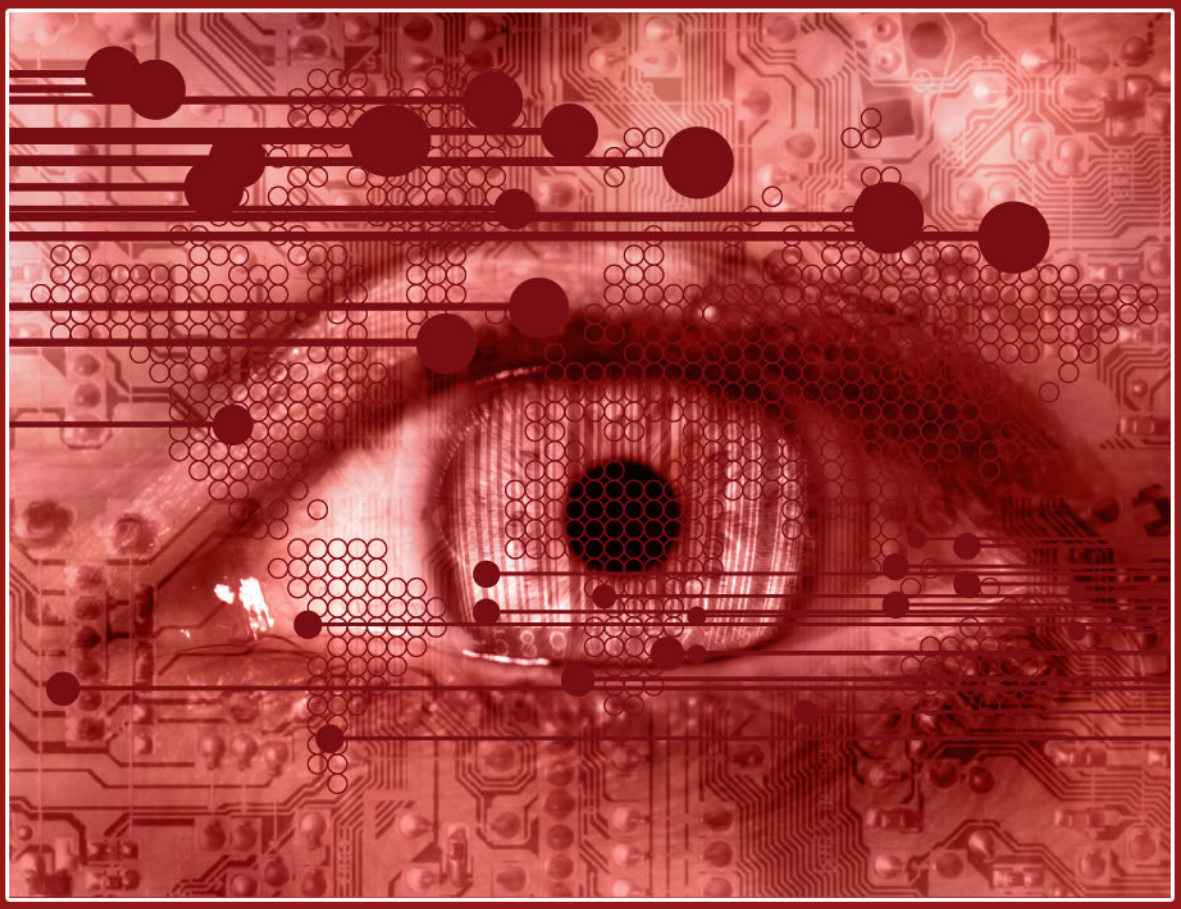
ICoFCS 2015

PROCEEDINGS OF THE NINETH INTERNATIONAL CONFERENCE ON

FORENSIC COMPUTER SCIENCE



www.icofcs.org

Proceedings of the Nineth International Conference on Forensic Computer Science - ICoFCS 2015
Brasília, Brazil, 2015, 113 pp. - Online ISBN 978-85-65069-10-6.

Address: SMPW 21 Conj. 02 Lote 08 Casa C – Parkway
CEP: 71.745-102 - Brasília/DF - Brazil
Email: presidente@cienciasforenses.org.br
Online ISBN 978-85-65069-10-6

TECHNICAL PROGRAM COMMITTEE (TPC)

GENERAL CHAIRS

Bruno Werneck Pinto Hoelz - Brazilian Federal Police, Brazil
João Paulo Carvalho Lustosa da Costa - University of Brasilia, Brazil
Jorge de Albuquerque Lambert- Brazilian Federal Police, Brazil

SESSION CHAIRS

Evandro Mário Lorens - Brazilian Federal Police, Brazil
Flavio Elias Gomes de Deus - University of Brasilia, Brazil
Juliano Zaiden Benvindo - University of Brasilia, Brazil
Paulo Quintiliano da Silva - Brazilian Federal Police, Brazil
Rafael Timóteo de Sousa Júnior - University of Brasilia, Brazil

REVIEWERS

Alex Sandro Roschildt Pinto - Federal University of Santa Catarina, Brazil
Amilton Soares Junior - Brazilian Federal Police, Brazil
Ana Cristina Azevedo Pontes de Carvalho - Mackenzie, Brazil
Bruno Gomes de Andrade - Brazilian Federal Police, Brazil
Bruno Werneck Pinto Hoelz - Brazilian Federal Police, Brazil
Carlos Alberto S. Lucietto - Brazilian Federal Police, Brazil
Daniel de Oliveira Cunha - Brazilian Federal Police, Brazil
Daniel França de Oliveira Melo - Brazilian Federal Police, Brazil
Díbio Leandro Borges - University of Brasilia, Brazil
Dominik Neudert-Schulz - Ilmenau University of Technology, Germany
Ebrahim Samer El Youssef - Federal University of Santa Catarina
Edison Pignaton de Freitas - Federal University of Rio Grande do Sul
Edna Dias Canedo - University of Brasilia, Brazil
Flavio Luis Vidal - University of Brasilia, Brazil
Frank Wilson Favero - Brazilian Federal Police, Brazil
Gustavo Guimaraes Parma - Brazilian Federal Police, Brazil
Gustavo Henrique Machado de Arruda - Brazilian Federal Police, Brazil
Harley Angelo de Moraes - Brazilian Federal Police, Brazil
Janine Zancanaro da Silva - Brazilian Federal Police, Brazil
Jayme Milanezi Junior – Brazilian Regulatory Agency, Brazil
José Rocha de Carvalho Filho - Brazilian Federal Police, Brazil
José Antônio Apolinário Junior - Military Institute of Engineering, Brazil
Karoll Haussler Carneiro Ramos - University of Brasilia, Brazil
Kefei Liu - Michigan University, USA
Luciano Lima Kuppens - Brazilian Federal Police, Brazil
Luiz Vinicius Gontijo Laborda Larrain - Brazilian Federal Police, Brazil
Michael Kirchhoff - Ilmenau University of Technology, Germany
Marco Antonio Marques Marinho- University of Brasilia, Brazil
Mariana Guimarães Pralon - Ilmenau University of Technology, Germany
Nerci Lino de Almeida - Brazilian Federal Police, Brazil
Paulo César Herrmann Wanner - Brazilian Federal Police, Brazil
Paulo Max Gil Innocencio Reis - Brazilian Federal Police, Brazil
Rafael de Oliveira Ribeiro - Brazilian Federal Police, Brazil
Ricardo Kehrle Miranda - University of Brasilia, Brazil
Rodrigo Gurgel Fernandes Távora - Brazilian Federal Police, Brazil
Samuel Machado Leal da Silva - Brazilian Army, Brazil
Sara Lais Rahal Lenharo - Brazilian Federal Police, Brazil
Stephan Haefner - Ilmenau University of Technology, Germany
Thiago Pereira de Brito Vieira – National Telecommunications Agency, Brazil
Willian Ferreira Giozza - University of Brasilia, Brazil

CONFERENCE SCOPE

Artificial Intelligence
Artificial Neural Network
Biometrics
Computer Crimes
Computer Forensics
Computer Forensics in Education
Computer Law
Computer Vision
Criminology
Cryptology
Digital Investigation
Image Analysis
Image Processing
Information Security
International Police Cooperation
Intrusion Prevention and Detection
Machine Learning
Management Issues
Network Security
Pattern Recognition
Secure Software Development
Semantic Web
Signal Processing
Simulation
Software Engineering

BEST PAPER AWARD

On this year, the “Best Paper Award” winner is the paper

Identificação do calibre de munições por meio da assinatura acústica dos estojos ejetados,

written by

Luiz Vinicius Gontijo Laborda Larrain
João Paulo Carvalho Lustosa da Costa
and Tadeu Junior Gross

The choice was made by the TPC members.

CONTENTS

FOREWORDS

The Technical Program Committee (TPC) of the 9th International Conference on Forensic Computer Science (ICoFCS) electronically releases the conference proceedings composed of fifteen selected publications on http://www.icofcs.org/2015/.

Less than 45 % of the papers were accepted for publication this year after a rigorous peer review selection. Moreover, the authors were required to present their results in the Forensic Academy track that took place at the Integrated Conference ICCyber / ICMedia 2015 held in Brasília from June 23rd, 2015 to the 25th. The complete program of the Integrated Conference is available at www.conferenciaintegrada.org.br.

The Brazilian Association of High Technology Experts (ABEAT) formally transfers the copyrights of the proceedings to the Brazilian Academy of Forensic Sciences (ABCF).

The TPC gratefully recognizes and acknowledges the inestimable support of ABCF, ABEAT, Technical-Scientific Board (DITEC) of the Federal Police Department (DPF) in Brazil and University of Brasilia (UnB).

DOI: 10.5769/C2015001 or http://dx.doi.org/10.5769/C2015001

Identificação do calibre de munições por meio da assinatura acústica dos estojos ejetados

Luiz Vinícius G. L. Larrain[1,2], João Paulo C. L. da Costa[1], Tadeu Junior Gross[2]

(1) Laboratório de Processamento de Sinais em Arranjos
Departamento de Engenharia Elétrica
Universidade de Brasília (UNB)
URL: http://www.redes.unb.br/lasp
(2) Gerência de Perícias em Áudio e Vídeo
Perícia Oficial e Identificação Técnica do Estado de Mato Grosso - POLITEC
URL: http://www.politec.mt.gov.br

***Resumo*— Gravações de disparos de armas de fogo podem ser cruciais para investigações forenses apesar de apresentarem vários desafios técnicos aos peritos. Propõe-se neste artigo uma abordagem com perspectiva diferente das tradicionais nesta área, focando o estudo nas assinaturas acústicas geradas pelas ações mecânicas intrínsecas ao funcionamento de armas de fogo, em específico da queda do estojo ejetado após o disparo. É proposta a modelagem do comportamento acústico do estojo como um tubo fechado, ou seja, com a presença de frequências ressonantes. O modelo é validado para quatro calibres de pistolas, verificando o seu potencial para a determinação de calibres a partir da assinatura acústica gerada pelo contato dos estojos ejetados, ao atingir superfícies rígidas.**

Palavras-Chave*— *Acústica Forense, Balística Forense, Gravações de Tiros, Identificação de armas de fogo.

***Abstract*— Audio gunshot recordings can be very helpful for crime scene investigation but also have many challenges for the forensic experts. In this article we proposed an untraditional perspective to solve this problems. Our focus is not in the muzzle blast or shock wave but in a specific mechanical action of the firearms: the ejection of spent cartridges. We propose an acoustical model to the casing inspired in closed tubes in physics acoustical. After we validated this model to four pistols caliber's and tested if the acoustical signature generated when the casing touch any rigid surface can be used to identify the ammunition caliber's in terms of audio forensics.**

Keywords*— *Audio Forensics, Forensic Ballistcs, Gunshot Recordings, Firearm identification.

I. INTRODUÇÃO

A difusão dos dispositivos portáteis de gravação, tais como os celulares *smartphones*, e suas integrações com as diversas redes sociais existentes, geram um volume exacerbado de registros audiovisuais contendo, em diversas ocasiões, eventos relacionados a prática de ações delituosas, incluindo o uso de armas de fogo.

A análise forense acústica demonstra-se uma importante e promissora área das ciências forenses para a investigação de crimes com emprego de armas de fogo. Isto porque as gravações desta natureza possuem informações preciosas com potencial para elucidar questionamentos, tais como: o reconhecimento do estrondo como um tiro real de arma de fogo, a identificação do calibre da arma de fogo empregada, a quantidade de armas envolvidas e a ordem dos autores dos disparos.

Em meio às abordagens para estes problemas encontram-se a modelagem e a caracterização acústica de disparos de armas de fogo [1-3], a detecção de tiros em ambientes ruidosos [4], a influência do ângulo e direção na gravação de tiros [5], a identificação do calibre da arma através das gravações de tiros [6], localização de atirador *sniper* [7,8] e a identificação de armas de fogo através de imagens rotacionadas do cartucho [9].

Para efetuar as análises o Perito deve compreender as distintas componentes emissoras de sons, possíveis de serem encontradas em gravações contendo disparos de armas de fogo. Dentre estas componentes encontram-se os sons oriundos da explosão provocada para a expulsão do projétil através do cano da arma; das ações mecânicas intrínsecas ao funcionamento desta, como o acionamento do gatilho e cão, a expulsão de estojos e recarregamentos de munição; da onda de choque provocada por projéteis supersônicos; e os sinais decorrentes dos efeitos de vibração em superfícies sólidas além do próprio solo [2].

Todas as abordagens supracitadas focam na abordagem acústica do tiro e seus efeitos na propagação. Este trabalho possui uma perspectiva diferenciada, focada no estudo de eventos acústicos das ações mecânicas, em específico para a queda do estojo ejetado. Apesar de uma ampla pesquisa bibliográfica na literatura relacionada, não foi identificada abordagem similar para o tratamento deste problema.

É proposto neste artigo um estudo acerca da natureza acústica do som produzido pelo contato dos estojos ejetados ao atingir o solo ou superfícies rígidas e o potencial destes eventos acústicos para a identificação do calibre de armas de fogo. Apresentam-se ainda os resultados experimentais obtidos para quatro tipos de calibres utilizados por pistolas.

Este artigo possui cinco seções, incluindo esta introdução. Inicialmente na Seção II são apresentados os aspectos teóricos e a modelagem física para o problema proposto. Em seguida, a Seção III aborda os ensaios realizados e os resultados obtidos. Apresentam-se na Seção IV as análises e considerações, comparando o modelo teórico adotado e os resultados

experimentais obtidos e, finalmente, as conclusões são apresentadas na Seção V.

II. MODELAGEM FÍSICA DO PROBLEMA

Na Seção II.A apresenta-se uma introdução aos mecanismos de expulsão do estojo, na Seção II.B são abordadas as características físicas e respectivas dimensões dos calibres delimitados neste artigo e na Seção II.C é proposta uma modelagem do comportamento acústico dos calibres delimitados.

A. Mecanismos de Expulsão do Estojo

As armas de fogo, quanto ao funcionamento, são classificadas em armas de tiro unitário e armas de repetição. As armas de tiro unitário são aquelas que comportam carga para um único disparo, mesmo disparando múltiplos projéteis simultaneamente, tendo portanto seu carregamento manual. Como exemplo, pode-se citar as espingardas de um cano, pistolas de tiro unitário e garruchas. Já as armas de repetição podem ser classificadas em não automáticas, semiautomáticas e automáticas [10].

Ainda de acordo com a literatura correlata [10], as armas de repetição não automáticas são aquelas cujos mecanismos de repetição e disparo dependem exclusivamente da força muscular do atirador, como os revólveres e a maioria das carabinas. Nas armas de repetição semiautomáticas o esforço do atirador é necessário apenas para o acionamento do disparo, aproveitando-se os gases da combustão para o acionamento automático do mecanismo de repetição, sendo exemplo desta categoria a maioria das pistolas. Para as armas de repetição automática tanto o mecanismo de disparo quanto o de repetição são acionados pela força expansiva dos gases, sendo possível a produção de tiro não só intermitente mas também contínuo, em rajada, como nas submetralhadoras e fuzis.

As armas de fogo de funcionamento por repetição semiautomáticas e automáticas, como as pistolas, submetralhadoras e fuzis por exemplo, apresentam mecanismos automáticos para a expulsão do estojo logo após o respectivo cartucho ser deflagrado, conforme ilustrado na Figura 1.

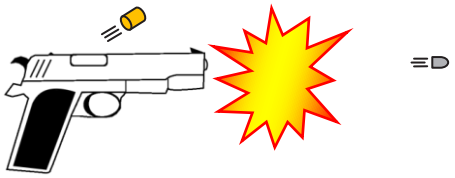


Fig. 1. Ilustração do mecanismo de expulsão do estojo em uma pistola

Tal mecanismo tem como objetivo o recarregamento automático da munição, permitindo que a arma esteja apta para efetuar outro disparo no menor tempo possível.

Desta forma verifica-se o potencial de aplicação da análise acústica da queda dos estojos preferencialmente para as armas de repetição semiautomáticas e automáticas, uma vez que a expulsão do estojo independe da ação voluntária do atirador, estando condicionada portanto exclusivamente ao disparo da arma de fogo. Por outro lado, esta abordagem também demonstra-se válida para determinadas armas de repetição não automática, condicionada à ação voluntária do atirador para a extração do estojo, como no caso de espingardas de repetição *pump action* ou na própria substituição dos cartuchos deflagrados no tambor do revólver.

B. Características Físicas do Estojo

O cartucho é composto, de maneira geral, por quatro componentes: o projétil, a carga propelente, o estojo e a espoleta, conforme apresentado na Figura 2. Para a ocorrência do tiro, a arma percute a espoleta, que provoca a queima da carga propelente, gerando um grande volume de gases que expulsa o projétil através do cano da arma. O estojo é o invólucro que permite a união mecânica de todos estes componentes, em uma única peça, facilitando o manejo da arma e a redução do intervalo de cada disparo.

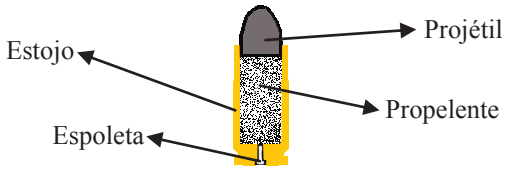


Fig. 2. Componentes presentes em um cartucho de fogo central

Atualmente a maioria dos estojos são construídos em metais não ferrosos, principalmente o latão, composto por liga de cobre e zinco, mas ainda assim é possível encontrar estojos de diversos tipos de materiais como alumínio, plástico, papelão, entre outros.

A forma e as dimensões do estojo indicam a configuração interna da câmara e consequentemente o calibre nominal da arma Apresenta-se na Figura 3 a classificação presente na literatura quanto ao formato dos estojos disponíveis. Cabe ressaltar ainda que, na prática, até para os estojos cilíndricos é possível a ocorrência de um pequeno afunilamento para facilitar o processo de extração.

Os estojos do tipo cilíndricos e cônicos são empregados predominantemente por revólveres, pistolas e submetralhadoras de maneira geral. Os estojos do tipo garrafa apresentam um estrangulamento no intuito de aumentar a quantidade de propelente para projéteis menores sendo utilizado por carabinas, rifles, fuzis e, ainda, em alguns tipos de pistolas, geralmente em armas que disparam com maior quantidade de energia.

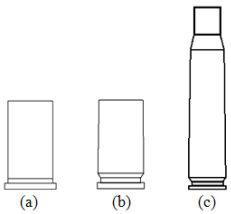


Fig. 3. Formatos de estojos: cilíndricos (a), cônicos (b) e garrafa (c)

Neste trabalho, o estudo foi delimitado para os calibres indicados na Tabela I, todos utilizados por pistolas.

TABELA I. MODELOS E DIMENSÕES DOS ESTOJOS [11]

Marca/ Calibre	Tipo	Comprimento Total (mm)	Diâmetro Boca (mm)	Diâmetro Corpo (mm)
CBC 9 MM LUGER	Cônico	19	9,55	9,84
CBC .40 S&W	Cilíndrico	21,56	10,69	10,74
CBC 32 AUTO	Cilíndrico	17,16	8,40	8,50
CBC 380 AUTO	Cilíndrico	17,27	9,45	9,45

C. *Frequências de Ressonância em Tubos Fechados*

Segundo a teoria física de ondas, quando as ondas sonoras se propagam no interior de um tubo, estas são refletidas nas extremidades, ainda que esta esteja aberta. Para certos comprimentos de onda, a superposição das ondas que se propagam nos tubos em sentidos opostos, produz uma onda estacionária. Os comprimentos de ondas para os quais estes fenômenos acontecem correspondem às frequências de ressonância do tubo.

A Figura 4 exibe algumas ondas sonoras estacionárias para um tubo com uma das extremidades fechadas e a outra aberta. Neste caso haverá um ponto de deslocamento máximo na extremidade aberta e um nó na extremidade fechada.

Para o cálculo com precisão do comprimento de onda da primeira harmônica, a correção final deve ser considerada. Esta correção final pode ser obtida através do produto do diâmetro do tubo por uma constante [12]. Através de pesquisa bibliográfica, foi observado a aplicação de coeficientes de correção final para tubos com uma extremidade fechado na faixa de 0,3 a 0,6. Para a modelagem deste problema em específico, foi adotado um coeficiente de correção final de 0,4.

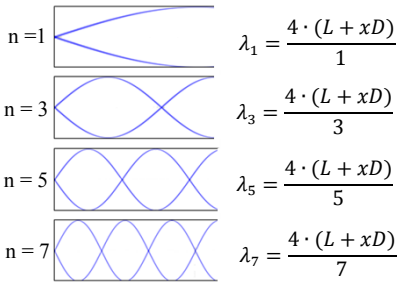


Fig. 4. Ondas estacionárias em tubos com apenas uma extremidade livre

Para a estimativa das frequências de ressonância teóricas, foi considerado o estojo como sendo um tubo cilíndrico ideal, ou seja, perfeitamente cilíndrico e homogêneo, com apenas uma das extremidades fechada. As frequências de ressonância podem ser estimadas por:

$$f_n = \frac{n \cdot v}{4 \cdot (L + 0{,}4D)} \qquad \text{para n = 1, 3, 5...} \qquad (1)$$

em que v, L e D são a velocidade do som, o comprimento total e o diâmetro interno do tubo, respectivamente.

Os resultados obtidos das frequências de ressonância de cada um dos calibres estudados são apresentados na Tabela II.

TABELA II. FREQUÊNCIAS TEÓRICAS DOS ESTOJOS TESTADOS

MARCA/CALIBRE	F_1 (Hz)	F_3 (Hz)	F_5 (Hz)
CBC 9 MM LUGER	3724	11172	18620
CBC .40 S&W	3289	9867	16445
CBC 32 AUTO	4142	12426	20710
CBC 380 AUTO	4038	12114	20190

III. ENSAIOS REALIZADOS

Na Seção III.A é detalhado o método empregado para a realização dos ensaios e na Seção III.B são apresentados os resultados obtidos.

A. *Método Utilizado*

Analisando as características físicas dos estojos, abordados na Seção II, observa-se semelhança aparente com o modelo de tubos fechados em uma extremidade, o que sugere que estes possam gerar sons em frequências de ressonância específicas.

Para verificação desta hipótese foram realizados ensaios em uma sala fechada de ruído ambiente de baixa intensidade. Os estojos deflagrados, foram soltos individualmente a aproximadamente 1,60 m do piso, de revestimento cerâmico, no interior de um círculo de raio de 1 m com o microfone ao centro, conforme esquematizado na Figura 4.

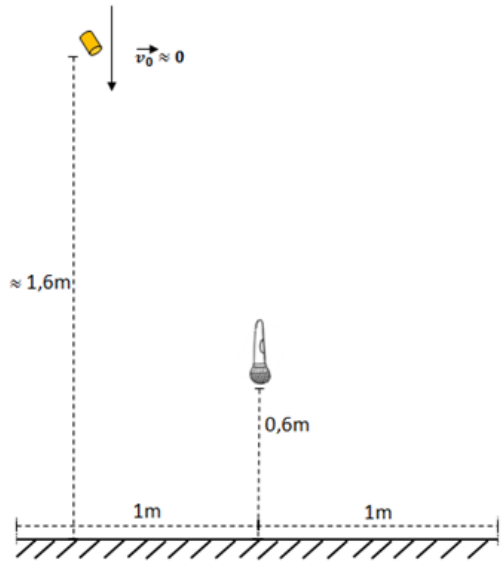


Fig. 5. Experimento realizado para coleta dos áudios

Foram utilizados 5 estojos distintos para cada um dos calibres estudados, com exceção do CBC 9 mm LUGER que foram utilizados apenas 3. A série de lançamento para cada um destes estojos foi reproduzida 50 vezes.

Os áudios foram capturados em formato WAV/PCM, sem compressão, com taxa de amostragem 48 kHz, 16 bits e em um único canal (mono).

B. *Resultados Obtidos*

Conforme esperado, ao analisar o espectrograma das gravações efetuadas, foi observada a presença de componentes frequenciais específicas em todos os calibres e ocorrência constante ao longo da série de lançamentos. Verificou-se ainda a predominância significativa de uma componente, com maior potência em relação às demais, conforme demonstrado no espectrograma da Figura 6, para um intervalo da série de lançamento do estojo número 1 CBC 380 AUTO.

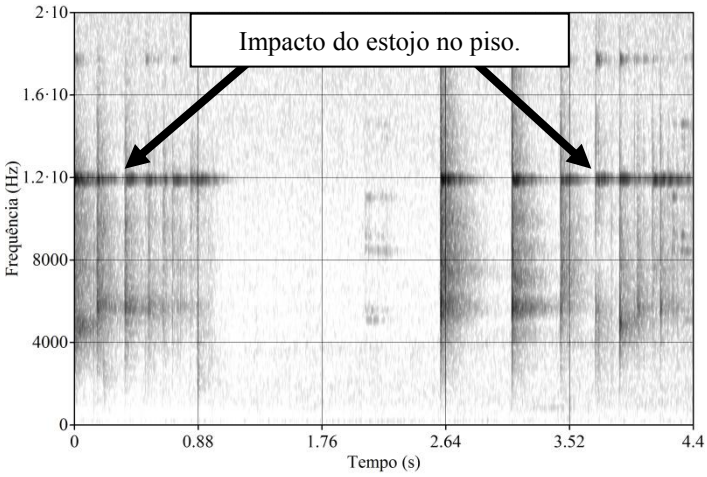


Fig. 6. Frequência de ressonância de maior potência para o estojo número 1 CBC 380 AUTO

Em seguida foi calculado o espectro médio de longo termo, do inglês *long-term average spectrum* (LTA), para cada uma das séries de lançamentos realizadas, e os resultados obtidos foram normalizados e agrupados por calibre.

Os espectros LTA obtidos para cada um dos calibres são apresentados nos gráficos das Figuras 7 a 10 com evidência para as frequências de ressonância de maior potência. Foi verificado no espectro LTA que as frequências de maior potência correspondem a segunda componente do modelo teórico (F_3).

Na Figura 7 têm-se os espectros LTA dos estojos CBC 9 mm LUGER. Verifica-se a ocorrência dos picos de frequências no intervalo de 11150 Hz a 1250 Hz.

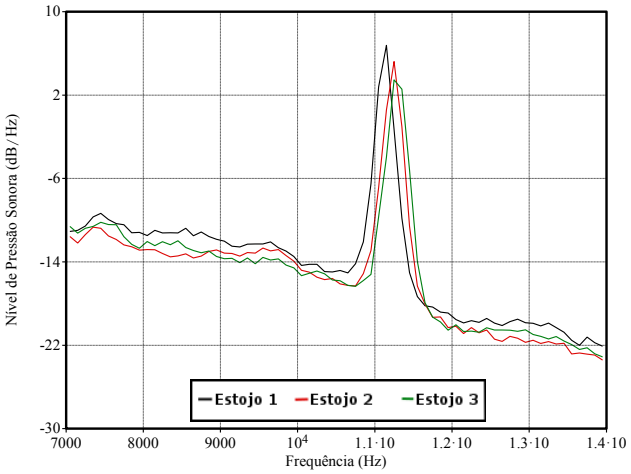


Fig. 7. Espectros LTAs das gravações dos estojos CBC 9 mm LUGER

Na Figura 8 têm-se os espectros LTA dos estojos CBC 32 AUTO. Verifica-se a ocorrência dos picos de frequências no intervalo de 12050 Hz a 12250 Hz.

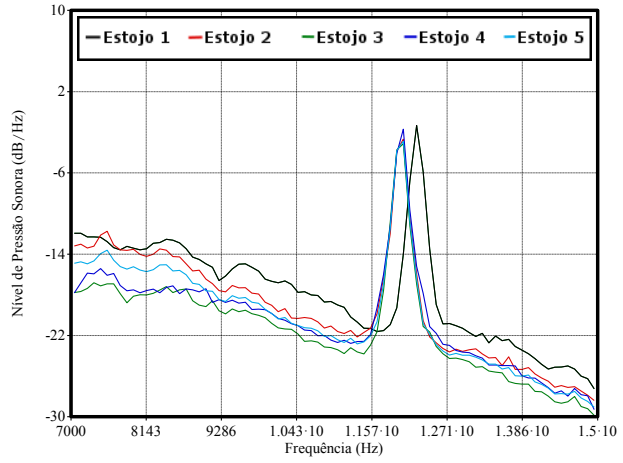


Fig. 8. Espectros LTAs das gravações dos estojos CBC 32 AUTO

Na Figura 9 têm-se os espectros LTA dos estojos CBC .40 S&W. Verifica-se a ocorrência dos picos de frequências no intervalo de 8050 Hz a 8450 Hz.

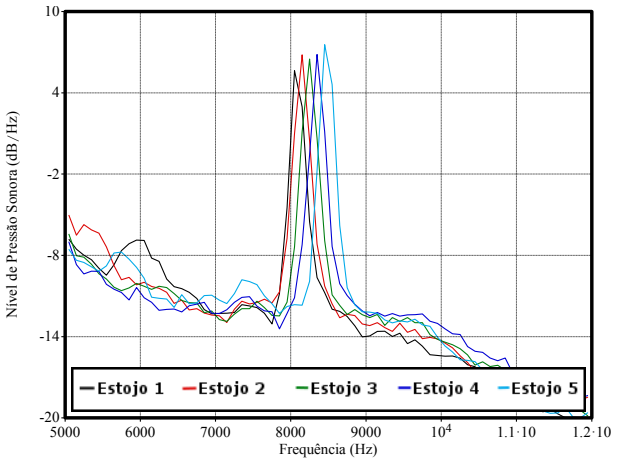


Fig. 9. Espectros LTAs das gravações dos estojos CBC .40 S&W

Na Figura 10 têm-se os espectros LTA dos estojos CBC 380 AUTO. Verifica-se a ocorrência dos picos de frequências no intervalo de 11650 Hz a 11950 Hz.

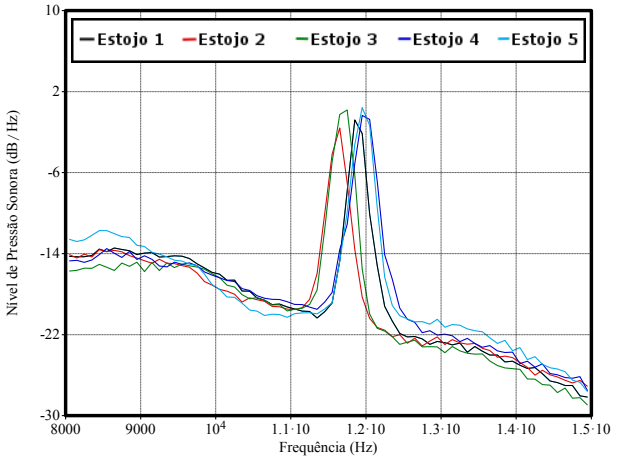


Fig. 10. Espectros LTAs das gravações dos estojos CBC 380 AUTO

IV. VALIDAÇÃO E CONSIDERAÇÕES

Os resultados obtidos através dos ensaios demonstraram a existência de convergência entre os distintos estojos de mesmo material e calibre. Também verificou-se diferenças significativas para alguns calibres distintos, conforme demonstrado na Figura 11, que apresenta a sobreposição normalizada das médias dos LTAs obtidos para os calibres delimitados neste estudo.

A exceção para os calibres estudados ocorreu, exclusivamente, para o CBC 32 AUTO e CBC 380 AUTO, em que houve uma clara sobreposição das médias dos espectros LTAs. A aproximação dos resultados obtidos para os calibres CBC 32 AUTO e CBC 380 AUTO era esperada em decorrência dos tamanhos destes estojos serem praticamente os mesmos conforme apresentado na Tabela I.

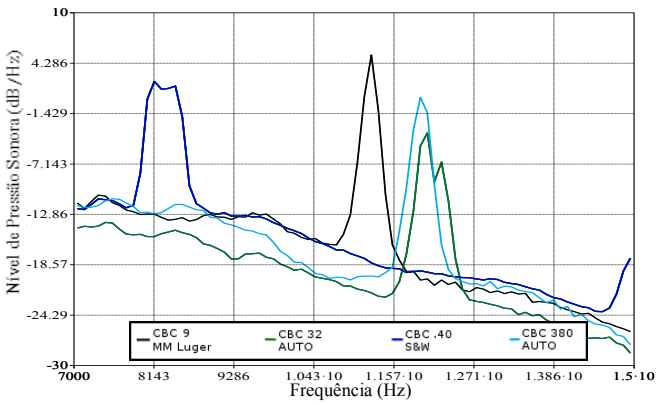


Fig. 11. Sobreposição das médias dos espectros LTAS dos calibres testados

Para validação do modelo proposto, foi calculada a raiz quadrada do erro quadrático médio normalizado, do inglês *normalized root-mean-square error* (NRMSE), definida pela expressão:

$$\mathrm{NRMSE} = \left[\frac{1}{N}\sum_{i=1}^{N}\left(\mathrm{F}_3 - \hat{\mathrm{F}}_{3\mathrm{i}}\right)^2\right]^{0,5} \cdot \frac{1}{\overline{\mathrm{F}}_3} \qquad (2)$$

em que F_3 e $\hat{\mathrm{F}}_3$ são as frequências observadas e teóricas respectivamente e $\overline{\mathrm{F}}_3$ é a média das frequências observadas.

Observa-se que os resultados práticos apresentaram razoável aderência ao modelo teórico proposto, ainda que simplificado, conforme evidenciado na Tabela III. A exceção, neste caso, ocorreu para o calibre CBC .40 S&W, que obteve o maior NRMSE em relação a modelagem física proposta.

TABELA III. VALIDAÇÃO DO MODELO PROPOSTO

MARCA/ CALIBRE	Faixa Frequências Observadas (Hz)	Valor Teórico $\hat{\mathrm{F}}_3$ (Hz)	NRMSE
CBC 9 MM LUGER	11150 a 11250	11172	0,0058
CBC .40 S&W	8050 a 8450	9867	0,1967
CBC 32 AUTO	12050 a 12250	12426	0,0286
CBC 380 AUTO	11650 a 11950	12114	0,0296

Dentre as vantagens da abordagem proposta, encontram-se a dispensabilidade de inúmeras variáveis citadas na literatura para a interpretação e análise satisfatória do tiro a partir de uma gravação, tais como: a disposição de objetos, obstáculos e natureza do próprio ambiente de ocorrência dos disparos, se aberto ou fechado, informações acerca da munição empregada, efeitos do vento e umidade, posição relativa e ângulo de direção do atirador em relação ao equipamento gravador [2,5]. Estes parâmetros em muitas ocasiões são desconhecidos pelo perito.

Como na abordagem proposta as frequências analisadas dependem exclusivamente do material e dimensões físicas do estojo, a constatação deste evento acústico, ainda que isoladamente, já possibilitaria ao perito estabelecer inferências independentemente de informações do local de crime.

Dentre as desvantagens do método encontra-se a limitação de sua aplicação, sendo predominantemente válido para as armas de fogo que possuam recarregamento automático ou semiautomático, ou seja, que expulsem involuntariamente o estojo, ou ainda para determinadas armas de repetição não automáticas, mediante a ação voluntária do atirador.

Além disto, ao contrário do tiro que se propaga por longas áreas, para a gravação deste evento acústico é imprescindível a proximidade do microfone em relação ao local dos disparos. Ressalta-se que para os eventos acústicos gravados nos ensaios, foram observados picos superiores a 6 dB/Hz em determinadas frequências, o que torna a viabilidade de aplicação desta abordagem bastante promissora.

Além do requisito de proximidade, considerando as faixas de frequência constatadas nos ensaios, para os calibres delimitados, o equipamento gravador deve suportar banda de no mínimo 12,5 kHz para que a gravação viabilize o emprego desta técnica. Esta limitação não é propriamente um problema muito grave para gravações presenciais de modo geral, uma vez que a grande maioria dos atuais equipamentos portáteis de gravação, incluindo celulares, já suportam bandas superiores a esta.

V. CONCLUSÕES

Este artigo confirmou a hipótese da presença de frequências de ressonância quando os estojos ejetados, após a ocorrência de um disparo de arma de fogo, atingem uma superfície rígida. Verificou-se ainda que ao menos uma frequência, para todos os calibres estudados, apresenta valores mais significativos de potência, ou seja, mais resistente ao distanciamento do microfone em relação ao atirador.

A modelagem física do problema, ainda que os estojos notoriamente não se tratem de cilindros ideais apresentou resultados bastante aderente aos obtidos nos ensaios práticos para todos os calibres testados.

Desta forma, a abordagem proposta foi validada satisfatoriamente para os quatro tipos de calibres de pistola, demonstrando-se portanto uma alternativa promissora para a atribuição de calibres de arma de fogo, a partir da assinatura acústica gerada pelo contato do estojo ejetado com superfícies rígidas. Ademais esta proposta mostra-se complementar as abordagens tradicionais já existentes sendo, portanto, mais um recurso de análise disponível ao perito.

Para trabalhos futuros torna-se necessário expandir os estudos e ensaios, abordando os demais calibres não contemplados neste artigo, inclusive de outros tipos de armas, especialmente os estojos do tipo garrafa. Recomenda-se ainda a verificação das máximas distâncias possíveis entre o transdutor e o atirador, que ainda assim permitam o registro da assinatura acústica do estojo e consequente aplicação do método, além de maiores estudos quanto à robustez desta técnica para artefatos de compressão, ambientes ruidosos, dentre outros fatores de degradação frequentes à casuística forense.

Outra sugestão para trabalhos futuros é o estudo acerca de outros sons intrínsecos as ações mecânicas das armas de fogo, como por exemplo: os sons gerados pelos sistemas de alimentação e repetição e da movimentação da arma pelo atirador.

AGRADECIMENTOS

Os autores agradecem às Agências brasileiras de pesquisa e inovação CAPES (Projeto FORTE, Edital CAPES Ciências Forenses 25/2014) e FINEP (Convênio RENASIC/PROTO 01.12.0555.00), pelo suporte a este trabalho.

Agradecem ainda a Diretoria Geral da Perícia Oficial e Identificação Técnica (POLITEC) do Estado de Mato Grosso, pelo apoio para a qualificação dos peritos deste órgão.

REFERÊNCIAS

[1] Maher, R.C. "Modeling and signal processing of acoustic gunshot recordings", em: Anais do IEEE Signal Processing Society 12th Digital Signal Processing Workshop, Jackson Lake, WY, pp. 257-261, 2006.

[2] Maher, R.C. "Acoustical characterization of gunshots", em: Anais do IEEE SAFE 2007: Workshop on Signal Processing Applications for Public Security and Forensics, Washington, DC, pp. 109-113, 2007.

[3] Maher, R.C., Shaw, S.R. "Deciphering gunshot recordings", em: Anais do AES 33rd Conference Audio Forensics - Theory and Practice, Denver, CO, 2008.

[4] Freire, I. L. e Apolinario Jr., J. A. "Gunshot detection in noisy environments", em: Anais do 7th International Telecommunications Symposium, Manaus, Brasil, 2010.

[5] Maher, R. C. e Shaw, Steven R. "Directional aspects of forensic gunshot recording", em: Anais do AES 39th International Conference Audio Forensics - Practices and Challenges, Hillerod, Dinamarca, 2010.

[6] Thumwarin, P., Matsuura, T., Yakoompai, K. "Audio forensics from gunshot for firearm identification", em: Anais do IEEE 4th Joint International Conference on Information and Communication Technology, Electronic and Electrical Engineering, Tailândia, pp. 1-4, 2014.

[7] Freire, I. L., e Apolinario, J. A. "GCC-based DoA estimation of overlapping muzzleblast and shockwave components of gunshot signals", em: Anais do IEEE Second Latin American Symposium on Circuits and Systems (LASCAS), Bogota, pp. 1-4, 2011.

[8] Calderon P., Manolo D., e Apolinario J. A. "Shooter Localization based on DoA Estimation of Gunshot Signals and Digital Map Information", em: IEEE (Revista IEEE America Latina) Latin America Transactions, 13.2: 441-447, 2015.

[9] Thumwarin, P.; Prasit, C.; e Matsuura, T. "Firearm identification based on rotation invariant feature of cartridge case", em: Anais do IEEE SICE Annual Conference, Tóquio, Japão, pp. 45-49, 2008.

[10] Tocchetto, D. "Balística Forense Aspectos Técnicos e Jurídicos", 7 ed. Campinas: Millenium Editora, 2013, p. 30-32.

[11] Sítio: "http://www.municion.org/". Acessado em 30 de abril de 2015.

[12] Boelkes, T. e Hoffmann, I. "Pipe Diameter and End Correction of a Resonant Standing Wave", International Scholastic Journal of Science Journal of Physics, Vol. 5, Iss. 1, 2011.

DOI: 10.5769/C2015002 or http://dx.doi.org/10.5769/C2015002

Proteção da Prova Documental Impressa e Digitalizada com a Utilização de *Watermarking*

Felippe Pires Ferreira

***Resumo*—O artigo propõe um método para disponibilização de documentos sigilosos durante o inquérito policial, que pode ser estendido a outros documentos, introduzindo o elemento de segurança conhecido como *watermarking*. Este elemento permitirá vincular uma cópia de documento a seu destinatário inicial, e em casos de vazamento de informação permitirá identificar a origem da cópia. O método possibilitaria incluir uma *watermark* em um documento eletrônico editável e recuperá-la em documentos impressos ou digitalizados, bastando apenas um fragmento do texto. O método é baseado na semelhança entre caracteres de diferentes fontes de texto, os quais serão utilizados para criação de uma codificação identificadora da origem do documento.**

Palavras-Chave*—*Marca d'água, Cópia de Documento, Cópia Impressa, Digitalização, Inquérito Policial, Fonte de Texto.

Abstract—**This article proposes a method for available classified documents during the police investigation, which can be extended to other documents, introducing the security element known as watermarking. This element will link a document copy to its initial recipient, and in cases of information leakage will identify the origin of the copy. The method would allow include a watermark in an editable electronic document and retrieve it in printed or scanned documents, just by a fragment of the text. The method is based on the similarity between characters of different text fonts, which will be used to create a code identifying the origin of the document.**

Keywords*—*Watermarking, Document Copy, Hard-Copy, Scan, Police Investigation, Text Font.

INTRODUÇÃO

A difusão da informação com o auxílio dos avanços da tecnologia foi inicialmente aceita como grande revolução na comunicação. A facilidade de se encontrar um documento editado por uma pessoa a quilômetros de distância permite rapidez e dinamismo ao processo de comunicação. Entretanto, a crescente difusão de conhecimento também proporciona a prática de publicação de material não autorizado através da Internet. Diversos livros, artigos e documentos de trabalho também foram objeto dessa popularização e disseminação da prática de compartilhamento de documentos on-line [1].

Neste novo cenário, surgiu a necessidade de criação de um mecanismo que protegesse a produção intelectual dos autores. Entretanto, essa proteção precisa ser robusta o suficiente para que não seja removível, capaz de ser recuperável, além de identificar o material. Dessa necessidade foi criado o conceito de *watermarking*, ou termo traduzido **marca d'água** [1].

Watermarking é o processo de inserir informações sobre o objeto no próprio objeto, e esta informação pode ser extraída posteriormente para ser verificada [2]. Diferente da esteganografia, a *watermarking* não procura ser totalmente eficiente contra detecções, mas visa identificar um material, e impossibilitar sua remoção ou alteração [3].

As *watermarkings* possuem diferentes aplicações [4] [5]. Elas permitem a **identificação do proprietário**, protegendo assim a distribuição de materiais como músicas e livros com sua utilização. Muitas vezes a marca d'água não está oculta e apresenta informações sobre o proprietário do material. A **autenticidade** do material pode ser verificada e, com outros mecanismos de segurança como a criptografia, contribui para a confirmação da origem da informação e sua veracidade. Caso ocorra alguma modificação do material, a verificação da **integridade** da marca d'água pode indicar que houve manipulação não autorizada sobre este. É possível realizar o controle de cópias personalizando as marcas d'água de acordo com o destinatário do material, possibilitando o rastreio do material em caso de vazamentos.

A literatura define que a *watermarking* tem propriedades que devem estar presentes no mecanismo quando inserido em um objeto, a fim de torná-la adequada para utilização [4]. A *watermarking* deve ser **robusta**, deste modo ser resistente a manipulação do material e ainda permanecer neste. Deve ser **não perceptível** a visão humana, sendo visível apenas durante processos de extração da informação. Ser **segura**, assim apenas o proprietário do material poderá recuperá-la, alterá-la ou removê-la. E por fim, deve ser capaz de armazenar informação/mensagem em um objeto [1] [6].

Outro aspecto são os diferentes suportes/materiais em que podem ser inseridas. De acordo com o tipo de arquivo de mídia são utilizadas técnicas diferentes. É possível utilizá-la em arquivos de texto, imagens, áudio e vídeo. Cada arquivo possui a sua peculiaridade, mas todos devem buscar as propriedades que tornam a *watermarking* resiliente.

O trabalho objetiva desenvolver um algoritmo para inclusão de marcas d'água em documentos eletrônicos textuais que permanecerá no documento mesmo após sua impressão, sendo passível de recuperação em processos de digitalização ou fotografia do material. Para atingir tal objetivo, o trabalho faz uso de manipulação de fontes de caracteres para inserção de um código identificador nos documentos. Os documentos textuais devem permitir edição, não sendo aplicável a imagem de documentos.

Na seção II serão apresentadas as metodologias de utilização de *watermarking* em documentos textuais, com ênfase na metodologia Baseada em Imagem, em que a técnica proposta neste trabalho se insere. Na seção III será ilustrado o

contexto em que se deseja aplicar a marca d'água, no caso em documentos que compõem o inquérito policial. A seção IV descreverá alguns trabalhos que serviram de direcionamento para construção da técnica do artigo. A seção V descreverá a técnica proposta através de esquemas e algoritmos, ressaltando as condições em que a técnica é aplicada. Na seção VI serão apresentados os resultados obtidos com a inserção de *watermark* de tamanhos diferentes, digitalizações de texto com diferentes configurações e a qualidade de duas imagens registradas por câmeras de *smartphones*, e posteriormente, é realizada a análise dos resultados, destacando as principais diferenças dos textos com variações nos tamanhos da *watermark* e os suportes em que se encontram, além da comparação entre a técnica proposta e métodos presentes na literatura. Por fim, a seção VII apresenta a conclusão da pesquisa e expectativas para trabalhos futuros.

WATERMARKING EM DOCUMENTOS TEXTUAIS

Além de aplicação em arquivos de imagens, áudios e vídeos, outra aplicação deste elemento de segurança são os documentos textuais. Diferentes técnicas de aplicação de *watermarks* foram criadas e agrupadas em quatro principais categorias [7]:

a. Baseada em Imagem - Nesta técnica o texto é tratado como imagem, sendo comparados aspectos visuais da composição do texto, como: alinhamento, espaçamento, caracteres, etc [8];

b. Sintática - Consiste em utilizar aspectos característicos de uma linguagem como substantivos, verbos, artigos, proposições, etc. Assim construindo uma árvore sintática que por sua vez será utilizada para inserção do elemento de segurança [9];

c. Semântica - Permite realizar mudanças de palavras por sinônimos ou utilizar abreviações e acrônicos para construir a marca d'água [10];

d. Estrutural - Utiliza características próprias da linguagem como a ocorrência de letras duplas ou preposições. O texto não é alterado, mas suas características são utilizadas na criação da marca d'água [8].

A pesquisa é classificada como Baseada em Imagem, porque utiliza aspectos visuais do texto, mais especificamente dos caracteres, para inserção e recuperação da *watermark*. Dentro dessa categoria podemos listar três técnicas [10]:

a. *Line Shifting* - as linhas do texto são deslocadas verticalmente, para baixo ou para cima, de tal forma que seja pouco perceptível aos olhos humanos. As linhas ímpares não são alteradas para que funcionem como linhas de controle. Entretanto, para recuperação da *watermark* , é necessário o documento original para fins de comparação;

b. *Word-Shift Coding* - deslocamento horizontal das palavras ou linhas de acordo com a *watermark* que se pretende inserir. As palavras são classificadas em grupos, nos quais os grupos pares variam de acordo com a *watermark* e os ímpares funcionam como controle e comparação para os deslocamentos. As palavras das extremidades não podem ser alteradas para que seja preservado o alinhamento. Assim como a técnica anterior, é necessário o documento original para recuperar a informação [11];

c. *Feature Coding* - utiliza variações das características de palavras ou letras como tamanho ou espaçamento para criação da *watermark*. Também faz uso do texto original para recuperação da informação.

As duas primeiras técnicas também precisam de um texto maior que a terceira técnica, uma vez que as alterações que aquelas proporcionam são baseadas nas linhas, enquanto esta é baseada em palavras ou letras.

APLICAÇÃO DO CONCEITO DE WATERMARKING EM DOCUMENTOS DO INQUÉRITO POLICIAL

O processo inquisitorial é marcado pelo seu sigilo durante a investigação e pela quantidade de provas produzidas que irão fundamentar a investigação e o futuro processo judicial. Apesar de sigiloso, algumas peças produzidas no inquérito policial são acessadas por personalidades que compõem esse trâmite, como advogados, policiais, promotores, juízes, entre outros. Esse acesso sem um controle eficaz pode gerar vazamento de informações e comprometer a própria investigação.

Pela facilidade na disseminação da informação, a detecção do responsável pelo vazamento se torna uma tarefa complexa, porque diferentes cópias são distribuídas e ainda não há maturidade suficiente nos processos de controle de cópias que possibilitem o rastreio da informação.

A definição de uma política de controle de cópias possibilita o rastreio da informação e a responsabilização do possível agente causador do vazamento. Uma das medidas seria a introdução de *watermarking* na produção de cópias de documentos, imagens e gravações, a fim de garantir a autenticidade, integridade e controle de cópia [5].

No entanto, o acesso ao material que foi alvo de vazamento não está disponível às autoridades, apenas o material original, porque são divulgados em veículos de comunicação como jornais e Internet, portanto os arquivos eletrônicos não estão à disposição para que sejam analisados. Logo, é preciso criar uma *watermarking* que possa ser recuperável nestes cenários.

TRABALHOS CORRELATOS

Alguns trabalhos utilizavam aspectos estruturais do texto ou dos caracteres para criarem as marcas d'águas. Diferentes abordagens para criação de *watermark* relacionados às características do texto foram encontradas:

a. *Watermark* criado através das características específicas dos documentos produzidos no software *Microsoft Word*. Cada parágrafo, palavra ou letra são tratados como objetos, possuindo propriedades, as quais possuem atributos especiais em que são possíveis esconder informações. Desta forma, permite que o texto possa ser distribuído pela Internet, mas ainda manter o mecanismo de controle de cópia dentro do documento [12];

b. *Watermarking* baseado na frequência das letras em sentenças escolhidas. Para cada sentença é criada um código de letras que será concatenado com outros códigos para criação da marca d'água. Para revelar o identificador é necessário comparar o *watermark* recebido com o *watermark* obtido durante uma nova execução do algoritmo sobre o texto. A técnica auxilia a verificação de possíveis alterações no texto [13];

c. Criação de *watermark* invisível que é inserida em páginas HTML através da *tag* <*meta*>. A marca d'água é criada, submetida a uma função de *Hash*, seu

resultado é convertido para oito dígitos e inserido no documento. Os caracteres são inseridos como espaços em branco, sendo ignorados por diversos programas [14];

d. A definição da *watermarking* está baseada em características como a distância entre as palavras [15] ou analisando características de espaçamento e tamanho entre caracteres [16];

e. Aplicação sobre o texto de pequenos pontos que se tornam pouco perceptíveis durante a leitura. Para recuperação da imagem é empregada autocorrelação entre os pontos, além da aplicação de pontos de registro para corrigir distorções geométricas [17];

f. Através da alteração de características dos caracteres, como o tamanho, é possível introduzir uma mensagem. A alteração de *pixels* do caractere permite a introdução da mensagem nas letras. Caso haja poucas alterações de *pixels*, a mensagem inserida torna-se menos robusta, entretanto menos perceptível. Em situação contrária, muitas alterações de *pixels* tornam a mensagem mais robusta e mais perceptível [18].

Analisando os trabalhos correlatos, o objetivo da pesquisa é criar uma técnica de *watermarking* que seja: não perceptível; mantenha informações sobre o detentor do documento; recuperável; e permaneça no documento mesmo após sua impressão ou digitalização.

TÉCNICA PROPOSTA

Como o contexto de aplicação da técnica criada é o inquérito policial, antes da execução da técnica é preciso definir o momento em que a técnica será aplicada sobre os documentos do inquérito.

É preciso definir uma nova fase no procedimento de geração de cópias de documentos. A técnica será aplicada a documentos digitais editáveis que serão impressos ou disponibilizados eletronicamente. Antes da execução do algoritmo propriamente dito, é necessário definir as etapas anteriores que serviram de subsídio para execução da técnica proposta em uma cópia de prova documental. O algoritmo necessita de uma *watermark* que seja identificadora do detentor da cópia. O processo de emissão de cópias precisa manter um registro histórico dos documentos emitidos e vincular o identificador a seu destinatário. Além de considerar que os documentos foram obtidos originalmente de um meio eletrônico.

Nesta fase, teremos as seguintes etapas após a elaboração do documento:

1. Criação de um código identificador, representado por uma sequência de bits, para o destinatário da cópia, devendo ser armazenado e registrado historicamente a vinculação do documento a este identificador;
2. Definição de um subconjunto C com as letras do alfabeto que terão suas formas alteradas no texto;
3. Inclusão, opcional, de preâmbulo ou mecanismo de segurança à marca d'água.

O código identificador pode ser acrescentado de elementos diferentes, com finalidades distintas. A marca d'água pode vir acompanhada de um código preâmbulo; ser submetida a uma função criptográfica; ou um código de verificação de erro. No caso do preâmbulo, é definido seu tamanho e se será inserido no inicio e/ou fim do código identificador, desta forma auxiliando na identificação da *watermark* no texto. A função criptográfica será aplicada com a finalidade de ocultação da mensagem proposta, entretanto pode gerar um *watermark* de tamanho maior. A terceira forma proposta consiste na inclusão de um código de verificação de erros associado ao código identificador. Uma abordagem que poderia ser utilizada com códigos identificadores pequenos é o código de Hamming(7,4) [19], o qual possibilita a detecção de até dois bits e a recuperação de até um bit da mensagem. A aplicação de um código de verificação contribui para solução de dois problemas desse contexto de marcas d'água em documentos impressos ou digitalizados: como existe a possibilidade de ruídos, é possível que alguma informação seja perdida durante o processo de recuperação da *watermark*; e no caso do código Hamming, permite a detecção da letra que sofreu interferência ou foi alterada propositalmente.

O subconjunto descrito anteriormente é uma das bases do método. A adição de uma marca d'água não perceptível em texto utiliza variações no formato das letras dos documentos. São pequenas variações que se tornam pouco perceptíveis quando inseridos em um documento completo. As diferenças das letras podem ser impostas através de fontes novas com divergências intencionais ou através de duas fontes textuais predefinidas em editores de texto que possuem letras com formas semelhantes, como a letra 'a' da fonte *Arial* e a mesma letra 'a' da fonte *Calibri*. A escolha dos caracteres que compõem o subconjunto deverá considerar características presentes no idioma em que o texto é escrito. Devem ser analisados aspectos como:

a. Frequência de ocorrência dos caracteres nas palavras do idioma;

b. Ocorrência de dígrafos, como: rr ou ss;

c. Um subconjunto C com muitos caracteres permite a inclusão de *watermark* em um menor fragmento de texto. Entretanto, pode comprometer a marca d'água, tornando mais frequentes e perceptíveis os caracteres alterados [18];

d. O tamanho dos caracteres no documento influenciará a percepção das divergências entre letras;

e. Distorções que possam acontecer nos caracteres durante o processo de digitalização.

A figura 1 exemplifica possíveis alterações no formato dos caracteres. As elipses em vermelho realçam as diferenças das letras. Apesar de ser relativamente fácil comparar a primeira linha com a segunda, deve ser considerado que as diferenças serão pequenas, a quantidade de letras ao redor será grande e o tamanho das letras será menor que o exemplo da figura 1.

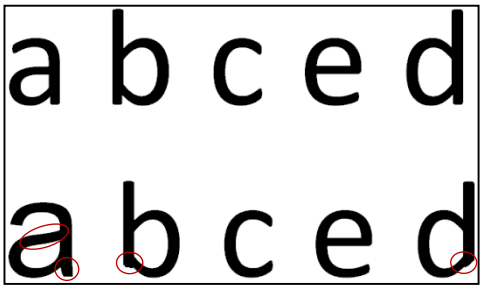


Fig. 1. Exemplo de caracteres com formas diferentes.

Definido a marca d'água, então é possível inserir o elemento de segurança ao texto a ser protegido. As figuras 2 e 3 ilustram o diagrama de inserção e recuperação das imagens nos documentos. As figuras exemplificam a utilização de um preâmbulo adicionado ao código identificador.

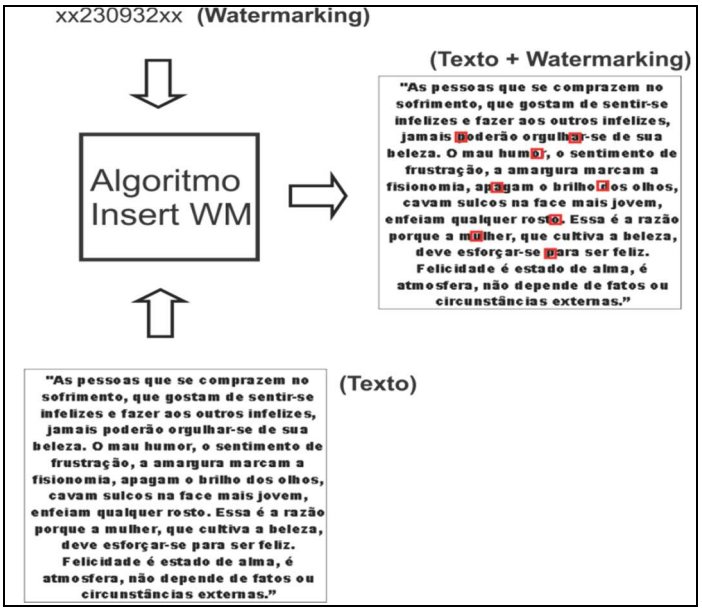


Fig. 2. Inserção da marca d'água no texto.

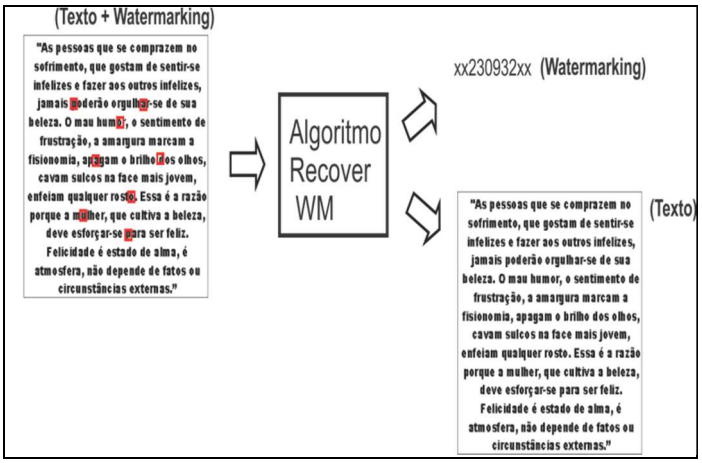


Fig. 3. Recuperação da marca d'água do texto protegido.

De posse da marca d'água e do texto, o algoritmo de inserção do elemento de segurança é iniciado (Tabela I). Como a *watermark* é uma sequência de bits, a cada representação '1' da marca d'água, o próximo caractere do texto, que está presente no subconjunto C, terá sua forma substituída. A cada representação '0', o próximo caractere do texto, que está presente no subconjunto C, terá sua forma original mantida. Desta maneira, a mudança de caracteres no texto dependerá da quantidade de números '1' presentes na *watermark*.

O algoritmo será executado sobre todo o texto, então após a execução do último dígito da *watermark* é realizada uma nova iteração da mesma marca d'água sobre os caracteres restantes do texto. Para evitar que seja percebida a alteração no texto por um leitor, algumas regras de modificação de caracteres são utilizadas:

- Caso ocorram casos de repetição de caracteres consecutivos, como: rr, ss, ee, oo, entre outras. O algoritmo ignora os caracteres;
- Caracteres em caixa-alta são ignorados;
- Títulos ou palavras em destaque que utilizem negrito, itálico ou que utilizem tamanho de fonte diferente do padrão do documento serão ignorados pelo algoritmo;
- Legendas de imagens, tabelas e notas de rodapé também são ignoradas pelo algoritmo.

Transformando essas especificações em algoritmos teríamos:

ALGORITMO PARA INSERÇÃO DE MARCA D'ÁGUA EM TEXTO

```
1. Obter Watermark;
2. Obter Subconjunto C; //Caracteres de forma variável
3. Enquanto texto não acabar Faça
4.   Próximo caractere do texto;
5.   Se letra pertence ao subconjunto C Então
6.     Se letra é um caractere válido Então //Regras
7.       Se próxima_letra_Watermark() == 1 Então
8.         Alterar forma da letra no texto;
9.       Fim_Se
10.    Fim_Se
11.  Fim_Se
12. Fim_Loop
```

ALGORITMO PARA RECUPERAÇÃO DE MARCA D'ÁGUA

```
1. Obter Subconjunto C;
2. Enquanto Texto não acabar Faça
3.   Próximo caractere do texto;
4.   Se letra pertence ao subconjunto C Então
5.     Se letra é um caractere válido Então  //Regras
6.       Se letra tem a forma alterada Então
7.         Watermark = Watermark + '1';
8.       Senão
9.         Watermark = Watermark + '0';
10.      Fim_Se
11.    Fim_Se
12.  Fim_se
13. Fim_Loop
```

Na linha 6 do algoritmo de inserção de *watermark* (Tabela I) e na linha 5 do algoritmo de recuperação de *watermark* (Tabela II), a verificação se o caractere é válido baseia-se nas regras descritas nesta seção para que o algoritmo ignore letras que são de tamanhos distintos, dígrafos, estão em caixa-alta ou fazem parte de alguma legenda.

Para ilustrar a utilização da técnica proposta foi definido um cenário de testes com duas marcas d'água com tamanhos diferentes. A primeira possui o tamanho de 16 bits, com preâmbulos de 4 bits e o código identificador de 8 bits. A segunda marca d'água utilizará uma palavra chave de 6 caracteres e será submetida ao algoritmo base64 [20], criando uma palavra de 64 bits no padrão ASCII. A função base64 é utilizada para demonstrar a possibilidade de utilização de uma função criptográfica sobre a marca d'água.

A marca d'água do primeiro teste será **1111100000101111**, enquanto a segunda marca d'água será **xx28xx**, que quando submetida à função base64 será **eHgyOHh4** em ASCII. Convertendo o texto para código binário, assumindo que cada letra é composta por oito bits, teremos **01100101 01001000 01100111 01111001 01001111 01001000 01101000 00110100**. O subconjunto de caracteres será: C = {**a, d, e, i, r**}, portanto esses caracteres poderão ser alterados no documento de acordo com as regras do algoritmo e da *watermark*

escolhida. O tamanho da fonte utilizando no texto de teste será 12 e fonte *Times New Roman*.

RESULTADOS E ANÁLISE

No exemplo, as letras do subconjunto C utilizam a fonte textual *Caladea* (Figura 4). Ressalta-se que foram utilizadas duas fontes textuais previamente definidas, o que não impossibilita que sejam utilizadas fontes novas, criadas especificamente para inserir a *watermark*.

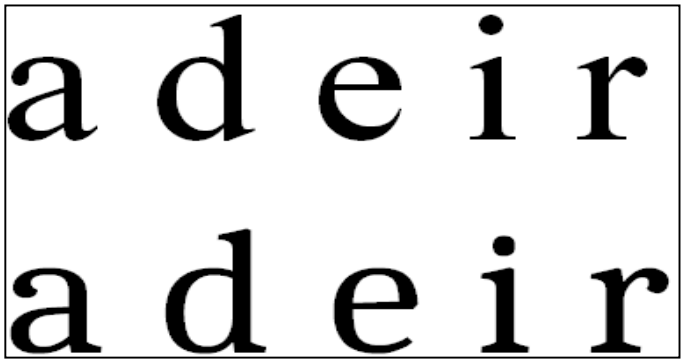


Fig. 4. Subconjunto C com os caracteres utilizados para teste. Na linha superior, os caracteres utilizam fonte *Times New Roman*. Na linha inferior utilizam fonte *Caladea*.

O texto utilizado para teste, como descrito na seção anterior, utiliza fonte *Times New Roman* de tamanho número 12, como ilustra a figura seguinte.

Balança enganosa é abominação para o SENHOR, mas o peso justo é o seu prazer. Em vindo a soberba, virá também a afronta; mas com os humildes está a sabedoria. A sinceridade dos íntegros os guiará, mas a perversidade dos aleivosos os destruirá.
De nada aproveitam as riquezas no dia da ira, mas a justiça livra da morte. A justiça do sincero endireitará o seu caminho, mas o perverso pela sua falsidade cairá.

Fig. 5. Texto original antes da inserção da marca d'água.

Três diferentes testes foram realizados sobre o texto. Primeiramente, duas *watermarks* de tamanhos diferentes foram aplicadas a duas cópias do texto. Posteriormente, o texto com a marca d'água será exibido após ser impresso e digitalizado, exemplificando quatro resoluções diferentes no equipamento de digitalização. Por fim, o material será exibido através de duas fotografias de *Smartphones* com câmeras de 5 e 8 *megapixels*. Os resultados são exibidos nas subseções seguintes.

Watermarks de tamanhos diferentes

Como descrito anteriormente, uma marca d'água terá 16 bits (Figura 6), enquanto a outra usará 64 bits (Figura 7).

Para facilitar a visualização da aplicação das marcas d'água, a cada iteração da *watermark* no texto seu início e término será demarcada com retângulos, e as letras que forma alteradas pelo algoritmo serão representadas pelos valores binários '0' ou '1' de acordo com a correspondência da *watermark* (Figuras 8 e 9).

Balança enganosa é abominação para o SENHOR, mas o peso justo é o seu prazer. Em vindo a soberba, virá também a afronta; mas com os humildes está a sabedoria. A sinceridade dos íntegros os guiará, mas a perversidade dos aleivosos os destruirá.
De nada aproveitam as riquezas no dia da ira, mas a justiça livra da morte. A justiça do sincero endireitará o seu caminho, mas o perverso pela sua falsidade cairá.

Fig. 6. Texto após a inclusão da *watermark* de 16 bits.

Balança enganosa é abominação para o SENHOR, mas o peso justo é o seu prazer. Em vindo à soberba, virá também a afronta; mas com os humildes está a sabedoria. A sinceridade dos íntegros os guiará, mas a perversidade dos aleivosos os destruirá.
De nada aproveitam as riquezas no dia da ira, mas a justiça livra da morte. A justiça do sincero endireitará o seu caminho, mas o perverso pela sua falsidade cairá.

Fig. 7. Texto após a inclusão da *watermark* de 64 bits.

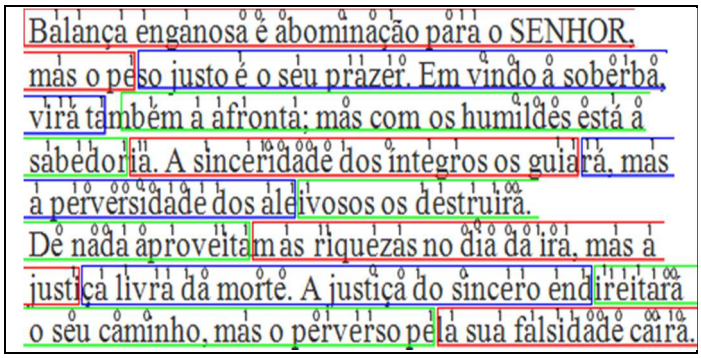


Fig. 8. Texto com a *watermark* de 16 bits e marcações.

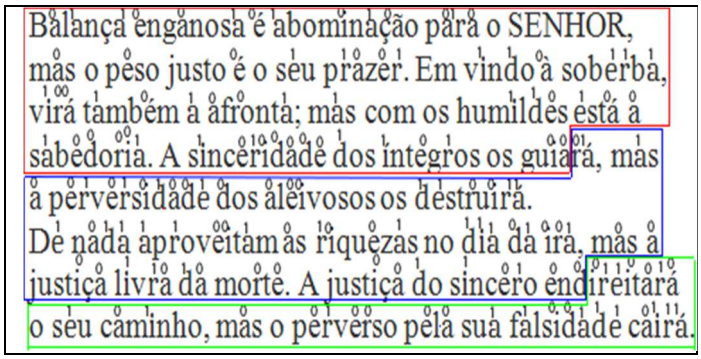


Fig. 9. Texto com a *watermark* de 64 bits e marcações.

Enquanto a marca d'água de 16 bits (Figura 8) utiliza pouco mais de uma linha para ser expressa, a segunda de 64 bits (Figura 9) utiliza quatro linhas para concluir sua primeira representação. A primeira *watermark* foi repetida nove vezes no trecho apresentado, enquanto a segunda apenas concluiu duas repetições no mesmo trecho de texto.

Os aspectos a serem destacados sobre a primeira *watermark* são:

- Capacidade maior de repetição;

- Possibilidade de recuperação da *watermark* em pequenos fragmentos de texto;
- Como esta utiliza preâmbulos, no pior caso de busca da marca d'água no texto serão verificados [2 * (tamanho em bits) - 1] caracteres pertencentes ao subconjunto C, e no melhor caso a quantidade de bits da *watermark*.

Os aspectos a serem destacados sobre a segunda *watermark* são:

- Utilização de uma função criptográfica antes da aplicação da técnica, a fim de proteger o código identificador;
- Apesar de ter um número de bits maior, isto aumenta a complexidade de recuperação da sequência binária. Além disso, a aplicação de uma função criptográfica pode retornar sequências de bits de tamanhos variados.

Digitalizações com diferentes DPI (dots per inch)

Foram selecionados quatro modos de digitalização disponíveis em uma impressora multifuncional. As opções selecionadas foram: digitalização em escala de cinza; resolução de 75 dpi, que é a menor resolução do equipamento; resolução de 200 dpi, que é a configuração padrão do dispositivo; e 600 dpi, que é a maior resolução do equipamento. Os resultados são apresentados em sequência nas figuras 10, 11, 12 e 13.

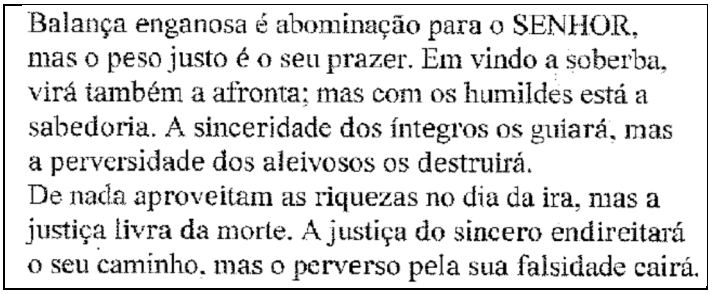

Balança enganosa é abominação para o SENHOR,
mas o peso justo é o seu prazer. Em vindo a soberba,
virá também a afronta; mas com os humildes está a
sabedoria. A sinceridade dos íntegros os guiará, mas
a perversidade dos aleivosos os destruirá.
De nada aproveitam as riquezas no dia da ira, mas a
justiça livra da morte. A justiça do sincero endireitará
o seu caminho, mas o perverso pela sua falsidade cairá.

Fig. 10. Digitalização em escala de cinza.

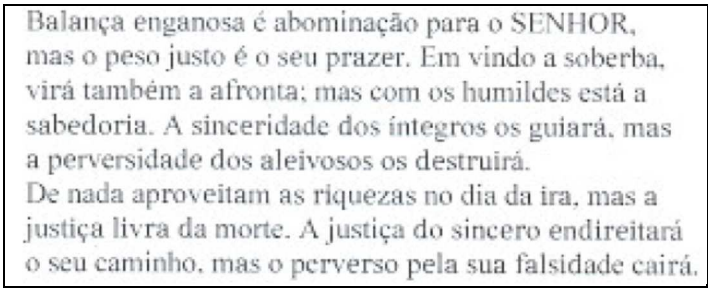

Balança enganosa é abominação para o SENHOR,
mas o peso justo é o seu prazer. Em vindo a soberba,
virá também a afronta; mas com os humildes está a
sabedoria. A sinceridade dos íntegros os guiará, mas
a perversidade dos aleivosos os destruirá.
De nada aproveitam as riquezas no dia da ira, mas a
justiça livra da morte. A justiça do sincero endireitará
o seu caminho, mas o perverso pela sua falsidade cairá.

Fig. 11. Resolução de 75 dpi.

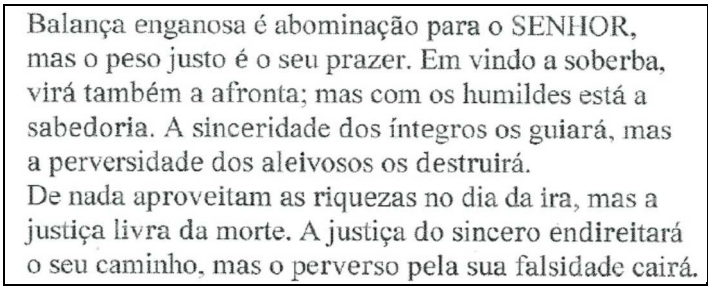

Balança enganosa é abominação para o SENHOR,
mas o peso justo é o seu prazer. Em vindo a soberba,
virá também a afronta; mas com os humildes está a
sabedoria. A sinceridade dos íntegros os guiará, mas
a perversidade dos aleivosos os destruirá.
De nada aproveitam as riquezas no dia da ira, mas a
justiça livra da morte. A justiça do sincero endireitará
o seu caminho, mas o perverso pela sua falsidade cairá.

Fig. 12. Resolução de 200 dpi.

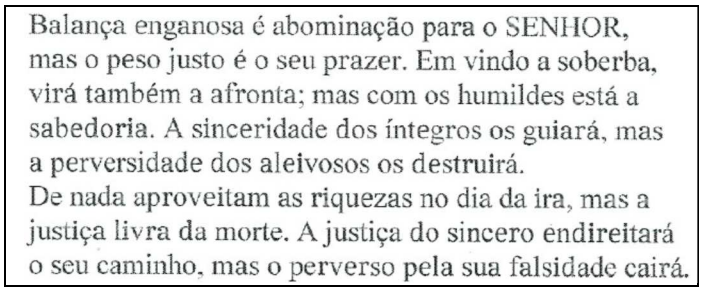

Balança enganosa é abominação para o SENHOR,
mas o peso justo é o seu prazer. Em vindo a soberba,
virá também a afronta; mas com os humildes está a
sabedoria. A sinceridade dos íntegros os guiará, mas
a perversidade dos aleivosos os destruirá.
De nada aproveitam as riquezas no dia da ira, mas a
justiça livra da morte. A justiça do sincero endireitará
o seu caminho, mas o perverso pela sua falsidade cairá.

Fig. 13. Resolução de 600 dpi.

As figuras 10, 11, 12 e 13 permitem concluir que a qualidade da digitalização de um documento pode trazer dificuldades para recuperação da *watermark*. Uma imagem muito degradada pode interferir na interpretação dos caracteres analisados. Entretanto, nos quatro cenários analisados, através da ampliação das imagens, ainda é possível verificar as diferenças entre os caracteres, como ilustra a figura 14. É perceptível que a degradação da imagem é mais acentuada com a resolução de 75 dpi, mas ainda existe a variação da silhueta da letra 'a' nos quatro cenários, o que a torna um caractere indicado para compor o subconjunto C. Propriedade esta que não é verdadeira para todos os caracteres de um alfabeto.

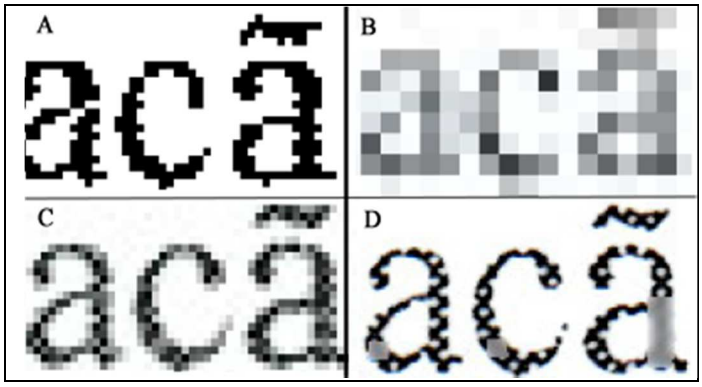


Fig. 14. Comparativo entre digitalizações do caractere 'a'. A) Digitalização em escala de cinza; B) Digitalização em resolução de 75 dpi; C) Digitalização em resolução de 200 dpi; D) Digitalização em resolução de 600 dpi.

Outro ponto a ser destacado é a distorção que algumas letras sofrem no processo de digitalização. Portanto, as variações dos caracteres selecionados para o subconjunto C devem ser avaliadas, para que não sejam ofuscadas pelo processo de digitalização.

Como a técnica de recuperação poderá ser empregada sobre imagens de baixa qualidade, o tratamento da imagem pode ser uma etapa anterior à aplicação do algoritmo. A manipulação de brilho e contraste de imagens podem acentuar as características das letras, como ilustra a figura 15, que foram submetidas a tratamento. Como é uma imagem em escala de cinza, através da equalização do histograma da imagem obtemos uma melhor distribuição de cores ao longo do histograma [21]. Assim, realçando as características das imagens e destacando as formas dos caracteres.

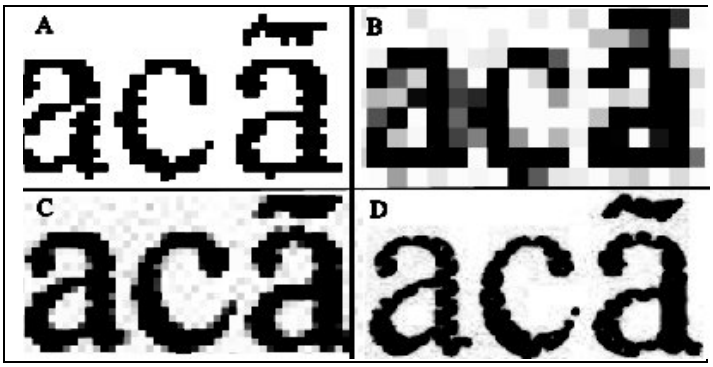


Fig. 15. Digitalizações do texto após o tratamento de brilho e contraste.

Imagem do texto através de fotografia

As imagens foram registradas por dois *Smartphones* com câmeras de 5 e 8 *megapixels*, respectivamente.

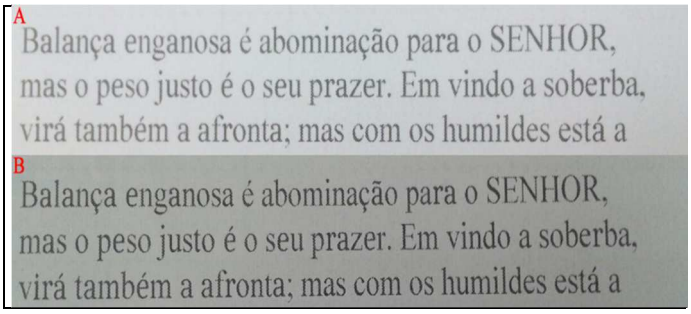


Fig. 16. Fotografia do texto após aplicação da marca d'água. A) câmera de 8 *megapixels*. B) câmera de 5 *megapixels*.

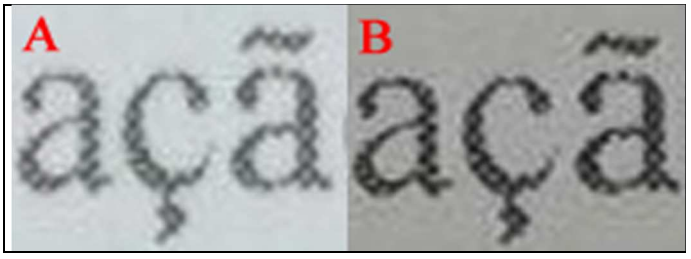


Fig. 17. Ampliação do texto fotografado após aplicação da marca d'água. A) câmera de 8 *megapixels*. B) câmera de 5 *megapixels*.

A análise da *watermark* através da fotografia permite concluir que as características identificadoras da marca d'água forma preservadas, permitindo a ampliação das imagens, mantendo os elementos identificadores dos caracteres, e obtendo resultados melhores que os textos digitalizados. Mesmo com o aumento da imagem (Figura 17), gerando distorções e ruído, ainda é possível identificar as letras que compõem a *watermark*.

Comparação entre a técnica proposta e outros métodos

Alguns critérios foram considerados, a fim de se comparar o método proposto com as técnicas encontradas na literatura. Os testes verificaram: a quantidade de informação introduzida no texto [11]; robustez contra ataques de alteração intencional de texto, deleção de sentença/linha e modificação de formatação do texto [19]; e detecção da *watermark* através de OCR. Foram selecionados os métodos *Line Shifting* e *Word-Shift Coding*.

Todas as técnicas foram aplicadas no texto da Figura 5. A Tabela III descreve a quantidade de bits inseridos no documentos com oito linhas, setenta e três palavras e trezentos e trinta e seis caracteres, desconsiderando os espaços em branco.

QUANTIDADE DE BITS POR TÉCNICA

Método	Nº de bits
Line Shifting	4
Word-Shift Coding	30
Método Proposto (subconjunto com 5 caracteres)	156

A análise quanto à robustez das técnicas, apenas o método proposto com a utilização de função criptográfica ou código de verificação de erros é capaz de detectar um ataque de alteração intencional do texto, entretanto somente utilizando o código de verificação de erros é possível recuperar o caractere alterado. Em relação à deleção de sentença/linha, os métodos *Line Shifting* e *Word-Shift Coding* não são robustos. O método proposto será robusto a depender do tamanho da *watermark* escolhida e dos caracteres do subconjunto C. Para o ataque de alteração de formatação, mudanças de alinhamento ou tamanho da fonte não comprometem a robustez do método proposto, o que não é verdade para os outros métodos testados.

Quanto à utilização de OCR para detecção das *watermarks*, foi utilizada a biblioteca *Tessaract* OCR [22], que quando executado com suas configurações padrões não detectou marcas d'água nas três técnicas.

CONCLUSÕES

A aplicação de *watermarking*, essencialmente, procura vincular uma produção, documental ou arquivo de mídia, a um autor. A técnica apresentada neste trabalho além de permitir a vinculação entre documento e autor, também possibilita a identificação do destinatário da cópia do documento. Dentro do contexto jurídico e policial, a guarda da prova documental pode afetar diretamente a condução de um processo ou inquérito. O vazamento de informações pode comprometer todo o trabalho realizado até o momento. Mas a partir do controle de cópia documental e da utilização de *watermarking*, são incluídos mais elementos fiscalizadores dentro do cenário descrito. Diferente de algumas técnicas de *watermarking* em documentos, a técnica proposta não precisa do documento original e nem de grandes fragmentos de texto para recuperação da *watermark*. Entretanto, a técnica ainda é vulnerável ao ataque de *re-typing*, que consiste em redigitar novamente o texto. Embora os vazamentos apresentados em veículos de comunicação prefiram apresentar os documentos em sua forma original com timbres, formatação e assinaturas do órgão emissor do documento.

Como a técnica utiliza imagens de textos para recuperação da marca d'água, é importante que o texto a ser analisado tenha qualidade em sua digitalização ou impressão compatível com a técnica apresentada. O texto submetido a processos de digitalização e impressão sequenciais influenciam a qualidade da informação e pode comprometer a eficiência da técnica. Uma alternativa para a baixa qualidade das imagens de texto seria o tratamento de imagem antes da aplicação da técnica.

A próxima etapa do trabalho será incluir a extração automatizada da *watermark* através de reconhecimento textual, como OCR, e melhorias na técnica para suportar ataques de *re-typing*. As variações de degradação dos caracteres durante o processo de digitalização precisam ser mensurados, a fim de

estipular um limite de qualidade do documento para ser submetido a um OCR ou mesmo ser passível de uma extração manual da marca d'água por meio da leitura do texto.

REFERÊNCIAS

[1] S. B. B. Ahmadi. *Digital Image Watermarking for Intellectal Property Protection. 4th International Scientific Conference of Iranian Academics*, 2014.

[2] S. S. Katariya. *Digital Watermarking: Review. International Journal of Engineering and Innovative Technology*, 2012.

[3] Y. Zhang e H. Qin. *A Novel Robust Text Watermarking For Word Document. 3º International Congress on Image and Signal Processing*, 2010.

[4] I. J. Cox et. Al. *Digital Watermarking and Steganography - 2º edition*. Ed. Elsevier, 2008.

[5] C. S. Woo. *Digital Image Watermarking Methods for Copyright Protection and Authentication. Thesis submitted in accordance with the regulations for Degree of Doctor of Philosophy*, 2007.

[6] S. P. Mohanty. *Digital Watermarking: A Tutorial Review. University of South Florida*, 1999.

[7] J. T. Brassil, S. Low e N. F. Maxemchuk. *Copyright Protection for the Electronic Distribution of Text Documents. Proceedings of the* IEEE, vol. 87, no. 7, pp.1181-1196, 1999.

[8] P. B. Devidas e P. N. Namdeo. *Text Watermarking Algorithm Using Structural Approach. World Congress on Information and Communication Technologies* - IEEE, 2012.

[9] M. Atallah et Al. *Natural Language Processing for Information Assurance and Security: An Overview and Implementations. Proceedings 9th ACM/SIGSAC New Security Paradigms Workshop, Cork, Ireland*, pp. 51-65, 2000.

[10] D. Huang e H. Yan. *Interword Distance Changes Represented by Sine Waves for Watermarking Text Image. Transactions on Circuits and Systems for Video Technology* - IEEE, Vol. 11, NO. 12, 2001.

[11] R. Davarzani e K. Yaghmaie. *Farsi Text Watermarking Based on Character Coding. International Conference on Signal Processing Systems*, 2009.

[12] S. Kaur. *A Zero-Watermarking algorithm on multiple occurrences of letters for text tampering detection. Internacional Journal on Coomputer Science and Engineering*, 2013.

[13] N. Mir. *Copyright for web content using invisible text watermarking, Computers in Human Behavior*, 2014.

[14] J. Cummins, P. Diskin e S. Lau. *Steganography and Watermarking. The University of Birmingham*, 2004.

[15] D. Huang e H. Yan. *Interword Distance Changes Represented by Sine Waves for Watermarking Text Images. Transactions on Circuits and Systems for Video Technology* - IEEE, Vol. 11, NO. 12, 2001.

[16] H. Lu et al. *A New Chinese Text Digital Watermarking for Copyright Protecting Word Document. International Conference on Communication and Mobile Computing*, 2009.

[17] H. Y. Kim e J. Mayer. *Data Hiding for Binary Documents Robust to Print-Scan, Photocopy and Geometric Distortions. Computer Graphics and Image Processing*, 2007.

[18] A. L. Varna, S. Rane e A. Vetro. *Data Hiding in Hard-Copy Text Documents Robust to Print, Scan and Photocopy Operations.* ICASSP, 2009.

[19] Q. Chen et al. *Word Text Watermarking for IP Protection and Tamper Localization*. IEEE, 2011.

[20] J. Linn. *Privacy Enhancement for Internet Electronic Mail. Request for Comments* - 989, 1987.

[21] R. C. Gonzalez e R. E. Woods. Processamento Digital de Imagens. 3ª Edição. Editora Pearson, 2010.

[22]Tesseract OCR engine, disponível em: http://code.google.com/p/tesseract-ocr/.

Felippe Pires Ferreira¸ Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília-DF, Brasil, E-mail: felippe_pires@yahoo.com.br.

DOI: 10.5769/C2015003 or http://dx.doi.org/10.5769/C2015003

Método para Análise Acústica e Reconhecimento de Vogais em Exames de Comparação de Locutores

Andréa Alves Guimarães Dresch, Hugo Vieira Neto, André Eugênio Lazzaretti e Rubens Alexandre de Faria

***Resumo*— Exames de Comparação Forense de Locutores apresentam características complexas, demandando análises demoradas quando realizadas manualmente. Propõe-se um método para reconhecimento automático de vogais com extração de características para análises acústicas, objetivando-se contribuir com uma ferramenta de apoio nesses exames. A proposta baseia-se na medição dos formantes através de LPC, seletivamente por detecção da frequência fundamental, taxa de passagem por zero, largura de banda e continuidade. Realiza-se o agrupamento das amostras através do método *k-means*, com centros iniciais determinados a partir dos histogramas dos primeiros formantes. Experimentos preliminares com uma base de dados pré-classificados forneceram resultados promissores, com localização de regiões correspondentes às vogais anteriores e posterior média-baixa, propiciando a visualização do comportamento do trato vocal de um falante.**

***Palavras-Chave*— Fonética Forense, Exame de Comparação de Locutores, Análise Acústica, Trapézio Fonético, software Praat.**

***Abstract*— Forensic Speaker Comparison exams have complex characteristics, demanding a long time for manual analysis. A method for automatic recognition of vowels, providing feature extraction for acoustics analysis is proposed, aiming to contribute as a support tool in these exams. The proposal is based in formant measurements by LPC, selectively by fundamental frequency detection, zero crossing rate, bandwidth and continuity. The k-means method is used for clustering, with initial centers determined from the first formants' histograms. Preliminary experiments, using a pre-classified database, have shown promising results, in which regions corresponding to front and lower-middle back vowels were successfully detected, providing visualization of a speaker's vocal tract behavior.**

Keywords*— *Forensic Phonetics, Forensic Speaker Comparison Exam, Acoustic analysis, Phonetic Trapezium, Praat software.

I. Introdução

A produção de provas por meio de registros de áudio, em especial após a promulgação da Lei 9296/96 que trata das interceptações telefônicas [1], tem crescido e consequentemente intensificado as demandas da área forense referente às perícias audiovisuais para atribuição de autorias.

O exame de Comparação de Locutores (CL) tem por finalidade verificar se dois registros de voz e fala foram produzidos por um mesmo indivíduo, consistindo na comparação entre um registro de áudio denominado questionado (sobre o qual pairam dúvidas quanto à autoria das falas) e um registro padrão (registros de fala de identidade conhecida). A importância desse exame reside na possibilidade de associar ou desvincular um indivíduo a um fato delituoso materializado através de um registro de áudio [2].

Relatórios de diagnóstico da Segurança Pública e da Perícia Criminal brasileira apontam a carência de peritos criminais [3] [4], que consequentemente culmina em passivo de laudos nos Institutos de Criminalística. O represamento de materiais a serem examinados prejudica a celeridade necessária para a produção de provas, o que, de acordo com Vargas e colaboradores [5], contribui para a morosidade de um processo penal.

Nesse contexto, agravado pela complexidade das análises envolvidas, uma vez que o exame de CL requer um tempo de execução muito superior à média dos demais exames periciais, a gestão de recursos humanos de Seções de Perícias Audiovisuais é dificultada, analogamente ao constatado por Vrubel e colaboradores em relação à Seção de Computação Forense [6].

Principalmente devido à interdisciplinaridade inerente a esse exame [7], e à construção de conhecimento que exige, o número reduzido de peritos criminais alocados para o mesmo é insuficiente. É desejável, portanto, que se busque o aperfeiçoamento das técnicas adotadas, para se otimizar a realização do exame - qualitativa e quantitativamente.

Em pesquisa realizada por Gold e French [8], foi efetuado um levantamento das técnicas utilizadas internacionalmente para esse exame em 13 países, sendo constatada a preponderância da utilização das análises classificadas como perceptivo-auditiva e acústico-instrumental. Os autores observaram que mesmo quando algum sistema de reconhecimento automático é utilizado, algum tipo de análise humana é feita, e que no Brasil são adotados os métodos perceptivo-auditivo e acústico-instrumental combinadamente.

A análise perceptivo-auditiva requer um profissional capacitado para identificar propriedades da qualidade da voz, traços linguísticos, padrões articulatórios, entre outros atributos. Por sua vez, a análise acústico-instrumental, ou simplesmente análise acústica, engloba medições de curto e de longo termo, nos domínios temporal e espectral.

Para realização dessa tarefa o software Praat é amplamente difundido [9], tanto no ambiente acadêmico como no forense. Contudo, algumas análises requerem extensiva segmentação de trechos com fonemas a serem submetidos à extração de parâmetros, o que, dependendo do volume do material, pode tornar o exame extremamente laborioso.

Sendo assim, a proposta desta pesquisa é o desenvolvimento de uma ferramenta para auxílio de análises acústicas que facilite a visualização de características úteis para o exame

Andréa Alves Guimarães Dresch, ICPR (Instituto de Criminalística do Paraná) e UTFPR (Universidade Federal Tecnológica do Paraná). Hugo Vieira Neto, UTFPR. André Eugênio Lazzaretti LACTEC (Instituto de Tecnologia para o Desenvolvimento). Rubens Alexandre de Faria, UTFPR. Curitiba-PR, Brasil. Emails: andrea.dresch@ic.pr.gov.br, hvieir@utfpr.edu.br, lazzaretti@lactec.org.br, rubens@utfpr.edu.br.

de CL (energia, frequência fundamental, frequência e banda de formantes, taxa de subida ou descida de formantes em um trecho).

Embora o foco seja forense, a ferramenta proposta também pode ser utilizada em outras áreas de linguística ou de fonoaudiologia. O intento é o reconhecimento de trechos vozeados de uma gravação, sem a obrigatoriedade de pré-segmentação manual, além da disponibilização de gráficos com possibilidade de seleção de áreas a serem reavaliadas com a visualização de oscilograma e espectrograma, com os trechos de interesse concatenados ou simplesmente etiquetados.

Tal funcionalidade seria útil, por exemplo, em análises do comportamento formântico a longo termo do trato vocal de um dado falante. Porém, nos casos em que tal hipótese não se confirme devido à interferência agressiva de ruído ou a particularidades da voz em questão, ou mesmo no caso de *outliers*, o analisador teria a possibilidade de confirmar perceptivamente o que ocorreu. Além disso, um padrão visual auxiliaria em análise intra e inter-sujeito, pois se espera em uma CL que sejam encontrados elementos estáveis o suficiente e que denotem similaridades em falas pertencentes a um falante, mas que não sejam comuns a outros indivíduos.

A medição dos formantes é feita pela técnica LPC (*Linear Predictive Coding*), conforme o método de Burg [10], com posterior ponderação de custos para determinação final dos valores de cada formante (com base na frequência e na banda). Serão descartados os pontos em que não houver detecção de F_0 (frequência fundamental), calculados através de autocorrelação nos *frames* (trechos em análise) com energia acima de limiar estabelecido e taxa de passagem por zero abaixo de limiares pré-estabelecidos.

Propõe-se ainda o reconhecimento de agrupamentos de pontos (ou *clusters*) referentes às regiões das vogais (anteriores/posteriores/centrais, altas/médias/baixas), idealmente encontrando a região de cada vogal (/a/,/e/, /ɛ/, /i/, /o/, /ɔ/ e /u/). Embora, como constatado por Escudero e colaboradores [11], no Português Brasileiro (PB) tal determinação possa ser feita por meio de várias combinações de parâmetros, a combinação dos formantes $F_1 \times F_2$ é a que melhor evidencia a distribuição das vogais. Para o reconhecimento serão realizados experimentos com os algoritmos *k-means* e distribuição Gaussiana.

O mecanismo desenvolvido deve permitir a análise das vogais, principalmente com base em seus valores de formantes, com medições realizadas sem necessidade de segmentação prévia. Neste trabalho são exploradas algumas estratégias para seleção dos instantes com valores válidos de formantes, tais como a detecção de frequência fundamental, determinação de limiares de taxa de passagem por zero e de energia de curto termo e continuidade de valores em amostras subsequentes, objetivando minimizar a interferência de fonemas consonantais.

A escolha do aproveitamento de interfaces do software Praat se deve pelo mesmo ser um software livre, e também pela familiaridade dos profissionais que trabalham com fonética. Pretende-se incorporar no futuro rotinas do software R [12], para a realização de cálculos estatísticos.

II. Fundamentação Teórica

A. Produção de Voz

A fala é um dos principais recursos de comunicação humana. Inicia-se por um processo interno do falante, que mentalmente formula a mensagem a ser transmitida, ocorrendo em seguida a ativação motora dos músculos e órgãos do aparelho fonador para a articulação da fala.

Após emissão da mensagem pelo falante e transmissão através do meio (o próprio ar ou um canal telefônico, por exemplo), terá vez o processo de percepção dos sons de fala pelo ouvinte. Tal processo é mais complexo do que a simples detecção de sinais acústicos (como tons puros ou ruído), pois é necessário identificar, categorizar e reconhecer esses sons em sua forma, para atribuir à fala seu significado (mensagem) [13] [14].

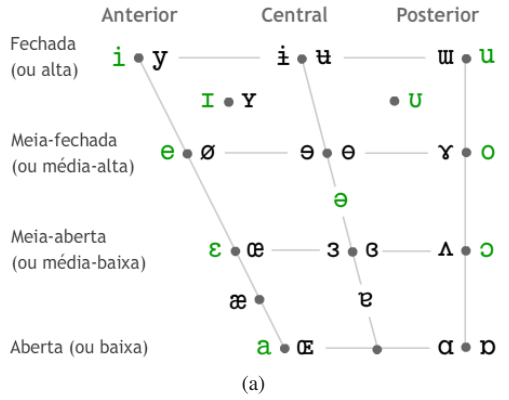
A voz é gerada pela conversão do fluxo contínuo de ar egresso dos pulmões em pulsos de ar (pulsos glóticos), quando ocorre a vibração das pregas vocais, responsável pela característica de vozeamento de vogais e de algumas consoantes. A frequência dessa vibração corresponderá à frequência fundamental (F_0), que possui como correlato acústico o *pitch* [15] [16].

As características anatômicas e fisiológicas do trato vocal provocam ressonâncias nos sons originados dos pulsos glóticos, conforme descrito no modelo fonte-filtro, que considera o sistema de geração do sinal de voz como uma composição de uma fonte de excitação (pulsos glóticos) acoplado a um filtro modelado pela anatomia do trato vocal. Durante a produção de fonemas vocálicos, as frequências amplificadas resultam nos formantes (F_1, F_2, F_3,...,F_n) [14]. Os primeiros formantes, F_1 e F_2, têm relação direta com a altura e o recuo da língua [13], sendo que a sua representação gráfica é normalmente realizada através do diagrama de Vogais Cardeais, também chamado de Trapézio Vocálico [17].

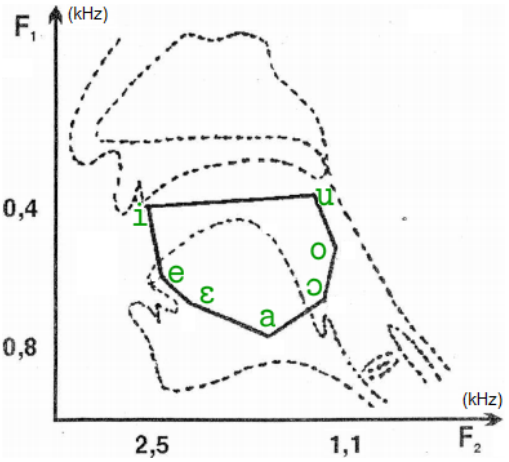
B. Fonemas do Português Brasileiro (PB)

As unidades linguísticas que organizam uma determinada língua são denominadas fonemas. No PB são subdivididos em vogais, semivogais ou *glides*, e consoantes.

- **Vogais:** representam o único tipo de segmento que pode atuar como núcleo silábico. São segmentos vozeados ou sonoros, devido à vibração das pregas vocais que sempre ocorre durante a sua articulação. Outro ponto importante para sua caracterização é que durante a sua produção o fluxo de ar não sofre obstruções no trato vocal, e como consequência os segmentos vocálicos geralmente apresentam maior energia que os segmentos consonantais [13] [14].
  Na Figura 1(a) é apresentado o trapézio fonético das vogais, em que as barras verticais e horizontais são alusivas à posição da língua nos respectivos eixos durante a produção de cada vogal. Dessa forma, cada vogal corresponde a uma configuração do trato vocal, interferindo diretamente nos valores dos formantes. O formante F_1 diz respeito à posição da língua no eixo vertical e F_2 à sua posição no eixo horizontal, conforme Figura 1(b), que ilustra as posições da língua durante a produção das



(a)



(b)

Fig. 1. Ilustração das posições das vogais no trapézio fonético e no gráfico $F_1 \times F_2$. (a) Trapézio fonético das vogais de acordo com o IPA (*International Phonetic Alfabet*), com destaque em verde nas vogais que ocorrem no PB – reproduzido de [18]; (b) Sobreposição do gráfico de $F_1 \times F_2$ (eixos com valores descrescentes para facilitar a análise) e de uma ilustração com a posição da língua durante a produção das vogais orais tônicas – adaptado de [13].

vogais orais tônicas **/a/** - "a", **/e/** - "ê", **/ɛ/** - "é", **/i/** - "i", **/o/** - "ô", **/ɔ/** - "ó" e **/u/** - "u" [13] [16].

- **Semivogais ou *glides*:** também são fonemas vozeados similares às vogais, porém com menor duração. No PB conectam-se a vogais para formar ditongos ou tritongos, podendo ser: /j/, como na palavra [paj] - "pai"; e /w/, como na palavra [maw] - "mau".
- **Consoantes:** ao contrário das vogais, durante a produção de uma consoante, o fluxo de ar egresso dos pulmões sofre obstrução total ou parcial. Assim, em decorrência do tipo de obstrução as consoantes podem ser: plosivas ou oclusivas (exemplos: /p/, /t/ e /g/ em [**p**ato] e [**g**ato]); fricativas (exemplos: /f/ em [**f**ɔka] - "foca", /s/ em [**s**apo], /ʃ/ em [**ʃ**ato] - "chato"); africadas (exemplo: /tʃ/ em [**tʃ**ia] - "tia"); nasais (exemplos: /m/ em [**m**ato] e /ɲ/ em [so**ɲ**o] -"sonho"); laterais (/l/ em [**l**ate]); tepes (exemplo: /ɾ/ em [ca**ɾ**o] - "caro"); e vibrantes (exemplo: /ř/ em [ca**ř**o] - "carro") [13] [16].
  Outra classificação diz respeito ao ponto da articulação: bilabial, labiodental, dental, alveolar, alveopalatal, palatal, velar ou glotal. As consoantes ainda podem ser vozeadas ou desvozeadas, sendo que na análise espectral de consoantes com mesmo ponto e modo de articulação (como por exemplo [f] e [v], de faca e vaca), a diferença pode ser observada através da barra de vozeamento (para o [v]).

C. Frequência fundamental

Estimadores de frequência fundamental procuram o componente frequencial que se sobressai em um trecho do sinal, valor que deverá ser equivalente ao período entre pulsos glóticos. Duas abordagens bastante utilizadas são a autocorrelação e a análise cepstral. Neste trabalho, optou-se pelo método de autocorrelação, por se mostrar mais robusto à presença de ruído [19].

O algoritmo nativo do software Praat calcula a autocorrelação de cada bloco de sinal submetido a uma janela de *Hanning* ou Gaussiana, sendo o resultado obtido pela divisão da função de autocorrelação do sinal pela autocorrelação da própria janela, como demonstrado na Equação (1), em que $r_x(\tau)$ corresponde à autocorrelação resultante, $r_{xw}(\tau)$ à autocorrelação do sinal janelado e $r_w(\tau)$ à autocorrelação da janela utilizada. Dessa forma, evita-se que harmônicos sejam confundidos com a frequência fundamental [20].

$$r_x(\tau) \approx r_{xw}(\tau)/r_w(\tau) \tag{1}$$

O algoritmo possui ainda refinamentos, com limiares de silêncio e de vozeamento e a atribuição de custos para transições de vozeamento/desvozeamento, valor de oitava e salto de oitava entre dois *frames* consecutivos. O tamanho da janela de análise também está atrelado ao limite inferior para busca de frequência (*pitch floor*) [9].

D. Formantes

Uma forma de reconhecer as regiões vocálicas de um sinal de voz é através da obtenção dos formantes, que pode ser feita pela aproximação do envelope espectral desse sinal através de uma análise de predição linear, ou LPC (*Linear Predictive Coding*), exemplificado na Figura 2. Tal técnica consiste em separar o sinal de excitação da resposta do trato vocal, extraindo justamente a informação de formantes que é de interesse para a análise [14].

A análise de predição linear parte do pressuposto de que cada amostra do sinal de fala é, aproximadamente, uma combinação linear das amostras anteriores. Normalmente é feita através de métodos de covariância ou de autocorrelação [21]. Uma representação deste modelo pode ser visualizada na Equação (2), em que $s[n]$ representa o sinal de saída, $x[n]$ o sinal de entrada e m o número de coeficientes que corresponderá a ordem do sistema. Uma vez que o sinal de entrada é desconhecido, o valor $\hat{s}[n]$ na Equação (3) seria uma estimativa do valor da amostra atual. O objetivo da análise preditiva é a determinação dos coeficientes $(a[i]|i = 1, \ldots, m)$ de forma que o erro de predição $e[n]$, constante na Equação (4), seja o menor possível.

$$s[n] = \sum_{i=1}^{m} a_i s[n-i] + x[n] \tag{2}$$

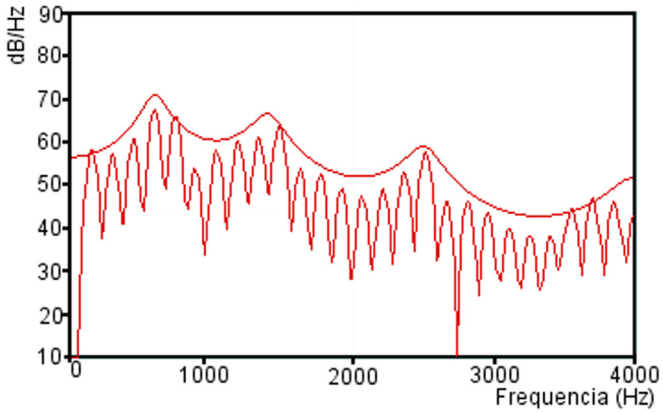


Fig. 2. Figura contemplando o espectro LPC (mais suavizado, em contraste com espectro de Fourier na parte inferior) de uma vogal /a/ produzida por um falante adulto do sexo masculino.

$$\hat{s}[n] = \sum_{i=1}^{m} a_i s[n-i] \tag{3}$$

$$e[n] = s[n] - \hat{s}[n] \tag{4}$$

Para este trabalho foi escolhido o algoritmo de Burg, por ser considerado um modelo estável e que apresenta bons resultados para gravações de curta duração [10]. O algoritmo de Burg considera, além da predição referente às amostras anteriores, y_n na Equação (5), também a referente às amostras posteriores, z_n na Equação (6). A escolha dos coeficientes é feita de forma a minimizar o erro de ambos os sentidos. A quantidade máxima de número de coeficientes, na prática, é determinada pelo valor da frequência de amostragem (em kHz) mais dois [21].

$$y_n = -\sum_{i=1}^{m} a_i x_{[n-i]} \tag{5}$$

$$z_n = -\sum_{i=1}^{m} a_i x_{[n+i]} \tag{6}$$

III. Materiais e Métodos

A. Amostras de dados

Para os experimentos preliminares, estão sendo utilizados arquivos de áudio produzidos em pesquisa realizada pelo Grupo de Estudos de Sons da Fala da UTFPR [22]. Trata-se de gravações de oito pesquisadoras, com a leitura de texto, apresentando aproximadamente 60 segundos de duração.

Os fonemas alvo daquele estudo eram interplosivos e presentes em sílabas tônicas, totalizando quatro repetições para cada uma das vogais orais tônicas do PB, as quais foram manualmente etiquetadas, servindo de referência.

Em uma próxima etapa do projeto serão utilizados os corpora *Spoltech* e "C-ORAL". O primeiro é um corpus compilado através do projeto "*CORPORA from CSLU: The Spoltech Brazilian Portuguese v1.0*" [23], que apresenta 8.080 trechos, com falas de 477 falantes, consistindo de leituras de sentenças foneticamente balanceadas e de respostas a perguntas. O segundo é um corpus compilado através do projeto "C-ORAL", desenvolvido pelo Núcleo de Estudos em Linguagem, Cognição e Cultura da Universidade Federal de Minas Gerais [24], que apresenta registros com fala espontânea, trazendo uma proximidade maior de situações reais.

B. Algoritmos utilizados

Para esta etapa está sendo utilizado, academicamente, o software de análise matemática *Matlab*, e sua *toolbox* de Processamento de Sinais. Após a seleção no Praat do arquivo ou do trecho a ser submetido à análise, inicia-se o processamento conforme ilustrado no fluxograma apresentado na Figura 3, cujos blocos principais estão enumerados e são descritos na sequência.

1) **Pré-processamento:** nesta etapa o sinal é reamostrado a uma taxa de 8kHz, e o nível DC removido através da subtração do nível médio do trecho.
2) **Rotina para Cálculo de ZCR:** o sinal é divido em frames (janelas) de 25ms de duração, sendo efetuado o cálculo do número de vezes que há alteração do sinal do valor da amostra (mudança de sinal de positivo para negativo e vice-versa). Após a finalização do processo acima, o resultado de todas as janelas são divididas pelo valor máximo para fins de normalização.
3) **Detecção de Frequência Fundamental:** no Praat é utilizada a opção "*To Pitch (ac)...*", por permitir a configuração dos parâmetros de inicialização, que incluem a definição das frequências mínima e máxima, além da escolha do tipo de janela (opção *Very accurate* para janela Gaussiana). O tamanho da janela não é definido, por ser uma função da frequência mínima. Neste primeiro momento mantiveram-se os valores de custo *default*. O objeto resultante é convertido para *PitchTier* e, em seguida para tabela, permitindo o armazenamento na forma de arquivo.
4) **Cálculo de Formantes:** no Praat é utilizada a opção "*To Formant (Burg)...*", que possibilita a escolha do número máximo de formantes a ser buscado, e o valor máximo da frequência. A largura da janela é configurada em 25ms, por ser um valor considerado (empiricamente) razoável para este tipo de análise. Mantem-se em 50 Hz o valor do filtro de pré-ênfase, que corresponde ao valor inicial em que o filtro atuará para corrigir a combinação da atenuação de altas frequências provocada pelo trato vocal e a amplificação associada à radiação (do som através da abertura dos lábios).
   Em seguida a matriz obtida é submetida à função "*Formant Track*", que considera os valores obtidos para cada *frame* como um candidato, ao qual é atribuído um custo referente ao valor da frequência, à banda, e à transição entre oitavas. O número máximo de formantes será menor, porém com maior exatidão dos valores obtidos. Após conversão para tabela, é realizada ainda uma limpeza de valores "*–undefined–*", para que o arquivo salvo possa ser corretamente carregado no Matlab.
5) **Seleção de amostras:** no Matlab as tabelas geradas

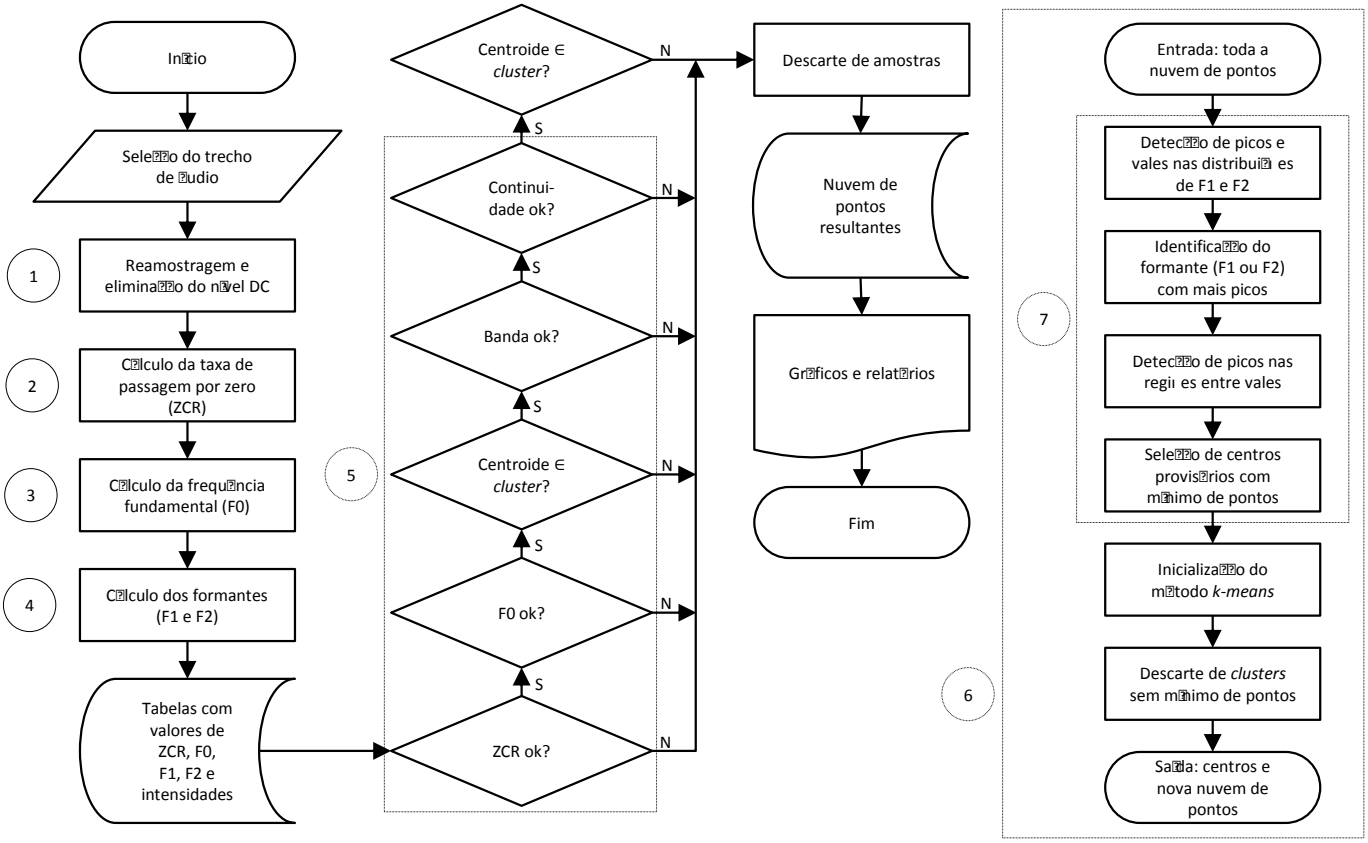


Fig. 3. Fluxograma dos algoritmos implementados. A numeração indicada ao lados dos blocos corresponde aos itens descritos na subseção III-B.

pelas rotinas do Praat são carregadas e salvas em matrizes. Realiza-se em seguida a etapa de seleção das amostras a terem os valores de formantes considerados. Inicialmente são eliminadas as que apresentem taxa de passagem por zero (ZCR) maior que 0,5 (50% do valor máximo), por apresentarem baixa probabilidade de serem voz.

Nos blocos não descartados anteriormente, mas nos quais não houve detecção de frequência fundamental, os valores dos formantes são desconsiderados. Em seguida, utiliza-se a função de busca por centroides (função explicada a seguir), em que só são mantidos os pontos que pertençam a um *cluster* (agrupamento) de tamanho mínimo . Dessa matriz são consideradas apenas as amostras que apresentem valores de banda de F_1 e F_2 menores que a média obtida para cada caso, e que com continuidade, ou seja, que o valor da amostra presente não tenha uma variação maior que 15% em relação aos blocos vizinhos. É feita uma nova busca por *clusters*, que resulta em uma matriz final e nos centroides identificados.

6) **Modelo de agrupamento:** a busca de centroides é realizado através do método *k-means*, o qual consiste no agrupamento sobre os padrões de entrada (formantes F_1 e F_2) em k grupos, sendo k um parâmetro definido *a priori*. O algoritmo é executado em duas etapas principais. Na primeira etapa, cada padrão de entrada é atribuído ao agrupamento mais próximo que esse se encontra, sendo a medida de proximidade representada pela distância euclidiana entre cada padrão e o centro do agrupamento. Na segunda etapa, é realizado o cálculo dos novos centros através da média aritmética entre a localização de todos os pontos associados a cada centro definido na primeira etapa. O processo se repete até que nenhuma nova alteração seja verificada nos agrupamentos, ou se um determinado número de iterações tenha ocorrido. No fim do algoritmo, cada padrão de entrada está associado a um dos agrupamentos definidos. Esse processo garante que a minimização da distância intra-grupos seja atingida no final das iterações, sendo essa a principal motivação da escolha deste método no contexto deste trabalho.

7) **Inicialização dos centroides:** nesta função chamada durante a seleção de amostras inicialmente são gerados histogramas suavizados (de forma a evidenciar máximos e mínimos das distribuições) para as matrizes de F_1 e F_2. É considerado aquele com maior número de máximos (o qual, intuitivamente mas não necessariamente, distinguiria melhor as regiões das diferentes vogais). A seguir o gráfico é subdividido em regiões (horizontais se $F1$ tem mais picos, ou verticais caso contrário), nas quais a geração de histograma suavizado é repetida. Com os valores desses máximos obtidos são determinados centroides temporários, para uma área delimitada pelos mínimos locais. Se essa área apresentar pelo menos 10% do número total de amostras, esse centro é considerado

válido. Caso o número de centros obtidos seja nulo, a função é repetida para o formante que inicialmente apresentou menor número de máximos.
Os centroides obtidos são utilizados para alimentar a função *k-means*, que na ausência de valores iniciais, a função estabeleceria os primeiros centros aleatoriamente, de forma que mesmo que houvesse um resultado convergente, este seria diferente a cada execução. Contudo, como há fornecimento dos valores iniciais no procedimento adotado, conforme descrição acima, a função torna-se determinística, fornecendo sempre os mesmos resultados sempre que executada.

IV. Resultados Preliminares

Nas análises do comportamento formântico foi observado que a simples remoção das amostras em que não houve detecção de frequência fundamental já resulta em um gráfico $F_1 \times F_2$ mais próximo do trapézio vocálico, conforme a Figura 1(a). Tal efeito pode ser visualizado na Figura 4(b), obtida a partir do processamento da nuvem de pontos da Figura 4(a), e na qual o contorno resultante se assemelha a um trapézio.

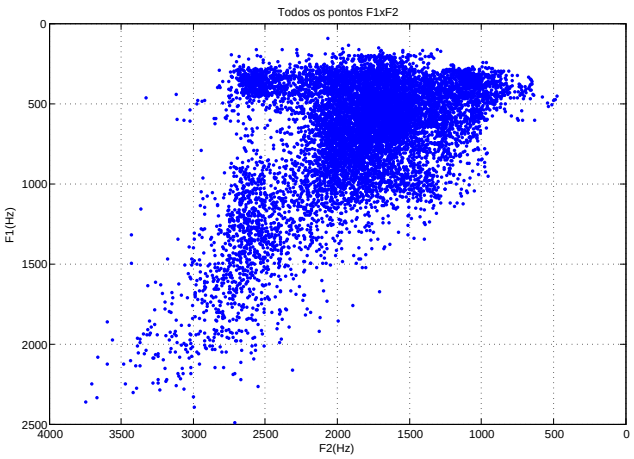
Outra forma de visualizar esse resultado é através da sobreposição da curva de formantes ao espectrograma. Conforme demonstrado na Figura 5, em que se observa em (a) a forma de onda de um trecho de áudio em análise com a sobreposição das funções do STE (energia) e de ZCR, em (b) um espectrograma de banda estreita com a sobreposição dos valores de frequência fundamental resultantes e em (c) o gráfico resultante para os valores dos formantes com a delimitação inicial dos trechos vozeados. É possível observar que os trechos considerados vozeados correspondem àqueles em que houve detecção frequência fundamental, apresentam uma energia relativa maior e baixa ZCR.

Conforme demonstrado na Figura 4(b), o gráfico resultante ainda apresenta pontos de frequências mais altas, possivelmente devido a efeitos de coarticulação, o que exigiu a aplicação dos demais algoritmos apresentados para que o conjunto resultante fosse mais consistente.

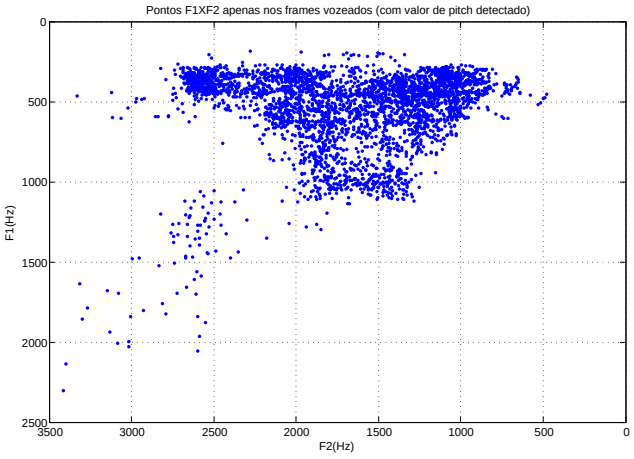
Após realização da etapa de busca de *clusters*, conforme fluxograma apresentado na Figura 3, obteve-se para as amostras da UTFPR um máximo de quatro centroides, com uma média de três.

Um exemplo de um dos gráficos resultantes é apresentado na Figura 6, no qual se observa a distribuição dos valores das amostras em um formato próximo a um trapézio. Os centroides obtidos durante a aplicação do método estão identificados pelos pontos em preto, enquanto que os valores de referência estão indicados pelos pontos vermelhos. É possível observar a proximidade dos centros com os valores de referência correspondentes, da esquerda para direita, às vogais **/e/**, **/ɛ/** e **/ɔ/**.

Na Tabela I são apresentados os valores de F_1 e F_2 obtidos para cada centro. Tais valores foram comparados com os valores de referência (Tabela II), referentes aos resultados da pesquisa realizada pelo Grupo de Estudos de Sons da Fala da UTFPR [22]. Para cada centro foi calculado, através de



(a)



(b)

Fig. 4. Exemplo de gráficos com resultados obtidos para uma das amostras de áudio do grupo de pesquisa da UTFPR. (a) Gráfico $F_1 \times F_2$ com todos os valores de formantes; (b) Gráfico $F_1 \times F_2$ com os valores de formantes nas janelas de análise com valor de frequência fundamental.

distância euclidiana, o valor de referência mais próximo, sendo determinada a vogal correspondente.

Calculou-se então, conforme apresentado na Tabela III, os percentuais das diferenças entre os formantes de cada centro e dos respectivos valores de referência, em relação aos próprios valores obtidos para os formantes. Tais percentuais variam entre 0,2% e 49,5% para F_1 e entre 0,1% e 16,1% para F_2. Entretanto, quando se leva em consideração, não apenas o ponto central, mas também a área da região correspondente através de seu desvio padrão das amostras referentes a cada cluster (Tabela IV), a distância entre o limite das regiões às referências é consideravelmente baixo, conforme pode-se observar nos valores da Tabela V.

Constata-se que em todos os casos houve centros identificados coincidindo com a referência para vogal posterior média-baixa (**/ɔ/**) e com a referência para vogal anterior alta ou média-alta (**/i/** ou **/e/**). Também se observa que centros coincidindo com a vogal anterior média-baixa (**/ɛ/**) ocorreram em nove amostras, coincidindo com vogal central baixa (**/a/**) foram menos frequentes, ocorrendo em apenas três casos (e

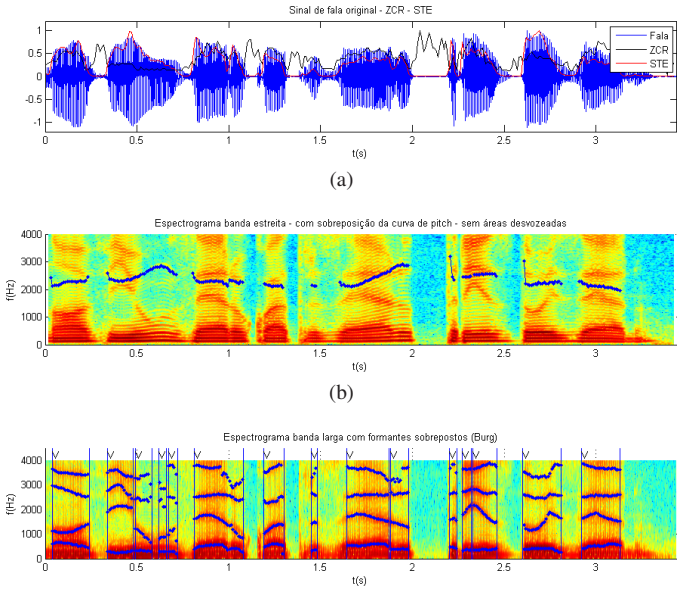


Fig. 5. Teste preliminar com um dos arquivos de "*Spoltech*", com a repetição: "nove, um, zero, quatro, zero, três, dois, zero". (a) Forma de onda com sobreposição dos gráficos de energia de curto prazo (STE) e taxa de passagem por zero (ZCR); (b) Espectrograma de banda estreita com sobreposição dos pontos de frequência fundamental; (c) Espectrograma de banda larga com sobreposição dos formantes das áreas vozeadas.

TABELA I

RESULTADOS DOS CENTROS ($F1, F2$) OBTIDOS PELO MÉTODO DE AGRUPAMENTO.

Amostra	Formante	Centro 1	Centro 2	Centro 3	Centro 4
GC1	F1 (Hz)	516,6	480,5		
	F2 (Hz)	1209,5	2286,5		
GC2	F1 (Hz)	455,2	400,1	848,5	
	F2 (Hz)	1190,4	2355,1	1663,1	
GC3	F1 (Hz)	527,7	697,8	612,1	
	F2 (Hz)	1254,7	2450,5	1812,8	
GC4	F1 (Hz)	611,4	705,2	442,0	
	F2 (Hz)	1117,5	1630,0	2183,7	
GC5	F1 (Hz)	608,6	523,1	595,2	
	F2 (Hz)	1297,7	2569,5	1793,7	
GC6	F1 (Hz)	548,3	476,5	595,4	608,9
	F2 (Hz)	1066,1	2401,4	1383,7	1878,1
GC7	F1 (Hz)	508,1	401,7	733,7	654,5
	F2 (Hz)	1044,9	2293,5	1415,1	1892,5
GC8	F1 (Hz)	501,1	691,2	454,8	
	F2 (Hz)	1239,3	1805,5	2446,6	
GC9	F1 (Hz)	533,2	517,8	697,4	
	F2 (Hz)	1090,8	2449,7	1729,6	
GC10	F1 (Hz)	596,0	493,8	386,0	967,5
	F2 (Hz)	1178,3	1991,6	2539,5	1938,8

naqueles em que houve quatro centros). Não houve nenhum centro coincidente com as vogais posteriores alta ou média-alta (**/u/** e **/o/**).

Tal comportamento sugere que a articulação das vogais anteriores média-alta ou alta e da posterior média-baixa, para os falantes em questão, seriam mais estáveis. Contudo, ressalta-se que faz-se necessária a investigação mais pormenorizada de cada região para verificar influência de coarticulação, de fonemas nasais, assim como de fonemas vocálicos átonos.

TABELA II

REFERÊNCIA - VALORES DE REFERÊNCIA OBTIDOS EM PESQUISA REALIZADA PELO GRUPO DE ESTUDOS DE SONS DA FALA DA UTFPR [22].

Amostra	Formante	/i/	/e/	/ɛ/	/a/	/ɔ/	/o/	/u/
GC1	F1 (Hz)	354,5	397,5	628,0	996,3	623,0	407,0	381,8
	F2 (Hz)	2442,0	2413,3	1937,3	1373,5	1055,3	736,0	828,3
GC2	F1 (Hz)	327,5	434,0	703,0	1036,5	680,5	421,5	349,8
	F2 (Hz)	2286,5	1649,3	1755,5	1520,3	1041,3	882,5	941,5
GC3	F1 (Hz)	328,5	429,8	555,8	886,0	619,0	442,5	412,8
	F2 (Hz)	2314,5	2369,3	2055,0	1624,0	1187,3	912,5	1003,5
GC4	F1 (Hz)	362,3	544,3	709,0	906,8	653,8	459,0	420,0
	F2 (Hz)	2323,5	2118,0	1808,8	1386,0	1007,5	817,8	860,3
GC5	F1 (Hz)	321,3	433,8	609,0	750,5	658,3	430,5	335,0
	F2 (Hz)	2388,8	2175,8	1979,0	1533,3	1197,3	830,3	768,8
GC6	F1 (Hz)	389,3	435,5	631,3	760,3	579,3	437,5	388,5
	F2 (Hz)	2251,0	2326,5	2085,3	1440,8	1042,5	704,5	859,0
GC7	F1 (Hz)	330,8	472,0	655,5	877,5	663,8	431,5	378,3
	F2 (Hz)	2205,0	2098,3	1767,5	1455,0	1043,8	904,5	863,8
GC8	F1 (Hz)	370,0	474,8	673,5	1063,5	681,8	478,8	442,3
	F2 (Hz)	2453,8	2175,8	2095,5	1653,5	1136,3	867,0	986,5
GC9	F1 (Hz)	290,8	464,0	719,8	894,0	687,3	471,3	386,0
	F2 (Hz)	2305,5	2303,5	1594,3	1387,3	1050,3	895,3	840,8
GC10	F1 (Hz)	299,0	474,8	612,8	1021,3	614,3	403,8	385,5
	F2 (Hz)	2620,3	2303,0	2057,0	1768,3	1023,8	800,5	785,3

TABELA III

DISTÂNCIA ENTRE OS CENTROS OBTIDOS E O VALOR DE REFERÊNCIA MAIS PRÓXIMO.

Amostra	Formante	Centro 1	Centro 2	Centro 3	Centro 4
GC1	Referência	/ɔ/	/e/		
	$\Delta F1/F1$	20,6%	17,3%		
	$\Delta F2/F2$	12,8%	5,5%		
GC2	Referência	/ɔ/	/i/	/ɛ/	
	$\Delta F1/F1$	49,5%	18,1%	17,2%	
	$\Delta F2/F2$	12,5%	2,9%	5,6%	
GC3	Referência	/ɔ/	/e/	/ɛ/	
	$\Delta F1/F1$	17,3%	38,4%	9,2%	
	$\Delta F2/F2$	5,4%	3,3%	13,4%	
GC4	Referência	/ɔ/	/e/	/ɛ/	
	$\Delta F1/F1$	6,9%	23,1%	0,5%	
	$\Delta F2/F2$	9,8%	3,0%	11,0%	
GC5	Referência	/ɔ/	/i/	/ɛ/	
	$\Delta F1/F1$	8,2%	38,6%	2,3%	
	$\Delta F2/F2$	7,7%	7,0%	10,3%	
GC6	Referência	/ɔ/	/e/	/a/	/ɛ/
	$\Delta F1/F1$	5,7%	8,6%	27,7%	3,7%
	$\Delta F2/F2$	2,2%	3,1%	4,1%	11,0%
GC7	Referência	/ɔ/	/i/	/a/	/ɛ/
	$\Delta F1/F1$	30,6%	17,7%	19,6%	0,2%
	$\Delta F2/F2$	0,1%	3,9%	2,8%	6,6%
GC8	Referência	/ɔ/	/i/	/ɛ/	
	$\Delta F1/F1$	36,1%	18,6%	2,6%	
	$\Delta F2/F2$	8,3%	0,3%	16,1%	
GC9	Referência	/ɔ/	/e/	/ɛ/	
	$\Delta F1/F1$	28,9%	10,4%	3,2%	
	$\Delta F2/F2$	3,7%	6,0%	7,8%	
GC10	Referência	/ɔ/	/i/	/a/	/ɛ/
	$\Delta F1/F1$	3,1%	22,5%	5,6%	24,1%
	$\Delta F2/F2$	13,1%	3,2%	8,8%	3,3%

V. CONSIDERAÇÕES FINAIS

Os resultados preliminares obtidos foram promissores, com a identificação média de centros correspondentes a três vogais no espaço $F_1 \times F_2$, delineando um padrão próximo ao trapézio vocálico esperado.

Considerando as necessidades forenses, espera-se que o sistema proposto possa ser efetivamente utilizado como uma ferramenta de apoio em exames de registros de áudio, principalmente em exames de Comparação de Locutores. Apesar da aplicação estar restrita a amostras de áudio de apenas um falante, ou que contenham arquivos de delimitação entre os

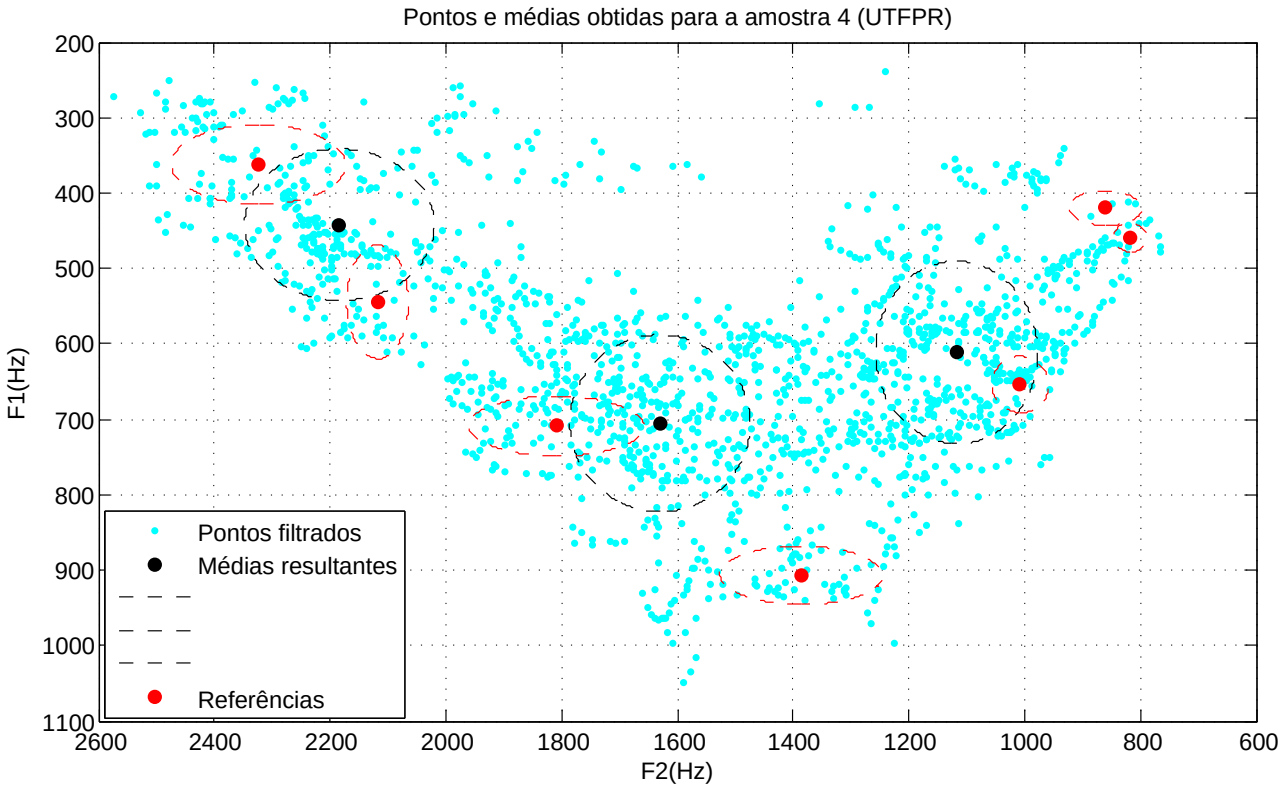


Fig. 6. Gráfico com pontos $F_1 \times F_2$ e centroides resultantes para a amostra 4 do grupo da UTFPR. Os valores de referência estão indicados pelos pontos vermelhos, enquanto que os centroides obtidos no trabalho estão indicados pelos pontos pretos. Os círculos maiores pontilhados representam a área das regiões obtidas.

TABELA IV

DESVIOS PADRÃO DOS VALORES PERTENCENTES À REGIÃO COM RELAÇÃO AO SEU CENTRO.

Amostra	Formante	Centro 1	Centro 2	Centro 3	Centro 4
GC1	F1 (Hz)	198,1	180,1		
	F2 (Hz)	221,0	301,7		
GC2	F1 (Hz)	172,5	129,2	140,9	
	F2 (Hz)	183,8	278,0	178,1	
GC3	F1 (Hz)	102,8	490,6	137,4	
	F2 (Hz)	137,9	156,8	167,9	
GC4	F1 (Hz)	121,0	117,2	100,4	
	F2 (Hz)	140,1	155,7	164,0	
GC5	F1 (Hz)	112,0	357,4	136,3	
	F2 (Hz)	153,7	202,8	148,0	
GC6	F1 (Hz)	64,4	160,7	114,1	131,5
	F2 (Hz)	107,8	118,2	104,8	134,3
GC7	F1 (Hz)	135,3	113,8	134,8	125,6
	F2 (Hz)	118,7	113,5	116,9	112,9
GC8	F1 (Hz)	97,9	224,3	152,7	
	F2 (Hz)	131,8	140,6	172,0	
GC9	F1 (Hz)	178,9	347,6	150,9	
	F2 (Hz)	159,1	207,4	226,3	
GC10	F1 (Hz)	183,0	130,9	82,2	184,9
	F2 (Hz)	205,2	166,3	139,4	235,1

TABELA V

DIFERENÇA PERCENTUAL DA DISTÂNCIA ENTRE A FRONTEIRA DE CADA REGIÃO E O VALOR DE REFERÊNCIA MAIS PRÓXIMO, QUANDO NÃO LOCALIZADO NO INTERIOR DA REGIÃO.

Amostra	Formante	Centro 1	Centro 2	Centro 3	Centro 4
GC1	F1	-	-		
	F2	-	-		
GC2	F1	-11,61%	-	-0,55%	
	F2	-	-	-	
GC3	F1	-	-	-	
	F2	-	-	-4,10%	
GC4	F1	-	-	-0,44%	
	F2	-	-1,41%	-	
GC5	F1	-	-	-	
	F2	-	-	-2,08%	
GC6	F1	-	-	-8,51%	-
	F2	-	-	-	-3,88%
GC7	F1	-4,00%	-	-1,23%	-
	F2	-	-	-	-0,64%
GC8	F1	-16,52%	-	-	
	F2	-	-8,27%	-	
GC9	F1	-	-	-	
	F2	-	-	-	
GC10	F1	-	-	-1,23%	-
	F2	-	-	-	-

turnos, acredita-se que terá utilidade em muitos casos.

Dadas as condições da maioria dos materiais encaminhados para esses exames, posteriormente será imprescindível a validação do mesmo também em condições degradadas, tais como: presença de ruído, compressões e limitações de frequência de canais telefônicos.

O presente trabalho ainda encontra-se em desenvolvimento, sendo que trabalhos futuros envolvem a integração com o software *R*, a possibilidade de interação com os gráficos para navegação dos trechos do áudio indicados através de pontos ou regiões selecionadas, a geração de relatórios e de registros de eventos (*logs*) para facilitar o elaboração de laudos e garantir a reprodutibilidade das etapas realizadas, assim como

o reconhecimento automático dos centroides equivalentes a cada vogal ou grupo de vogal (em função de suas classificações relativas à anterioridade e altura). Além dos valores de F_0, F_1 e F_2, o intuito é futuramente acrescentar outras dimensões para análise, tais como a banda de cada formante, e a variação de seus valores no decorrer da produção de um fonema.

É necessária ainda a realização de experimentos com diferentes durações de fala exclusiva, de um mesmo falante, para determinação da mínima duração a fim de se evidenciar graficamente o padrão formântico. Para tais ensaios serão utilizados os corpora *Spoltech* e "C-ORAL".

Também é importante permitir formas de validação dos resultados, como a possibilidade de treinamento ou de particionamento de uma amostra de áudio para verificar a coerência entre os resultados obtidos para cada segmento. O mecanismo desenvolvido deve propiciar análises de variações intra e inter-sujeito, importantíssimas de serem diferenciadas em exames de CL, permitindo que o examinador exclua elementos não-servíveis (isto é, que tenham grande variação intra-sujeito).

Posteriormente tais códigos serão migrados para rotinas do Praat ou outra linguagem que permita que todos os pacotes programados estejam em plataforma de software livre. A finalização de um ambiente de testes requer ainda que o examinador possa salvar um projeto com configurações realizadas, e a disponibilização de relatórios com as rotinas executadas e parâmetros utilizados.

Agradecimentos

Os autores agradecem a Denise de Oliveira Carneiro e Marilisa Exter Koslovski, peritas criminais do Instituto de Criminalística do Paraná, pelas importantes discussões acerca de ferramentas úteis para apoio ao exame de Comparação Forense de Locutores, a Eduardo Tondin Ferreira Dias e Philipe Ambrózio Dias, colegas do Laboratório de Processamento de Imagens e Sinais da UTFPR, pelas críticas e sugestões para melhoria do conteúdo do presente artigo, e ainda ao Grupo de Estudos dos Sons da Fala da UTFPR, liderado pela professora Maria Lúcia de Castro Gomes, pela cessão de amostras de áudio utilizadas neste trabalho.

Referências

[1] Brasil, "Lei nº 9.296 (lei das interceptações telefônicas), de 24 de julho de 1996." *Diário Oficial da República Federativa do Brasil*, 1996.

[2] A. C. M. Braid, *Fonética Forense*, 2nd ed., ser. Tratado de Perícias Criminalísticas. Campinas, SP: Editora Millenium, 2003.

[3] SENASP, "Diagnóstico da perícia criminal no brasil," Secretaria Nacional de Segurança Pública, Tech. Rep., 2012.

[4] ENASP, "Relatório nacional da execução da meta 2: um diagnóstico da investigação de homicídios no país," Conselho Nacional do Ministério Público, Tech. Rep., 2012.

[5] J. D. Vargas, I. Blavatsky, and L. M. L. Ribeiro, "Metodologia de tratamento do tempo e da morosidade processual na justiça criminal," Secretaria Nacional de Segurança Pública, Tech. Rep., 2006.

[6] A. Vrubel, A. Brondani, M. Silva, and L. Grochocki, "Modelo matemático para a gestão de recursos humanos baseados em controles estatísticos de demanda e produtividade," *Anais do VI Congresso CONSAD de Gestão Pública*, 2013.

[7] M. L. C. Gomes, L. Richert, and J. Malakoski, "Identificação de locutor na área forense: a importância da pesquisa interdisciplinar," in *Anais do X ENCONTRO DO CELSUL*, Cascavel, 2012.

[8] E. Gold and P. French, "International practices in forensic speaker comparison," *The International Journal of Speech, Language and the Law*, vol. 18, pp. 293–307, 2011.

[9] P. Boersma and D. Weenink, "Praat, doing phonetics by computer (version 5.4.08)," 2015. [Online]. Available: http://www.praat.org/

[10] C. Collomb, "Burg's method, algorithm and recursion," 2009. [Online]. Available: http://ccollomb.free.fr/

[11] P. Escudero, P. Boersma, A. S. Rauber, and R. A. H. Bion, "A cross-dialect acoustic description of vowels: Brazilian and european portuguese," *Journal of the Acoustical Society of America*, vol. 126, pp. 1379–1393, 2009.

[12] R. Core-Team, "R: A language and environment for statistical computing," 2013. [Online]. Available: http://www.R-project.org/

[13] I. Russo and M. Behlau, *Percepção da Fala: Análise Acústica do Português Brasileiro*. Editora Lovise, 1993.

[14] L. R. Rabiner and R. W. Schafer, *Theory and Applications of Digital Speech Processing*. Pearson, 2011.

[15] J. R. Deller Jr., J. H. L. Hansen, and J. G. Proakis, *Discrete-Time Processing of Speech Signals*. Wiley-IEEE Press, 2000.

[16] A. P. P. F. Engelbert, *Fonética e Fonologia da Língua Portuguesa*. Curitiba: Ibpex, 2011.

[17] T. Cristófaro-Silva, *Dicionário de Fonética e Fonologia*. Editora Contexto, 2011.

[18] T. Cristófaro-Silva and H. C. Yehia, *Sonoridade em Artes, Saúde e Tecnologia*. Belo Horizonte: Faculdade de Letras, 2012. [Online]. Available: http://fonologia.org

[19] T. Shimamura and H. Kobayashi, "Weighted autocorrelation for pitch extraction of noisy speech," *IEEE Transactions on Speech and Audio Processing*, vol. 9, pp. 727–730, 2001.

[20] P. Boersma, "Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound," *IFA Proceedings*, vol. 17, 1993.

[21] L. M. J. Barbosa, *Processamento de Sinais em Fonética Forense*, Departamento da Polícia Federal, 2012.

[22] M. L. C. Gomes, "An acoustic description of vowels brazilian portuguese in normal and disguised voice," in *IAFPA 2013 Annual Conference*, 2013.

[23] M. C. Schramm, L. F. R. Freitas, A. Zanuz, and D. Barone, "A brazilian portuguese language corpus development." in *International Conference on Spoken Language Processing 2000*. ISCA, 2000.

[24] T. Raso and H. Mello, *C-ORAL BRASIL I - Corpus de referência do português brasileiro falado informal*. Belo Horizonte, MG: Editora UFMG, 2012.

DOI: 10.5769/C2015004 or http://dx.doi.org/10.5769/C2015004

Continuous Authentication via Localization Using Triangulation of Directions of Arrival of Line of Sight Components

Marco A. M. Marinho, Paulo Roberto de Lira Gondim, and João Paulo C. L. da Costa

***Abstract*—A larger number of users work from a desktop computer and use their smartphones, tablets, and home computers to communicate, buy, organize, and store sensitive information. With the growth of the adoption of the Internet for tasks such as online banking and shopping, an increased focus has been given on the development of tools that enable secure transactions.**

This manuscript proposes the usage of direction of arrival estimation tools to provide continuous authentication. The location of a user within the network can be estimated by using triangulation of the user's wireless signal. The location estimates can be used to track a user's movement within a wireless network. The movement pattern can then be analyzed for possible indicators of fraud.

***Index Terms*—Continuous authentication, DOA estimation, MIMO**

I. Introduction

Verifying one's identity electronically has become the focus of extensive research. Not only does the user need to be authenticated to use a given system, but also the system itself so that the user can trust it. A picture of the importance of electronic authentication in the recent landscape is given by the revenue lost by companies due to Internet fraud. According to [1], approximately 3.4 billion dollars were lost in the year of 2011 due to on-line fraud. Therefore, considerable attention has been devoted by the scientific community to the development of new ways of improving security at every part involved in Internet transactions.

Most of the systems that rely on electronic authentication verify the user's identity in a single authentication step and then allow them to freely use the system either until they log out or for a given amount of time when they must be re-authenticated. The process of constant re-authentication in a system is known as continuous authentication. For systems that rely on continuous authentication users must constantly prove to the system who they are in order to continue operating it. Although these types of re-authentication improve security, they will most likely result in a negative impact on the perception a user has of the ease-of-use of a system.

Most of the modern user's systems are connected to core networks by wireless access networks. Such networks have become omnipresent in today's large cities and most of today's workplaces. They enable user mobility while delivering satisfactory data rates, and can be an economic choice in comparison to cabling of an entire floor or building. To keep up with the demand for higher networking speeds, most wireless networking standards have adopted the Multiple Input Multiple Output (MIMO) technology. Systems that employ MIMO also provide the physical requirements for the application of mathematical tools of array signal processing.

Array signal processing has developed many techniques towards the estimation of the direction of arrival (DOA) of a radio signal. Knowledge of the DOA offers outstanding benefits, such as spatial filtering. We refer here to many important DOA estimation methods, such as Iterative Quadratic Maximum Likelihood (IQML) [2], Root-WSF [3] and Root-MUSIC [4], Expectation Maximization (EM) [5], [6], [7], [8], the space Alternating Generalized Expectation Maximization (SAGE) [9], [10], [11] and Estimation of Signal Parameters via Rotational Invariance (ESPRIT) [12], which can be applied.

This manuscript proposes taking advantage of using components present at modern MIMO wireless communication systems for the estimation of the user location by means of DOA estimation. By obtaining the location of the user when he/she first authenticates, for example, using biometrics or a password, it is possible to enforce authentication on multiple levels.

The remainder of this work is divided into five sections. In Section II the problem of DOA estimation is detailed and explained. The localization of a user within a network is shown in Section III. In Section IV three ways of employing the estimated location to user authentication are discussed. In Section V numerical simulations are presented considering mobility models and the average location error as a metric. Finally, in Section VI conclusions are drawn.

II. Direction of Arrival Estimation

This section describes the steps involved in estimating the DOAs of a set of received signals in an antenna array. Subsection II-A details the data model. In Subsection II-B the estimation of the DOAs using the ESPRIT algorithm is presented.

A. Data Model

The baseband signal received at the m-th antenna of an antenna array composed of M antennas at time snapshot t

Marco Antonio Marques Marinho, Paulo Roberto de Lira Gondim, and João Paulo Carvalho Lustosa da Costa, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília-DF, Brasil, E-mails: marco.marinho@ieee.org, pgondim@ene.unb.br, joaopaulo.dacosta@ene.unb.br.

can be represented as

$$x_m(t) = \sum_{i=1}^{d} s_i(t) e^{j\cdot(m-1)\cdot\mu(\theta_i)} + n_m(t), \quad (1)$$

where $s_i(t)$ is the complex symbol transmitted by the i-th source at time snapshot t and $n(t)$ is the zero mean circularly symmetric (ZMCS) additive white Gaussian noise present at the antenna m at time snapshot t. $\mu(\theta_i)$ represents the spatial frequency of the signal transmitted by the i-th source. For this work the spatial frequencies of a signal impinging over the uniform linear array (ULA) are given by

$$\mu(\theta_i) = 2\pi \frac{\Delta}{\lambda} \cos(\theta_i), \quad (2)$$

where θ_i is the direction of arrival of the i-th signal, Δ is the separation between the antenna elements and λ is the wavelength of the incoming signal.

Equation (1) can be rewritten in matrix notation as

$$\mathbf{X} = \boldsymbol{AS} + \boldsymbol{N} \in \mathbb{C}^{M\times N}, \quad (3)$$

where $\boldsymbol{S} \in \mathbb{C}^{d\times N}$ is the matrix containing the N symbols transmitted by each of the d sources, $\boldsymbol{N} \in \mathbb{C}^{M\times N}$ is the noise matrix with its entries drawn from $\mathcal{CN}(0, \sigma_n^2)$, and

$$\boldsymbol{A} = [\boldsymbol{a}(\theta_1), \boldsymbol{a}(\theta_2), ..., \boldsymbol{a}(\theta_d)] \in \mathbb{C}^{M\times d}, \quad (4)$$

where θ_i is the azimuth angle of the $i-$th signal and $\boldsymbol{a}(\theta_i) \in \mathbb{C}^{M\times 1}$ is the array response, obtained by measurements, whose elements are $e^{j\cdot(m-1)\cdot\mu(\theta_i)}$.

The received signal covariance matrix $\boldsymbol{R_{XX}} \in \mathbb{C}^{M\times M}$ is given by

$$\boldsymbol{R_{XX}} = \mathrm{E}\{\boldsymbol{XX}^{\mathrm{H}}\} = \boldsymbol{AR_{SS}A}^{\mathrm{H}} + \boldsymbol{R_{NN}}, \quad (5)$$

where $(\centerdot)^{\mathrm{H}}$ stands for the conjugate transposition, and

$$\boldsymbol{R_{SS}} = \begin{bmatrix} \sigma_1^2 & \gamma_{1,2}\sigma_1\sigma_2 & \cdots & \gamma_{1,d}\sigma_1\sigma_d \\ \gamma_{1,2}^*\sigma_1\sigma_2 & \sigma_2^2 & & \vdots \\ \vdots & & \ddots & \\ \gamma_{1,d}^*\sigma_1\sigma_d & \gamma_{2,d}^*\sigma_2\sigma_d & \cdots & \sigma_d^2 \end{bmatrix}, \quad (6)$$

where σ_i^2 is the power of the $i-$th signal and $\gamma_{a,b} \in \mathbb{C}$, $|\gamma_{a,b}| \leq 1$ is the cross correlation coefficient between signals a and b. $\boldsymbol{R_{NN}} \in \mathbb{C}^{M\times M}$ is a matrix with σ_n^2 over its diagonal and zeros elsewhere. An estimate of the signal covariance matrix can be obtained by

$$\hat{\boldsymbol{R}}_{\boldsymbol{XX}} = \frac{\boldsymbol{XX}^{\mathrm{H}}}{N}. \quad (7)$$

B. ESPRIT

For DOA estimation this works uses the ESPRIT method since it is a closed form algorithm that can be very easily extended to multidimensional scenarios.

The ESPRIT parameter estimation technique is based on subspace decomposition. Matrix subspace decomposition is usually done by applying the Singular Value Decomposition (SVD). The SVD of the matrix $\boldsymbol{X} \in \mathbb{C}^{M\times N}$ is given by

$$\boldsymbol{X} = \boldsymbol{U\Lambda V}^{\mathrm{H}}, \quad (8)$$

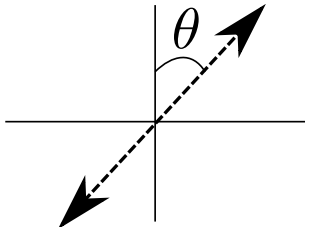


Figure 1. Depiction of possible ambiguity in signal propagation direction.

where $\boldsymbol{U} \in \mathbb{C}^{M\times M}$ and $\boldsymbol{V}^{N\times N}$ are unitary matrices called the left-singular vectors and right-singular vectors of $\boldsymbol{X}$ and $\boldsymbol{\Lambda} \in \mathbb{C}^{M\times N}$ is pseudo diagonal matrix containing the singular values of $\boldsymbol{X}$. The signal subspace $\boldsymbol{E}_S \in \mathbb{C}^{M\times \hat{L}}$ of $\boldsymbol{X}$ can be constructed by selecting only the left singular vectors related to the d largest singular values. The remaining singular vectors form the noise subspace $\boldsymbol{E}_N \in \mathbb{C}^{M\times M-d}$ of $\boldsymbol{X}$.

Equivalently, eigenvalue decomposition can be applied on the auto correlation matrix $\hat{\boldsymbol{R}}_{\boldsymbol{XX}}$ of $\boldsymbol{X}$ spanning the same subspace

$$\hat{\boldsymbol{R}}_{\boldsymbol{XX}} = \boldsymbol{E\Sigma E}^{-1}, \quad (9)$$

where $\boldsymbol{E} \in \mathbb{C}^{M\times M}$ and $\boldsymbol{\Sigma} \in \mathbb{C}^{M\times M}$ contains the eigenvectors and eigenvalues of $\boldsymbol{R}_{\boldsymbol{XX}}$. The eigenvectors related to the $\hat{L}$ largest eigenvalues span the same signal subspace $\boldsymbol{E}_S$ of the single value decomposition. The same holds for the noise subspace of the EVD and left singular vectors of the SVD, $\boldsymbol{E}_N$. With this subspace estimate at hand the Total Least Squares (TLS) ESPRIT [12] is applied.

The high accuracy provided by the ESPRIT algorithm is capable of yielding very precise results for the position estimation.

For multidimensional arrays another option is to employ methods based on the PARAFAC decomposition such as [13] [14] instead of ESPRIT.

III. Localization

It is important to notice that the DOA is given with respect to the reference of the $\boldsymbol{x}-$axis and cannot distinguish between front or back. Figure 1 displays this ambiguity.

The result is that each sensor possesses an estimated line in the ground plane where the transmitting node may be located. However, the acquisition of a set of line estimates enables obtaining a single estimate of the transmitting user localization. Figure 2 shows an example of imprecise estimates from three receiving nodes being used to estimate the position of the transmitter node. The problem is reduced to the least squares problem of finding the point of minimum distance from any of the possible combination of line estimates.

By writing the representing the line estimates as line equations of the type $Ax + By + C = 0$ in the sensor coordinate system, an estimate of the sensor position is given by

$$\{\hat{x}_0, \hat{y}_0\} = \min_p \frac{|A_{p_1}x_0 + B_{p_1}y_0 + C_{p_1}|}{\sqrt{A_{p_1} + B_{p_1}}} + \frac{|A_{p_2}x_0 + B_{p_2}y_0 + C_{p_2}|}{\sqrt{A_{p_2} + B_{p_2}}} + ..., \quad (10)$$

where p is an index set containing the possible combinations of estimated lines. While more than three sensors can be used to obtain increased accuracy, it also results in a higher

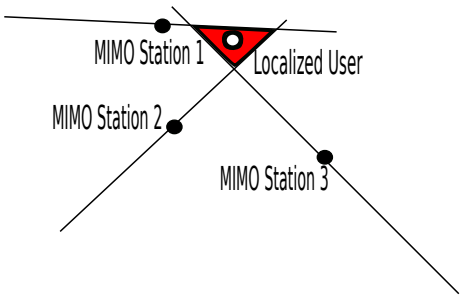


Figure 2. Example of a sensor triangulation using only the DOAs of the reference nodes.

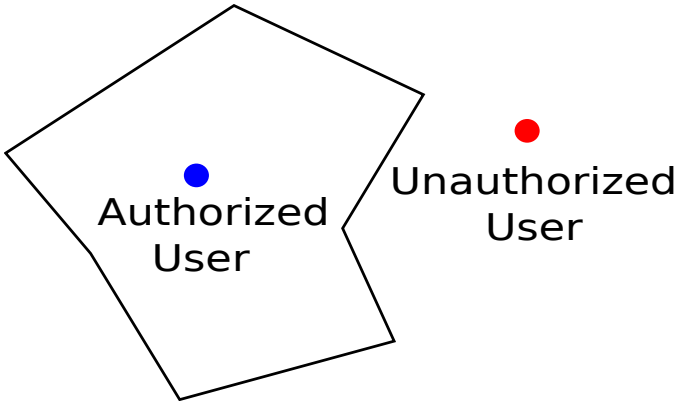


Figure 3. Example of border control via user localization.

computational load involved in the calculation of the minima. Once the lines have been choosen the final location estimate $S_0 = [x_0, y_0]$ can be found by taking the derivative of the above with respect to x_0 and y_0 and finding the point where it is equal to zero.

Furthermore, this technique may be used in conjunction with other localization methods such as the received signal strength indicator (RSSI). A set of candidate locations can be selected to choose the candidate that best fits the RSSI information.

IV. Authentication Mechanism

This section presents ways of using an estimation of the user's position for continuous authentication. Subsection IV-A shows how position estimation can be used to assure that a user only accesses a system when he/she is in an authorized area. In Subsection IV-B the position estimation is used to ensure a user has a speed compatible with network access metrics. Finally, Subsection IV-C shows how postion estimation can be used together with behavioaral movement metrics to provide continuous authentication.

A. Border Enforcing

Depending on the type of information being accessed, it may be important to enforce that the user only accesses a certain system or information if he/she is within a given area or set of areas. For instance, such approach could be beneficial to ensure that a user accessing a system is being monitored by a system of cameras. Thus, an unauthorized access could be registered on tape for further inquiry in the future. This is the simplest method for providing continuous authentication since there is only a single metric: the user is within the authorized access area. Figure 3 shows an example of an area constituted by a single polygon.

While the problem of verifying whether a point is inside a polygon or not has been thoroughly studied, the solution, in the case of irregular polygons, is resorting to ray tracing. Ray tracing may become complex depending on the nature and number of polygons, and the computational load can easily grow if the system needs to ensure proper location for a large number of users. However, since such calculations can be performed in a centralized authentication structure, large computational capacity can be provided.

As an example application, let us consider that a cellular system is divided into switching centers, and each switching center is divided into location areas, used for location of mobile terminals, when a mobile terminating call is presented to the cellular system. Each mobile terminal has some identities, e.g, International Mobile Subscriber Identity (IMSI) and Temporary MSI (TMSI). There is an association between location and security functions, since for each Location Area (LA) there is an assignment of a TMSI. In this sense, for each LA transition that occurs during the terminal movement, a new TMSI is assigned. Thus it is possible to reduce the area to be searched for localization purposes, limiting the search to cells included in the current LA.

B. Verification of Movement Speed

In networks that are not encrypted the problem of spoofing, i.e when a user pretending to be someone else, appears. Even when the system is encrypted, if the security key has been compromised an adversary may act as a spoofer in the network. This is a critical security problem and can be avoided by using the proposed method for user localization.

When a user first accesses the system or data he/she performs a standard authentication method to prove his/her identity to the system. During the process of authentication the users location is estimated by the system. The user must then transmit a set of pilot symbols at a fixed time interval so that his location can be reestimated at each transaction. By checking if the user is moving at a reasonable speed, the system can ensure that the one transmitting is still the original user who was authenticated.

Since, for this type of security, we assume that the system is either not encrypted or that the encryption key has been compromised, once the system detects an adversary operating as a spoofer, it must then warn the user of the presence of the spoofer, and, provide the estimated localization of the adversary to the authority in charge. Thus, in the case of an encrypted system, the user can obtain a new encryption key, and in unencrypted networks, the transactions are halted until the spoofer has been dealt with, avoiding any potential danger.

By measuring the distance between the current estimate A and the previous estimate B with

$$D_{\overrightarrow{AB}} = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}, \qquad (11)$$

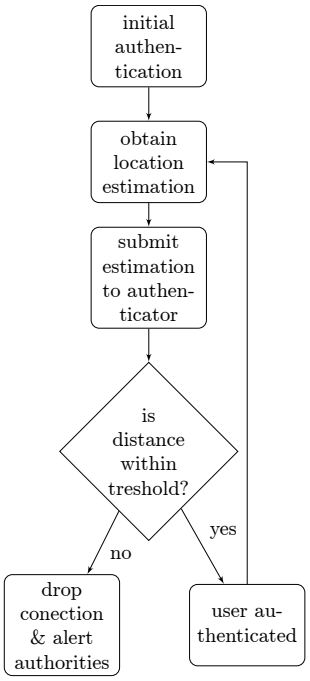


Figure 4. Flowchart of the proposed authentication method.

and defining a distance threshold between estimates it is possible to identify spoofers. Figure 4 shows a flowchart of the proposed authentication method. This simple estimation method can properly identify intruders posing as authorized users in MIMO wireless networks.

C. Analysis of Behavioral Movement

The last proposed method for providing continuous authentication while relying on the estimation of a user location analyzes the user movement behavior over the network coverage. Consider, for instance, a working environment. Most people have a frequent routine within their workspace. They remain seated in their workspace most of their working hours, they may approach their nearby colleagues at a given frequency, go to the bathroom, to the lunchroom, and visit the manager's desk. All this information defines the behavior of a person.

The first part of this approach is to obtain a statistical model of a person's movement within the coverage of a wireless network. This statistical model can be obtained, for instance, as a heat-map of a discretized grid of the coverage of the wireless network. This heat-map would represent a statistical model of the placement behavior of a given user. Figure 5 shows an example of the heat-map of a person that remains at a desk within a room. The comparison between the stored statistical model ant the heat-map obtained from a user during the day can provide the means of authenticating the user.

Although this approach requires a training stage, it is, possibly, the one of capable of enforcing the most efficient authentication. If a person's movement within a given space is enough to provide unique identification, this approach provides unique identification without requiring any user's input.

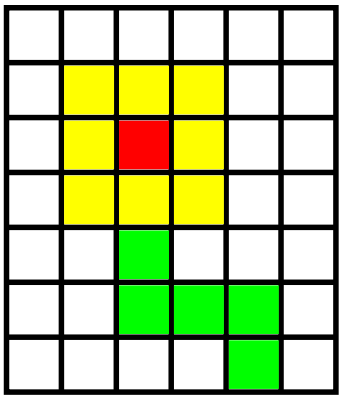


Figure 5. Example of movement heat-map

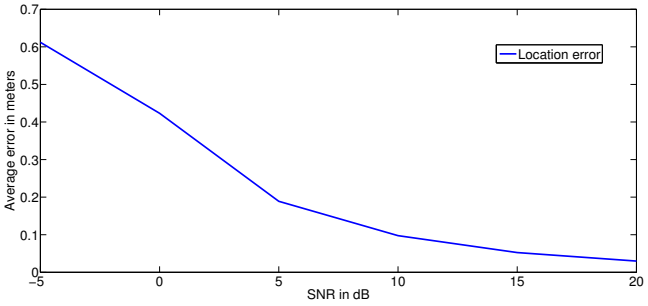


Figure 6. Average location error using the DOA based triangulation technique.

V. Numerical Simulations

In order to study the performance of the proposed solution in the presence of noise a set of numerical simulations was performed. The scenario tries to simulate an area of 45×45 m covered by three WiFi hotspots equipped with three antennas each. The number of snapshots used for the DOA estimation at each hotspot is $N = 50$. The snapshots used for the DOA estimation do not depend on the symbols transmitted over the 802.11 protocol, since the DOA estimation depends only on the narrow-band carrier signal.

We analyze the average location error for the given scenario in different SNR conditions. To perform this simulation a user is placed at a random location within the simulation area and its location is estimated. Figure 6 shows the average location error for 10000 random points.

For a SNR of 20 dB the technique proposed in Subsection IV-A is precise to within approximately 3 cm, i.e, the method can provide precise and online border enforcing for wireless system users under the simulated scenario. The precision of the method varies according to the number of WiFi hotspots and antennas present at each hotspot. The results shown in Figure 6 also validate the approach proposed in Subsection IV-B.

VI. Conclusion

This manuscript addressed an initial discussion on possible methods for providing continuous authentication while relying on DOA estimation in MIMO wireless networks. The mathematical tools for providing a precise DOA estimation

in antenna arrays were presented. Once a DOA estimate has been obtained, the system can then estimate the user's location within the network coverage.

By using the estimated user position, a set of rules can be enforced for the sake of obtaining a robust authentication system. The first set of rules regards the containment of the user within a set of boundaries. This can be easily implemented with the estimate of the users location. The second set regards the analyzes of the movement speed of the user within the network coverage. If the speed exceeds a certain value, the system considers the transaction is no longer safe due to the possible existence of a spoofer within the network. The third set of rules is the most complex, yet the one of highest potential for continuous authentication. By analyzing the behavior of a user's movement within the network, the authentication system could measure the difference between the momentary movement of the user and the stored behavior. If the user's movement is not similar enough to the stored behavior, the authentication fails. This is interesting, for instance, when a smartphone is lost or stolen. A user that has taken the smartphone will probably be distant from the owner of the phone. The system is then notified that the actions taken at the device are suspicious and should probably not be authorized.

The methods proposed in this work are still hard to be implemented in real systems, since the application of MIMO is in its infancy. The goal of the work is to lay down the foundation for providing continuous authentication based on user location inside a wireless network. Although it may not be efficient enough to provide precise authentication by itself, it can be a useful tool for improving the accuracy of a multimodal continuous authentication system that relies on other inputs, such as movement gait.

Acknowledgments

The authors wish to thank the Brazilian research and innovation Agencies CAPES (FORTE Project, CAPES Forensic Science Notice 25/2014) and FINEP (Grant RENASIC/PROTO 01.12.0555.00), for their support to this work.

References

[1] CyberSource, "2012 online fraud report," Tech. Rep., 2012.

[2] Y. Bresler and A. Macovski, "Exact Maximum Likelihood Estimation of Superimposed Exponentials Signals in Noise," *IEEE ASSP Magazine*, vol. 34, pp. 1081–189, 1986.

[3] P. Stoica and A. Nehorai, "A Novel Eigenanalysis Method for Direction Estimation," in *Proceedings IEEE F.*, 1990.

[4] A. J. Barabell, "Improving the Resolution Performance of Eigenstructured Based Direction-Finding Algorithms," in *Proceedings of ICASSP 83*, 1983.

[5] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statistical Soc. B.*, vol. 39, no. 1, 1977.

[6] T. K. Moon, "The Expectation-Maximization Algorithm," *IEEE Signal Processing Magazine*, November 1996.

[7] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. John Wiley & Sons, Inc., New York, 1997.

[8] M. Miller and D. Fuhrmann, "Maximum-Likelihood Narrow-Band Direction Finding and the EM Algorithm," *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. 38, pp. 1560–1577, 1990.

[9] J. A. Fessler and A. O. Hero, "Space-Alternating Generalized Expectation-Maximization Algorithm," *IEEE Transactions on Signal Processing*, vol. 42, no. 10, October 1994.

[10] F. A. Dietrich, "A Tutorial on Channel Estimation with SAGE," *Technical Report TUM-LNS-TR-06-03*, 2006.

[11] F. Antreich, J. Nossek, G. Seco-Granados, and A. Swindlehurst, "The Extended Invariance Principle for Signal Parameter Estimation in an Unknown Spatial Field," *IEEE Transactions on Signal Processing*, vol. 59, no. 7, pp. 3213–3225, July 2011.

[12] R. Roy and T. Kailath, "ESPRIT - estimation of signal parameters via rotation invariance techniques," *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. 17, 1989.

[13] J. P. C. L. da Costa, D. Schulz, F. Roemer, M. Haardt, and J. A. A. Jr., "Robust R-D Parameter Estimation via Closed-Form PARAFAC in Kronecker Colored Environments," in *Proc. 7-th International Symposium on Wireless Communications Systems (ISWCS 2010)*, 2010.

[14] J. P. C. L. da Costa, F. Roemer, M. Weis, and M. Haard, "Robust R-D parameter estimation via closed-form PARAFAC," in *Proc. ITG Workshop on Smart Antennas (WSA'10)*, 2010.

DOI: 10.5769/C2015005 or http://dx.doi.org/10.5769/C2015005

Extração de dados em *smartphones* com sistema Android usando substituição da partição *recovery*

Sibelius Lellis Vieira e Adriano Rodrigues da Cruz

Resumo—**Os *smartphones* são telefones celulares que podem armazenar dados de grande valia para investigação de crimes. Contudo, o processo de aquisição de dados destes aparelhos pode ser inviável quando estes estiverem bloqueados e com a função de depuração USB desabilitada. O objetivo deste trabalho é descrever uma técnica e os procedimentos associados que podem ser aplicados na aquisição de dados em *smartphones* que utilizam o sistema operacional Android, estão bloqueados e com a função de depuração USB desabilitada. Ao final, ilustra-se o método de extração através da substituição da partição *recovery* e os resultados dos testes realizados com este método**.

Palavras-Chave—Evidências Digitais, Smartphone, Android, Extração de dados.

***Abstract*—Smartphones are mobile phones that store valuable in a crime investigation. However, the process of data acquisition can be unfeasible when the devices are locked and the USB debugging is disabled. This work describes a technique and its procedures that can be applied to mobile data acquisition in which the Android operating system is used, the mobile is locked and its USB debugging is disabled. At the end, we present a method of data extraction by replacing the recovery partition and the results of tests that employed this method.**

Keywords—Digital evidence, Smartphone, Android, Data extraction.

I. INTRODUÇÃO

A computação forense é uma área da computação científica cujo objetivo é examinar dispositivos computacionais com a intenção de identificar, preservar, recuperar e apresentar evidências digitais que possam ser úteis para tipificação de crimes. Dentre estes dispositivos, destacam-se os computadores, *notebooks*, *laptops*, *tablets*, telefones celulares, máquinas fotográficas entre outros. De acordo com a estimativa da International Communication Union (ITU), o número de linhas ativas de telefones celulares, em 2014, é de quase 7 bilhões [1]. Isto torna o telefone celular um importante alvo da análise forense, uma vez que pode ser utilizado como meio para o ilícito penal (envio de mensagens de ameaça, calúnia, armazenamento de imagens relacionadas à pedofilia, etc), ou mesmo em crimes informáticos próprios [2].

Existem telefones celulares de diversas marcas e modelos. Os aparelhos construídos com maior poder de processamento e conectividade são denominados *smartphones*. Vários sistemas operacionais foram desenvolvidos para serem utilizados pelos *smartphones*. Os mais populares são o Android, iOS, Symbian, Series 40, BlackBerry, Samsung e Windows [3].

A. *Definição do problema*

O Android se tornou o sistema operacional móvel mais popular do mundo no começo de 2011 [4]. Desta forma, é natural que a quantidade de aparelhos apreendidos para perícia com este sistema também seja proporcionalmente grande. Um dos recursos de segurança que o Android possui é permitir o bloqueio da tela do aparelho. Esse bloqueio pode ser feito de diferentes maneiras, tais como: senha numérica, senha alfanumérica, padrões etc. Outro recurso é a ativação da depuração USB, utilizada pelos desenvolvedores para acesso ao telefone através do PC durante a depuração de aplicativos e utilizada também pelos peritos criminais, para extração dos dados do usuário.

Após a apreensão de um aparelho celular, o primeiro passo para uma análise pericial é realizar a extração dos dados do aparelho para um computador, de forma a preservar o artefato original e não comprometer a integridade dos dados extraídos. A extração de dados de *smartphones* pode ser feita de forma física ou de forma lógica. A extração física é mais complexa, pois envolve *hardwares* especiais e conhecimento em eletrônica [4]. A extração lógica é feita com o uso de *softwares* que se conectam ao aparelho através do Android Debug Bridge (ADB), serviço que é executado no Android quando a função de depuração USB está ativada. Existem diversas técnicas e ferramentas para extração dos dados através da depuração USB. No entanto, quando esta opção está desabilitada e não é possível habilitá-la, o trabalho do perito é dificultado, podendo tornar a extração inviável [5].

Visto que não é possível acessar o ADB se a função de depuração USB estiver desativada, o primeiro procedimento a ser adotado pelo perito é a ativação desta opção. Contudo, em alguns casos isto não é possível. Por exemplo, se o aparelho apreendido possui bloqueio de tela ativo e a senha padrão ou PIN de desbloqueio não for conhecido, o perito fica impossibilitado de se conectar ao ADB e não é possível ter acesso ao menu do sistema para habilitar a depuração USB.

B. *Objetivo geral*

O objetivo geral deste trabalho é analisar, propor e testar um método para extração de dados de *Smartphones* de diversas marcas, que se enquadram no cenário mencionado anteriormente, a saber, Sistema Operacional Android, com bloqueio de tela ativado e a opção de depuração USB desabilitada.

Sibelius Lellis Vieira, Professor do Departamento de Computação, PUC GOIAS e Perito Criminal do Estado de Goiás, Goiânia-GO, Brasil, sibelius@pucgoias.edu.br, e Adriano Rodrigues da Cruz, Bacharel em Ciência da Computação pela PUC GOIAS, Programador Sênior CTI/SENAC-GO, Goiânia, GO, Brasil, adrianohck@gmail.com.

II. REFERENCIAL TEÓRICO

A. *Computação forense*

A computação forense tem como objetivo desenvolver técnicas e ferramentas para a investigação e análise de potenciais evidências digitais [6]. Tais técnicas e ferramentas podem ser empregadas na investigação e tipificação de crimes que envolvam dispositivos computacionais. Entende-se por dispositivos computacionais qualquer aparelho capaz de processar informação, ou seja, a computação forense pode ser utilizada na investigação de computadores, *notebooks*, *laptops*, celulares, *tablets*, máquinas fotográficas digitais, televisores digitais etc.

Uma evidência digital é qualquer informação, armazenada ou transmitida por um dispositivo computacional, que pode ser utilizada como prova em um processo judicial para tipificar um crime ou estabelecer uma ligação entre um crime e sua vítima ou um crime e seu autor [7].

B. *Android OS*

O Android é um sistema operacional móvel de código aberto, baseado no *kernel* 2.6 do Linux e gerenciado pela Open Handset Alliance, um grupo de empresas de tecnologia lideradas pelo Google. Este sistema está presente principalmente em telefones celulares. Porém, também é possível encontrá-lo em *tablets*, mini-PCs, televisores e GPS. No começo de 2011, se tornou o sistema operacional mais popular do mundo para celulares [4].

É por meio dos aplicativos que o Android oferece funcionalidades para o usuário do celular. Existem vários tipos de aplicativos, tais como jogos, redes sociais, organizadores pessoais, calendários etc. De fato, até as funcionalidades básicas do celular, tais como enviar e receber mensagens e originar e receber ligações, são aplicativos [8].

Alguns aplicativos armazenam dados do usuário. O aplicativo de telefone, por exemplo, armazena as chamadas originadas e recebidas e duração das mesmas. O aplicativo de mensagens SMS armazena as mensagens enviadas e recebidas pelo usuário [8]. Estes dados podem ser utilizados pelo perito forense na elaboração de um laudo pericial, por exemplo, e são armazenados basicamente em dois locais: interno e externo [4].

O armazenamento externo dos dados é feito por Secure Digital Card (cartão SD), geralmente formatado com o sistema de arquivos Microsoft FAT32 [4], o que facilita o trabalho do perito, uma vez que o cartão SD pode ser removido e analisado em outra máquina.

Os dados são armazenados internamente em um *chip* de memória *flash*. Além de dados de usuário, a memória também armazena arquivos de sistema. O armazenamento interno é gerenciado pela Application Program Interface (API) do Android e segue uma estrutura pré-determinada. Assim que os aplicativos são instalados, uma pasta para o aplicativo é criada no subdiretório /data/data. O navegador padrão do Android, por exemplo, armazena os dados no subdiretório /data/data/com.android.browser [4].

O Android Software Development Kit (SDK) é um conjunto de bibliotecas, documentos, utilitários e compiladores necessários para a codificação, compilação, teste e distribuição de aplicativos. O SDK contém, por exemplo, o utilitário adb usado para depuração e o utilitário fastboot, utilizado para *flash* de partições.

O SDK do Android permite que os desenvolvedores criem banco de dados SQLite para os aplicativos. O SQLite é um banco de dados leve, pequeno, de código fonte aberto e que possui as características básicas, tais como tabelas, gatilhos e visões, necessárias para a estruturação de dados [9]. Outra característica é que todos os dados são armazenados em um único arquivo *cross-platform*, ou seja, o arquivo de dados pode ser lido tanto na implementação do SQLite para Android quanto na implementação para Windows.

Estes bancos de dados são armazenados normalmente no subdiretório /data/data/<app>/databases [4]. A análise destes bancos de dados de aplicativos como telefone e mensagens permite que o perito identifique, por exemplo, chamadas originadas para determinado número ou troca de mensagens suspeitas.

C. *Android Debug Bridge*

O Android Debug Bridge (ADB*)* é uma ferramenta do próprio SDK que permite a comunicação entre um computador e um dispositivo com Android. Ela assemelha-se ao (Secure Shell) SSH do Linux. Entre as várias utilidades desta ferramenta, destacam-se: instalação de aplicativos, execução de comandos diretamente no *shell* do dispositivo e cópia de arquivos entre o computador e o dispositivo e vice-versa. O ADB possui três componentes [10]:

a) um utilitário de linha de comando, que é executado pelo usuário para emitir os comandos;
b) um processo servidor, que executa no mesmo computador do usuário e é responsável por receber os comandos do utilitário e transmiti-los para o dispositivo;
c) um serviço, que executa como processo de segundo plano no dispositivo e é responsável por receber os comandos transmitidos pelo processo servidor.

O serviço do ADB no dispositivo fica ativo somente se a opção *Depuração USB* estiver habilitada. Portanto, se esta opção estiver desabilitada, não é possível utilizar o ADB para comunicar-se com o aparelho. A Tabela I lista alguns comandos do cliente ADB.

TABELA I. COMANDOS DO CLIENTE ADB

Comando	**Descrição**
adb shell	Inicia um *prompt* de comando no dispositivo.
adb push <local> <remoto>	Copia o arquivo especificado do computador local para o dispositivo.
adb pull <remoto> <local>	Copia o arquivo remoto especificado do dispositivo para o computador local.
adb install app.apk	Instala um aplicativo no aparelho.

Fonte: [10].

Caso o aparelho apreendido esteja com a opção de depuração USB desabilitada, o perito pode habilitá-la no menu de configurações de opções do desenvolvedor, conforme mostrado na Figura 1. Quando o aparelho possui a tela

bloqueada, primeiro é necessário desbloqueá-la para acessar o menu de configurações. Se o desbloqueio não for possível, não há como acessar tal tela.

Após habilitar a depuração USB, o perito deve conectar o cabo USB ao telefone e tentar se conectar ao aparelho usando o ADB. Caso obtenha sucesso, alguns aplicativos podem ser instalados ou até mesmo o comando adb *pull* pode ser usado para extrair os dados do aparelho para a máquina do examinador.

Em uma situação em que o telefone possua bloqueio de tela ativado, o perito ainda pode tentar conectar o cabo USB no aparelho para verificar se o usuário deixou a opção de depuração USB ativada.

Fig. 1. Ativação da depuração USB.

D. *Senhas, padrões e PIN Lock*

O Android fornece para o usuário diversas maneiras de configurar o bloqueio de tela. Algumas são básicas e úteis apenas para o travamento do teclado. Já outras são mais sofisticadas e permitem o desbloqueio da tela apenas para quem conhece a senha, padrão ou Personal Identification Number (PIN) [11].

O PIN é essencialmente numérico e não pode ser combinado com letras e outros caracteres. Também é possível bloquear a tela por um padrão, desenhando ligações entre pontos em uma matriz. Outra forma é usar uma senha tradicional, combinando letras e números. A Figura 2(a) exibe a tela de desbloqueio por PIN. A Figura 2(b) exibe a tela de desbloqueio por padrão. A Figura 2(c) exibe a tela de desbloqueio por senha.

(a) (b) (c)

Fig. 2. Bloqueios de tela.

Caso não seja possível realizar o desbloqueio da tela, somente chamadas de emergência podem ser feitas. Também não é possível acessar a tela de configuração do sistema. Existem alguns aplicativos que tentam realizar a quebra do bloqueio de tela. Porém, eles só podem ser instalados se a opção de depuração USB estiver habilitada.

E. *Partições típicas*

Uma partição é uma divisão lógica de um dispositivo de armazenamento de dados. Embora o fabricante do aparelho possa modificar o esquema de partições padrão, as partições mostradas na Tabela II estão presentes na maioria dos *smartphones* com Android. As partições de sistema, dados do usuário, *boot*, *cache* e *recovery* tipicamente estão presentes nos aparelhos [12]. Os aplicativos que armazenam dados do usuário na memória interna do telefone gravam estes dados na partição /data, no caminho /data/data/<app> [4].

TABELA II. TABELA DE PARTIÇÕES TÍPICAS DO ANDROID

Caminho	Nome	Sistema de Arquivos	Ponto de Montagem	Descrição
/dev/mtd/mtd0	pds	yaffs2	/config	Dados de configurações
/dev/mtd/mtdl	misc	-	N/A	Memória
/dev/mtd/mtd2	boot	Bootimg	N/A	Inicializável (partição padrão de *boot*)
/dev/mtd/mtd3	recovery	Bootimg	N/A	Inicializável (partição *recovery*)
/dev/mtd/mtd4	system	yaffs2	/system	Arquivos de sistemas e aplicativos
/dev/mtd/mtd5	cache	yaffs2	/cache	Arquivos de *cache*
/dev/mtd/mtd6	userdata	yaffs2	/data	Dados do usuário
/dev/mtd/mtd7	kpanic	-	N/A	Logs de falhas

Fonte: [12].

F. *Modos de inicialização*

Os aparelhos podem ser inicializados de diferentes modos. Alguns fabricantes disponibilizam *softwares* específicos para isto. Porém, na maioria dos dispositivos o modo de inicialização pode ser alterado através de uma combinação de teclas enquanto o dispositivo está sendo ligado.

Embora o Android forneça para os desenvolvedores de aplicativos certo nível de abstração de *hardware*, existe uma grande diversidade de fabricantes e modelos. Alguns aparelhos possuem grande quantidade de teclas. Já outros possuem apenas uma ou duas teclas. Justamente por causa desta diversidade, nem sempre a mesma combinação de teclas para alternar o modo de inicialização funciona em modelos diferentes. A Tabela III exibe as combinações de teclas para alternar o modo de inicialização de alguns modelos [12].

TABELA III. MODOS DE INICIALIZAÇÃO DE ALGUNS APARELHOS COM ANDROID

Modelo	Modo	Combinação	Descrição
Motorola Droid	*Flash*	D-Pad UP + power	Modo que permite *flashing* via *RSD Lite*
Motorola Droid	*Flash*	camera +power	Modo que permite *flashing* via *RSD Lite*
Motorola Droid	*Recovery*	power + x	*Boot* na partição *recovery* (após, camera + volup

			para mostrar menu)
HTC G1	*Flash*	power + back	Modo *Fastboot*
HTC G1	*Flash*	power + câmera	Modo de *boot* (‘*back*’ para trocar para *Fastboot*)
HTC G1	*Recovery*	power + home	*Boot* na partição *recovery*
Samsung Captivate	*Flash*	volup + voldn (então insira USB)	*Boot* no modo “Samsung *force download*”
Samsung Captivate	*Recovery*	power + volup + voldn	*Boot* na partição *recovery*
Samsung Galaxy Tab	*Flash*	power +voldn	*Boot* no modo “Samsung *force download*”
Samsung Galaxy Tab	*Recovery*	power + volup	*Boot* na partição *recovery*

Fonte: [12].

Quando o aparelho é ligado normalmente sem nenhuma combinação de teclas, ele está no modo normal. Neste modo, o sistema principal, comumente instalado na partição *system*, é iniciado. Para que o aparelho seja inicializado em um modo diferente, é necessário ligá-lo, pressionando a combinação de teclas corresponde ao modo desejado. Um modo especial de inicialização é chamado de modo de recuperação ou modo *recovery*. Ao ligar o aparelho, pressionando a combinação de teclas para inicialização no modo *recovery*, os arquivos da partição de mesmo nome, partição *recovery*, são carregados [13].

A inicialização no modo *recovery* permite ao usuário formatar o dispositivo, restaurar as configurações de fábrica, limpar dados de *cache* e realizar tarefas de manutenção. Esse modo de inicialização também é utilizado pelo próprio Android para aplicação de pacotes de atualização [4].

A partição *recovery* contém os arquivos de inicialização para o modo *recovery*. Ela possui seu próprio *kernel* Linux, separado do *kernel* do sistema principal do Android [14] e pode ser iniciada mesmo que a instalação principal do sistema esteja com problemas. O modo *recovery* padrão de fábrica normalmente oferece apenas funcionalidades básicas e sem acesso ao ADB [4].

Esta partição geralmente possui um tamanho pequeno e seu dispositivo associado pode ser diferente, dependendo do modelo e fabricante. Detalhes sobre esta partição podem ser vistos examinando o arquivo /proc/mtd [4], conforme mostra a Figura 3.

```
ahoog@ubuntu:~$ adb shell cat /proc/mtd
dev:    size   erasesize  name
mtd0: 00040000 00020000 "misc"
mtd1: 00500000 00020000 "recovery"
mtd2: 00280000 00020000 "boot"
mtd3: 04380000 00020000 "system"
mtd4: 04380000 00020000 "cache"
mtd5: 04ac0000 00020000 "userdata"
```

Fonte: [4].

Fig. 3. Detalhes da partição *recovery*.

G. *A extração de dados*

Existem vários métodos para extração de dados. Estes métodos são basicamente divididos entre extração física e extração lógica [15].

De acordo com [4], a extração física pode ser classificada em extração por *hardware* e por *software*. A extração por *hardware* é realizada utilizando duas técnicas conhecidas como *chip-off* e Joint Test Action Group (JTAG). Este tipo de extração só é útil quando os dados armazenados na memória *flash* não estão criptografados. Do contrário, os dados podem até ser extraídos, mas não serão adequadamente utilizados. Já a extração por *software* utiliza a técnica de executar *software* no aparelho, fornecendo uma imagem física completa das partições [4]. É a técnica normalmente empregada em diversas ferramentas de extração de dados.

As técnicas de extração lógica dos dados são menos destrutivas ao aparelho, pois não necessitam de alterações de *hardware* no dispositivo a ser analisado. Segundo [4], as técnicas lógicas de extração de dados apenas necessitam que a opção de depuração USB esteja habilitada.

A técnica conhecida como *ADB pull* utiliza o comando *pull* do ADB para realizar uma cópia recursiva dos diretórios e arquivos a serem analisados do aparelho para a máquina do perito. Apesar de simples, essa técnica nem sempre é viável, pois na maioria dos casos, o usuário sob o qual o *ADB* é executado não possui permissão de leitura nos diretórios dos aplicativos. A partição de maior interesse é a /data, onde residem todos os arquivos do usuário. Se o *ADB* possui acesso de *root*, esta partição pode ser inteiramente copiada.

O aplicativo AFLogical também pode ser utilizado para extração dos dados [9]. Este aplicativo foi desenvolvido pela empresa viaForensics e pode ser instalado no aparelho através do ADB. Uma vez instalado, ele extrai os dados de diversos aplicativos como SMS, contatos, registros de chamada, Facebook, *browser*, entre outros. Os dados extraídos são armazenados no cartão SD, em formato csv [16]

Outra maneira de analisar os dados é examinando o cartão SD externo. Diversos aplicativos permitem que o usuário realize um *backup* dos dados para a memória externa. O cartão SD pode ser removido do aparelho e analisado na máquina do perito. O problema é que nem sempre os dados do *backup* estarão atualizados.

Caso o bloqueio de tela esteja ativo, não é possível realizar *backup* dos aplicativos. Se a opção de depuração USB estiver desabilitada, também não é possível instalar aplicativos ou executar comandos. Neste cenário, ainda existe outra possibilidade: a substituição da partição *recovery*.

H. *A substituição da partição recovery*

A troca da partição *recovery* padrão pode ser realizada em aparelhos cujo *boot loader* seja compatível com o modo *fastboot* ou ofereça a opção de substituição de partições (também conhecida como *flash* de partições). Existem diversas partições *recovery* modificadas que podem ser utilizadas para substituição da partição padrão. A maioria destas partições permite acesso via ADB como *root*. Desta forma, o aparelho pode ser inicializado no modo *recovery* e o perito pode utilizar, por exemplo, a técnica de *ADB pull* para extração dos dados.

O *boot loader* é um pequeno programa responsável por carregar o sistema operacional. No ambiente Linux, os principais *boot loaders*, GRUB e LILO, estão presentes na maioria das distribuições. Em dispositivos com Android, o *boot loader* é responsável por carregar o sistema operacional Android ou a partição *recovery* [14]. Alguns fabricantes desenvolvem seus próprios b*oot loaders* e *softwares* para

interagir com eles. Uma das funções destes *softwares* é permitir a substituição de partições do aparelho [4]. Alguns exemplos destes *softwares* são o Motorola RSD Lite, o Samsung Odin Multiloader e o LG Flashtool.

O processo de substituição de partições é específico de cada aparelho. A imagem da partição a ser substituída deve ser compatível com o aparelho em questão e nem sempre a mesma imagem pode utilizada em modelos diferentes.

O *boot loader* pode estar travado ou não. Um *boot loader* travado somente carrega sistema operacional com assinatura válida. Da mesma forma, não é possível instalar uma imagem personalizada na partição *recovery*. Nestes casos, é necessário destravar o próprio *boot loader* primeiro. Esse procedimento varia de acordo com o aparelho e pode violar a garantia do fabricante [17].

I. *O modo fastboot*

O modo *fastboot* foi inicialmente implementado em um Android Developer Phone (ADP), fabricado pela HTC. Neste modo, é possível usar o utilitário de linha de comando *fastboot*, que já vem compilado com a SDK do Android, para gravação de imagens em partições do aparelho. Para utilizar o *fastboot*, é necessário que o *boot loader* do aparelho seja compatível com o modo *fastboot* e esteja destravado. Então o aparelho deve ser conectado na porta USB e ligado (ou reiniciado) segurando-se as teclas VOLDN e BACK. Essa combinação pode ser diferente dependendo do aparelho. Ao entrar neste modo, é mostrada na tela do aparelho a palavra *FASTBOOT* [4]. Neste momento, é possível emitir comandos para listar aparelhos conectados, como mostrado na Figura 4. Depois da confirmação de que o aparelho realmente está conectado, é possível fazer a gravação de novas partições.

```
C:\Users\Adriano>fastboot devices
0910D4D11800F00C        fastboot

C:\Users\Adriano>
```

Fig. 4. Comando *fastboot devices*.

Existem várias imagens da partição *recovery* modificadas que podem ser utilizadas em substituição à partição de fábrica. Na escolha de uma imagem apropriada, deve ser levado em consideração se a nova imagem permite ou não acesso como *root* via ADB. As mais populares são:

a) ClockworkMod: escrita por Koush Dutta, é baseada na imagem da partição *recovery* do Android 2.1. Possui diversas opções como *backup*, restauração, atualização do aparelho através de arquivos .zip e acesso via ADB habilitado [13];
b) TWRP: Team Win Recovery Project possui, além das opções padrão, funções como *backup* e restauração e acesso via ADB habilitado. Sua interface é sensível ao toque e é personalizável [18].

III. MATERIAIS E MÉTODOS

Para a realização dos experimentos deste trabalho, quatro modelos diferentes de *smartphones* foram utilizados. As especificações de cada aparelho foram descritas em cada experimento. Além disso, acessórios como cabos USB e carregadores compatíveis com cada modelo de aparelho foram necessários. Para a elaboração do método de extração de dados proposto, optou-se incialmente pela realização de uma ampla pesquisa bibliográfica sobre perícia forense em dispositivos móveis. A partir da fase inicial de pesquisa, foi possível encontrar estudos que tratavam especificamente do tema proposto por este trabalho e analisar diversas características do sistema operacional Android e elaborar um método. O método foi elaborado baseando-se nos princípios gerais da computação forense. Todavia, durante a pesquisa bibliográfica foi verificado que o processo de análise pericial em *smartphones* é sempre mais invasivo do que análises de computadores pessoais. Desta forma, o método proposto foi criado no intuído de preservar ao máximo a integridade dos dados, apesar de ser intrusivo.

Após a proposição do método, diversos experimentos foram realizados. Com estes experimentos foi possível constatar a aplicabilidade e viabilidade do método proposto. Também foi possível identificar cenários nos quais a aplicação do método é inviável, ora por questões de incompatibilidade, ora por questões de restrições do aparelho.

Esta seção é dedicada a descrever os experimentos realizados com o método proposto. Para tal foram utilizados quatro aparelhos de telefonia celular no seguinte cenário: sistema operacional Android, bloqueio de tela ativo com padrão, senha ou PIN de desbloqueio desconhecido e opção de depuração USB desabilitada.

A. *O método proposto*

A substituição da partição *recovery* padrão como método para extração dos dados do usuário é utilizada neste trabalho. A partição *recovery* original é substituída por outra que possibilite o acesso através do ADB, permitindo, desta forma, que o perito examinador seja capaz de realizar a extração dos dados do aparelho.

Contudo, a decisão de aplicação ou não deste método deve ser tomada pelo perito levando em consideração diversos aspectos, entre eles, a possibilidade da restauração das configurações de fábrica. A substituição da partição *recovery* pode causar incompatibilidade com o sistema Android instalado, levando-o a não inicializar novamente. Embora esta situação não prejudique a perícia em si e nem comprometa a integridade dos dados do usuário, esta possibilidade deve ser analisada pelo perito, pois, neste caso, a única maneira de deixar o celular utilizável novamente é restaurando a imagem original do Android para o aparelho, o que leva à perda de todos os aplicativos, históricos e configurações do usuário. Neste trabalho os experimentos foram realizados utilizando tanto a imagem ClockworkMod quanto a TWRP.

B. *Instalação do Software Development Kit (SDK)*

A instalação do SDK no sistema operacional Windows 7 pode ser feita baixando-se o arquivo de instalação direto do portal do desenvolvedor para Android (https://developer.android.com/sdk/). Após obter o arquivo de instalação é necessário executá-lo, aceitar termos de uso e confirmar os locais de instalação.

Um ponto chave da instalação do Android SDK é a escolha correta do nível da API (API *Level*). A cada alteração no *framework* de desenvolvimento são acrescentadas e removidas funções, suporte a novas plataformas, entre outras. Para solucionar problemas de compatibilidade de aplicativos, foi criado o conceito de nível de API. Uma determinada versão do Android suporta instalação de aplicativos criados

até certo nível de API. Aplicativos criados com níveis de API mais recentes não podem ser instalados em versões antigas do Android [19]. A Tabela IV relaciona a versão do Android ao nível de API suportado.

TABELA IV. VERSÕES DO ANDROID E SEUS NÍVEIS DE API

Versão	Nível da API	Codinome
Android 4.4	19	KITKAT
Android 4.3	18	JELLY_BEAN_MR2
Android 4.2.2 Android 4.2	17	JELLY_BEAN_MR1
Android 4.1.1 Android 4.1	16	JELLY_BEAN
Android 4.0.4 Android 4.0.3	15	ICE_CREAM_SANDWICH_MR1
Android 4.0.2 Android 4.0.1 Android 4.0	14	ICE_CREAM_SANDWICH
Android 3.2	13	HONEYCOMB_MR2
Android 3.1.x	12	HONEYCOMB_MR1
Android 3.0.x	11	HONEYCOMB
Android 2.3.4 Android 2.3.3	10	GINGERBREAD_MR1
Android 2.3.2 Android 2.3.1 Android 2.3	9	GINGERBREAD
Android 2.2.x	8	FROYO
Android 2.1.x	7	ECLAIR_MR1
Android 2.0.1	6	ECLAIR_0_1
Android 2.0	5	ECLAIR
Android 1.6	4	DONUT
Android 1.5	3	CUPCAKE
Android 1.1	2	BASE_1_1
Android 1.0	1	BASE

Fonte: [19].

Após a instalação, é possível verificar qual o nível da API está instalado usando o programa SDK Manager, instalado junto com o SDK. Nele também é possível acrescentar ou remover outros níveis de API, como ilustrado na Figura 5.

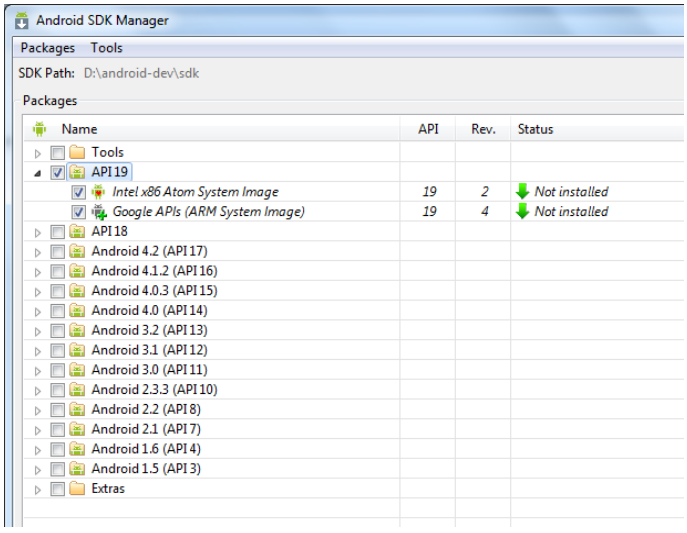


Fig. 5. Android SDK Manager.

C. *Procedimento de extração de dados do Samsung Galaxy S2 (GT-I9100)*

O GT-I9100, desenvolvido pela companhia sul coreana Samsung e lançado em abril de 2011 com o nome comercial Samsung Galaxy S II, possui processador Dual-Core Cortex A9 de 1.2 Ghz, *chipset* Exynos 4210, memória RAM de 1 GB e armazenamento interno de 16 GB ou 32 GB. Sua versão original vem com Android 2.3.4 (Gingerbread) que pode ser atualizada para a versão 4.1 (Jelly Bean).

O perito deve fazer uma avaliação inicial do aparelho para decidir quais procedimentos devem ser adotados, verificando a presença de cartão SD, cartão SIM e bateria. Para que seja possível realizar o acesso via ADB ao telefone, os *drivers* compatíveis com o modelo GT-I9100 devem ser instalados. No experimento realizado, os *drivers* foram obtidos do *site* da própria Samsung e instalados em um computador com Windows 7. Após a constatação de que a opção de depuração USB está desabilitada, o bloqueio de tela está ativo e o padrão, PIN ou senha de desbloqueio não é conhecido, o próximo passo é inicializar o celular no modo *recovery*.

A inicialização do GT-I9100 no modo *recovery* é feita pressionando simultaneamente as teclas de Volume (+), *Home* e *Power*, com o aparelho desligado. No experimento realizado, a tela do aparelho apresentou o modo *recovery* padrão, como mostrado na Figura 6.

Fig. 6. GT-I9100: tela do modo *recovery* padrão.

Em seguida, o cabo USB deve ser conectado ao aparelho e o comando adb *devices* executado, como ilustrado na Figura 7. Caso nenhum aparelho seja listado, a partição *recovery* presente não permite acesso ADB.

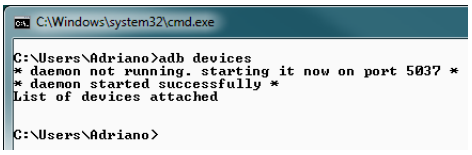


Fig. 7. GT:I9100: Comando `adb devices`.

Neste experimento, a substituição da partição *recovery* foi feita utilizando o *software* Odin3 [13]. Para usar este *software*, o celular deve estar no modo *download* (também conhecido como Odin *mode*). A inicialização no modo *download* é feita pressionando as teclas Volume (-), *Home* e *Power* com o aparelho desligado [20]. Ao ligar o aparelho segurando estas teclas, é mostrada uma tela de confirmação. Em seguida, a tecla de Volume (+) deve ser pressionada e o telefone entrará no modo *download*.

(a) (b)

Fig. 8. GT-I9100: Modo *download*.

A seguir, o celular deve ser conectado à USB e o *software* Odin3 v.1.85 inicializado. Ao iniciar, o Odin3 exibe a mensagem "Added", indicando que o aparelho foi reconhecido [13]. Caso a aparelho não tivesse sido reconhecido, os *drivers* deveriam ser reinstalados.

A versão compatível com o GT-I9100 do ClockworkMod utilizada neste experimento foi a 6.0.2.9. Para fazer a substituição, o arquivo do ClockworkMod deve ser selecionado na opção PDA do Odin3 e depois, o botão *Start* deve ser clicado. Quando o procedimento finaliza, o Odin3 exibe a mensagem "PASS", como pode ser observado na Figura 9.

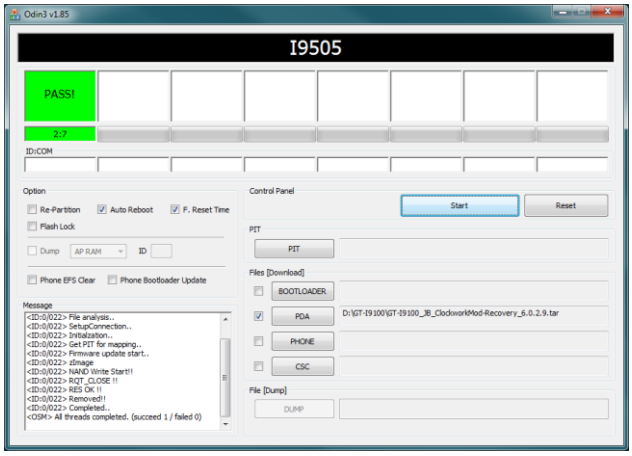


Fig. 9. GT-I9100: conclusão da substituição da partição *recovery*.

Após a conclusão deste procedimento, o aparelho deve ser imediatamente desconectado da USB e a bateria removida. Isto foi feito porque algumas versões do Android para aparelhos da marca Samsung, no momento da inicialização, restauram a partição *recovery* para a versão original caso esta partição tenha sido modificada. Em seguida, ele foi ligado novamente no modo *recovery*. Neste momento, a tela do ClockworkMod é apresentada indicando que o procedimento ocorreu com sucesso.

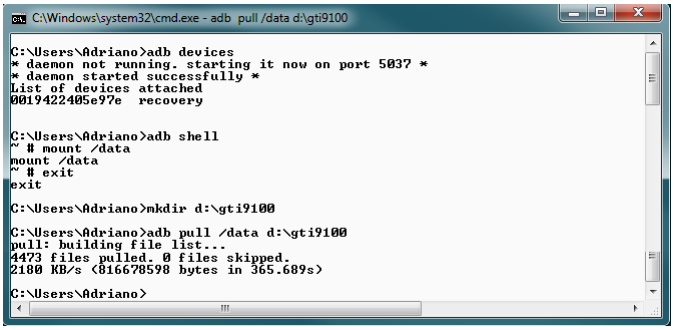


Fig. 10. GT-I9100: comando `adb pull /data`.

A próxima etapa é fazer a extração dos dados da partição /data. Para isto, é necessário acessar o aparelho via ADB e montar a partição com o comando *mount* /data. Em seguida, um diretório deve ser criado na máquina do examinador e os dados copiados através do comando adb *pull*. Esses comandos foram executados no GT-I9100 do experimento e o resultado mostrado na Figura 10.

D. Procedimento de extração de dados do LG Optimus 3D (P920h)

O P920h, denominado comercialmente de LG Optimus 3D, foi desenvolvido pela companhia sul coreana LG Eletronics em 2011. Sua principal característica é a tela que pode exibir imagens em terceira dimensão sem o uso de óculos especiais. Esse modelo vem com memória Random Access Memory (RAM) de 512 MB, processador ARM Cortex A9 de 1 GHz Dual Core, *chipset* OMAP4430 desenvolvido pela Texas Instruments, tela de 4.3 polegadas e Android 2.2 [21].

A inicialização do P920h no modo *recovery* é feita pressionando simultaneamente as teclas *Power*, Volume (-) e 3D, com o aparelho desligado. Assim que o aparelho for iniciado, a imagem da partição *recovery* é carregada.

No caso em tela, a substituição da partição *recovery* original do P920h foi necessária, pois esta partição não permite conexão via ADB. Essa substituição pode ser feita utilizando a ferramenta LGTool (http://www.lgtool.net/) ou utilizando o *fastboot*. Neste experimento, a substituição foi feita com o utilitário *fastboot*, pois a ferramenta LGTool é proprietária e necessita de ativação.

Embora o *boot loader* padrão do P920h forneça suporte, não existe uma combinação de teclas documentada para iniciar o telefone no modo *fastboot*. Para isto, foi utilizado o *software* Omap4Boot-for-optimus, desenvolvido pela comunidade XDA Developers e de código fonte aberto [22]. Este *software* foi criado para ser utilizado em procedimentos de recuperação de celulares com *chipset* OMAP44XX que não estejam inicializando, permitindo que o usuário consiga fazer o *boot* tanto no modo *download* como no modo *fasboot*. Neste trabalho, a ClockworkMod versão 6.0.1.9 compatível com o P920h foi utilizada.

Em seguida, o celular deve ser ligado no modo *fastboot,* sendo o *software* Omap4Boot-for-optimus utilizado nesse momento. Após a descompactação, o arquivo start_fastboot.bat deve ser executado. Então é apresentada uma tela solicitando qual o modelo do telefone. A opção 2 (Optimus 3D P920) deve ser selecionada. Agora, a mesma tela apresenta uma mensagem informando que o telefone deve ser conectado a USB, conforme ilustra a Figura 11.

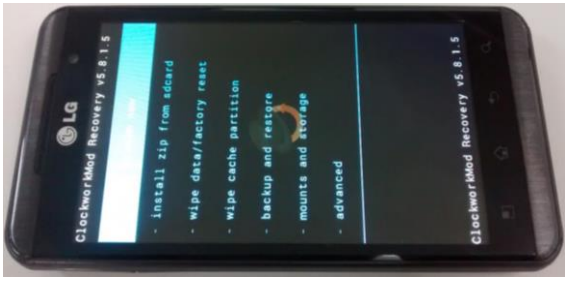


Fig. 11. P920h apresentando o *recovery mode* com ClockworkMod.

Nesse momento, o telefone desligado deve ser conectado à USB sem a bateria. Então, o *software* identifica o dispositivo OMAP4430, instala os *drivers* necessários e para em um segundo estágio. Só então a bateria deve ser acoplada novamente. Se o Windows não conseguir encontrar os *drivers* na pasta do Omap4Boot e instalar automaticamente, a instalação deve ser feita de forma manual e o procedimento repetido.

Feito isso, a tela do celular deve apresentar o logotipo da LG em tom de cinza e o texto "fastboot v0.5" no canto superior esquerdo. Esse mesmo procedimento pode ser feito para ligar o telefone no modo *download*, bastando segurar a tecla Volume(+).

A próxima etapa é a substituição da partição em si. Isso pode ser feito com o comando *fastboot flash recovery* recovery.img. Este comando substitui a partição *recovery* pela imagem recovery.img fornecida. Por fim, o telefone deve ser ligado novamente no modo *recovery*. Se todas as etapas foram concluídas com sucesso, a tela do aparelho deve apresentar a interface da ClockworkMod, como ilustra a Figura 12.

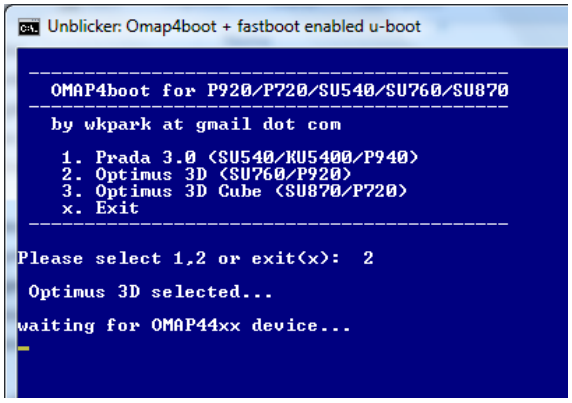


Fig. 12. Software OMAP4Boot aguardando conexão com o P920h

O método de extração de dados utilizado neste experimento também foi o adb *pull*, de forma similar ao aparelho GT-I9100.

E. Procedimento de extração de dados Motorola Moto G Dual SIM (XT1033)

O XT1033 foi desenvolvido pela Motorola e lançado no mercado em janeiro de 2014. Possui processador Quad-core Cortex-A7 de 1.2 Ghz, *chipset* Qualcomm MSM8226 Snapdragon 400, memória RAM de 1GB e armazenamento interno de 8 GB ou 16 GB. Sua versão original vem com Android 4.3 (Jelly Bean).

O XT1033 não aceita cartões de memória externos. Logo, todos os dados do aparelho são armazenados na memória interna. Então, na análise inicial do aparelho o perito deve observar a presenta de cartões SIM. No aparelho deste experimento, dois SIMs foram encontrados, pois este aparelho possui a tecnologia Dual SIM.

A partição *recovery* do XT1033 pode ser substituída através do *fastboot*. Porém, esta substituição só pode ser feita em aparelhos cujo *boot loader* esteja destravado. Caso o *boot loader* esteja travado, é necessário destravá-lo primeiro. Esse procedimento pode ser feito através de um código obtido no *site* da própria Motorola [23]. Para obter este código, é necessário digitar os dados de destravamento do telefone no *site* da Motorola. Estes dados são conseguidos através do próprio *fastboot*.

F. Procedimento de extração de dados Motorola Moto G 2ª Geração (XT1069)

O XT1069, também conhecido como Motorola Moto G2, foi desenvolvido pela Motorola e lançado no mercado em setembro de 2014. Possui processador e *chipset* Qualcomm Snapdragon 400 MSM8226 / ARM Cortex-A7 de 1.2 Ghz, memória RAM de 1GB e armazenamento interno de 16 GB. Sua versão original vem com Android 4.4.4 (KitKat), atualizável para Android 5.0 (Lollipop). Todas as considerações anteriores feitas ao XT1033 em relação à substituição da partição recovery também se aplicam ao XT1069.

IV. RESULTADOS E DISCUSSÃO

A. Resultados para o Samsung Galaxy S2 (GT-I9100)

Ao final deste experimento, os dados da partição /data e os dados do cartão SD interno (partição /sdcard) foram armazenados em diretórios da máquina do examinador, sendo a extração efetivada utilizando o adb *pull*.

No experimento realizado, após a extração dos dados, o aparelho foi ligado segurando-se somente a tecla *Power*. Durante a inicialização, o logotipo da Samsung foi mostrado na tela, juntamente com um ícone de advertência. O aparelho ficou travado nesta tela por cerca de 10 segundos. Após esse período, ele desligou. A tentativa de inicialização normal do aparelho foi repetida por mais três vezes, sem sucesso.

Desta forma, é possível concluir que em aparelhos do modelo GT-I9100, a substituição da partição *recovery* pode afetar também o sistema operacional Android, levando o aparelho a não inicializar normalmente. No caso do celular deste experimento, a imagem original do sistema Android foi restaurada usando o próprio Odin3. Após a restauração, o celular iniciou normalmente, porém com as configurações originais de fábrica. Todos os aplicativos instalados, contatos, históricos e mensagens se perderam.

Esta é uma hipótese que deve ser levada em consideração na decisão do perito em fazer ou não a substituição da partição *recovery*. Embora isto não prejudique a análise pericial em si, uma vez que os dados já tinham sido copiados para a máquina, após o procedimento de substituição, o celular pode se tornar inoperantese o sistema Android instalado for incompatível com a nova partição *recovery*. E a maneira encontrada neste trabalho para deixar o celular utilizável novamente envolve a restauração da imagem original do Android, o que leva à perda dos dados do aparelho.

B. Resultados para o LG Optimus 3D (P920h)

Conforme apresentado na seção III.D, a substituição da partição *recovery* foi realizada, o que permitiu a inicialização do aparelho no modo *recovery* com o ClockworkMod, e a subsequente extração de dados via adb *pull*, conectando-se ao aparelho através do comando adb *shell*. A partir daí, todos os arquivos da partição de dados puderam ser copiados para a máquina do examinador.

C. *Resultados para o Motorola Moto G Dual SIM (XT1033)*

No experimento realizado com o XT1033, o *boot loader* estava travado. Para fazer esta verificação, o aparelho foi ligado no modo *fastboot*, pressionando simultaneamente as teclas Volume (-) e *Power* com o aparelho desligado, por 3 segundos. Assim que estas teclas foram liberadas, a tela do modo *fastboot* foi mostrada. Nela foi possível verificar o texto "*Device is LOCKED*", conforme mostrado na Figura 13.

Fig. 13. XT1033: tela do modo fastboot.

Neste momento o telefone estava no modo *fastboot* mostrando a informação de que o *boot loader* estava travado. Então, o cabo USB foi conectado ao telefone e ligado ao computador e o comando *fastboot oem get_unlock_data* executado (Figura 14). O retorno da execução deste comando é o código de verificação a ser passado para o site da Motorola.

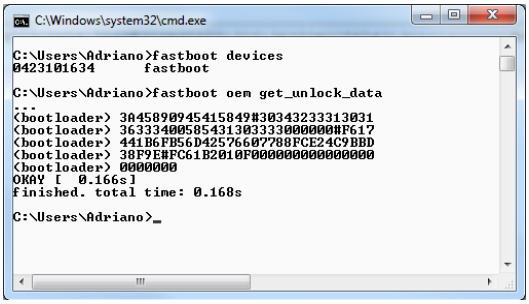


Fig. 14. XT1033: execução do comando para obter unlock_data.

O código de verificação é tratado de forma a não conter espaços ou informações adicionais. Ao entrar com o código no *site* da Motorola, a mensagem "*Your device does not qualify for bootloader unlocking.*" foi mostrada, conforme ilustra a Figura 15. Desta forma, conclui-se que este aparelho não pode ter seu *boot loader* destravado.

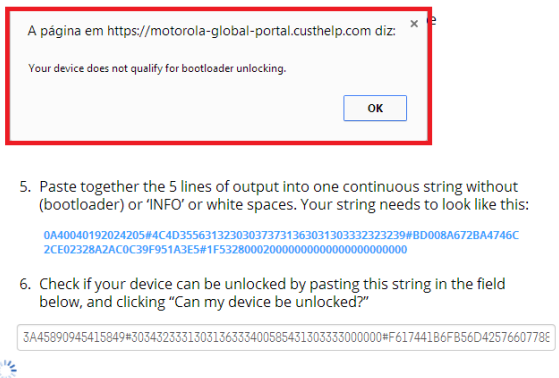


Fig. 15. XT1033: mensagem indicando dispositivo *unlockable*.

Sem o código de destravamento, não foi possível destravar o *boot loader*. Logo, também não foi possível fazer a substituição da partição *recovery* do XT1033 neste experimento. Isto, porém, não invalida a aplicação deste método em modelos cujo *boot loader* esteja destravado.

D. *Resultados para o Motorola Moto G 2ª Geração (XT1069)*

Assim como ocorreu com o XT1033, o *boot loader* estava travado. Procedeu-se, então, à tentativa de destravamento, através da execução do comando *fastboot oem get_unlock_data* e a obtenção do código de verificação. Desta vez, a entrada do código de verificação no *site* da Motorola gerou uma mensagem eletrônica enviada ao usuário do aparelho, com o código de destravamento do *bootloader*, conforme pode ser observado parcialmente na Figura 16.

Fig. 16. Mensagem eletrônica retornando o código de destravamento.

O destravamento é efetivado executando o comando *fastboot oem unlock* UNIQUE_KEY, sendo UNIQUE_KEY o código retornado na mensagem de correio eletrônico. Uma vez destravado, a substituição da partição em si pôde ser realizada também utilizando o comando *fastboot flash recovery* recovery.img. O arquivo imagem compatível com o XT1069 do TWRP utilizado neste experimento foi a twrp-2.8.6.0-titan.img. Por fim, o telefone deve ser ligado novamente no modo *recovery*. Após a conclusão das etapas com sucesso, a tela do aparelho apresentou a interface da TWRP, como ilustra a Figura 17. A opção de backup permite a realização do backup do aparelho, que pode ser copiado para o computador através do comando adb *pull*. Entretanto, o destravamento do bootloader acarretou o retorno do aparelho às configurações originais de fábrica, implicando na perda das informações anteriormente presentes.

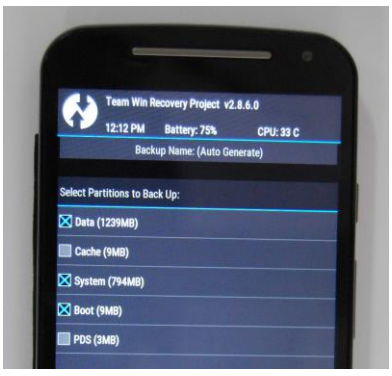


Fig. 17. XT1069: tela de *backup* do modo *recovery* TWRP.

V. CONCLUSÕES

Este trabalho teve como principal objetivo propor, analisar, e demonstrar a viabilidade de se utilizar um método para o problema de extração de dados de telefones *smartphones* com sistema operacional Android, bloqueio de tela ativo e senha, PIN ou padrão de desbloqueio desconhecido e função de depuração USB desabilitada.

A hipótese de trabalho pressupõe que a substituição da partição *recovery* padrão por outra que possibilite o acesso via ADB possibilita a extração dos dados do usuário, sem comprometer a integridade das informações extraídas.

Para verificação da hipótese proposta, quatro experimentos foram realizados. O primeiro experimento foi feito com um Samsung Galaxy S2. Neste modelo, foi possível realizar a extração dos dados. Entretanto, a substituição da partição *recovery* afetou a instalação do sistema principal, levando a aparelho a não inicializar novamente. O segundo experimento apresentado foi relativo a um aparelho LG Optimus 3D. A partição *recovery* padrão foi substituída e os dados extraídos com sucesso. O terceiro experimento apresentado foi feito com um aparelho Morotola Moto G Dual SIM. Neste aparelho, não foi possível realizar a substituição da partição *recovery*, uma vez que o *boot loader* estava travado e não foi possível destravá-lo. Por fim, o quarto experimento, também com um aparelho Motorola com o *boot loader* travado, embora tenha permitido a substituição da partição *recovery*, não garantiu a extração com sucesso dos dados, uma vez que foram apagados no processo de destravamento do *boot loader*.

Com base nos experimentos realizados neste trabalho, é possível concluir que a substituição da partição *recovery*, como método para extração de dados de *smartphones*, pode ser realizada em aparelhos que possuem o *boot loader* destravado. Nos aparelhos em que o método pode ser aplicado, os dados foram extraídos com sucesso. Contudo, em um deles a instalação principal do Android foi prejudicada e o aparelho não iniciou novamente. Embora isto não tenha comprometido a extração e a integridade dos dados em si, esta é uma possibilidade que deve ser considerada antes de aplicar este método.

Como proposta de trabalho futuro, pretende-se analisar as situações em que a manipulação do aparelho acaba por submetê-lo a um *reset* de fábrica, o que não implica, necessariamente, na remoção de dados presentes [24]. A análise do aparelho com Android nestas condições poderia ainda permitir a recuperação de dados pessoais importantes para a perícia.

REFERÊNCIAS

[1] INTERNATIONAL TELECOMMUNICATION UNION. Disponível em: <http://www.itu.int/en/ITU-D/Statistics/Pages/stat/default.aspx>. Acesso em: 06 mai. 2015.

[2] VECCHIA, E. D. Perícia Digital – Da Investiação à Análise Forense. Millenium Editora, 2014.

[3] MAHAPATRA, L. *Tech / Sci. International Business Times*. 2013. Disponível em: <http://www.ibtimes.com/android-vs-ios-whats-most-popular-mobile-operating-system-your-country-1464892>. Acesso em: 26 nov. 2013.

[4] HOOG, Andrew. *Android Forensics - Investigation, Analysis and Mobile Security for Google Android*. Waltham: Elsevier, 2011.

[5] SIMÃO, André Morum de L. *Proposta de método para Análise Pericial em Smartphone com Sistema Operacional Android*. Dissertação. (Mestrado)- Departamento de Engenharia Elétrica, Universidade de Brasília. Brasília, DF, 2011.

[6] ELEUTÉRIO, P. M. S. e MACHADO, M. P. Desvendando a Computação Forense. Novatec, 2011.

[7] CASEY, Eoghan. *Digital Evidence and Computer Crime*. Waltham: Elsevier, 2011.

[8] HASEMAN, C. *Android Essentials*. New York: Apress, 2008.

[9] GUPTA, Aditya. *Learning Pentesting for Android Devices*. Packt Publishing, 2014.

[10] ANDROID DEVELOPERS. *Android Debug Bridge*. Disponível em: <http://developer.android.com/tools/help/adb.html>. Acesso em: 13 jan. 2014a.

[11] SUN, Chen, WANG, Yang and ZHENG, Jun. Dissecting Pattern Unlock: The effect of pattern strength meter on patterin selection. *Journal of Information Security and Applications*, 19, 308-320, 2014.

[12] VIDAS, Timothy; ZHANG, Chengye; CHRISTIN, Nicolas. Toward a general collection methodology for Android devices. *Digital Investigation: The International Journal of Digital Forensics & Incident Response*, 8, p. S14-S24, ago. 2011.

[13] TYLER, J. and VERDUZCO, W. *XDA Developers' AndroidTM Hacker's Toolkit*. John Wiley and Sons, 2012.

[14] XDA DEVELOPERS. *Recovery*. Disponível em: <http://forum.xda-developers.com/wiki/Recovery>. Acesso em: 01 mar. 2014a.

[15] SON, Namheun; LEE, Yunho; KIM, Dohyun; JAMES, Joshua; LEE, Sangjin; LEE, Kyungho. A study of user data integrity during acquisition of Android devices. *Digital Investigation: The International Journal of Digital Forensics & Incident Response*, 10, p. S3-S11, ago., 2013.

[16] VIAFORENSICS. *AFLogical tool*. Disponível em: <https://viaforensics.com/resources/tools/android-forensics-tool/>. Acesso em: 03 mar. 2014.

[17] XDA DEVELOPERS. *Boot Loader*. Disponível em: < http://forum.xda-developers.com/wiki/Bootloader>. Acesso em: 03 mar. 2014b

[18] TEAM WIN. *Team Win Recovery Project*. Disponível em: <http://teamw.in/project/twrp2>. Acesso em: 03 mar. 2014.

[19] ANDROID DEVELOPERS. *What is API Level?* Disponível em: <http://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>. Acesso em: 01 abr. 2014b.

[20] SAMSUNG ELECTRONICS. *Service manual for GSM telephone GT-I9100*. 2011. 103 p.

[21] LG ELETRONICS. LG-P920h. Manual do usuário, 2011.

[22] GITHUB – OMAP4BOOT. *Tools to boot omap4xx over USB*. Disponível em: <https://github.com/swetland/omap4boot>. Acesso em: 18 abr. 2014.

[23] MOTOROLA BOOTLOADER UNLOCK. *Unlock your Bootloader*. Disponível em: <https://motorola-global-portal.custhelp.com/app/standalone/bootloader/unlock-your-device-a/action/auth >. Acesso em: 21 de abr. 2014.

[24] SCHWAMM, Riqui and ROWE, Neil. Effects of the Factory Reset on Mobile Devices. *The Journal of Digital Forensics, Security and Law*, 9(2), 205-220, 2014.

DOI: 10.5769/C2015006 or http://dx.doi.org/10.5769/C2015006

Método de recuperação de mensagens apagadas do *SQLite* no contexto do aplicativo *WhatsApp* para plataforma *Android*

Alberto Magno M. Soares, João Paulo C. de Sousa, Juliano K. M. Oya

Resumo. **Discutimos um método de recuperação de mensagens apagadas do aplicativo *WhatsApp*, cujo armazenamento se dá em base de dados utilizando sistema gerenciador de banco de dados *SQLite*, disponível na plataforma *Android*. No desenvolvimento da técnica, foram analisadas as estruturas internas do arquivo da base de dados *SQLite* com potencial interesse para uma investigação forense dessa natureza. Especificamente, foram exploradas as regiões não alocadas (*freespace*) e desalocadas (*freeblocks*) das bases de dados *SQLite*, com foco na recuperação estruturada de mensagens do aplicativo *WhatsApp*.**

Palavras-chave: *Análise Forense, Android, Recuperação de dados, WhatsApp, SQLite, Freeblock, Freespace*

Abstract. **We discuss a recovery method for WhatsApp application's deleted messages, which occurs in stored database using database management system SQLite, available on the Android platform. In the technique's development, was examined the internal structures of the SQLite database file with potential interest for a forensic investigation of this nature. Specifically, the unallocated (freespace) and deallocated (freeblocks) regions of SQLite databases were explored, focusing on structured recovery of WhatsApp application messages.**

Keywords: *Forensics Analysis, Android, Data Recovery, WhatsApp, SQLite, Freeblock, Freespace*

I. Introdução

WhatsApp é uma aplicação de mensagens instantâneas com versões disponíveis para *smartphones* com os sistemas *Android*, *BlackBerry*, *iPhone*, *Windows Phone* e *Symbian*, e até janeiro de 2015 atingiu a marca de 700 milhões de usuários [1].

A plataforma *Android* já contabilizou, segundo Gartner [2], mais de um bilhão de usuários em 2014, liderando o mercado de sistemas operacionais [2].

Em 2014, os exames de informática em aparelhos de telefonia celular que utilizam *WhatsApp* sobre plataforma *Android* representaram grande parte dos exames periciais em dispositivos móveis que foram requisitados ao Instituto de Criminalística da Polícia Civil do Distrito Federal e, em muitos casos, o examinador necessitou pesquisar a existência de vestígios de mensagens apagadas ou fragmentos destas no dispositivo.

O método que será apresentado permite recuperação de mensagens apagadas do aplicativo *WhatsApp*, cujo armazenamento se dá em base de dados utilizando sistema gerenciador de banco de dados *SQLite*, disponível na plataforma *Android* [3].

O presente trabalho inicia com a análise de características do aplicativo *WhatsApp* e do sistema gerenciador de banco de dados *SQLite*. Em seguida, são apresentados detalhes de interesse forense, incluindo análise das estruturas internas da base de dados envolvidas no processo de apagamento e recuperação de mensagens. Ao final, é detalhado o algoritmo de recuperação de mensagens, construído com base no método apresentado.

II. Materiais e Métodos

Para o estudo, foi utilizado ambiente *Android* emulado (*Android SDK for x86, Android 5.1 (Lollipop), Kernel 3.4.67+digit@tyrion.par.corp.google.com#3, Build sdk_phone_x86-eng 5.1 LKY45 1737576*), possibilitando acesso irrestrito ao sistema de arquivos da memória interna.

Para construção da base *SQLite* de exame, foi ativada conta no aplicativo WhatsApp de testes no dispositivo emulado, e foram enviadas mensagens no formato 'Mensagem #', com cenários de deleção individual ou de várias mensagens simultâneas.

A análise da estrutura interna dos arquivos SQLite foi realizada mediante uso do software *FTK Imager*, versão 3.2.0.0 [4].

Detalhamento do método de recuperação desenvolvido está descrito nos itens IV e V.

III. Referencial Teórico

Nesta seção são apresentados os conceitos relacionados ao aplicativo *WhatsApp* e ao banco de dados *SQLite*.

A. Banco de dados SQLite

O *SQLite* é uma biblioteca de *software* de código aberto que implementa um sistema de banco de dados SQL transacional, sem a necessidade de um servidor dedicado e com pouca ou nenhuma configuração para seu funcionamento. Além disso, é um sistema autossuficiente e extremamente compacto – *SQLite* lê e escreve diretamente para arquivos de disco comuns e não possui um processo tipo servidor separado – e, dessa forma, é uma escolha bastante popular para o uso em dispositivos móveis como celulares. Um banco de dados *SQLite* completo contendo tabelas, índices, gatilhos e visões, subsiste em um único arquivo em disco [5] [6].

Sobre sua arquitetura, os componentes do *SQLite* são agrupados em categorias denominadas *Core*, *SQL Compiler*, *Backend* e *Acessories*. Especificamente na categoria *Backend*, há os componentes diretamente relacionados com a estrutura de armazenamento de dados, que são as árvores-b (*b-tree*) e as

páginas (*Page*) [5]. A Figura 1 apresenta uma visão geral desses componentes e seus principais relacionamentos.

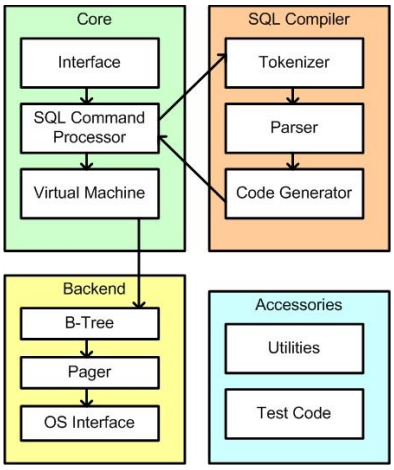


Fig. 1. Visão geral da arquitetura do sistema gerenciador de banco de dados *SQLite*.
Fonte: adaptado de [5]

Uma visão geral da estrutura de um arquivo de banco de dados do *SQLite* pode ser vista na Figura 2. Assim, um arquivo de banco de dados é dividido em várias unidades de armazenamento, chamadas páginas. As páginas são numeradas sequencialmente – iniciando em 1 – e possuem o mesmo tamanho, que pode ser entre 512 (2^9) e 65.536 (2^{16}) *bytes*.

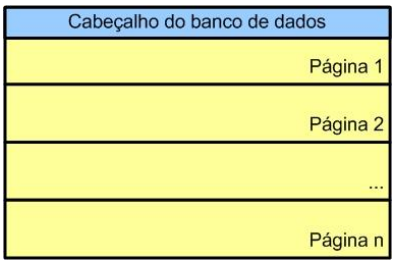


Fig. 2. Visão geral da estrutura de um arquivo de banco de dados.

No início da primeira página é armazenado o cabeçalho do banco de dados (uma sequência de 100 *bytes*), cujos primeiros 16 *bytes* são a assinatura `SQLite format 3`, indicando que se trata de um banco de dados *SQLite*. Ainda no cabeçalho pode ser extraída a informação do tamanho das páginas (*offset* 16, tamanho de 2 *bytes* os quais devem ser interpretados como um inteiro de 16 *bits* no formato *big endian*). Essa mesma página é raiz de uma *b-tree* que contém uma tabela especial denominada `sqlite_master` que armazena o esquema completo do banco de dados.

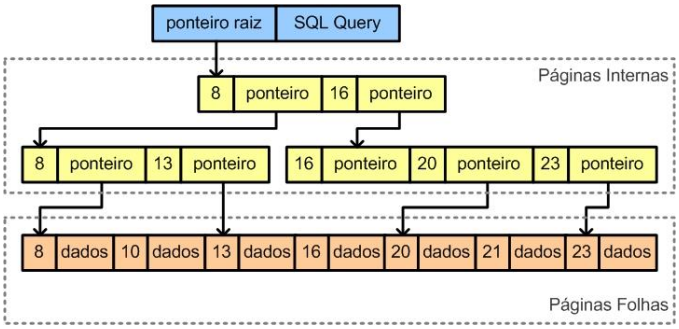


Fig. 3. Exemplo de uma estrutura de árvore balanceada.
Fonte: adaptado de [6]

Uma grande parte do arquivo do banco de dados *SQLite* é organizada em uma ou mais estruturas *b-tree*. Uma única estrutura *b-tree* é armazenada usando uma ou mais páginas, e cada página contém um único nó da *b-tree*. As páginas usadas para armazenar uma única estrutura de *b-tree* não precisam formar um bloco contíguo [5]. Essa estrutura é exemplificada na Figura 3.

Ainda na Figura 3, podemos observar que as páginas podem ser de tipos diferentes, de acordo com o tipo de informação que ela armazena. Dentre os tipos documentados em [5], as páginas folhas da *b-tree* são as de maior interesse pericial, já que nelas são armazenados os dados dos registros.

Uma página folha pode ser identificada pelo primeiro campo do seu cabeçalho, que são 2 *bytes* com o valor de *flag* `0x0D`. Logo após o cabeçalho da página, fica armazenada uma lista de ponteiros (inteiros de 2 *bytes* no formato *big endian*) cujos valores são os *offsets* para cada célula de dados da página. Na Figura 4 é exemplificada a estrutura de uma página.

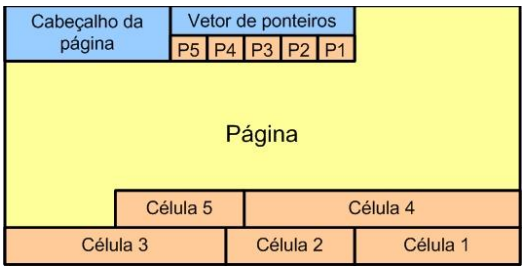


Fig. 4. Visão geral da estrutura de uma página do *SQLite*.

Assim, dentro de uma página, a célula é a unidade elementar de armazenamento de dados, sendo, portanto, a unidade de armazenamento dos registros do banco de dados. Cada célula possui uma estrutura exemplificada nas Figuras 5 e 6. No cabeçalho da célula estão os dados do tamanho da célula, excluindo o seu cabeçalho, e o campo *row id*, cujo formato é no padrão *varint* (*variable length integers*) [5].

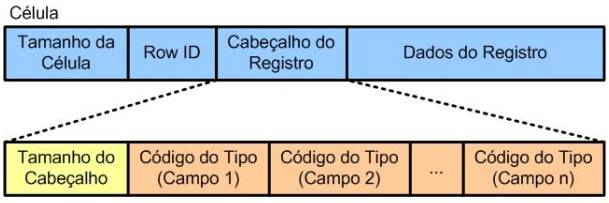


Fig. 5. Estrutura de uma célula e do cabeçalho do *payload*.

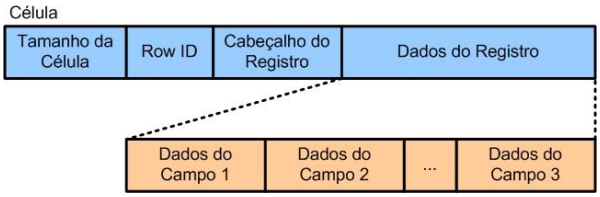


Fig. 6. Estrutura de uma célula e do *payload*.

Ainda na Figura 5, no cabeçalho do registro é armazenado o tamanho do cabeçalho (no formato *varint*) e, em seguida, o tipo de cada campo, onde cada *byte* representa o tipo de dado armazenado na área de dados, como mostrado na Figura 6. O valor do *byte* e o tipo de dado que ele representa é mostrado na Tabela I.

TABELA I. DESCRIÇÃO DOS CÓDIGOS DO TIPO DOS DADOS DO REGISTRO.

Valor do Tipo	Descri□□o	Tamanho
0	0	NULL
1 a 4	Inteiro com sinal	1 a 4
5	Inteiro com sinal	6
6	Inteiro com sinal	8
7	Ponto flutuante	8
8	Constante de valor 0	0
9	Constante de valor 1	0
N ≥ 12 e par	BLOB	(N-12)/2
N ≥ 13 e par	*String*	(N-13)/2

B. Aplicativo WhatApp

O *WhatsApp* é um aplicativo de mensagens multiplataforma que permite trocar mensagens através do celular e da *Internet*. Tal aplicativo está disponível em diversas plataformas de comunicação móvel e seu uso é bastante popular em telefones celulares do tipo *smartphone*.

Em investigações criminais, tal aplicativo é bastante utilizado como meio de comunicação pelos criminosos e o conteúdo das mensagens pode ser indício da materialidade do crime, da possível autoria e do modo de operação de uma associação criminosa. Versões atuais do aplicativo, na plataforma *Android*, usam o *SQLite*, embarcado na plataforma *Android*, como sistema gerenciador de banco de dados.

Dentre várias tabelas de armazenamento do aplicativo, há a tabela `messages`, onde são armazenadas as mensagens enviadas e recebidas pelo usuário do aplicativo [7]. Para ilustrar, alguns dos campos dessa tabela foram extraídos de uma mensagem real e interpretados, conforme a descrição dos tipos da Tabela I, e o resultado é apresentado na Tabela II.

TABELA II. DESCRIÇÃO DOS CAMPOS DA TABELA MESSAGES.

Nome	C□digo	Tipo	Tamanho
key_remote_jid	0x43	*String*	27
key_id	0x25	*String*	12
status	0x01	Inteiro	1
data	0x21	*String*	10
timestamp	0x05	Inteiro	6
send_timestamp	0x01	Inteiro	1
received_timestamp	0x05	Inteiro	6

Na Figura 7 é apresentado o registro com dados dessa mensagem. Considerando os valores de tamanhos obtidos na Tabela II, é possível extrair o conteúdo de cada campo da `messages`.

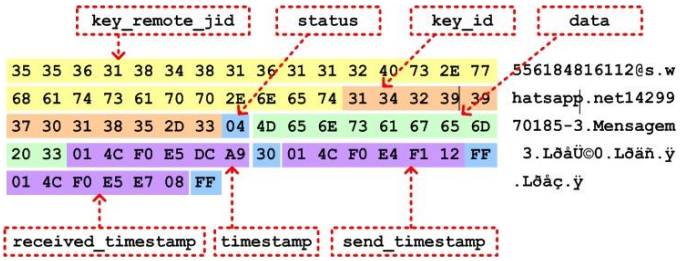


Fig. 7. Exemplo de interpretação da estrutura de dados de um registro de mensagem do aplicativo *WhatsApp*.

IV. RECUPERAÇÃO DE MENSAGENS

A. Recuperação de dados na região desalocada (freeblock) de páginas folha.

No cenário de recuperação de dados, deve-se primeiramente compreender o funcionamento das páginas folha, visto que estas páginas são as responsáveis por armazenar o conteúdo dos registros em um banco de dados *SQLite*. Uma página folha, assim como qualquer outra página, é estruturada em células. Cada página folha possui um cabeçalho composto de 8 *bytes*, que contém as informações descritas na Tabela III.

O cabeçalho é seguido por uma lista de *offsets* de 2 *bytes* que apontam para a área na página onde cada célula está posicionada. Caso haja "n" células na página, existirão "n" ponteiros de 2 *bytes* para os conteúdos das células, que estarão ordenados pelo valor das chaves dos registros. As células são alocadas do endereço final para o endereço inicial da página, conforme ilustrado na Figura 8.

TABELA III. CAMPOS DO CABEÇALHO DE UMA PÁGINA FOLHA

Offset/Tamanho	Descri□□o
0 / 1	Um *byte* indicando o tipo de p□gina. O valor 13 (0x0D) □ usado para indicar que a p□gina □do tipo folha.
1 / 2	Quando um registro □ exclu□do, a c□lula correspondente ao registro torna-se desalocada (*freeblock*), liberando mais espa□o na p□gina para futuro uso. Os dois *bytes* deste campo indicam o *offset* para o in□cio do primeiro *freeblock*. O valor 0 indica que n□o h□ regiı es desalocadas. Cabe ressaltar que o valor □0□ n□o indica que a p□gina est□ cheia, mas sim que n□o h□ c□lulas removidas.
3 / 2	Dois *bytes* indicando o n□mero de c□lulas da p□gina.
5 / 2	Dois *bytes* indicando o *offset* do in□cio da □rea com conte□do de c□lulas.
7 / 1	Um *byte* indicando o n□mero de fragmentos na p□gina. O *freeblock* requer pelo menos 4 *bytes* de espa□o. Grupos de 1,2 ou 3 *bytes* isolados compı em os fragmentos.

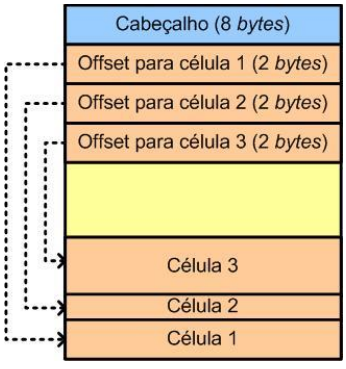


Fig. 8. Organização de uma página folha.

Quando um registro do banco de dados é excluído, algumas alterações são realizadas na estrutura da página folha. A célula com o conteúdo é desalocada, liberando espaço na página para inserção de novos registros. Entretanto, para que o espaço da célula seja liberado, tanto o cabeçalho da página, quanto os

offsets para os conteúdos das células devem ser reajustados. Quando a célula é excluída, o campo do cabeçalho da página que registra o número de células deve ser decrementado. Com a liberação de espaço, o *offset* para o início do primeiro *freeblock* deve ser ajustado. Assim, quando uma nova célula precisar ser alocada, a região referente ao *freeblock* poderá ser utilizada. Além disso, os *offsets* para as células são organizados sequencialmente, devendo, também, ser reajustados. A Figura 9 ilustra a remoção de uma célula:

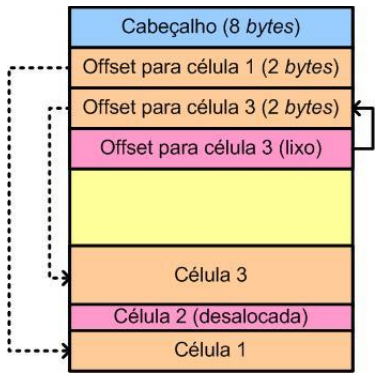


Fig. 9. Organização de uma página folha, após a remoção de uma célula.

Para a recuperação de conversas do aplicativo *WhatsApp*, as regiões desalocadas das páginas possuem grande utilidade forense. Será mostrado, adiante, como é possível recuperar mensagens dessas regiões, realizando-se uma análise do banco de dados em nível de *bytes*.

Em sistemas *Android*, o banco de dados *SQLite* que armazena as conversas é denominado `msgstore.db`. O referido banco tem em sua estrutura interna a tabela `messages` que é responsável por armazenar, além de outras informações, o conteúdo das conversas. No exemplo mostrado na Figura 10, foram enviadas três mensagens entre dois usuários de teste do aplicativo *WhatsApp*, sendo uma delas removida logo após o envio das três mensagens.

Fig. 10. Troca de mensagens através do aplicativo *WhatsApp*.

Realizando uma consulta (SQL *query*) na tabela `messages`, é possível observar que, antes da exclusão da "mensagem 2", havia, na tabela, quatro registros. O primeiro registro é criado quando o *WhatsApp* inicia uma conversa e não possui dados sobre as mensagens. Os demais registros estão associados às mensagens e possuem informações relevantes. Após a remoção da "mensagem 2", um dos registros é apagado, reduzindo as entradas na tabela `messages`. Tal processo pode ser visualizado na Figura 11.

Fig. 11. Consulta no *SQLite* mostrando diferenças no banco antes e após a exclusão de uma mensagem.

Analisando a estrutura interna do banco, é possível observar que os dados são armazenados em células no interior de uma página folha. Antes da "mensagem 2" ser excluída, a página que armazena os dados possui quatro células e, como o banco nunca tinha sido utilizado anteriormente, toda a região vazia está preenchida com zeros. Não existe também nenhuma informação relativa a *freeblocks*, ou seja, o campo no cabeçalho da página associado a este tipo de informação está zerado. A Figura 12 ilustra a estrutura da página antes da exclusão da "mensagem 2". Ressalte-se que os *offsets* mostrados são relativos ao início da página.

Após a exclusão da mensagem, é possível observar uma série de mudanças na estrutura da página. Primeiramente, a célula que armazena o registro apagado é liberada para uso futuro, tornando-se parte dos *freeblocks*. A região que armazenava a célula excluída passa a ter grande relevância forense, uma vez que os dados são desalocados, porém a informação continua integra até que uma nova mensagem seja incluída e faça uso do *freeblock*. Desta forma, é possível criar mecanismos de buscas nestas regiões visando recuperar conversas do aplicativo *WhatsApp*.

Para tornar a página livre, algumas alterações devem ser feitas no cabeçalho da página. Primeiramente, o número de células é reduzido, implicando na alteração do campo do cabeçalho que guarda no número de células. Como a célula é liberada, o *offset* para o início dos *freeblocks* é alterado. Neste caso específico, o *offset* para o *freeblock* passa a ter o antigo valor do *offset* para a célula excluída. Os primeiros *bytes* da célula excluída também são alterados para manutenção da estrutura de ponteiros dos *freeblocks*. Outros campos também podem ser alterados à medida que são realizadas operações na página, podendo, inclusive, gerar fragmentação interna. Os *offsets* para as páginas também precisam ser rearranjados. A Figura 13 mostra o conteúdo da página após a remoção da "mensagem 2".

Como é possível observar, com exceção dos primeiros *bytes* que são alterados para manter a estrutura de ponteiros do *freeblock*, a região desalocada continua mantendo os dados da conversa, que pode ser recuperada por uma análise forense. Estas informações podem ser de grande relevância em uma investigação e não são visualizadas por meio das ferramentas de navegação SQLite convencionais.

B. Recuperação de dados na região não alocada (freespace) de páginas folha.

Na arquitetura do banco *SQLite*, as páginas folha estão constantemente sofrendo alterações. Em algumas situações, todas as células da página podem ser liberadas, tornando a página livre para novas inserções de registros. Como visto anteriormente, o conteúdo das células, mesmo após a exclusão de uma mensagem, permanece acessível e recuperável. Contudo, à medida que novos registros vão sendo inseridos, a região desalocada vai sendo ocupada, destruindo a informação das conversas mais antigas.

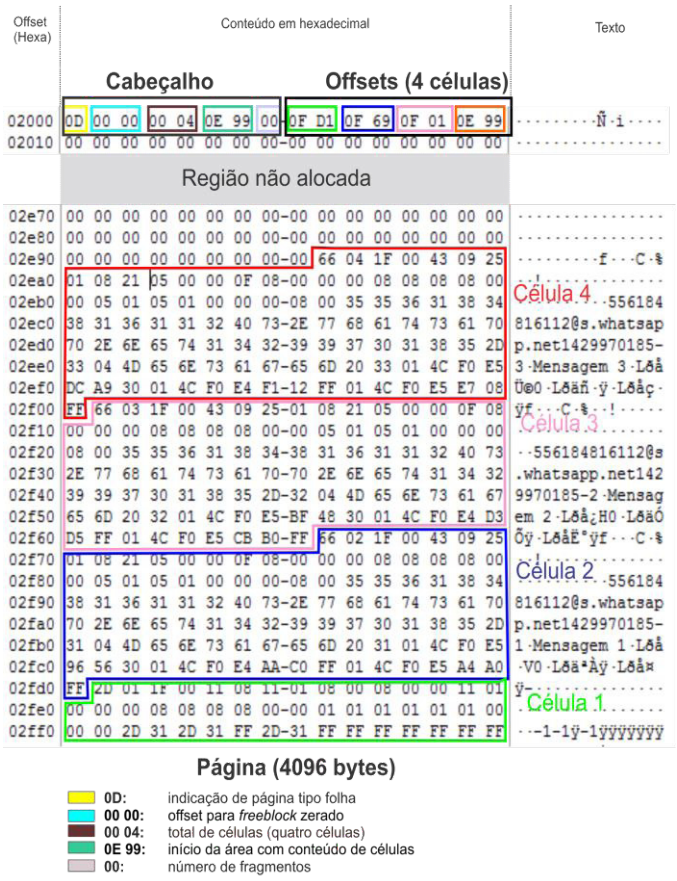


Fig. 12. Estrutura interna da página folha antes da exclusão da "mensagem 2". Conteúdo gerado pelo software *FTK Imager* [4] e posteriormente adaptado.

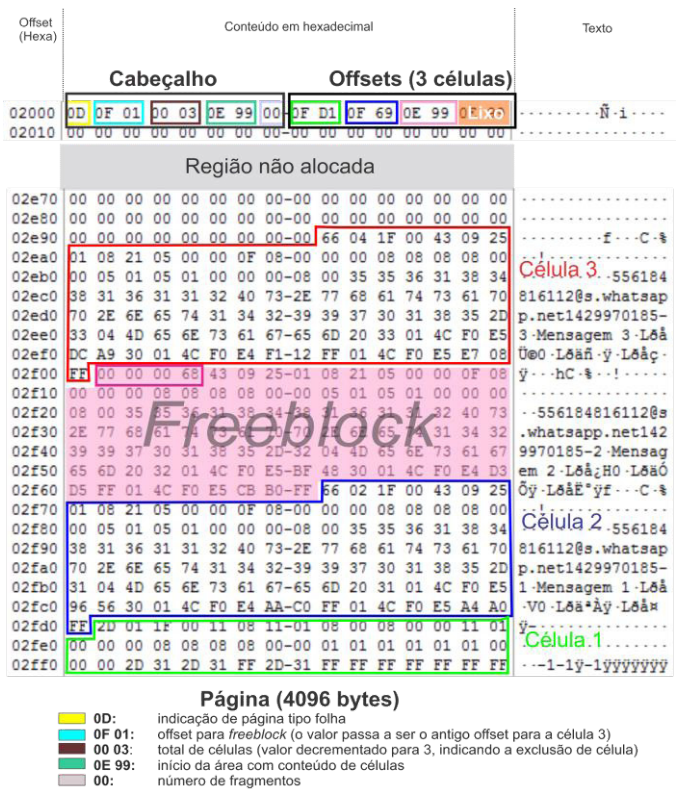


Fig. 13. Estrutura interna da página folha após exclusão da "mensagem 2". Conteúdo gerado pelo software *FTK Imager* [4] e posteriormente adaptado.

No cenário ilustrado na Figura 14, foram enviadas diversas mensagens ao usuário "Teste WhatsApp 2". Após o envio, todas as conversas foram excluídas e uma nova mensagem foi enviada ao usuário "Teste WhatsApp 3".

Fig. 14. Chats com os usuários "Teste WhatsApp 2" e "Teste WhatsApp 3".

Neste cenário, mesmo com a exclusão do chat com o usuário "Teste WhatsApp 2" e liberação das células, o banco irá manter as conversas recuperáveis. Todavia, quando a mensagem do usuário "Teste WhatsApp3" é incluída na página folha, parte dos dados são perdidos, tornando a recuperação das mensagens e fragmentos de mensagens mais complexa. A Figura 15 ilustra, em nível de *bytes*, como a inclusão da nova mensagem altera o conteúdo da região não alocada.

C. Outras possibilidades de recuperação de mensagens.

De acordo com o observado nos exames periciais cotidianos, um usuário do aplicativo *WhatsApp*, geralmente, realiza poucas ações de exclusão de mensagens, e quando o faz, é comum a exclusão de poucas linhas de conversa, com isso, grande parte dessas ações atuam em regiões analisadas de páginas de dados desalocadas e livres.

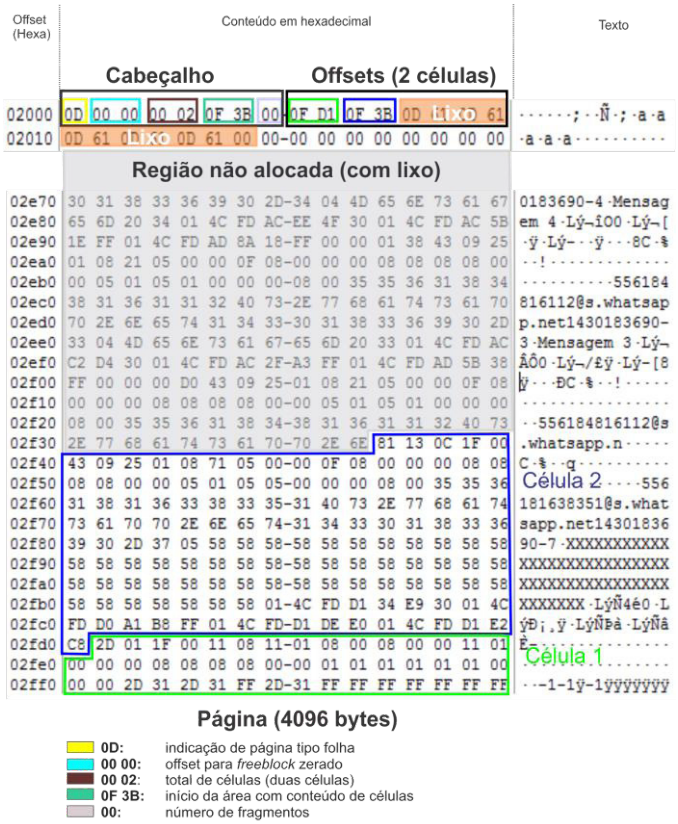


Fig. 15. Estrutura interna da página folha após exclusão do chat com o usuário "Teste WhatsApp 2" e do envio de mensagem para o usuário "Teste WhatsApp 3". Conteúdo gerado pelo software *FTK Imager* [4] e posteriormente adaptado.

Contudo, a especificação do formato *SQLite* define como deve ser feito o armazenamento de registros extensos que extrapolem o espaço disponível da página de dados. Para isso, é indicado ao final da célula, um ponteiro de 4 *bytes*, para página de *overflow*, onde é armazenado o conteúdo excedente.

Outra possibilidade de recuperação é quando há exclusão de páginas inteiras com mensagens, que resultariam no descarte de toda página, e no registro dessa como página desalocada. Tais páginas são armazenadas em uma lista de páginas livres (*freelist page*) e podem conter informação de relevância pericial.

O *SQLite* conta ainda com sistema de *rollback journal*, mantido em arquivo de log auxiliar, que pode conter mensagens recuperáveis não gravadas na base dados devido a interrupções inesperadas do funcionamento da *engine*. Existe mecanismo semelhante de operação de log, que pode ser ativado opcionalmente a partir da versão 3.6.0 do *SQLite*, denominado *Write-Ahead-Log*, conforme [5], onde há a possibilidade de recuperação de mensagens não gravadas em definitivo na base de dados, mas na versão do aplicativo *WhatsApp* analisado e em versões anteriores, não foi detectado ativação desse recurso como parte do mecanismo de manutenção de consistência.

Por abranger grande parte do foco usual dos exames executados, as regiões tratadas no método abordado restringiram-se inicialmente às regiões no interior das páginas de dados, mas, para um exame mais aprofundado, pretende-se estender o método para todas as estruturas do arquivo da base de dados e do sistema jornalístico cujas estruturas possam conter mensagens excluídas recuperáveis.

V. Algoritmo de Recuperação

Nesta seção é apresentado algoritmo de recuperação de mensagens apagadas do aplicativo *WhatsApp*, utilizando os conceitos apresentados.

A. Acesso a base de dados SQLite

O aplicativo *WhatsApp* para plataforma *Android*, na sua versão 2.12.58, armazena os seus arquivos de banco de dados de mensagens no diretório, de acesso restrito, `/data/data/com.whatsapp/databases`, da memória interna, onde se encontra o arquivo da base de dados *SQLite* denominado `msgstore.db`, objeto de análise. Existe mecanismo de *backup* do próprio aplicativo que realiza cópia criptografada do arquivo de mensagens para mídia de armazenamento removível do dispositivo, sendo que a chave criptográfica simétrica está armazenada no arquivo denominado `key`, localizado no diretório, de acesso restrito, `/data/data/com.whatsapp/files`, dificultando a extração.

Existem técnicas de extração do arquivo da base de dados através da exploração de vulnerabilidades da aplicação ou do próprio sistema operacional. A partir das últimas versões do aplicativo, há um aumento dos mecanismos de segurança, dentre eles a restrição com relação à extração de dados do aplicativo através do mecanismo de *backup* do sistema *Android*, muitas vezes utilizado em análise forense para ganho de acesso a diretórios restritos, o que exige do examinador o desenvolvimento contínuo de novas técnicas de extração.

B. Processo de recuperação

O método de recuperação de mensagens *WhatsApp* que é apresentado adota estratégia baseada na recuperação de páginas de dados, tipo folha, da estrutura b-*tree*, e inspeção das áreas de *freespace* (área não alocada) e *freeblock* (região desalocada) dessas páginas, em busca de células marcadas como apagadas que possibilitem recuperação das mensagens.

Para o processo de recuperação de mensagens, foi desenvolvido programa, na linguagem JAVA, para percorrer os dados do arquivo da base de dados, conforme algoritmo descrito na Figura 16.

Essencialmente, o algoritmo desenvolvido realiza leitura do tamanho das páginas do arquivo, inspeciona a área desalocada e levanta informações do esquema de dados da tabela `messages`. Iterativamente percorre todas as páginas de dados examinando o espaço não-alocado (*freespace*) e os blocos desalocados (*freeblocks*) e, em cada uma dessas áreas, faz análise léxica dos *bytes* no formato definido no esquema, localizando a posição da sequência de caracteres *@s.whatsapp*, característica do campo denominado `key_remote_jid` que deve conter identificação do interlocutor nos registros pertencentes à tabela de mensagens. Esta técnica é semelhante à utilizada em [8], para recuperação de registros da tabela `moz_places` da base de dados *SQLite* do *browser Firefox*.

Como exemplo de recuperação na área não-alocada (*freespace*) execução do algoritmo, inicialmente, foi analisado cenário da base de dados com exclusão simultânea de várias mensagens de uma conversa. O programa identificou mensagens em área não alocada, conforme descrito na Figura 17, indicando que aquelas mensagens foram apagadas e que a página foi restruturada pelo *engine SQLite*. Essa restruturação demonstra o aumento do tamanho da área não alocada, englobando a região das mensagens apagadas. Na Figura 18 pode-se observar registros da mensagem recuperada.

No cenário para recuperação em área desalocada (*freeblock*), foi recuperada mensagem fruto de uma deleção individual, conforme saída do processamento descrita na Figura 19.

Fig. 16. Processo de recuperação de mensagens.

```
Tamanho da Página do banco SQLite:4096
->Varrendo Free SPACE de 3887 byte(s)
  localizado no offset:8204
Buscando célula de mensagem WhatsApp ...
Offset: 8204
Tamanho:3887
<início>
?a?a?a?a?a? .... <bytes omitidos>
... ??????????????????????????????pC?%??!??????????????
????????556184816112@s.whatsapp.net1430183690-6?Mensagem
6?L??1?0?L??????L???8????C?%??!??????????????????????5
56184816112@s.whatsapp.net1430183690-5?Mensagem
5?L????0?L??????L?????????C?%??!??????????????????????5
56184816112@s.whatsapp.net1430183690-4?Mensagem
4?L???O0?L??[???L????????8C?%??!??????????????????????5
56184816112@s.whatsapp.net1430183690-3?Mensagem
3?L????0?L??/???L??[8?????C?%??!??????????????????????
556184816112@s.whatsapp.n
<fim>
```

Fig. 17. Trecho de saída do programa de recuperação que contem trecho da área não alocada (*freespace*).

```
1 Campo:id-Tipo:0-Tamanho:0-Conteúdo:0
2 Campo:key_remote_jid-Tipo:67-Tamanho:27-
  Conte□do:556184816112@s.whatsapp.net
3 Campo:key_from_me-Tipo:9-Tamanho:0-Conteúdo:1
4 Campo:key_id-Tipo:37-Tamanho:12-Conteúdo:1430183690-6
5 Campo:status-Tipo:1-Tamanho:1-Conteúdo:4
6 Campo:needs_push-Tipo:8-Tamanho:0-Conteúdo:0
7 Campo:data-Tipo:33-Tamanho:10-Conte□do:Mensagem 6
8 Campo:timestamp-Tipo:5-Tamanho:6-
  Conte□do:1430185259494(27/04/2015 22:40:59)
9 Campo:media_url-Tipo:0-Tamanho:0-Conteúdo:0
  ...
```

Fig. 18. Exemplo de trecho de mensagem recuperada da área não-alocada (*freespace*).

```
Varrendo Free BLOCK de 3723 byte(s)
  localizado no offset:8206
Buscando celula de mensagem WhatsApp ...
Offset: 12033
Tamanho:104
<início>
C?%??!??????????????????????556184816112@s.whatsapp.net
1429970185-2?Mensagem 2?L??H0?L??????L????f???
<fim>
<início - em HEX>
430925010821050000F0800000008080808000005010501000000080
0035353631383438313631313240732E776861747361706702E6E6574
3134323939373031383352D32044D656E736167656D2032014CF0E5BF
4830014CF0E4D3D5FF014CF0E5CBB0FF66021F00
<fim>
 1 Campo:id-Tipo:0-Tamanho:0-Conteúdo:0
 2 Campo:key_remote_jid-Tipo:67-Tamanho:27-
   Conte□do:556184816112@s.whatsapp.net
 3 Campo:key_from_me-Tipo:9-Tamanho:0-Conteúdo:1
 4 Campo:key_id-Tipo:37-Tamanho:12-Conteúdo:1429970185-2
 5 Campo:status-Tipo:1-Tamanho:1-Conteúdo:4
 6 Campo:needs_push-Tipo:8-Tamanho:0-Conteúdo:0
 7 Campo:data-Tipo:33-Tamanho:10-Conte□do:Mensagem 2
 8 Campo:timestamp-Tipo:5-Tamanho:6-
   Conte□do:1429970730824(25/04/2015 11:05:30)
 9 Campo:media_url-Tipo:0-Tamanho:0-Conteúdo:0
10 Campo:media_mime_type-Tipo:0-Tamanho:0-Conteúdo:0
  ...
```

Fig. 19. Exemplo de trecho de mensagem recuperada da área desalocada (*freeblock*).

Para mensagens muito longas, cujo conteúdo extrapola o espaço especificado na célula, são definidos 4 *bytes* no final da área destinada à célula que contém o endereço da página de *overflow* com o restante da mensagem que pode ser recuperada.

Observou-se também a possibilidade de recuperação de fragmentos de mensagens em áreas residuais, tipo *slack*, das áreas desalocadas (*freeblock*); porém, essas áreas são apresentadas pelo algoritmo para inspeção direta, pois devido a inconsistências detectadas nos dados, não há como ser efetuada leitura de acordo com a definição dos tipos definidos.

VI. Conclusão e Trabalhos Futuros

A recuperação de mensagens ou de fragmentos de mensagens excluídas do aplicativo *WhatsApp* pode ser de grande importância forense.

Neste artigo, foi apresentado um método desenvolvido para recuperação de mensagens apagadas do aplicativo *WhatsApp* , versão 2.12.58, sobre a plataforma *Android*, baseado na inspeção das estruturas internas do arquivo de base de dados *SQLite*, na localização de regiões livres ou desalocadas e na análise dos dados dessas áreas, permitindo a extração de dados e a potencial recomposição de mensagens excluídas.

Observou-se que é possível recuperar as mensagens apagadas das regiões referidas, inclusive com a possibilidade de os dados estarem íntegros.

Como trabalhos futuros, existem as possibilidades de recuperação de mensagens com grande extensão (que extrapolam o conteúdo de uma página de dados), de recuperação de páginas livres (*freepages*) e de recuperação de mensagens do arquivo do sistema de *journaling*.

Além disso, como trabalhos futuros, também podem ser produzidas abordagens para recuperação de mensagens excluídas em diferentes versões do aplicativo *WhatsApp*, bem como diferentes sistemas operacionais que façam uso de bases de dados *SQLite*.

Referências

[1] PORTAL G1, "WhatsApp atinge os 700 milhões de usuários por mês em todo o mundo", Disponível em <http://g1.globo.com/tecnologia/noticia/2015/01/whatsapp-atinge-os-700-milhoes-de-usuarios-por-mes-em-todo-o-mundo.html>. Acesso em 09 de maio de 2015.

[2] STAMFORD, Conn, "Android to Surpass One Billion Users Across all Devices in 2014", 2014, Disponível em <http://www.gartner.com/newsroom/id/2645115>. Acesso em 30 de maio de 2015.

[3] KOMATINENI,Satya; MACLEAN,Dave, "Pro Android 4", 2012, APRESS, ISBN:978-1-4302-3931-4, http://www.it-ebooks.info/go.php?id=657-1393515887-0abbc5d732ddac7ff07e9f6a175c8667.

[4] ACCESS DATA CORPORATION. FTK imager. Disponível em, <http://www.accessdata.com/support/product-downloads>; Acesso em 30 de maio de 2015.

[5] SQLite. Disponível em <http://www.sqlite.org>. Acesso em 27 de abril de 2015.

[6] JEON, Sangjun, et al. "A recovery method of deleted record for SQLite database. Personal and Ubiquitous Computing 16.6",2012,p 707-715.

[7] ANGLANO, Cosimo, "Forensic analysis of WhatsApp Messenger on Android smartphones", 2014,Elsevier, Digital Investigation nº11,p.201-213 http://www.journals.elsevier.com/digital-investigation.

[8] PEREIRA, Murilo Tito, "Forensic analysis of the Firefox 3 Internet history and recovery of deleted SQLite records", 2009,Elsevier, Digital Investigation nº5,p.93-103 http://www.journals.elsevier.com/digital-investigation.

Peritos Criminais Alberto Magno M. Soares, João Paulo C. de Sousa, Juliano K. M. Oya
Seção de Perícias em Informática, Instituto de Criminalística – Polícia Civil do Distrito Federal, Brasília-DF, Brasil
E-mails: alberto.magno@gmail.com, jpclaudino@gmail.com, juliano.oya@gmail.com

DOI: 10.5769/C2015007 or http://dx.doi.org/10.5769/C2015007

Investigação em Ambientes de Jogo Multijogadores *Online*

Juliano K. M. Oya, Cleber Scoralick Junior e Bruno W. P. Hoelz

Resumo—**Jogos *online* por meio da *Internet* podem ser utilizados para auxiliar ou cometer crimes como exploração sexual de crianças e adolescentes, racismo, injúria, difamação, calúnia e associação criminosa. Além disso, crimes praticados fora do ambiente *online*, como furtos, podem apresentar vestígios inesperados nesse ambiente. Neste trabalho, realizou-se um estudo exploratório do ambiente de jogos *online* e definiu-se uma metodologia para a investigação de crimes nesse mesmo ambiente. A metodologia proposta, baseada na análise de tráfego de redes e na identificação de endereços IP, foi aplicada em um estudo de caso real, no qual foi possível estabelecer com sucesso a autoria do crime.**

Palavras-Chave—crimes cibernéticos, análise de tráfego de redes, jogos online.

Abstract—**Online games over the Internet have been increasingly used to aid or commit crimes such as child exploitation, racism, slander, defamation, and crime association. In addition, crimes committed offline, such as theft, can present unexpected traces in the online world. In this work, we carried out an exploratory study of an online gaming environment and set up a methodology for the investigation of crimes in such environments. The proposed methodology, based on network traffic analysis and identification of IP addresses, was applied in a real case study, in which it was possible to successfully establish the authorship of the crime.**

Keywords—cyber crime, network traffic analysis, online games.

I. INTRODUÇÃO

A análise de tráfego de redes é uma atividade que consiste no uso de *hardware* e *software* para a coleta e análise do tráfego de mensagens de protocolos de rede, como o TCP, UDP e IP. É, geralmente, realizada por administradores de redes para detectar anomalias na rede; encontrar pontos de bloqueios na rede; descobrir equipamentos e cabeamentos defeituosos; observar importantes mensagens não mostradas pelas aplicações; detectar falhas de segurança, instalação ou *bugs* em serviços disponíveis na rede; e descobrir o tráfego "pirata" dentro da rede [7].

Por meio da análise de tráfego de redes é possível investigar os ambientes virtuais de jogos eletrônicos, também conhecidos como jogos multijogadores *online*. Nesse tipo de ambiente a interação entre os usuários (ou jogadores) ocorre, em nível de aplicação, por meio da troca de mensagens e comandos. Cada usuário, dentro desse mundo virtual, é representado por um *nickname* ou um *avatar*, os quais podem ter pouca relação com os verdadeiros nomes reais de cada usuário.

Ambientes virtuais baseados na *Internet* são cada vez mais utilizados para auxiliar ou cometer crimes [3] como exploração sexual de crianças e adolescentes, racismo, injúria, difamação, calúnia, formação de quadrilha, entre outros. Além disso, crimes praticados fora do ambiente *online*, como furtos, podem apresentar vestígios inesperados nesse ambiente.

Apesar do anonimato criado pelo uso de *nickname* ou *avatar*, é possível identificar o usuário por meio da análise das mensagens dos protocolos TCP, UDP e IP. Cada interação entre os usuários de um ambiente virtual gera um conjunto de segmentos TCP e UDP que são encapsulados em pacotes IP [1].

Neste artigo, é proposto um método de trabalho para a coleta e análise de dados de tráfego de rede com o objetivo de identificar possíveis autores de um delito por meio do endereço IP. Assim, serão apresentadas as atividades que constituem o método de trabalho, quais sejam: preparação do ambiente de coleta, coleta de dados de tráfego de redes, análise dos dados coletados e finalmente a preparação de um relatório para a autoridade competente.

Este trabalho é divido em três seções:

a) Referencial teórico, na qual são apresentados os conceitos relacionados aos ambientes de jogos multijogadores *online*, os vestígios digitais e a identificação de autoria por meio de endereços IP;

b) Metodologia, na qual são descritos os instrumentos e procedimentos e é apresentado o estudo de caso no qual o método de investigação é aplicado;

c) Conclusão, na qual são discutidos os resultados obtidos e os possíveis trabalhos futuros..

II. REFERENCIAL TEÓRICO

Este tópico apresenta os fundamentos relacionados às técnicas, ferramentas e conceitos envolvidos na investigação em ambientes de jogo multijogadores *online*. A *Seção A* descreve os ambientes de jogo multijogadores *online*. A *Seção B* descreve o procedimento para a identificação de autoria por meio de endereços IP.

A. Ambientes de jogo multijogadores online

Mcshaffry e Graham [6] apresentam um modelo lógico da arquitetura geralmente utilizada pelos sistemas de jogos. Na Figura 1, são apresentados os principais módulos e interfaces que compõe esse modelo.

Como os mesmos autores explicam, a camada de aplicação se preocupa com a máquina na qual o jogo é executado. Nesta

camada é onde estará localizado o código que realiza a comunicação com dispositivos de *hardware* (como o *mouse*, o teclado e o monitor), serviços do sistema operacional (tais como as comunicações de rede) e operações como a inicialização e desligamento do jogo.

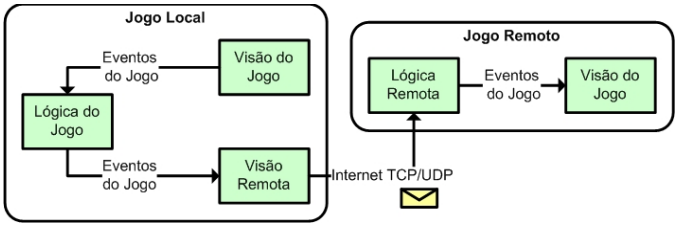


Fig. 1. Visão geral de uma arquitetura lógica de um jogo.
Fonte: Adaptado de Mcshaffry e Graham [6].

Na camada de lógica, ainda segundo Mcshaffry e Graham [6], é possível encontrar os subsistemas de gestão de estado do jogo, que é responsável por comunicar as mudanças de estado para outros sistemas, assim como aceitar comandos de entrada de outros sistemas. A camada de visão do jogo é responsável por apresentar o estado do jogo e traduzir entrada em comandos de jogo que são, então, enviados para a lógica do jogo. Por fim, a camada de visão remota é responsável por enviar, em jogos que utilizam a rede de computadores, dados de sincronização entre o jogo local e o jogo remoto.

Esse modelo de arquitetura é utilizado na grande maioria dos sistemas de jogos, assim como para os modelos de interação entre os jogadores e o sistema de jogo apresentados por Fullerton [5], os quais são: *single player*; *player vs. player*; *multilateral competition*; *team competition*; *multilateral team competition*; *unilateral competition*; *multiple individual vs. game*; *multiple players compete against the system*; *cooperative*.

Os jogos do tipo *single player* ou *player vs. player* não utilizam, na maioria das vezes, a conexão com a *Internet* durante a execução do jogo. Já os jogos do tipo *multiplayer* utilizam, na maioria das vezes, a conexão com a *Internet* para que os jogadores interajam entre si.

A respeito do ambiente de conexão, McShaffry e Graham [6] apresentam 5 modelos tipicamente utilizados pelos jogos atuais, que são: jogo individual sem conexão, jogo individual com conexão, jogo multijogador com conexão direta, jogo multijogador com conexão rede local, jogo multijogador com conexão *Internet*. A Figura 2 mostra graficamente a forma de organização e interconexão dos equipamentos que compõem esses modelos, sendo que em alguns casos pode haver a conexão com a *Internet* para possibilitar a troca de dados entre os jogadores. Nessa Figura, em 2-A é apresentado o modelo jogo individual sem conexão. Em 2-B são apresentados os modelos jogo multijogador com conexão direta e jogo multijogador com conexão rede local. Em 2-C é apresentado jogo individual com conexão *Internet*. Por fim, na Figura 2-D, é apresentada uma arquitetura híbrida de comunicação, na qual os jogadores se comunicam entre si e também com um servidor central de jogo.

A técnica de análise de tráfego de rede pode ser aplicada para os modelos de conexão que envolvam múltiplos jogadores por meio de uma conexão com a rede local ou com a *Internet*, como mostrado na Tabela I. Nela são relacionados os modelos apresentados por McShaffry e Graham [6] e a possibilidade de se realizar a captura do tráfego de dados na rede.

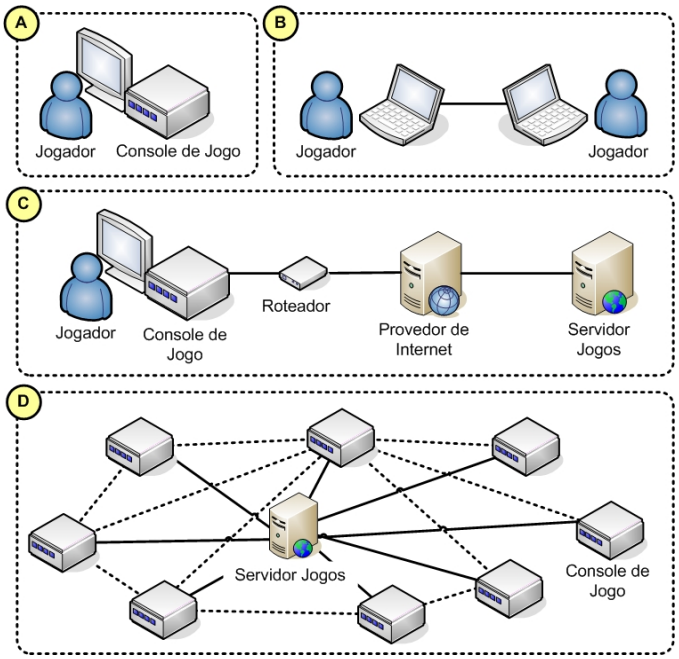


Fig. 2. Modelos de conexão dos sistemas de jogos
Fonte: Adaptado de McShaffry e Graham [6].

TABELA I. PRINCIPAIS CARACTERÍSTICAS DOS MODELOS DE CONEXÃO.

	Interação dos jogadores	Conexão com Internet	Provedor de Conexão	Servidor de Jogos	Análise de Tráfego Local	Análise de Tráfego Remota
Jogo Individual s/ Conexão	✗	✗	✗	✗	✗	✗
Jogo Individual c/ Conexão	✗	✓	✓	✓	✓	✗
Jogo Multijogador c/ Conexão Direta	✓	✗	✗	✗	✓	✗
Jogo Multijogador c/ Conexão Rede Local	✓	✗	✗	✗	✓	✗
Jogo Multijogador c/ Conexão Internet	✓	✓	✓	✓	✓	✓

O método de investigação apresentado neste trabalho é adequado para os modelos de arquitetura de jogos de Rabin [9] do tipo *multiplayer*, ou seja, para aqueles jogos que envolvem 2 ou mais jogadores interagindo entre si por meio de uma conexão de *Internet* e, ainda, sabendo-se o nome virtual do alvo e interagindo com ele.

B. . Identificação de autoria por meio de endereços IP

Os jogos multijogadores *online* utilizam o protocolo TCP/IP para a troca de dados entre os jogadores, tais como eventos de sincronização do sistema do jogo. Assim é possível utilizar a análise de tráfego de redes para extrair informações da camada de redes para identificar um possível alvo.

Na Figura 3, são apresentadas as camadas do modelo OSI [12] utilizadas para realizar a comunicação entre o sistema de jogo e os jogadores

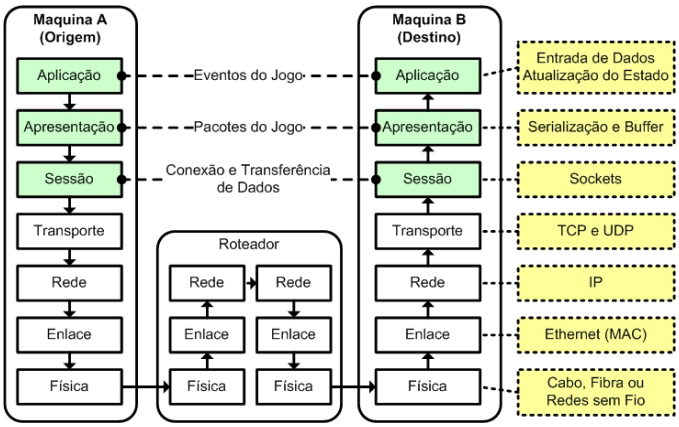


Fig. 3. Modelo OSI com as camadas de comunicação entre os jogos *online*. Fonte: Adaptado de Rabin [9].

O protocolo IP tem como principal característica a identificação única de um dispositivo de rede através de seu endereçamento. Existem duas versões do protocolo IP: IPv4 e IPv6. O IPv4 possui um endereçamento de 32 *bits* e o IPv6 utiliza um endereçamento de 128 *bits*.

Uma estratégia adotada por algumas instituições ou estabelecimentos para atenuar a escassez de endereços IPv4 é a utilização dos endereços privados de `10.0.0.0` a `10.255.255.255`, de `172.16.0.0` a `172.31.255.255` e de `192.168.0.0` a `192.168.255.255` por seus usuários. Assim, os dispositivos de rede, que utilizam esses endereços IPs, não podem comunicar-se diretamente com outros dispositivos na *Internet*, necessitando de um serviço de tradução de endereços IPs privados para endereços roteáveis na *Internet*. Comumente o serviço utilizado é o NAT (*Network Address Translation*) [11].

Para organizar o endereçamento IP, faixas de endereçamentos são distribuídos de forma hierárquica, sendo a IANA (*Internet Assigned Numbers Authority*) [17] a autoridade central responsável. No Brasil, a autoridade regional é a CGI.br [4], que, dentre outras atribuições, realiza a alocação de endereços IP no âmbito nacional.

Dessa forma, um provedor de conexão de *Internet* (também chamados de ISP – *Internet Service Provider*) deve contratar faixas de endereços IP através dos registros regionais (ou nacionais, quando houver) e, quando um cliente desse provedor se conectar à *Internet*, ele deverá receber um dos endereços IP pertencentes à faixa contratada [11].

Para poder utilizar a *Internet*, o dispositivo que interliga a rede local ao ISP deve se autenticar no provedor. O registro destas autenticações é de extrema importância para a investigação, pois através dele é possível identificar que um cliente iniciou uma conexão em determinada data e horário, qual o endereço IP recebeu, e a data e horário da desconexão. Em uma nova conexão, o mesmo usuário pode receber outro endereço IP, por isso é de suma importância que a investigação saiba, além do endereço IP, o momento exato que uma atividade ilícita ocorreu [11].

Assim, os IPs descobertos durante uma investigação estão vinculados aos ISPs, os quais podem ter sido utilizados por algum de seus clientes. Nesse caso, é necessário solicitar ao ISP qual o cliente que utilizava o endereço IP no dia e hora especificados. Essa solicitação muitas vezes é feita através de mandado judicial [8] (e que algumas vezes é precedida de uma requisição cautelar para preservar os dados de *logs*).

Da mesma forma, pode-se fazer uma solicitação para o provedor de aplicação para que o mesmo forneça os *logs* de utilização de seus serviços. Assim, o provedor de aplicação pode fornecer os dados do usuário de seus serviços como o IP e data e hora de utilização.

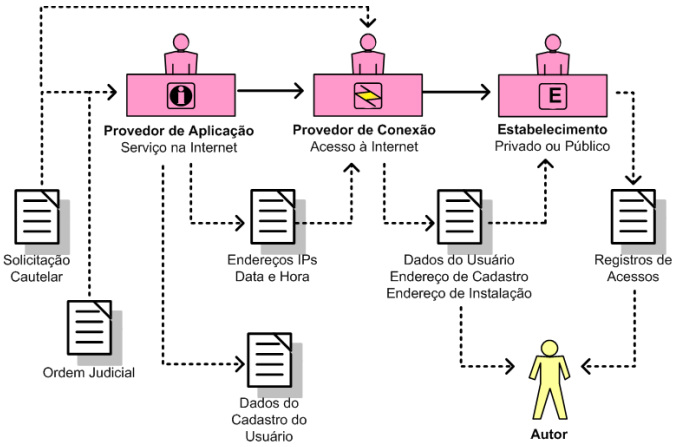


Fig. 4. Passos para verificar identificar o IP da conexão utilizada por um usuário.

Na Figura 4, é ilustrado o fluxo de informações fornecidas pelos provedores de aplicação, provedores de conexão e pelo estabelecimento. Uma solicitação cautelar de preservação de registros é feita pela autoridade policial, até que seja emitida a ordem judicial para que os provedores de aplicação e de conexão forneçam os registros de conexão (endereços IPs, data e hora) e os dados de cadastro do usuário que realizou as conexões. Eventualmente o acesso à *Internet* do autor pode ocorrer de forma indireta, através de um serviço de tradução de endereços IPs e, nesse caso, serão necessários os registros de acesso do estabelecimento que forneceu esse serviço ao autor. No final do processo, os dados de cadastro do usuário ou os registros de acesso permitirão localizar o possível autor de um delito.

Sobre os registros de *logs*, de acordo com o Marco Civil da *Internet* [14], os provedores de conexão devem manter os registros de conexão por um prazo de 1 ano. Já os provedores de aplicações devem manter seus registros por um prazo de 6 meses.

Assim, se os IPs de onde foram originadas as ações investigadas forem identificados, será possível descobrir o local de onde a conexão foi realizada, sendo também possível indicar a autoria das ações.

III. METODOLOGIA

Inicialmente foi criado um ambiente controlado de testes para verificar os procedimentos, as técnicas e as ferramentas de análise de tráfego de redes necessários para identificar os endereços IPs. Em seguida, foi realizado um estudo de caso, no qual foi aplicado e verificado os resultados dos instrumentos e procedimentos descritos na fase de testes. Tanto na fase de testes como na de estudo de caso foram colhidos os dados de tráfegos de redes, os quais foram filtrados e analisados. Por fim, são apresentados os resultados dessas análises.

A. Instrumentos e procedimentos

O perito deve coletar e analisar um conjunto de vestígios relacionados ao caso investigado. No caso específico de

investigação em ambiente de jogos *online*, onde se busca a identificação do autor através do endereço de IP utilizado por ele, o método de trabalho, mostrado na Figura 5, é sugerido para a realização de coleta e análise de dados de tráfegos de rede.

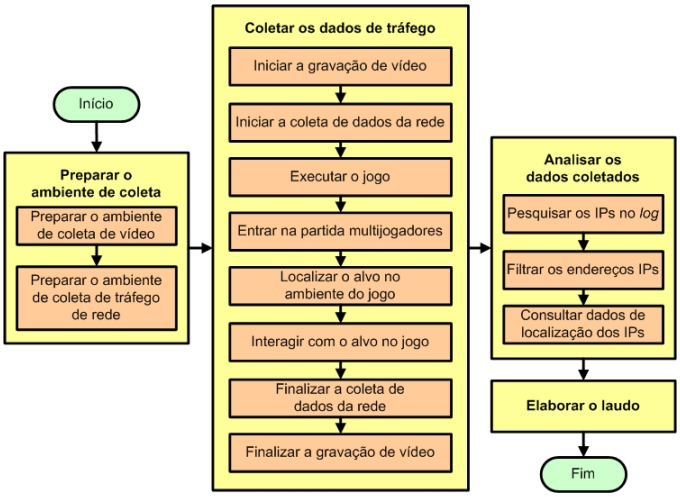


Fig. 5. Visão geral das fases de investigação e coleta de dados de tráfego de rede para jogos *online*.

O método apresentado consiste em 3 fases principais, quais sejam: (1) preparar o ambiente de coleta, (2) coletar os dados de tráfego, (3) analisar os dados coletados e (4) elaborar o laudo.

A fase de preparação do ambiente de coleta consiste na preparação do ambiente para coleta de dados de vídeo e dados de tráfego de redes. A coleta de vídeo é necessária para identificar os momentos do jogo nos quais ocorreram ou foram executados eventos de interesse pericial. Na Figura 6, são apresentados dois possíveis modelos de infraestrutura de equipamentos para realizar a coleta de dados de vídeo e rede, nos quais são utilizados uma câmera de vídeo, um monitor, um console de jogos, um computador coletor de dados da rede, um roteador e acesso à *Internet*.

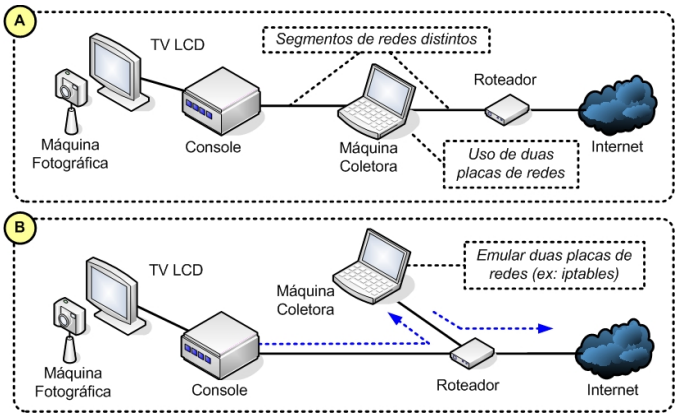


Fig. 6. Modelos de infraestrutura de equipamentos para a coleta de dados.

Na Figura 6-A, a máquina coletora possui duas placas de rede e fisicamente está *inline* no meio de acesso à *Internet*. Já na Figura 6-B, a máquina coletora emula duas placas de rede. Em ambos os casos, o console de jogos é configurado para utilizar a *interface* da máquina coletora como *default gateway*. Alternativamente, a máquina coletora pode ser configurada como ponte, conectando dois segmentos de redes distintos e operando em um nível mais baixo do modelo OSI (nível 2).

Na fase de coleta de dados de tráfego, deve-se iniciar a gravação da execução do jogo e a coleta dos dados de rede, os quais podem ser extraídos através do uso de *softwares* específicos como o *Wireshark* [12] e o *Tcpdump* [10]. Após a execução do jogo, deve-se localizar dentro do ambiente virtual do jogo o alvo. Nesse momento será necessária alguma interação com o alvo para que ocorra alguma troca de mensagens – e dessa forma ocorra a transmissão de pacotes IP.

Durante a fase de análise dos dados coletados, o perito deverá aplicar os filtros necessários para extrair os IPs possivelmente relacionados com o alvo. Pode-se proceder, então, a busca pela localização aproximada (e também as informações do provedor de conexão) de onde foi realizada a conexão que utilizou cada um dos IPs extraídos.

Por fim, os resultados dessas pesquisas deverão ser reportados no laudo pericial.

IV. ESTUDO DE CASO

O método descrito na *Seção III* foi aplicado na investigação de um caso real de furto em uma residência. Dentre os objetos furtados nessa residência havia um equipamento *Sony Playstation 3* (PS3) [15]. Dessa forma, foi percebido pela vítima que o criminoso (ou receptador) estava utilizando o perfil da vítima e jogando *online* o jogo *Call of Duty: Black Ops 2* [2] por meio do PS3 furtado, pois esse equipamento havia sido configurado para fazer o *login* automático na *PlayStation Network* (PSN) [16] com o perfil da vítima. Diante desse cenário, a autoridade policial solicitou a identificação do endereço IP de onde partiam os acessos do usuário da vítima para conectar na PSN.

A. Preparação do ambiente de coleta

Para a realização do exame é necessário que o console *Playstation 3* esteja conectado à *Internet*, para que ele possa acessar a PSN, e que os pacotes IP que chegam e saem do console sejam capturados, para análise posterior. Para isso, utilizou-se um computador entre o *Playstation 3* e a *Internet*, o qual realizava o roteamento dos pacotes e a sua captura. Assim, o modelo de infraestrutura utilizada foi o modelo ilustrado na Figura 6-B.

Buscando entender o funcionamento da PSN, foram realizados testes de diversas funcionalidades da rede do *Playstation 3*, tendo sido determinado que, para obter o endereço IP de um determinado jogador era necessário estar jogando o mesmo jogo e estar na mesma partida que ele.

Assim, através da conexão com a *Internet* é possível utilizar vários serviços da PSN. Há vários serviços de interação com outros usuários. Entretanto, a maioria das interações é intermediada pelos servidores da PSN, tais como *login*, *chat* e troca de mensagens. Entretanto, em jogos *online* de multijogadores, durante a interação direta entre os jogadores, ocorre a troca de mensagens entre esses jogadores, assim como entre os jogadores e o servidor da PSN. Nesse caso, como é mostrado na Figura 7, ocorre a troca de pacotes TCP e UDP com endereços IP específicos de cada usuário.

Dessa forma, durante o passo da coleta de dados da rede, os dados mais relevantes são aqueles coletados durante a interação com o alvo no ambiente virtual. Nesse sentido, a gravação em vídeo da execução da partida se torna relevante, já que através dela é possível identificar os momentos de interação com o alvo e, em seguida, filtrar os dados de tráfego relacionados a esse período.

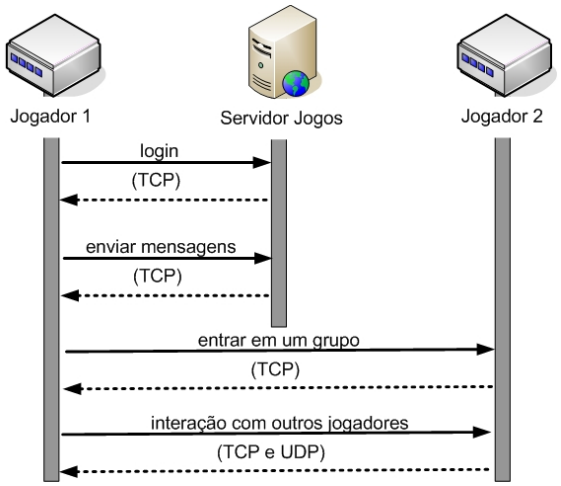


Fig. 7. Principais operações e protocolos utilizados durante a execução do jogo.

B. *Coleta dos dados de tráfego de rede*

O *software sniffer* instalado na máquina coletora foi o *tcpdump*. Através do comando `tcpdump -n` é possível iniciar a captura do tráfego. Assim, será mostrado todo o tráfego de rede, que passa pela primeira *interface* listada com o comando `tcpdump -D`, sem resolver nomes. Isso permitirá a visualização do tráfego em tempo real [7] [10].

Foram necessárias inúmeras tentativas para entrar na mesma partida ou jogo do alvo. Essa é a fase de formação dos grupos. Em geral, uma partida é iniciada com um grupo de 12 jogadores, divido em duas equipes rivais de 6 jogadores. Durante a partida alguns jogadores podem sair e outros jogadores podem entrar.

Na Figura 8, é apresentada a execução do jogo *Call of Duty: Black Ops 2*, no qual pode-se acessar quais são os nomes dos usuários que participam da partida. Na mesma figura é possível identificar os nomes utilizados pelo perito e pelo alvo dentro do ambiente do jogo.

Fig. 8. Imagem extraída do jogo *Call of Duty: Blacks Ops 2* [2].

A arquitetura frequentemente utilizada nos jogos multijogadores nos quais a posição de cada jogador é relevante é aquela baseada em mapas de 2 dimensões (2D). Tais mapas são divididos em quadrantes e quando ocorre a aproximação de jogadores no mesmo quadrante há a troca de pacotes TCP e UDP entre esses jogadores e a troca de pacotes TCP com o servidor de aplicação.

Na Figura 9, é mostrado graficamente um exemplo de uso de mapas 2D. Dessa forma, a fim de coletar os dados de tráfegos relevantes para a investigação, o perito deve "caçar" o alvo dentro do ambiente virtual.

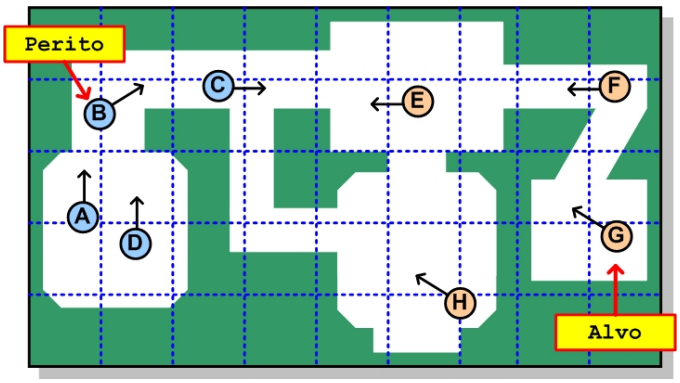


Fig. 9. Modelo de mapa 2D utilizado pelo jogo.
Fonte: Adaptado de Brackeen et al. [1].

Assim, foram feitas diversas tentativas para entrar no mesmo jogo e interagir com o alvo. As tentativas foram feitas em dias e horários diferentes. Durante essas tentativas foram capturados os pacotes IP que trafegavam entre o PS3 e a *Internet*, bem como foram tiradas fotos dos jogadores e gravados vídeos do jogo no ambiente virtual. Como resultado, foram coletados 3 arquivos de captura de tráfego de 3 jogos distintos nos quais o alvo esteve presente.

C. *Análise dos dados coletados*

Os pacotes IP de cada jogo foram filtrados, eliminando-se os IPs dos servidores da empresa *Sony* e pacotes não relacionados ao jogo (tais como pacotes de consultas DNS). A análise dos dados coletados foi realizada no ambiente *GNU Linux*, por meio de um interpretador de comandos *Shell* [18].

Os comandos destacados no Quadro 1 foram utilizados para realizar a extração ordenada dos endereços IPs, através do comando `egrep`, que pesquisa e apresenta os somente as partes do arquivo que possuem o padrão da entrada, e do comando `sort`, que ordena o resultado.

QUADRO 1. COMANDOS DE FILTRAGEM DE ENDEREÇOS IPS.

```
cat captura-jogo3.txt
↳ | egrep -o '[0-9]{1,3}\.[0-9]{1,3}\.
↳ [0-9]{1,3}\.[0-9]{1,3}'
↳ | sort > lista-ips-jogo.txt
> 177.17.138.229
> 177.17.138.229
> ...
> 177.193.12.107
> 177.193.12.107
> ...
> 177.41.254.5
> 177.41.254.5
> ...
```

Em seguida, foram extraídos os IPs do servidor de jogos da PSN, assim como os endereços da rede local (ver Quadro 2).

QUADRO 2. LISTA DE IPS DE REDE LOCAL.

```
>    10.0.0.0 -  10.255.255.255(10/8 prefix)
>  172.16.0.0 -  172.31.255.255(172.16/12 prefix)
> 192.168.0.0 - 192.168.255.255(192.168/16 prefix)
```

Sobre a lista de IPs resultantes, foram contadas o número de ocorrências de cada IP, como apresenta o Quadro 3. Nesse quadro, o comando `uniq -c` é usado para contar o número de repetições da entrada ordenada, e o comando `sort -r` ordena o resultado em ordem reversa. O resultado é apresentado em

duas colunas, que são o número de ocorrências e o endereço IP relacionado.

QUADRO 3. COMANDOS DE CONTAGEM DE FREQUÊNCIA DOS ENDEREÇOS IP.

```
cat lista-ips-jogo3.txt | uniq -c
↳ | sort -r > lista-ips-count-rev.txt
> 3385 177.80.118.68
> 2630 201.52.52.162
> 2540 200.206.146.200
> 1835 186.220.199.241
> 1830 177.193.12.107
> 1715 187.39.163.177
> 1255 189.102.126.154
>  905 189.123.225.196
>  220 187.39.71.57
>  154 201.92.213.21
>  120 179.197.122.237
>  100 189.120.250.17
>   55 177.41.254.5
>   55 177.17.138.229
>   45 190.176.222.87
```

Após essa filtragem, os endereços IP restantes possuíam quantidade compatível com o número de jogadores presentes em cada jogo. Deve-se ressaltar que, em alguns jogos, a quantidade de IPs ultrapassa o número máximo de jogadores em cada partida (12 jogadores) devido ao fato de, durante o jogo, alguns jogadores deixarem a partida e outros entrarem em seus lugares.

Em seguida, foi realizado um cruzamento de dados para verificar qual IP ou qual par provedor/cidade se mantinha constante nos três jogos. Foi necessário analisar o par provedor/cidade, e não somente o IP, pois, como a maioria dos acessos domésticos à *Internet* utilizam alocação dinâmica de IP, é possível que o IP de um determinado jogador tenha sido trocado entre os três jogos analisados, principalmente nos que ocorreram em dias distintos.

Uma das consultas de localização foi através do `geoiplookup`, utilizando os comandos do Quadro 4, cujo resultado da execução dos comandos são duas colunas com o endereço IP e os dados de localização.

QUADRO 4. COMANDOS DE CONSULTAS DE LOCALIZAÇÃO DE ENDEREÇOS IPS.

```
for i in $(cat lista-ips-uniq.txt);
   do r=$(geoiplookup $i);
   echo $i - $r;
done
> 177.17.138.229  - GeoIP: BR, Brazil
> 177.193.12.107  - GeoIP: IP Address not found
> 177.41.254.5    - GeoIP: BR, Brazil
> 177.80.118.68   - GeoIP: BR, Brazil
> 186.220.199.241 - GeoIP: BR, Brazil
> 187.39.163.177  - GeoIP: BR, Brazil
> 187.39.71.57    - GeoIP: BR, Brazil
> 189.1.174.20    - GeoIP: BR, Brazil
> 189.102.126.154 - GeoIP: BR, Brazil
> 189.120.250.17  - GeoIP: BR, Brazil
> 189.123.225.196 - GeoIP: BR, Brazil
> 190.176.222.87  - GeoIP: AR, Argentina
> 200.206.146.200 - GeoIP: BR, Brazil
> 201.52.52.162   - GeoIP: BR, Brazil
> 201.92.213.21   - GeoIP: BR, Brazil
```

Além da consulta de localização do `geoiplookup` outras ferramentas para levantamento de dados podem ser utilizadas: `geoip`, `nslookup`, `wget` e `whois`. O Quadro 5 apresenta um exemplo de resultado da consulta `whois`.

QUADRO 5. RESULTADO DA CONSULTA DO COMANDO WHOIS.

```
whois 177.193.12.107
> ...
> inetnum:      177.192/14
> aut-num:      AS28573
> abuse-c:      GRSVI
> owner:        NET Servicos de Comunicacao S.A.
> ownerid:      000.108.786/0001-65
> responsible:  Grupo de Seguranca da Informacao Virtua
> country:      BR
> owner-c:      GRSVI
> tech-c:       GRSVI
>...
```

A Tabela 2 mostra o resultado consolidado das coletas realizadas em 3 jogos distintos (nos quais houve a participação do usuário alvo), com os dados de IP, de localização e do provedor de conexão.

TABELA II. RESULTADO DA ANÁLISE DOS IPS COLETADOS

Jogo 1	Jogo 2	Jogo 3	Provedor/Cidade
	54.9.9.4		Woodbridge – NJ - USA
177.133.29.165	**177.133.29.165**		**GVT - Sobradinho - DF**
177.143.201.90			Virtua - Farroupilha - RS
177.4.237.10			**Brasil Telecom S/A - Filial DF**
		177.17.138.229	**GVT Brasília – DF**
	177.32.41.148		Virtua - Sao Paulo - SP
		177.41.254.5	GVT - Joinville - SC
177.42.208.116			GVT - Salvador - BA
		177.80.118.68	Virtua - Sao Paulo - SP
177.96.164.233			GVT - Palhoça - SC
177.96.177.158			GVT - Curitiba - PR
177.96.38.186			GVT - Palhoça - SC
177.100.114.223			VCB - Macaé - RJ
177.106.213.223			CTBC - Uberlândia - MG
177.106.245.43			CTBC - Uberlândia - MG
	177.141.117.100		Virtua - São Paulo - SP
	177.179.234.217		Oi Velox - Rio de Janeiro - RJ
		177.193.12.107	Virtua - São Luís - MA
		179.197.122.237	Oi Velox - Brasil
		186.220.199.241	Virtua - Sao Paulo - SP
	187.35.24.94		Vivo - São Paulo - SP
		187.39.71.57	Virtua - Pindamonhangaba - SP
		187.39.163.177	Virtua - Bento Gonçalves - RS
187.112.240.107			GVT - Cascavel - PR
		189.1.174.20	Hostlocation - São Paulo - SP
	189.6.13.13		Virtua - Brasil
	189.27.84.106		GVT - Campo Grande - MS
189.73.249.39			BR Telecom DF - MS
189.81.47.23			Velox - João Pessoa - PB
189.85.178.19			Newsite - Palhoça - SC
	189.102.45.194		Virtua - São Paulo - SP
		189.102.126.154	Virtua - São Paulo - SP
		189.120.250.17	Virtua - São Paulo - SP
		189.123.225.196	Virtua - Curitiba - PR
		190.176.222.87	Telefonica – Argentina
200.193.245.146			**BR Telecom DF - Brasilia - DF**
		200.206.146.200	Vivo - Indaiatuba - SP
	201.22.89.163		GVT - Maringá - PR
		201.52.52.162	Virtua - Sao Paulo - SP
	201.74.34.15		Virtua - São Bernardo - SP
201.86.0.95			GVT - Curitiba - PR
		201.92.213.21	Vivo - Santana de Parnaíba - SP

São destacados na Tabela II, em negrito, os endereços IPs sediados no Distrito Federal. O IP `177.133.29.165` foi o

único a se manter nos jogos 1 e 2 e o IP 177.17.138.229 foi o único do Distrito Federal no jogo 3. Ademais, ambos IPs são da empresa GVT, sendo, portanto, os IPs mais prováveis de estarem vinculados ao usuário alvo.

Foi realizado, também, o cruzamento dos nomes dos jogadores capturados por fotografias durante as partidas. Ressalta-se que as fotografias mostram os jogadores presentes no jogo em determinado momento, pois ocorrem algumas trocas de jogadores durante o jogo.

Assim, foi preparado o laudo para a autoridade policial, relatando o método de trabalho, assim como os resultados obtidos. Esses resultados incluem um conjunto de registros contendo os dados de IP, provedor, cidade, data e hora. Esses dados são suficientes para buscar, por meio do provedor de conexão, o usuário que utilizou tais IPs nos períodos especificados.

V. Conclusões

Os ambientes virtuais criados na *Internet,* tais como os jogos *online* multijogadores são cada vez mais utilizados para cometer crimes. Por meio de uma metodologia de investigação baseada na análise de tráfego de redes é possível coletar mensagens de protocolos de rede como o TCP, UDP e IP para auxiliar no estabelecimento da autoria de um crime, até mesmo de crimes iniciados fora desse ambiente, como um furto.

Para isso, é necessário entender o contexto dos jogos *online*. Dessa forma, foi apresentado o modelo lógico da arquitetura de um sistema de jogo, tendo como principais camadas a aplicação, lógica, visão do jogo e visão remota. Também é necessário entender as várias formas de interação entre os jogadores e o sistema de jogo (*single player*, *multiplayer*, etc.).

A metodologia proposta neste trabalho é adequada para os modelos de arquitetura de jogos do tipo *multiplayer*, ou seja, para aqueles jogos que envolvem 2 ou mais jogadores interagindo entre si através de uma conexão de *Internet* e, ainda, sabendo-se o nome virtual do alvo e interagindo com ele. Ela inclui a preparação do ambiente, a coleta de dados de tráfego de redes, a análise dos dados coletados e a apresentação do laudo.

A aplicação da metodologia em um caso real demonstrou a viabilidade de coletar dados de vídeo e rede em serviços de jogos *online*. Também revelou a importância das informações de vídeo para identificar os eventos de interesse pericial, nos quais ocorreram a interação com o alvo, os quais, após um processo de análise e filtragem, permitiram identificar os IPs relacionados ao alvo da investigação.

Esses endereços IP (em data e hora delimitados) são essenciais para o prosseguimento da investigação, que passará a depender de dados fornecidos pelo provedor de aplicação, provedor de conexão e, em alguns casos, de estabelecimento público ou privado. Sem eles, não é possível descobrir o local de onde a conexão foi realizada e, consequentemente, estabelecer a autoria das ações.

A aplicação da metodologia em outras plataformas e serviços de jogos *online* é uma das possibilidades de trabalhos futuros. Outra possibilidade é a avaliação do impacto da utilização de servidores *proxy* (e outras técnicas de camuflagem) na metodologia proposta, especialmente na coleta de vestígios.

Referências

[1] Brackeen, David; Barker, Bret; Vanhelsuwé, Laurence. “Developing Games in Java”. New Riders, 2003.

[2] *Call of Duty* disponível em http://www.callofduty.com/, acessado em 30/05/2015.

[3] Cardoso, Nágila Magalhães; Hashimoto, Yuri Campos; da Silva, Keith Maila Domingos; Maia, Anderson Trindade. “Redes sociais a nova arma do crime cibernético: O efeito do uso da engenharia social e da esteganografia”. The International Journal of Forensic Computer Science (ICoFCS), 2011.

[4] CGI, disponível em http://cgi.br/, acessado em 30/05/2015.

[5] Fullerton, Tracy. “Game Design Workshop: A Playcentric Approach to Creating Innovative Games”. A K Peters/CRC Press, 2014.

[6] Mcshaffry, Mike; Graham, David. “Game Coding Complete”. Course Technology PTR, 2012.

[7] Mota Filho, João Eriberto. “Análise de Tráfego em Redes TCP/IP: utilize tcpdump na análise de tráfego em qualquer sistema operacional”. Novatec Editora, 2013.

[8] Peron, André; de Deus, Flávio Elias Gomes; de Sousa Júnior, Rafael Timóteo. “Ferramentas e Metodologia para Simplificar Investigações Criminais Utilizando Interceptação Telemática”. The International Journal of Forensic Computer Science (ICoFCS), 2011.

[9] Rabin, Steve. “Introduction to Game Development”. Course Technology PTR, 2009.

[10] Tcpdump, disponível em http://www.tcpdump.org/, acessado em 30/05/2015.

[11] Vecchia, Evandro Della. “Perícia Digital: da investigação a análise forense”. Millenium Editora, 2014.

[12] Zimmermann, Hubert. "OSI Reference Model - The ISO Model of Architecture for Open Systems Interconnection". IEEE Transactions on Communications, 1980.

[13] Wireshark, disponível em http://www.wireshark.org/, acessado em 30/05/2015.

[14] Brasil. Lei no 12.965, de 23 de abril de 2014.

[15] *Sony Playstation 3*, disponível em http://br.playstation.com/ps3/, acessado em 30/05/2015.

[16] *PlayStation Network*, disponível em http://br.playstation.com/psn/, acessado em 30/05/2015.

[17] *Internet Assigned Numbers Authority*, disponível em http://www.iana.org, acessado em 30/05/2015.

[18] Neves, Julio Cezar. “Programação Shell Linux”. Editora Brasport, 2010.

Juliano K. M. Oya e Cleber Scolarick Junior, Peritos Criminais, Seção de Perícias em Informática, Instituto de Criminalística – Polícia Civil do Distrito Federal, Brasília-DF, Brasil. E-mails: juliano.oya@gmail.com e scoralick@gmail.com

Bruno W. P. Hoelz, Perito Criminal, Instituto Nacional de Criminalística – Polícia Federal, Brasília-DF, Brasil. E-mail: werneck.bwph@pdf.gov.br

DOI: 10.5769/C2015008 or http://dx.doi.org/10.5769/C2015008

Maldetect: Uma metodologia automatizável de detecção de *malwares* desconhecidos

Leandro Silva dos Santos, Dino Macedo Amaral

***Resumo*— O cenário de ataques cibernéticos está atingindo níveis cada vez mais altos de complexidade. Com isso as ferramentas tradicionais de detecção e remoção de ameças estão cada vez menos eficiente, pois utilizam um abordagem de detecção baseada em assinatura. Este trabalho propõe uma metodologia automatizável de detecção de *malwares* desconhecidos, ou seja, aqueles que não foram detectados pelas ferramentas tradicionais. A metodologia *Maldetect* apresentada neste artigo coleta e correlaciona características comportamentais típicas de códigos maliciosos, sendo independente de sistema operacional. Foi construída uma ferramenta usando as linguagens de programação PHP e Python, denominada *Maldetect Tool* que automatiza a metodologia proposta. A partir do *dump* da memória volátil, a *Maldetect Tool* gera um relatório contendo os processos que mais realizam atividades típicas de *malwares*. A *Maldetect Tool* analisou de maneira automatizada *dumps* de memória de estações infectadas e foi capaz de detectar os artefatos maliciosos a partir da análise da memória volátil.**

***Palavras-Chave*— Forense, análise de memória, detecção de *malwares* desconhecidos, volatility, maldetect**

***Abstract*— The scenario of cyber attacks is reaching ever higher levels of complexity. Thus the traditional tools of threats detection are becoming less efficient because they use mainly signature-based detection. This work proposes a automatable methodology of unknown malware detection, ie those that were not detected by traditional tools. The Maldetect methodology presented in this paper collects and correlates typical behavioral characteristics of malicious code and is independent of operating system. A so-called Maldetect Tool that automates the proposed methodology was built using Python and PHP programming languages. From the dump of volatile memory, Maldetect Tool generates a report containing the processes that perform more typical activities of malware. The Maldetect Tool analyzed in an automated approach memory dumps from infected machines and was able to detect malicious artifact from the analysis of volatile memory.**

***Keywords*— Forensics, memory analysis, unknown malwares detect, volatility, maldetect**

I. Introdução

O cenário de ataques cibernéticos acompanha a modernização das ferramentas de detecção e remoção, o que os tornam cada vez mais complexos. A indústria de antivírus (AV) tem se demonstrado ineficiente contra ameaças avançadas, principalmente por utilizar detecção baseada em assinaturas. Este tipo de detecção é facilmente burlado com técnicas de polimorfismos e metamorfismo[2]. Apesar disso, o AV ainda possui seu espaço no arsenal de segurança da informação.

Leandro Silva dos Santos, Dino Macedo Amaral. Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília-DF, Brasil, E-mails: holyminds@gmail.com, dinoamaral@gmail.com

Inicialmente, os *malwares* tinham objetivos simples, como apagar arquivos ou provocar erros, ou ainda executar atividades indesejadas em um computador, as quais eram percebidas pelos usuários. Porém, com o avanço dos *malwares*, os mesmos são capazes de capturar e até sequestrar dados relevantes das vítimas. Este último tipo de *malware* (chamados de *rasomware*[1]) criptografa os arquivos da vítima e pedem pagamento pela decifração destes arquivos [3].

Existe ainda o conceito de *Advanced Persistent Threats* - APT (Ameaça Persistente Avançada), a qual geralmente possui alvos específicos e utiliza técnicas avançadas, como a exploração de uma ou mais vulnerabilidades *0-day* e o uso de certificados falsificados, para comprometer as estações de seus alvos[21]. Assim, na maior parte dos casos, não são produzidos por indivíduos isolados, mas sim por instituições, crime organizado ou governos que mediante objetivos específicos ajudam a financiar tais atividades[3]. Este tipo de ameaça geralmente é usada em atividades de espionagem ou sabotagem [10].

Diante deste cenário, a análise de memória volátil consiste em umas das principais técnicas para analisar ameaças avançadas, por ser eficiente na identificação de características comportamentais típicas de *rootkits* e outros tipos de *malwares*[18]. Além disso, a análise de memória permite reconstruir o estado original do sistema, quais arquivos estão sendo acessados, as conexões de redes que foram abertas, dentre outros dados relevantes para a identificação de código malicioso[11].

Dessa forma, este trabalho propõe uma metodologia automatizável de análise de memória volátil, denominada *Maldetect*. Esta é capaz de coletar caraterísticas comportamentais e correlacionar as informações de forma a identificar quais são os processos candidatos a *malware*. Além da metodologia, é apresentada uma implementação da Maldetect que automatiza a análise do *dump* de memória volátil do sistema operacional Window 7, bem como os resultados obtidos dessa análise.

Este trabalho está organizado da seguinte forma: a seção II descreve os trabalho relacionados e mostra algumas soluções de automatização de análise de memória volátil que já foram propostas. A seção seguinte apresenta três metodologias de análise de memória para detecção de artefatos maliciosos. A seção IV descreve a ferramenta Volatility usada para implementação da automatização da Maldetect. A seção V detalha as fases da metodologia Maldetect para detecção de *malwares* desconhecidos a partir da análise do *dump* de memória volátil. A seção VI descreve as técnicas utilizadas na construção de uma ferramenta que implementa a metodologia Maldetect para o sistema operacional Windows 7. A seção VII apresenta os resultados obtidos a partir da execução da Maldetect para detecção de artefatos maliciosos em *dumps* de memória volátil de estações infectadas e por fim, na última

seção, são feitas as considerações finais.

II. Trabalhos Relacionados

Análise de memória foi um dos principais temas do desafio de 2005 do Digital Forensic Research Workgroup (DFRWS), o que motivou um esforço de pesquisa e desenvolvimento de ferramentas nesta área[4]. Este desafio deu início aos estudos de análise de memória usando técnicas forenses. Em [20], no ano de 2011, Vömel publicou um *survey* que apresenta várias técnicas de aquisição de memória baseada em software e hardware. Mostrando também várias técnicas de análise de processos, de recuperação de chave criptográfica, de análise de registro de sistemas, de redes, de arquivos, do estado do sistema e de uma aplicação específica baseada nas estruturas de memória do sistema operacional. Essas técnicas são amplamente utilizadas em análise manual de memória.

Em [7], no ano de 2013, Liang Hu et al. mostrou a importância de automatizar o processo de análise de memória volátil. O artigo propõe automatizar a análise de memória baseada em dois fluxos de análise (*DLL flow* e *Process flow*). Em cada fluxo são coletados diversos dados que serão processados e correlacionados gerando um relatório. Porém apenas os dados do fluxo de DLL são processados automaticamente.

Em [5], no ano de 2014, outra solução de automatização da análise de memória foi apresentada por Fahad - *Associate Director – Security Research and Analytics UBS AG*. Ela é composta de três fases: primeiro aquisição da memória para um *drive* seguro que fique oculto para o usuário. Segundo, é a execução do Volatility para extração das informações relevantes contidas no *dump* da memória. As duas primeiras fases serão executas a cada 30 minutos. Por fim, essas informações são enviadas para um servidor central que fará a análise. Esta fase executará um algoritmo de comparação do *dump* atual com as informações contidas na base de dados. Assim é possível identificar a criação de novas conexões de redes, novos serviços, alteração e criação de chaves de registros, entre outros dados. O problema é o aumento do processamento do *host* e o volume de dados sendo transferidos pela rede.

Como a análise de memória tem sido amplamente aplicada na identificação de código malicioso, existem alguns cuidados que devem ser levados em consideração na fase de coleta da memória volátil, pois o processo de aquisição, geralmente, requer a execução de código na máquina infectada. Este processo pode ser interferido pelo *malware* em execução. Em [18], no ano de 2013, Johannes analisou várias técnicas antiforense que interferiam na aquisição da memória e testou as principais ferramentas com o objetivo de identificar quais delas seriam resistentes a estas técnicas. O resultado encontrado pode ser observado na figura 1 e nenhuma ferramenta foi resistente a todas as principais técnicas antiforense.

Dessa forma, este trabalho propõe uma metodologia automatizável de detecção de *malwares* desconhecidos[1] baseada em características comportamentais dos processos, DLLs, e drivers em execução no sistema operacional. O dados serão extraídos da imagem da memória volátil e correlacionados com outras fontes, como o VirusTotal[2] e *blacklist* de IP's. Além disso, a metodologia é independente de sistema operacional.

[1] *Malwares* que não foram detectados pelas ferramentas tradicionais de detecção de códigos maliciosos

III. Metologias de Análise de Memória

No campo da computação forense, a análise de memória pode trazer resultados mais proveitosos que a análise de artefatos de disco, já que a análise de memória identifica as ações que estão sendo executadas pelo sistema operacional e pelos aplicativos em execução no momento da coleta do *dump* da memória volátil. Além disso, a análise de memória pode prover várias informações sobre o estado do sistema em tempo de execução, por exemplo: quais processos estão em execução, conexões de rede abertas e comandos executados recentemente. Os dados que ficam criptografados no disco, geralmente não estão criptografados quando executados na memória. Também é possível encontrar na memória chaves criptográficas, arquivos confidenciais e histórico dos *browsers* no modo de navegação anônima [12].

Dessa foram, a seguir são descritas três metodologias de análise de memória volátil para detecção de artefatos maliciosos, as quais constituirão a base da metodologia proposta neste trabalho.

A. Metodologia do SANS - System Administration, Networking and Security

A metodologia do SANS de análise de memória está focada em busca de artefatos maliciosos residentes em memória, ou seja, em execução no sistema operacional. Sua descrição não está concentrada em um único documento, mas é descrita em alguns casos de uso e *posters* públicos. Em [9] a metodologia é apresentada como sendo o nono passo da busca por um *malwares* desconhecidos, como mostra a Figura 2. A metodologia proposta pelo SANS é composta de seis passos, alguns desses passos já são executados em uma análise padrão de memória, e outros são específicos para encontrar artefatos maliciosos. A análise de memória volátil nos fornece melhores resultados para identificar técnicas usadas por *rootkits*, os quais procuram dificultar sua detecção.

- **Identificar processos estranhos**

  Na fase de análise de processos devemos coletar algumas informações, tais como: nome do processo, caminho em disco, processo "pai", linha de comando, hora de inicialização e SIDs. Esses dados serão usados para: identificar processos legítimos; verificar a escrita correta do nome; identificar caminhos suspeitos dos processos; verificar o "pai" do processo; e identificar parâmetros de linha de comando que iniciou o processo.

- **Analisar DLLs e handles[3] de processos**

  Existe uma diferença fundamental entre programa e processo. Um programa é uma sequencia estática de

[2] VirusTotal é um serviço gratuito que analisa arquivos e URL's suspeitas e facilita a rápida detecção de vírus, worms, cavalos de tróia e todos os tipos de *malwares*. Acesso em: https://www.virustotal.com/

[3] Referência abstrata para um recurso[16].

Acquisition tool	Version	Format	KDBG	MmGetPhysical memory-ranges()	MmMap-MemoryDump-Mdl()
Memoryze	2.0	raw	PASS	FAIL	PASS
FTK Imager	3.1.2	raw	PASS	FAIL	PASS
Win64dd	1.4.0	raw	PASS/FAIL	FAIL	FAIL
Win64dd	1.4.0	dmp	FAIL	FAIL	FAIL
DumpIt	1.4.0	raw	PASS	FAIL	FAIL
WinPmem	1.3.1	raw	FAIL	FAIL	PASS
WinPmem	1.3.1	dmp	FAIL	FAIL	PASS
WindowsMemoryReader	1.0	raw	PASS	FAIL	PASS
WindowsMemoryReader	1.0	dmp	PASS	FAIL	PASS

Fig. 1. Resultado da aquisição de memória com técnicas de antiforense ativada[18].

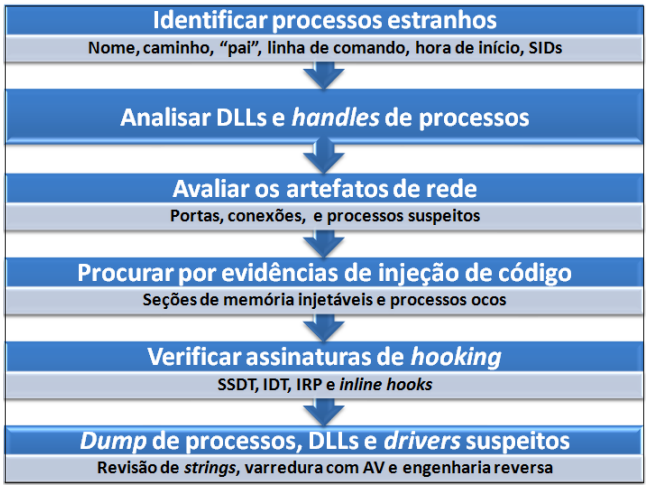


Fig. 2. Metodologia de análise de memória do SANS, adaptado de [9].

instruções; já um processo contém um conjunto de recursos usados por uma instância de um programa. Em [16] são apresentados componentes de um processo do sistema operacional Windows:

* Um espaço de endereçamento virtual;
* Um programa executável;
* Um lista de *handles* para vários recursos do sistema (por exemplo: portas de comunicação, semáforos e arquivos abertos);
* Uma lista de *Threads*;
* Um contexto de segurança; e
* Lista de DLLs (*dynamic-link libraries*) associadas.

Nesta fase são analisados os *handles* associados aos processos de maneira a identificar alguma atividade maliciosa, como por exemplo DLLs que apesar de terem nomes legítimos estão em caminhos diferentes no sistema de arquivos do sistema operacional. Além disso, muitos *malwares* usam um *handle* do tipo *mutex*[4] para identificar se o *malware* já foi instalado na máquina da vítima e assim não executar nada.

- **Avaliar os artefatos de rede**

Durante a análise dos artefatos de rede deve-se identificar as portas TCP suspeitas e os processos associados a elas, bem como os indicativos da presença de *backdoors* e a reputação dos IP's que a máquina está conectada. Dessa forma, podemos identificar atividades tipicas de códigos malicioso com relação as conexões de rede[19].

- **Procura por evidências de injeção de código**

Em [19], são apresentadas duas técnicas de injeção de código:

* *Injeção de DLLs*: muito utilizada pelos *malwares*, eles utilizam algumas chamadas de sistema, tais como: *VirtualAllocEx(), CreateRemoteThread e SetWindowsHookEx()* para carregar uma DLL em um processo já em execução; e
* *Processos ocos*: o *malware* cria uma nova instância de um processo legítimo e substitui a área de código do processo legítimo pelo seu código malicioso e obtêm as DLLs, os *handles* e outros recursos do processo original.

A detecção de injeção de código pode ser feita varrendo a memória procurando setores marcados com permissão de escrita e execução e que não tenham um mapeamento para um arquivo. Também pode ser feita uma comparação entre o código de memória e código do arquivo em disco para verificar o nível de similaridade[19].

- **Verificação de assinaturas de *hooking***

Basicamente existem quatro técnicas utilizadas pelos *rootkits* que devem ser verificadas nesta fase da metodologia. Deve-se verificar a *System Service Descriptor Table* (SSDT) - tabela que contém um *array* de ponteiros para as funções de tratamento de cada *system call*. As entradas da SSDT podem ser alteradas pelos *rootkits* e assim alterar a saída ou a entrada das chamadas de sistema, de forma a esconder processos, arquivos e chaves de registros [16].
Além disso, *drivers* maliciosos podem utilizar a *Interrupt Descriptor Table* - estrutura de dados que armazena os endereços das funções de tratamento de interrupção e exceções de processos - para realizar um *hook*[5] em todas

[4]Mecanismo de sincronização usado para serializar o acesso à um recurso [16].

[5]Técnica utilizada por *rootkits* para modificar o comportamento normal de uma ação do sistema operacional ou processo.

rotinas de tratamento ou em apenas um ponto [11].
Outra técnica utilizada é o *hook driver* (*Hooking the I/O Request Packet* - IRP), na qual os *rootkits* alteram a IRP - estrutura de dados que contém códigos para identificar as operações desejadas e *buffers* de dados que serão lidos ou escritos pelo *driver*. Geralmente, o módulo tcpip.sys é atacado com esta técnica[11].
Por último, pode ser usada a *inline hook*, também conhecida como *Dynamic code patching*, que sobrescreve os primeiros *bytes* de um função com a instrução de *jump* (instrução JMP do *assembly* [8]) para redirecionar a execução para a função do código malicioso, e ao final de sua execução retornar para a função original [15].

- **_Dump_ de processos, DLLs e _drivers_ suspeitos**

  Nesta fase, espera-se obter uma lista dos possíveis artefatos malicioso que precisam de uma análise mais profunda. Para esses possíveis *malwares* deve-se realizar um *dump* do processo correspondente da memória e realizada uma revisão de strings, varredura com antivírus, engenharia reversa e outras técnicas que possibilitem a detecção de atividades maliciosas.

B. Caçando Malwares nos processos em memória

Em [12], Michael Hale et al. descreve sete objetivos da análise de memória para se encontrar um *malware*, que são:

- **Recuperar linhas de comandos e caminho dos processos**

  O *Process Enviroment Blobk* (PEB), que é membro da estrutura de memória _EPROCESS, contém o caminho completo do processo, a linha de comando que iniciou o processo, ponteiros para a *heap* do processo, entre outras informações. Essas informações ajudam a localizar o arquivo executável no disco e descobrir informações sobre como o processo foi instanciado na memória da estação infectada.

- **Analisar _heaps_**

  Os dados que as aplicações manipulam (dados recebidos via rede, ou textos digitados em um processador de texto) possuem uma boa chance de estarem armazenados na *heap* do processo, assim não perde-se muito tempo pesquisando em regiões de memória que contém DLLs, arquivos mapeados e a pilha.

- **Inspecionar variáveis de ambiente**

  Existem famílias de *malwares* que marcam sua presença com a criação de variáveis de ambiente. Outros *malwares* manipulam os valores das variáveis de ambientes para gerar comportamentos maliciosos em outros processos. Algumas variáveis que são tipicamente manipuladas por códigos maliciosos, são:
  * PATH: armazena o caminho dos executáveis;
  * PATHEXT: extensões atribuídas aos programas executáveis;
  * Caminho dos diretórios temporários;
  * Caminho dos diretórios de documentos, histórico de internet e dados de aplicações dos usuários; e
  * ComSpec: localização do cmd.exe;

- **Detectar _backdoors_ com _handles_ padrões**

  Identifique se a entrada e saída de um processo estão sendo direcionados a um *socket* de rede remoto para um atacante. Uma técnica muito comum, usada pelos *backdoors* é criação de um *socket* de rede associado a um processo cmd.exe de tal forma que toda saída do processo seja transmitido pela rede e toda entrada do *socket* seja transformada em entrada para o processo.

- **Enumerar DLLs**

  Os *Dynamic Link Libraries* possuem códigos e recursos que podem ser compartilhados entre processos, por isso é muito comum entre os *malwares* a técnica de injetar DLLs em processos legítimos. Durante a análise de DLLs deve-se verificar se existe alguma não vinculada, se o caminho das DLLs no sistema de arquivos são adequados e o contexto em que as mesmas estão carregadas.

- **Extrair arquivos PE da memória**

  Pode ser realizado o *dump* do conteúdo em memória dos programas executáveis para uma análise mais profunda deste artefato. Porém um processo ao ser carregado na memória sofre algumas alterações que devem ser levadas em consideração durante a análise do artefato extraído. Por exemplo, o *hash* md5 do *dump* do processo extraído da memória pode não ser o mesmo do *hash* do arquivo no disco, mas é possível usar um *fuzzy hash*[22] para determinar o grau de similaridade.

- **Detectar injeção de código**

  São apresentados quatro técnicas de injeção de código:
  * Injeção remota de DLLs: o processo malicioso força o processo alvo a carregar uma DLL específica;
  * *Injeção remota de código*: o processo malicioso escreve código na área de memória do processo alvo e força sua execução;
  * *Injeção reflexiva de DLL*: o processo malicioso escreve o código da DLL no espaço de memória do processo alvo; e
  * *Injeção em processo oco*: o processo malicioso inicia uma nova instância de um processo legítimo em modo suspenso e então é feita uma sobrescrita da área de código do processo legítimo pelo código malicioso e sua execução é iniciada.

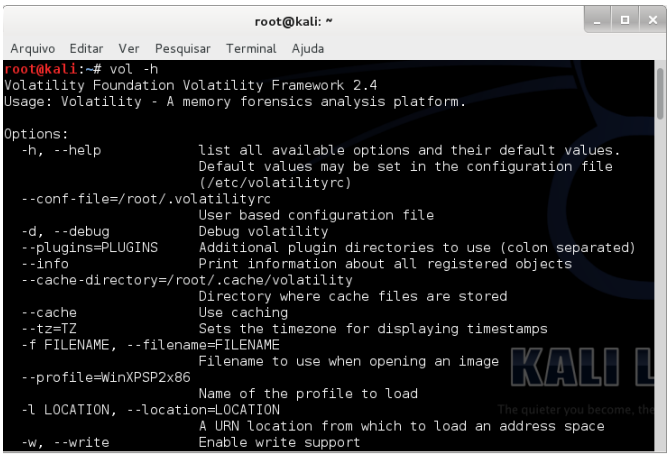


Fig. 3. Volatility em linha de comando.

C. *Metodologia de análise de memória*

Em [14], são descritos os objetivos da análise de memória, especificamente no contexto de análise de código malicioso:

- Coletar os metadados disponíveis, tais como: detalhes de processos, conexões de rede, e outras informações associadas ao potencial *malware*;
- Para cada processo de interesse, se possível, recuperar o arquivo executável da memória para análise; e
- Para cada processo de interesse extrair mais dados da memória, por exemplo, usuários, senhas e chaves criptográficas.

IV. *Volatility Framework*

O *Volatility Framework* é uma coleção de ferramentas, implementada em Python, capaz de extrair artefatos digitais de um *dump* da memória volátil (RAM). O Volatility é licenciado pela *GNU General Public License 2*, possui código aberto e é gratuito. Sua arquitetura permite a inclusão de novas funcionalidades através da criação de novos *plugins* [12].

O Volatility é capaz de analisar o *dump* de memória das versões 32-bits e 64-bits dos sistemas operacionais Windows, Linux, Mac e 32-bits do Android. O Volatility suporta a inclusão de novos sistemas operacionais devido a sua arquitetura modular. Porém o volatility não é uma ferramenta de aquisição de memória e também não possui interface gráfica, seu uso é através de linha de comando [12], como mostra a Figura 3.

Em [6], são apresentados os *plugins* do volatility, os quais são agrupados na seguintes categorias:

* *Image Identification*: identificação do sistema operacional e suas estruturas de dados;
* *Processes and DLLs*: lista os processos e DLLs carregadas na memória;
* *Process Memory*: recupera informações específicas de um ou mais processos;
* *Kernel Memory and Objects*: lista e verifica componentes do *kernel*;
* *Networking*: recupera atividades de rede do *dump* da memória;
* *Registry*: recupera dados armazenados nos registros do sistema operacional;
* *Crash Dump, Hibernation e Conversion*: executa o *parser* e analisa informações de arquivos de hibernação e *crash dumps*, bem como realiza a conversão entre esse tipos de arquivos; e
* *Miscellaneous*: agrupa os *plugins* de tipos diversos.

V. Metodologia Maldetect

A metodologia proposta é uma adaptação da metodologia do SANS descrita na seção III. Esta metodologia é independente de sistema operacional, ou seja, os conceitos podem ser aplicados a qualquer S.O. A Maldetect coleta e correlaciona informações comportamentais dos artefatos residentes no *dump* de memória volátil e identifica quais dessas características são típicas de códigos maliciosos. A figura 4 apresenta um resumo de cada fase e a seguir estas serão descritas em detalhes.

A. ***Pré-análise***

A pré-análise é uma fase de preparação e otimização da metodologia. Possui como objetivo reaproveitar o conhecimento aprendido das execuções anteriores da metodologia. Nesta fase, o *dump* de memória a ser analisado é comparado com a base de conhecimento de indicativos de comprometimentos[6] (IOC) de análises já realizadas, caso não exista uma base de conhecimento prévia, esta etapa poderá ser suprimida. Além disso, nesta etapa pode ser criada uma linha base com dados de atividades consideradas normais para máquina que será analisada.

B. ***Análise de processos***

Deve-se possuir uma lista dos processos do núcleo do sistema operacional a ser analisado, assim como as atividades normais que estes processos possuem. As características desses processos devem ser confrontadas com os processos correspondentes do *dump* da memória em análise de forma a identificar as possíveis anomalias e armazená-las para serem correlacionadas com os resultados das outras fases. Para realizar esta atividade devem ser coletados os seguintes dados de todos os processos em execução: nome, caminho completo, PID, PPID, linha de comando de inicialização, hora de inicialização, hora de término, processos filhos, prioridade de execução, dono do processo, sessão em que está rodando, número de *threads* em execução e número de *handles*. Ainda nesta fase, devem ser usadas várias técnicas de listagem de processos em execução, com o objetivos de encontrar aqueles que usam técnicas de ocultação, ou seja, processos que apesar de estarem em execução não seriam listados no gerenciador de tarefas do sistema operacional. Por fim, deve-se verificar se os binários estão sendo executados a partir de pastas temporárias e se existem processos com nomes similares ao processos do núcleo do sistema operacional.

[6]Descrição das atividades maliciosas que caracterizam determinado *malware*

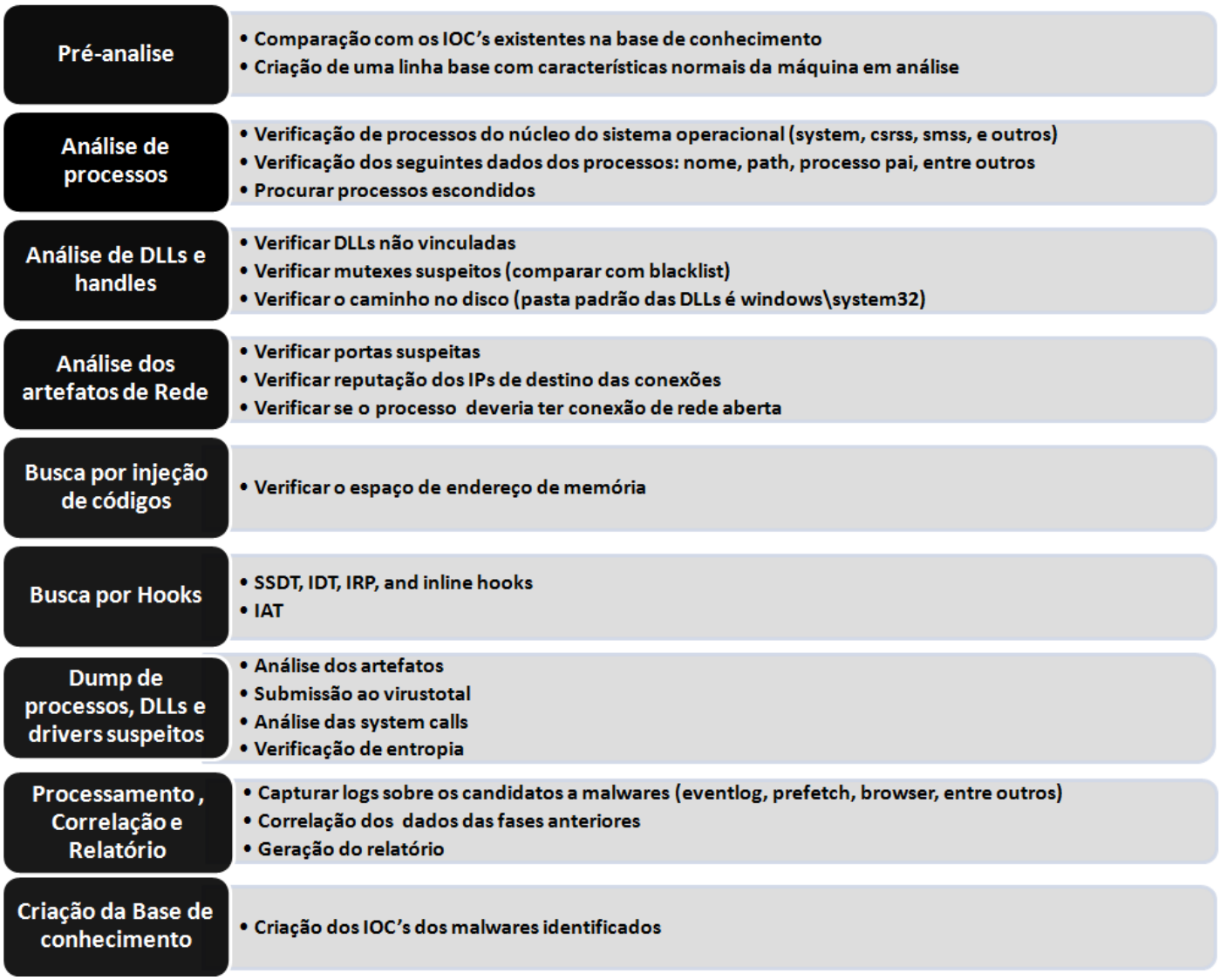


Fig. 4. Metodologia Maldetect

C. *Análise de DLLs e handles*

Durante a análise de DLLs deve-se verificar a existência de DLLs não vinculadas, o contexto onde as mesmas foram carregadas, o caminho do disco onde as DLLs estão armazenadas e a escrita correta de seus nomes. Além disso, os *handles* do tipo *mutexes* devem ser comparados com uma *blacklist* de nomes usados por *malwares* conhecidos. Geralmente este é um recurso muito utilizado pelos códigos maliciosos para identificar se a máquina já foi infectada. Também, verifique se existe um *pipe* (mecanismo que redireciona a saída de uma programa como entrada de outro) redirecionando entradas e saídas do processo "cmd.exe" para um programa remoto, pois esta técnica é muito usada por *backdoors*.

D. *Análise de Artefatos de Rede*

O acesso a rede é muito utilizado pelos *malwares* para diferentes fins, tais como: extravio de informações, *download* de novos componentes, comunicação com a central de comandos, infecção de novas vítimas, disparo de envio de e-mails em massa e ataques de negação de serviços. Dessa forma, deve-se listar as portas abertas em modo *listening* do protocolo TCP e verificar quais portas são consideradas como suspeitas. As informações coletadas na pré-análise pode ajudar a identificação destas portas suspeitas, pois caso a máquina em análise seja um servidor FTP, a porta 21 estará marcada como normal, porém caso estas informações não sejam conhecidas previamente deve-se considerar as portas que não pertencem ao funcionamento normal do sistema operacional como anomalias a serem correlacionadas na fase apropriada. Além disso, deve-se realizar uma verificação do IP's de destino de conexão com uma *blacklist* de IP's maliciosos. E por fim, verifique se existem interfaces de rede em modo promíscuo, quais processos estão fazendo uso das conexões de rede e se estes deveriam fazer uso deste recurso.

E. *Busca por injeção de código*

As técnicas usadas nesta etapa deve ser capaz de identificar as assinaturas de injeção de código, tais como a injeção

de DLLs, verificando áreas de memória marcadas como READ/WRITE/EXECUTE. Além disso, deve ser capaz de verificar se existem *Hollow Process*, esta técnica está descrita na seção III.

F. *Busca por hooks*

O objetivo dessa etapa é identificar os artefatos que usam técnicas avançadas para dificultar sua detecção. As técnicas usadas nesta fase devem ser capazes de identificar os principais tipos de *hooks* usados pelo *malwares*. Deve-se identificar a existência de módulos não vinculados, *hooks* de *system calls* (SSDT), *inline hooks*, alterações na *Interrupt Descriptor Table* (IDT), nos *handles* da *I/O Request Packets* (IRP). Além disso, devem ser identificados os *hooks* da *Import Address Table* (IAT). Também, identifique quais processos estão executando em modo *debug* e a existência de *threads* órfãs. Essa técnicas foram descritas na subseção III-A.

G. *Dump de processos, DLLs e drivers suspeitos*

Essa etapa da metodologia tem o objetivo de aprofundar a análise dos possíveis códigos maliciosos identificados nas outras fases. Estes artefatos serão reconstruídos a partir do *dump* de memória que está sendo analisado. Os artefatos podem ser processos, *drivers* ou DLLs. O *hash* destes arquivos devem ser comparadas com uma base de *hash* de *malwares* conhecidos, como o Virustotal.

Deve-se procurar strings suspeitas, tais como url, email, CPF e nomes de arquivos de sistema. Além de identificar chamadas de sistemas comuns entre os *malwares* e a entropia dos arquivos.

H. *Processamento, Correlação e Relatório*

Os dados coletados e armazenados nas fases anteriores são correlacionados nesta fase e o relatório da análise é gerado. Para complementar as informações podem ser usadas diferentes fontes de *logs* sobre os possíveis *malwares*. Essas fontes podem ser histórico de navegadores, *eventlog*, *prefetch*, anomalias de *timeline* e data de compilação do arquivo executável.

I. *Criação da base de conhecimento*

Na última fase, deve-se utilizar padrões de descrição de indicativos de comprometimentos (IOC) para alimentar a base de conhecimento. Como exemplo, podem ser gerados IOC's baseado no *framework* OpenIOC (*framework open source* capaz de descrever as características comportamentais dos *malwares*[13]). No caso da Maldetect algumas verificações não estão descritas nestes padrões abertos e por isso será usado um padrão próprio para descrição dos IOC's encontrados.

VI. Técnicas de Automatização da Metodologia Maldetect para Windows

Foi construída uma ferramenta que implementa cada fase da metodologia Maldetect. A figura 5 mostra a tela inicial desta ferramenta, denominada de Maldetect Tool, a qual foi

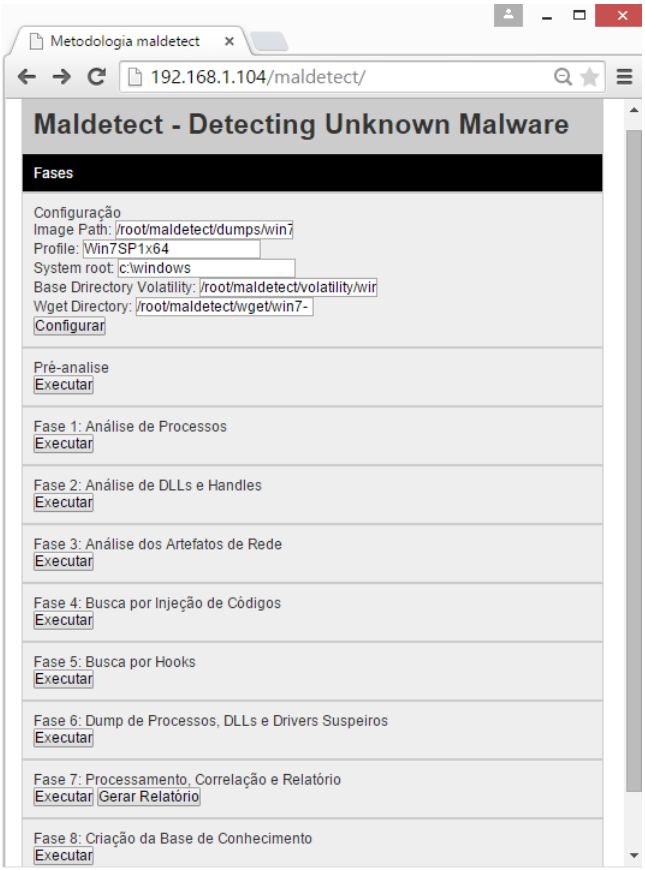


Fig. 5. Tela inicial da versão beta da *Maldetect Tool*.

implementa usando a linguagem PHP e Python. A figura 6 mostra a interação da Maldetect Tool com os recursos externos utilizados. Os principais recursos são: Volatility, VirusTotal, IPVoid[7] e a base de conhecimento. Foi criado um repositório no github (https://github.com/maldetect/maldetect) para disponibilizar os resultados das análises realizadas com a Maldetect Tool. A seguir serão descritas as técnicas utilizadas na construção da Maldetect Tool para o caso da análise de memória volátil do sistema operacional Windows 7.

A. *Pré-análise*

Esta fase recebe como entrada os IOC's gerados por execuções anteriores da metodologia maldetect. Estes arquivos serão criados no formato XML na última fase da metodologia. Dessa forma, é possível otimizar a detecção de códigos maliciosos existente na base de conhecimento. Além disso, deve ser coletada informações que auxiliarão a execução da metodologia, como por exemplo, portas de redes que devem ser consideradas como normais.

B. *Análise de processos*

Foi construído um *plugin*, denominado de procinfo, em Python para o Volatility para extrair as seguintes informações

[7]IPVoid é um serviço *online* e gratuito que faz análise de IP e DNS baseado em *blacklist*. Acessado em http://www.ipvoid.com/

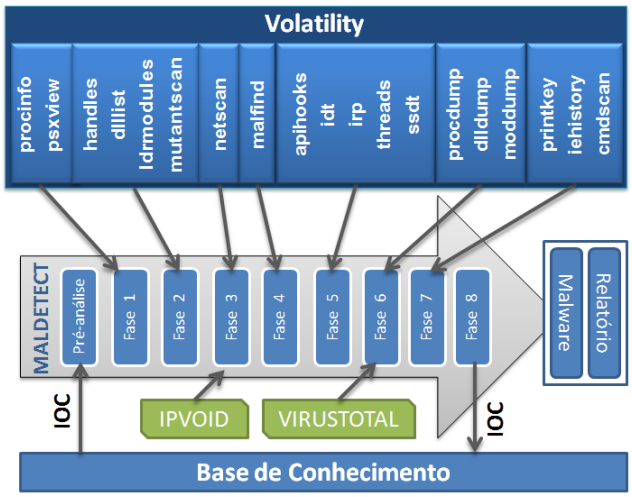


Fig. 6. Arquitetura da Maldetect Tool para o caso de análise de memória volátil do sistema operacional Windows 7. O *plugin* procinfo não faz parte da lista de *plugins* do Volatility e foi criado especialmente para a automatização da metodologia Maldetect.

do *dump* de memória a ser analisado: PID, PPID, PROCESS_NAME, BASEPRIORITY, PATH, COMMAND_LINE, SESSIONID, CREATE_TIME, EXIT_TIME, HANDLES, THREADS e USERNAME. A criação deste *plugin* foi necessária, pois os que já existem não mostram todas as informações consideradas relevantes para as verificações que devem ser realizadas nesta fase. O *process_analyser* realiza um *parser* das informações extraídas pelo procinfo e as carrega em um banco de dados MySQL. Nesta fase, os valores desses atributos são verificados se estão de acordo com a documentação da Microsoft para os seguintes processos do sistema operacional: System, Smss, Crss, Wininit, Services, Lsass, Svchost, Lsm, Winlogon, Explorer, Conhost, Rundll32, Taskhost e IExplore. Para cada um desse processos foi feita um pesquisa no banco de dados por processos com nomes com grau de similaridade maior que 80% com relação aos nomes dos processos do sistema operacional.

Para detectar processos ocultos (*hidden process*) foi utilizado o *plugin psxview* do Volatility, pois este acessa a lista de processos da estrutura _EPROCESS de maneiras diferentes tendo a capacidade de detectar um processo não vinculado.

C. Análise de DLLs e handles

Para verificar os *handles* do tipo *mutex*, muito usado pelos *malwares*, foi utilizado o *plugin mutantscan* do Volatility. Durante as análises pode ser criada uma *blacklist* de *mutex* maliciosos. Em [17], no ano 2014, foi apresentado que o *mutex* "2gvwnqjz1" é muito usado pelos *malwares*, pois em todos os casos onde foi encontrado estava associado a um código malicioso. A *blacklist* utilizada nesta implementação está disponível no repositório do github.

Para verificar quais DLLs não estão vinculadas foi utilizado o *plugin ldrmodules* do Volatility, o qual percorre a lista de DLL de formas diferentes. Foi usado o *plugin dlllist* do Volatility com o objetivo de listar as DLLs carregadas de cada processo e seu respectivo caminho no disco. Assim, foi possível realizar uma pesquisa no banco de dados para descobrir quais processos carregavam DLLs de conexão de rede (winsock32.dll, ws2_32.dll, wininet.dll e urlmon.dll), e com esse resultado é possível verificar e marcar quais desse processos tipicamente não fazem uso de conexões de rede, ou seja, averiguar possível discrepância entre as funcionalidades das DLLs e do processo que as carregou.

Nesta fase, foi feita a busca por características típicas de *backdoor*. Através do *plugin handles* do Volatility foi possível verificar se os processos cmd.exe possui um *handle* do tipo *File* com valor \Device\Afd\Endpoint, pois é um comportamento comum dos *Backdoors*[12].

D. Análise de Artefatos de Rede

Foi utilizado o *plugin netscan* do Volatility para listar as conexões de rede. A análise desta fase foi separada em duas etapas (portas em modo *Listening* e conexões estabelecidas). Para as portas em modo *Listening* foi feito um mapeamento das portas TCP que normalmente são abertas por uma máquina Windows 7 livre de *malwares*. Dessa forma, ao analisar o *dump* de memória em questão as portas que não estão nesta *whitelist* são marcadas como suspeitas. Nas portas com conexões estabelecidas, os IP's remotos foram submetidos ao IPVoid para verificação de sua reputação.

E. Busca por injeção de código

O *plugin malfind* do Volatility foi utilizado para mapear quais áreas de memória estão marcadas como PAGE_EXECUTE_READWRITE, pois essas são as possíveis áreas de injeção de código. Assim os processos associados a essas áreas são marcados como suspeitos.

F. Busca por hooks

Para cada técnica de *hook* utilizada pelos *rootkits* foi usado o *plugin* correspondente do Volatility. Para os hooks de IAT, EAT e *inline hook*, foi usado o *plugin apihooks*, e os outros os nomes dos *plugins* do Volatility correspondem a técnica de *hook* que eles detectam. Portanto foram usados os seguintes *plugins* do Volatility: *ssdt*, *irp*, *idt*. Além disso, foi feita a verificação de *threads* órfãs usando o *plugin threads* do Volatility.

G. Dump de processos, DLLs e drivers suspeitos

Nesta fase, são listados os processos em ordem decrescente do número de anomalias encontradas nas fases anteriores e o analista pode escolher os artefatos que serão extraídos da memória. Para realizar a extração destes artefatos foi utilizado os seguintes *plugins* do Volatility: *procdump*, *ddldump* e *moddump*. Após a extração, usando a API do VirusTotal é feita uma consulta pelo *hash* sha256 do artefato extraído do *dump* da memória.

H. *Processamento, Correlação e Relatório*

Todas as informações obtidas nas fases anteriores são processadas e correlacionadas de tal forma que os possíveis códigos maliciosos recebam uma pontuação para cada anomalia encontrada, quanto maior sua pontuação mais anomalias o processo possui. Para cada antivírus que retorna como *malware* a Maldetect Tool eleva a pontuação de anomalia do artefato. A descrição da cada anomalia é armazenada na base de dados para fins de relatório. Além disso, para compor o relatório final são coletadas as seguintes informações:

* valores da chave de registro Microsoft\Windows\CurrentVersion\Run (usada para armazenar programas que serão executados junto com a inicialização do windows);
* histórico de acesso do Internet Explorer;e
* histórico de comandos do cmd.exe.

Para capturar essas informações foram utilizados respectivamente os seguintes *plugins* do Volatility: *printkey*, *iehistory* e *cmdscan*. A Maldetect Tool gera um relatório em PDF contendo todos os artefatos que receberam uma pontuação de anomalias, assim como as anomalias de rede, o histórico de comandos do cmd.exe e acessos do Internet Explorer.

I. *Criação da base de conhecimento*

Na última fase da metodologia, o analista pode escolher quais processos ele gostaria de gerar o arquivo de IOC para compor a base de conhecimento. Este arquivo possui o formato XML onde são descritas as anomalias encontradas durante as fases anteriores da metodologia. Este arquivo irá alimentar a fase de pré-análise da próxima execução.

VII. Resultados e Discussões

Como prova de conceito da metodologia Maldetect e validação da implementação realizada pela ferramenta Maldetect Tool, foi feita uma análise automática de quatro *dumps* de memória de uma máquina Windows 7 infectada com os seguintes *malwares*: jackal[8], nfe.xml.exe[9], CiGPxdM.exe[10] e NF-e 18454310845.exe[11]. Foi analisado um *malware* menos conhecido pelos antivírus como é o caso do jackal.exe. Inclusive, antivírus como Kaspersky não o detectaram como sendo um código malicioso. A tabela I mostra a taxa de detecção desses *malwares* no VirusTotal.

TABELA I
Resultado da submissão dos códigos maliciosos ao VirusTotal.

Malwares	Taxa de detecção	Data da Análise
jackal.exe	20 / 57	26/04/2015 15:30:39 UTC
nfe.xml.exe	34 / 57	26/04/2015 15:34:43 UTC
NF-e 18454310845.exe	40 / 57	26/04/2015 15:40:05 UTC
CiGPxdM.exe	46 / 57	26/04/2015 15:37:12 UTC

[8]md5: e0208ab8930434036cbeef5683418d23
[9]md5: f6be0475e183335e00ffe363cf62a2bc
[10]md5: 116addcf779c596ad11a3fe910050c9e
[11]md5: f9856997401fd45a38790dcb1402537e

Foi feito um *dump* da memória da estação infectada com cada um dos códigos maliciosos. Os *dumps* de memória obtidos foram analisados automaticamente utilizando a Maldetect Tool. A ferramenta não sabia previamente nenhuma informação sobre o artefato malicioso que havia nos *dumps* de memória. A tabela II mostra os artefatos considerados maliciosos pela Maldetect Tool e as respectivas atividades típicas de *malwares* encontradas. Além dessas anomalias, o relatório também apresenta os registros do histórico do Internet Explorer, as anomalias de rede e o histórico de comandos dos processos cmd.exe que estavam em execução no momento da aquisição da memória. O relatório completo gerado para cada *malware* está disponível no repositório do Github[12].

TABELA II
Atividades típicas de códigos maliciosos encontrados e seus respectivos artefatos

Malwares	Artefato	Atividade maliciosa
jackal.exe	jackal.exe.exe	Cria dois processos cmd.exe! Mutex malicioso encontrado: __Dassara__! Porta ou conexão suspeita! Backdoor! Taxa de Detecção do Virustotal: 12 / 57! Carrega duas DLLs num contexto suspeito!
nfe.xml.exe	MALDETECT-PC.e	Este processo esta sendo executado a partir da pasta appData! Carrega uma DLL num contexto suspeito! Possui área de memória com a flag de write_exec!
CiGPxdM.exe	svchost.exe	Pai não encontrado! Caminho incorreto! Username incorreto! Falta parâmetro -k! Possui área de memória com a flag de write_exec!
NF-e 18454310845.exe	svchost.exe	Pai não encontrado! Caminho incorreto! Username incorreto! Falta parâmetro -k! Possui área de memória com a flag de write_exec!

Percebeu-se que os *malwares* tentaram dificultar sua detecção usando nomes de processos correspondentes à nomes de processos legítimos do sistema operacional, no caso foi usado o nome svchost.exe (nome de processo que pertence ao núcleo do sistema operacional Windows 7). O jackal foi executado a partir da pasta c:\windows\system32 na tentativa de se camuflar como um processo legítimo do Windows 7. Todos códigos maliciosos testados foram detectados e a automatização proposta, implementada pela Maldetect Tool, reduziu consideravelmente o tempo de análise da memória volátil em relação a análise manual. Por fim, o relatório gerado pela Maldetect Tool apresenta todas as informações relevantes coletadas durante a análise e direciona a atenção do analista para os artefatos que realmente realizam atividades

[12]https://github.com/maldetect/maldetect

típicas de *malware* deixando claro qual foi o artefato malicioso encontrado.

A Maldetect Tool ainda está em desenvolvimento, e está na versão *beta* para testes e melhorias das técnicas de detecção de anomalias comportamentais típicas de códigos maliciosos.

VIII. Conclusao

A metodologia *Maldetect* coleta e correlaciona informações comportamentais dos processos, DLLs e *drivers* de um *dump* de memória volátil (RAM) e identifica quais desses comportamentos são típicos de códigos maliciosos. A metodologia pode ser aplicada para detectar as ameças avançadas modernas e *malwares* desconhecidos. Além disso, é uma metodologia automatizável e independente de sistema operacional.

A metodologia *Maldetect* demonstra que a verificação de características comportamentais é mais eficaz que a detecção baseada em assinatura usada pela maioria dos antivírus. Utilizando a *Maldetect Tool*, que implementa a metodologia proposta, foi possível detectar os artefatos maliciosos baseado em características comportamentais. Além disso, a *Maldetect Tool* detectou *malwares* com baixa taxa de detecção no VirusTotal. Apesar do aumento da complexidade e do avanço das técnicas usados pelos *malwares* modernos, coletar e correlacionar informações comportamentais de várias fontes é uma das maneiras eficientes de detectá-los.

Em trabalhos futuros, sugere-se a ampliação da base de conhecimento dos indicativos de comprometimentos dos vários tipos de códigos malicioso com o objetivos de aplicar técnicas de aprendizado de máquina para verificar se ocorre agrupamento entre as características dos tipos de artefatos maliciosos e assim determinar qual o melhor tipo classificador.

Agradecimentos

Agradeço primeiramente a Deus, que me sustentou, capacitou e me deu saúde para realizar mais um projeto em minha vida. Aos professores, coordenadores, e funcionários do Departamento de Engenharia Elétrica da Universidade de Brasília que nos proporcionaram o ambiente saudável para pesquisa e desenvolvimento. Também ao orientador Dino, que sempre esteve disponível e paciente para me auxiliar nesta jornada. E por último, mas não menos importante, a minha esposa e família que me apoiaram e incentivaram nessa caminhada.

Referências

[1] Anand Ajjan. Ransomware: Next-generation fake antivirus. A SophosLabs technical paper, 2013. http://www.sophos.com/en-us/medialibrary/PDFs/technical%20papers/SophosRansomwareFakeAntivirus.pdf?la=en.pdf?dl=true.

[2] Michael Bailey, Jon Oberheide, Jon Andersen, Z Morley Mao, Farnam Jahanian, and Jose Nazario. Automated classification and analysis of internet malware. In *Recent advances in intrusion detection*, pages 178–197. Springer, 2007.

[3] Bill Blunden. *The Rootkit Arsenal: Escape and Evasion in the Dark Corners of the System*. Jones & Bartlett Publishers, 2011.

[4] DFRWS. The dfrws 2005 forensic challenge, 2005. http://www.dfrws.org/2005/challenge/index.shtml.

[5] Fahad Eshan. Memory forensics & security analytics: Detecting unknown malware, 2014. http://www.isaca.org/chapters5/Ireland/Documents/2014%20Event%20Presentations/Detecting%20Unknown%20Malware%20Memory%20Forensics%20and%20Security%20Analytics%20-%20Fahad%20Ehsan.pdf.

[6] Volatility Foundation. Command reference, 2015. https://github.com/volatilityfoundation/volatility/wiki/Command-Reference.

[7] Liang Hu, Shinan Song, Xiaolu Zhang, Zhenzhen Xie, Xiangyu Meng, and Kuo Zhao. Analyzing malware based on volatile memory. *Journal of Networks*, 8(11):2512–2519, 2013.

[8] Randall Hyde. *The art of assembly language*. No Starch Press, 2010.

[9] Rob Lee. Finding unknown malware – step-by-step. SANS DFIR Faculty, 2013. http://digital-forensics.sans.org/media/poster_fall_2013_forensics_final.pdf.

[10] Frankie Li. A detailed analysis of an advanced persistent threat malware. *SANS Institute InfoSec Reading Room*, 2011.

[11] Michael Ligh, Steven Adair, Blake Hartstein, and Matthew Richard. *Malware analyst's cookbook and DVD: tools and techniques for fighting malicious code*. Wiley Publishing, 2010.

[12] Michael Hale Ligh, Andrew Case, Jamie Levy, and Aaron Walters. *The Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory*. John Wiley & Sons, 2014.

[13] Hun-Ya Lock. Using ioc (indicators of compromise) in malware forensics, 2013. http://www.sans.org/reading-room/whitepapers/forensics/ioc-indicators-compromise-malware-forensics-34200.

[14] Cameron H Malin, Eoghan Casey, and James M Aquilina. *Malware forensics: investigating and analyzing malicious code*. Syngress, 2008.

[15] Mihály Oroszlány. Rootkits under windows os and methods of their detection. Masaryk University Faculty of Informatics, 2008. http://is.muni.cz/th/139801/fi_b/Bc.pdf.

[16] Mark Russinovich, David A. Solomon, and Alex Ionescu. *Windows Internals, Part 1. 6th Edition*. Microsoft Press, 2012.

[17] Rob Seger. Hunting the mutex, 2014. http://researchcenter.paloaltonetworks.com/2014/08/hunting-mutex/.

[18] Johannes Stüttgen and Michael Cohen. Anti-forensic resilient memory acquisition. *Digital Investigation*, 10:S105–S115, 2013.

[19] Chad Tilbury. Memory forensics. SANS Computer Forensics and Incident Response, 2012. http://software.msu.montana.edu/free/ISSA/Memory%20Forensics%20Made%20Easy%20Solving%20Cases%20with%20the%20New%20Breed%20of%20Tools-Tilbury-5-22-2012.pdf.

[20] Stefan Vömel and Felix C Freiling. A survey of main memory acquisition and analysis techniques for the windows operating system. *Digital Investigation*, 8(1):3–22, 2011.

[21] Guangmingzi Yang, Zhihong Tian, and Wenliang Duan. The prevent of advanced persistent threat. *Journal of Chemical and Pharmaceutical Research*, 6(7):572–576, 2014.

[22] Boyun Zhang, Jianping Yin, and Jingbo Hao. Using fuzzy pattern recognition to detect unknown malicious executables code. In *Fuzzy Systems and Knowledge Discovery*, pages 629–634. Springer, 2005.

DOI: 10.5769/C2015009 or http://dx.doi.org/10.5769/C2015009

Desenvolvimento de um Ambiente Honeynet Virtual para Aplicação Governamental

Gildásio Antonio de Oliveira Júnior, Rafael Timóteo de Sousa Júnior e Danilo Fernandes Tenório

Departamento de Engenharia Elétrica, Universidade de Brasília (UnB)
Campus Universitário Darcy Ribeiro – Asa Norte – 70910-900 – Brasília-DF – Brasil

jrgildasio@oi.com.br, desousa@unb.br, daniloftenorio@gmail.com

***Resumo*—Constitui uma prática comum aplicar técnicas de detecção de intrusos para detectar tráfego malicioso. Por conta do extenso número de vulnerabilidades em sistemas de informação e da grande criatividade dos invasores, torna-se cada vez mais necessário atualizar permanentemente as técnicas de detecção utilizadas. Portanto, é crucial operacionalizar um ambiente cibernético que propositadamente esteja preparado para ser invadido e comprometido, com a finalidade de permitir ao profissional de segurança analisar e verificar a evolução dos diversos tipos de ataques e vulnerabilidades exploradas por invasores. Este trabalho apresenta uma solução de segurança projetada especificamente para a pesquisa e a obtenção de informações de intrusos. Esse mesmo ambiente pode ser utilizado para a preservação de evidências de ataques para efeito forense.**

Palavras-Chave*—*Honeynets, Honeypots, Intrusion Detection Systems (IDS), Controle de Dados de Intrusões, Captura de Dados de Intrusões.

***Abstract*—It constitutes a common practice to apply intrusion detection techniques to detect malicious traffic. Because of the large number of vulnerabilities in information systems and the great creativity of the intruders, it becomes increasingly necessary to continuously update the employed detection techniques. Therefore, it is crucial to operationalize a cyber-environment that purposely is prepared to be invaded and compromised, in order to allow the security professional to analyze and verify the evolution of the several types of attacks as well as the vulnerabilities exploited by attackers. This paper presents a security solution projected specifically for research and for obtaining information from intruders. This same cyber-environment can be used for the preservation of attack evidences to forensic effect.**

Keywords*—*Honeynets, Honeypots, Intrusion Detection Systems (IDS), Intrusion Data Control, Intrusion Data Capture.

I. Introdução

Atualmente um dos principais problemas de segurança enfrentados no ciberespaço é a invasão de redes de computadores. Conforme estatísticas apresentadas em [1], no ano de 2014 houve 467.621 notificações de tentativas de fraudes e um aumento de 54% de ataques em servidores *Web*, em relação ao ano anterior. A rápida expansão do volume de informações acessadas através da Internet aumentou o interesse por novas formas de atividades intrusivas. Por conta desse crescimento, o ciberespaço se tornou um campo de guerra cibernética, uma guerra invisível e interminável. Desta forma, torna-se fundamental proteger os ativos de rede contra ameaças e garantir a integridade, a confidencialidade e a disponibilidade dos dados trafegados.

Verifica-se, entretanto, que grande parte das ferramentas e técnicas utilizadas para segurança da informação com a finalidade de combater ataques, tais como *firewall*, sistemas de criptografia, *hash*, assinatura digital, *antivírus*, dentre outros, não são suficientes para assegurar a segurança de redes e sistemas.

As tecnologias de detecção de intrusão trabalham em conjunto com outros mecanismos de segurança, buscando sempre indícios da ocorrência de ataques. Segundo [2], detecção de intrusão é o processo de monitoramento de eventos que ocorre em um sistema de computador ou rede para detectar sinais de possíveis incidentes. Tais tecnologias são classificadas de acordo com as técnicas utilizadas para investigar os citados indícios. Por exemplo, a arquitetura de um IDS (*Intrusion Detection System*) depende da localização do sistema e da forma como os dados são coletados, havendo duas categorias: baseado em host ou HIDS (*Host Intrusion Detection System*) e baseado em rede ou NIDS (*Network Intrusion Detection System*).

O grande problema da operacionalização de um IDS é que tal sistema limita-se a garantir certo nível de segurança, sobretudo no que se refere à detecção de ataques que sejam conhecidos previamente ou que se possa prever de alguma forma, ou seja, ataques que tenham alguma assinatura verificável previamente conhecida. Coloca-se por consequência a questão complementar de como coletar e analisar os dados de tráfego que incluam possíveis ataques ainda desconhecidos. Para tanto, prepara-se o IDS para detectar não somente ataques por assinatura, mas também indícios de ataques que correspondam a anomalias no tráfego ou na operação de sistemas. A detecção por anomalia, ainda que efetiva como método para bloquear os ataques, deixa em aberto a necessidade de analisar detalhadamente a anomalia para descrever o ataque e deste obter a respectiva assinatura.

A solução comum para a questão da análise dos ataques, tanto daqueles conhecidos quanto daqueles detectados por anomalia, consiste em construir um recurso de segurança chamado *honeynet* que, de acordo com [3], é uma ferramenta projetada especificamente para pesquisa e obtenção de informações de intrusos. Em uma *honeynet*, sistemas operacionais e serviços são emulados, de modo a parecer com sistemas reais funcionais, atraindo assim a atenção dos atacantes. Esta rede particular, deixada propositadamente ao

alcance dos intrusos, pode então ser usada para verificar e analisar os diversos tipos de ataques e as correspondentes vulnerabilidades exploradas nesses ataques.

Na gerência da segurança das redes, sistemas e aplicações governamentais, considera-se que o gestor deve aplicar toda medida a seu alcance, no sentido de proteger a informação e os sistemas, sendo então as medidas voltadas à captura e análise de novos ataques um importante pilar da gestão da segurança.

Outra característica de relevo das *honeynets* é a sua possível aplicação na área forense, visto que uma *honeynet* tem a possibilidade de capturar o *modus operandi* das operações de intrusão, bem como preservar evidências dos atos que indiquem ocorrência de crimes e também dos meios tecnológicos empregados em tais crimes.

Tais possibilidades das *honeynets* constituem uma justificativa do presente trabalho, pois se considera a importância de ter um ambiente cibernético dentro de um Órgão Governamental para acompanhar a evolução dos diversos tipos de ataques, a fim de impedir intrusões maliciosas nos sistemas, garantindo, desta forma, o funcionamento dos serviços essenciais à população. Por outro lado, foram considerados os temas de pesquisa correlatos no domínio da informática forense e segurança da informação. Diante disso, a estratégia do trabalho foi dividida em três fases distintas: construção de um ambiente *honeynet* virtual de autocontenção, validação do ambiente e um estudo de caso.

Este artigo está organizado da seguinte forma. Na Seção 2 apresentamos os conceitos relacionados a *honeynets*. É detalhada na Seção 3 a arquitetura do ambiente de *honeynet* virtual de autocontenção proposto propositadamente para ser invadido e comprometido. Na Seção 4 validamos o ambiente, à luz dos requisitos de controle, captura e análise dos dados. Na Seção 5 simulamos dois ataques de força bruta, para demonstrar de forma detalhada o funcionamento do ambiente *honeynet*. Por fim, a Seção 6 apresenta as considerações finais e propostas de trabalhos futuros.

II. HONEYNETS

De acordo com [3], usando o conceito de *honeypot* como sendo um sistema, serviço ou aplicação emulada propositadamente para tornar-se alvo de um ataque, denomina-se *honeynet* um conjunto de *honeypots* de alta interatividade, integrados em uma solução projetada especificamente para ser invadida e comprometida. Diferentemente dos *honeypots* de baixa interatividade, que apenas emulam sistemas operacionais e serviços, os *honeypots* de alta interatividade fornecem sistemas operacionais e aplicações reais com as quais os intrusos possam interagir. Essa interatividade faz com que pesquisadores possam observar o comportamento de um intruso em um sistema real, a fim de descobrir novas técnicas de invasão, identificar novas vulnerabilidades e aprender como esses intrusos se comunicam.

Referências sobre mecanismos para monitoração e análise de atividades intrusivas surgiram em meados da década de 1980. Em agosto de 1986, um usuário malicioso atacou computadores dos laboratórios do LBL (Lawrence Berkeley Laboratory) para roubar dados. Com a finalidade de monitorar esses tipos de usuários, Clifford Stoll criou um projeto governamental para rastrear com detalhes os ataques até sua origem [4].

Em 1990, Bill Cheswick descreveu o acompanhamento de uma invasão no laboratório da AT&T, invasão esta em que foram exploradas falhas no serviço *Sendmail*, obtendo-se acesso ao *gateway* do laboratório. A finalidade desta experiência consistiu em localizar e aprender sobre as técnicas que foram utilizadas pelos intrusos. Uma técnica de mudança de diretório *root*, *chroot*, foi construída para observar todas as atividades que o intruso queria fazer [5].

A primeira solução de *honeypot* baseada em *software* foi chamada de DTK (*Deception Toolkit*) [6]. Ela foi desenvolvida em 1998 por Fred Cohen. Esta ferramenta tinha uma coleção de *scripts Perl* e código C, que emulava várias vulnerabilidades conhecidas do *Unix*, com o propósito de obter informações e enganar atacantes. Este *toolkit* pode ser utilizado também para alertar e aprender sobre vulnerabilidades conhecidas.

Em 1999, Lance Spitzer liderou um grupo, sem fins lucrativos, de 30 profissionais de segurança, dedicados a aprender técnicas, táticas e motivações de intrusos. Em 2001, os membros do projeto lançaram o livro "*Know Your Enemy*", baseado em dois anos de pesquisas e descrevendo em detalhes as tecnologias *Honeynet* [3] e [7]. Ainda em dezembro de 2001, o *Honeynet Project* anunciou a "*Research Alliance Honeynet*" com o objetivo de melhorar as pesquisas e desenvolvimento de *honeynets*. Depois do *Honeynet Project* muitos autores criaram definições e classificações que serão descritas nas próximas seções.

Já em [8] e [9], é apresentada uma comunidade de agentes de *software* que captura ataques e redireciona o tráfego para uma *honeynet*, de modo a permitir a análise dos detalhes dos ataques e a criação de proteções, em solução predecessora à deste trabalho.

A. Arquitetura de uma Honeynet

O sucesso de um projeto *honeynet* depende da correta definição da arquitetura, verificando-se que a construção e a manutenção de uma *honeynet* dependem de três requisitos críticos: controle de dados, captura de dados e coleta de dados [10]. O controle e a captura dos dados são os requisitos mais importantes da arquitetura. O terceiro requisito se aplica nas configurações que tenham vários *honeypots* em ambientes distribuídos.

1) Controle de Dados

Trata-se de um requisito muito crítico, cuja finalidade é a de controlar os dados de entrada e saída para reduzir os riscos dentro da *honeynet*. Isto garante que sistemas comprometidos não sejam usados para atacar sistemas de produção de outras redes [3] e [10]. O tráfego de dados deve ser controlado de modo automático, para reduzir de forma rápida qualquer dano no sistema, e transparente, visando garantir que intrusos não percebam que suas atividades estão sendo controladas.

Ou seja, o controle de dados deve ser utilizado para separar a *honeynet* das outras redes, tais como: *Internet*, administrativa e produção. Para tanto, cada pacote deve ser controlado e inspecionado quando entra ou sai da *honeynet*. Geralmente, os ambientes permitem apenas que qualquer sistema inicie conexões com a *honeynet*, consentindo que intrusos sondem, identifiquem e explorem os sistemas vulneráveis dentro da *honeynet*.

2) Captura de Dados

Tratam-se das operações de captura de dados relativos a todas as atividades dos intrusos dentro da *honeynet*, incluindo as conexões de entrada, as atividades de rede e de sistema. Conforme [3] e [10], tais operações são tão críticas para o sucesso do projeto, que é melhor ter múltiplos métodos de captura de dados operantes.

Entretanto, nenhum dado capturado deve ser armazenado localmente nos *honeypots*, visto que dados armazenados localmente podem ser detectados por intrusos e utilizados para

comprometer o sistema. Além disso, estes dados podem ser modificados e destruídos. Em consequência, tais dados devem ser armazenados em outro local que seja seguro e confiável.

3) *Coleta de Dados*

A coleta de dados é um requisito aplicado em organizações que possuam várias *honeynets* em ambientes distribuídos. Neste caso, todos os dados capturados deverão ser transferidos a uma coletora central, para armazenamento e para poderem ser correlacionados e aumentar a efetividade das *honeynets* de captura.

Conforme [10], se a *honeynet* faz parte de um ambiente distribuído, então quatro requisitos específicos para coleta de dados devem ser aplicados:

- Cada *honeynet* deverá ter um identificador único;
- Os dados deverão ser transmitidos dos sensores para uma coletora de forma segura, garantido sua confidencialidade, integridade e autenticidade;
- O anonimato dos dados deverá ser garantido; e
- Um serviço de sincronização de relógios, como o *Network Time Protocol* (NTP) deverá ser utilizado para garantir que os dados capturados na *honeynet* distribuída estejam devidamente sincronizados.

B. *Honeynets Reais*

As *honeynets* reais fornecem sistemas operacionais reais com quem os intrusos possam interagir. O objetivo dessa interação é aprender como os intrusos invadem os sistemas, como se comunicam e qual a finalidade do ataque [3], [10] e [11]. Estas informações podem ser de extrema importância para que Órgãos Governamentais compreendam e protejam seus sistemas contra ameaças e ataques.

Neste tipo de *honeynet*, todos os dispositivos e mecanismos de segurança (*honeypots*, contenção, alerta e coleta de informações) são físicos [3], [7] e [10]. A Tabela I apresenta as principais vantagens e desvantagens das *honeynets* reais.

TABELA I. VANTAGENS E DESVANTAGENS DAS *HONEYNETS* REAIS

Vantagens	Desvantagens
• Intrusos interagem com dispositivos físicos reais • Ambiente distribuído (tolerante a falhas)	• Custo de implementação e espaço físico • Dificuldade de instalação e administração • Complexidade de manutenção

C. *Honeynets Virtuais*

Por suas características, as *honeynets* reais são difíceis e complexas de construir. Além disso, sua implementação exige uma variedade de sistemas físicos e mecanismos de segurança. Por outro lado, as *honeynets* virtuais permitem executar todos os sistemas operacionais, aplicações e serviços no mesmo *hardware* através de um *software* de virtualização [10] e [11]. A Tabela II apresenta as principais vantagens e desvantagens das *honeynets* virtuais.

TABELA II. VANTAGENS E DESVANTAGENS DAS *HONEYNETS* VIRTUAIS

Vantagens	Desvantagens
• Custo e espaço físico reduzidos • Facilidade de manutenção e administração	• Limitação e risco de comprometimento do software de virtualização (neste caso, o intruso poderá controlar toda a *honeynet*) • Risco de *fingerpriting* (os intrusos poderão detectar se os sistemas estão sendo executados em um software de virtualização)

As *honeynets* virtuais estão divididas ainda em duas categorias: autocontenção e híbridas. Na primeira, todos os dispositivos, incluindo os de captura e coleta de dados, geração de alertas e *honeypots*, estão implementados em um único computador. Já as híbridas representam uma combinação entre *honeynets* reais e virtuais. Nesta categoria, por exemplo, operações de captura, controle de dados e sistemas de *logs* são implementados em dispositivos físicos distintos, enquanto os *honeypots* são configurados em um único computador através de um *software* de virtualização.

III. PROPOSTA DE UM AMBIENTE HONEYNET VIRTUAL DE AUTOCONTENÇÃO

Esta seção descreve os aspectos relacionados ao desenvolvimento de um ambiente *honeynet* virtual de autocontenção para ser invadido e comprometido. A arquitetura proposta neste trabalho tem como objetivo detectar e capturar ataques novos e desconhecidos em Órgãos Governamentais. Neste ambiente, uma parte substancial do tráfego capturado terá origem ilícita ou maliciosa, ou seja, estará comprometida por códigos maliciosos.

Para atingir este objetivo, o desenvolvimento da arquitetura foi dividido em três fases distintas [3], [10]: arquitetura proposta e modelo de solução (fase 1), controle de dados (fase 2) e captura de dados (fase 3). Dessa forma, por um lado é possível atender os requisitos definidos na Seção anterior e, por outro lado, trata-se de uma solução em camadas e verifica-se que, quanto mais camadas de informações o ambiente tiver, mais fácil será analisar e aprender com os intrusos.

A arquitetura do ambiente *honeynet* virtual de autocontenção dispõe de um servidor físico (*Dell PowerEdge 2950*), sete servidores virtuais e um firewall (*CISCO ASA 5520*). A Figura 1 apresenta a estrutura do ambiente utilizado.

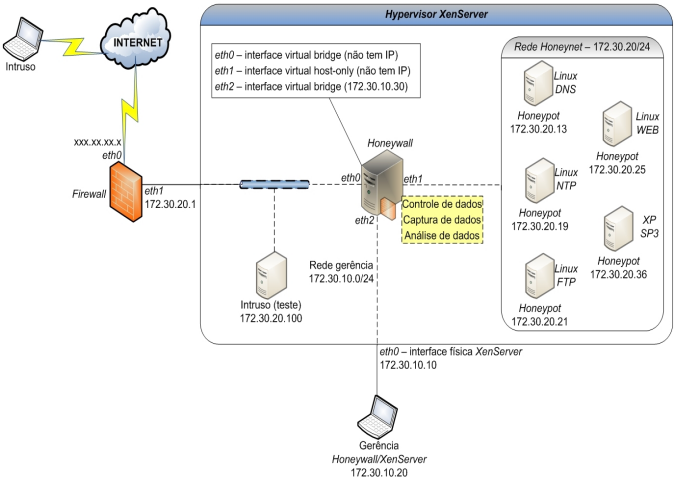


Fig. 1. Arquitetura proposta e modelo de solução.

A. *Fase 1: Arquitetura Proposta*

Nesta fase, a arquitetura foi desenvolvida com o *Hypervisor XenServer*, uma solução de virtualização de código aberto que possibilita gerenciar infraestruturas virtuais [12] [13]. Toda a estrutura do projeto foi feita em apenas um servidor físico. Recomenda-se a utilização de *hardware* dedicado para que as máquinas virtuais possam ser executadas corretamente. A Tabela III apresenta as características do servidor e plataforma de virtualização utilizada.

TABELA III. CARACTERÍSTICAS DO SERVIDOR

Servidor Dell PowerEdge 2950	Configuração
Hypervisor XenServer	Processador Intel *Xeon* E5410 2.33 GHz 8 núcleos com tecnologia Intel VT , 8 GB de memória RAM, 2 HDs de 250 GB, 2 HDs de 500 GB e 2 placas de rede 10/100/1000
	Plataforma *XenServer* 6.2 com as *features* XS62ESP1, XS62E001, XS62E002, XS62E004, XS62E005, XS62E007, XS62E008, XS62E009, XS62E010, XS62E011, XS62E012, XS62E013

O servidor foi configurado com RAID 10, para garantir desempenho e redundância dos dados. Portanto, foram utilizados quatro HDs (2 HDs de 250 GB e 2 HDs de 500 GB) para realizar esta configuração. Isso permitiu utilizar o próprio servidor como *storage* para armazenar as máquinas virtuais.

O *Hypervisor XenServer* foi implementado no servidor físico para criar a estrutura da *honeynet* virtual de autocontenção. A configuração e gerência das máquinas virtuais no *XenServer* foi feita através do *XenCenter 6.2*, por ser uma ferramenta de código aberto sob licença BSD (*Berkeley Software Distribution*). A Figura 2 mostra os *honeypots* virtuais que foram criados pelo *XenCenter*. Vale ressaltar que todos os sistemas foram testados e funcionaram com sucesso dentro do *XenServer*. As máquinas virtuais foram configuradas de acordo com a Tabela IV.

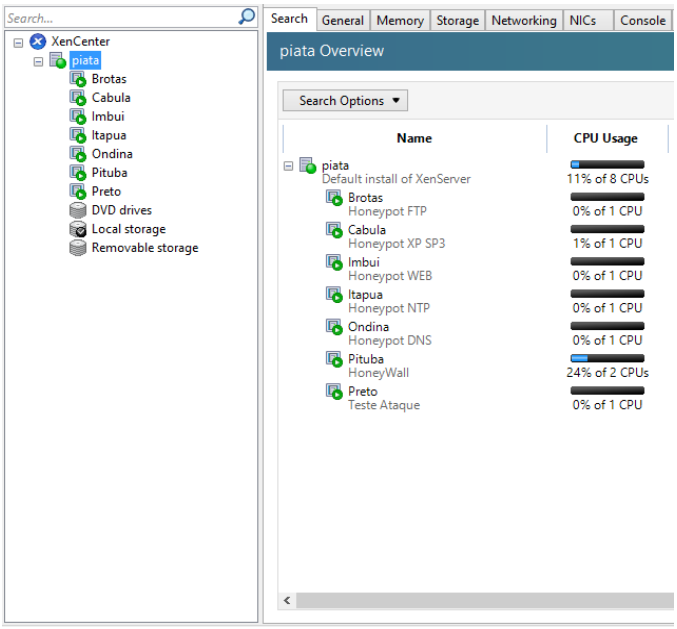


Fig. 2. Gerência do *XenServer* com *XenCenter*.

Conforme a Figura 1, o ambiente possui três redes distintas: a internet, uma rede não confiável (lugar de onde vêm os ataques); a rede *honeynet* 172.30.20.0/24, integrada por um conjunto de *honeypots* para serem comprometidos; e a rede de gerência 172.30.10.0/24, para gerência do *honeywall e XenServer*.

O *honeywall* [14], [15] foi configurado com três interfaces virtuais. A primeira interface virtual *eth0* comunica-se com o *firewall* (IP 172.30.20.1). A segunda interface virtual *eth1* é utilizada para se comunicar com a rede *honeynet*. As duas interfaces virtuais (*eth0* e *eth1*) estão configuradas como *bridge* (camada 2), portanto não possuem endereço IP. Por fim, a terceira interface *eth2* (IP 172.30.10.30) é utilizada para gerência e coleta de dados do *honeywall*. Vale ressaltar que o funcionamento deste dispositivo na camada 2 apresenta duas grandes vantagens: a primeira é que não há *hops* de roteamento nem decremento do TTL (*Time To Live)* no cabeçalho IP; a segunda vantagem é a dificuldade por parte dos intrusos em detectar o ambiente.

TABELA IV. MÁQUINAS VIRTUAIS E SUAS CONFIGURAÇÕES

Máquinas Virtuais	Configuração
Honeywall	Processador com 2 núcleos, 2 GB de memória RAM, 200 GB de HD e 3 interfaces virtuais de rede (*eth0, eth1 e eth2*)
	Versão *Roo-1.4* baseado no Sistema Operacional *CentOS release* 5 (final) com os serviços *snort, iptables* e *sebek-3.0.3-6*
Intruso (teste)	Processador com 1 núcleo, 512 MB de memória RAM, 8 GB de HD e uma interface virtual de rede (*eth0*)
	Sistema Operacional *Linux Kali 1.0.9* com ferramentas para testes de penetração e forense digital
Honeypot DNS	Processador com 1 núcleo, 512 MB de memória RAM, 8 GB de HD e uma interface virtual de rede (*eth0*)
	Sistema Operacional *Linux Debian Wheezy 7.2* com os serviços *bind9, ntpdate e OpenSSH-6.0-p1*
Honeypot NTP	Processador com 1 núcleo, 512 MB de memória RAM, 8 GB de HD e uma interface virtual de rede (*eth0*).
	Sistema Operacional *Linux Debian Wheezy 7.2* com os serviços *ntp-4.2.6-p5 e OpenSSH-6.0-p1*
Honeypot FTP	Processador com 1 núcleo, 512 MB de memória RAM, 8 GB de HD e uma interface virtual de rede (*eth0*)
	Sistema Operacional *Linux Debian Wheezy 7.2* com os serviços *proftpd-1.3.4a, ntpdate e OpenSSH-6.0-p1*
Honeypot WEB	Processador com 1 núcleo, 512 MB de memória RAM, 8 GB de HD e uma interface virtual de rede (*eth0*)
	Sistema Operacional *Linux Debian Wheezy 7.2* com os serviços *apache2.2.22, php5-5.4.4-14, mysql-server-5.5.31, ntpdate e OpenSSH-6.0-p1*
Honeypot XP	Processador com 1 núcleo, 512 MB de memória RAM, 8 GB de HD e uma interface virtual de rede (*eth0*)
	Sistema Operacional *Windows* XP SP3 com instalação padrão. Não foram instalados outros serviços

Todos os *honeypots* foram implementados na rede *honeynet*, que foi configurada como *host-only* (rede virtual privada) para fazer a comunicação entre os *honeypots e* a interface virtual *eth1* do *honeywall*. No *link* da rede externa, existe ainda uma máquina virtual (IP 172.30.20.100) configurada para testar a configuração do ambiente *honeynet* virtual de autocontenção.

A rede de gerência é uma rede confiável usada para coletar e analisar remotamente os dados. Esta rede deverá ser utilizada ainda para administrar o *honeywall* (IP 172.30.10.30) e o *XenServer* (IP 172.30.10.10*).* A gerência vabe a um *host* dedicado exclusivo para esta finalidade. Um cabo *crossover* é utilizado para fazer *link* entre o *host* de gerência e o servidor físico.

Todos os *honeypots* foram configurados com a instalação padrão do *Linux Debian Wheezy 7.2* e *Windows* XP SP3. Foram feitas instalações e configurações dos serviços de DNS, FTP, NTP e WEB. Vale ressaltar que não foi aplicado nenhum processo de *hardening* para manter os sistemas mais seguros.

B. *Fase 2: Controle de Dados*

O controle de dados recebidos e enviados no ambiente *honeynet* virtual de autocontenção tem como finalidade filtrar quais dados podem ir para qual destino. Este controle cabe ao *firewall* (elemento que tem a finalidade de filtrar pacotes e de separar as duas redes: internet e *honeynet*) e pelo *iptables* configurado no *honeywall*. Foram definidas três regras para controlar o fluxo do tráfego:

- Qualquer individuo poderá realizar uma conexão da internet para a *honeynet*. Isso permite que um intruso explore os *honeypots*;
- O *firewall* controlará conexões feitas da rede *honeynet* com a internet para evitar que os intrusos usem os *honeypots* comprometidos para atacar outros sistemas em produção. Esta regra será replicada também no *firewall iptables* implementado no *honeywall*, para que haja uma redundância de controle de fluxo;
- A rede *honeypot* e a rede gerência não poderão se comunicar. Isso garante que os *honeypots* comprometidos não modifiquem ou destruam os dados coletados.

Ao mesmo tempo, um *script rc.firewall* (implementado no *iptables* do *honeywall*) é utilizado com a mesma finalidade, ou seja, prevenir ataques de dentro da rede *honeynet* para outros sistemas. O principal objetivo deste *script* é limitar o número de conexões (UDP, TCP, ICMP) que podem ser feitas para fora da rede *honeynet* em uma escala de tempo (mensal ou diária).

Nesta fase, foram implementadas duas camadas de segurança para diminuir o impacto de falhas durante o controle do tráfego de dados.

C. *Fase 3: Captura de Dados*

A captura de dados tem como finalidade coletar todas as atividades que ocorrem dentro da rede *honeynet*. Quanto maior o número de camadas (métodos de captura), mais se espera ter sucesso no projeto. O *script rc.firewall* (implementado no *honeywall*) registrará todas as conexões de entrada e saída em */var/log/messages* para indicar o início de um ataque.

Nesta arquitetura, o software detector de intrusões *snort*, implementado no *honeywall*, foi configurado com regras atualizadas e utilizado para capturar todo o tráfego da interface virtual *eth1* do *honeywall*. Isto foi feito para registrar todo o tráfego de entrada e saída da rede *honeynet*.

Finalmente, a última camada de captura de dados fica por conta da ferramenta *sebek*. Esta ferramenta tem como objetivo principal obter registros que permitam mais tarde recriar com precisão ataques em um *honeypot* [16]. Em cada *honeypot* foi instalado e configurado o cliente *sebek* para ser executado no *kernel*, com a finalidade de capturar todas as atividades dos invasores (teclas pressionadas, *upload* de arquivos, senhas). Estas atividades serão enviadas por um canal seguro para o servidor *sebek* instalado no *honeywall* através do protocolo UDP (porta 1101).

IV. Validação do Ambiente

A validação do ambiente foi feita através de vários testes na *honeynet* virtual de autocontenção para verificar se o sistema estava realmente funcionando. Desta forma, foi criada uma máquina virtual com o sistema operacional *Linux Kali* [17] (IP 172.30.20.100) na rede externa para simular alguns ataques.

O primeiro teste foi realizado quanto ao requisito controle de dados do ambiente para verificar se o *honeywall* estava coletando todos os dados de entrada e saída. Portanto, foi feito um *ping* da máquina virtual *Linux Kali* (172.30.20.100) para o *Honeypot* DNS (172.30.20.13). Com base na configuração feita no conjunto de regras do *snort*, foi possível capturar os pacotes ICMP (Figura 3).

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/27-15:56:18.831361 26:0:78:DC:4E:F8 -> 2E:C:53:6B:2B:85 type:0x800 len:0x62
172.30.20.100 -> 172.30.20.13 ICMP TTL:64 TOS:0x0 ID:29641 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:6594   Seq:2  ECHO
25 5C 3E 55 00 00 00 00 1B 58 0B 00 00 00 00 00  %\>U.....X......
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F  ................
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F   !"#$%&'()*+,-./
30 31 32 33 34 35 36 37                          01234567

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/27-15:56:18.831979 2E:C:53:6B:2B:85 -> 26:0:78:DC:4E:F8 type:0x800 len:0x62
172.30.20.13 -> 172.30.20.100 ICMP TTL:64 TOS:0x0 ID:55708 IpLen:20 DgmLen:84
Type:0  Code:0  ID:6594  Seq:2  ECHO REPLY
25 5C 3E 55 00 00 00 00 1B 58 0B 00 00 00 00 00  %\>U.....X......
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F  ................
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F   !"#$%&'()*+,-./
30 31 32 33 34 35 36 37                          01234567

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Fig. 3. Captura do pacote ICMP.

Ainda neste teste, o *snort* mostrou algumas assinaturas relacionadas ao pacote ICMP que são de extrema importância para determinar o sistema operacional utilizado pelo intruso para realizar o ataque:

- TTL: observa-se que o campo TTL (IP utilizado pelo intruso) está configurado como 64. Com base nestas informações foi possível deduzir que este pacote foi enviado por um computador executando o *Linux*;
- Tamanho do datagrama: requisições de eco ICMP geradas através do utilitário *ping* terão 84 *bytes* de comprimento em sistemas operacionais UNIX e semelhantes ao UNIX;
- Conteúdo da carga útil: dados de uma requisição eco ICMP enviados através do utilitário *ping* em sistemas operacionais UNIX, ou semelhante ao UNIX serão compostos exclusivamente por números e símbolos.

O *snort* foi configurado também para converter quaisquer informações ASCII encontradas no *payload* do pacote para o arquivo *snort.log*. Este procedimento é fundamental para analisar rapidamente as seções de texto simples, tais como as seções de FTP, TELNET ou IRCs.

O segundo teste teve como propósito verificar os limites de conexões de saída para o protocolo ICMP. Para realizá-lo, foi executado um *ping* do *Honeypot* XP (172.30.20.36) para a máquina virtual de teste *Linux Kali* (172.30.20.100). Este procedimento indicará que o *Honeypot* XP foi comprometido e que um intruso está tentando realizar conexões para fora do ambiente, podendo ser um ataque.

```
May 11 09:26:59 pituba kernel: OUTBOUND ICMP: IN=br0 OUT=br0 PHYSIN=eth1 PHYSOUT=eth0 SRC=172.30.20.36 DST=172
.30.20.100 LEN=60 TOS=0x00 PREC=0x00 TTL=128 ID=266 PROTO=ICMP TYPE=8 CODE=0 ID=512 SEQ=7424
May 11 09:27:00 pituba kernel: OUTBOUND ICMP: IN=br0 OUT=br0 PHYSIN=eth1 PHYSOUT=eth0 SRC=172.30.20.36 DST=172
.30.20.100 LEN=60 TOS=0x00 PREC=0x00 TTL=128 ID=268 PROTO=ICMP TYPE=8 CODE=0 ID=512 SEQ=7680
May 11 09:27:01 pituba kernel: OUTBOUND ICMP: IN=br0 OUT=br0 PHYSIN=eth1 PHYSOUT=eth0 SRC=172.30.20.36 DST=172
.30.20.100 LEN=60 TOS=0x00 PREC=0x00 TTL=128 ID=270 PROTO=ICMP TYPE=8 CODE=0 ID=512 SEQ=7936
May 11 09:27:02 pituba kernel: OUTBOUND ICMP: IN=br0 OUT=br0 PHYSIN=eth1 PHYSOUT=eth0 SRC=172.30.20.36 DST=172
.30.20.100 LEN=60 TOS=0x00 PREC=0x00 TTL=128 ID=272 PROTO=ICMP TYPE=8 CODE=0 ID=512 SEQ=8192
May 11 09:27:03 pituba kernel: OUTBOUND ICMP: IN=br0 OUT=br0 PHYSIN=eth1 PHYSOUT=eth0 SRC=172.30.20.36 DST=172
.30.20.100 LEN=60 TOS=0x00 PREC=0x00 TTL=128 ID=274 PROTO=ICMP TYPE=8 CODE=0 ID=512 SEQ=8448
May 11 09:27:04 pituba kernel: Drop icmp > 30 attemptsIN=br0 OUT=br0 PHYSIN=eth1 PHYSOUT=eth0 SRC=172.30.20.36
 DST=172.30.20.100 LEN=60 TOS=0x00 PREC=0x00 TTL=128 ID=276 PROTO=ICMP TYPE=8 CODE=0 ID=512 SEQ=8704
[root@pituba log]#
```

Fig. 4. Limites de conexões de saída do protocolo ICMP.

Quando o limite de conexão de saída ICMP (*HwICMPRATE=30*) for atingido, o *script rc.firewall* executará uma entrada “DROP ICMP” e bloqueará durante uma hora estas conexões de saída (Figura 4). Todas as conexões serão registradas pelo *iptables* no *honeywall* em */var/log/iptables*.

O terceiro teste (referente ao requisito de captura de dados) teve como objetivo verificar se a base de assinaturas do *snort* no *honeywall* estava atualizada e configurada para detectar ataques. Primeiro foi feito um *portscan* com o *nmap* da máquina atacante (172.30.20.100) para o *honeypot* WEB (172.30.20.25) com a finalidade de sondar e verificar quais eram os serviços que estavam sendo executados no *honeypot* (Figura 5).

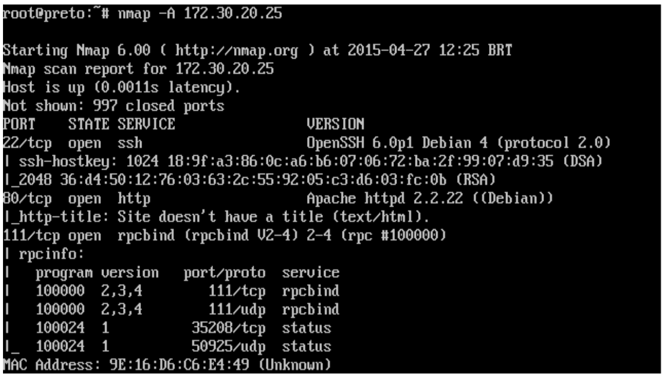


Fig. 5. Ataque *portscan* com *nmap* no *Honeypot* WEB.

Este ataque mostrou (vide Figura 5) que o *Honeypot* WEB estava executando os seguintes serviços: *OpenSSH 6.0p1* (porta 22/*tcp*), *Apache httpd 2.2.22* (porta 80/*tcp*) e *rpcbind v2-4* (porta 111/*tcp*). O intruso conseguiu ainda verificar o endereço *MAC* (9E:16:D6:C6:E4:49) e versão do sistema operacional (*Linux Debian*) utilizado.

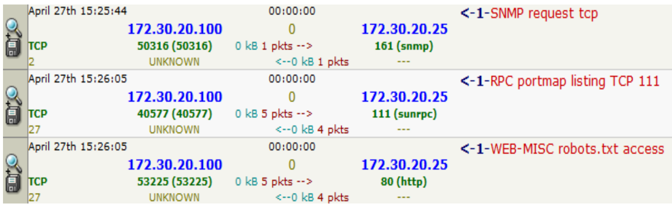


Fig. 6. Captura do ataque *portscan* feito pelo *nmap* no *Honeypot* WEB.

Conforme Figuras 5 e 6, o *snort* conseguiu detectar três ataques (em vermelho na Figura 6): *portscan* executado pelo *nmap* como uma tentativa de obter informações do *Honeypot* WEB através do protocolo SNMP direcionado para a porta 161/TCP. O ataque WEB-MISC *robots.txt Access* [18] detectado pelo *snort* informa que houve uma tentativa de coleta de informações a uma aplicação *web* potencialmente vulnerável.

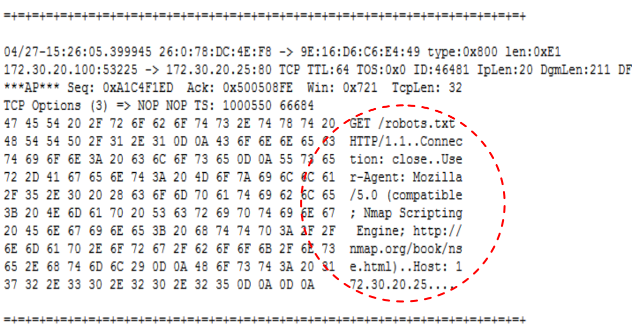


Fig. 7. Captura do ataque *portscan* em formato hexadecimal e ASCII.

Este mesmo ataque pode ser visto ainda de uma forma mais detalhada pelo administrador através da *payload* do pacote em dois formatos diferentes. O primeiro formato é dado em hexadecimal (coluna da esquerda). O segundo formato é a conversão em ASCII (coluna da direita). A Figura 7 informa que foi executado um *portscan* através do *nmap*.

V. Simulações e Resultados Obitidos

O objetivo deste estudo de caso é mostrar como os recursos apresentados neste trabalho podem ser utilizados por Órgãos Governamentais como fonte de pesquisa para coletar, analisar e estudar ataques e vulnerabilidades exploradas por invasores.

A. *O Ataque*

Neste estudo de caso, foram realizados dois ataques de força bruta que geralmente são utilizados para comprometer severamente um sistema. O ataque foi feito da máquina virtual *Linux Kali* (172.30.20.100) para o *Honeypot* FTP (172.30.20.21). O serviço *proftpd-1.3.4a* do *honeypot* foi configurado para aceitar apenas conexões com autenticação. Para esta simulação o intruso será representado pela máquina virtual *Linux Kali* (172.30.20.100).

Primeiramente foi executado um *portscan* com o *nmap* pelo intruso (Figura 8). Após a varredura, verificou-se que vários serviços estavam com estado OPEN, inclusive o FTP e SSH.

O primeiro ataque foi realizado no protocolo FTP através da ferramenta *xHydra*. Esta ferramenta faz escalação de privilégios através de quebra de senha online. A Figura 9 apresenta os usuários (luiza e marcia) e as senhas (senhaluiza e senhamarcia) que foram encontrados pelo *xHydra* durante o ataque.

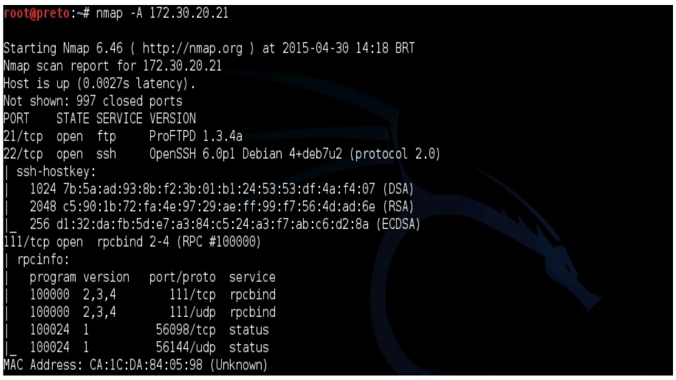


Fig. 8. Ataque *portscan* com *nmap* no *Honeypot* FTP.

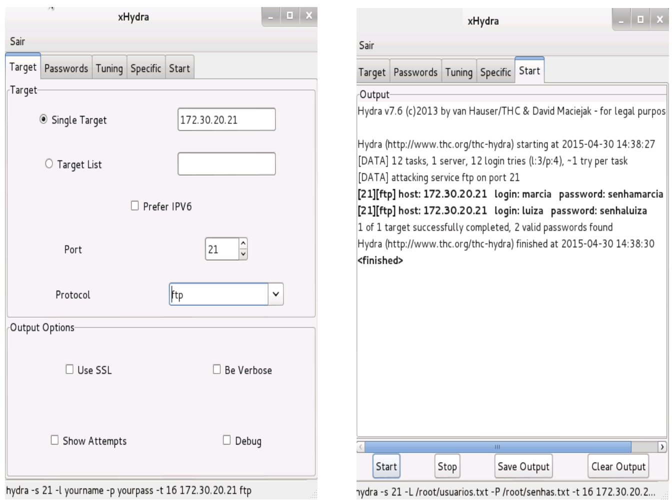


Fig. 9. Ataque de força bruta com *xHydra*.

Foram utilizadas as ferramentas *netcat* e *medusa* para realizar o segundo ataque no protocolo SSH. Primeiro foi executado o *netcat* para levantar o *banner* do serviço SSH. Depois executamos a ferramenta *medusa* para realizar o ataque de força bruta como mostrado na Figura 10. Após o ataque, a ferramenta retorna os usuários (luiza e marcia) e as senhas (senhaluiza e senhamarcia) que foram encontrados.

Em uma última etapa (Figura 11), o intruso realizou uma conexão FTP com o *honeypot* (172.30.20.21). Nesta conexão foram executados os seguintes comandos: *ls*, *cd Documentos*, *get seminario.txt*, *delete seminario.txt e exit.*

```
root@preto:~# nc 172.30.20.21 22
SSH-2.0-OpenSSH_6.0p1 Debian-4+deb7u2
^C
root@preto:~# medusa -M ssh BANNER:SSH-2.0-OpenSSH_6.0pl -h 172.30.20.21 -U /root/usuarios.txt
 -P senhas.txt | grep SUCCESS
ACCOUNT FOUND: [ssh] Host: 172.30.20.21 User: luiza Password: senhaluiza [SUCCESS]
ACCOUNT FOUND: [ssh] Host: 172.30.20.21 User: marcia Password: senhamarcia [SUCCESS]
root@preto:~#
```

Fig. 10. Ataque de força bruta com *medusa*.

```
root@preto:~# ftp 172.30.20.21
Connected to 172.30.20.21.
220 ProFTPD 1.3.4a Server (FTP RAEOO) [::ffff:172.30.20.21]
Name (172.30.20.21:root): luiza
331 Password required for luiza
Password:
230-Welcome, archive user luiza@172.30.20.100 !
230-
230-The local time is: Thu Apr 30 17:44:34 2015
230-
230-This is an experimental FTP server.  If you have any unusual problems,
230-please report them via e-mail to <root@brotas.raeoo.com.br>.
230-
230 User luiza logged in
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful
150 Opening ASCII mode data connection for file list
drwxr-xr-x   2 root     root         4096 Apr 27 14:28 Arquivos
drwxr-xr-x   2 root     root         4096 Apr 30 16:15 Documentos
drwxr-xr-x   2 root     root         4096 Apr 27 14:28 Palestras
-rw-r--r--   1 root     root          170 Sep  4  2014 welcome.msg
226 Transfer complete
ftp> cd Documentos
250 CWD command successful
ftp> ls
200 PORT command successful
150 Opening ASCII mode data connection for file list
-rw-r--r--   1 root     root           76 Apr 30 16:15 seminario.txt
226 Transfer complete
ftp> get seminario.txt
local: seminario.txt remote: seminario.txt
200 PORT command successful
150 Opening BINARY mode data connection for seminario.txt (76 bytes)
226 Transfer complete
76 bytes received in 0.01 secs (14.7 kB/s)
ftp> delete se
seminario.txt  senhas.txt
ftp> delete seminario.txt
550 seminario.txt: Permission denied
ftp> exit
221 Goodbye.
root@preto:~#
```

Fig. 11. Conexão FTP realizada pelo intruso.

B. *Analisando os Ataques no Ambiente*

Normalmente os usuários maliciosos iniciam um ataque com a coleta de informações. Eles precisam explorar quais vulnerabilidades e *backdoors* existem nos sistemas. Em 30 de abril, o *snort* detectou um ataque *portscan* no *Honeypot* FTP. Neste ataque, o intruso tentou explorar quais eram os serviços que estavam sendo executados no sistema.

No mesmo dia, o *snort* alertou que um dos *honeypots* havia sido comprometido. Neste caso, o ataque foi detectado e registrado conforme Figura 12. Este alerta do *snort* nos informou sobre uma tentativa de conexão SSH com um dos nossos *honeypots*. A seguir apresentamos as informações de cabeçalho do primeiro pacote (Figura 13):

- O pacote foi capturado em 30 de abril às 14h16min;
- O pacote foi enviado da porta 37492 da máquina 172.30.20.100 para porta 22 do *honeypot* 172.30.20.21;
- Esse pacote encapsula o protocolo TCP com TTL (*Time to Live*) 64, TOS (*Type of Service*) igual a zero, ID 19439 e comprimento de cabeçalho IP de 20 *bytes*;
- Número de seqüência 0xA834A7ED, número de confirmação *Ack* 0xC8892E29, *Win* (tamanho da janela) 0xE5 e *TcpLen* (Comprimento de cabeçalho TCP) 32 bytes;
- As opções TCP com dois NOPs e um TS (*timestamp*).

Para obter informações detalhadas sobre os pacotes enviados, analisamos os dados que foram detectados no *payload* do pacote. Conforme a Figura 14, confirmamos que o intruso realizou um ataque de força bruta através da ferramenta *Medusa* no serviço SSH.

Ainda no mesmo dia, às 14h 35min recebemos outro alerta referente a uma conexão FTP no mesmo *honeypot* (Figura 15). Verificamos que a conexão foi feita da mesma máquina que realizou o ataque anterior. Este alerta do *snort* nos informou que a máquina 172.30.20.100:38237 estava tentando realizar uma conexão FTP com o *honeypot* (172.30.20.21:21).

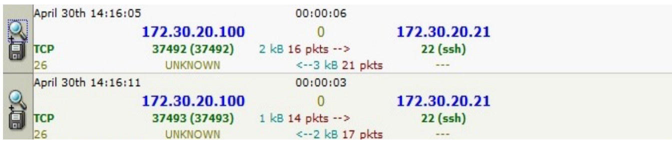


Fig. 12. Captura do ataque de força bruta com *medusa*.

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/30-14:16:05.155091 0:15:5D:47:1:8 -> CA:1C:DA:84:5:98 type:0x800 len:0x56
172.30.20.100:37492 -> 172.30.20.21:22 TCP TTL:64 TOS:0x0 ID:19439 IpLen:20 DgmLen:72 DF
***AP*** Seq: 0xA834A7ED  Ack: 0xC8892E29  Win: 0xE5  TcpLen: 32
TCP Options (3) => NOP NOP TS: 3730433 119684
```

Fig. 13. Informações de cabeçalho do ataque de força bruta.

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/30-14:16:05.155091 0:15:5D:47:1:8 -> CA:1C:DA:84:5:98 type:0x800 len:0x56
172.30.20.100:37492 -> 172.30.20.21:22 TCP TTL:64 TOS:0x0 ID:19439 IpLen:20 DgmLen:72 DF
***AP*** Seq: 0xA834A7ED  Ack: 0xC8892E29  Win: 0xE5  TcpLen: 32
TCP Options (3) => NOP NOP TS: 3730433 119684
53 53 48 2D 32 2E 30 2D 4D 45 44 55 53 41 5F 31  SSH-2.0-MEDUSA_1
2E 30 0D 0A                                      .0..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/30-14:16:05.164615 CA:1C:DA:84:5:98 -> 0:15:5D:47:1:8 type:0x800 len:0x69
172.30.20.21:22 -> 172.30.20.100:37492 TCP TTL:64 TOS:0x0 ID:64942 IpLen:20 DgmLen:91 DF
***AP*** Seq: 0xC8892E29  Ack: 0xA834A801  Win: 0x712  TcpLen: 32
TCP Options (3) => NOP NOP TS: 119687 3730433
53 53 48 2D 32 2E 30 2D 4F 70 65 6E 53 53 48 5F  SSH-2.0-OpenSSH_
36 2E 30 70 31 20 44 65 62 69 61 6E 2D 34 2B 64  6.0p1 Debian-4+d
65 62 37 75 32 0D 0A                             eb7u2..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Fig. 14. Captura do ataque de força bruta em hexadecimal e ASCII.

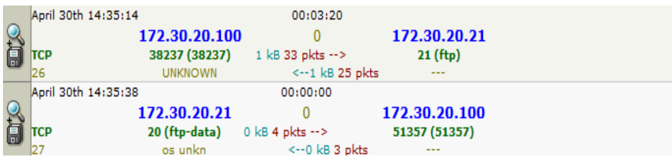


Fig. 15. Captura do ataque no *Honeypot* FTP.

Analisando novamente o *payload* do pacote, verificamos que a conexão FTP foi feita em texto simples, ou seja, os dados não foram criptografados. Isso significa que podemos

decodificar os dados e capturar todas as teclas digitadas. Portanto, foi possível verificar detalhes do ataque em formato hexadecimal e ASCII durante a conexão da máquina atacante com o *honeypot* FTP (Figura 16).

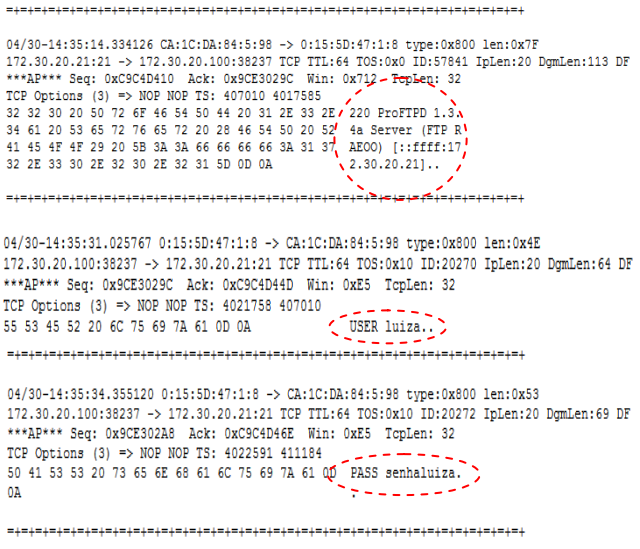


Fig. 16. Captura do ataque em *hexadecimal* e ASCII no *Honeypot* FTP.

Dentre as informações obtidas, pode-se verificar: endereço MAC (00:15:5D:47:01:08), endereço IP (172.30.20.100) e porta de origem (38237) da máquina que estava atacando; a porta de destino (21); a versão do servidor FTP (*ProFTPD 1.3.4a*), o usuário (*luiza*) e a senha (senhaluiza) utilizados pelo intruso para realizar a autenticação no servidor.

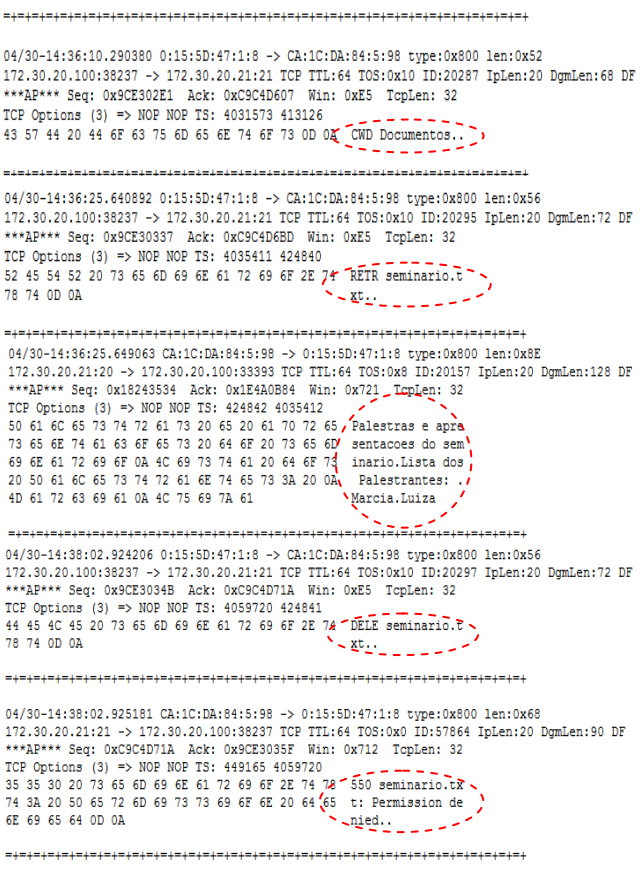


Fig. 17. Exploração de ataque no *Honeypot* FTP.

Neste mesmo cenário, continuando a análise do ataque no *Honeypot* FTP, a Figura 17 mostra ainda que o intruso conseguiu entrar no diretório */Documentos* do servidor e fazer *upload* do arquivo *seminário.txt*. O intruso tentou também remover o arquivo *seminario.txt*, mas a captura do pacote mostra que ele não teve sucesso (*Permission denied)*.

VI. Conclusões

Neste trabalho, um ambiente *honeynet* virtual de autocontenção foi desenvolvido como solução de pesquisa para analisar vulnerabilidades e acompanhar novas formas de atividades de intrusos em redes. O desenvolvimento do ambiente proposto foi dividido em três fases, buscando uma otimização dos processos apresentados. Na primeira fase mostramos a arquitetura proposta, definindo o *hypervisor* utilizado e descrevendo como o *honeywall* e os *honeypots* foram configurados. Na segunda fase, um *firewall* e um *script* implementados no *iptables* do *honeywall* foram utilizados para controlar o fluxo de dados. Na terceira fase, foram implementadas três camadas para coletar as atividades dentro da *honeynet*: um *script* a fim de registrar conexões de entrada e saída; o *snort*, configurado com regras atualizadas para capturar todo o tráfego; e a ferramenta *sebek*, utilizada para recriar com precisão os ataques sofridos nos *honeypots*.

Com o objetivo de validar o ambiente, vários testes foram feitos. O primeiro teste foi realizado no requisito controle de dados para verificar se o *honeywall* estava coletando todos os dados de entrada e saída. O propósito do segundo teste foi verificar os limites de conexões de saída do protocolo ICMP. O terceiro e último teste teve como finalidade verificar se a base de assinaturas do *snort* no *honeywall* estava configurada e atualizada para detectar os ataques. Por fim, fizemos um estudo de caso através da simulação de dois ataques de força bruta para mostrar o funcionamento do ambiente e obter os resultados.

Tais procedimentos indicam a validade de utilizar o ambiente em aplicações governamentais, visto que são providas todas as funcionalidades ao alcance do gestor para a captura de detalhes de ataques, seja visando a atualização de medidas de proteção, seja para efeito de demonstração forense.

Como trabalhos futuros, cabe testar outras formas de ataques; executar o ambiente por um período determinado de tempo e verificar os ataques reais oriundos da Internet para analisá-los e descrever o que aconteceu, o que foi aprendido; criar imagens de *honeypots* comprometidos para uma análise forense mais detalhada; extrair e analisar dados a partir de *dumps* de memória e incluir *honeypots* com sistemas operacionais utilizados por *smartphones.*

Agradecimentos

Os autores agradecem às Agências brasileiras de pesquisa e inovação CAPES (Projeto FORTE, Edital CAPES Ciências Forenses 25/2014), FINEP (Convênio RENASIC/PROTO 01.12.0555.00), pelo suporte a este trabalho..

Referências

[1] CERT.BR. *Incidentes Reportados ao CERT.br – Janeiro a Dezembro de 2014*. Disponível em: <http://www.cert.br/stats/incidentes/2014-jan-dec/analise.html>. Acessado em: 15/04/2015.

[2] SCARFONE, K. e MELL, P. *Guide to Intrusion Detection and Prevention Systems (IDPS)*. Recommendations of the National Institute of Standards and Technology, Gaithersburg, 2007.

[3] PROJECT, Honeynet. *Conheça seu inimigo - O Projeto Honeynet*. São Paulo: Pearson Education do Brasil, 2002.

[4] STOLL, C. *Stalking the wily hacker*. Commun. ACM, ACM, New York, NY, USA, v. 31, n. 5, p. 484–497, 1988. ISSN 0001-0782.

[5] CHESWICK, B. *An evening with berferd in which a cracker is lured, endured, and studied*. In: In Proc. Winter USENIX Conference. [S.l.: s.n.], 1990. p. 163–174.

[6] COHEN F. *A Note on the Role of Deception in Information Protection - 1998*. Disponível em: <http://all.net/journal/deception/deception.html/>. Acessado em 17/04/2015.

[7] THE HONEYNET PROJECT. Disponível em: < https://www.honeynet.org//>. Acessado em: 01/04/2015.

[8] SOUSA JR, R. T.; SILVA, T. A.; ALBUQUERQUE, R. O. *Ambiente baseado em agentes de software para o auxílio na detecção e estudo de ataques em redes de computadores*. Proceedings of the 1st International Conference on Cyber Crime Investigation ICCyber'2004. Brasília, 2004. p. 156-161.

[9] SILVA, T. A.; ALBUQUERQUE, R. O.; BUIATI, F. M.; PUTTINI, R. S.; SOUSA JR, R. T. *A Community of Agents for Trapping Attacks Against Network Services and Redirecting Traffic Attacks to a Honeynet*. Proceedings of the First International Conference on Internet Technologies and Applications (ITA 05), Wrexham (UK), 2005. p. 135-143.

[10] SPITZNER, L. *Honeypots – Tracking Hackers*. *Indianápolis, IN: AddisonWesley*, 2002.

[11] THE HONEYNET PROJECT. *Know Your Enemy: Defining Virtual Honeynets*. Janeiro de 2003. Disponível em: <http://old.honeynet.org/papers/virtual/>. Acessado em: 01/04/2015.

[12] CITRIX. *Optimized server virtualization for all your workloads*. Disponível em < http://www.citrix.com/products/xenserver/overview.html>. Acessado em 11/04/2015.

[13] XENSERVER. *XenServer Open Source Virtualization Plataform*. Disponível em < http://xenserver.org/>. Acessado em 10/04/2015.

[14] THE HONEYNET PROJECT. *Know Your Enemy: Honeywall CDROM Roo*. Agosto de 2005. Disponível em:<http://old.honeynet.org/papers/cdrom/roo/>. Acessado em: 02/04/2015.

[15] THE HONEYNET PROJECT. *Roo CDROM User's Manual*. Maio de 2007. Disponível em: <http://old.honeynet.org/tools/cdrom/roo/manual/6-maintain.html/>. Acessado em: 02/04/2015.

[16] THE HONEYNET PROJECT. *Know Your Enemy: Sebek*. Novembro de 2003. Disponível em: <http://www.honeynet.org/papers/sebek/>. Acessado em: 05/04/2015.

[17] KALI LINUX. *Our Most Advanced Penetration Testing Distribution, Ever*. Disponível em < https://www.kali.org/>. Acessado em 20/04/2015

[18] SNORT. *Snort FAQ*. Disponível em <https://www.snort.org/faq/readme-thresholding>. Acessado em 22/04/2015.

DOI: 10.5769/C2015010 or http://dx.doi.org/10.5769/C2015010

Estudo de Rótulos de Tempo em Sistemas de arquivos HFS+

Arelian Monteiro Maia, Felippe Pires Ferreira e Lindeberg Pessoa Leite

***Resumo*— Para análise pericial de um sistema de arquivo, os metadados armazenam dados relevantes, principalmente os rótulos de tempo. Dessa modo, este trabalho objetiva determinar o comportamento dos rótulos de tempo do sistema de arquivos HFS+ em diversos cenários na plataforma OS X, versões Mavericks e Yosemite. Em uma máquina virtual Mavericks e Yosemite, realizaram-se simulações de operações comuns com o intuito de entender o comportamento dos metadados de rótulos de tempo no sistema de arquivo HFS+. Para exposição dos resultados dos experimentos, foram elaboradas tabelas que apresentam o mapeamento entre a ação executada e as alterações nos rótulos de tempo.**

***Palavras-Chave*— Sistema de arquivo, Metadados, Rótulos de tempo, HFS+**

***Abstract*— For expert analysis of a file system, metadata store relevant data, especially the labels of time. In this way, this study aims to determine the behavior of the HFS+ File System timestamps in diverse scenarios in OS X platform, versions Mavericks e Yosemite. In a virtual machine Mavericks e Yosemite, there were simulations of common operations in order to understand the behavior of timestamps metadata in the HFS+ file system. To display the results of the experiments, tables were prepared presenting the mapping between the action taken and the changes on the labels of time.**

***Keywords*— File system, Metadatas, Timestamp, HFS+**

I. Introdução

Ao realizar uma análise pericial, informações temporais de um arquivo, como por exemplo, data de criação, data de modificação e data de acesso são elementos essenciais para criar uma linha do tempo (*timeline*). Entretanto, devido aos rótulos de tempo serem influenciados por vários fatores como o sistema de arquivo, *hardware*, o sistema operacional em execução e suas configurações, normalmente a extração das informações temporais não são diretas [1]. Desse modo, conhecer como são utilizados os registro de rótulos de tempo é fundamental para subsidiar Laudos periciais.

A demanda por exames periciais em dispositivos da Apple na Perícia da Polícia Federal vem crescendo, consequência do aumento de sua presença no mercado. Portanto, o estudo do funcionamento e utilização do sistema de arquivos destes dispositivos é uma necessidade.

O Hierarchical File System Plus (HFS+) é o principal sistema de arquivos da linha de produtos da Apple, substituindo o Hierarchical File System (HFS) em sistema Mac OS X e também é um dos formatos utilizados em sistema iOS. [7]

Arelian Monteiro Maia, Felippe Pires Ferreira e Lindeberg Pessoa Leite Mestrandos do Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília-DF, Brasil. E-mails: arelianmaia@gmail.com, felippepipipe@gmail.com, lindpessoa@gmail.com.

A. Objetivo

Este trabalho possui o objetivo de determinar o comportamento dos rótulos de tempo do sistema de arquivos HFS+ em diversos cenários na plataforma OS X, versões Mavericks e Yosemite.

B. Método

O método adotado neste trabalho foi executar em máquinas virtuais a plataforma OS X, nas versões Mavericks e Yosemite, realizando operações comuns de usuários como criar, copiar, mover, compactar, entre outras. A análise dessas operações na estrutura do sistema de arquivos HFS+ servirá de base para fazer afirmações acerca das alterações que os metadados de arquivos e diretórios sofrem e auxiliar a criação de uma *timeline*.

Foram criados dois cenários de teste. No primeiro cenário foi utilizado a interface gráfica para realizar as operações. Enquanto no segundo cenário foi utilizado linhas de comando para executar as operações. Com auxílio da interface gráfica do Mac OS X foram criadas duas partições HFS+. Em uma partição, criaram-se dez pastas, onze arquivos de texto e um arquivo de imagem. As operações foram executadas sobre esses objetos conforme descritas na tabela na seção Resultados.

As partições foram espelhadas e por meio do FTK Imager 3.2.0.0 [1] e do software HFSExplorer 0.23 [2] foram registrados os valores das datas de cada pasta e arquivo. Posteriormente, realizaram-se as operações de copiar, colar, mover, compactar, etc. Após isso, gerou-se uma nova imagem das partições para serem realizadas as comparações entre as datas antes e depois das operações de cada arquivo e pasta.

No segundo cenário, foi elaborado um script com os comandos de manipulação de arquivos e pastas. Antes da execução de cada comando de manipulação, os arquivos e pastas eram submetidos ao comando stat para registrar as datas antes da manipulação, e posteriormente as datas eram novamente chegadas para fins comparativos. Os dois cenários foram implementados com a utilização de discos rígidos e mídias removíveis.

[1]Forensic Toolkit, ou FTK, é um software de computação forense criado pela AccessData. Ele é capaz de processar um dispositivo, como um disco rígido, em busca de informações diversas. [9]

[2]É uma aplicação para visualizar e extrair arquivos de um volume HFS + ou em um volume HFSX, que estão localizados em uma unidade física, como uma imagem de um disco .dmg, ou em um dump de sistema de arquivos em formato raw[10]

C. Trabalhos correlatos

Um estudo do comportamento de rótulos de tempo em sistemas NTFS foi realizado por Junior, Cleber Scoralick [1]. Seu trabalho baseou-se em Chow et al [2], que apresentaram regras gerais baseadas em alguns dos rótulos de tempo existentes. Ele também inspirou-se em Bang et al. [3] e Bang, Yoo e Lee [4] que avaliaram mais rótulos de tempo e um número maior de operações. Este artigo busca realizar um estudo semelhante, mas aplicado ao sistema de arquivo HFS+.

II. Aspectos Técnicos

O HFS+, também chamado Mac OS Extended, foi introduzido em 1998 para superar os problemas da HFS e se tornar o sistema de arquivos principal usado em computadores Mac. HFS+ é uma versão melhorada do HFS suportando arquivos e volumes maiores pelo o uso de endereços de blocos de alocação de 32 bits e Unicode para nomes de arquivos. Ele também suporta múltiplos atributos para arquivos, como *journaling*, registros de textitinline attribute data, lista de controle de acesso baseado em arquivos de segurança e compatibilidade com os modelos de permissão de arquivo em outra plataformas como Windows. [5]

Assim como HFS, HFS+ divide o volume em setores de 512 bytes e agrupa-os em blocos de alocação, normalmente 8, e atribui a um arquivo. Blocos de alocação são endereçados por ponteiros de 32 bits [6]. Para reduzir a fragmentação do arquivo, blocos de alocação contíguas chamados *Clumps* são atribuídos aos arquivos. O número de blocos de alocação por *Clump* é fixa e é especificado em Volume Header. Os primeiros 1024 bytes e últimos 512 bytes de volume são reservados. O *Volume Header* está localizado imediatamente após primeiros 1024 bytes e é fixo. O *Alternate Volume Header* que é réplica do *Volume Header* está localizado nos 1024 bytes antes do final do volume e também é fixo. [7]

O *Volume Header* armazena rótulos do tempo, o número de arquivos sobre o volume, localização de outras estruturas sobre o volume, tamanho de blocos de alocação, tamanho de aglomerados, etc. [7]

Um volume HFS+ tem cinco arquivos especiais que são utilizados para organizar o espaço do volume utilizado para armazenar pastas, arquivos e atributos. São eles:

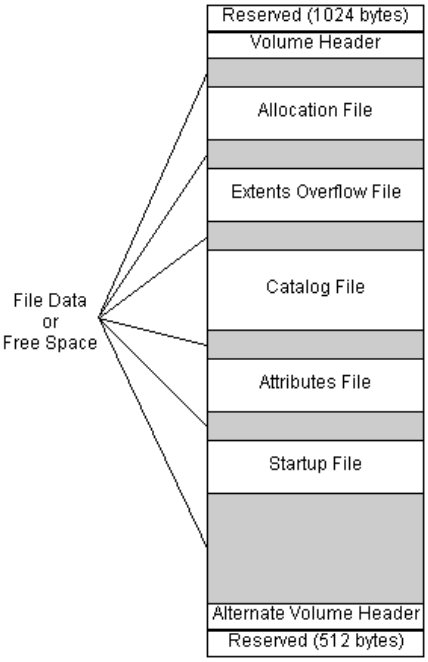


Fig 1. Layout do HFS+ [5]

A. Allocation File

O *Allocation File* é um bitmap que controla a utilização dos blocos de alocação. Ele mantém o controle do que está livre e em uso pela representação de cada bloco por um bit. Isso é equivalente ao *Bitmap Volume* do HFS. A principal diferença entre *Bitmap Volume* e *Allocation File* é que este é um arquivo comum que pode existir em qualquer lugar no volume, podendo diminuir ou aumentar de tamanho e não precisa ser contíguo. Enquanto aquele sempre reside em área reservada e seu tamanho é fixo. A localização do primeiro extent do *Allocation File* é armazenado em *Volume Header*. Esta arquitetura de *File Allocation* induz a flexibilidade no sistema de arquivos HFS+ não encontrada em HFS. [7]

B. Catalog File

O *catalog file* é uma árvore-B que armazena a hierarquia de pastas e arquivos. Ele descreve todos os arquivos e pastas do volume incluindo os arquivos especiais e da hierarquia no volume. É semelhante ao *Catalog File* do HFS. O *Catalog File* é organizado em uma árvore-B para permitir pesquisas rápida e eficientes por meio de uma grande hierarquia. Este arquivo contém informações vitais sobre cada arquivo e pasta juntamente com as informações do catálogo. A principal diferença entre os registros em HFS e HFS+ é que no *Catalog File* em HFS+ os nós da árvore-B relativo aos arquivos e pastas contêm mais informações e podem ter diferentes tamanho ao contrário do HFS. A localização do primeiro extent do *Catalog File* é armazenado no *Volume Header*. *Catalog File* contém nó de cabeçalho, nós índices, nós folhas e, se necessário, mapa de nós. Cada arquivo ou pasta do *Catalog File* é identificado por um único *Catalog Node ID* (CNID). Para pasta, CNID é chamado FolderID e para arquivos FileID. [7]

C. *Extent Overflow File*

O *Extents Overflow File* é utilizado para mapear os extents (áreas contíguas de um arquivo) extras dos arquivos que contêm mais que oito extents. Os primeiros oito extents são listados no registro correspondente ao arquivo no *catalog file*. Está estruturado como uma árvore-B. [7]

D. *Bad Block File*

Bad Block File é usado para marcar e registrar as áreas do volume que contém blocos danificados. O *Extent Overflow File* é usado para armazenar informações sobre os extents de *Bad Block File*. [7]

E. *Attributes File*

O *attribute file* contém metadados adicionados em pastas e arquivos pelas aplicações. Ele é um arquivo especial que não possui uma entrada no *Catalog File*. Um volume não pode ter *Attributes File* em caso de sua descrição no *Volume Header* para alocação de blocos seja 0. *Attributes File* é uma Arquivo B-tree estruturado onde os nós podem conter registros conhecidos como atributos. Um *Attribute File* pode ter 3 tipos de atributos [7]:

- *Inline Data Attributes* que contêm pequena atributos;
- *Attributes Data Fork* que contêm referências para um máximo de 8 extents;
- *Extended Attributes* que contêm referências a mais 8 extents para os atributos de dados;

F. *Startup File*

O *startup file* é um arquivo especial que facilita o boot em computadores não-Mac. O *boot loader* pode encontrar o arquivo *startup File* sem total conhecimento do formato de uma partição HFS+. Em vez disso, o *Volume Header* contém a localização dos primeiros oito extents do *startup File*. Este arquivo pode conter mais de oito extents, os quais serão colocados no *Extents Overflow File*. [7]

G. *Rótulo de tempo no HFS+*

HFS+ armazena rótulos de tempo em várias estruturas de dados, incluindo *Volume Header* e registros de catálogo. Estas datas são armazenados em inteiros de 32 bits sem sinal (UInt32) contendo o número de segundos desde 01/jan/1904 00:00:00 GMT, tendo como data máxima 06/fev/2040 06:28:15 GMT. Como as datas registradas estão entre 1900 e 2100, não se considera os segundos bissextos. [5]

A implementação é responsável por converter esses tempos para o formato esperado pelo software cliente. Por exemplo, o gerenciador de arquivos do Mac OS converte rótulos de tempo para hora local; a implementação Mac OS HFS+ converte a hora local para GMT, quando apropriado. [5]

A documentação oficial da Apple sobre a implementação do HFS+ [8], encontram-se descritos rótulos de tempo no *volume header/alternate volume header* e nas informações sobre pastas e arquivos no *Catalog File*:

Volume Header e *Alternate Volume Header* [7]

- *createDate* - Rótulo de tempo de quando o volume foi inicializado.
- *modifyDate* - Rótulo de tempo de quando o volume foi modificado pela última vez.
- *backupDate* - Rótulo de tempo de quando foi feito o último backup do volume. Deve ser atualizado por alguma aplicação de backup e não pelo sistema.
- *checkedDate* - Rótulo de tempo de quando foi realizado a última verificação de consistência no volume. Tipicamente alterado na utilização de ferramentas de checagem de disco como *Disk First Aid*.

Registro de pasta no *catalog file* [7]

- *createDate* - Rótulo de tempo de quando a pasta foi criada. Diferentemente da *createDate* do *volume header*, essa data está armazenada em GMT.
- *contentModDate* - Rótulo de tempo da última modificação do conteúdo da pasta, isto é, a última vez em que um arquivo ou pasta foi criado ou deletado dentro dessa pasta, ou quando um arquivo ou pasta foi movido para outra pasta.
- *attributeModDate* - Rótulo de tempo da última vez em que qualquer campo no registro de catálogo da pasta foi alterado.
- *accessDate* - Rótulo de tempo em que o conteúdo da pasta foi lido pela última vez. Criado para compatibilidade do Mac OS X com o POSIX, tem o valor de zero quando criado pelo Mac OS tradicional.
- *backupDate* - Rótulo de tempo de quando foi feito o último backup da pasta. Deve ser atualizado por alguma aplicação de backup e não pelo sistema.

Registro de arquivos no *Catalog File* [7]

- *createDate* - Rótulo de tempo de quando o arquivo foi criado. Diferentemente da *createDate* do *volume header*, essa data está armazenada em GMT.
- *contentModDate* - Rótulo de tempo em que o conteúdo do arquivo foi modificado. Entenda-se conteúdo não apenas os dados do arquivo, mas qualquer informação associada a ele (resource) como ícone, vídeo *QuickTime*, som, e outros.
- *attributeModDate* - Rótulo de tempo da última vez em que qualquer campo no registro de catálogo do arquivo foi alterado.
- *accessDate* - Rótulo de tempo em que o conteúdo do arquivo foi lido pela última vez. Criado para compatibilidade do Mac OS X com o POSIX, tem o valor de zero quando criado pelo Mac OS tradicional.
- *backupDate* - Rótulo de tempo de quando foi feito o último backup da pasta. Deve ser atualizado por alguma aplicação de backup e não pelo sistema.

III. Resultados obtidos

As tabelas abaixo mostram os resutados obtidos no OS X, versões Mavericks 10.9.0 e Yosemite 10.10.3 com o sistema de arquivo HFS+, versão 4. Os softwares FTK Imager 3.2 e HSFExplorer 0.23 foram utilizados para acessar a estrutura do sistema de arquivos.

Ação	Data de Criação	Data de Modificação	Data de Alteração de Atributos	Data de Acesso	Data de Backup
Mover diretório na mesma partição					
Mover arquivo na mesma partição				x	
Copiar diretório na mesma partição			x	x	
Copiar arquivo na mesma partição			x	x	
Mover diretório para partição diferente			X (diretório na partição de destino alterado)	X (diretório na partição de origem e de destino alterados)	
Mover arquivo para partição diferente			X (arquivo na partição de destino alterado)	X (arquivo na partição de origem e de destino alterados)	
Copiar diretório para partição diferente			X (diretório na partição	X (diretório na partição	

Tabela 1: Pela interface gráfica do sistema operacional

Ação	Data de Criação	Data de Modificação	Data de Alteração de Atributos	Data de Acesso
Alterar permissões do arquivo: chmod xxx			X	
Alterar permissões de pasta: chmod xxx			X	
Arquivo oculto Chflags hidden			X	
Pasta oculta Chflags hidden			X	

Tabela 3: Linha de comando para manipulação de atributos

Ação	Data de Criação	Data de Modificação	Data de Alteração de Atributos	Data de Acesso
Cat				X
Sed				x

Tabela 4: Linha de comando para leitura de arquivo

Ação	Data de Criação	Data de Modificação	Data de Alteração de Atributos	Data de Acesso
Compactar Pasta: tar -cf				x
Compactar Arquivo: tar -cf				x
Descompactar Pasta: tar –xf	x		x	X
Descompactar Arquivo: tar –xf			x	x
Arquivo .tar após descompactação				x
Compactar Arquivo: gzip			x	
Descompactar Arquivo: gzip -d			x	x
Compactar Pasta: zip				
Compactar Arquivo: zip				x
Descompactar Pasta: unzip	x		x	
Descompactar Arquivo: unzip	x		x	x
Arquivo .zip após descompactação				x

Tabela 2: Linha de comando para compactação e descompactação

Ação	Data de Criação	Data de Modificação	Data de Alteração de Atributos	Data de Acesso
Ls				X
Cd				
Criação de diretório filho (mkdir) alteração no diretório pai		x	x	
Remoção de arquivo (rm) alteração no diretório pai				x
Remoção de diretório filho (rm -rf) alteração no diretório pai		x	x	
Cópia de arquivo (cp) na mesma partição:				
Arquivo na origem				x
Arquivo no destino	x	X	X	x
Diretório de origem		X	X	
Diretório de destino		x	x	
Cópia de arquivo (cp) em outra partição:				
Arquivo na origem				x
Arquivo no destino	X	X	X	x
Cópia de arquivo (cp -a) preservando atributos:				
Arquivo na origem				x
Arquivo no destino			x	
Cópia de diretório (cp -a) preservando atributos:				
Diretório de origem				x
Diretório de destino			x	x
Cópia de diretório (cp -r) na mesma partição:				
Diretório de origem				
Diretório de destino				
Diretório filho	x	x	x	x

Tabela 5: Linha de comando para manipulação

IV. Conclusões

Este artigo objetivou determinar o comportamento dos rótulos de tempo do sistema de arquivo HFS+ em diversos cenários na plataforma OS X, versões Mavericks e Yosemite.

É possível observar comportamentos semelhantes em alguns comandos executados pela interface gráfica e pela interface de linha de comando. Entretanto, não é possível afirmar isso em todos os casos, visto que a implementação da ação pode executar instruções diferentes ou sequências diferentes. Um dos comportamentos semelhantes observados foi a execução de compactação e descompactação utilizando o formato .zip. tempo.

Algumas ações semelhantes tiveram comportamentos diferentes, como a manipulação de arquivos e diretórios. Enquanto na utilização de interface gráfica a data de criação foi preservada, em alguns casos de execução de comandos pelo terminal esta data sofreu alterações.

Grande parte dos comandos utilizados fazem alterações na data de acesso do objeto, com exceção dos comandos de manipulação de atributo. Logo, ações como leitura, listagem ou cópia fazem alterações neste campo. Uma *timeline* baseada neste campo pode auxiliar na criação de uma trajetória de utilização do sistema de arquivos pelo usuário.

O Backup descrito nos resultados da seção anterior foi realizado através do software *Time Machine* do Mac OS X. Os arquivos e diretórios originais que sofreram backup não tiveram suas datas alteradas, entretanto, os arquivos resultantes do procedimento sofreram alterações em suas datas de modificação de atributos e de acesso. Apesar de existir um atributo de data de backup (*backupDate*) em cada arquivo e pasta, este atributo não foi alterado após a execução do backup.

O experimento foi realizado com discos rígidos e removíveis com o sistema operacional em estudo. Foram simuladas operações de arquivos entre discos rígidos, entre discos removíveis e entre disco rígido e disco removível. Apesar da variação das mídias utilizadas, os resultados obtidos foram os mesmos, independente se eram discos rígidos ou removíveis. O que diferenciava os resultados eram apenas os comandos executados ou as partições envolvidas. Logo, a cópia de um arquivo entre discos rígidos apresentava o mesmo comportamento que a cópia de um arquivo entre discos removíveis. Mas apresentava resultados diferentes a depender se a copia ocorreria entre diferentes partições ou se a cópia ocorria na mesma unidade.

Como trabalho futuro, pode-se usar as datas para criar uma ferramenta de *timeline*. Desse modo, é possível verificar a consistência nas datas, de modo a encontrar possíveis adulterações intencionais. Ademais, repetir os testes no hfsx e em outros sistemas operacionais como o GNU/Linux para verificar se o comportamento descrito neste artigo se mantém quando executado em outras plataformas ou versões do sistema de arquivos. Por fim, é possível ainda analisar o comportamento dos CNID (catalog node ID) e relacioná-lo com as datas de criação.

Referências

[1] Júnior, Cleber Scoralick, *Estudo de Rótulos de Tempo em Sistemas NTFS Baseado em estrutura do Sistema de Arquivos e do Sistema Operacional*. Mestrado Profissional em Informática Forense e Segurança da Informação - UNB, 2012.

[2] CHOW, K. et al, *The rules of time on ntfs File system. Systematic Approaches to Digital Forensic Engineering, IEEE International Workshop on, IEEE Computer Society, Los Alamitos*, 2007

[3] BANG, J. et al, *Analysis of time information for digital investigation. Networked Computing and Advanced Information Management, International Conference on, IEEE Computer Society, Los Alamitos, CA*, 2009

[4] BANG, J.; YOO, B.; LEE, *S. Analysis of changes in File time attributes with File manipulation*, 2011

[5] TN1150, *HFS Plus Volume Format, http://developer.apple.com/*, Acessado em abril de 2015

[6] Mac OS X, *Mac OS Extended format (HFS Plus) volume and file limits, http://support.apple.com/*, Acessado em abril de 2015

[7] Wasim Ahmad Bhat , S. M. K. Quadri, *A Quick Review of On-Disk Layout of Some Popular Disk File Systems*, Global Journal of Computer Science & Technology, 2011

[8] Technical Note TN1150, *HFS Plus Dates, http://dubeiko.com/development/FileSystems/HFSPLUS/tn1150.html*, Acessada em 23/04/2015

[9] FTK Imager, *http://en.wikipedia.org/wiki/Forensic_Toolkit*, Acesso 27/05/2015

[10] HFSExplorer, *http://en.wikipedia.org/wiki/Forensic_Toolkit*, Acessado em 27/05/2015

[11] A Quick Review of On-Disk Layout of Some Popular Disk File Systems, *http://en.wikipedia.org/wiki/Forensic_Toolkit*, Acessado em 27/05/2015

DOI: 10.5769/C2015011 or http://dx.doi.org/10.5769/C2015011

Brasil e Ciberterrorismo: desafios para o Rio 2016

Bruna Toso de Alcântara

***Resumo*— Sendo o país que sediará os jogos olímpicos de 2016 o Brasil possui a responsabilidade de se preparar para que os mesmos se deem de forma segura em território nacional. Assim, questões como o Ciberterrorismo, ainda que pareçam longe da realidade brasileira, devem ser levadas em consideração. Desta forma o presente artigo apresenta os desafios para as quais o Brasil deve atentar em prol de proteger o país durante o grande evento em 2016.**

Palavras-Chave—Ciberterrorismo, Brasil, Rio 2016.

***Abstract*— Being the country that will host the Olympic Games in 2016, Brazil has the responsibility to prepare itself to guarantee the safety of the games within national territory. Thus, issues such as Cyberterrorism, even though appearing to be far from the Brazilian reality, should be taken into consideration. Therefore, this paper presents the challenges for which Brazil should pay attention for the sake of protecting the country while hosting the big event in 2016.**

Keywords—*Cyberterrorism, Brazil, Rio 2016.*

I. INTRODUÇÃO

Com as organizações terroristas utilizando-se cada vez mais do ciberespaço para espalhar sua narrativa e com o crescente entrelaçamento das Infraestruturas Críticas dos Estados aos meios tecnológicos, o temor do Ciberterrorismo ronda diversos países ao redor do mundo.

Contudo, sem uma acordada definição internacional sobre os efeitos e características do que venha a ser esse tipo de terrorismo, proteger-se do mesmo se torna uma tarefa difícil e trabalhosa, que envolve não somente preparação, mas estudos acerca de suas possíveis consequências.

Nesse sentido, em que pese o Brasil não tenha atos de terrorismo como uma realidade próxima, enquanto anfitrião dos jogos olímpicos em 2016, ele deve se preparar para possíveis ataques cibernéticos nesse formato. Afinal, se tratando do ciberespaço, não podemos assumir padrões e muito menos que certos países não serão atingidos direta ou indiretamente por ações de terceiros.

Assim, o presente artigo se divide em duas partes. A primeira visa elucidar como o país pode entender o termo "Ciberterrorismo" e como esse fenômeno preocupa os países a nível internacional.

Já a segunda parte pretende mostrar os desafios para os quais o Brasil, em específico, deve atentar nessa seara e quais as possíveis medidas que podem ajudar a contorná-los.

II. CIBERTERRORISMO: ENTENDIMENTO, FORMAS DE ATUAÇÃO E COMBATE.

Em que pese o termo Ciberterrorismo tenha aparecido pela primeira vez em um artigo de Barry Collin [1], nos anos 1980, significando o perigo de ataques conduzidos à longa distância (como consequência da interseção entre mundo físico e virtual) e tendo como alvos Infraestruturas Críticas de um país (fazendo com que a população de um país não conseguisse "comer, beber, se locomover, ou viver") ainda não existe uma definição internacional padrão para esse fenômeno. Assim, o debate acerca das ameaças que o Ciberterrorismo apresenta continua vivo e ganhando cada vez mais relevância.

De fato, em estudo conduzido há pouco tempo pela Swansea University [2] 36% dos entrevistados admitiram ser muito importante que tomadores de decisão *(policymakers*) tivessem uma resolução das questões de definição em torno de terrorismo, e 35% marcaram como **quase** essencial a necessidade de uma definição específica de Ciberterrorismo para os mesmos. Ademais, 87% dos entrevistados considerou como elemento característico do Ciberterrorismo a motivação política e ideológica.

Sob esse prisma, talvez, o que possa ser mantido para as análises do Ciberterrorismo seja os quatro motivos clássicos que impulsionam atividades terroristas: 1) terrorista com só um foco (ou seja, a motivação deles vem de um assunto em particular, como os direitos dos animais), 2) terroristas ideológicos (que usam da violência para promover sua ideologia politica, a qual se pauta nos extremos da direita ou da esquerda) 3) terroristas nacionalistas (os quais buscam independência de um dado Estado ou entrar de um Estado para outro por razões étnicas ou geográficas) e 4) terroristas politico religiosos (que podem se tornar letais dado que entendem suas ações como atos sob ordens divinas)[3]

Por outro lado, como Awan [4] explica, "dado que tipos de comportamento podem ser ligados a problemas e movimentos sociais, isso nos permite olhar para o Ciberterrorismo através das lentes da mudança social". Nesse sentido, pensar em motivos nos leva a tentar entender o Ciberterrorismo a nível social, enquanto parte das mudanças que vem ocorrendo desde a Revolução Informacional. Igualmente, se nos basearmos em uma concepção construtivista de mundo - para a qual 1) o mundo é constituído parcialmente por nossas crenças e ideias sobre ele 2) nosso conhecimento de mundo é socialmente construído e mantido, e 3) há uma importante dinâmica de interações entre o mundo das ideias e o das coisas de tal maneira que nossas ideias e realidades sociais são moldadas, reforçadas e impactadas uma na outra- o debate passa a se focar mais em como o conceito é construído e produzido

socialmente do que em sua definição com características *per se* [5]

De qualquer forma, parece que a nível prático, Weimann [6] é um autor que consegue elucidar de forma precisa o que vem ocorrendo atualmente [7]. Em outras palavras, ele explica que o uso de computadores feito por terroristas serve principalmente como um "facilitador de suas atividades, seja para propaganda, recrutamento, difamação de comunicações ou outros propósitos que não simplesmente o Ciberterrorismo". Ademais, Weimann [8] coloca em pauta que há uma confusão entre atividades de hacktivismo e Ciberterrorismo, fazendo com que atos menores tomem proporções maiores através da mídia. Todavia, Weimann ressalta que [...] mesmo assim o hacktivismo realça a ameaça do Ciberterrorismo [...]" uma vez que os terroristas podem se utilizar dos caminhos já trilhados pelos hacktivistas, mas daí para alcançar seus próprios propósitos de atingir governos. Ademais, segundo o autor, zonas cinzentas podem existir entre essas duas modalidades no ciberespaço, se os terroristas forem capazes de recrutar ou contratar hacktivistas ou se hacktivistas decidirem ir mais além e operar a nível de Infraestruturas Críticas.

Nesse sentido, o uso da internet por grupos terroristas deve ser levado em consideração e analisado de forma profunda, abrangendo principalmente análises sobre a Darknet ou Deep Web, uma vez que 99.8% das atividades terroristas ocorrem nesse (sub) mundo do ciberespaço [9].

Em realidade, se faz necessário entender primeiramente porque os terroristas estão se interessando pelo uso de computadores. Assim em linhas gerais podemos elencar: 1)oferece abrangência para espalhar sua narrativa de forma rápida e barata, atingindo o maior número possível de recrutas, 2) forma fácil de manter o anonimato dos participantes e a troca de informações, 3) facilidade de acesso a dados abertos que podem ser úteis aos planos terroristas 4) facilidade para obter financiamento, principalmente através do uso de entidades de caridade como fachada [10].

Em segundo lugar, dentro desses pontos elencados, o que tange o recrutamento e abrangência para a narrativa se torna potencialmente mais perigosos, tendo em vista que, segundo uma pesquisa feita pela RAND Corporation em 2013, a internet pode aumentar as oportunidades para a radicalização. Contudo não necessariamente acelerando o processo e não substituindo o contato físico necessário para a própria radicalização (*self-radicalisation*)[11].Como exemplo recente dessa atividade podemos citar o caso do grupo ISIS (Estado Islâmico), o qual com sua campanha nas redes sociais atraiu para sua causa mais de 18.000 combatentes estrangeiros de mais de 90 países[12] entre eles o Brasil [13].

Tomando o exemplo do ISIS percebe-se que o uso da internet por terroristas aumenta o fenômeno, já discutido mundialmente, do recrutamento de combatentes estrangeiros (FTF, em inglês), ou seja, pessoas que viajam para um estado diferente do seu com o objetivo de se juntarem ou receberem treinamento em apoio a atividades terroristas [14]. De fato de acordo com o Fórum Global Contra o Terrorismo (GCTF, em inglês), órgão criado em 2011 por 29 países e a União Europeia (incluindo a Colômbia como membro fundador e a Organização dos Estados Americanos como um *stakeholder*), um grupo de trabalho especifico nesse fenômeno já tem lugar e usa como base o documento chamado "Memorando de Marrakesh em Boas Práticas para uma Resposta mais Efetiva ao fenômeno do FTF".

O Memorando de Marrakesh não é compulsivo perante as leis internacionais e se apresenta dividido em quatro grandes grupos, que se subdividem em um total de 19 boas práticas. Esses grandes grupos foram intitulados: detecção e intervenção contra a violência extremista; detecção e intervenção no recrutamento facilitação; detecção e intervenção contra viagem e combate; e detecção e intervenção sobre retornos. [15] Tendo como intuito principal coletar e difundir boas práticas entre uma variedade de países para combater o FTF [16]

Além disso, a RES 2178 (2014) do Conselho de Segurança da ONU também aborda o assunto, "[...] expressando grave preocupação que os combatentes terroristas estrangeiros estão usando sua ideologia extrema para promover o terrorismo". [17]

Diante desse quadro, é interessante a proposta de Ginkel [18] em medidas repressivas em duas categorias: duras e suaves. As primeiras seriam

> [...] focadas na negação do acesso as narrativas extremistas dos grupos e apoiadores terroristas, através do bloqueio de mensagens ou a retirada do ar de websites, e da proibição de distribuição e comunicação de conteúdo radical, além de repressão criminal para as pessoas por trás de tais ações.

Contudo, como alerta Ginkel, e corroboramos com a autora, implementar tais medidas em países com uma democracia já estabelecida podem ser contraproducentes e mesmo que as páginas sejam "desligadas" de forma efetiva, outras , cada vez mais escondidas e de difícil acesso surgirão.

Já as medidas suaves seriam mais ao nível de contrapropagandas, podendo se dar na forma de campanhas publicas (a exemplo do Centro para Comunicações Estratégicas Contra o Terrorismo, nos Estados Unidos); narrativas alternativas (a exemplo do Radical Middle Way, do Reino Unido, como uma tentativa de diálogo com grupos mulçumanos sobre o papel da religião no século XXI); contra narrativas (a exemplo do programa "Say no to Terror" a qual se utiliza de vários mecanismos para criar uma contra narrativa sobre elementos selecionados da narrativa terrorista).

Por fim haveria ainda o papel da mídia convencional, que como Ginkel coloca, é um tema controverso, uma vez que depende dos meios convencionais de comunicação deixar-se ou não influenciar pela narrativa terrorista.

Como último ponto de medidas no combate ao uso terrorista da internet, e da prática do Ciberterrorismo (enquanto veículo de radicalização e recrutamento) vale a pena lembrar o compromisso, também não compulsório legalmente, da NETMundial, que ocorreu em Abril de 2014 no Brasil, cujo documento final menciona a questão da segurança da internet, de forma mais geral, baseada em uma "forte cooperação entre os diferentes *stakeholders*"[19]. Ainda, exemplos que podem inspirar o Brasil podem basear-se em países individualmente, como Estados Unidos e Reino Unido, que possuem estratégias contra o Ciberterrorismo, uma vez

que ambos sofreram consequências físicas da distribuição livre da narrativa terrorista[1].

Diante disso, o que se torna claro é que independente do que for realmente o Ciberterrorismo, e de como evoluirá o debate conceitual sobre ele, um país hoje deve atentar para 1) atividades hacktivistas em suas redes, uma vez que elas podem ser o "caminho das pedras" para ataques mais perigosos 2)abrangência da narrativa terrorista, a qual incita a radicalização e pode proporcionar a participação de estrangeiros em causas terroristas e 3) medidas contra a narrativa terrorista que não causem pânico à população nem prejudiquem o abertura democrática da internet.

Tendo isso em mente, na próxima seção adentramos para as especificidades de um país que não vê o terrorismo como uma realidade próxima e assim, consequentemente, tem que superar alguns desafios se quiser proporcionar um nível de segurança excelente durante o Rio 2016.

III. Desafios para o Brasil

O Brasil já sediou de forma exitosa os Jogos Mundiais Militares em 2011; a Rio+20 em 2012; a Jornada Mundial da Juventude, com a presença de Sua Santidade o Papa Francisco, a Copa das Confederações em 2013 e, mais recentemente, a Copa do Mundo em 2014. Contudo, em que pese à experiência que o país adquiriu quanto à segurança em grandes eventos, algumas questões ainda merecem atenção para o Rio16, sendo uma delas o terrorismo, e mais especificamente o Ciberterrorismo.

Uma vez que o país está em evidência internacional, como anfitrião de grandes eventos, e dada sua extensão assegurar a segurança das Infraestruturas Críticas e das conexões cibernéticas que envolvem o território nacional se torna primordial. Afinal, mesmo que o Brasil não tenha um histórico de atos terroristas, sempre se deve lembrar que grandes eventos se tornam por si mesmos uma "grande vitrine"[20] para atuações terroristas, sejam elas oriundas de organizações específicas ou de lobos solitários.

Assim, com base nas recentes discussões acadêmicas, políticas e militares foram identificadas quatro grandes áreas as quais o Brasil precisa atentar no que tange ao terrorismo, e consequentemente a sua vertente cibernética, para encerrar com o devido prestígio esse ciclo de grandes eventos. Estes seriam: 1) Necessidade de uma legislação interna, a qual tipifique o que é terrorismo para o Brasil; 2) a construção de uma cultura de segurança nessa seara, a nível real e virtual; 3) o fato de conseguir assegurar a capilaridade das comunicações eletrônicas dentro e fora do território nacional; e 4) a necessidade do fortalecimento das instituições responsáveis pela tomada de ações antiterroristas.

A. Legislação interna

O Brasil não conseguiu até o presente momento chegar a um consenso sobre o que seria um ato terrorista, e por isso não possui tipificação a nível penal sobre o assunto. Assim, como aponta André Luís Woloszyn, especialista em segurança, defesa e inteligência e analista de assuntos estratégicos [21] há a existência de um paradoxo frente à atitude brasileira, ou seja, ao mesmo tempo em que o país repudia internacionalmente e considera o terrorismo um crime a nível interno não há uma tipificação frente ao mesmo.

De fato a nível internacional

> Brasil tem apoiado as decisões da AGNU e do CSNU contra o terrorismo, sendo parte das doze principais convenções no âmbito das Nações Unidas, das quais **nove já foram internalizadas e ratificadas**. No âmbito regional, o Brasil é signatário das três convenções da Organização dos Estados Americanos (OEA) relacionadas ao terrorismo. [22]

Já internamente, a Constituição da República Federativa do Brasil de 1988 traz como preceito fundamental o repúdio ao terrorismo (artigo 4º, inciso VIII), complementado pelo artigo 5° (inciso XLIII) o qual declara o terrorismo como crime inafiançável e insuscetível de graça ou anistia**.** [23]Também possuímos algumas leis que tratam do assunto (Lei de Crimes Hediondos-Lei nº 8.072, de 1990- e Lei de Segurança Nacional-Lei nº 7.170, de 1983).Contudo, como explica Woloszyn [24] elas se tornam "inócuas" uma vez que ferem tanto o principio da objetividade jurídica (a qual exige a definição clara e precisa das ações constituidoras dos tipos penais [25]) quanto o princípio constitucional da reserva legal (o qual atesta que não há crime sem que haja lei anterior que o defina)[26].

Lasmar [27] aponta alguns argumentos, que podem explicar a formação do paradoxo explicitado por Woloszyn. Estes seriam: 1) qualquer tratamento da questão do terrorismo poderia estigmatizar a população muçulmana brasileira, 2) o reconhecimento da existência de atividades terroristas em território brasileiro poderia afetar o turismo internacional no Brasil 3) existência de um corpo normativo de combate ao terrorismo ou o reconhecimento de sua existência levariam a uma construção de uma imagem de alinhamento brasileiro com a política externa estadunidense da "Guerra Global Contra o Terror" e isso poderia ser visto como uma política externa e interna provocativa, que poderia atrair problemas políticos e de segurança para o Brasil 4) existência do temor de que grupos de movimentos sociais legítimos venham a ser taxados de grupos terroristas e 5) envolvimento de vários políticos da alta cúpula governamental em atividades ou grupos que se utilizaram da violência política durante a ditadura militar brasileira a fim de combatê-la.

Quanto ao primeiro e quarto argumento, uma possibilidade seria tentar fazer uma processo mais aberto e democrático quanto ao texto jurídico da dita tipificação. Talvez através de uma plataforma online, de modo que tanto a sociedade mulçumana quanto grupos de movimento social legítimos sejam abarcados.

O terceiro argumento parece um pouco deslocado. Afinal como Embaixador Samuel Pinheiro Guimarães colocou, em um Seminário recente sobre o terrorismo:

> Enquanto o Brasil mantiver internamente um convívio pacífico, harmonioso entre as

[1] Quando se fala em consequências físicas de Ciberterrorismo me refiro ao caso dos Irmãos Tsarnaev, que plantaram bombas na maratona de Boston em Abril de 2013 e ao estudante universitário Roshnara Choudry que esfaqueou um membro do Parlamento britânico com uma faca em 2010. Em ambos os casos a inspiração para instauração do terror (radicalização) teve origem em conteúdo online. [12]

> diferentes comunidades, de um lado; e, no sistema internacional, ter posições que façam com que ele continue na linha de defesa da paz, de defesa do desarmamento, de repúdio ao terrorismo, de solução pacífica das controvérsias, nós estaremos criando as principais condições para evitar que o Brasil seja incluído no rol das nações que são objeto de eventuais atentados terroristas. [28]

Por fim, quanto ao quinto argumento, infelizmente essa é uma realidade na história brasileira e deve ser superada socialmente, através de mecanismos como a Comissão da Verdade.

Portanto de maneira geral, constata-se a necessidade de uma tipificação do terrorismo na legislação interna, lembrando a igual necessidade, de um debate no corpo jurídico que se desenvolverá para a vertente cibernética. Fazendo a ressalva de que ao se analisar a vertente cibernética o melhor caminho seria a não direção aos extremos. Em outras palavras, não se pode tipificar o Ciberterrorismo enquanto proporções de uma guerra cibernética, tendo em vista que ainda não se sabe ao certo o que o mesmo significa ou como se desenvolverá. A única certeza que temos é que podemos tipifica-lo enquanto uma forte ferramenta de influência para atos terroristas, mesmo assim, o debate deve ser levado para a sociedade, atingindo principalmente os grupos mais vulneráveis a essa influência online.

B. Construção de uma cultura de segurança real e no ciberespaço

Tendo em vista que o Brasil, não possui um histórico contendo atos terroristas, esse é um aspecto que não está na cultura brasileira. Como explicado por Salaberry [29]

> A nossa cultura é de os homens colocarem a carteira no bolso da frente e as mulheres a bolsa para frente quando andam em um lugar que não conhecem. Mas alguém tem medo de passar ao lado de um cesto de lixo?

Assim, existe a necessidade de trabalhar a percepção da população acerca do terrorismo, desmistifica-lo como distante, e improvável de ocorrer no país – principalmente quando se trata de sua vertente no ciberespaço. Afinal, se 2016 será o fim do ciclo de grandes eventos ele pode ser também o início de uma cultura de segurança, com o cuidado de que ela não se "torne paranoia, ao ponto de desrespeitar os direitos individuais, os direitos civis, os direitos humanos" [30].

Em especifico ao Ciberterrorismo, aspectos como higiene cibernética, e conscientização via palestra e *workshops* aparecem como possibilidades. Aumentar sensibilização da população se torna uma ferramenta essencial.

Contudo a divulgação dessas iniciativas deve ser mais massificada e atingir principalmente a população mais vulnerável (adolescentes, imigrantes de zonas de conflito e a comunidade israelita e mulçumana) A referência à população mais vulnerável se dá uma vez que já estamos vivenciando um problema de valores, que perseguirá muito a próxima geração.

Esse problema de valores se reflete na sociedade desde os ataques de 11 de Setembro de 2001 e se consolida no aumento de partidos de extrema – tanto direita quanto esquerda – ao redor do mundo. Por isso, a necessidade de abordar adolescentes, imigrantes de zonas de conflito e a comunidade israelita e mulçumana brasileiras.

Ademais, o uso da contra narrativa frente às informações terroristas disponíveis se torna uma alternativa. Mas como todo assunto sensível, essa narrativa deve ser ponderada em sua forma de abrangência, e nesse ponto a cooperação internacional pode vir a ajudar. Experiências de países que já lidam com o assunto podem ajudar o Brasil a traçar um perfil do que fazer e, principalmente, do que não fazer.

C. Assegurar a capilaridade das comunicações eletrônicas dentro e fora do território nacional

O fato de o Brasil ser um país "gigante pela própria natureza" não só apresenta uma dificuldade quanto a possíveis atos de terrorismo físico como também no âmbito cibernético. Afinal, mesmo que durante os grandes eventos recentes se tenha utilizado da tecnologia para manter a capilaridade da atuação militar [31], esse mesmo uso traz vulnerabilidades para a manutenção de comunicações efetivas e seguras.

Nesse ponto, ainda que haja um engajamento integrado de órgãos de inteligência, segurança e defesa [32] existe a necessidade de reforçar linhas de comunicação. Em outras palavras, se a comunicação entre as unidades de defesa, segurança ou inteligência forem cortadas (total ou parcialmente) e supondo hipoteticamente um fim terrorista por trás dessa ação, criar estratégias alternativas de comunicação ou mesmo um sistema de *backup* que restaure rapidamente as conexões, se torna imperativo.

Devemos lembrar que priorizar a capilaridade em meio a um território tão extenso é um assunto importante e sério, ainda mais dado o fluxo de pessoas que grandes eventos como os jogos olímpicos podem proporcionar. Lugares de difícil comunicação e acesso também devem ser levados em consideração, e adequados com infraestrutura ou pessoal qualificado.

Não só os canais de comunicação eletrônica internos devem ser protegidos, como também as vias de tráfego de informações com a INTERPOL[2], AMERIPOL[3] e MERCOSUL [33] devem ser asseguradas, para que não haja falseamento de informações tampouco para que haja a falta delas em uma situação de emergência.

Por fim, mapear caminhos deixados pelos hackers e a partir daí perceber até onde a narrativa terrorista está chegando se torna também imprescindível. Afinal com cinco dos dez grupos de hackers mais ativos do mundo [34], o Brasil pode, sem perceber, ter mais vias cibernéticas de acesso a seu território do que se imagina.

D. Fortalecer instituições responsáveis

Juntamente com a construção de uma cultura de segurança é necessário que recursos materiais, tecnológicos e principalmente humanos atinjam os órgãos responsáveis pela proteção da esfera cibernética do país.

O engajamento com outros setores da sociedade [35] e a instrução oriunda de países estrangeiros [36] são partes da

[2] INTERPOL, em português significa "Organização Internacional de Polícia Criminal".

[3] AMERIPOL em português significa: "Comunidade de Polícia da América"

solução, mas a necessidade de aumentar proporcionalmente esses esforços é existente.

A cultura de segurança a ser construída deve levar em consideração o trabalho já desempenhado para a proteção do país pelas instituições responsáveis. Não me refiro a explicitar métodos, mas uma boa divulgação de casos exitosos poderia contribuir para o entendimento da necessidade de investimentos na área de segurança, em especial na cibernética.

Outro ponto é que o Ciberterrorismo em específico ganhe um lugar no debate com a sociedade, talvez através da grade curricular da Escola Nacional de Defesa Cibernética [37]. Nesse sentido, elaborar um plano de carreira para manter o pessoal especializado dentro do setor público é fundamental, assim podemos criar um banco de profissionais brasileiros trabalhando efetivamente dentro do território nacional.

Por fim, vale a ressalva de que no âmbito do ciberespaço a máxima de "quanto mais nacional melhor" é a que vale atualmente. Assim, investimentos são necessários, a percepção desses investimentos é imprescindível e a conduta ética dos profissionais em um ambiente tão aberto é fundamental.

IV. CONCLUSÕES

O Brasil não é nenhum país inexperiente no que tange a organização e segurança de grandes eventos. Todavia isso não pode se tornar sinônimo de uma conduta leviana com a segurança nacional.

Tendo isso em mente e partindo especificamente para a questão do terrorismo cibernético, existem áreas de preocupação nacional que devem merecer atenção se quisermos que o país encerre seu ciclo de evidência - enquanto país-sede de grandes eventos- com "chave-de- ouro".

Assim, o Brasil deve atentar para quatro grandes áreas antes que os jogos olímpicos aconteçam em 2016. Essas áreas seriam: 1) Necessidade de uma legislação interna, a qual tipifique o que é terrorismo para o Brasil; 2) a construção de uma cultura de segurança voltada para a cibernética, ao nível de implicações reais e virtuais; 3) o fato de conseguir assegurar a capilaridade das comunicações eletrônicas dentro e fora do território nacional; e 4) a necessidade do fortalecimento das instituições responsáveis pela tomada de ações antiterroristas.

Cada uma dessas áreas possui especificidades e compõem, em parte, preocupações de ordem internacionais (como por exemplo, o manejo de contra narrativas e uma tipificação do terrorismo, e de sua vertente cibernética). Se formos capazes de supri-las, isso não só garantirá uma boa administração dos jogos Rio2016, como também poderia nos ajudar na construção de uma doutrina de segurança que equilibre os anseios de um estado de direito democrático com medidas securitárias apropriadas.

REFERÊNCIAS

[1] Collin B. C (1997) The future of cyberterrorism. *Crime & Justice International* 13 (2). Disponível em: http://www.cjimagazine.com/archives/cji4c18.html?id=415 Acesso em 19 de Abril de 2015. (tradução nossa)

[2] Macdonald, S., Jarvis, L., Chen, T. & Lavis, S. (2013). *Cyberterrorism: A Survey of Researchers*. Cyberterrorism Project Research Report (No. 1), Swansea University. Disponível em: www.cyberterrorism-project.org Acesso em 17 de Abril de 2015. (tradução nossa)

[3] Awan, I.(2014) Debating the term cyber-terrorism: Issues and problems, *Internet Journal of Criminology.ISSN 2045-6743 [online]* . Disponível em: http://www.internetjournalofcriminology.com/ Acesso em 19 de Abril de 2015 p.02 (tradução nossa)

[4] Charvat, J P I A G. *Cyber Terrorism: A New Dimension in Battlespace*. Disponível em: https://ccdcoe.org/publications/virtualbattlefield/05_CHARVAT_Cyber%20Terrorism.pdf.Acesso em 19 de Abril de 2015. P.02 (tradução nossa)

[5] Jarvis, L., Nouri, L. & Whiting, A. (2014) 'Understanding, Locating and Constructing Cyberterrorism'. In: Chen, T., Jarvis, L. & Macdonald, S. (eds) (2014) *Cyberterrorism: Understanding, Assessment and Response* (New York: Springer) Disponível em: http://www.springer.com/computer/information+systems+and+applications/book/978-1-4939-0961-2 Acesso em: 19 de Abril de 2015. p.34-35 (tradução nossa)

[6] Weimann, G. (2005) The sum of all fears? *Studies in Conflict & Terrorism,* 28 (2). p.133

[7] Weimann, G (2004) *Us Institute of Peace December*. Special Report 119 [online] Disponível em: http://webcache.googleusercontent.com/search?q=cache:W8chjRrx0AMJ:www.usip.org/sites/default/files/sr119.pdf+&cd=1&hl=pt-BR&ct=clnk&gl=br Acesso em 19 de abril de 2015.p.05 (tradução nossa)

[8] Para entendimento maior sobre a preferencia do uso do autor ver: Awan, I.(2014) Debating the term cyber-terrorism: Issues and problems, *Internet Journal of Criminology.ISSN 2045-6743 [online]*

[9] Algemene Inlichtingen en Veiligheidsdienst (AIVD), (2012) *Jihadism on the Web: A Breeding Ground for Jihad in the Modern Age*. Disponível em: https://www.aivd.nl/english/publications-press/@2873/jihadism-web/ Acesso em 19 de Abril de 2015. p.05 (tradução nossa)

[10] Weimann, G. (2005) The sum of all fears? *Studies in Conflict & Terrorism,* 28 (2); Ginkel, B. van. *Responding to Cyber Jihad: Towards an Effective Counter Narrative*. International Center Counter Terrorism (ICCT) Research Paper March 2015. Disponível em: http://www.clingendael.nl/publication/responding-cyber-jihad-towards-effective-counter-narrative Acesso em 19 de Abril de 2015. (tradução nossa)

[11] Behr, I. von; Reding, A; Edwards, C. Luke G. (2013) *Radicalization in the Digital Era*. Research reports RAND Corporation. Disponível em: http://www.rand.org/pubs/research_reports/RR453.html Acesso em 19 de Abril de 2015 (tradução nossa)

[12] Liang, C.S. (2015) *Cyber Jihad: Understanding and Countering Islamic State Propaganda*. Geneva Center of security Policy (GCSP) Policy Paper 2015/2. Disponível em: http://www.gcsp.ch/Emerging-Security-Challenges/Publications/GCSP-Publications/Policy-Papers/Cyber-Jihad-Understanding-and-Countering-Islamic-State-Propaganda Acesso em 19 de Abril de 2015. P.02 (tradução nossa)

[13] Canthanéde, E; Matais A.(2015) Governo detecta recrutamento de jovens pelo Estado Islâmico. *Estadão [online]* Disponível em: http://internacional.estadao.com.br/noticias/geral,governo-detecta-recrutamento-de-jovens-pelo-estado-islamico,1655354 Acesso em 19 de Abril de 2015.

[14] Global CounterTerrorism Forum. The Hague-*Marrakech Memorandum on Good Practices for a More Effective Response to the FTF Phenomenon*. Disponível em: https://www.thegctf.org/web/guest/foreign-terrorist-fighters;jsessionid=2C659E5ECACBD47050353F462BA883CF.w142 Acesso em 19 de Abril de 2015.p.01(tradução nossa)

[15] Ibid.

[16] Ibid

[17] S/RES/2178 (2014) p.02 (tradução nossa)

[18] Ginkel. Op. cit. p.04 (tradução nossa)

[19] *NETMundial.Multistakeholder Statement* (2014) Disponível em http://netmundial.br/netmundial-multistakeholder-statement/ Acesso em 19 de Abril de 2015. P.05

[20] Campos, A. J. de *Terrorismo e Grandes Eventos*. In: Seminário Internacional Terrorismo e Grandes Eventos (2013: Brasília, DF).Terrorismo e grandes eventos [recurso eletrônico].Câmara dos Deputados, Comissão de Relações Exteriores e de Defesa Nacional. Brasília: Câmara dos Deputados, Edições Câmara, 2014. P.58

[21] Woloszyn, A. L *O Terrorismo do Século 21 e a Democracia*. In: Seminário Internacional Terrorismo e Grandes Eventos (2013: Brasília, DF).Terrorismo e grandes eventos [recurso eletrônico].Câmara dos Deputados, Comissão de Relações Exteriores e de Defesa Nacional. Brasília: Câmara dos Deputados, Edições Câmara, 2014. P.34 (grifo nosso)

[22] Mariani, C.B. *Como o Brasil está inserido no combate internacional ao terrorismo?* Disponível em: http://relacoesinternacionais.com.br/politica-externa/como-o-brasil-esta-inserido-no-combate-internacional-ao-terrorismo/ Acesso em 22 de Abril de 2015 (grifo nosso)

[23] BRASIL. Constituição (1988). *Constituição da República Federativa do Brasil:* texto constitucional promulgado em 5 de outubro de 1988, com as alterações adotadas pelas Emendas Constitucionais nº 1/92 a 67/2010 e pelas Emendas Constitucionais de Revisão nº 1 a 6/94. Brasília, DF: Senado Federal, Subsecretaria de Edições Técnicas, 2011

[24] Woloszyn Op. cit.

[25] Mesquita, L.E.G. *O Terrorismo e a sua probabilidade de ocorrência no Brasil.*(2012) Trabalho de Conclusão de Curso. Escola Superior de Guerra. Rio de Janeiro 2012. P.37

[26] Woloszyn Op. cit.

[27] Lasmar, J.M. A legislação brasileira de combate e prevenção do terrorismo quatorze anos após 11 de Setembro: limites, falhas e reflexões para o futuro. *Revista de Sociologia Política*, v. 23, n. 53, p. 47-70, mar. 2015.

[28] Guimarães, S.P. *Terrorismo e Grandes Eventos*. In: Seminário Internacional Terrorismo e Grandes Eventos (2013: Brasília, DF).Terrorismo e grandes eventos [recurso eletrônico].Câmara dos Deputados, Comissão de Relações Exteriores e de Defesa Nacional. Brasília: Câmara dos Deputados, Edições Câmara, 2014. p.71

[29] Salaberry, L.A.S. *Terrorismo e Grandes Eventos*. In: Seminário Internacional Terrorismo e Grandes Eventos (2013: Brasília, DF).Terrorismo e grandes eventos [recurso eletrônico].Câmara dos Deputados, Comissão de Relações Exteriores e de Defesa Nacional. Brasília: Câmara dos Deputados, Edições Câmara, 2014. p.64

[30] Daher, D. *Terrorismo e Grandes Eventos* In: Seminário Internacional Terrorismo e Grandes Eventos (2013: Brasília, DF).Terrorismo e grandes eventos [recurso eletrônico].Câmara dos Deputados, Comissão de Relações Exteriores e de Defesa Nacional. Brasília: Câmara dos Deputados, Edições Câmara, 2014. P.78

[31] Arruda, J.C. *Terrorismo e Grandes Eventos* In: Seminário Internacional Terrorismo e Grandes Eventos (2013: Brasília, DF).Terrorismo e grandes eventos [recurso eletrônico].Câmara dos Deputados, Comissão de Relações Exteriores e de Defesa Nacional. Brasília: Câmara dos Deputados, Edições Câmara, 2014. P. 70

[32] Daher Op. cit. p.76

[33] Ibid. p.77

[34] Raposo, A. C. Terrorismo e Contraterrorismo: desafio do século XXI**.** *Revista Brasileira de Inteligência*. Brasília: Abin, v. 3, n. 4, set. 2007. p. 46

[35] Salaberry, Op. cit. p.65

[36] Arruda, Op. cit. p.72

[37] Matsuura, S. Brasil terá Escola Nacional de Defesa Cibernética. *O Globo [online*] disponível em http://oglobo.globo.com/sociedade/tecnologia/brasil-tera-escola-nacional-de-defesa-cibernetica-15914957 Acesso em 22 de Abril de 2015.

:

Bruna Toso de Alcântara, graduanda de Relações Internacionais da Universidade Federal de Santa Maria, Santa Maria-RS, Brasil, E-mail: brunatoso@hotmail.com

DOI: 10.5769/C2015012 or http://dx.doi.org/10.5769/C2015012

Um Levantamento sobre o Mercado de Exploração de Vulnerabilidades do Espaço Cibernético

Robson de Oliveira Albuquerque, Rafael Timóteo de Sousa Júnior e João Paulo C. Lustosa da Costa

Departamento de Engenharia Elétrica, Universidade de Brasília (UnB)
Campus Universitário Darcy Ribeiro – Asa Norte – 70910-900 – Brasília-DF – Brasil

robson@redes.unb.br, desousa@unb.br, joaopaulo.dacosta@ene.unb.br

***Resumo*— O espaço cibernético é a base de sustentação de múltiplos setores da economia, constituindo-se como fonte de geração de recursos e capitais, bem como a projeção de poder de vários Estados. Logo, se tornou comum explorar vulnerabilidades da informação neste ambiente. As ferramentas de exploração de vulnerabilidades encontram-se no centro de um processo de produção e comercialização que, com seus diversos atores, constituem um mercado a parte. É importante para o profissional de segurança da informação e combate ao cibercrime ter conhecimento desse mercado específico, no sentido de organizar proteções de sistemas de informação, e também realizar investigações de evidências do cibercrime. Neste artigo, apresentam-se as definições dos elementos básicos desse mercado, além de um levantamento de algumas das principais empresas participantes. As conclusões apontam a necessidade de forte planejamento para correta atuação nessa área.**

***Abstract* – Cyberspace is now essential to support multiple economy sectors, as it constitutes a source of resources and wealth, as well as representing the projection of power from multiple nations. Due to its characteristics, cyberspace has become a place where the exploitation of information vulnerabilities occurs continuously. Tools for vulnerability exploitation are in the middle of a production and trade process, which, with a variety of actors, form a specialized market, dealing with large amounts of money. For law and enforcement agencies that fights cybercrime it is very important to have knowledge of such market, not only to protect information systems, but to perform cybercrime investigation and forensics activities. In this paper, definitions are given for the basic elements that take part in this market and relevant concepts related to exploits, as well as a review of enterprises that operate in this market. The conclusions point that strategic planning is a critical requirement to approach this cyber security area.**

***Keywords* – Security, Vulnerabilities, Exploits, Exploit and Vulnerability Market.**

I. INTRODUÇÃO

No cenário atual, a Internet que integra o espaço cibernético não pode ser vista apenas como mais uma rede de computadores mundial, mas sim a base de sustentação de múltiplos setores da economia como, por exemplo, hardware, software, telecomunicações, redes sociais, sistemas financeiros, moedas virtuais, serviços de indexação, armazenamento massivo e jogos online. Em diversos aspectos o espaço cibernético é a fonte de geração de recursos e capitais, bem como a projeção de poder de vários Estados, através do controle de recursos tecnológicos que compõem tal espaço.

O espaço cibernético está em franca evolução, em função do desenvolvimento ininterrupto de novas tecnologias e consequente desenvolvimento de soluções que têm o objetivo de atender às demandas crescentes do mercado. Este processo evolutivo gera um volume cada vez maior de informações estratégicas, corporativas e pessoais e que, idealmente, deveriam ser acessadas apenas por aqueles que teriam a legitimidade de tratá-las. Entretanto, dependendo do valor da informação que estiver sendo tratada, outros integrantes desse ecossistema, que não deveriam ter acesso à informação, têm não só interesse em acessá-la, mas também de copiá-la, modificá-la e, eventualmente, destruí-la.

Nesse espaço cibernético, está inserida uma área particular e específica de segurança da informação, área esta que trata de exploração de vulnerabilidades e de falhas de segurança. As explorações são realizadas das mais variadas formas e com os mais variados tipos de meios e tecnologias. Normalmente são realizadas com o intuito de se obter vantagem estratégica e competitiva, e também para a obtenção de lucros.

Este ambiente de exploração de falhas cibernéticas tem três raízes principais: o crime organizado, atividades hacktivistas e as ações de nações-estado [1], [2], [3]. De comum, os três fazem uso de técnicas de exploração de falhas e desenvolvem de mecanismos técnicos de garantia de obtenção do dado ou da informação desejada para posterior uso. Para a consecução do objetivo relacionado ao acesso à informação privilegiada, um atacante ou intruso utiliza as mais variadas técnicas e processos, mas é possível destacar o emprego de ferramentas de software projetadas especificamente para tirar proveito de alguma falha em um sistema computacional, normalmente para fins danosos como o de instalar um código malicioso, em inglês, *malware*.

Tais tipos de ferramentas, formalmente denominadas de *exploits* em inglês, vêm sendo objeto de todo um mercado de desenvolvimento, comercialização, manutenção e suporte, voltado para o crime cibernético (*cybercrime*). Sob a ótica deste cenário, o mercado de *exploits* é uma atividade complexa, de alto nível técnico e profissional, amplamente competitiva entre os atores envolvidos, e altamente lucrativa sob o aspecto financeiro e estratégico, no sentido de obtenção de algum tipo de vantagem.

Em função desse contexto, o objetivo deste artigo é o de demonstrar um levantamento da situação desse mercado, em especial do ponto de vista daqueles serviços e produtos que se encontram disponíveis para utilização em diversas situações.

Os dados coligidos são importantes para que o profissional de segurança da informação e combate ao *cybercrime* tenha conhecimento desse mercado específico, no sentido de organizar proteções de sistemas de informação, e de realizar investigações tanto para prevenção de crimes cibernéticos quanto para a investigação de evidências da ocorrência desses crimes. Além disso, o artigo aponta a necessidade de planejamento focado e com objetivos bem definidos para a inserção nesse tipo de atividade.

De maneira geral, este artigo está organizado nas seguintes divisões. A Seção II explica o que são *exploits* e temas correlatos, assim como o processo básico de emprego dessas ferramentas. Na Seção III tem-se uma discussão que trata de como o mercado de *exploits* está estruturado. Já a Seção IV apresenta alguns dos principais atores envolvidos. Por fim, as conclusões do trabalho encontram-se na Seção V.

II. DEFINIÇÕES E TERMINOLOGIA

O mercado de *exploits* pode ser analisado em uma perspectiva de consumidor (cliente) e fornecedor (especialista/empresa). Indiferente da perspectiva, para se abordar esse mercado é necessário planejamento, foco e objetivo concreto. É um mercado profissional que conta com desenvolvimento de técnicas e estudo de tecnologias em constante evolução, empregando pessoal técnico altamente capacitado.

A participação neste mercado como eventual gerador de necessidade não prescinde de um planejamento bem estruturado, de preferência com suporte e objetivo estratégico, que conte com infraestrutura própria e ferramentas específicas. Por consequente, tal participação requer recursos financeiros e técnicos para implementação de laboratórios avançados para testes de funcionalidades e técnicas de exploração, já que para se atender a uma necessidade, um objetivo claramente definido é fundamental.

Além disso, estes tipos de recursos técnicos e financeiros são aplicados de diversas formas. Por exemplo, para permitir a replicação das características técnicas e de ambiente de um alvo específico para testes de efetividade e exploração de uma oportunidade real. Note-se que esta atividade deve ser desenvolvida por pessoal qualificado na área de segurança cibernética. Esta área envolve diversas tecnologias e arquiteturas, tornado a atividade claramente multidisciplinar com viés técnico e especializado.

Assim sendo, podem ser considerados como requisitos mínimos para atuação neste tipo de mercado, o domínio e entendimento de tecnologias relacionadas a arquiteturas de hardware dos mais variados tipos, arquiteturas e plataformas de software para avaliação, desenvolvimento e testes funcionais, bem como domínio avançado de tecnologias e protocolos de redes de comunicação, sistemas operacionais, equipamentos de redes de comunicação, serviços de telecomunicação, entre outros requisitos.

Isso posto, um dos fundamentos básicos deste mercado é o *exploit* em si. Sob o aspecto de segurança, ele é o ponto de partida visando uma necessidade específica de emprego de recursos técnicos ligados a um objetivo concreto. É o *exploit* que oferece as condições mínimas de sucesso na exploração de falhas. Entretanto, para que o *exploit* obtenha sucesso, diversas etapas precisam ser entendidas corretamente.

Os tópicos a seguir detalham alguns aspectos de forma a explicar o que são vulnerabilidades *Zero-day*, o que são *exploits* e o processo básico de emprego dos mesmos. Para fins deste artigo, uma vulnerabilidade é um tipo de falha que permite ao atacante obter sucesso na exploração de um recurso tecnológico, seja ele na forma de um hardware ou de um software.

A. VULNERABILIDADES ZERO-DAY

Sob o aspecto de segurança da informação, uma vulnerabilidade *Zero-day* é um tipo de falha para a qual não existe defesa prévia eficiente devido ao fato de não existir uma correção ou uma atualização que a remova do produto sendo explorado. Isso permite a um atacante, considerando o seu objetivo e o nível de aprofundamento da exploração desejada, uma elevada taxa de sucesso na exploração da falha e, consequentemente, alcançar o objetivo desejado seja ele relacionado à obtenção de informações, seja ao controle do recurso explorado, seja à negação de serviço ao recurso por parte de outrem.

Relacionado ao desenvolvimento de produtos e aplicações na área de tecnologia e segurança da informação, um *Zero-day* é uma falha cujo responsável pelo produto ou aplicação não teve tempo hábil para correção antes de a falha ser explorada, seja por total desconhecimento prévio da falha, seja por inexistência de tempo hábil de publicação de correção antes de a falha ser conhecida e explorada.

Após a falha ser publicamente conhecida através de canais oficiais de divulgação, lista de segurança, entre outros, ela deixa de ser considerada uma vulnerabilidade *Zero-day* e passa a ser uma falha explorável em condições que atendam a versão que é atingida por determinada falha.

Devido as características de desenvolvimento de determinados sistemas, uma falha atinge uma única versão de um determinado produto, ou atinge múltiplos produtos em múltiplas versões. Exemplo disso é o caso da falha estar em uma biblioteca que é base de desenvolvimento de vários produtos.

B. EXPLOIT

Do ponto de vista técnico, um *exploit* [4] é um trecho de código de programa desenvolvido e compilado para uma arquitetura de hardware ou para uma arquitetura de software ou para um tipo de aplicativo. Este código é criado com o intuito de explorar uma vulnerabilidade associada a um recurso tecnológico, podendo inclusive ser para exploração de vulnerabilidade *Zero-day*.

Em um maior detalhamento, um *exploit* é um trecho de código que normalmente utiliza técnicas de linguagem de máquina de baixo nível, por meio de instruções de registradores em linguagem *assembly* que são capazes de manipular recursos de entrada e saída de memória e de recursos de processamento. Através deste tipo de manipulação, onde exista uma falha capaz de ser explorada em uma aplicação ou hardware, um *exploit* faz com que um conjunto arbitrário de instruções desejada pelo atacante sejam executadas, em detrimento do conjunto original de instruções que deveria ser executado na aplicação explorada.

Do ponto de vista de segurança da informação um *exploit* pode ser visto e entendido como uma arma cibernética. É um recurso tecnológico especializado que faz uso de técnicas específicas relacionadas à arquitetura de hardware e software, cujo efeito gera vantagem estratégica a um atacante e causa prejuízos variados a quem se torne alvo de uma arma deste tipo.

O *exploit* é um recurso técnico eficiente se bem empregado e se bem controlado, que permite o acesso não autorizado a recursos computacionais, possibilitando a escalação de privilégios ou realizando a negação de serviços. Caso um *exploit* seja mal empregado, ele se torna um recurso perdido. Uma vez utilizado fora do ambiente de controle, um *exploit* deixa de ter a eficiência desejada. Ainda, caso mal empregado, permite a implicação de autoria, e como consequência, de possíveis acusações envolvendo crime e espionagem cibernética.

Há basicamente dois tipos de *exploit* – local e remoto. Um *exploit* local é aquele onde o atacante já possui acesso a um determinando recurso computacional e executa localmente um determinado código com o intuito de aumentar suas permissões no recurso explorado, instalar determinadas ferramentas auxiliares ou permitir ainda que seja possível o controle remoto do recurso computacional mediante exploração de falhas. Um *exploit* remoto é aquele onde o atacante é capaz de executar um determinado código por meio de um canal de comunicação ou uma rede, permitindo a exploração de falha no recurso computacional, sendo possível o seu controle remotamente.

Assim, um *exploit* é um recurso técnico que pode atingir qualquer usuário em qualquer ambiente cibernético, seja ele conectado ou isolado, sendo que, neste último caso, o *exploit* pode ser inserido no ambiente por meio de dispositivos removíveis. Normalmente, um *exploit* [5] pode alterar o funcionamento do hardware ou do software, permitindo que sejam instalados ou manipulados outros recursos de acordo com o objetivo do atacante.

C. Processo Básico de Emprego de um Exploit

Um *exploit* pode ser utilizado das mais variadas formas e técnicas. Por exemplo, pode ser um pequeno arquivo executável para uma plataforma de hardware e software, pode ser embutido em páginas na Internet, pode ser um arquivo anexo a uma mensagem de correio eletrônico, pode vir na forma de uma mensagem de texto em um celular, pode estar embutido em arquivos digitais com as mais variadas extensões.

Uma vez executado, o *exploit* permite ao atacante realizar ações no ambiente cibernético, como por exemplo, permitir o controle remoto de determinados dispositivos através de um canal de comunicação, ou causar falha ou interrupção de serviços computacionais e de sistema de controle de outrem. Para ser eficiente, um *exploit* faz uso de técnicas de exploração de falhas que permitem ao atacante a execução de instruções visando o controle do recurso computacional desejado.

O desenvolvimento de *exploits* passa por etapas específicas que variam de acordo com o recurso a ser explorado. Desta forma, para uma vulnerabilidade ser passível de exploração em tempo hábil é necessário, no mínimo, o domínio da tecnologia a ser explorada. Isto envolve a capacidade de entender ou replicar o ambiente a ser explorado. Entretanto, o *exploit* pode ser desenvolvido por meios próprios ou ser adquirido para um propósito particular, observada uma janela de oportunidade.

De maneira geral, o processo de emprego de um *exploit* segue um ciclo de vida, conforme a Figura 1, cujas etapas são explicadas na Tabela 1. Observa-se nessa figura que a janela de oportunidade, durante a qual o recurso está explorável, é variável em função de vários aspectos, como tempo de atualização, tempo de descobrimento, complexidade da falha, entre outras.

Conforme o estudo de Bilge e Dumitras [7], são conhecidos alguns detalhes da janela de oportunidade. Por exemplo, o tempo médio de detecção de um ataque utilizando uma vulnerabilidade *Zero-day* é de 300 dias após o recurso computacional já se encontrar em fase de exploração por um atacante. Nesse período, normalmente não estão disponíveis atualizações, assinaturas de antivírus ou sistemas de detecção de intrusão, o que faz com que a detecção seja mais difícil e complexa. Ainda em [7], encontra-se a informação de que a duração de ataques que exploram falhas *Zero-day* varia entre 19 dias e 30 meses.

Fig. 1. Processo Básico de Emprego de um *Exploit* (Adaptado de [6])

TABELA I. Etapas do Ciclo de Emprego de um *Exploit*

Etapa	Descrição
1 – Lançamento de um produto	Nesta etapa é disponibilizado um produto de um fabricante no mercado de tecnologia da informação. Este produto pode ser um *hardware* com *software* customizado, pode ser um *software* aplicativo, um sistema operacional, uma plataforma cliente-servidor ou qualquer recurso técnico que faça uso de algum recurso computacional.
2.1 – Descobrimento privado de vulnerabilidade	Nesta etapa, por meio de pesquisa e desenvolvimento, um determinado grupo, empresa ou pessoa, descobre uma determinada vulnerabilidade. Caso esta vulnerabilidade não seja divulgada publicamente, ela se torna uma vulnerabilidade *Zero-day*.
2.2 – Relatório Público de vulnerabilidade	É quando uma vulnerabilidade é descoberta e publicada na Internet, seja por um grupo de pesquisa, uma empresa, seja pelo próprio fabricante do produto. Nesta etapa, normalmente, o responsável pelo produto é previamente avisado para que desenvolva a respectiva correção do produto antes da divulgação em massa da falha. Vale observar que antes da correção, qualquer produto atingido pela vulnerabilidade é passível de exploração.
3 – Criação de Prova de Conceito (PoC)	Dado o conhecimento da vulnerabilidade, é desenvolvido um código prévio que demonstra a capacidade da exploração da vulnerabilidade. Normalmente a PoC é um passo para a criação de um *exploit* para o produto vulnerável.
4 – Desenvolvimento de atualização do produto	O fabricante é capaz de desenvolver uma nova versão do produto, de maneira a evitar que o *exploit* seja eficiente em termos de exploração da falha de segurança.
5 – Disponibilização de atualização	O responsável pelo produto torna público a sua atualização, de forma que os usuários desse produto possam fazer a respectiva atualização e consequentemente evitar que o produto seja explorado.
6 – Aplicação da atualização	Os usuários devem ter um processo de manutenção de atualização do produto, de forma a fazer com que o produto seja atualizado com uma versão que não seja explorada.

Cabe ressaltar que ataques que utilizam falhas *Zero-day* normalmente possuem alvos direcionados, o que os tornam

mais difíceis de detectar através de uso de técnicas convencionais, já que muitas vezes, quem é alvo de tais ataques não torna público o ataque e muito menos divulga detalhes do mesmo. O estudo [7] mostra ainda que tornar pública uma vulnerabilidade resulta em aumento significativo do número de sistemas atacados. Isso pode ser ainda confirmado atualmente conforme relatório de segurança da informação divulgado pela Verizon [8].

A respeito do processo de emprego de um *exploit*, observa-se que algumas etapas são fundamentais. Por exemplo, o atacante precisa conhecer no mínimo o recurso a ser explorado, ser capaz de desenvolver o código para a falha a ser explorada e observar, além da efetividade do código, o tempo efetivo de exploração, dentro da janela de oportunidade.

Cabe ressaltar que, para uma determinada falha para ser explorada, é precisa haver um objetivo bem definido, incluindo a especificação do recurso a ser explorado, e os processos de análise da falha e de desenvolvimento do método e das ferramentas de exploração. Por exemplo, explorar uma falha para controlar remotamente o dispositivo computacional, em termos de código, é totalmente diferente de copiar dados do ou para o recurso, que por sua vez é diferente do código que altera a funcionalidade do recurso, e assim por diante. Além disso, uma falha é explorável dentro de uma janela de oportunidade, que é variável.

III. LEVANTAMENTO SOBRE MERCADO DE *EXPLOITS*

O que determina um mercado comercial é basicamente a lei da oferta e da demanda. No espaço cibernético existem diversas fontes relacionadas à negociação de *exploits*, seja na forma de comércio entre pessoas, entre empresas e pessoas, entre empresas, entre empresas e governo. Isto é variável e normalmente é dependente da necessidade de cada um que esteja envolvido em tal processo.

O fato é que existe tal mercado. Muitas vezes ele é considerado legal, uma vez que existe a negociação lícita, dentro do que rege a lei, sendo negociado a cifras elevadas por entidades e governos. Entretanto, ele também pode se caracterizar como ilegal em muitos aspectos, já que há também produtos adquiridos por meio de fóruns eletrônicos sem nenhum tipo de controle e com as mais variadas formas de pagamento, inclusive relacionadas a atividades de crimes cibernéticos, encontrando-se até mesmo definidos na legislação em diversos países. Diante dessa situação, caracteriza-se como uma necessidade social o estabelecimento de processos e técnicas forenses relativas a tais ambientes.

O mercado de *exploits* é incentivado pelo descobrimento de falhas de segurança em sistemas computacionais que não são reportadas ao fabricante desses sistemas para a devida correção. Assim, os responsáveis pelo descobrimento têm a possibilidade de vender a falha, havendo casos de valores de venda substanciais, como U$ 500.000,00, segundo reportagem da revista Forbes [9]. Constata-se que os valores são variáveis dependendo da utilidade e efetividade do *exploit* sob o ponto de vista do interesse do comprador, da abrangência da falha ou da dificuldade de exploração de um determinado recurso.

O mercado de *exploits* também está ligado ao mercado do crime cibernético. Um estudo realizado pela empresa Hewlett-Packard (HP) aponta que existe um crescimento do impacto financeiro relacionado à segurança da informação da ordem de 40% [10]. Esse estudo também aponta que, em uma análise de organizações dos EUA, o custo médio do crime cibernético para essas organizações foi da ordem de U$ 8.900.000,00 em 2012. Esta cifra representa um aumento de 6% quando comparado aos dados de 2011 e 38% quando comparado a 2010. O estudo apontou que em 2012 houve aumento de 42% no número de ataques cibernéticos, com as organizações passando por uma média de 102 ataques com sucesso por semana, com custo médio de solução na ordem de valores de U$ 591.780,00.

Se uma organização é alvo de um ataque cibernético furtivo mediante uso de falhas *Zero-day*, é provável que ela não descubra o efeito desse ataque antes de a falha se tornar pública e, muitas vezes, antes que o fabricante tenha disponibilizado uma correção. Neste caso, o prejuízo pode ser incalculável sob os mais diversos aspectos, gerando diversas formas de perdas para a empresa.

Outro estudo, este da empresa Symantec Labs [11], lista dados do ano de 2013 que revelam características do mercado de *exploits*:

- Houve 91% de aumento em campanhas de ataques direcionados a um alvo específico;
- Aumento de 62% no número de falhas de segurança;
- Mais de 552 milhões de identidades foram expostas;
- Pelo menos 23 falhas *Zero-day* descobertas;
- Cerca de 38% de usuários de plataformas móveis experimentaram problemas com crimes cibernéticos;
- Aumento de 66% no volume de mensagens não solicitadas; e
- 1 em cada 392 e-mails continha ataques de roubo de senhas.

De acordo com um estudo conduzido por Goncharov [12], da empresa Trend Micro, há no mercado cibernético Russo um submundo que é ativo de maneira organizada desde 2004 e é utilizado como mercado para troca de informações sobre vulnerabilidades e *exploits*. Nesse caso, alguns dos atores, como zloy.org, DaMaGeLab e XaKePoK.NeT, são bastante utilizados e constituem a bastante tempo focos de atividades relacionadas ao mercado de crimes cibernéticos em geral. Também existem diversos outros casos mais recentes, tais como o 1337Day para *exploits*, e, em um domínio diverso mas aparentado pela suscetibilidade ao crime organizado, os casos SILKROAD e AGORA para drogas.

A. *ASPECTOS ESTRATÉGICOS*

O ponto chave do estudo de Goncharov [12] é a descrição do alto nível de especialização de muitas partes do mercado de crime cibernético russo. O relatório analisado deixa claro que um *hacker* com uma boa cadeia de relacionamentos e contatos não precisa mais criar todas as suas armas cibernéticas. Tais armas simplesmente podem ser compradas de outro *hacker*, ou se pode alugar uma plataforma de negação de serviço, ou terceirizar determinadas funções. Nesse mercado, existem especialistas para qualquer tipo de atividade no espaço cibernético, incluindo cifração, ataques de negação distribuídos, redirecionamento de tráfico, serviços de *Pay-Per-Install* (Pague por instalação de *malware*), entre outros.

O estudo ainda apontou que houve depreciação em valores de acordo com o tipo de serviço. Por exemplo, a Tabela II aponta que no caso do furto de cartões de crédito, em vários países, o preço unitário encontra-se em queda desde 2011.

TABELA II. PREÇO (US$) PARA DADOS DE CARTÕES DE CRÉDITOS FURTADOS (FONTE: TREND MICRO [12])

País	2011	2012	2013
Austrália	7	5	4
Canadá	5	5	4
Alemanha	9	5	6
Reino Unido	7	6 – 8	5
Estados Unidos	3	1	1

No que se refere a contas de serviços de e-mails e redes sociais, conforme dados da Tabela III, o estudo aponta que o preço médio de sequestro de contas está em declínio. De maneira geral, os dados mostram que, dependendo do tipo de atividade, o custo tem diminuído em função do aumento da oferta por esses serviços no mercado russo.

TABELA III. PREÇO (US$) PARA CONTAS HACKEADAS (ADAPTADO DE TREND MICRO [12])

Serviço	2011	2012	2013
Facebook	200	160	100
Gmail	117	120	100
Hotmail	107	100	100
Mail.ru	74	70	50
Twitter	167	40	--

Além disso, o relatório aponta que a oferta excessiva também tem influenciado na qualidade da oferta. Já que a competitividade é alta, a qualidade dos produtos ofertados muitas vezes é duvidosa e estes não fazem a função para a qual foram comprados. Este aspecto apenas reforça a necessidade de entender-se o que está sendo negociado, o que torna fundamental a capacidade de verificação técnica, para obter uma exata caracterização do produto comercializado.

Um aspecto a ser observado é que a questão vai além do preço de produtos, sendo importante se observar quem é que está pagando por *exploits Zero-day*. Em uma reportagem da revista Forbes [9], um negociador intermediador (*broker*), conhecido pelo codinome Grugq, relata que a maior parte dos clientes é de governos ocidentais, tipicamente EUA e Europa, simplesmente pelo fato de que eles pagam mais do que russos e chineses, por exemplo.

Na entrevista realizada [9], o citado *broker* reportou que vender determinados produtos para a máfia russa é mau negócio, porque além de pagar pouco, há grande chance de que o produto tenha pouca utilidade em poucos dias, caso seja um *exploit*. Além disso, há na Rússia muitos criminosos cibernéticos, o que faz com que o preço deste tipo de produto derive da alta competitividade e custo muito baixo. Isso ainda é agravado pelo fato de haver muita desonestidade na negociação de produtos versus sua real efetividade no mercado russo.

Ainda segundo o entrevistado em [9], no caso do mercado chinês, que também possui um número elevado de *hackers*, a venda de armas cibernéticas por *hackers* chineses é exclusiva para o governo chinês, e o preço é muito baixo. A reportagem [9] ainda aponta que outros mercados de artefatos cibernéticos, como Oriente Médio e Ásia não superam o preço ofertado pelos países ocidentais.

B. MERCADO CIRCUNSCRITO

O mercado de *exploits* apresenta um agravante por se constituir um mercado circunscrito, muito particular e fechado. Em consequência de ser um mercado reservado, nele existem vulnerabilidades que são conhecidas apenas por grupos privilegiados e restritos. Nesse cenário, estão inseridos criminosos cibernéticos específicos e *brokers*, que normalmente possuem uma rede privilegiada de contatos.

Além disso, agências de diversos governos também estão inseridas neste mercado restrito, tanto no papel de consumidores quanto de membros ativos. A atuação de agências de governo se concretiza influenciando na criação de produtos com falhas de segurança. Também ocorre pela interferência na criação de padrões de mercado com dificuldades técnicas que proporcionam vantagens para agências de inteligência dotadas de recursos técnicos e alta capacidade de processamento e armazenamento.

Reitera-se que, por compreende elevada capacidade técnica e recursos financeiros significativos para investimento em pesquisa de novas vulnerabilidades e desenvolvimento de novos *exploits*, mercado circunscrito representa um alto risco para quem lida diretamente com desenvolvimento de software e para fabricantes que possuem produtos na lista de interesse da comunidade de inteligência, tais como sistemas operacionais, hardware para comutadores de rede, roteadores, memórias de computadores, soluções de segurança da informação, entre outros.

Um dos pontos fundamentais desse mercado circunscrito reside no fato de que os grupos participantes têm acesso à informação crítica, o que permite que eles possam comprometer sistemas vulneráveis sem que o público jamais conheça essas ameaças.

Nesse aspecto, um estudo de 2013 da NSS Labs [13] analisou uma série com dados de diversos anos de dois grandes programas de vulnerabilidades e os resultados apontaram que nos últimos 3 anos anteriores ao estudo, em qualquer dia, grupos privilegiados tinham acesso a pelo menos 58 vulnerabilidades, tendo como alvo sistemas de empresas como Microsoft, Apple, Oracle ou Adobe. Além disso, essas vulnerabilidades continuaram privadas por uma média de 151 dias. Neste caso, a janela de oportunidade (vide Figura 1) é da ordem de meses para os detentores de tais armas cibernéticas que nesse intervale tiveram a oportunidade de fazer uso de técnicas de infiltração, exfiltração de dados, controle remoto, vigilância cibernética, entre outras, virtualmente sem serem detectados ou combatidos.

C. CUSTOS DE EXPLOITS

É muito difícil definir o real custo de um *exploit*, haja vista a necessidade de empregar vários critérios de análise, além do tipo do *exploit*. Normalmente, características como a complexidade ou facilidade do uso, a efetividade, a especificação do alvo, a janela de oportunidade, localização do alvo, dificuldade de penetração, etc., são fatores que influenciam na composição do preço final de uma arma cibernética.

Na citada reportagem da Forbes [9], alguns números puderam ser correlacionados ao tipo de produto do mercado de tecnologia da informação afetado eventualmente por um *exploit* para uma vulnerabilidade *Zero-day*, conforme resumo na Tabela IV.

Em um levantamento específico para este artigo, atualizado com valores para 2015, algumas ferramentas com uso direcionado e de maneira restrita foram encontradas oferecidas por um único *hacker* (http://apt0.no-ip.biz/). A Tabela V indica alguns valores para produtos específicos, embora os dados não se adequem a uma comparação de preços nem a uma análise de depreciação. Entretanto, indicam que, uma vez que um determinado *Exploit* funcione em uma determinada versão do sistema alvo, os interessados em adquirir tal produto podem pagar um valor que depende de fatores que são inerentes ao

próprio *Exploit* (eficiência, por exemplo). Vale observar que a Tabela V não contempla *exploits Zero-day*.

TABELA IV. PREÇO (US$) MÉDIO DE UM *EXPLOIT ZERO-DAY* (ADAPTADO DE TREND MICRO [12])

Produto	Preço Estimado
Adobe Reader	5.000 – 30.000
Mac OSX	20.000 – 50.000
Android	30.000 – 60.000
Flash ou Plugins Java para Navegadores	40.000 – 100.000
Microsoft Word	50.000 – 100.000
Windows	60.000 – 120.000
Firefox ou Safari	60.000 – 150.000
Chrome ou Internet Explorer	80.000 – 200.000
IOS	100.000 – 250.000

TABELA V. PREÇOS (US$) DE *EXPLOITS ENCONTRADOS EM 2015*

Ferramenta	Sistema Alvo	Preço
Exploits, *Kits* de teste de penetração, infeção silenciosa	Firefox 22-27	200 (reduzido)
Mozilla Firefox *Bootstrapped Code Execution*	Firefox addon (Windows 7, Windows XP, Window 8.1)	400 (reduzido)
OLE automation array remote code execution	Internet Explorer <= 11	800
WolfPack	Java 1.6.0* Java 1.7.0_06 Java 1.7.0_10 Java 1.7.0_17 Java Applet (Windows 7, Windows XP, Window 8.1)	1000
Exploits, *Kits* de teste de penetração, infeção silenciosa	Firefox 31-34 (Windows 7)	800
Polymorphism IE11 Exploit Source Code	Internet Explorer 11 (Windows 7)	1200
Polymorphism Firefox 31-34 Exploit Source Code	Firefox 31-34 (Windows 7)	1000
Insanity - Infection Kit	Locação de infraestrutura para testes de penetração em sistemas	2000 (por mês)

D. PROGRAMAS DE RECOMPENSA PARA DESCOBERTA DE FALHAS

E. PREÇO MÉDIO (US$) PAGO POR *BUG BOUNTY PROGRAMS* (ADAPTADO DE FREI [13])

Empresa	Preço (US$)	Descrição
Google	~ 580.000	Preço médio durante 3 anos por 501 vulnerabilidades descobertas no navegador Chrome. Equivale a 28% de atualizações no mesmo período.
Mozilla	~ 570.000	Preço médio durante 3 anos por 190 vulnerabilidades no navegador Firefox. Equivale a 24% de atualizações no mesmo período.
Facebook	~ 1.000.000	Preço médio pago desde 2011 em seu programa de vulnerabilidades.
Microsoft	~ 130.000	Preço médio desde 2013 para programa de reporte de novas técnicas de exploração de falhas.

Além do setor de mercado de produtos e infraestrutura, verifica-se um setor de programas de recompensa para descoberta de falhas (*Bug Bounty programs*), cujos dados encontram-se na Tabela VI, conforme o estudo conduzido pela NSS Labs [13].

F. FORMA DE COMERCIALIZAÇÃO

Verifica-se um aumento do número de empresas que oferecem *exploits Zero-day* para os clientes, em um modelo de venda por assinaturas. Tais empresas normalmente não revelam seus clientes. Entretanto, grandes compradores incluem agências de governo. Neste caso, algumas empresas oferecem 25 *exploits Zero-day* por ano, ao custo de US$ 2.500.000,00 (preço de 2010) [9].

Outros dados disponíveis na Internet apontam que o preço médio da assinatura está entre US$ 40.000,00 e US$ 160.000,00 para clientes restritos de determinados países (explicitamente, Estados Unidos e Europa), relativamente a uma média de 100 *exploits* exclusivos por ano.

Existem ainda plataformas de desenvolvimento e de distribuição de *exploits* com vulnerabilidades *Zero-day*. Uma delas, amplamente conhecida para técnicas de exploração e comando e controle, é o Metasploit [14], disponibilizada tanto em versão pública como em versão comercial, sendo ambas as versões coordenadas e mantidas por uma empresa americana.

Outra plataforma, menos difundida, mas aparentemente muito conhecida no mercado de segurança cibernética, é conhecida como *Elderwood Project*, tendo sido reportada publicamente pela equipe de segurança da empresa Symantec [15]. Tal plataforma, relacionada a um grupo específico de *hackers*, permite a criação de vetores de ataques e códigos para vulnerabilidades em sistemas Adobe Flash e navegadores Internet Explorer, por exemplo.

Existem dados em fontes abertas que apontam o uso desta plataforma em ataques conhecidos como a "Operação Aurora". Também existem relatos de arquivos de *malware* relacionados ao *Elderwood Project* em países como Estados Unidos, Canadá, China, Austrália, Hong Kong, entre outros. Uma característica marcante do grupo relacionado ao *Elderwood Project* é que o seu foco de ataque são empresas que atuam como fornecedores de equipamentos eletrônicos, cujos principais clientes são grandes empresas da área de defesa.

IV. EMPRESAS ESPECIALIZADAS

Conforme o mercado se movimenta, existem empresas que fornecem serviços especializados na área de *exploits*. A seguir, são apresentadas algumas dessas empresas cujos dados estão disponíveis na Internet.

A. VUPEN

A Vupen é uma empresa francesa que participa ativamente do mercado de *exploits* e *bug bounty programs*. Segundo dados da própria empresa, todas as vulnerabilidades da Vupen são de desenvolvimento próprio e permitem a agências de governo e à comunidade de inteligência a condução de operações de rede em suas missões cibernéticas.

Segundo dados oriundos de fontes abertas [16], a lista de clientes da Vupen inclui governos, empresas de segurança em geral, empresas financeiras, de saúde, seguros, manufatura, tecnologia entre outros. Entre os clientes, a lista inclui países-membro da Organização do Tratado do Atlântico Norte (OTAN). Algumas fontes abertas [17] inclusive reportam que a estadunidense National Security Agency (NSA) comprou *exploits Zero-day* fornecidos pela Vupen.

B. REVULN

A ReVuln é uma empresa europeia situada em Malta. Segundo dados da própria empresa, seus serviços estão no estado da arte de pesquisa em segurança cibernética e em

soluções de segurança para clientes pelo mundo. É uma empresa especializada em pesquisa de vulnerabilidades para medidas ofensivas e defensivas em segurança cibernética.

Alguns relatos de fontes abertas [18] apontam que a ReVuln atuou na geração de códigos para serem explorados em usuários de redes de jogos online e que a lista de clientes inclui a NSA e adversários americanos como o grupo Guardas Revolucionários do Irã.

C. NETRAGARD

A Netragard é uma empresa americana que atua no mercado de segurança, com desenvolvimento de *exploits* e testes de penetração. A empresa trabalha com uma plataforma proprietária denominada *Real Time Dynamic Testing*, que segundo a própria empresa é diferente de produtos comuns e ferramentas automatizadas. A empresa mantém times permanentes de pesquisa em vulnerabilidades e desenvolvimento de *exploits*.

Segundo algumas fontes abertas [19] a lista de clientes desta empresa é restrita aos Estados Unidos.

D. ENDGAME SYSTEMS

A *Endgame Systems* é uma empresa americana que lida com inteligência em segurança da informação e análise de dados. Segundo dados da própria empresa, os produtos ofertados permitem uma visibilidade em tempo real em seus domínios digitais e através de uma variedade de aplicações que permitem resolver a mais variada gama de problemas digitais.

Segundo relatos de fontes abertas [20], a lista de clientes inclui a NSA, a Marinha e a Força Aérea Americanas, além da CIA, Inteligência Britânica e Comando Cibernético Americano.

E. EXODUS INTELLIGENCE

A *Exodus Intelligence* é uma empresa americana que lida com pesquisa e desenvolvimento de soluções de segurança cibernética. Sua missão institucional é prover aos clientes informações confiáveis e relacionadas a um contexto para suas vulnerabilidades e *exploits Zero-day* exclusivas.

Entre a lista de parceiros estratégicos desta empresa estão o Departamento de Defesa Americano, a DARPA, o NSSLabs, a Fortinet, entre outros.

F. HACKING TEAM

Hacking Team é uma empresa italiana que provê tecnologias ofensivas de comando e controle para agências de inteligência e de segurança. Seu principal produto faz uso de *exploits* e vulnerabilidades, para ser capaz de permitir controle remoto de diversas plataformas.

Existem relatos em fontes abertas [21] que apontam o uso de produtos desta empresa em pelo menos 30 países, incluindo Sudão, Azerbaijão, Arábia Saudita, Marrocos e Emirados Árabes.

G. AGT

A *Advanced German Technology* (AGT) é uma empresa alemã que oferece diversos produtos e soluções. Entre elas, é ofertado um sistema de comando e controle com técnicas avançadas de furtividade e exploração de vulnerabilidades para controle remoto de diversos dispositivos.

A lista de clientes da AGT inclui agências de segurança e forças da lei.

H. RAPID7

A Rapid7, uma empresa americana, é o braço comercial do *framework* de exploração de falhas *opensource* conhecido como Metasploit [14]. A empresa possui diversos produtos e serviços que variam de gerenciamento de vulnerabilidades até exploração de falhas e desenvolvimento de *exploits*.

A lista de clientes da Rapid7 inclui governos, agências de saúde, de telecomunicação, de energia, de finanças, entre outros.

I. IMMUNITYSEC

A *ImmunitySec* é uma empresa americana que trabalha com plataforma de comando e controle, desenvolvimento e implementação de *exploits*. A empresa possui aplicações e diversos *exploits* que são amplamente testados antes da sua disponibilização a seus clientes.

J. COREIMPACT

A CoreImpact é uma empresa americana que oferta sistemas de comando e controle e *exploits* para sistemas com o intuito de realizar testes de penetração em diversos cenários, plataformas e dispositivos.

K. GFI

A Gfi é uma empresa americana que oferta serviços na área de gerenciamento de vulnerabilidades para redes de comunicação e aplicação de atualizações de segurança contra falhas exploradas remotamente e localmente.

L. BEYONDTRUST

A BeyondTrust é uma empresa americana que provê soluções para o gerenciamento de vulnerabilidades. Seus produtos lidam com a identificação e resposta a vulnerabilidades de ataques cibernéticos.

M. BLUECOAT

A BlueCoat é uma empresa americana que provê soluções na área de *Advanced Persistent Threats* (APTs), análise de *malware* e soluções de análise de vulnerabilidades dos mais variados tipos. Os produtos são voltados para o mercado de segurança cibernética e possuem as mais diversas aplicações.

N. FINFISHER

A *FinFisher* é uma empresa alemã que atua no segmento de segurança cibernética oferecendo soluções para agências governamentais de segurança, comunidade de inteligência e governos. Seu foco de atuação está voltado principalmente ao combate a atividades criminosas no espaço cibernético. Ela atua em pesquisa e desenvolvimento e intrusão em sistemas de tecnologia da informação, segundo dados da própria empresa.

V. CONCLUSÕES

A inserção de uma instituição, ou mesmo de um profissional individual, na área de monitoração do mercado de vulnerabilidades, assim como no domínio geral da segurança cibernética, requer capacidades de análise, de acompanhamento constante, empregando pessoal técnico com formação avançada e treinamentos atualizados. Por sua importância nos domínios de segurança da informação, seja ela pública ou privada, é importante que a atuação nessa área venha provida de forte presença com competência e capacidade técnica, o que requer necessariamente investimento e planejamento de longo prazo.

Um planejamento estratégico definido é fator crítico de sucesso para atuação no mercado de vulnerabilidades, assim

como é fundamental para a correta atuação e inserção na área de segurança cibernética, mais especificamente para investigação no mercado de vulnerabilidades de sistemas computacionais, seja ela com o intuito de trabalho forense, seja ela para atuação contra o crime cibernético.

Do levantamento apresentado neste artigo, conclui-se que essa atuação parece ser considerada por diversos países como atividade sistemática, contínua, merecedores de alocação de recursos humanos, tecnológicos e financeiros, caracterizando-se, portanto, como estratégica para tais países. Vale ressaltar que a área cibernética não é um espaço para ações estanques e isoladas, sem foco e muito menos sem objetivo concreto. Assim, é estratégica também a geração de conhecimento específico sobre os processos e tecnologias de produção de *exploits*, o que é fundamental para a consecução do entendimento e avanço da análise de ameaças cibernéticas.

A oferta de *exploits* por parte de empresas ou grupos é uma realidade efetiva que pode talvez levar ao entendimento de que a exploração de vulnerabilidades se trataria apenas de comprar *exploits* de prateleiras e empregá-los contra determinados alvos. Em uma perspectiva de monitoração, atuação preventiva contra crimes, investigação forense, e atividades correlatas, requisita-se um entendimento mais abrangente, que inclua o esclarecimento de questões como: existência de alvo claramente definido; capacidade técnica, em termos de pessoas e recursos técnicos, suficiente para a definição e avaliação de ferramentas a serem adquiridas; diferentes implicações quando da aplicação de eventuais ferramentas, que podem ser usadas tanto para ofender e atacar, quanto para prevenir e defender; necessidade de atribuição de origem de manobras consideradas ofensiva no espaço cibernético; as medidas que devem ser tomadas em função do sucesso ou insucesso no emprego de determinada ferramenta; etc.

Algumas dessas questões têm, não somente implicações estratégicas, mas devem muitas vezes ser detalhadas em seus aspectos táticos e operacionais, de acordo com o nível de ação que se deseje empreender no mercado de vulnerabilidades. Por outro lado, visto que muitas das ferramentas aí disponíveis são usadas também no sentido de contribuir com medidas de proteção de sistemas cibernéticos, é importante ressaltar que utilizar dessa forma tais ferramentas requer preparação, não apenas adquiri-las, mas, principalmente, para ter um planejamento claro e definido do seu emprego, bem como ter capacidade de analisá-las, seja na perspectiva forense, seja na perspectiva de crime cibernético.

O presente trabalho, de caráter exploratório, apresenta atividades que têm a possibilidade de, ao menos parcialmente, ser objeto de automação, seja no levantamento, nas análises e na aquisição de conhecimentos, reunindo tecnologias de mineração de dados, filtragem, descoberta de padrões, representação de ontologias de segurança na área cibernética, etc., assuntos que constituem possíveis trabalhos futuros.

AGRADECIMENTOS

Os autores agradecem às Agências brasileiras de pesquisa e inovação CAPES (Projeto FORTE, Edital CAPES Ciências Forenses 25/2014) e FINEP (Convênio RENASIC/PROTO 01.12.0555.00), pelo suporte a este trabalho.

REFERÊNCIAS

[1] Clarke, Richard A.; Knake, Robert K.; “Cyber war”. Tantor Media, Incorporated, 2014.

[2] Shackelford, Scott J.; “Managing Cyber Attacks in International Law, Business, and Relations: In Search of Cyber Peace.”; Cambridge University Press, 2014.

[3] Hyppönen, Mikko H. "Information Security"; Proceedings of the IATUL Conferences 2014; Purdue University; Purdue e-Pubs; Disponível em http://docs.lib.purdue.edu/iatul/2014/keynotes/1/.

[4] Wu, J.; Arrott, A.; Colon Osorio, F.C., "Protection against remote code execution exploits of popular applications in Windows," Malicious and Unwanted Software: The Americas (MALWARE), 2014 9th International Conference on, vol., no., pp.26-31, 28-30 Oct. 2014; DOI: 10.1109/MALWARE.2014.6999416.

[5] Mahaffey, Kevin, John G. Hering, and James Burgess. "Security Status and Information Display System." U.S. Patent No. 20,140,373,162. 18 Dec. 2014.

[6] Tiedata. “What are Web Based *Exploits*?” Disponível em: http://www.tiedata.com/web*exploits*.asp. Acessado em maio de 2014.

[7] Bilge, Leyla; Dumitras, Tudor; “Before We Knew It - An Empirical Study of *Zero-day* Attacks In The Real World”. Symantec Research Labs. Outubro de 2012.

[8] Verizon Report. “The 2015 Data Breach Investigations Report (DBIR)”; 2015. Disponível em http://www.verizonenterprise.com/DBIR/2015/.

[9] Revista Forbes; “Shopping For *Zero-day*s: A Price List For Hackers' Secret Software *Exploits*”. Edição de 23 de março 2012.

[10] HP Research. “Cybercrime Costs Rise Nearly 40 Percent, Attack Frequency Doubles”. Publicado em 08 de outubro de 2012.

[11] Symantec Labs. “2014 Internet Security Threat Report”. Volume 19. Publicado em Abril de 2014.

[12] Goncharov, Max; “The Russian Underground, Revisited”. Cybercriminal Underground Economy Series. A Trend Micro Research Paper. Publicado em 28 de abril de 2014. Disponível em http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp-russian-underground-revisited.pdf.

[13] Frei, Stefan; “The Known Unknowns: Empirical Analysis of Publicly Unknown Security Vulnerabilities”; Analyst Brief; NSS Labs; 05 de Dezembro de 2013.

[14] Metasploit; “The Metasploit Project”; Rapid7, 2015. Disponível em http://www.metasploit.com/.

[15] O’Gorman, Gavin; McDonald, Geoff. “The Elderwood Project”. Symantec Security Response. 06 Sep 2012.

[16] Fidler, Mailyn; “Anarchy or Regulation: Controlling the Global Trade in Zero-Day Vulnerabilities”; Thesis submitted to the Interschool Honors Program in International Security Studies; Center for International Security and Cooperation Freeman Spogli Institute for International Studies; Stanford University; May 2014.

[17] Walker, Danielle; “NSA sought services of French security firm, zero-day seller Vupen”; SC Magazine; September 18, 2013. Disponível em http://www.scmagazine.com/nsa-sought-services-of-french-security-firm-zero-day-seller-vupen/article/312266/.

[18] Perlroth, Nicole; Sangerjuly, David E. “Nations Buying as Hackers Sell Flaws in Computer Code”; New York Times, July, 2013.

[19] Ungerleider, Neal; “How Spies, Hackers, and the Government Bolster a Booming Software Exploit Market”; Disponível em http://www.fastcompany.com/3009156/the-code-war/how-spies-hackers-and-the-government-bolster-a-booming-software-exploit-market.

[20] Harris, Shane; “The Mercenaries. Ex-NSA hackers and their corporate clients are stretching legal boundaries and shaping the future of cyberwar”. Disponível em http://www.slate.com/articles/technology/future_tense/2014/11/how_corporations_are_adopting_cyber_defense_and_around_legal_barriers_the.html.

[21] Reporters without borders; “The Enemies of Internet Special Edition: Surveillance”; Disponível em http://surveillance.rsf.org/en/hacking-team/.

DOI: 10.5769/C2015013 or http://dx.doi.org/10.5769/C2015013

Extração de dados da Web relativos a licitações e contratos públicos para inferência por reconhecimento de padrões estatísticos: estudo de caso.

Cirilo Max Macedo de Morais e Díbio Leandro Borges

***Resumo*—Este artigo relaciona técnicas para extração de dados da Web relativos a licitações e contratos públicos para inferência por reconhecimento de padrões estatísticos, utilizando a linguagem R, e apresenta um estudo de caso para identificação da correlação entre o maior número de contratos realizados por uma unidade administrativa com uma mesma empresa e o número total de contratos celebrados por esta unidade administrativa. As informações foram extraídas do sítio da internet www.dados.gov.br.**

Palavras-Chave— extração de dados da Web, contratos públicos, reconhecimento de padrões, dados abertos, dados governo.

***Abstract*—This article lists techniques for extracting Web data relating to tenders and public contracts for statistical inference for pattern recognition, using the R language, and presents a case study to identify the correlation between the maximum number of public contracts from an administration unity with the same company and the total number of contracts from this administration unity. The information was extracted from the Web site www.dados.gov.br.**

Keywords—Web data extraction, public contracts, pattern recognition, open data, government data.

I. INTRODUÇÃO

Identificar padrões em licitações e contratos públicos fraudulentos é de fundamental importância para a melhor utilização dos recursos humanos dos órgãos de controle governamentais. A enorme quantidade de licitações públicas, juntamente com a escassez de pessoal nos órgãos de controle, faz com que as verificações dos contratos sejam realizadas em um pequeno percentual. O critério de escolha dos contratos a serem auditados é realizado por denúncias, amostragem aleatória, montante dos recursos envolvidos, ausência de prestação de contas ou outros. Tais critérios, por sua vez, implicam na não identificação de irregularidades em muitos contratos analisados e permitem que o restante dos contratos não sofra nenhum tipo de auditoria externa.

A fim de criar um critério alternativo, uma análise automática em todos os contratos realizados na administração pública poderia identificar padrões por tipos de objeto e apontar aqueles com maior possibilidade de ocorrência de irregularidades.

Além disso, o surgimento de nova modalidade de licitação como o Regime Diferenciado de Contratação – RDC, que flexibiliza a obrigatoriedade de existência prévia de projeto executivo e orçamento detalhado para contratação de obras e serviços de engenharia, torna ainda mais necessário o conhecimento prévio de padrões estatísticos de contratos similares.

Como fonte de dados para o reconhecimento de padrões, seriam utilizados os dados públicos disponíveis na Internet, disponibilizados pelo governo federal, que elaborou uma ferramenta para isto. Esta ferramenta, conhecida como Portal Brasileiro de Dados Abertos, apresenta no sítio www.dados.gov.br informações sobre convênios, compras governamentais e outros. Os dados são disponibilizados através de APIs (*Application Programming Interfaces*).

Para exemplificar a possibilidade de inferência estatística nestes dados públicos, o artigo apresenta um estudo de caso mostrando os resultados obtidos mediante regressão linear e sua correspondência à hipótese formulada.

No estudo de caso, pretende-se verificar se existe uma correlação entre o número de contratos celebrados por uma mesma unidade administrativa com uma mesma empresa e o número total de contratos desta unidade administrativa, e se esta correlação varia de acordo com a modalidade de licitação.

A hipótese é que como as licitações públicas buscam celebrar o contrato mais vantajoso para a União e, para tal, devem respeitar, dentre outros, os princípios da moralidade e isonomia, então o número total de contratos celebrados entre uma unidade administrativa e uma mesma empresa deveria supostamente ser proporcional ao número total de contratos celebrados por uma unidade administrativa, preservando uma mesma correlação, independentemente da modalidade de licitação escolhida.

A apresentação do artigo é realizada, inicialmente, mostrando a fonte de dados disponibilizada pelo governo federal e utilizada para o estudo de caso. Em seguida são mostradas algumas técnicas, na linguagem R, utilizadas para a obtenção de dados na internet. Posteriormente é apresentada uma descrição resumida das principais modalidades de licitação. Após a abordagem dos principais assuntos envolvidos no problema analisado, inicia-se a explicação dos procedimentos de extração, análise dos dados e resultados obtidos no estudo de caso. Por fim são apresentadas as conclusões e possibilidades de futuras pesquisas.

.

II. ESTUDO DE CASO

A. Fonte de Dados

Para a realização do estudo de caso, foram utilizados dados extraídos do portal de dados abertos brasileiro, cujas características são descritas em seguida.

O Brasil tornou-se membro do *Open Gorvernment Partnership*, um iniciativa multinacional para adoção mundial do *Open Government Data* (OGD) em setembro de 2011 [2]. O portal de dados abertos brasileiro foi lançado em 2012 e

Cirilo Max Macedo de Morais, Mestrado Profissional em Engenharia Elétrica com ênfase em Informática Forense e Segurança da Informação - Departamento de Engenharia Elétrica, Universidade de Brasília, Campus Universitário Darcy Ribeiro, Prédio CIC/EST, Asa Norte, 70910-900 - Brasília, DF – Brasil, e-mail: cirilomax@gmail.com. e Díbio Leandro Borges, Departamento de Ciência da Computação, Universidade de Brasília, Campus Universitário Darcy Ribeiro, Prédio CIC/EST, Asa Norte, 70910-900 - Brasília, DF – Brasil, e-mail: dibio@unb.br.

disponibilizado no sítio da internet www.dados.gov.br [3]. O desenvolvimento do projeto é parte da Infraestrutura Nacional de Dados Abertos (INDA) e tem por objetivo a centralização de acesso ao maior conjunto possíveis de dados públicos governamentais. Em abril de 2015, estavam disponíveis para a consulta 859 conjuntos de dados. Nestes conjuntos, estão disponíveis vários tipos de dados, como da saúde suplementar, do sistema de transporte, de segurança pública, indicadores de educação, gastos governamentais, processo eleitoral e outros.

Como fonte para este estudo de caso, foi utilizado o conjunto de dados de compras públicas do governo federal. Tais dados abertos são do Sistema Integrado de Administração e Serviços Gerais – SIASG, onde se operacionalizam as compras do Governo Federal. Os dados estão disponíveis através de uma API de Dados Abertos em versão beta. Por meio desta API é possível acessar dados dos fornecedores, do catálogo de materiais, do catálogo de serviços, das licitações e dos contratos.

O acesso aos dados é feito através de URLs com recursos Web nos formatos XML, JSON, CSV e HTML. É possível realizar em cada consulta uma série de parâmetros de filtro que devem compor a URL. Para acessar os dados da API é necessário informar o formato de resposta (HTML, XML, JSON ou CSV) e qual informação é desejada[8]. No processo de extração dos dados escolheu-se o formato JSON como formato de resposta.

Para o objetivo do estudo de caso, as informações necessárias seriam os contratos celebrados em cada unidade administrativa (UASG) e as respectivas empresas contratadas. Tais informações podem ser obtidas utilizando-se o método de consulta Contratos disponível no módulo Contratos da API.

A API possibilita a utilização de vários parâmetros na consulta. Como cada consulta fornece informações de apenas 500 contratos, faz-se necessária a utilização do parâmetro *offset*. O parâmetro *offset* corresponde à quantidade de registros ignorados a partir do início da lista de resultados ordenados pelo ID.

Suspeitando-se que a modalidade da licitação pudesse influenciar no número de contratos realizados com uma mesma empresa, procurou-se filtrar os dados por modalidade de licitação. Para tal faz-se necessária a utilização do parâmetro modalidade. O parâmetro modalidade corresponde ao código da modalidade de licitação.

Desta forma, a API filtra os contratos por modalidade de licitação mediante a utilização das seguinte URLs:

http://compras.dados.gov.br/contratos/v1/contratos.json?modalidade=01, para contratos na modalidade Convite.

http://compras.dados.gov.br/contratos/v1/contratos.json?modalidade=02, para contratos na modalidade Tomada de Preços.

http://compras.dados.gov.br/contratos/v1/contratos.json?modalidade=03, para contratos na modalidade Concorrência.

http://compras.dados.gov.br/contratos/v1/contratos.json?modalidade=05, para contratos na modalidade Pregão.

http://compras.dados.gov.br/contratos/v1/contratos.json?modalidade=06, para contratos na modalidade Dispensa de Licitação.

http://compras.dados.gov.br/contratos/v1/contratos.json?modalidade=07, para contratos na modalidade Inexigibilidade de Licitação.

B. R-Project

Antes da descrição dos procedimentos para obtenção dos dados utilizados no estudo de caso, nesta seção são apresentadas algumas maneiras de extração de dados da Web.

A extração de dados da Web pode ser realizada em vários níveis, desde a análise de baixo nível dos formatos HTML / XML / JSON até a chamada final de funções simples implementadas por vários pacotes de conveniência, que podem por exemplo fazer uso internamente de especificações de API [5].

A linguagem R fornece uma gama de funcionalidades e recursos que possibilitam a extração e análise de dados da Web de uma maneira rápida e simples. Sua utilização cresceu bastante no mundo nos últimos anos. Por este motivo foi escolhida para a realização do estudo de caso.

R é uma linguagem de programação livre e um ambiente de *software* que fornece uma grande variedade de técnicas estatísticas e gráficas.

A CRAN (*Comprehensive R Archive Network*) é uma rede de FTP e servidores Web ao redor do mundo que armazena versões do código e documentação do R idênticos e atualizados.

Dentro da comunidade R muitos projetos contribuíram para fornecer infraestrutura que permitissem a interação do R com a Web. Recentemente, uma grande quantidade de pacotes de correspondência com tecnologias Web tem sido desenvolvida e é agora coletada, organizada e estruturada em uma vista de tarefas de tecnologias Web CRAN disponível no http://cran.r-project.org/web/views/WebTechnologies.html [5].

Os pacotes do R que permitem a extração de dados da Web em baixo nível são XML, RCurl e rjson/RJSONIO/jsonlite. O pacote XML contém funções para análise XML e HTML e suporta XPath para pesquisa em XML. Uma importante função em XML para extrair dados de uma ou mais tabelas HTML é readHTMLTable(). O pacote RCurl faz uso da biblioteca libcurl para transferência de dados usando vários protocolos, permite compor solicitações HTTP e fornece convenientes funções para busca de URLs, métodos get/post e outros, além de processar os resultados retornados pelo servidor Web. JSON (JavaScript Object Notation) é um formato de intercâmbio de dados que se tornou o mais comum da Web. Os pacotes rjson, RJSONIO, e jsonlite convertem objetos R em objetos JSON e vice-versa[5].

C. Modalidades de Licitações

No estudo de caso realizado, procurou-se distinguir os contratos celebrados por diferentes modalidades de licitação. De uma maneira sucinta são relacionadas as principais modalidades nesta seção.

A lei nº 8.666, de 21 de junho de 1993, instituiu as normas para licitação e contratos da administração pública. De acordo com a referida lei, existem várias modalidades de licitação, dentre as quais, carta-convite, tomada de preços e concorrência. A lei nº 10.520 de 17 de julho de 2002 instituiu a modalidade de licitação denominada pregão para a aquisição de bens e serviços comuns, sendo posteriormente regulamentada na forma eletrônica através do Decreto nº 450, de 31 de maio de 2005.

A dispensa de licitação foi autorizada em casos especiais de compra, sem desrespeitar os princípios da moralidade e isonomia, devendo-se limitar à aquisição de bens e serviços indispensáveis ao atendimento da situação de emergência ou dos casos descritos no artigo 24 da lei 8.666/93, dentre eles contratação de pequeno valor.

A inexigibilidade de licitação ocorre devido à inviabilidade de competição em situações descritas no artigo 25 da lei 8.666/93, como existência de fornecedor exclusivo, de empresa ou pessoa física com notória especialização, profissional artista, dentre outros.

Antes da abordagem dos dados obtidos é necessário apresentar algumas técnicas para extração de dados da Web.

Como exemplos de diferenças entre as citadas modalidades, pode-se citar o prazo entre o aviso de publicação do edital e o recebimento das propostas e os valores máximos do contrato permitido para cada modalidade.

Quanto à antecedência de publicação do edital para licitações que não envolvam técnica ou empreitada integral, os prazos mínimos são:

a) convite – 5 dias úteis;

b) tomada de preços – 15 dias;

c) concorrência – 30 dias;

d) pregão – 8 dias úteis;

Quanto ao valor do contrato, há distinção entre os objetos. Para obras e serviços de engenharia, o valor máximo de cada modalidade é maior que para compras e serviços. Os valores máximos para obras e serviços de engenharia e para compras e serviços, respectivamente, são mostrados a seguir:

a) convite - até R$ 150.000,00 (cento e cinqüenta mil reais) e R$ 80.000,00 (oitenta mil reais) ;

b) tomada de preços - até R$ 1.500.000,00 (um milhão e quinhentos mil reais ou R$ 650.000,00 (seiscentos e cinquenta mil reais);

c) concorrência - acima de R$ 1.500.000,00 (um milhão e quinhentos mil reais) ou de R$ 650.000,00 (seiscentos e cinquenta mil reais);

d) pregão – sem limite de valor.

Deve-se ressaltar que a lei 12.462, de 5 de agosto de 2011, instituiu o Regime Diferenciado de Contratações Públicas – RDC. Este regime foi inicialmente previsto para aplicação nas licitações e contratos necessários para os grandes eventos como Copa do Mundo, Copa das Confederações e Olimpíadas. Atualmente, muitas outras leis ampliaram o escopo do objeto do RDC. Neste modelo, a seleção do fornecedor se processa pela escolha da menor proposta realizada por meio de lances públicos e diferencia-se das modalidades de licitação, citadas anteriormente, dentre outros fatores, pelo fato de que os concorrentes não conhecem o orçamento, nem possuem o projeto executivo da obra, mas sim, no caso da contratação integrada, um projeto básico que caracteriza a obra com base nas indicações dos estudos técnicos preliminares.

No estudo de caso realizado foram utilizados somente os contratos relacionados às modalidades de licitação: carta-convite, tomada de preços, concorrência, pregão, dispensa de licitação e inexigibilidade de licitação.

D. Extração de Dados

Nesta seção é apresentada a metodologia utilizada para a extração de dados do estudo de caso.

Para a extração de dados foram utilizados os pacotes RCurl e RJSONIO.

Utilizou-se para consulta a seguinte URL descrita anteriormente:

"http://compras.dados.gov.br/contratos/v1/contratos.json"

Para fornecimento dos parâmetro de consulta e obtenção do objeto JSON, utilizou-se a seguinte função:

getForm(url, .params = list (modalidade = mod, offset = as. integer(off))), onde mod e off representam os valores dos parâmetros de consulta.

Para transformação do objeto JSON em objeto R, utilizou-se a função fromJSON().

Para a obtenção dos dados em objeto R, foi necessária também a implementação de procedimentos para padronizar suas estruturas de dados e automatizar a consulta em todas as páginas correspondentes à modalidade de licitação desejada. Tais procedimentos não estão descritos neste artigo.

Ao final do processo de extração dos dados, foram obtidos 6 *dataframes,* correspondentes aos contratos constantes do portal de dados abertos brasileiro, divididos por modalidade de licitação (Tabela I).

TABELA I- DADOS EXTRAÍDOS

Dataframe	Modalidade de Licitação	Número Contratos
1	carta-convite	34.009
2	tomada de pre?os	39.394
3	concorr?ncia	24.035
4	preg?o	219.647
5	dispensa de licita??o	89.478
6	inexigibilidade	45.378
TOTAL		**451.941**

Cada Dataframe era constituído por 17 campos contendo as seguintes informações (Tabela II):

TABELA II – CAMPOS DOS DATAFRAMES

Nº	Campo	Descrição
1	identificador	Identificador do Contrato
2	uasg	Campo de seis digitos que indica o código da UASG contratante.
3	modalidade_licitacao	Número e o ano da licitação que originou a contratação.
4	numero_licitacao	Número do aviso da licitação que originou a contratação.
5	tipo_contrato	Tipo de Contrato.
6	licitacao_associada	Referência à licitação que originou a contratação.
7	origem_licitacao	Origem da licitação que gerou o contrato: Preço praticado(SISPP) ou Registro de preço(SISRP).
8	numero	Campo seguido pelo número do contrato, seguido do respectivo ano.
9	objeto	Descrição do objeto, a partir de uma descrição de item/serviço informada.
10	numero_aditivo	Quantidade de termos aditivos de um contrato.
11	numero_processo	Número do processo de contratação.
12	cnpj_contratada ou cpf_contratada	CNPJ da empresa contratada ou CPF da contratada, varia de acordo com a modalidade de licitação
13	data_assinatura	Data de assinatura do contrato.
14	fundamento_legal	Fundamento legal do processo de contratação.
15	data_inicio_vigencia	Data de início de vigência dos contratos.
16	data_termino_vigencia	Data de término de vigência dos contratos.
17	valor_inicial	Valor inicial do contrato.

E. Análise dos Dados

Após a extração dos dados passou-se ao tratamento e análise dos mesmos.

Inicialmente identificaram-se os campos necessários para a busca dos padrões de acordo com a descrição da hipótese, ou seja, o CNPJ ou CPF do fornecedor contratado e a unidade administrativa contratante (UASG).

Em seguida, para cada *dataframe*, elaborou-se um novo *dataframe* compreendendo apenas os campos UASG, CNPJ ou CPF contratado (conforme modalidade de licitação e o número de contratos celebrados ente os dois. Para tal utilizou-se a seguinte função R:

ddply(dataframe, c("uasg","cnpj_contratada"), summarise,
ContrPorEmpresa=length(modalidade_licitacao)
),

onde *ContrPorEmpresa* passou a ser o campo do novo *dataframe* que informa a quantidade de contratos celebrados entre uma unidade administrativa e uma mesma empresa.

Finalmente elaborou-se um último *dataframe*, a partir do anterior, compreendendo o campo UASG e criando os campos *NumEmpresas* (número de empresas contratadas por UASG), *NumContratos* (número total de contratos celebrados por cada UASG) e *MaiorNumContMesmaEmpresa* (maior número de contratos que cada UASG celebrou com uma mesma empresa). Para tal foi utilizada a seguinte função R:

ddply(dataframe1, c("uasg"), summarise,
NumEmpresas = length(ContrPorEmpresa),
NumContratos=sum(ContrPorEmpresa),
MaiorNumContMesmaEmpresa=max(ContrPorEmpresa)
)

Cada *dataframe* final, relativo a cada modalidade de execução, obteve os seguintes dados (Tabela III):

TABELA III – DADOS FINAIS PARA ANÁLISE

Dataframe	Modalidade de Licitação	Número Uasgs	Número Contratos	Número Empresas	M. Num. Cont. Mesma Empr.
1	carta-convite	1.773	34.009	24.055	43
2	tomada de pre□os	1.781	39.394	25.343	50
3	concorr□ncia	1.247	24.035	13.217	43
4	preg□o	2.429	219.647	131.301	61
5	dispensa de licita□□o	2.590	89.478	49.729	599
6	inexigibilidade	2.160	45.378	24.626	207

Para a verificação da correlação entre o número de contratos de uma unidade administrativa (uasg) e o máximo de contratos celebrados com uma mesma empresa por essa unidade administrativa, foram estudadas as formas de regressão linear e não-linear.

Para a regressão linear, buscou-se encontrar os coeficientes que produzissem a melhor aproximação, utilizando-se o critério dos mínimos quadrados, conforme a seguinte função de regressão:

$$Y \approx \beta_0 + \beta_1 X$$

$$\text{MaiorNumContMesmaEmpresa} \approx \beta_0 + \beta_1 \text{NumContratos}$$

Verificando-se a regressão linear para os contratos da modalidade de licitação carta-convite, obteve-se os seguintes coeficientes (Fig.1):

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.496936   0.052235   28.66   <2e-16 ***
NumContratos 0.054072   0.001238   43.69   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.959 on 1771 degrees of freedom
Multiple R-squared:  0.5187, Adjusted R-squared:  0.5184
F-statistic:  1909 on 1 and 1771 DF,  p-value: < 2.2e-16
```

Fig.1 – Coeficientes da regressão linear na Modalidade Carta-convite

Percebe-se que o valor p é praticamente nulo. Além disso os coeficientes são bem maiores que o desvio padrão, levando à obtenção de valores t grandes. Tais condições afastam a hipótese dos coeficientes β_0 e β_1 serem nulos.

Além disso, obtém-se um coeficiente de determinação $R^2 = 0{,}51$. Tal coeficiente indica a proporção de variabilidade presente nas variações da variável reposta Y que é explicada pela variável independente X no modelo de regressão.

Assim os coeficientes usados no modelo de regressão linear, para os contratos da modalidade carta-convite, foram:

$\beta_0 = 1.497$

$\beta_1 = 0.054$

Procurando-se agora verificar se um modelo não linear se ajustaria melhor, realizou-se a comparação da regressão linear com funções polinomiais até o grau 5,:

$$Y \approx \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \beta_4 X^4 + \beta_5 X^5$$

```
> anova(lmMnunemp.1,lmMnunemp.2,lmMnunemp.3,lmMnunemp.4,lmMnunemp.5)
Analysis of Variance Table

Model 1: MaiorNumContMesmaEmpresa ~ NumContratos
Model 2: MaiorNumContMesmaEmpresa ~ poly(NumContratos, 2)
Model 3: MaiorNumContMesmaEmpresa ~ poly(NumContratos, 3)
Model 4: MaiorNumContMesmaEmpresa ~ poly(NumContratos, 4)
Model 5: MaiorNumContMesmaEmpresa ~ poly(NumContratos, 5)
  Res.Df    RSS Df Sum of Sq       F    Pr(>F)
1   1771 6797.5
2   1770 5860.8  1    936.68 294.957 < 2.2e-16 ***
3   1769 5763.0  1     97.84  30.810 3.277e-08 ***
4   1768 5693.2  1     69.77  21.969 2.983e-06 ***
5   1767 5611.4  1     81.84  25.770 4.251e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Fig.2 – Comparação entre polinômios de diferentes graus.

Pelos valores p e F, percebe-se que os modelos de 3º,4º e 5º graus apresentam um ajuste melhor que a regressão linear simples. Para o modelo com polinômio de 5º grau, o coeficiente de determinação, R^2, é igual a 0,60, ou seja, superior ao da regressão linear simples (Fig.2). A Figura 3 a seguir apresenta os diferentes modelos no gráfico:

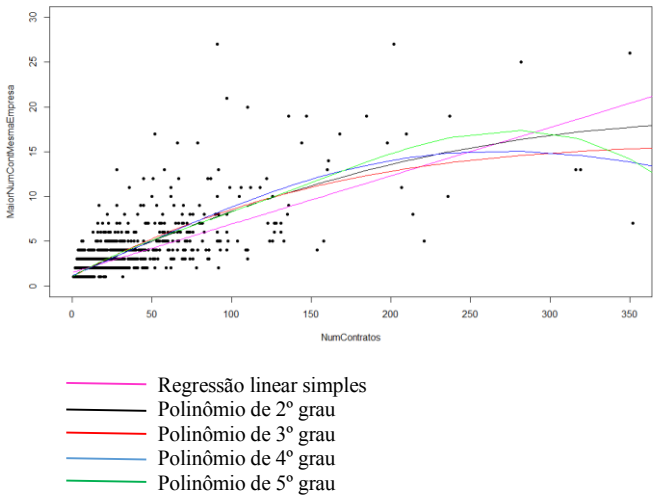


Fig.3 – Modelos de ajuste.

No entanto, devido à proximidade com a regressão linear simples no intervalo de números de contrato contido entre 1 e 150 contratos (maior concentração de unidades administrativas), optou-se pela realização da comparação entre as diversas modalidades de licitação mediante a utilização da regressão linear simples.

A partir dos dados obtidos, foram elaborados os seguintes gráficos para cada modalidade de licitação (Figuras 4 a 9), onde a linha central representa a função de regressão e as linhas verdes as margens superior e inferior de acordo com o nível de tolerância.

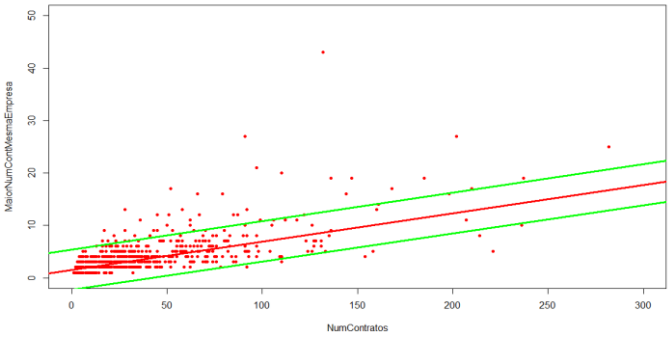


Fig.4 – Modalidade Carta-convite

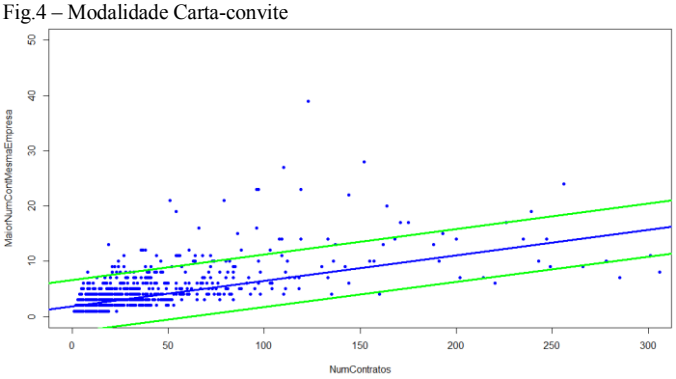


Fig.5 – Modalidade tomada de preços

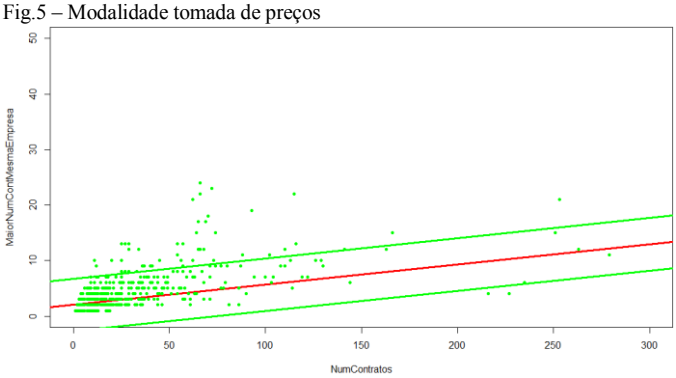


Fig.6 – Modalidade concorrência

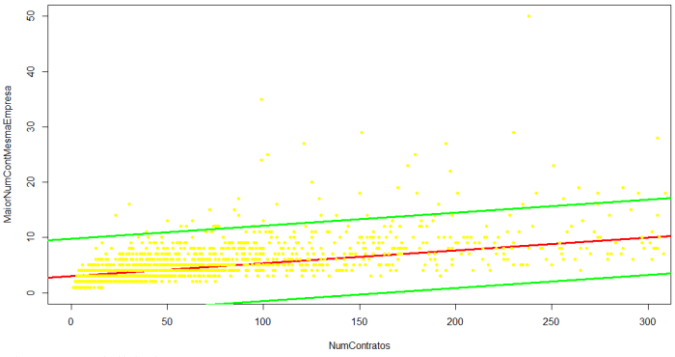


Fig.7 – Modalidade pregão

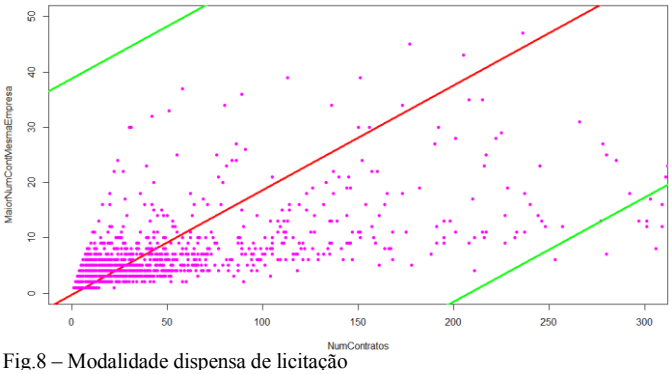


Fig.8 – Modalidade dispensa de licitação

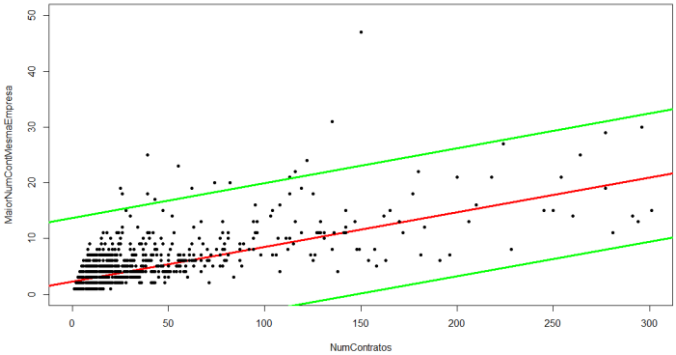


Fig.9 – Modalidade inexigibilidade de licitação

Pelos gráficos anteriores pode-se perceber, para todas as modalidades de licitação, a existência de uma correlação entre o maior número de contratos celebrados com uma mesma empresa e o número total de contratos da unidade administrativa. Ressalta-se que na modalidade dispensa de licitação existe uma dispersão maior.

Comparando-se agora todas as funções de regressão de acordo com as modalidades de licitação obtêm-se o seguinte gráfico (Figura 10):

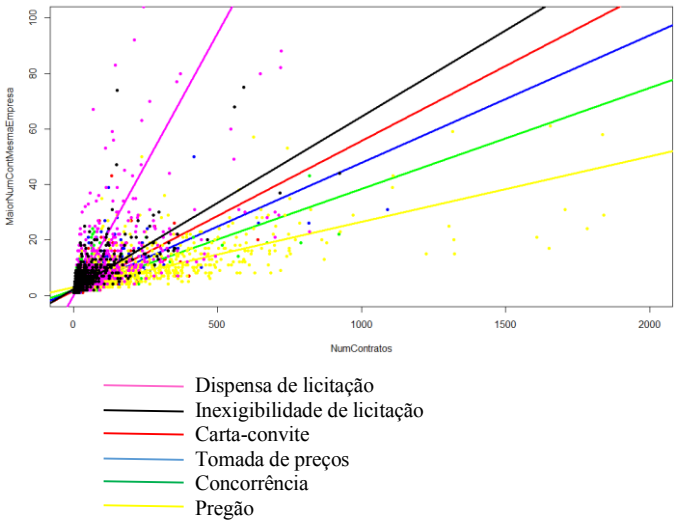


Fig.10 – Comparação entre as modalidades de licitação

F. Resultados Obtidos

A partir da análise da correlação entre os contratos celebrados com uma mesma empresa e o número de contratos celebrados por uma unidade administrativa foi possível estabelecer uma correlação entre as mesmas.

Verificou-se que o número de contratos celebrados com uma mesma empresa possui uma correlação diferente com número

total de contratos da unidade administrativa, variando de acordo com a modalidade de licitação escolhida.

Tal variação pode ser explicada pelas diferentes maneiras de seleção da empresa a ser contratada. Na dispensa de licitação e inexigibilidade de licitação não existe certame, sendo a empresa selecionada diretamente pela unidade administrativa e, portanto, mais fácil de haver uma possível preferência. A modalidade de licitação carta-convite prevê a disputa entre três empresas escolhidas livremente pela administração, o que também diminui a possibilidade de direcionamento, mas ainda possibilita uma certa preferência. Nas demais modalidades de licitação, a exigência de divulgação do edital aumenta, sendo, na modalidade pregão, a disputa realizada através da internet, o que dificulta o direcionamento pela unidade administrativa.

Por outro lado, as modalidades de licitação são realizadas de acordo com o valor, sendo, de uma maneira geral, a dispensa de licitação realizada para os contratos de menor valor e a concorrência para os contratos de maior valor. Outra explicação para os diferentes modelos de regressão poderia ser o aumento do interesse das empresas em licitações de maior valor e consequentemente maior distribuição dos contratos.

Observa-se, assim, que quanto maior a exigência de divulgação do edital de licitação e quanto maior o valor do contrato (modalidade de licitação) menor é a correlação entre o número máximo de contratos celebrados com uma mesma empresa e o número total de contratos celebrado com uma mesma unidade administrativa.

O estudo encontra-se em caráter preliminar e não foram exauridas todas as formas de ajustes possíveis para caracterização do melhor modelo. Da mesma forma, muitas outras informações dos contratos poderiam ser utilizadas para formação deste modelo, como valor do contrato, data de assinatura e outros. No entanto pode-se destacar que é possível inferir um resultado a partir de reconhecimento de padrões estatísticos.

Tal inferência poderia contribuir para a identificação de possíveis irregularidades na contratação, pois a existência de valores muito superiores aos padrões obtidos poderia levantar a suspeita de ocorrência de favorecimento ou direcionamento de licitação. Um tratamento mais refinado nos dados obtidos poderiam, assim, apontar para os órgãos de controle quais unidades administrativas e empresas mereceriam uma análise pormenorizada dos contratos, por exemplo.

III. CONCLUSÕES

A partir da extração de dados da Web relativos a licitações e contratos públicos é possível obter inferências por reconhecimento de padrões estatísticos.

Como importante fonte de dados relacionados a contratos e licitações para obtenção de padrões estatísticos, pode-se utilizar o portal de dados abertos brasileiro, www.dados.gov.br. Além de licitações e contratos, tal portal fornece uma grande quantidade de informações relacionadas a vários tipos de dados como da saúde suplementar, do sistema de transporte, de segurança pública, indicadores de educação, gastos governamentais, processo eleitoral e outros.

Em um estudo de caso preliminar para obter inferências por reconhecimento de padrões estatísticos entre o número de contratos celebrados por uma unidade administrativa com uma mesma empresa e o número total de contratos celebrados por esta unidade administrativa, foi possível estabelecer uma correlação entre os mesmos.Com a elaboração de um modelo de regressão linear, o estudo de caso explicitado neste artigo, utilizando a linguagem R para extração de dados relacionados a licitações e contratos do portal de dados abertos brasileiros, comprovou que quanto maior a exigência de divulgação de dados do edital e quanto maior o valor do contrato, menor é a correlação entre o número de contratos celebrados entre uma unidade administrativa e uma mesma empresa e o número total de contratos desta unidade.

O estudo de caso abordou apenas três variáveis de dezessete variáveis extraídas dos contratos. Variáveis como valor do contrato, data de assinatura, número de aditivos e outros não fizeram parte do modelo apresentado.

Muitas outras informações, correlações e padrões entre licitações e contratos, não abordados neste artigo, podem ser objeto de estudos.

A partir do conhecimento destes padrões estatísticos, a observância de grandes desvios poderia apontar aos órgãos de controle a suspeita de possíveis irregularidades.

Acrescenta-se que o surgimento de nova modalidade de licitação, como o Regime Diferenciado de Contratação – RDC, que flexibiliza a obrigatoriedade de existência prévia de projeto executivo e orçamento detalhado para contratação de obras e serviços de engenharia, torna ainda mais necessário o conhecimento prévio de padrões estatísticos de contratos similares.

AGRADECIMENTOS

Aos professores do mestrado profissional em engenharia elétrica da UNB pelas orientações, à família e aos colegas pelo constante incentivo e motivação.

REFERÊNCIAS

[1] MUNZERT, Simon et al. Automated Data Collection with R: A Practical Guide to Web Scraping and Text Mining. John Wiley & Sons, 2014.

[2] HASTIE, Trevor et al. An Introduction to Statistical Learning with Applications in R. 2013.

[3] DOS SANTOS BRITO, Kellyton et al. Brazilian government open data: implementation, challenges, and potential opportunities. In: Proceedings of the 15th Annual International Conference on Digital Government Research. ACM, 2014. p. 11-16.

[4] SEERS, Blake M.; SHEARS, Nick T. New Zealand's Climate Data in R—An Introduction to clifro. 2015.

[5] MAIR, Patrick; CHAMBERLAIN, Scott. Web Technologies Task View. A peer-reviewed, open-access publication of the R Foundation for Statistical Computing, p. 178, 2014.

[6] VARELA, Sara et al. rAvis: an R-package for downloading information stored in Proyecto AVIS, a citizen science bird project. PloS one, v. 9, n. 3, p. e91650, 2014.

[7] COSTA, Jefferson de J. et al. Uma análise da qualidade dos dados relativos aos boletins de ocorrências das rodovias federais brasileiras para o processo de Mineração de Dados, 2013.

[8] _____, "Dados Abertos Governamentais" (2012). [Online]. Disponível em: http://www.governoeletronico.gov.br/acoes-e-projetos/Dados-Abertos. Acessado em: 27 abr. 2015.

[9] Karin Breitman, Percy Salas, Marco Antonio Casanova, Daniel Saraiva, Vinicius Gama, Jose Viterbo, Regis Pires Magalhães, Ednylton Franzosi, Miriam Chaves, "Open Government Data in Brazil", IEEE Intelligent Systems, vol.27, no. 3, pp. 45-49, May-June 2012, doi:10.1109/MIS.2012.25.

[10] Abu-Mostafa, Y; Magdon-Ismail, M. & Lin, H. Learning from Data, AMLBook, EUA, 2012.

DOI: 10.5769/C2015014 or http://dx.doi.org/10.5769/C2015014

Catálogo de Fraudes da RNP: 7 anos de experiência no tratamento de fraudes eletrônicas brasileiras

Italo Brito, José Lucas Borges, Lucas Ayres, Paula Tavares, Rogerio Bastos [1]
Edilson Lima, Liliana V. Solha [2]

***Resumo*— As fraudes eletrônicas disseminadas na Internet tornaram-se uma ameaça constante para a população em geral. A cada dia novas e mais sofisticadas técnicas de fraudes são empregadas, levando usuários menos preparados ou atentos à serem vítimas desses ataques. Criado em 2008, o Catálogo de Fraudes da Rede Nacional de Ensino e Pesquisa (RNP), consolida-se como um importante repositório de fraudes eletrônicas brasileiras disseminadas por e-mail. Este artigo apresenta o funcionamento do Catálogo de Fraudes da RNP, estatísticas e tendências observadas, além de oportunidades de trabalho que podem ser desenvolvidos para melhorar a segurança dos usuários de Internet brasileiros.**

***Palavras-Chave*— Fraudes, Catálogo de Fraudes, fraude eletrônica, e-mail, *phishing*, *ICCyber*.**

***Abstract*— The electronic frauds all over the Internet have become a recurring threat to all the people. Everyday, newer and more sophisticated fraud techniques are deployed, causing the less prepared or the less alerted users to be victims of those attacks. The Frauds Catalog of the Brazilian Academic and Research Network (RNP), created in 2008, consolidates itself as an important repository of brazilian electronic frauds disseminated through e-mail. This paper presents RNP Frauds Catalog, how it works, observed statistics and trends, and future work opportunities that can improve the security of Brazilian Internet users.**

***Keywords*— Frauds, frauds catalog, eletronic fraud, e-mail, *phishing*, *ICCyber*.**

I. Introdução

Comunicação sempre foi o principal fator no estabelecimento de relações. Diversas formas de comunicação fáceis e de grande agilidade apresentam-se nos dias atuais e o e-mail continua sendo um dos principais meios de comunicação digital. Apesar de ter sofrido uma desaceleração nos últimos anos, o número de contas de e-mail continua aumentando, bem como o volume de mensagens transitadas. Atribui-se como motivo desse contínuo crescimento, a facilidade de uso e a maior formalidade e confiança atribuída à mensagem de e-mail, se comparada a outros meios de comunicaçao. Em 2014, segundo resultados apresentados pelo grupo Radicati [1], cerca de 4,1 bilhões de contas de e-mails sendo utilizadas em todo mundo e 108,7 bilhões de mensagens trafegando por dia.

Embora seja um serviço de comunicação utilizado em larga escala desde o final da década de 90, as ferramentas de e-mail ainda possuem, em sua maioria, uma verificação do conteúdo precária, o que se presta a um possível mal uso. Por esse motivo, o e-mail é bastante utilizado para disseminação de fraudes eletrônicas [2]. As fraudes frequentemente apelam para a inocência, tal como a não validação de uma suposta mensagem de seu banco solicitando dados pessoais; ou curiosidade dos usuários, por exemplo, a possibilidade de ver fotos exclusivas de um evento recentemente noticiado pela mídia.

Os e-mails fraudulentos estão, em sua maioria, relacionados a golpes digitais como o *phishing*. O *phishing* ocorre quando o golpista, utilizando-se de meios digitais e de engenharia social, tenta obter dados pessoais, senhas ou informações financeiras da vítima. O usuário pode ser seduzido de diversas formas, como por exemplo por meio de uma página de compras em promoção, solicitações de atualização ou recadastramentos, que caso não ocorram acarretarão em prejuízo ao usuário, ou ainda como promoções relacionadas a cartões de crédito, a companhias aéreas ou outras envolvendo o preenchimento de formulários, levando o usuários desavisados a clicarem em links falsos criados pelo atacante.

Muitas vezes o link falso dá origem ao *download* de um *malware*, um tipo de programa desenvolvido unicamente para a execuçao de ações maliciosas. Ao ser instalado ou executado no computador da vítima, o *malware* pode iniciar funções tais como envio de spam, roubo de informações confidenciais dos usuários ou até mesmo realizar ataques contra outras máquinas.

O Catálogo de Fraudes da RNP foi criado em 2008, com o objetivo de coletar fraudes recebidas por e-mail pela população em geral e analisar, filtrar e catalogar essas fraudes, criando um repositório de mensagens conhecidamente fraudulentas e alertando a comunidade sobre como se proteger desse tipo de ataque. Criado pelo Centro de Atendimento a Incidentes de Segurança da RNP (CAIS/RNP) e mantido atualmente em parceria com o Ponto de Presença da RNP na Bahia (PoP-BA/RNP), até onde se tem conhecimento, o Catálogo de Fraudes da RNP é a única fonte de informações aberta e online sobre fraudes eletrônicas no Brasil, sendo bastante utilizado pela população em geral como uma base de conhecimento para validação de mensagens de e-mail suspeitas.

Este artigo está estruturado em seis seçoes, sendo esta Seçao da Introduçao a primeira. Na Seção II são apresentados outros trabalhos, todos internacionais, que abordam a análise e registro de fraudes eletrônicas. Já na Seção III, discute-se o processo de tratamento de fraudes, detalhando-se as etapas deste processo, que incluem: recebimento, triagem, interação com a fraude e catálogo. A Seção IV traz algumas estatísticas

[1]Italo Brito, José Lucas Borges, Lucas Ayres, Paula Tavares, Rogerio Bastos. Ponto de Presença da RNP na Bahia- PoP-BA/RNP, Universidade Federal da Bahia - UFBA, Salvador-BA, E-mails: {italo,lucasborges,lucasayres,paulatavares,rogeriobastos}@pop-ba.rnp.br. [2]Edilson Lima, Liliana V. Solha. Centro de Atendimento a Incidentes de Segurança da RNP - CAIS/RNP, Campinas-SP, E-mails: {edilson.lima,liliana.solha}@rnp.br.

e tendências observadas no tratamento das fraudes e a Seção V apresenta alguns benefícios que o Catálogo de Fraudes proporciona para a comunidade em geral. Por fim, a Seção VI conclui e apresenta trabalhos futuros para esta iniciativa.

II. Trabalhos Relacionados

Os trabalhos anteriores na área de fraudes eletrônicas (*phishing*) focam em mecanismos de detecção automática de fraudes, técnicas utilizadas pelos atacantes e relatórios de atividades envolvendo *phishing*.

Detecção automática é um assunto bastante abordado em artigos dessa área. Basnet et al. em [7] tentou criar um sistema de detecção automática baseado em regras. Estas regras levam em consideração diversas características, como IPs nas URLs, palavras-chave comuns a sites fraudulentos, número de caracteres especiais na URL, envio de formulários utilizando TLS/SSL, presença na blacklist do Google Safe Browsing [5], número de pontos e tamanho da URL, entre outras. Fette et al. em [8] propõe um método de classificação de e-mails baseado no potencial que cada e-mail tem de ser isca para um ataque de *phishing*. Seu algoritmo de detecção leva em conta características únicas identificadas nos emails de *phishing* e a saída de um filtro de spam, que eles identificaram ser bastante eficiente para este tipo de detecção.

McGrath et al. em [9] examina o modus operandi dos atacantes através da anatomia das URLs e domínios dos sites fraudulentos, do registro e tempo de ativação dos domínios de *phishing*, e das máquinas utilizadas para hospedar tais sites. Os resultados podem ser utilizados como heurísticas na filtragem de e-mails de *phishing* e na identificação de registros de domínios suspeitos. Garera et al. em [10] estuda as URLs utilizadas em ataques de *phishing* e tenta descobrir se elas realmente pertencem a um ataque de *phishing*, sem utilizar qualquer conhecimento da página em si. Ele identificou algumas técnicas utilizadas pelos *phishers* para enganar as vítimas, como mascarar o *host* com um endereço de IP, mascarar o *host* com outro domínio, criar domínios similares ao de organizações conhecidas e criar URLs muito grandes para confundir a vítima.

A RSA [11] produz relatórios mensais da quantidade de ataques envolvendo *phishing* e suas principais características. O Anti-Phishing Working Group (APWG) publica relatórios trimestralmente [12] com as tendências de ataques de *phishing*, informando a quantidade de marcas utilizadas como alvo pelos *phishers*, os setores mais atacados da indústria, os países que mais hospedam sites de *phishing* e cavalos-de-tróia utilizados nestes ataques, e a distribuição dos *phishings* por TLD (top-level domain).

III. Processo de tratamento

Desde 2008, o Centro de Atendimento a Incidentes de Segurança da RNP (CAIS/RNP) mantém uma base de dados de fraudes eletrônicas encaminhadas por usuários da Internet. Todos os e-mails recebidos pelo CAIS apontados como alerta de uma potencial fraude são analisados e catalogados em uma ferramenta web, disponibilizando as informações coletadas como fonte de consulta através do site da iniciativa[1]. O objetivo do catálogo é, portanto, apoiar a comunidade brasileira na identificação e conscientização sobre os principais golpes eletrônicos que estão sendo veiculados na Internet.

A partir dos e-mails recebidos pelo CAIS, é realizada uma triagem inicial. Atualmente, cerca de 15.000 mensagens são tratadas a cada mês, desse total são descartados os spams e as mensagens em língua estrangeira, posteriormente, descarta-se as mensagens repetidas. Com isso, são catalogadas uma média de 200 novas fraudes por mês. Após a mensagem ser classificada como fraude, inicia-se o processo de identificação das principais características do e-mail, tais como o uso de redirecionamento para sites falsos e/ou a presença de arquivos maliciosos em anexo ou disponíveis para *download*.

São registrados, no Catálogo de Fraudes, o corpo do e-mail na forma de texto e imagem através de captura de tela. As mensagens que direcionam o usuário para sites fraudulentos, também têm as páginas do site registradas como imagem, para isso é feito uma interação com esses sites, a fim de coletar o maior número de informações. Quando há um *malware* anexado ou disponível para *download*, é utilizada a ferramenta VirusTotal [3] para análise do arquivo malicioso, sendo registrado o nome do arquivo que está relacionado a sua ação em uma máquina, como por exemplo "trojan" que dá acesso a usuários maliciosos à máquina da vítima e *malware* contendo a palavra "Win" geralmente são direcionados a máquinas com sistema operacional Windows. Estes dados são utilizados para melhor informar o usuário.

As informações, tais como assunto da mensagem, tipo, classificação, nome do *malware*, hash md5 do *malware* são registradas no catálogo juntamente com as imagens. As principais tarefas relacionadas ao Catálogo de Fraudes são atualmente executadas manualmente, no entanto, objetivando aumentar a eficiência e os resultados obtidos, encontram-se em desenvolvimento novas ferramentas para a automatização de algumas etapas deste processo. Estas melhorias visam tornar este processo mais eficaz e possibilitar a identificaçao de um número maior de fraudes.

IV. Experiências obtidas na manutenção do Catálogo de Fraudes

Ao longo desses sete anos de tratamento e análise de fraudes, foi possível observar diversas tendências no comportamento dos e-mails fraudulentos. O entendimento dessas tendências oferece condições de desenvolver guias de recomendações e ferramentas automatizadas que possam ajudar no combate à disseminação dessas fraudes. Essa seção apresenta algumas estatísticas, tais como: quantidade de e-mails recebidos, número de fraudes analisadas, número de fraudes por categoria, etc. Assim também, são apresentados resultados de análises de URLs maliciosas e tendências observadas no Catálogo de Fraudes da RNP.

A. Estatísticas de fraudes catalogadas

Os e-mails recebidos passam por uma triagem e são classificados como: spam, fraude ou mensagem em língua estrangeira.

[1] http://www.rnp.br/servicos/seguranca/catalogo-fraudes

Grande parte dos e-mails encaminhados por usuários são mensagens de spam. A figura 1 contrasta a quantidade total de e-mails recebidos com a quantidade de mensagens classificadas como fraudes ou mensagens em língua estrangeira, no período de abril de 2014 a abril de 2015.

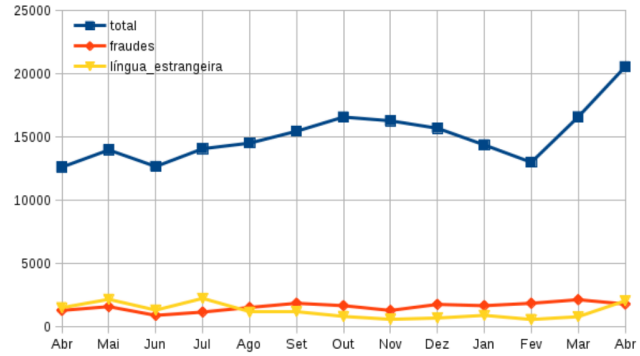


Fig. 1. E-mails tratados em 2014/Abr - 2015/Abr

Com base nos assuntos abordados nas mensagens classificadas como fraude, estas são categorizadas da seguinte maneira: bancos, documentos, cartões de crédito, e-commerce, empresas aéreas, governo, redes sociais e internet, seguradoras, serviços de pagamento, e outras.

O gráfico da figura 2 mostra o acumulado de fraudes por categoria, no período de abril de 2014 a abril de 2015. As fraudes mais frequentes exploram assuntos relacionados às instituições bancárias, reunidas na categoria bancos, e a transações financeiras, tais como pagamento de boletos e faturas, cheques, comprovantes de depósitos e transferências, orçamentos, dentre outros, reunidos na categoria documentos.

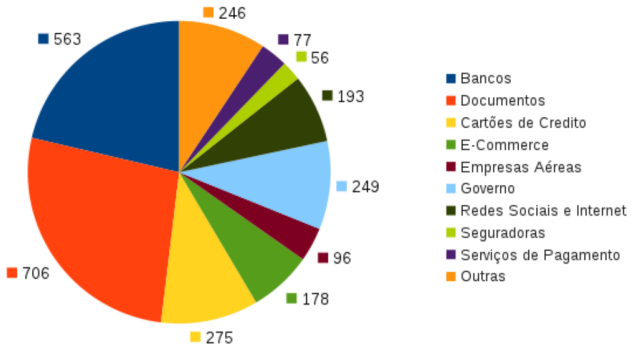


Fig. 2. Categorização das fraudes catalogadas em 2014/Abr - 2015/Abr

Também se destacam-se as fraudes relacionadas a empresas de cartões de crédito e fraudes que utilizam o nome de instituições governamentais. É comum temas como intimações judiciais, serviços postais, cadastro em promoções de empresas de cartões de crédito, dentre outros.

B. Análise de URLs maliciosas

Analisando-se os e-mails de fraudes, constata-se que cerca de 90% contêm URLs para sites ou arquivos maliciosos no corpo da mensagem. Diante dessa informação, verificou-se que o bloqueio das URLs maliciosas é um importante mecanismo de proteção para os usuários.

Atualmente, há diversos serviços de reputação de URLs. O Google Safebrowsing [5] é o serviço do Google para verificação de URLs maliciosas. Navegadores web como o Firefox, Google Chrome e Safari utilizam esse serviço para alertar os usuários quando as mesmas são acessadas.

O Phishtank [4] é uma comunidade baseada no serviço anti-phishing, usado por empresas como Opera, WOT, Yahoo Mail para verificar se uma URL é considerada *phishing*. A Microsoft oferece o filtro SmartScreen [6] para os aplicativos do seu sistema operacional, contudo não disponibiliza uma API de consulta pública.

Utilizando as APIs de consulta dos serviços Google Safebrowsing e Phishtank realizamos uma análise das URLs encontradas nas fraudes catalogadas. As figuras 3 e 4 mostram a quantidade de URLs analisadas e a quantidade de URLs detectadas como maliciosas nos serviços Google Safebrowsing e Phishtank.

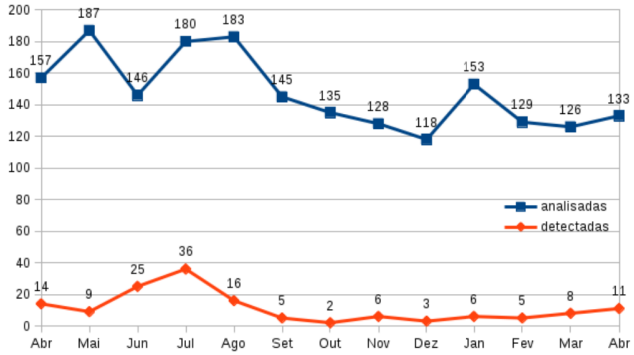


Fig. 3. Análise de URLs no Google Safebrowsing

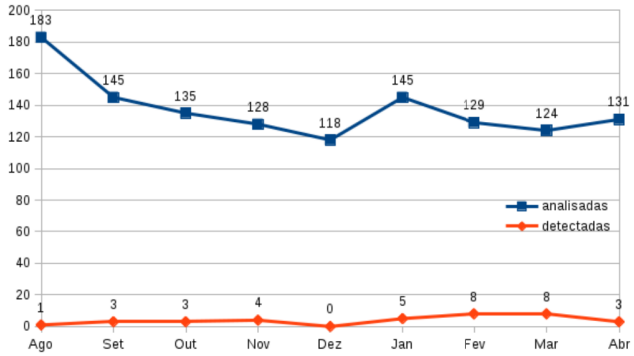


Fig. 4. Análise de URLs no Phishtank

Como pode ser observado nos gráficos apresentados, os serviços de reputação avaliados são pouco eficientes para as URLs utilizadas nas fraudes destinadas aos usuários brasileiros. Isso mostra a necessidade de um serviço direcionado a estes usuários. Diante dessa demanda, a RNP iniciou o desenvolvimento de um serviço de verificação de URLs maliciosas voltado para a comunidade brasileira.

C. Principais fraudes

Similar às campanhas de spam, as fraudes eletrônicas também apresentam campanhas direcionadas a determinados acontecimentos ao longo do ano. No ano de 2014 e inclusive em 2015, puderam ser identificadas algumas fraudes que se destacaram por abordarem assuntos relacionados a acontecimentos de relevância nacional.

No período da Copa do Mundo este tema foi largamente abordado relacionado a promoções e compra de ingressos que redirecionava o usuário para páginas falsas que requeriam seus dados pessoais e financeiros. Na época das eleições também foi grande a quantidade de *phishings* abordando temas como cadastramento biométrico, regularização do título eleitoral, falsas notícias sobre os candidatos à presidência, processo de seleção de mesários para zonas eleitorais e listas a favor de um Impeachment, fraudes que, em sua maioria continham arquivos maliciosos utilizados para infectar a máquina do usuário. O ENEM foi utilizado como tema em diversas fraudes que requeriam o recadastramento devido a irregularidades na inscrição, porém, a URL citada na suposta mensagem do ENEM levava o usuário a executar inadivertidamente um *malware*. Assuntos como imposto de renda deram origem a fraudes contendo falsos documentos.

Algumas campanhas fraudulentas apresentam-se regularmente durante todo o ano. Fraudes relacionadas a bancos são enviadas cotidianamente solicitando atualizações e recadastramentos para que supostamente não ocorra bloqueio da conta ou indisponibilização de serviços. Fraudes associadas a comprovantes, boletos, notas fiscais, faturas, pedidos de orçamento estão relacionadas em sua maioria com arquivos maliciosos. Essas fraudes que se repetem ao longo do ano possuem maiores chances de ludibriar a vítima por retratar assuntos comuns, de forma simples e de interesse pessoal.

V. Benefícios do catálogo

O Catálogo de fraudes é uma importante contribuição da RNP tanto para a comunidade acadêmica como também à comunidade brasileira em geral. As fraudes ali catalogadas, por serem rigorosamente analisadas, são uma fonte confiável de informações sobre fraudes eletrônicas brasileiras. Os usuários, em geral, podem fazer uso dessas informações para consultar por período, identificar campanhas de fraudes, e também comparar o texto e imagem das mensagens recebidas com fraudes conhecidas.

Como consequência do trabalho com as fraudes eletrônicas, alguns guias de boas práticas, respostas para perguntas frequentes e cartilhas de segurança foram desenvolvidas e disponibilizadas no site da iniciativa[2].

Já para a comunidade de Segurança da Informação, esse catálogo pode ser usado para maior entendimento das fraudes direcionadas ao público brasileiro, uma vez que a maioria dos trabalhos anteriores apresenta dados de fraudes internacionais. Nota-se, inclusive, a carência por repositórios dessa natureza aqui no Brasil, de forma que possam ser usados por ferramentas de segurança automatizadas para evitar que os usuários se tornem vítimas (ex: filtros de conteúdo, *plugins* de navegadores web, etc). Neste sentido, é interesse da RNP incorporar ao Catálogo de Fraudes o registro das URLs utilizadas nas fraudes, agregando um mecanismo de reputação às URLs de forma que possam ser usadas em ferramentas de clientes - este trabalho atualmente encontra-se em andamento. Por outro lado, a análise dessas URLs pode desvendar padrões, comumente utilizados pelos atacantes, para iludir seus usuários, comparando-os com características das fraudes identificadas ao redor do mundo.

[2]http://www.rnp.br/servicos/seguranca/educacao-e-conscientizacao-seguranca

VI. Conclusões

O crescimento na disseminação de fraudes eletrônicas, via e-mail, redes sociais e outras mídas eletrônicas, associado com a falta de conhecimento e até discernimento de muitos usuários na Internet, evidenciam a necessidade e a importância de uma base de conhecimento das fraudes eletrônicas, que possa ser usada não apenas como uma fonte de consulta e validação de mensagens desconhecidas, mas também que possa ser usada pelas organizações na implantação de filtros e outros mecanismos de segurança a fim de evitar que os usuários sejam vítimas desse tipo de ataque.

Este artigo apresentou o processo de análise, triagem e registro do Catálogo de Fraudes da RNP, bem como algumas estatísticas e tendências observadas nesse processo. É notório observar o crescimento na quantidade das fraudes e também o nível de sofisticação nas páginas de captura de informações dos usuários. Dessa maneira é importante que novos mecanismos de filtro automatizado de sites e mensagens fraudulentas sejam implantados, a fim de minimizar a quantidade de usuários que estão sujeitos a esses golpes. Apresentou-se também uma análise de ambientes de filtros de URL, a saber, o projeto *Google Safe browsing* e *Phishtank*, onde evidênciou-se a necessidade de uma base de dados voltada para a realidade brasileira de fraudes eletrônicas, haja visto que a grande maioria das fraudes não é encontrada nessas bases de dados internacionais (7.6% de detecção pelo *Safe browsing* e 2.82% de detecção pelo *Phishtank*).

Como trabalhos futuros, espera-se aprimorar a apresentação do Catálogo de Fraudes para os usuários, adicionando novos mecanismos de busca no site, gráficos, e formulários de validação de e-mail; implantar um catálogo de URLs maliciosas observadas nacionalmente, fornecendo uma API de consulta para viabilizar filtros automatizados nas instituições (trabalho em andamento), investigar e propor mecanismos de detecção automatizada de fraudes, para que o processo de triagem do Catálogo de Fraudes possa ser automatizado e assim a quantidade de fraudes analisadas seja maior.

Referências

[1] The Radicati Group, Inc. Email Statisctics Report, 2014-2018. http://www.radicati.com/?p=10644, acessado em 12/05/2015.

[2] Ollmann, Gunter. *The Phishing Guide: Understanding & Preventing Phishing Attacks.* IBM Int. Sec. Sys. Disponível em http://www-935.ibm.com/services/us/iss/pdf/phishing-guide-wp.pdf, acessado em 29/04/2015.

[3] VirusTotal. https://www.virustotal.com/, acessado em 29/04/2015.

[4] Phishtank. https://www.phishtank.com/, acessado em 29/04/2015.

[5] Google's Safe Browsing Diagnostic Tool. https://developers.google.com/safe-browsing/, acessado em 04/05/2015.
[6] Filtro SmartScreen. http://windows.microsoft.com/pt-br/internet-explorer/products/ie-9/features/smartscreen-filter, acessado em 02/05/2015.
[7] Basnet, Ram B., Andrew H. Sung, and Quingzhong Liu. *Rule-based phishing attack detection.* International Conference on Security and Management (SAM 2011), Las Vegas, NV. 2011.
[8] Fette, Ian, Norman Sadeh, and Anthony Tomasic. *Learning to detect phishing emails.* Proceedings of the 16th international conference on World Wide Web. ACM, 2007.
[9] McGrath, D. Kevin, and Minaxi Gupta. *Behind Phishing: An Examination of Phisher Modi Operandi.* LEET 8 (2008): 4.
[10] Garera, Sujata, et al. *A framework for detection and measurement of phishing attacks.* Proceedings of the 2007 ACM workshop on Recurring malcode. ACM, 2007.
[11] RSA Online Fraud Resource Center. http://www.emc.com/emc-plus/rsa-thought-leadership/online-fraud/index.htm#!resources, acessado em 29/04/2015.
[12] APWG Reports. http://www.antiphishing.org/resources/apwg-reports/, acessado em 29/04/2015.

DOI: 10.5769/C2015015 or http://dx.doi.org/10.5769/C2015015

Perícia Computacional em Artefatos Digitais de Interceptações Telefônicas

Wilson Leite da Silva Filho

***Resumo*—Este artigo apresenta o trabalho pericial realizado em três casos de interceptações telefônicas. São discutidos os aspectos operacionais e técnicos das interceptações telefônicas automatizadas pelos Sistemas Vigia e Guardião. Esses sistemas geram uma grande quantidade de artefatos digitais, que foram questionados quanto a sua integridade e integralidade. A defesa também trouxe à discussão o princípio da igualdade de armas, alegando que não possui os mesmos recursos de análise que a parte acusatória. Desses questionamentos, que envolvem uma grande quantidade de dados, surgiu a necessidade de desenvolvimento de um software pericial específico para atender a essas demandas. Tal software foi utilizado nas três perícias e está pronto para ser usado e/ou adaptado para outros casos semelhantes que demandem os mesmos tipos de análise nos artefatos digitais produzidos pelos sistemas de interceptação telefônica.**

Palavras-Chave*—*Interceptações telefônicas, Sistema Guardião, Sistema Vigia, Software Pericial.

***Abstract*—This article presents the forensics work done in three cases involving phone calls interceptions. It´s discussed the operational and technical aspects of the interception, done by the Vigia and Guardiao systems. These systems produce a huge amount of digital artifacts that has been questioned regarding their integrity and integrality. The defense also discuss the principle of weapons parity, claiming that they don´t have the same analysis resources that the accusatory part. From these questionings, involving a great amount of data, comes the necessity to develop a forensics software specially designed for this task. This software was used in the three cases and is available to be used or adapted for other cases where there is similar digital artifacts generated by phone call interception systems.**

Keywords— Phone Calls Interceptions, Guardião System, Vigia System, Forensic Expert Software.

I. INTRODUÇÃO

A interceptação de comunicação telefônica é uma operação que se constitui na interferência de um terceiro na captação de uma comunicação entre duas ou mais pessoas, sem o conhecimento dos interlocutores. É a modalidade específica de interceptação legal realizada por órgãos policiais na investigação criminal, com autorização judicial e de acordo com os dispositivos legais das leis 9.296/96 e 10.217/01, devendo ser conduzida por autoridade policial a quem cabe a responsabilidade das operações técnicas, a integridade da prova colhida e a análise de seu conteúdo [1].

Os processos de interceptações são realizados por procedimentos amplamente apoiados por sistemas de informação automatizados, que fornecem aos órgãos de persecução criminal ferramentas para a gravação, análise e geração de relatórios com os resultados das interceptações. Os áudios e os relatórios resultantes das operações são adicionados aos processos criminais e seguem para análise das partes interessadas.

Ao serem anexados aos autos, a defesa tem acesso ao conjunto de provas geradas nas interceptações telefônicas. Esse conjunto é formado por artefatos digitais, tais como arquivos de áudio e registros que relacionam alvos interceptados, datas e horas das chamadas. Foi dos questionamentos sobre a integridade das provas geradas pelos sistemas de interceptação que surgiu a necessidade da perícia computacional abordada nesse artigo.

II. OBJETIVO

O objetivo principal do trabalho é apresentar os procedimentos que foram realizados nas análises de três casos de perícia em artefatos digitais de interceptações telefônicas.

Serão discutidas as dificuldades encontradas devido à necessidade de análise da grande quantidade de dados oriundos de fontes diversas. Será apresentado um software desenvolvido pelo autor, específico para esse tipo de perícia.

Por fim, serão apresentados os resultados dos processamentos dos dados realizados pelo software desenvolvido.

III. SISTEMAS DE INTERCEPTAÇÃO TELEFÔNICA

Dois sistemas que são comumente utilizados nos Estados e na União para a automatização e viabilização das interceptações telefônicas são os Sistemas Vigia, da empresa Suntech, e o sistema Guardião, da empresa Digitro.

Segundo a Suntech, o Sistema Vigia é uma solução completa para gerenciar todo o processo de interceptação legal e retenção de dados para qualquer serviço ou subsistema de comunicação de qualquer tecnologia ou fornecedor [2]. No contexto das interceptações telefônicas analisadas nas perícias, o Sistema Vigia foi utilizado pelas operadoras de telefonia para programar as centrais telefônicas e realizar os desvios dos telefones alvos das interceptações.

Segundo a Digitro, o Sistema Guardião realiza monitoração de voz e dados e oferece recursos avançados de análise de áudio e identificação de locutores. É uma solução feita

especialmente para as operações de investigação legal [3]. No contexto das interceptações telefônicas analisadas, o Guardião foi usado para receber as ligações telefônicas desviadas pelas operadoras. Era, portanto, o sistema instalado no Órgão Policial com as funções de realizar a coleta de áudio e gerar os relatórios resultantes das interceptações.

Cabe salientar que neste trabalho foram analisadas apenas as características dos sistemas relacionadas às perícias em questão e a forma de utilização que os sistemas foram submetidos para a geração dos dados. Não é objetivo do trabalho qualquer forma de auditoria ou análise crítica sobre os recursos e capacidades que cada sistema oferece.

A. Processo de Interceptação Telefônica

O processo de interceptação telefônica realizado pelo Órgão Policial por meio do Sistema Guardião segue várias etapas que envolvem o Poder Judiciário, o Ministério Público, os membros das forças de segurança pública e as operadoras de telefonia móvel e fixa.

A primeira etapa do processo é a solicitação ao Poder Judiciário de autorização para realizar interceptação telefônica. A Autoridade Policial prepara um documento com tal solicitação, elencando quais os números de telefones que precisam ser interceptados e as respectivas operadoras às quais eles pertencem.

Após receber a autorização judicial, é feita a programação do Sistema Guardião, por um técnico operador do sistema. Nesta configuração, é estabelecido um canal de desvio (canais DDR – PABX) para cada número de telefone a ser monitorado. Nesse momento, o Sistema Guardião está pronto para registrar passivamente os áudios desviados pela operadora por meio do Sistema Vigia.

Na etapa seguinte, envia-se uma solicitação juntamente com o mandado judicial para a operadora de telefonia. Nela, são informados quais telefones deverão ser interceptados e em quais canais de desvio os respectivos áudios deverão ser enviados. Geralmente, é dado um prazo judicial de 24 horas para que a operadora configure os desvios de canal.

Somente a partir daí o Guardião passa a receber as interceptações. Durante as gravações, as conversas podem ser acompanhadas em tempo real pelas equipes de investigação.

Expirado o prazo da interceptação solicitada, a operadora encerra o desvio dos números alvos para as linhas monitoradas pelo Sistema Guardião.

A figura 1 apresenta os agentes envolvidos nos procedimentos de interceptação descritos anteriormente.

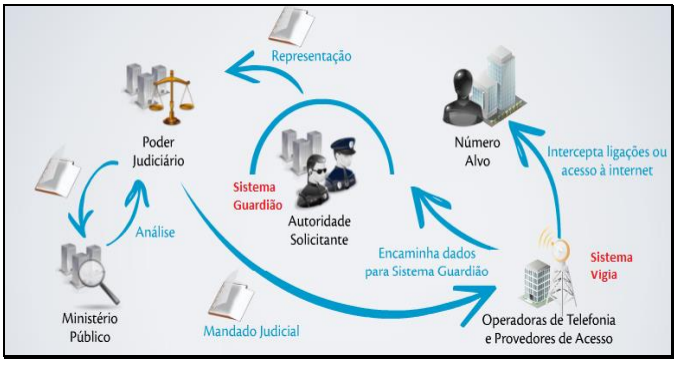


Figura 1 – Agentes envolvidos nas interceptações telefônicas (compilação do passo a passo disponível no site do Guardião [3])

IV. DOS QUESTIONAMENTOS DA DEFESA

As principais indagações feitas pela defesa, por meio de seus assistentes técnicos, e aceitas pelos Juízes, correspondem aos seguintes quesitos: a) se o conjunto dos artefatos digitais anexados aos autos correspondem à completude das interceptações telefônicas; b) se os registros e áudios estão íntegros, ou seja, não foram apagados ou editados; c) se há algum registro de interceptação que esteja fora da data de autorização judicial; d) que seja feito um cotejamento entre os registros do Sistema Vigia, usado pelas operadoras de telefonia, e os registros do Sistema Guardião, usado pelo Órgão Policial.

A defesa também questiona, baseada no princípio da paridade de armas, o direito de ter as mesmas condições de análise que a parte acusatória. Tal princípio, segundo Ferrajoli, diz que para que a disputa se desenvolva lealmente e com paridade de armas, é necessária a perfeita igualdade entre as partes: em primeiro lugar, que a defesa seja dotada das mesmas capacidades e dos mesmos poderes da acusação; em segundo lugar, que o seu papel contraditor seja admitido em todo estado e grau do procedimento e em relação a cada ato probatório singular, das averiguações judiciárias e das perícias ao interrogatório do imputado, dos reconhecimentos aos testemunhos e às acareações [4].

A defesa argumenta que a parte acusatória possui acesso total ao Sistema Guardião, que possibilita várias ferramentas para a análise dos dados. À defesa restariam apenas os registros digitais das interceptações, gravados em mídia ótica e acessados por um leitor específico. Não haveria ferramentas para fazer um cruzamento dos dados. Dessa necessidade, os defensores formularam mais um quesito: e) que os dados sejam importados para algum Sistema de Banco de Dados para que possam ser analisados, ou que, não sendo possível a importação, que se aponte no laudo as dificuldades do cruzamento da grande quantidade de dados provenientes das interceptações telefônicas.

Por fim, para dirimir eventuais dúvidas em relação à prova, a Autoridade Judicial determinou que os peritos extraíssem na íntegra e com verificação de integridade todos os registros referentes às interceptações telefônicas.

V. OS PROCEDIMENTOS PERICIAIS

Durante o processo de interceptação, os dados desviados pela operadora para o Sistema Guardião foram armazenados em disco rígido. Ao fim da operação policial, esses dados foram gravados em mídias óticas e as mídias anexadas aos autos correspondentes. O volume de dados gerado por uma operação de interceptação telefônica, que geralmente envolve vários alvos, por períodos de meses ou até anos é muito grande. Um dos casos envolveu aproximadamente 7.000 horas de áudio, volume que precisou ser gravado em 18 DVDs, ou seja, cerca de 80GB.

Devido à capacidade de armazenamento limitada do Sistema Guardião que foi periciado e ao número de operações de interceptação telefônicas simultâneas em andamento, não era possível manter por muito tempo os dados de uma operação de interceptação, ou faltaria espaço para operações subsequentes. Em razão disso, os discos rígidos do Sistema foram usados de forma rotativa, ou seja, os áudios de interceptações de operações já finalizadas eram apagados para liberar espaço para novas interceptações.

Cumpre frisar que as informações gravadas pelo Sistema Guardião correspondem a registros de dados em que constam, por exemplo, o número de telefone alvo e o período de interceptação, e arquivos de áudio associados a cada um desses registros. No processo de liberação de espaço de forma rotativa para novas operações, apenas os arquivos de áudio são sobrescritos. Já os registros de dados permanecem no Sistema indefinidamente. No início das análises periciais, o processo de liberação rotativa de espaço já tinha atingido as operações foco da perícia. Dentro do Sistema Guardião existiam, portanto, apenas os registros de dados da operação, sem os áudios correspondentes.

Os arquivos de áudio, além de anexados aos autos em mídias óticas, haviam sido também guardados em disco rígido externo ao Sistema Guardião. Esses arquivos são armazenados com criptografia, e somente podem ser lidos pelo Guardião Reader. Esse software, além de reproduzir o áudio das interceptações, é capaz de verificar sua integridade por meio de algoritmo proprietário, que atesta unicamente a ocorrência ou não de adulteração nos arquivos, e não que eles correspondem à totalidade das gravações coletadas durante o período de interceptação.

Desse cenário, constata-se a existência de dois conjuntos de informações, armazenadas em locais distintos. Para que se cumpra o requisitado pela Autoridade Judicial é necessário que se integre estes dois conjuntos de dados, formando um todo válido. O primeiro conjunto de informações são os arquivos de áudio, armazenados no disco rígido externo. Este conjunto de áudio é tecnicamente íntegro, ou seja, os áudios que ali estão não foram adulterados, conforme atestado pelo algoritmo de verificação de integridade. O segundo conjunto de informações são os registros de dados (sem os áudios) relativos a toda operação, que está armazenado dentro do Sistema Guardião. Para determinar se os áudios do disco rígido externo e dos DVDs correspondem aos áudios da operação de interceptação, deve-se relacionar esses arquivos de áudio com os registros de dados presentes dentro do Sistema Guardião.

Tais registros de dados armazenados no Guardião, correspondentes às três interceptações telefônicas periciadas, compreendem aproximadamente 600.000 linhas de informação organizadas em aproximadamente 12.000 páginas de relatório. Como pode-se notar, o processo de comparação de dados é custoso e exige um esforço que só pode ser realizado em tempo razoável com o auxilio de recursos computacionais. Desta forma, desenvolveu-se um software específico para esta tarefa. Em linhas gerais, o software deve ler o relatório emitido pelo Sistema Guardião na presença do perito, e interpretar cada linha deste relatório, extraindo informações que serão usadas para procurar o arquivo correspondente nas mídias anexas aos autos e no disco rígido de *backup*. O sucesso desta comparação é similar ao processo de extração dos áudios de dentro do Sistema Guardião, caso eles ainda estivessem lá. Os módulos e o fluxo dos dados envolvidos nessa operação estão demonstrados nos diagramas das figuras 2, 3 e 4.

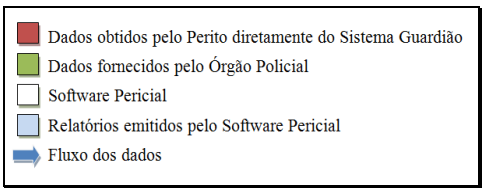


Figura 2 – Legenda dos fluxogramas

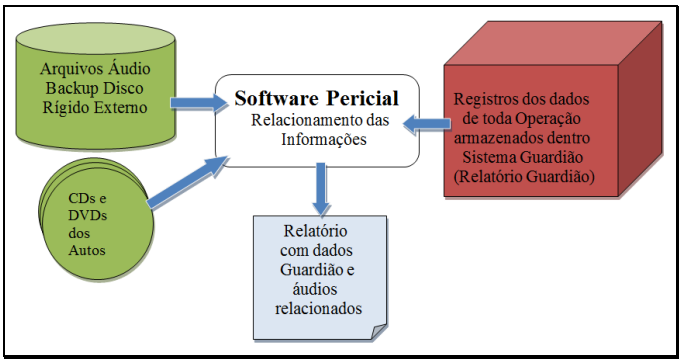


Figura 3 – Fluxograma do software pericial

Com esse módulo do software foi possível responder aos quesitos: a) se o conjunto dos artefatos digitais anexados aos autos correspondem a completude das interceptações telefônicas; b) se os registros e áudios estão íntegros, ou seja, não foram apagados ou editados.

O próximo quesito a ser respondido foi: c) se há algum registro de interceptação que esteja fora da data de autorização judicial. Para isso foi desenvolvido um módulo que verifica os prazos judiciais autorizados e compara com cada registro, dizendo se a interceptação correspondente está ou não autorizada.

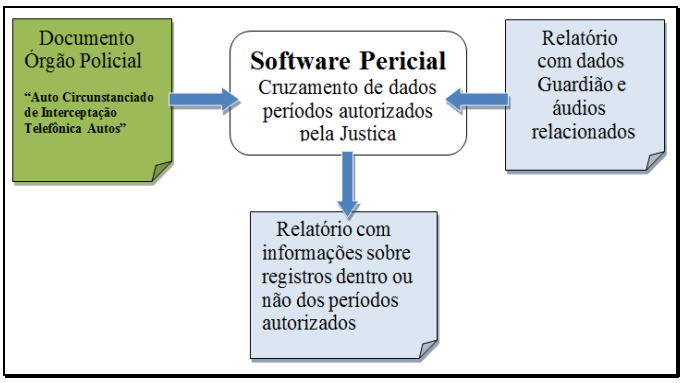


Figura 4 – Fluxograma do software pericial

O próximo quesito a ser respondido, refere-se ao cotejamento dos dados do Sistema Vigia com os dados do Sistema Guardião. Há ressalvas importantes a serem feitas sobre essa tarefa. Os dois sistemas são independentes, foram projetados de formas distintas. Nem todos os eventos de um sistema estão diretamente relacionados com os do outro. Além disso, há diferença de horário em que as chamadas começam a ser registradas e possíveis diferenças de ajuste do relógio dos módulos relacionados ao Sistema Vigia. Por exemplo, o Sistema Vigia registra o início do horário de uma chamada no momento em que a chamada é atendida. Já o Sistema Guardião registra a chamada logo ao término da discagem do número. Outro exemplo que geraria discrepâncias é que os relógios das

centrais telefônicas das operadoras podem não estar precisamente sincronizados com o horário oficial de Brasília. Os relógios podem estar atrasados ou adiantados em relação a ele, ou ainda, em desacordo quanto ao horário de verão. Essa falta de sincronismo causa diferenças nos registros das chamadas entre os Sistemas Vigia e Guardião.

Dessa forma, esse módulo do software deve realizar um trabalho de aproximação entre as datas e horários para tentar relacionar os registros compatíveis entre os dois sistemas e apresentar esses resultados. Em caso de dúvidas não resolvidas pelo algoritmo do software, uma análise pontual deverá ser feita em cada caso de interesse. Em uma das perícias realizadas, do total de aproximadamente 270.000 registros, 180.000 foram relacionados. Nos 90.000 registros restantes, o relacionamento automático não logrou êxito.

O último quesito a ser respondido baseia-se no princípio da paridade de armas, no qual a defesa pede: e) que os dados sejam importados para algum Sistema de Banco de Dados para que possam ser analisados.

Durante todo o processamento, o software pericial fez uso de Banco de Dados Relacional e arquivos de relatório. Foram criadas tabelas que modelam os registros que estão sendo questionados. Essas tabelas podem ser exportadas para que a defesa possa realizar as consultas que achar necessárias.

O modelo relacional das tabelas que modelam os registros envolvidos nas interceptações telefônicas está ilustrado na figura 5.

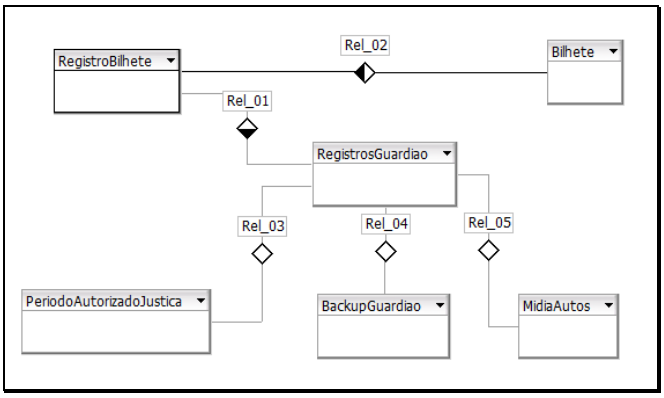


Figura 5 – Modelo relacional dos dados das interceptações

VI. CARACTERÍSTICAS TÉCNICAS DO SOFTWARE PERICIAL

O software pericial citado no artigo foi desenvolvido em Java. Para acesso ao sistema de arquivo e para a listagem e procura de arquivos em discos e mídias óticas optou-se por fazer chamadas aos comandos de busca em disco do sistema operacional Linux. Tal decisão foi baseada no contato prévio que o autor tinha com esses comandos (*locate* e *updatedb)* e em sua confiabilidade. Para realizar as chamadas foi usado o método *exec* do ambiente Java.

Para as análises dos arquivos texto e relatórios fornecidos pelos Sistemas Guardião e Vigia foram desenvolvidos interpretadores dos relatórios. Para o relatório principal do Sistema Guardião foi especificado uma gramática formal, usando-se EBNF[5], e criado um interpretador dessa gramática. EBNF é um código que expressa a gramática de uma língua formal. Uma gramática definida em EBNF consiste em símbolos terminais e regras de produção não-terminais. Restrições relativas, como símbolos terminais, podem ser combinados em uma sequência válida. Exemplos de símbolos terminais incluem caracteres alfanuméricos, sinais de pontuação e caracteres de espaço em branco. EBNF é utilizado na definição de linguagens formais e pode estar envolvida no processo de construção de interpretadores e compiladores [5].

Para as pesquisas e manejo dos dados foi usado o banco de dados relacional. Incialmente, as primeiras versões do software utilizavam arquivos texto e pesquisas diretas em objetos instanciados na memória por meio da linguagem Java. Essa abordagem, entretanto, se mostrou lenta no processamento de uma grande quantidade de dados. Na versão atual, as estruturas de dados alojadas anteriormente em memória e salvas em arquivos texto foram modeladas para tabelas em banco de dados. Foi alcançado um ganho considerável de desempenho com essa abordagem, além de possibilitar recursos mais sofisticados de pesquisa sobre os dados processados. O banco de dados escolhido foi o PostgresSQL [6], acessado via JDBC [7].

A interface do software é toda em linha de comando. É possível usar alguns parâmetros de entrada para escolher determinadas funções. Algumas configurações também podem ser feitas em arquivos texto.

Porém, os recursos foram sendo acrescentados conforme demanda específica. Para se utilizar o software de forma genérica, ainda é necessário um esforço para tornar a interface mais amigável e principalmente para melhorar a documentação, que atualmente se restringe a comentários nos códigos fonte.

VII. CONCLUSÕES

O processo de interceptação de ligações telefônicas é amplamente utilizado pelos Órgãos Policiais em suas investigações. A integridade e confiabilidade das provas geradas pelos sistemas de interceptações telefônicas foram questionadas em três casos distintos. Além desses questionamentos, a defesa argumenta que não possui paridade de armas em relação aos órgãos acusatórios. Este questionamento levou à necessidade de perícia em um ambiente com grande quantidade de dados, que só foi viabilizada com o desenvolvimento de um software específico para este fim. Com o auxílio desse software foi possível responder aos quesitos formulados pelos Juízes e pela defesa, bem como fornecer à defesa dados estruturados para que ela pudesse fazer suas próprias análises.

O software pericial foi utilizado em três perícias distintas e está pronto para ser usado e/ou adaptado para outros casos semelhantes que demandem os mesmos tipos de análise nos artefatos digitais produzidos pelos sistemas de interceptação telefônica.

O cotejamento entre os registros dos dois sistemas envolvidos nas interceptações apresentou dificuldades técnicas devido a diferenças entre seus dados. Foi utilizado um algoritmo que tenta relacioná-los por aproximação. Essa tarefa logrou um sucesso parcial. Como trabalho futuro, haveria espaço para se estudar mais detalhadamente essa relação e com isso propor um aprimoramento desse algoritmo.

REFERÊNCIAS

[1] Ferro Junior, Celso Moreira; Oliveira Filho, Edemundo Dias de; Preto, Hugo César Fraga; "Segurança Pública Inteligente (Sistematização da Doutrina e das Técnicas da Atividade); Editora Kelps: Goiânia, 2008.

[2] Vigia. http://www.suntech.com.br/pt/solucoes/lawful-interception/vigia/, acessado em 18 de abril de 2015.

[3] Guardião. http://www.digitro.com/pt/index.php/component/content/article/89Itemid=1, acessado em 18 de abril de 2015.

[4] Ferrajoli, Luigi. Direito e razão: teoria do garantismo penal. 2. ed. São Paulo: Revista dos Tribunais, 2006, p. 565

[5] Watt, David A.; Brown, Deryck F. Programming Language Processors in Java – Compilers and Interpreters, Prentice Hall, Great Britain: 2000

[6] PostgresSQL. http://www.postgresql.org/, acessado em 12 de junho de 2015.

[7] Silberschatz, Abraham, Korth , Henry F. e Sudarshan, S. - Sistema de Banco de Dados – Tradução da 5ª edição; Editora Elsevier: Rio de Janeiro, 2006.

Wilson Leite da Silva Filho, Perito Criminal - Instituto Geral de Perícias de Santa Catarina; Mestre em Engenharia de Computação - IPT/USP; Especialista em Segurança da Informação – ITA; Graduando em Psicologia – UFSC.