Instant Messaging Forensics

Gabriel Menezes Nunes, Laboratório ACME! Computer Security Research – UNESP – Universidade Estadual Paulista "Júlio de Mesquita Filho"

***Resumo*— Milhões de pessoas em todo mundo usam aplicações de mensagens instantâneas todos os dias para se comunicarem com amigos, colegas de trabalho, fornecedores, clientes ou até mesmo para enviar informações sensíveis com objetivos militares.**

Com o aumento na demanda de uso desse tipo de tecnologia, surgiu a necessidade de analisar que tipo de informações estão sendo trafegadas nessas conexões.

Esse artigo apresenta um novo modelo de análise forense, através do qual o investigador será capaz de localizar evidências com facilidade em protocolos de mensagens instantâneas. Além disso, esse artigo apresenta a ferramenta MSN Shadow, a qual é capaz de capturar e examinar diversas informações que trafegam no protocolo MSNP da Microsoft, além de realizar técnicas de investigação ativa.

Palavras Chaves – segurança computacional, perícia forense, msn, instant messaging, tcp hijack, ip spoofing, sniffing.

I. INTRODUÇÃO

Atualmente o uso de mensagens instantâneas [1] se tornou algo comum tanto para o entretenimento de usuários finais quanto para a diminuição de custos de comunicação dentro de uma empresa, ou até mesmo comunicações militares [2].

O protocolo mais usado atualmente é o Microsoft Notification Protocol [3], conhecido como MSN, o qual vem implementado num software cliente instalado por padrão em diversas versões do sistema operacional Windows. Esse protocolo não permite apenas conversas por texto, mas também vídeo conferência e transferência de arquivos.

Muitos desenvolvedores fora da Microsoft implementam esse protocolo em outros sistemas operacionais como Linux, Mac OS e clientes para celulares, PDA e muitos outros dispositivos que tenham a capacidade de se conectarem à rede mundial de computadores.

Essa ubiqüidade desse tipo de protocolo deve aumentar a preocupação de todos que gerenciam tecnologia da informação, principalmente da equipe de segurança.

Como qualquer outra tecnologia, esta também pode ser usada tanto para o bem quanto para o mal. Comunicações facilitadas com fornecedores, clientes e outros que trabalham num sistema cooperativo [4] são exemplos de bom uso desse sistema síncrono de comunicação. Por outro lado, casos como espionagem industrial e uso abusivo fora da política de segurança da empresa podem diminuir a produtividade e causar grandes perdas (financeiras ou de propriedade intelectual) .

O único meio de controlar o uso desses protocolos é através de ferramentas conhecidas como *NFAT* (*Network Forensics Analysis Tools*) [25], ferramentas de análise forense de rede, que dão uma visão em tempo-real do que está acontecendo na sua rede e fornecem mecanismos que melhoram o exame [5] e tornam o trabalho de análise [5] mais fácil para o investigador.

Este artigo apresentará um novo modelo técnico de investigação de protocolos de mensagens instantâneas, o qual tem o objetivo de guiar o analista na busca por evidências nesse meio de comunicação. Além disso, será apresentada a ferramenta de análise forense MSN Shadow.

Objetivos

O objetivo desse projeto é a criação de um método científico de investigação, cuja função é servir como base para a análise de protocolos de mensagens instantâneas.

O modelo que será apresentado e a ferramenta desenvolvida demonstrarão, tanto na teoria, quanto na prática, a eficácia da metodologia descrita.

Dependendo da quantidade de tráfego de uma rede ou o nível de uso de clientes MSN, tanto para uso produtivo ou uso abusivo, haverá um grande problema na análise, pois milhares de pacotes podem trafegar pela rede num determinado período de tempo. Logo, uma ferramenta que torne o exame desse tráfego mais rápido e melhore a posterior análise que será feita pelo investigador é necessária.

A ferramenta MSN Shadow é capaz de capturar o tráfego da rede correspondente apenas ao protocolo MSNP em tempo-real, facilitando o trabalho de reconstrução de fluxo e busca por evidências [6], além de realizar técnicas de investigação ativa. Esse software pode ser utilizado para verificar se a política de segurança de uma determinada empresa está sendo seguida, além de descobrir fontes de vazamento de informações por espionagem industrial e uso abusivo da tecnologia como, por exemplo, excesso de conversas desnecessárias entre colegas de trabalho ou casos de pedofilia. Também é possível seqüestrar uma conexão a fim de investigar de forma mais ativa uma determinada situação, como, por exemplo, pedofilia, no qual um policial intercepta uma conexão para uma suposta criança, e, a partir daí, tenta conseguir mais informações ou marcar um encontro para realizar a prisão do criminoso.

II. *INSTANT MESSAGING*

Sistemas de presença e mensagens instantâneas permitem os usuários se conectarem uns aos outros e trocarem informações de estado e mensagens. [1]

Diferentemente de sistemas de correio eletrônico [7], o modelo de mensagens instantâneas foi criado para ser síncrono, ou seja, em tempo-real. Esse tipo de modelo, o qual se tornou extremamente popular, está sendo implementado não apenas para o divertimento dos usuários finais, mas também para a diminuição de custos de comunicação entre filiais e funcionários de uma mesma empresa e, também, comunicação rápida com clientes e fornecedores.

O modelo descrito em [1] define dois serviços:

- Serviço de presença
- Serviço de mensagens instantâneas

Serviço de Presença

A principal função do serviço de presença é distribuir mensagens de estado entre os clientes conectados.

O serviço de presença pode ter dois tipos de clientes:

- Presentes: entidades que fornecem informações de presença para serem armazenadas e distribuídas
- Observadores: entidades que apenas recebem as informações de presença.

Na maioria das implementações, todo cliente é um presente e um observador, pois além de enviar informações de estado, também as recebe de seus contatos. O servidor de presença é, quase sempre, aquele fornecido pelo fabricante, ao qual o cliente se conecta e se autentica.

Existem diversos tipos de mensagens de estado como aviso de conexão, desconexão, troca de mensagens do *display*, adição e deleção de contatos, dentre outros.

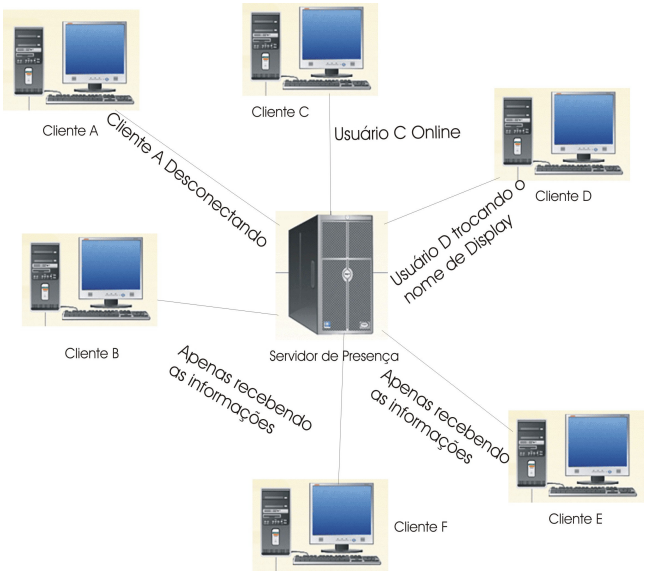

Figura 1 Rede de comunicação entre clientes e o servidor de presença.

Serviço de Mensagens Instantâneas

Serviço que tem como função o envio de mensagens entre clientes. O funcionamento desse serviço é muito simples:

1. O emissor envia uma mensagem para o serviço de mensagens instantâneas.
2. O serviço mensagens instantâneas tenta enviar a mensagem para a caixa de entrada instantânea correspondente (*instant inbox*).
3. Caso a caixa de entrada correspondente seja localizada, o receptor terá acesso à mensagem.
4. Caso a caixa de mensagem correspondente não seja encontrada, deve ser retornada uma mensagem de erro para o emissor.

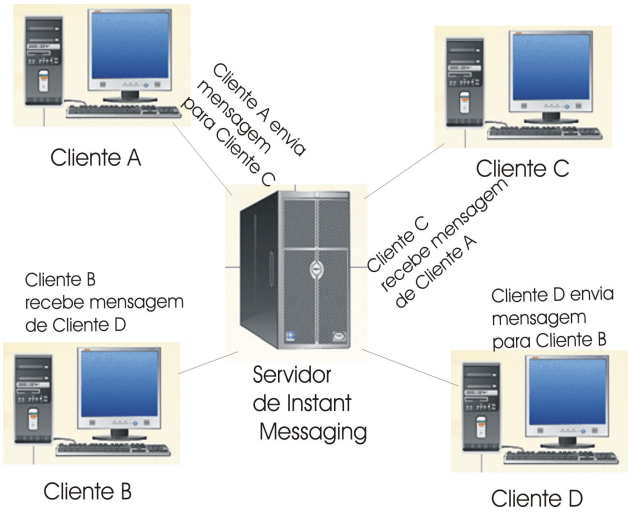

Figura 2 Comunicação entre clientes intermediada por um servidor de *Instant Messaging*.

Protocolos de Mensagens Instantâneas

Há diversos protocolos de mensagens instantâneas no mercado, os quais os mais usados são: *MSN, AIM, ICQ, Yahoo e Jabber/XMPP*; sendo todos esses proprietários, exceto o último citado.

Apesar de existirem diversos softwares clientes que implementam os mais diferentes protocolos de mensagens instantâneas, as funcionalidades oferecidas são praticamente iguais:

- Lista de contatos
- Troca de mensagens instantâneas
- Envio de mensagens offline
- Envio de arquivos
- *Emoticons*
- Conferência por voz
- Conferência por vídeo
- Envio de *e-mails*
- Envio de mensagens *sms*
- *Chat* multi-usuário
- Compartilhamento de arquivos

Perigos no Uso de Instant Messaging

Como qualquer outra tecnologia, esta também pode ser abusada e adiciona uma camada a mais de insegurança na rede. Problemas mais comuns no uso de *instant messaging* são:

1. Códigos Maliciosos: Como quase todos os clientes de mensagens instantâneas têm a capacidade de envio de arquivos, o uso dessa tecnologia facilita a entrada de vírus, *worms, trojans* e outros tipos de códigos maliciosos no ambiente corporativo. As prevenções para esse tipo de ataque são o treinamento do usuário e a atualização de softwares de segurança como anti-*spywares* e anti-vírus.
2. Falhas de software: Como qualquer outro tipo de software, os programas mensageiros também podem estar vulneráveis aos ataques clássicos de estouro de pilha*, format strings,* estouro de *heap*, dentre outros. Com a adição de processadores *html* em diversos clientes, há também a possibilidade falhas *web*. [8]
   A prevenção é uma boa política de *patches* e atualizações.
3. Vazamento de informações: Adicionando uma camada a mais de comunicação, há também a soma de uma camada a mais de vazamento, caso a implementação não seja bem planejada. Grande parte dos clientes mensageiros utilizados atualmente usa servidores internos das empresas que os fornece como *Microsoft, Yahoo e America Online*. Deve haver um planejamento da equipe interna de TI para verificar se é mesmo possível e seguro permitir que dados da empresa trafeguem por servidores externos. Além disso, é necessário o uso de clientes que permitem sessões criptografadas, o que não é o caso de protocolos como o da *Microsoft*.
4. *Phishing*: Ataques que se tornaram freqüentes atualmente. Com o aumento no uso de mensagens instantâneas, há possibilidade dos atacantes se voltarem a esse meio para ludibriarem suas vítimas.
5. Negação de serviço: O uso irrestrito dessa tecnologia e a liberação de seu tráfego pode tornar a rede em questão vulnerável à ataques de negativa de serviço dirigida a esses protocolos.

Há diversos problemas de segurança nos clientes atuais de mensagens síncronas [9]. No entanto, analisando apenas o nível de rede, a melhora de segurança com a implementação de criptografia é considerável, já que impediria ataques de captura e manipulação de informações. Apesar disso, apenas o protocolo *Jabber/XMPP* tem clientes que suportam criptografia.

Protocolo do MSN

O *Microsoft Notification Protocol*, usado na rede do *Microsoft Live Messenger*, foi criado em 1999 [10] pela *Microsoft*. Ao longo dos anos, o protocolo teve diversas evoluções e melhorias, e hoje, está na sua versão 15, conhecido como *MSNP15*.

O funcionamento do *Microsoft Notification Protocol* exige dois tipos de servidores, os quais residem na rede do fornecedor: Servidor de Notificação e o *SwitchBoard.*

O servidor de notificação é o principal, ao qual o cliente se conecta, e é neste que é feita a autenticação. Após esta fase, é possível trocar diversas mensagens com o servidor para realizar várias tarefas como adicionar ou excluir contatos, mudar o nome do *display,* requisitar conversações, dentre outras. Este é o servidor que atuará como entidade de presença, logo, se houver desconexão com este host, o usuário não mais estará ativo para a sua lista de contatos. [10]

O *switchboard* é o servidor que irá fazer a ponte na conversação entre dois clientes, ou seja, ele irá atuar como *proxy*. No *MSNP* não é possível conversar com o cliente via conexões ponto-a-ponto, é necessário sempre um servidor da *Microsoft*, o qual intermeará essa conexão. Este é o servidor que atuará como entidade de mensagem instantânea. Além de redirecionar as mensagens instantâneas trocadas entre os clientes, esse servidor também repassa mensagens de transferência de arquivo, voz e vídeo.

O *Microsoft Notification Protocol* usa conexões fora da banda (*out-of-band*), para tratar mensagens trocadas entre clientes, ou seja, dados de controle enviados para o servidor de notificação são trafegados em conexões diferentes das mensagens trocadas com servidores s*witchboard*. Logo, caso um atacante corrompa uma conexão, com ataques de forjamento*[11]* ou seqüestro de sessão *[11],* por exemplo, a conexão com outros clientes ou servidor de notificação não será afetada, dificultando uma possível detecção do ataque. Outros tipos de protocolos, como o *Jabber/XMPP,* usam a mesma conexão tanto para envio de informações de presença quanto envio de mensagens de conversação, então, caso essa conexão seja atacada, todas as conexões do usuário serão afetadas. Outra questão diante da arquitetura do *MSNP*, é que as conexões com servidor de notificação e com os servidores *switchboards* são independentes, ou seja, caso a conexão com o servidor de notificação seja terminada, as conexões com *switchboard* não serão terminadas até que o software cliente as finalize. [10] Também foi verificado que é possível um cliente ter duas ou mais conexões advindas de diferentes *switchboards*, mas levando à mesma janela de conversação, causando um problema no qual o cliente está recebendo mensagens de diferentes remetentes, mas com o mesmo nome de usuário, ou seja, mesmo que o usuário real esteja mantendo contato com o cliente atacado, é possível que o invasor forje mensagens, as quais não serão vistas como falsas para o cliente.

III. *MODELO*

Modelo Abstrato de Investigação Digital de Protocolos de Mensagens Instantâneas - MAIDPMI (The Abstract Instant Messaging Protocols Digital Investigation Model)

O modelo proposto neste trabalho, inspirado no modelo teórico e genérico criado pelo Departamento de Justiça dos Estados Unidos [12] e no Modelo Abstrato de Forense Digital [14], é uma nova abordagem tanto para o estudo e perícia de

protocolos de mensagens instantâneas quanto para criações de tais ferramentas.

Esse novo modelo técnico, chamado Modelo Abstrato de Investigação Digital de Protocolos de Mensagens Instantâneas, ou MAIDPMI, tem 7 componentes, os quais podem ser vistos na figura 3:

- Identificação do protocolo.
- Coleta de informações.
- Exame de informações cliente-servidor.
- Exame de informações cliente-cliente.
- Análise de informações cliente-servidor.
- Análise de informações cliente-cliente.
- Apresentação final.

O novo modelo apresentado tem o objetivo de suprir a falta de um processo mais técnico que é necessário para determinados tipos de protocolos. Por exemplo, protocolos de mensagens instantâneas são diferentes do protocolo *FTP [15]*, o qual é diferente de protocolos de *E-Mail* como *SMTP[16] e POP3 [17]*.

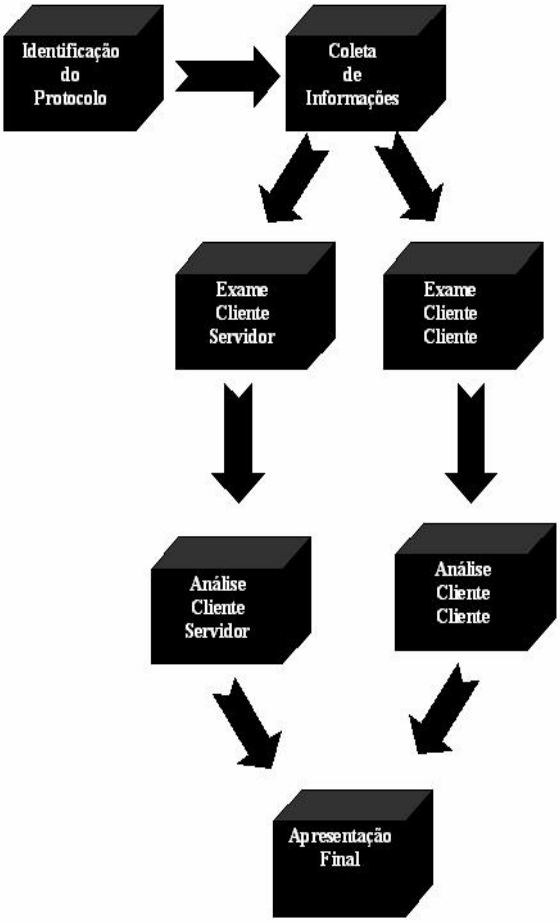

Figura 3 Fases do Modelo Abstrato de Investigação Digital de Protocolos de Mensagens Instantâneas.

A diferença em diversos tipos de protocolo não está apenas na arquitetura como também na porta *TCP [18]* utilizada, no cabeçalho e quais são as informações relevantes. No protocolo de transferência de arquivos o importante é o próprio arquivo transferido, enquanto que nos protocolos de correio eletrônico, as informações mais importantes são o remetente, destinatário e a mensagem enviada (talvez os servidores nos quais a mensagem passou também sejam importantes). Portanto, são necessárias diferentes abordagens de investigação para cada uma dessas tecnologias.

O MAIDPMI é um processo que pretende abranger todas as peculiaridades de protocolos de mensagens instantâneas, independentemente do fornecedor utilizado. Através desse modelo, o investigador será capaz de identificar qual o protocolo deverá ser periciado e como será feita essa perícia de modo que nada seja perdido e apenas as informações importantes sejam analisadas a fim de otimizar e facilitar seu trabalho. Além disso, o analista terá uma base para a criação de ferramentas como o *MSN Shadow*.

Identificação do Protocolo

Na primeira fase do modelo proposto o analista necessita identificar qual o protocolo será periciado. Algumas técnicas para essa identificação são:

- Verificação dos *softwares* clientes oficiais utilizados
- Captura de pacotes
- identificação do usuário

Verificação dos softwares clientes utilizadas

Uma técnica que pode ser usada para identificação do protocolo é a verificação do programa cliente oficial. É necessário que seja o oficial, pois muitos clientes como Kopete, Pidgin e outros têm suporte a diversos protocolos, logo apenas sua identificação não será suficiente para determinar qual protocolo será analisado.

Identificando o programa cliente oficial utilizado, a determinação do protocolo será trivial.

Alguns mais conhecidos:

Cliente Oficial	Protocolo
MSN Live Messenger	MSN Protocol
ICQ[ICQ08] e AIM	OSCAR
Yahoo! Messenger	Yahoo! Messenger Protocol
Skype	Skype Protocol
Psi*	XMPP/Jabber
mIRC	IRC Protocol
Gadu-Gadu	Gadu-Gadu

Tabela 1 Relação entre *software* clientes e protocolos de *instant messaging*.

Há uma observação no *Psi* pois o *XMPP/Jabber* é um protocolo aberto, logo não existe cliente "oficial", mas o *Psi* é um dos primeiros a adotar esse protocolo e é considerado pela comunidade como o principal cliente *Jabber*

Caso o protocolo analisado pelo perito seja novo ou proprietário, serão necessários outros tipos de pesquisa como o uso de mecanismos de busca como o *Google* ou utilização de outras técnicas para identificação.

Captura de Pacotes

Através da captura, o analista poderá reconhecer padrões que determinam o protocolo alvo: formato de cabeçalhos, conteúdo dos pacotes e portas utilizadas.

Formato de cabeçalhos e conteúdo dos pacotes

A verificação do cabeçalho e conteúdo dos pacotes possibilita o analista perceber padrões que facilitam a identificação do protocolo. Para isso, é necessário que ele já tenha conhecimento prévio do formato dos protocolos.

Portas Utilizadas

Cada protocolo utiliza portas TCP diferentes para a troca de dados. Tendo conhecimento da relação porta/serviço é possível identificar qual mecanismo de *Instant Messaging* será estudado.

Protocolo	Porta
MSNP	1863
XMPP/Jabber	5222
Yahoo! Messenger	5050
Gadu-Gadu	1550
IRC	6667
OSCAR	5190

Tabela 2 Relação entre protocolos e portas utilizadas

Identificação do Usuário

Em algumas arquiteturas de *Instant Messaging* é possível identificar o protocolo usado apenas com a identificação do usuário, como por exemplo usuario@hotmail.com sendo do MSNP e usuario@jabber.org pertencendo ao Jabber/XMPP. Por outro lado, outras arquiteturas não possuem o formato com '@', logo são necessários outros meios de identificá-las. Além disso, há arquiteturas que suportam nomes de usuários de outros fornecedores, como é o caso do protocolo de notificação da Microsoft. Portanto, esse método deve ser usado em conjunto com outros, já que não fornece uma identificação totalmente confiável do protocolo alvo a ser investigado.

Coleta de Informações

O próximo passo, após a identificação do protocolo que será periciado, é a coleta de informações. Há duas formas de realizar essa busca por dados: Localmente ou Remotamente.
No método local, o analista busca no sistema operacional por arquivos de *logs* ou qualquer outro tipo de dado que registre o uso de programas *Instant Messaging*. As vantagens desse processo são:

- Velocidade, pois o perito terá rápido acesso às informações.
- Exame já realizado pelo software que gerou o *log*.
- Não necessidade de uma infra-estrutura anterior.

A desvantagem é que são poucos os programas de i*nstant messaging* que criam, por padrão, arquivos de *logs* das conversas entre usuários. Além disso, as mensagens entre cliente e servidor não são gravadas.
O processo de coleta remota tem a vantagem de capturar todas as informações, tanto de cliente-cliente, quanto de cliente-servidor. A desvantagem é a necessidade de uma infra-estrutura anterior, como a instalação de um sistema de captura na rede. Após a instalação desse sistema, será possível o analista configurá-lo de forma adequada, usando uma filtragem correta de pacotes, por exemplo, a fim de otimizar a captura e não gastar processamento ou espaço com dados inúteis para a investigação.

Exame de pacotes Cliente-Servidor

Caso o analista tenha usado o processo de coleta remota, é necessário que se realize um exame dos dados a fim de retirar tudo que possa atrapalhar a posterior análise.
Antes de examinar os pacotes trocados entre Cliente-Servidor, o perito deve entender o funcionamento do protocolo e analisar quais são informações relevantes que podem gerar provas eficazes.

Tipo de pacote	Significado
Autenticação	Pacotes que podem mostrar usuário e senha caso não sejam criptografados.
Lista de Contatos	Indica quais são os contatos do usuário, a que grupo pertencem, etc.
Adição de Contatos	Indica que o usuário adicionou um contato na sua lista.
Deleção de Contatos	Indica que o usuário removeu um contato da sua lista.
Pedidos de conversa	Alguns protocolos exigem que se contate o servidor antes de mandar mensagens. Esses pacotes mostram tal processo.
Envio de *Emails*	Alguns protocolos permitem que sejam enviados *emails* durante uma sessão de *instant messaging*.
Display Name	Pacotes que mostram o *display name* do usuário e seus contatos.

Tabela 3 Mensagens relevantes trocadas entre cliente e servidor.

Após identificar os tipos de pacotes interessantes para a investigação, o analista terá que examinar os pacotes. Algumas técnicas utilizadas para facilitar a análise:

- Organização de fluxos *TCP/IP*.
- Remoção dos cabeçalhos.
- Decodificação do *payload* (para protocolos binários).

Organização de fluxos

Uma técnica muito utilizada por diversos programas de captura como o *Wireshark* e ferramentas forense como o *MSN* Shadow é a organização de fluxos *TCP/IP*. Os fluxos são organizados por endereços *IP* e portas *TCP* formando uma sessão. Posteriormente o perito será capaz de analisar cada sessão separadamente, facilitando a visualização de evidências.

Remoção dos cabeçalhos

Outra técnica usada para facilitar a fase de análise é a remoção dos cabeçalhos. A função desse processo é retirar qualquer elemento irrelevante do pacote a fim de deixar apenas a mensagem importante.

Pacotes de mensagens entre cliente e servidor não possuem apenas as informações a serem transportadas, mas também códigos e outros dados de controle que dizem como o servidor deve manipular essa requisição. Alguns desses dados são inúteis para o analista, pois ele não quer manipulá-los e sim visualizá-los. Logo, cabe ao perito a separação entre dados inúteis e dados úteis.

```
SYN 26 2008-03-23T12:55:17.9-07:00 2008-03-27T06:14:31.46-07:00 77 3
GTC A
BLP BL
PRP MFN Usuário
PRP MBE N
PRP WWE 0
LSG Colegas%20de%20trabalho e4044582-6b67-4f29-91c7-019d86cfe4b6
LSG Fam..lia 3dcb3d0f-70c8-4c48-b413-7c86d284255c
LSG Amigos af750ee3-139f-4928-aa14-f6723799b141
LST N=usuario@hotmail.com F=Usuario1 C=1e696892-2ac7-4b26-b697-01ecda94f6ee 11 1
LST N=usuario2@hotmail.com F=Usuario2 C=5a6553ef-9f63-42bb-91fa-0259656b3254 11 1
LST N=usuario3@hotmail.com F=Mensagem do Usuario3 C=582b7b77-6893-42c1-82de-036ad1275876 11 1
```

Tabela 4 Exemplo de pacote trocado numa sessão entre um cliente e servidor.

Na tabela 4 está um exemplo de pacote enviado pelo servidor para o cliente. A informação enviada é sua lista de contatos e nome dos grupos de usuários. Como pode ser visto, há diversos códigos misteriosos para quem não tem conhecimento do protocolo, portanto é necessário que o analista filtre e transforme os dados numa forma na qual seja possível buscar por evidências mais eficientemente. No caso, o mais importante seriam os endereços dos usuários (parâmetro do "*N=*" depois do *LST*) e os nomes dos grupos (parâmetro do *LSG*). As informações restantes no pacote são dados de controle trocados entre cliente e servidor, cujo valor é menos relevante para uma investigação forense. Por exemplo, o parâmetro *PRP MFN* define o nome (*display name*) que será mostrado à lista de contatos.

Decodificação do payload

A decodificação de pacotes que têm como conteúdo texto plano não é necessária, mas alguns protocolos ou serviços oferecidos são binários, logo é necessário o respectivo *codec* (codificador/decodificador) ou algoritmo.

Há diversos algoritmos e *codecs* para diferentes serviços oferecidos hoje pelos mensageiros mais modernos: voz, vídeo, imagens, transferência de arquivos. A análise de arquivos pode ser feita trivialmente por meio de comandos do Unix como *file* e *strings*. O comando file retornará o tipo de arquivo, então o analista poderá buscar qual é o *software* que manipula o artefato recuperado. Já o comando *strings* retorna qualquer tipo de *string* encontrada no arquivo, o que pode ser interessante para localizar informações dentro de arquivos binários.

Há diversos algoritmos de compressão de imagens como *BMP, JPEG e PNG*, mas a maioria dos programas de visualização de imagens tem suporte a todos eles, logo sua visualização é trivial.

A manipulação de fluxos de voz e vídeo é mais complicada. O analista tem que ter em mãos o *codec* necessário, senão terá que usar outras técnicas para periciar as informações, como engenharia reversa.

Apesar das semelhanças, alguns protocolos de mensagens instantâneas têm arquiteturas diferentes, como é o caso do *XMPP/Jabber* e do *MSNP*. O protocolo da Microsoft trata as mensagens trocadas entre clientes em uma conexão *out-of-band*, como utilizado no *FTP*, onde as mensagens de controle e a de dados fluem por conexões diferentes. Já no *XMPP/Jabber*, todos os pacotes, independente se o destino é o servidor ou outro usuário, são enviados pela mesma conexão *TCP/IP*.

Arquiteturas a parte, o MAIDPMI é viável em qualquer desses casos, pois é proposto métodos para diferenciar pacotes interessantes de pacotes fúteis, sem levar em consideração o método utilizado para a captura dessas informações.

Exame de pacotes Cliente-Cliente

Todas as técnicas citadas para perícia de pacotes Cliente-Servidor podem ser usadas nos pacotes Cliente-Cliente. A diferença é a relevância dos pacotes trocados entre clientes e que tipo de informação carregam.

Tipo de pacote	Significado
Mensagens	Conversação entre usuários.
Voz	Conversação usando *VoIP*.

Vídeo	Conversação usando Vídeo.
Transferência de Arquivos	Troca de arquivos entre usuários.
Digitação	Avisa que o usuário está digitando uma mensagem.

Tabela 5 Mensagens trocadas em sessões entre clientes.

```
MSG 14 A 139
MIME-Version: 1.0
Content-Type: text/plain; charset=UTF-8
X-MMS-IM-Format: FN=Helvetica; EF=; CO=000000; CS=0; PF=22
mensagem enviada
```

Tabela 6 Exemplo de cabeçalho de conversação por texto

Na tabela 6 é possível observar um exemplo de pacote Cliente-Cliente enviado. Há o cabeçalho informando o número da mensagem, o número de caracteres enviados, dados de codificação, fonte e cor.

Numa investigação essas informações não são relevantes, logo é recomendável que o analista retire esses dados antes da fase de análise.

Análise Cliente-Servidor

Após a fase de exame, o perito terá que analisar as informações obtidas e verificar o valor destes para o caso [13].

Na maioria das situações, as mensagens trocadas entre Cliente-Servidor são pouco relevantes, mas algumas podem informar dados que complementem ou guiem uma investigação. Por exemplo, um dos pacotes trocados entre cliente e servidor é o envio da lista de contatos. Com acesso a essa informação é possível verificar com quem um usuário possivelmente está se comunicando. Uma empresa que tenha uma política de segurança rígida para mensagens instantâneas corporativas poderá ter acesso à esses contatos e poderá decidir se o usuário está ou não obedecendo as normas estabelecidas.

O excesso de contatos não profissionais de um usuário, por exemplo, pode ser uma evidência de que ele não está produzindo, pois fica muito tempo em conversações com pessoal de fora da empresa. Obviamente, isso por si só não é suficiente. É necessária uma investigação mais detalhada nesse usuário para determinar se ele gasta muito tempo em assuntos desnecessários para o trabalho.

Contatos de concorrentes na lista de um empregado pode ser considerado estranho. Há a possibilidade de apenas amizade entre colegas, mas também de vazamento de informações. A possibilidade ou não de se manter esse contato dependerá da política estabelecida.

A desatualização de uma lista de contatos dos clientes pode causar perda de eficiência e credibilidade, já que o cliente não consegue se comunicar com o prestador de serviços, ocasionando perda de receita da empresa.

Análise Cliente-Cliente

A análise Cliente-Cliente pode ser considerada o passo mais importante numa perícia de um protocolo de mensagem instantânea, pois é nesse tipo de conexão que as informações mais importantes trafegam.

Após o exame desse tipo de pacote, o qual inclui a decodificação de mensagens binárias e redução de dados úteis, o analista está pronto para adequar as evidências e transformá-las em provas.

O processo de análise está intimamente ligado à política de segurança, pois nela estão definidas as regras de quem terá acesso a esses dados analisados e como estes serão armazenados, portanto é necessário que o perito siga rigorosamente essas diretrizes para que a privacidade e informações importantes para a investigação não sejam comprometidas.

Alguns problemas a serem abordados numa perícia dessa natureza são: abuso no uso de mensagens instantâneas, violação de privacidade e vazamento de informações.

Um dos objetivos de uma perícia forense de rede pode ser a avaliação do uso de uma determinada tecnologia, como *software* de mensagens instantâneas. O abuso nesse tipo de protocolo é muito comum, sendo alguns exemplos o excesso de conversas desnecessárias que podem levar a diminuição da produtividade de um funcionário, até o uso desse tipo de programa para transferência de arquivos ilegais como aqueles protegidos por direitos autorais ou arquivos de pedofilia.

Com os dados prontos para serem analisados, o perito terá a capacidade de verificar se existe abuso na rede. Umas das técnicas pode ser o número de pacotes enviados/recebidos por um determinado endereço IP na rede interna. Caso esse número seja elevado, é necessário analisar o conteúdo desse tráfego e verificar se existe uma razão para esse tempo gasto em conversação. Outra técnica, caso a política de segurança permita, é verificar se o conteúdo das mensagens enviadas é adequado para a conversação corporativa.

A violação de privacidade pode ser identificada verificando os pacotes duplicados na rede e seus endereços de camada de enlace. Quando o mesmo pacote é enviado para destinos diferentes, é provável que esteja acontecendo um ataque de falsificação *ARP*. Verificando-se os endereços *MAC*, será possível visualizar quem é o atacante e quem é a vítima. O atacante é o primeiro a receber o pacote com *MAC* forjado e depois ele irá reenviar o mesmo pacote, mas com endereço *MAC* da vítima. Verificando esse tipo de comportamento na rede, o perito será capaz de identificar o atacante.

O caso de vazamento de informações pode ser considerado o mais grave dentre todos aqueles que devem ser abordados dentro de uma política de segurança. Esse processo é muito delicado, pois exige que o analista tenha acesso a todo conteúdo trafegado pelo usuário; e a determinação de qual analista será o responsável pela perícia e como esses dados serão armazenados são de difícil planejamento.

O vazamento ou roubo de informações pode ser identificado de diversas formas: informações de texto, informações de vídeo, informações de voz e transferência de arquivos.

A verificação das transferências de arquivos executadas pelo usuário é muito importante não apenas para gestão de tráfego como também averiguação de vazamento de dados sigilosos. Tendo conhecimento apenas do destinatário, já é possível pressupor o tipo de informação trafegada, como por exemplo, um funcionário interno enviando um arquivo de banco de dados para um funcionário concorrente. Esse tipo de atitude é no mínimo estranho.

Nas informações trafegadas por texto, um *insider* (atacante interno da empresa), é capaz de enviar qualquer informação considerada binária como arquivos de escritório (*Word e Excel,* por exemplo), algoritmos internos da empresa, informações de lançamentos de novos produtos, etc. Com o uso de ferramentas como o *uuencode*, é possível codificar um arquivo binário em texto, e assim enviar tal informação numa janela de conversação qualquer.

Caso o atacante perceba que esse tipo de tráfego está sendo monitorado, é possível que ele use a conversação por vídeo para roubo de informações. Um exemplo desse ataque é o uso de uma *webcam* para filmar e enviar imagens de documentos importantes da empresa, imagens do *desktop* do usuário ou algo escrito numa simples folha de papel.

Outro meio de transferir dados internos de uma empresa é através de voz sobre IP. Com o uso dessa tecnologia é possível se comunicar com qualquer pessoa em qualquer parte do mundo, muitas vezes de graça ou a custos baixíssimos. Mas também é possível trafegar informações sigilosas nesses canais, e sem o devido monitoramento, é possível o atacante usar a própria infra-estrutura da empresa para prejudicá-la.

Apresentação Final

Na fase de Apresentação Final o analista fará um relatório de tudo que foi capturado, examinado e analisado. A partir daí, dará suas conclusões do que aconteceu e também deverá mostrar soluções em casos que exijam alterações na conduta da empresa e/ou empregados.

IV. MSN SHADOW

O objetivo do projeto apresentado é a criação de uma ferramenta de análise forense direcionada à arquitetura de mensagens instantâneas, com foco no protocolo de notificação da Microsoft.

Ao longo de seu desenvolvimento, diversas características foram criadas para facilitar o trabalho de um analista nessa área.

- Decodificação de conversação por texto.
- Decodificação de conversação por video.
- Criação de pacotes-texto falsos.
- Seqüestro de sessão *MSN.*
- Queda de conexão.
- Captura de lista de contatos.
- Relatório em *HTML.*
- Armazenamento do fluxo de vídeo em formato *.AVI.*
- Leitura de arquivos *PCAP.*

Como pode ser observado, na lista acima, o software desenvolvido nesse trabalho permite ao analista visualizar diversas informações relevantes numa arquitetura de mensagem instantânea, além de não precisar ter vastos conhecimentos na área, pois o programa já realiza diversas fases do exame, deixando apenas o interessante para o investigador.

Bibliotecas utilizadas

Diversas bibliotecas de programação foram utilizadas durante o desenvolvimento desse projeto.

A captura dos pacotes é feita usando a *libpcap*. Essa *API*, utilizada por diversos softwares conhecidos como *Wireshark e Ettercap*.

Quando um pacote texto é capturado pelo projeto apresentado, toda a programação é feita utilizando a API QT [23], a qual tem diversos métodos e classes que ajudam na manipulação de *strings* e outros objetos gráficos.

A decodificação de pacotes de vídeo é mais complicada e exige uma biblioteca específica para essa tarefa: a *libmimic* [19].

Essa API foi criada pelo programador Ole André Vadla Ravnås [20], o qual realizou engenharia reversa no cliente oficial da Microsoft, a fim de ter acesso à codificação utilizada pelo MSN, a MIMIC v2.x.

As técnicas de investigação ativa, realizadas por este projeto, têm como base as falhas inerentes da suíte *TCP/IP*. Para a criação dos pacotes *TCP e IP* forjados, foi utilizada a API de *sockets* UNIX [21].

O software *MSN Shadow* mantém em memória todas as mensagens trocadas entre clientes e servidores e atualiza os números de sincronização e reconhecimento do *TCP*. Assim que o analista, que estiver usando o software, precisar seqüestrar a sessão, o *MSN Shadow* criará um pacote *raw*, usando *API UNIX*, e o enviará para a rede. É necessário que o usuário seja *root* para ter permissões de criação de pacotes.

Ambiente de testes

Para a realização da fase de testes, foi criado um ambiente no qual diversos *softwares* e diversos sistemas operacionais fossem testados e analisados, a fim de verificar a competência do software *MSN Shadow*.

Sistema Operacional	Cliente de Instant Messaging
Microsoft Windows XP	*Messenger Live!*
Ubuntu Linux	*aMSN*
Ubuntu Linux	*Kopete*

Tabela 7 Ambiente de testes.

Na tabela 7 pode ser visto a variedade de sistemas operacionais e clientes, sobre os quais, o projeto apresentado foi testado.

Essa diversificação no parque tecnológico é importante devido às várias versões do protocolo de notificação da Microsoft. Dependendo da versão utilizada pelo software cliente, é possível que a manipulação de determinados tipos de pacotes sejam diferentes, o que necessitará em diferentes implementações no projeto final.

Configuração do sistema de monitoramento

Durante a fase de testes na captura e decodificação, além do software desenvolvido nesse projeto, outra ferramenta foi utilizada para auxiliar o monitoramento: o *arpspoof* [22].

Como a rede de testes utiliza *switch*, é necessário realizar o envenenamento de *cache* para forçar o sistema operacional a enviar o pacote para o sistema de captura. Nem sempre esses softwares são usados para ataques, como nesse caso.

```
arpspoof -t 192.168.0.1 192.168.0.5
arpspoof -t 192.168.0.5 192.168.0.1
```

Tabela 8 Comandos executados para envenenar o *cache* do sistema alvo.

Na tabela 8 é mostrado os dois comandos executados para que tanto o sistema monitorado quanto o seu *gateway* sejam configurados para enviar dados para o sistema de monitoramento. Isso é necessário, pois o protocolo escolhido é desenvolvido para a *Internet*, logo não funciona localmente, e sem os pacotes advindos do *gateway*, parte da investigação será perdida.

O próximo passo é configurar o sistema de captura para reenviar os pacotes recebidos e que não são destinados a ele.

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Tabela 9 Ativando o redirecionamento de pacotes no kernel do Linux.

O comando mostrado na tabela 9 deve ser executado como usuário *root*, já que é uma alteração no *kernel* e exige nível de administrador.

Captura e decodificação de pacotes texto

Na fase de testes de captura e decodificação de pacotes texto, a ferramenta forense apresentada nesse projeto foi instalada no sistema de monitoramento já configurado.

Na figura 4 é mostrada janela de configuração do software MSN Shadow. Nessa janela é possível configurar diversas informações necessárias para o trabalho de análise e captura das informações:

- *Rule for Sniffing*: Nesse campo é possível que o analista informe a regra que será usada para captura dos pacotes tanto de vídeo como de texto. Essa regra será passada à *libpcap*, a qual será encarregada de capturar os pacotes, logo é necessário que essa informação siga o padrão *Berkeley Packet Filter*.
- *Ignore packets with source MAC of the interface*: Essa caixa de seleção permite o analista configurar o software para que este ignore pacotes com endereço MAC de origem igual ao da interface monitorada. Essa opção é importante, pois quando o sistema de captura está num ambiente no qual é necessário redirecionamento de pacotes, o software irá capturar duas vezes o mesmo pacote, mostrando informações redundantes. Logo, com essa opção, é possível o programa de monitoramento ignorar pacotes reenviados. A escolha dessa opção também gera um efeito colateral, no qual o sistema de captura não pode ser usado como cliente *MSN*, já que seus pacotes serão ignorados e parte da investigação será perdida.
- *Interface*: Nessa caixa de seleção é possível escolher dentre as interfaces instaladas no sistema de monitoramento, qual será aquela utilizada.
- *Mencoder Path*: Nesse campo é necessário informar o caminho do binário do software ***mencoder*** [24], o qual será utilizado para gravação da captura de vídeo. Esse programa não é necessário para a captura do vídeo, mas é necessário caso o analista deseje armazenar a captura em arquivo.

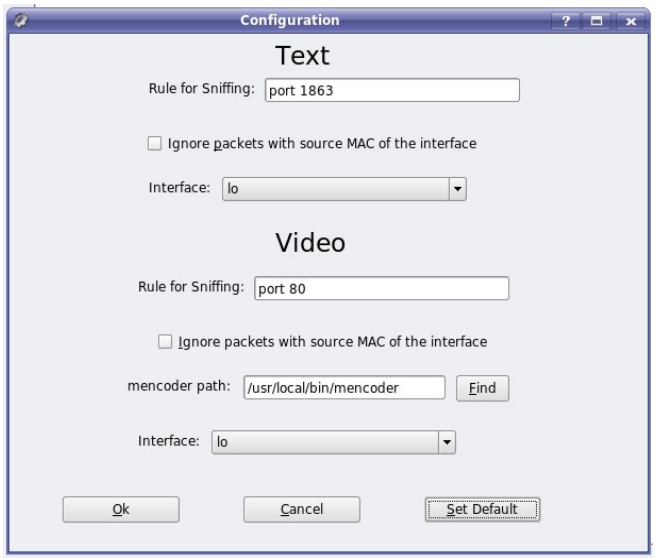

Figura 4 Janela de configuração do *MSN Shadow*

Como observado na figura 5, a ferramenta forense *MSN Shadow* mostra as conversações texto já decodificadas, ou seja, sem os dados desnecessários do cabeçalho de controle do protocolo. Além disso, apresenta para o analista os horários nos quais os pacotes foram recebidos na placa de rede e os nomes de usuários dos participantes. Com a opção de leitura de arquivos *.pcap*, é possível um investigador constituir conversações que já aconteceram.

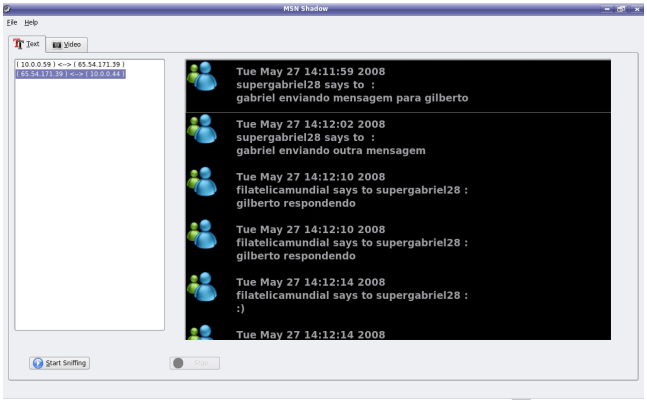

Figura 5 Resultados obtidos durante a captura de conversação por texto.

A figura 6 mostra o mesmo fluxo de pacotes, mas decodificado pelo *software Wireshark*. Na imagem é possível observar que o pacote de conversação inclui mais informações, as quais, na sua grande maioria, são inúteis para uma investigação forense com foco em mensagem instantânea. Além disso, fica difícil reconstituir de onde vem cada mensagem, pois o pacote apenas informa o usuário de origem, e a associação do usuário de destino com endereço *IP* deve ser feita manualmente.

O software apresentado é capaz de dividir os fluxos de acordo com o endereço *IP* de origem e o endereço *IP* de destino, como pode ser observado na figura 5, ao lado esquerdo.

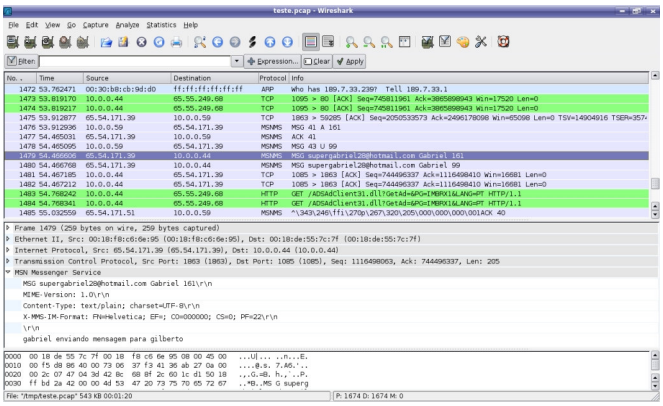

Figura 6 Pacotes do protocolo MSNP apresentados no s*oftware Wireshark.*

Todo esse processamento é realizado sobre pacotes enviados no modo *online,* pois foi constatado que o cliente mensageiro cifra as mensagens enviadas quando o usuário destino não está conectado ou este ativa a opção de aparecer *offline*. Nesse caso, o usuário não irá enviar a mensagem através de um servidor *switchboard*, mas irá abrir um canal de comunicação criptografado com um dos servidores da *Microsoft*, utilizando a tecnologia *SSL/TLS*, e, a partir daí, o servidor irá retransmitir essas mensagens, também de modo cifrado, para o cliente destino.

Usando essa curiosa arquitetura, é possível assegurar as comunicações desse protocolo apenas habilitando o modo desconectado nos dois usuários comunicantes. Dessa forma, o tráfego cliente-cliente estará protegido, mas não a conexão cliente-servidor, a qual ainda pode ser manipulada.

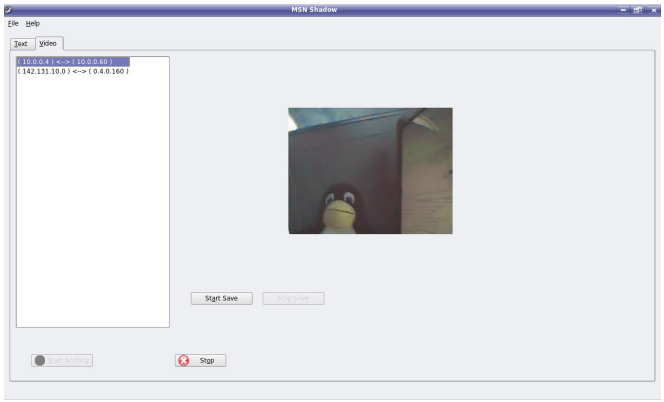

Figura 7 Janela do *MSN Shadow* durante a captura de fluxo de vídeo.

Captura e decodificação de pacotes de vídeo

Outra característica interessante do software apresentado é a capacidade de capturar e decodificar pacotes de transmissão de vídeo sobre o protocolo do *MSN*.
Como pode ser visto na figura 7, a divisão do fluxo de vídeo é feita da mesma forma da captura de texto. Além disso, é possível ver o vídeo capturado em tempo-real e gravá-lo em arquivo no formato *.AVI*.

A parte do projeto com relação à decodificação do vídeo foi interessante, pois houve necessidade de se fazer contato com o criador da API *libmimic* para que este esclarecesse uma dúvida. A questão abordada foi a possibilidade de decodificar vídeos sem ter o quadro de inicialização necessário para iniciar a API. Segundo o autor, seria impossível sem tal informação.

Então foi implementado diretamente no código fonte, um *frame* de inicialização capturado durante uma sessão normal de vídeo. A cada momento que um usuário do *MSN Shadow* inicializa uma sessão de monitoramento de vídeo, esse quadro é utilizado, e a captura acontece normalmente, ou seja, independentemente do quadro de inicialização, é possível capturar e decodificar pacotes de vídeo-conferência do *Microsoft Notification Protocol.*

Seqüestro de sessão e forjamento de mensagens

O seqüestro de sessão e o forjamento de mensagens são muito semelhantes, a única diferença é a configuração de regras no IPTables para que pacotes sejam bloqueados durante o seqüestro de uma sessão.

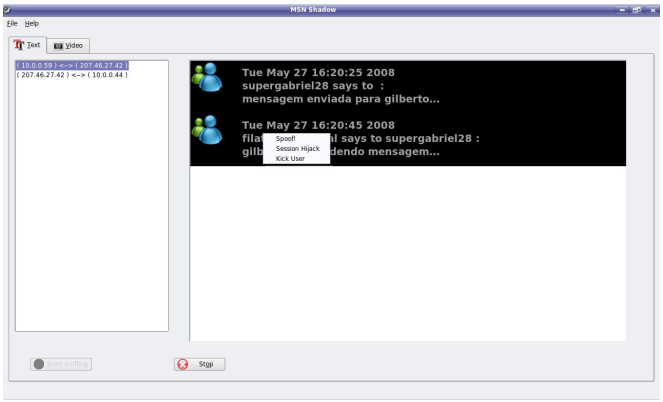

Figura 8 Menu de escolha.

Na figura 8 é mostrado o menu de seleção que aparece quando o botão direito do *mouse* é pressionado sobre as mensagens capturadas.

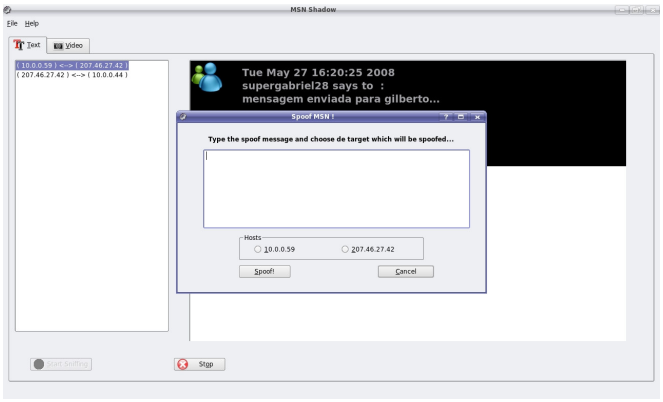

Figura 9 Janela de forjamento de mensagens.

O menu permite escolher dentre várias opções:

- *Spoof!* : Permite o analista forjar um pacote em particular.
- *Session Hijack*: Permite o analista seqüestrar toda a sessão *MSN*.
- *Kick User*: Permite o analista desconectar um usuário da rede *MSN*. Apenas é possível para usuários da rede interna.

A figura 9 mostra a janela de *spoofing*. Nessa janela é possível escolher qual dos dois endereços IP será forjado e qual a mensagem será enviada. O *software* irá montar automaticamente o cabeçalho necessário desde a camada de rede, até a camada de aplicação.

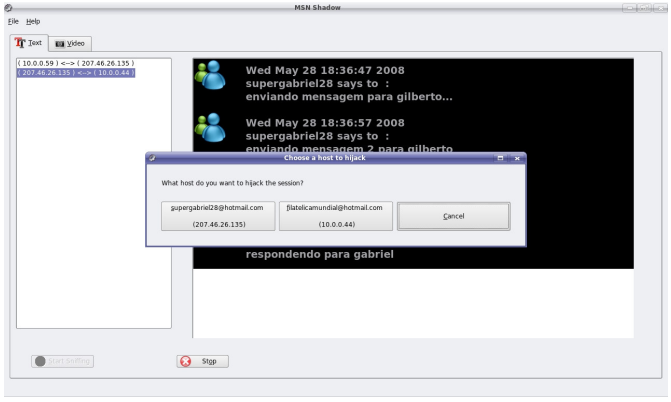

Figura 10 Escolha da conexão alvo.

A imagem 10 mostra a janela na qual é possível escolher qual será a conexão seqüestrada.
Na imagem 11 é apresentado os resultados de um seqüestro de sessão *MSN*. As mensagens enviadas tanto pelo analista que está atacando a conexão, quanto o as mensagens enviados pelo usuário atacado serão mostradas numa nova tela, como mostra a figura 11.

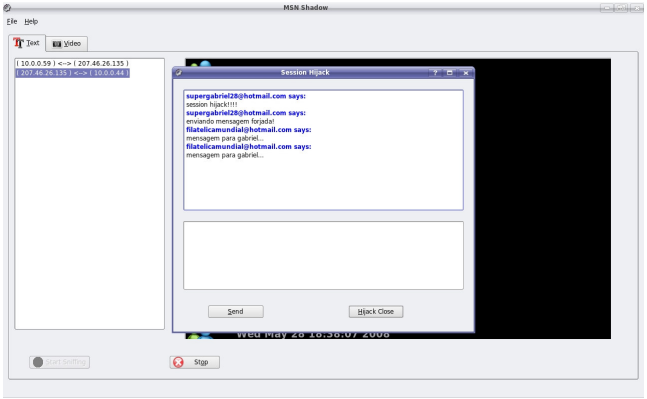

Figura 11 Janela de seqüestro de sessão.

Captura e decodificação de lista de contatos

A figura 12 mostra a tela correspondente à captura de uma lista de contatos.

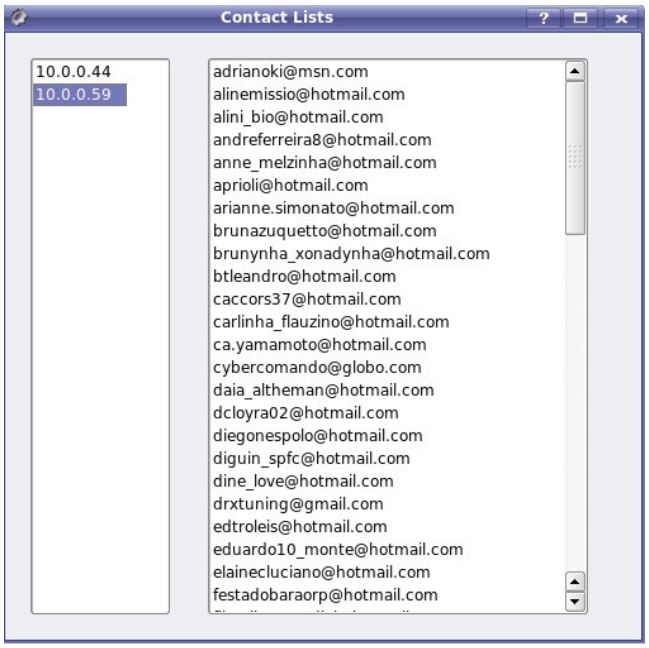

Figura 12 Captura da lista de contatos.

Geração de relatórios HTML

Na imagem 13 é mostrado o relatório HTML gerado após a captura de uma sessão MSN.

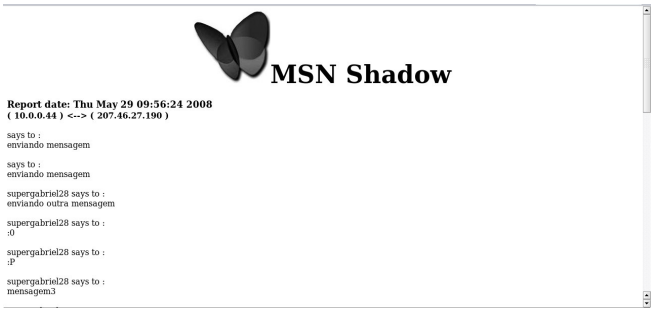

Figura 13 Relatório gerado em formato HTML.

V. CONCLUSÃO

Ferramenta e modelo desenvolvidos

O modelo Abstrato de Investigação Digital de Protocolos de Mensagens Instantâneas foi criado com a intenção de ser um guia para qualquer profissional com o dever de analisar protocolos de mensagens instantâneas, além de ajudar no desenvolvimento de ferramentas como a apresentada nesse trabalho. Abordando todas as peculiaridades desse tipo de tecnologia, o modelo proposto é importante para que nenhuma informação seja perdida nesses tipos de investigações, e, além disso, define quais são as informações irrelevantes que devem ser ignoradas a fim de otimizar a posterior análise manual do investigador.

A ferramenta *MSN Shadow*, criada sobre o modelo proposto, vem a ser um software poderoso de perícia forense de mensagem instantânea, com foco no protocolo de notificação da *Microsoft*. Essa ferramenta tem diversas características que ajudam muito o trabalho do profissional que necessita de informações advindas dessa tecnologia: captura e decodificação de pacotes de conversação por texto, vídeo, lista de contatos, leitura de arquivos de captura e técnicas de investigação ativa.

Resultados Obtidos

Os resultados obtidos e demonstrados no capítulo anterior provam a eficiência tanto do modelo proposto quanto da ferramenta desenvolvida, pois apresenta diversas imagens que ilustram a facilidade de uso e eficácia no exame de informações realizados pelo software *MSN Shadow*.

Caso o analista encarregado de realizar alguma investigação em tecnologia de mensagens instantâneas não tenha acesso à esse tipo de software, será necessário que diversas informações sejam analisadas manualmente com ferramentas como *tcpdump* ou *Wireshark*. Essa análise manual, além de ser lenta e custosa, pode levar a erros e perdas de informações relevantes.

A ferramenta forense *MSN Shadow* é capaz de diminuir o tempo de trabalho do analista, retirando do fluxo de mensagens instantâneas informações inúteis e apresentando apenas aquilo que é relevante para a investigação.

Trabalhos futuros

A cada dia surgem novas tecnologias e a computação, principalmente a *Internet*, está cada vez mais presente na vida de todos os usuários, não apenas na dos técnicos, mas também na dos leigos. Há também um grande aumento no uso de tecnologia para a realização de fraudes, golpes e outros tipos de crimes, logo, é necessária a criação de técnicas que permitam investigações e busca por evidências em qualquer tecnologia, independente se este é um *hardware* como um *pendrive*, um protocolo ou um *software*.

O projeto apresentado está focado num único protocolo, o *Microsoft Notification Protocol*, mas existem inúmeros outros protocolos de mensagens instantâneas, cada um com seus cabeçalhos e informações de controle, portanto é necessária a implementação de diferentes códigos para diferentes protocolos.

Além disso, outras informações importantes que são trafegadas por fluxo de mensagens instantâneas como arquivos, não são manipulados pelo projeto apresentado, mas também são fontes de evidências.

Outro fator a ser abordado num futuro próximo é o uso de criptografia no fluxo de rede, impedindo a sua captura. Essa abordagem é interessante para o aumento na segurança do tráfego, mas também impede a análise forense neste meio. Por isso, a tendência num futuro não muito distante, é a migração de ferramentas de análise de rede para ferramentas de análise de *host*, as quais devem executar em nível de *kernel* do sistema operacional e capturar todas as informações antes que sejam criptografadas. Apenas assim, as evidências poderão ser coletadas.

VI. REFERÊNCIAS

[1] M. Day, J. Rosenberg, and H. Sugano. A Model for Presence and Instant Messaging, RFC 2778. IETF, Feb 2000.

[2] S. M. Cherry, “IM means business,” IEEE Spectrum Online, vol. 39, pp. 28–32, Nov. 2002.

[3] HYPOTHETIC.ORG. MSN Messenger Service 1.0 Protocol. Disponível em <http://www.hypothetic.org/docs/msn/ietf_draft.txt>. Acessado em: 22 mar. 2008.

[4] ET Nakamura, PL de Geus - Editora Berkeley, Sao Paulo, Brasil, 2007

[5] E. Casey, “Network traffic as a source of evidence: tool strengths, weaknesses, and future needs”, *Journal of Digital Investigation* 1, 1 (2004)

[6] V. Corey, C. Peterman, S. Shearin, M. S. Greenberg and J. Van Bokkelen (2002 December) "Network Forensics Analysis". IEEE INTERNET COMPUTING pages 60-66

[7] P. Resnick. Internet message format. RFC 2822, IETF, April 2001.

[8] OPEN WEB APPLICATION SECURITY PROJECT. Top Ten. Disponível em:<http://www.owasp.org/index.php/Top_10_2007>. Acessado em 22 mar. 2008.

[9] Mannan, Mohammad. and van Oorschot, Paul. Secure public Instant Messaging: A survey. Proceedings of Privacy, Security and Trust, 2004.

[10] MSNPIKI. Unofficial MSN protocol documentation. Disponível em:<http://msnpiki.msnfanatic.com/index.php/Main_Page>. Acessado em 22 mar. 2008.

[11] S. M. Bellovin, “Security problems in the TCP/IP protocol suite”, *Computer Communications Review*, 19(2):32 48, Apr. 1989.

[12] National Institute of Justice.(July 2001) Electronic Crime Scene Investigation A Guide for First Responders.

[13] Baryamureeba, Venansius and Florence Tushabe. The enhanced digital investigation process model. In Digital Forensics Research Workshop (DFRWS), Baltimore, Maryland, August 2004.

[14] M.Reith, C. Carr and G. Gunsch, “An Examination of Digital Forensic Models”, *International Journal of Digital Evidence*, Fall 2002,Volume 1, Issue 3.

[15] J. Postel and J. Reynolds, “RFC 959: File Transfer Protocol (FTP)”, *Technical report*, Oct. 1985.

[16] JB Postel - Simple Mail Transfer Protocol, August, 1982.

[17] J Myers, M Rose - Post Office Protocol-version 3, 1996.

[18] W.R. Stevens, TCP/IP Illustrated, Volume 1: The Protocols, *Addison-Wesley*, 1994.

[19] FAIRSIGHT. libmimic. Disponível em:<http://farsight.sourceforge.net/>. Acessado em 22 mar. 2008.

[20] SLASHDOT. Logitech MSN Webcam Codec Reverse-Engineered. Disponível em:<http://linux.slashdot.org/article.pl?sid=05/04/05/0240236&tid=215&tid=188&tid=106>. Acessado em 22 mar. 2008.

[21] R. Stevens, “Unix Network Programming, Volume 1”, 2nd ed., *Prentice Hall*, 1998.

[22] DSNIFF. arpspoof. Disponível em:<http://www.monkey.org/~dugsong/dsniff/>. Acessado em 22 mar. 2008.

[23] TROLLTECH. qt. Disponível em:<http://trolltech.com/>. Acessado em 22 mar. 2008.

[24] MPLAYER - THE MOVIE PLAYER. mencoder. Disponível em:<http://www.mplayerhq.hu/>. Acessado em 22 mar. 2008.

[25] Nikkel Bruce J., "Improving evidence acquisition from live network sources", *Digital Investigation 2006;* 3(2), pp89 -- 96