# Arquitetura de Agentes Inteligentes no auxílio à perícia forense computacional

Petroni, B. C. A., Sanchez, P. L. P.

***Abstract* — A arquitetura proposta neste trabalho demonstra a aplicabilidade de agentes inteligentes no auxílio à perícia forense computacional. Aplicações que utilizam técnicas embasadas em sistemas inteligentes estão merecendo atenção, devido ao excelente retorno de desempenho e confiabilidade, conforme visto em trabalhos como Nogueira [16] e Muhammad [17] dentre outros. O principal objetivo deste, é demonstrar uma arquitetura onde agentes inteligentes atuam em uma rede de computadores na proteção contra intrusões, conhecida como sistemas IDS, e em seguida é elaborado um breve comparativo ilustrados em gráficos com outra ferramenta que possui sua arquitetura embasada na técnica redes neurais artificiais. Importante ressaltar que neste trabalho é acrescentada uma contribuição através de uma análise comparativa de uma aplicação real, não se restringindo somente a descrições ou modelos de códigos fontes em ambientes de pesquisas, o que atualmente observa-se com artigos que utilizam dessas técnicas, especificamente para as aplicações contra intrusões em redes de computadores. Por fim, este trabalho apresenta alguns questionamentos com relação a análises periciais onde essa arquitetura pode contribuir para identificações e questionamentos.**

***Index Terms* — Inteligência Artificial, Agentes Inteligentes, TCP/IP, Computação Forense, Perícia.**

## I. INTRODUÇÃO

ESTE trabalho tem como objetivo principal fazer uma proposta de arquitetura de sistema que se baseia na tecnologia de detecção de intrusos ou IDS (Intrusion Detection System) em redes de computadores com protocolos TCP/IP, que é orientada por Agentes Inteligentes, que é uma das várias tecnologias embasadas em Inteligência Artificial, da área de Sistemas Inteligentes.

A verificação desses intrusos é necessária pelo fato de que as intrusões são advindas de ataques a sistemas computacionais, que afetam tanto computadores de usuários residenciais como de organizações empresariais.

Posteriormente a partir da análise mencionada a idéia é fazer um comparativo e retirar alguns dados estatísticos a fim de verificar as qualidades da nova técnica em confronto com uma tecnologia já existente e utilizada.

Por fim, fazemos alguns questionamentos da aplicabilidade desta nova arquitetura no auxílio das atividades do perito judicial.

## II. CRIMES POR COMPUTADOR

É publico e notório que a informação é o elemento fundamental na busca pelo conhecimento para todos os usuários de computadores.

Porém, o fato de possuir informações está diretamente ligado à transmissão e delegação de poderes para as pessoas. Paralelo a isso existe outra questão, que é a manutenção dessas informações em sigilo e metodologias de segurança, que devem ser aplicadas na disseminação controlada da informação. Essas questões acabam gerando uma grande preocupação na sociedade e nas organizações como um todo.

Conforme [1], com o avanço e a popularização do computador, no inicio da década de 1990, p mundo começou a assistir a uma verdadeira revolução nos costumes, à medida que estas máquinas invadiam, cada vez mais, os lares das famílias de todo o planeta. Em paralelo e como decorrência deste fenômeno, a Internet experimentava um gigantesco *boom* com seu crescimento.

Com a passagem para o novo milênio, pessoas de todas as classes e lugares passaram a ficar a cada dia mais próxima de tecnologias de comunicação e armazenamento e, cada vez mais acessam a grande rede através de computadores instalados em suas casas, trabalho e lugares públicos.

A idéia deste trabalho é apresentar uma arquitetura utilizando a técnica de Inteligência Artificial – (IA) conhecida como Agentes Inteligentes – (AI) aplicável tanto na prevenção de incidentes como no auxílio à computação forense no trabalho do perito judicial atuante nas áreas de Sistemas de Informação – (SI), e Tecnologia da Informação – (TI).

A arquitetura apresentada permite um monitoramente e observação de todos os serviços oferecidos aos usuários em uma rede de computadores que utiliza o protocolo TCP/IP para comunicação.

### *A. Detecção de Intrusão*

Intrusão é uma violação da política de segurança do sistema computacional. Esta definição é geral o suficiente para abranger todo tipo de ataque. Em muitos casos as políticas de segurança não podem ser traduzidas num conjunto de regras rígido, haja vista a própria natureza mutante dos eventos de segurança.

Enquanto a política de segurança define as metas que devem ser cumpridas num sistema, a detecção das violações desta política requer o conhecimento dos passos ou ações que possam resultar nessa violação [4].

As técnicas utilizadas para a detecção de intrusões podem ser classificadas em duas categorias: detecção de anomalias e detecção de uso indevido, ambas com vantagens e inconvenientes [6].

Dentro das classificações descritas têm-se: [4].

- **Anomalia**: baseada na determinação de comportamento anômalo no uso de recursos do sistema. Por exemplo: se normalmente um determinado usuário A do departamento de vendas de uma empresa somente usa sua conexão de rede de segunda a sexta-feira entre 08:00 e 16:00 horas, uma atividade noturna na conta desse usuário, é anormal, e pode significar uma intrusão. Nesse caso tenta-se quantificar o comportamento usual ou aceitável, indicando-se outros comportamentos irregulares como sendo potencialmente intrusivos. Esta técnica é sujeita a um certo grau de incerteza.
- **Detecção por uso abusivo**: refere-se a intrusões que seguem um padrão bem definido de ataque, que explora vulnerabilidades no sistema ou nos softwares aplicativos. Esses modelos podem ser prévia e precisamente escritos ou identificados, o que oferece um maior grau de certeza do resultado.

A investigação sobre intrusões e detecção de falhas foi iniciada no início da década de 1980 com a introdução do conceito de "*computer threats*" e "detecção de uso indevido" [6].

### *B. Computação Forense*

A computação forense é uma ciência voltada para o estudo avaliação, investigação e análises de evidências dentre várias situações que envolvam a computação como meio para cometer crimes. Muitas pessoas acreditam que a atividade pericial na computação é recente, devido ao pouco tempo em que a informática vem fazendo parte de nossas vidas, mas essa ciência é um pouco mais antiga do que nós imaginamos [1].

A computação forense trabalha como coleta e análise de dados de sistemas computadorizados, redes ou quaisquer outros dispositivos que permitam armazenamento de informações.

Isso porque com o computador, vieram os crimes que o têm com instrumento ou como alvo. No Brasil, de uma maneira geral, os trabalhos periciais ainda estão sendo realizados de forma incipiente, mas alguns Institutos de Criminalística têm-se preocupado em criar uma estrutura mínima para realizar o trabalho com qualidade, deixando de lado o hábito de passar perícias dessa natureza para aquele perito que "entende de informática", mas que, sem nenhuma estrutura e cheio de boa vontade, ainda tenta resolver o problema, trazendo para si todos os riscos de manipular provas extremamente voláteis sem estar equipado e sem os conhecimentos necessários.

Os crimes na verdade podem ser cometidos através de ataques, e as pessoas que realizam esses ataques, buscam obter alguma vantagem ilícita.

Com isso é criado um grande problema para os usuários e organizações. Por exemplo, muitas vezes informações que representam uma valiosa propriedade intelectual são ilicitamente obtidas ou adulteradas, sendo casos típicos o acesso a bases de dados de clientes, informações de parceiros ou outras informações úteis [2].

## III. AGENTES INTELIGENTES

A definição exata do termo Agente Inteligente ainda não está bem estabelecida, visto que existem diferentes posicionamentos relativos a essa questão, o que, de certa forma, dificulta o entendimento dessa tecnologia.

Vários autores dispõem de conceitos diversos como:

- Um agente inteligente deve ter o comportamento de maneira semelhante ao de um ser humano, e em sua essência ser um interlocutor entre o sistema computacional – interface – e o homem. [12].
- Um agente inteligente é um sistema de computador que está situado em determinados ambientes, e que é capaz de ações autônomas neste ambiente numa ordem que satisfaça os objetivos do projeto. [3].
- Outro conceito sobre Agente Inteligente seria um artefato que possui a capacidade de perceber seu ambiente através de sensores (ex: microfone, teclado etc.) e agir sobre ele através de efetuadores (ex: vídeo, alto-falante, impressora, braços mecânicos, entre outros), [10].

Para complementar, Costa [9], compreende agente como toda a entidade capaz de interagir em determinado ambiente, em geral, orientada por objetivos, e as decisões sobre seu uso devem ser tomadas a partir da percepção, de forma que tenham como conseqüência, a execução de uma determinada atividade.

Agentes Inteligentes é uma técnica de Inteligência Artificial que pode ser aplicada em interfaces de sistema computacionais auxiliando usuários durante sua interação. Domínios de aplicação como interfaces, permitem explorar dos Agentes Inteligentes algumas características como autonomia, habilidade social, reatividade e pró-atividade [3].

Agentes Inteligentes podem inclusive podem inclusive ser difundidos, como explica Naedele e Dzung [5], no sentido de diferenciar o objetivo da segurança juntamente com o mecanismo da segurança.

Um Agente Inteligente na verdade é um programa que pode auxiliar o usuário a executar alguma tarefa, e a classificação de Agente Inteligente vem do fato de que esse programa pode ser executado em um ambiente onde ele possa se mover a apresentar algumas propriedades particulares como:

- **Autonomia**: habilidade de tomar decisões sem a interferência do usuário;
- **Reatividade**: capacidade de reagir ao ambiente no qual está inserido, através da detecção de ações externas;
- **Pró-atividade**: capacidade de iniciativa para realizar ações que levem ao cumprimento de seus objetivos;

A massificação e utilização de computadores e redes com um custo mais acessível ampliou o problema de acesso não-autorizado, bem como na possibilidade de manipulação e ataque a dados e recursos. O crescimento da interconexão de redes não só possibilita o rápido acesso a uma maior variedade de computadores, como também garante acesso a dados que podem vir de qualquer lugar. Com freqüência ocorrem

situações onde intrusos facilmente superam os mecanismos de autenticação de senhas, projetados para proteger os sistemas.[4].

Com isso, o crescimento da necessidade de proteção se torna cada vez mais necessárias.

As características mencionadas com relação aos Agentes Inteligentes são muito importantes quando o assunto é a utilização de sistemas computacionais, uma vez que, o meio de relação do usuário com a máquina se dá através da interface computacional e também independente de em determinados momentos estar ou não conectados a rede.

A seguir a Figura 1 ilustra a arquitetura de um Agente Inteligente.


Figura. 1. Arquitetura de um Agente Inteligente[1]

## IV. REDES DE COMPUTADORES COM PROTOCOLO TCP/IP

O Departamento de Defesa dos Estados Unidos patrocinou o desenvolvimento da ARPANET, precursora da grande rede mundial que conhecemos hoje como Internet. O desenvolvimento baseou-se na interligação de centenas de universidades e repartições públicas, usando linhas telefônicas dedicadas.

Posteriormente foram criadas as redes de rádio e satélite, e também começaram a surgir problemas com os protocolos de redes existentes, o que forçou a criação de uma nova arquitetura de referência. Desse modo, a habilidade para conectar várias redes de maneira uniforme foi um dos principais objetivos do desenvolvimento da arquitetura que ficou conhecida como Modelo de Referência TCP/IP (Transmission Control Protocol / Identify Protocol).

Diante da preocupação do Departamento de Defesa dos EUA de que seus preciosos hosts, roteadores e gateways de interconexão de redes fossem destruídos de uma hora para outra, definiu-se também que a rede deveria ser capaz de sobreviver à perda do hardware de sub-redes, com as conversações existentes sendo mantidas em atividade.

Basicamente o Departamento de Defesa dos EUA queria que as conexões permanecessem intactas enquanto que os computadores de origem e os computadores de destino estivessem funcionando, mesmo que algumas máquinas ou linhas de transmissão intermediárias deixassem de operar repentinamente.

O que fora descrito acima é a origem da internet e o início da preocupação de pessoas e organizações no que diz respeito à transmissão de informações via essa rede.

Além disso, tornou-se necessária uma arquitetura flexível, capaz de se adaptar a aplicações com requisitos divergentes como, por exemplo, a transferência de arquivos.

Com isso surgiu o protocolo TCP/IP que serviu para habilitar os programas a estabelecerem conexões entre seus aplicativos.

O TCP envia dados através de circuitos virtuais. Os serviços de rede do protocolo TCP/IP são subdivididos em camadas ou níveis, sendo eles:

Tabela 1: Nível e serviços de rede do protocolo TCP/IP.

| Nível | Serviço de Rede |
|---|---|
| Aplicação | DNS, TELNET, FTP, SMTP |
| Transporte | TCP, UDP |
| Rede | IP, ICMP |
| Enlace de dados | ARP, RARP e interface de rede |
| Físico | Cabo ou outro dispositivo |

Cansian [4], nos atenta a alguns problemas de segurança envolvendo redes que utilizam o protocolo TCP/IP, sendo:

### *A. Captura de arquivos de senhas*

Trata-se de um dos métodos de ataque mais comuns e, portanto, digno de destaque. Neste cenário um atacante usa falhas nos processos que envolvem o TCP/IP para obter acesso a informações privilegiadas, principalmente com relação ao arquivo de senhas do sistema, por exemplo, em um sistema Linux que fica no arquivo (*/etc/passwd/)* ou equivalente em outro sistema operacional.

### *B. Correio eletrônico*

Além de ser o serviço mais usados nas redes, e provavelmente o mais importante, está entre os mais vulneráveis a ataques e abusos, sendo que os servidores de correio eletrônico apresentam muitas vezes mecanismos de autenticação e validação não confiáveis.

### *C. Servidores WWW*

Serviços, tais como hipermídia, que rodam sobre o protocolo HTTP[2], têm apresentado falhas de segurança graves e freqüentes. Estas falhas têm sido relacionadas principalmente com execução privilegiada de Scripts[3] ou programas que permitem acesso a informações sensíveis.

### *D. DNS – Domain Name System*

Uma base de dados que trata da tradução dos nomes de hosts (computadores) para endereços IP[4]. Um intruso que consiga interferir na operação do DNS pode montar uma variedade de ataques, incluindo a obtenção de informações

[1]Adaptado de: COSTA, Ernesto; S., Anabela. Inteligência Artificial: fundamentos e aplicações. Lisboa: FCA Editora de Informática Ltda, 2004.

[2] Hypertext Transfer Protocol: é um protocolo de comunicação utilizado para sistemas de informação de hipermídia distribuídos e colaborativos.

[3] São linguagens de programação executadas do interior de programas e/ou de outras linguagens de programação, não se restringindo aos seus ambientes, servido para estender a funcionalidade de um sistema.

[4] Internet Protocol: número atribuído a um computador na internet.

privilegiadas.

*E. Telnet*

Utilizado como meio intermediário para os ataques a outros protocolos ou processos. O processo de validação é baseado em uma única combinação de usuário e senha.

*F. FTP – File Transport Protocol*

Requer os mesmos cuidados com relação ao Telnet, com a diferença de que com o FTP é possível a conexão de usuários anônimos, sendo necessário evitar que usuários alheios aos sistemas tenham acesso a informações privilegiadas.

## V. Arquitetura Proposta

O modelo proposto neste trabalho baseia-se no conceito de sistema de monitoramente de segurança de rede.

O objetivo dessa arquitetura é propor a detecção de comportamentos intrusivos utilizando o modelo de detecção por abuso, detectando pacotes em formatos de informações para descobrir o comportamento intrusivo, analisar e fornecer informações para a tomada de decisão.

Nessa arquitetura utilizou-se do desenvolvimento dos Agentes Inteligentes utilizando a tecnologia Java[5], linguagem essa que possibilita a utilização em qualquer plataforma de sistema operacional.

Após vermos uma definição básica do conceito de Agentes Inteligentes, vamos agora fazer uma análise mais técnica a fim de ilustrar a arquitetura que será proposta.

Segundo Russel e Norvig [14] , existem alguns tipos básicos de programas que incorporam os princípios essenciais de Agentes Inteligentes. Dessa forma, dois modelos destacam-se:

- **Agentes reativos simples**: nesse modelo mais simples de agente reativo, a principal característica é a utilização de ações com base na percepção atual, quando se ignora o restante do histórico de percepções, pois utiliza o modelo de regra condição-ação, sob uma visão computacional. O funcionamento ocorre da seguinte maneira: em toda a percepção (interpretação do estado atual com entradas provocadas por "estímulos/respostas"), verifica-se a existência de regra(s) correspondente(s) na base de conhecimento, contendo as condições e ações associadas ao estado atual. Posteriormente, após a identificação do estado, localiza-se a regra e é retornada a ação correspondente vinculada à regra. A Figura  a seguir ilustra esse caso.

```
função AGENTE-REATIVO-SIMPLES(percepção) retorna uma ação
    variáveis estáticas: regra, um conjunto de regras condição-ação
        estado ← INTERPRETAR-ENTRADA (percepção)
        regra ← REGRA-CORRESPONDENTE (estado, regras)
        ação ← AÇÃO-DA-REGRA[regra]
        retornar ação
```

[5] Disponível em: http://www.java.com/pt_BR/download/index.jsp

Figura. 2. Modelo de Agentes reativos simples [6].

- **Agentes reativos baseados em modelo**: esse modelo de agente trabalha com a possibilidade de observações parciais e pode controlar parte do ambiente que ele não pode ver no momento. O funcionamento ocorre da seguinte maneira: o agente se encontra em um estado interno observando o ambiente (percepção) para que se possa retornar uma ação. Internamente as regras estão definidas na base de conhecimento, e suas ações serão localizadas através de seu estado (percepções), após a localização na base de conhecimento, retornará a ação a ser feita, sob o formato da(s) regra(s) correspondentes. Para ilustrá-lo, segue a Figura 3 .

```
função AGENTE-REATIVO-COM-ESTADOS(percepção) retorna uma ação
    variáveis estáticas: estado, uma descrição do estado atual do mundo
        regras, um conjunto de regras condição-ação
        ação, a ação mais recente, inicialmente nenhuma
        estado ← ATUALIZAR-ESTADO (estado, ação, percepção)
        regra ← REGRA-CORRESPONDENTE (estado, regras)
        ação ← AÇÃO-DA-REGRA[regra]
        retornar ação
```

Figura. 3. Modelo Agentes reativos simples.

No caso da arquitetura a ser utilizada, será utilizada a função de Agente-Reativo-Simples.

A Figura 4 seguir ilustra o modelo de teste, dois servidores e uma máquina cliente para serem utilizados.

Para este teste foi inserida duas interfaces de redes, uma para cada servidor afim de evitar que ambos os programas de controle em execução pudessem proporcionar algum conflito, uma vez que possui como características comuns a utilização do protocolo TCP/IP.

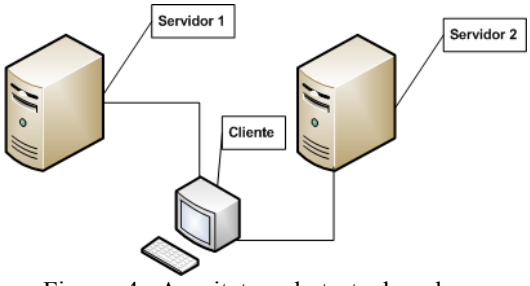


Figura. 4. Arquitetura do teste de rede.

Na Figura 5 está o modelo da arquitetura de checagem de intrusos durante o teste nos dois servidores.

[6] Adaptado de: Petroni, Benedito Cristiano Aparecido AISI – Um sistema inteligente para melhoria da usabilidade e da interação de uma interface web / Benedito Cristiano Aparecido Petroni. – Campinas: PUC – Campinas, 2006.154 p.

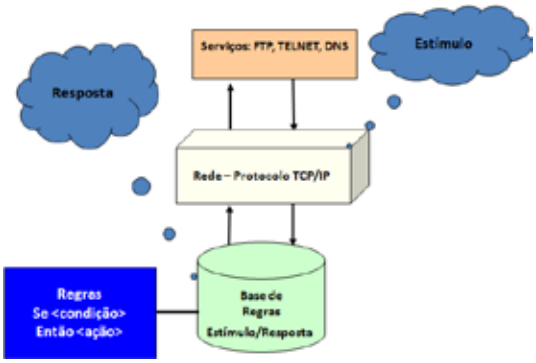


Figura. 5. Arquitetura do Agente reativo simples.

Na Figura 5 pode-se observar que existe a checagem de informações feitas numa Base de Regras, informando os dados que são trafegados pelos serviços oferecidos.

Quando é detectada alguma intrusão, através de um controle conhecido como estimulo, é verificada na Base de Regras e posteriormente vem a Resposta informando sobre a intrusão detectada.

A Base de Regras por sua vez, possui informações que representam características como:

- Caminhos esperados de sessão (origem e destino da sessão) que representam riscos;
- Quais as portas de origem de destino e, por conseqüência, quais serviços estão sendo executados;
- Relação de domínios existentes nas máquinas.

## VI. TESTES COM A ARQUITETURA PROPOSTA

O modo proposto para verificação de resultados foi a utilização de um cenário comparando essa proposta de IDS com uma ferramenta similar, denominada Beholder [13]. O Beholder é um IDS de rede baseado em Redes Neurais Artificiais e que utiliza uma rede neural MLP (Multilayer Perceptron), diferenciando-o do modelo proposto que utiliza Mapas de Kohonen.

Ao contrário do que fora apresentado em [16], esta arquitetura, após sua implementação em testes, será comparada com outra arquitetura baseada em outra técnica de Inteligência Artificial, Redes Neurais Artificiais e seus resultados comparados numericamente e suas descrições utilizando gráficos.

O cenário da arquitetura para testes é o mesmo do projeto Beholder, sendo:

- **Servidor 1**: Sistema Operacional Linux; Servidor Web Apache; Servidor FTP; Servidor SSH; IDS;
- **Servidor 2**: Sistema Operacional Linux; Servidor Web Apache;
- **Cliente**: Sistema Operacional Linux;

A condução para a realização dos testes foi separar em duas etapas, sendo uma etapa num acesso normal aos servidores e outra etapa num acesso simulando um ataque.

A máquina cliente teve dois tipos de comportamentos seguindo também duas etapas, a primeira como atacante efetuando um tipo de varredura de porta, e outra utilizando os serviços da rede de computadores como Web e FTP em ambos os servidores. Os procedimentos de ataque foram feitos pela ferramenta Nmap[7] em sua versão 4.20 e o IDS proposto foi colocado para as duas redes.

### A. *Comparativos de desempenho*

A seguir a Tabela 2 ilustra os resultados comparativos das duas ferramentas IDS.

Tabela 2: Testes de Acesso normal ao Servidor 1.

| Arquitetura Proposta | Beholder |
|---|---|
| **Teste 1**: acesso da máquina cliente no servidor 1 Web. | |
| **Resultado**: nenhuma suspeita, 100% do tráfego reconhecido como normal. Analisaram-se 150 pacotes. | **Resultado**: nenhuma suspeita, 100% do tráfego reconhecido como normal. Analisaram-se 140 pacotes. |
| **Teste 2**: máquina cliente acessa serviço FTP no servidor 1. | |
| **Resultado**: nenhuma suspeita, 100% do tráfego reconhecido como normal. Analisaram-se 73 pacotes. | **Resultado**: nenhuma suspeita, 100% do tráfego reconhecido como normal. Analisaram-se 73 pacotes. |
| **Teste 3**: máquina cliente acessa serviço SSH no servidor 1. | |
| **Resultado**: nenhuma suspeita, 100% do tráfego reconhecido como normal. Analisaram-se 600 pacotes. | **Resultado**: nenhuma suspeita, 100% do tráfego reconhecido como normal. Analisaram-se 600 pacotes. |
| **Teste 4**: máquina cliente acessa serviço no servidor 2. | |
| **Resultado**: nenhuma suspeita, 100% do tráfego reconhecido como normal. Analisaram-se 150 pacotes. | **Resultado**: nenhuma suspeita, 100% do tráfego reconhecido como normal. Analisaram-se 150 pacotes. |

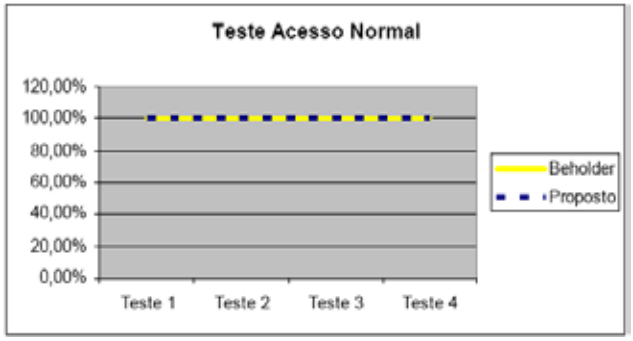


Figura 6. Gráfico de testes normais.

Pode-se observar que tanto pelas informações na Tabela 2 e no gráfico ilustrado na Figura 6, a arquitetura proposta possui o mesmo desempenho que a arquitetura do Beholder.

A seguir será apresentada na Tabela 3 os resultados referentes aos testes de varreduras de ataque ao Servidor 1, feitos com o auxílio do Nmap.

[7] Disponível em: http://nmap.org/man/pt-br/

Tabela 3: Testes de Ataque ao Servidor 1.

| Arquitetura Proposta | Beholder |
|---|---|
| **Teste 1**: máquina cliente efetua scan connect() no Servidor 1. (nmap –sT). | |
| **Resultado**: 2,01% dos 250 pacotes reconhecidos como suspeitos. | **Resultado**: 0,5% dos 250 pacotes reconhecidos como suspeitos. |
| **Teste 2**: máquina cliente efetua scan SYN no Servidor 1. (nmap –sS). | |
| **Resultado**: 99,4% dos 389 pacotes reconhecidos como suspeitos. | **Resultado**: 99,6% dos 389 pacotes reconhecidos como suspeitos. |
| **Teste 3**: máquina cliente efetua scan FIN no Servidor 1. (nmap -sF | |
| **Resultado**: 99,4% dos 570 pacotes reconhecidos como suspeitos. | **Resultado**: 99,6% dos 570 pacotes reconhecidos como suspeitos. |
| **Teste 4**: máquina cliente efetua scan Xmas no Servidor 1. (nmap –sX). | |
| **Resultado**: 99,6% dos 585 pacotes reconhecidos como suspeitos. | **Resultado**: 99,6% dos 570 pacotes reconhecidos como suspeitos. |
| **Teste 5**: máquina cliente efetua scan Null no Servidor 1. (nmap –sN). | |
| **Resultado**: 99,8% dos 527 pacotes reconhecidos como suspeitos. | **Resultado**: 99,6% dos 527 pacotes reconhecidos como suspeitos. |
| **Teste 6**: máquina cliente efetua scan RPC no Servidor 1. (nmap –sR). | |
| **Resultado**: 99,4% dos 350 pacotes reconhecidos como suspeitos. | **Resultado**: 91,3% dos 350 pacotes reconhecidos como suspeitos. |

Tabela 4: Testes de Ataque ao Servidor 2.

| Arquitetura Proposta | Beholder |
|---|---|
| **Teste 1**: máquina cliente efetua scan connect() no Servidor 2. (nmap –sT). | |
| **Resultado**: 6,6% dos 85 pacotes reconhecidos como suspeitos. | **Resultado**: 1,3% dos 85 pacotes reconhecidos como suspeitos. |
| **Teste 2**: máquina cliente efetua scan SYN no Servidor 2. (nmap –sS). | |
| **Resultado**: 98,7% dos 325 pacotes reconhecidos como suspeitos. | **Resultado**: 98,7% dos 325 pacotes reconhecidos como suspeitos. |
| **Teste 3**: máquina cliente efetua scan FIN no Servidor 2. (nmap -sF | |
| **Resultado**: 99,6% dos 280 pacotes reconhecidos como suspeitos. | **Resultado**: 98,6% dos 280 pacotes reconhecidos como suspeitos. |
| **Teste 4**: máquina cliente efetua scan Xmas no Servidor 2. (nmap –sX). | |
| **Resultado**: 99,6% dos 324 pacotes reconhecidos como suspeitos. | **Resultado**: 98,6% dos 324 pacotes reconhecidos como suspeitos. |
| **Teste 5**: máquina cliente efetua scan Null no Servidor 2. (nmap –sN). | |
| **Resultado**: 99,7% dos 1385 pacotes reconhecidos como suspeitos. | **Resultado**: 99,7% dos 1385 pacotes reconhecidos como suspeitos. |
| **Teste 6**: máquina cliente efetua scan RPC no Servidor 2. (nmap –sR). | |
| **Resultado**: 99,4% dos 327 pacotes reconhecidos como suspeitos. | **Resultado**: 91,3% dos 327 pacotes reconhecidos como suspeitos. |

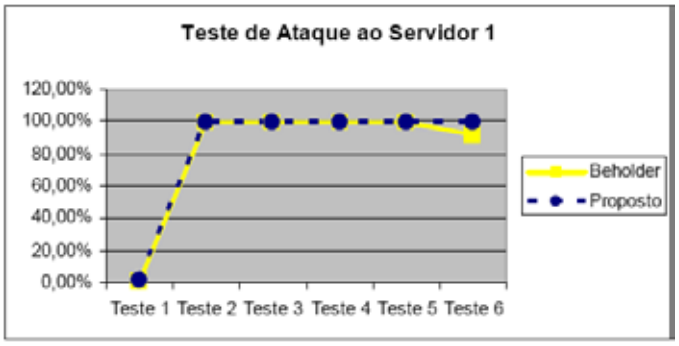


Figura 7. Gráfico de testes de Ataque ao Servidor 1.

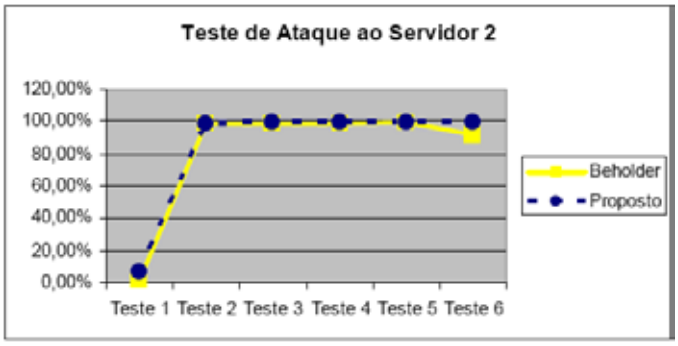


Figura 8. Gráfico de testes de Ataque ao Servidor 2.

Pode-se observar que tanto pelas informações na Tabela 3 e no gráfico ilustrado na Figura 7, a arquitetura proposta possui um desempenho um pouco maior do que o desempenho da arquitetura do Beholder.

A seguir será apresentada na Tabela 4 os resultados referentes aos testes de varreduras de ataque ao Servidor 2, feitos com o auxílio do Nmap.

Pode-se observar que tanto pelas informações na Tabela 4 e no gráfico ilustrado na Figura 8, a arquitetura proposta possui um desempenho um pouco maior do que o desempenho da arquitetura do Beholder no teste efetuado no Servidor 2.

## VII. Considerações Finais

Com a intenção de avaliar a arquitetura proposta do sistema

IDS baseado em Agentes Inteligentes realizaram-se alguns testes, afim de uma comparação com uma ferramenta de IDS já em funcionamento..

Na realização dos testes realizados nos dois servidores e o computador cliente, pode-se observar que ambas as arquiteturas de ferramentas IDS tiveram praticamente o mesmo desempenho nos três tipos de testes realizados.

Porém mesmo sendo tecnologias diferentes, percebe-se que Agentes Inteligentes utilizados na arquitetura proposta como ferramenta IDS podem sensivelmente através de seus resultados apresentados, serem capazes de efetuarem as detecções baseadas em intrusões.

Os testes realizados com a arquitetura sugerida foram apresentados em um ambiente de rede local, e conforme ilustrado na Figura 4, o modelo utilizou dois computadores como servidores e um computador como cliente, representando varreduras de portas (a partir de informações do cabeçalho IP e do cabeçalho de transporte, e quais serviços) sendo operados pelo usuário.

Com base nestes testes ficam alguns questionamentos a serem realizados, se pensarmos num cenário de uma organização, e esta arquitetura estiver em funcionamento numa situação de intrusão, e num processo de perícia poder-se-ia analisar com relação a:

- **Vestígio**: localização de algum rastro produzido em decorrência de fato;
- **Evidências**: tudo o que for encontrado ou localizado, poderá ser aproveitado no processo;
- **Indício**: auxilie nas circunstâncias em que fora realizada toda a tarefa de intrusão;
- **Provas**: tornar um meio lícito para que seja um conteúdo marcante e finalístico para juízes;

E como consideração final, elevando a necessidade de um melhor aprofundamento com relação a desenvolvimento dessa arquitetura, pode-se pensar que as tecnologias existentes de Inteligência Artificial podem representar inúmeros auxílios para a área da computação forense. E por último que este artigo teve como principal contribuição, sua aplicação num ambiente de produção, e principalmente o comparativo com outra técnica embasada em sistemas inteligentes bem como a apresentação de todos os resultados demonstrados numericamente e também através de gráficos.

## REFERENCES

[1] Costa, Marcelo Antonio S.Lemos "Computação Forense" Campinas, Millennium, 2003.

[2] Ianelli Nicholas, Aaron Hackworth, *Botnets as a Vehicle for Online Crime* (e-forensic press). The International Journal of Forensic Computer Science, 2007, pp. 1, 19–39.

[3] Wooldridge, Michael J., An introduction to multiagent systems. West Sussex, England: John Wiley & Sons Ltd., 2002. 348 p.

[4] Cansian, A.M., "Desenvolvimento de um Sistema Adaptativo de Detecção de Intrusos em Redes de Computadores", PhD Thesis, Instituto de Física de São Carlos, USP, São Carlos, SP, 1997.

[5] Naedele, M., and Dzung, D., "Industrial Information System Security – Part I, Part2, and Part 3", ABB Review 2005

[6] J. P. Anderson, "Computer Security Threat Monitoring and Surveillance", James P.Anderson, Co. Fort Washington, PA, 1980.

[7] R. A. Kemmerer and G. Vigna, "Intrusion Detection: A Brief History and Overview",IEEE Computer, Security and Privacy - Supplement, April 2002, pp. 27-29.

[8] Dorothy E. Denning, "An Intrusion-Detection Model", IEEE Trans. on Software Eng.,February 1987.

[9] Costa, Ernesto; S., Anabela**.** Inteligência Artificial: fundamentos e aplicações. Lisboa: FCA Editora de Informática Ltda, 2004. 608 p.

[10] Russel, S.,Norvig P. Artificial intelligence: a modern approach. New Jersey: Prentice-Hall, 1995. 932 p.

[11] Thomas, D.S. and K. Forcht. Legal Methods of using Computer Forensics Techniques for Computer Crime Analysis and Investigation. Issues in Information System 5(2). 2004.

[12] Etzioni, Oren & Weld, Daniel S. Intelligent Agents on the Internet: Fact, Fiction, and Forecast. In: IEEE EXPERT, v.10, n.3, p. 44-49, 1995.

[13] Bombonato, F., Sistemas de detecção de intrusos baseados em redes neurais, Projeto final (Bacharelado em Ciência da Computação), Instituto de Informática, Universidade Católica de Brasília, Brasília, 2003.

[14] Russel, S., Norvig, P. Inteligência Artificial 2.ed. São Paulo: Editora Campus, 2004. 1056p.

[15] Petroni, Benedito Cristiano Aparecido AISI – Um sistema inteligente para melhoria da usabilidade e da interação de uma interface web / Benedito Cristiano Aparecido Petroni. – Campinas: PUC – Campinas, 2006.154 p.

[16] Nogueira, J. H. M. *Mobile Intelligent Agents to Fight Cyber Instrusions* (e-forensic press). The International Journal of Forensic Computer Science, 2006, pp. 1, 28–32.

[17] Muhammad Naeem Ahmed Khan, Chris R. Chatwin and Rupert C.D. Young Extracting Evidence from Filesystem Activity using Bayesian Networks The International Journal of Forensic Computer Science, 2007, pp. 1, 50–63.

**Benedito Cristiano Ap.Petroni** é mestrando (aluno ouvinte) em Engenharia Legal pela Universidade de São Paulo – USP, mestre em Sistemas de Computação pela Pontifícia Universidade Católica de Campinas (2006), Pós Graduado em Design e Aplicações para Internet pela Universidade São Francisco Itatiba (2001), e Graduado em Análise de Sistemas pela Universidade São Francisco - Itatiba (1998). Atualmente é Professor Associado do Centro Paula Souza (FATEC – Jundiaí e Bragança Paulista.) Possui experiência na área de Ciência da Computação, com ênfase em desenvolvimento de sistemas em Instituições de Ensino Superior, plataformas de Ensino à Distância - EAD. Pesquisa a área de Sistemas Inteligentes através da técnica de Inteligência Artificial, (Agentes Inteligentes- AI), no auxílio e melhorias para área de perícia judicial em computação forense.

**Pedro Luís Próspero Sanchez** é engenheiro eletricista, doutor e livre-docente em Engenharia Elétrica pela Escola Politécnica da Universidade de São Paulo. É bacharel em direito pela Faculdade de Direito da Universidade de São Paulo. É professor livre-docente do Departamento de Engenharia de Sistemas Eletrônicos da Escola Politécnica da Universidade de São Paulo, onde lidera a área de ensino e pesquisa em Engenharia Legal, Ciência e Tecnologia Forenses. É coordenador do Grupo de Engenharia Legal, Ciência e Tecnologia Forenses da Universidade de São Paulo. Na Universidade de São Paulo ministra a disciplina "Engenharia Legal" no nível de graduação, e em pós-graduação ministra as disciplinas "Tópicos de Direito Tecnológico", "Metodologia da Prova Pericial", "Fundamentos de Ciência Forense" e "Ciência Forense Aplicada a Sistemas de Informação".