Apresentando os *Honeypots*: Definições, Conceitos Gerais e Implantação

Higor da Costa Oliveira

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação

Orientador: Profº. Eduardo Pagani Julio

Juiz de Fora, MG

Dezembro de 2009

Higor da Costa Oliveira

Honeypots: Definições, Conceitos Gerais e Implantação

Monografia submetida ao corpo docente do Departamento de Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora, como parte integrante dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Profº. Eduardo Pagani Julio

JUIZ DE FORA - MG

DEZEMBRO, 2009

Honeypots: Definições, Conceitos Gerais e Implantação

Higor da Costa Oliveira

Monografia submetida ao corpo docente do Departamento de Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora, como parte integrante dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em _____ de _____ de 2009.

Comissão Examinadora:

_____

Eduardo Pagani Julio
Mestre em Computação, UFF

_____

Marcelo Lobosco
Doutor em Eng. De Sistemas e Computação, UFRJ

_____

Eduardo Barrere
Doutor em Eng. De Sistemas e Computação, UFRJ

JUIZ DE FORA
DEZEMBRO, 2009

Resumo

Este trabalho tem como objetivo inicial apresentar as ameaças virtuais mais conhecidas a fim de facilitar o entendimento dos termos que são mais utilizados em toda a abordagem de *honeypots*. Posteriormente são apresentadas as definições e conceitos de *honeypots*, além de trazer conceitos mais avançados, como níveis de interação. Uma das motivações deste trabalho é mostrar as principais vantagens e desvantagens de se utilizar *honeypots*, e como podem ter importância na segurança dentro de uma organização. O principal foco deste trabalho, no entanto, será exemplificar, com uma implantação simples, como *honeypots* funcionam, mostrando a sua capacidade quanto ferramenta de segurança.

Palavras-chave: *honeypot*, segurança

Abstract

This work aims to present the most known malware in order to facilitate the understanding of the terms that are frequently used throughout the approach of honeypots. Subsequently it will be presented definitions and concepts of honeypots, in addition to bringing more advanced concepts, such as levels of interaction. One of the motivations of this work is to show the main advantages and disadvantages of using honeypots, and how they can be important in security within an organization. The main focus of this work, however, is to illustrate, with a simple implantation, how honeypots work, showing its capacity as security tool.

Keywords: honeypot, security

Lista de Figuras

Sumário

Lista De Abreviaturas

IDS	Intrusion Detection System
TCP	Transmission Control Protocol
DHCP	Dynamic Host Configuration Protocol
HD	Hard Disk
DoS	Denial of Service
HTML	HyperText Markup Language
DDoS	Distributed Denial of Service
DTK	Deception Toolkit
CERT	Computer Emergency Response Team
DMZ	DeMilitarized Zone
UDP	User Datagram Protocol
IP	Internet Protocol
DNS	Domain Name System
HTTP	HyperText Transfer Protocol
SMTP	Simple Mail Transfer Protocol
ICMP	Internet Control Message Protocol
FTP	File Transfer Protocol
POP3	Post Office Protocol
ARP	Address Resolution Protocol
SSH	Secure Shell

1. Introdução

Existem indivíduos que utilizam a *Internet* para fins ilegais, como o acesso de informações não autorizadas, a espionagem industrial, resultando em uma concorrência desleal, destruição de arquivos importantes, entre outros. Estes são alguns dos exemplos de ameaças que enfrentamos no dia-a-dia, principalmente em ambientes de forte concorrência.

Para empresas, essas ameaças significam prejuízo. As invasões de sistemas, os vírus e o roubo de informações são um grande problema enfrentado pelos administradores de rede atualmente. Todas essas ameaças devem ser identificadas e neutralizadas, e o indivíduo que as cometeu deverá sofrer as punições recorrentes da lei de cada local.

Um termo que nasceu para designar esses indivíduos e aparece cada vez mais na mídia em geral, mas usado erroneamente, é o termo *hacker*. Esse indivíduo possui grandes conhecimentos em redes de computadores, sistemas operacionais e programação. O termo correto para designar indivíduos que usam estes conhecimentos para realizar ataques virtuais é *blackhats*[1].

Desde então, uma série de medidas vem sendo tomadas para impedir que estes indivíduos continuassem a proliferar com seus ataques. Um grupo oriundo dos próprios *hackers,* os chamados especialistas de segurança, ou *whitehats,* acabaram criando metodologias, como ferramentas para detecção e captura destes invasores.

Com o avanço dos estudos na área de segurança e com a necessidade de uma atualização crescente das ferramentas, abriu-se um vácuo para a necessidade de entender as técnicas e as filosofias por trás dos atacantes. Para isso, uma nova classe de ferramentas apareceu e entrou de maneira ainda que modesta nas empresas: os *honeypots.* A proposta deste trabalho é realizar o estudo, a apresentação geral e a implantação dessa importante ferramenta de segurança.

[1] Indivíduo com grandes conhecimentos de redes, sistemas operacionais e programação que tem como objetivo invadir sistemas computacionais com intenção maliciosa (PROVOS, 2007).

Para protegermos um sistema computacional que esteja conectado dentro de uma rede de computadores, contamos com o apoio de ferramentas como um *firewall* e/ou um *IDS* (*Intrusion Detection System*). O *firewall* realiza análise do tráfego que entra e sai de uma rede, bloqueando determinado tráfego e isolando a rede interna da *Internet* em si. Já o *IDS* monitora as atividades dos pacotes que entram na rede, identificando ataques através de assinaturas dos mesmos. Portanto, ambos são sistemas que tem como objetivo principal a segurança de uma rede de computadores. Como foi dito, os *honeypots* entram em cena para preencher um objeto de estudo diferente, que é a análise dos ataques e a exploração de vulnerabilidades de um sistema (PROVOS, 2007).

Utilizados inicialmente por pesquisadores como uma forma de atrair *blackhats* a um sistema de rede, para que seus movimentos e comportamentos pudessem ser estudados, os *honeypots* hoje passam a representar um importante papel na segurança de uma empresa. Isso se deve ao fato de os *honeypots* proporcionarem a detecção preventiva de atividades de rede não autorizadas. Com isso a prevenção de ataques e a implantação de segurança dentro de uma empresa se tornam mais viáveis e produtivas (MARCELO e PITANGA, 2003).

Através do estudo e consequentemente do entendimento dos comportamentos, táticas, ferramentas utilizadas no ataque e até mesmo o tipo dos atacantes que o sistema recebeu, os *honeypots* hoje são muito importantes para os profissionais de segurança de Tecnologia de Informação.

Este trabalho está organizado como segue. O Capítulo 2 descreve as ameaças e os tipos de ataques que um sistema pode receber, assim como a terminologia usada por profissionais de segurança. O Capítulo 3 define o conceito de *honeypots*, mostra a história por trás dos mesmos e define os tipos usados em diferentes necessidades. O Capítulo 4 mostra a importância e o papel dos *honeypots* dentro da segurança de uma organização, listando as vantagens e desvantagens de se utilizar este componente de segurança. O Capítulo 5 mostra todos os níveis de interação de *honeypots* e comparativos sobre como

utilizar cada nível de interação, além de trazer exemplos de ferramentas de *honeypots.* O Capítulo 6 detalha a implantação e a utilização de um *honeypot,* mostrando os resultados obtidos.

2. Entendendo as Ameaças

Este capítulo aborda os tipos de ameaças existentes para um sistema computacional, além de exibir estatísticas da comunidade de segurança e o entendimento dos tipos de ataques existentes.

2.1. Os Atacantes

Na série de artigos, *Know Your Enemy* (SPITZNER, 2002), é apresentado os principais personagens que podem ser capturados por um *honeypot*. Abaixo é definido a classe de cada um destes atacantes:

- *script kiddies:* este tipo de atacante é o invasor mais inexperiente, utilizando-se de tutoriais e livros para tentar realizar seu ataque. É o mesmo que tentar repetir uma receita de bolo sob a forma de tutorial, repetindo o ataque contido ali e tentando obter sucesso. Se esse atacante conseguir êxito no seu intento, o mesmo não saberá o que fazer quando estiver dentro do sistema, pois não possui conhecimentos técnicos suficientes. Normalmente esse atacante é barrado pelo *firewall;*
- *lammer:* é o tipo de atacante curioso, pois possui certo grau de conhecimento sobre sistemas operacionais, redes entre outros. Assim como o *script kiddie,* é um repetidor de fórmulas de ataque. Normalmente estes atacantes invadem sites inseguros, deixando uma espécie de assinatura dizendo que ele foi o responsável pelo ataque. Assim, na maioria das vezes ele é pego por deixar rastros no sistema, devido a esse tipo de comportamento arrogante;
- *blackhat:* este é o mais perigoso atacante. O termo *hacker* às vezes é utilizado de maneira errada para designar este atacante. O objetivo do *blackhat* é o conhecimento, ficando oculto em um sistema invadido para utilizá-lo em outros

ataques. A sua vitória é invadir um sistema, obter informação, ativar um *backdoor*[2] e ficar totalmente oculto;

- *hacker:* indivíduo que possui amplos conhecimentos de redes, sistemas operacionais e programação;
- *cracker:* profundo conhecedor de programação (C e *Assembly*), este atacante encontra vulnerabilidades em sistemas, depurando-o através do chamado *exploit*. Este tipo de programa explora a vulnerabilidade de um sistema e permite a invasão do mesmo. O *cracker* quando é bem sucedido na tarefa de programar um *exploit*, libera o mesmo para seus contatos de confiança;
- *carder:* este tipo de *hacker* concentra seus esforços no roubo de senhas de cartão de crédito. No Brasil há quadrilhas especializadas neste tipo de crime, sendo procuradas pela polícia federal;
- *phreaker:* muito raro no Brasil por tratar-se de um indivíduo com conhecimentos em eletrônica, telefonia fixa e celular. Sua especialidade é burlar sistemas telefônicos, clonar celulares e executar operações que permitam ligações gratuitas. Um dos mais famosos *hackers,* Kevin Mitnick, era também um dos mais talentosos *phreakers.*

2.2. As Ameaças

Além dos ataques cometidos por indivíduos, também existem ataques oriundos de programas planejados com um determinado objetivo, como:

- vírus: *softwares* de computador projetados para causar algum dano específico em um sistema invadido. Têm comportamento semelhante ao vírus biológico: multiplicam-se de máquina para máquina, precisam de uma máquina hospedeira,

[2] Falha de segurança que pode existir em um *software* ou sistema operacional, permitindo a invasão do sistema por um *blackhat* (MARCELO e PITANGA, 2003).

esperam o momento certo para o ataque e tentam esconder-se para não serem exterminados;

- *worms:* tipos de vírus mais sofisticados. A diferença está na forma de propagação. Os *worms* espalham-se rapidamente para outras máquinas, seja pela *Internet,* seja por meio de uma rede local. A contaminação ocorre de maneira discreta e o usuário só percebe o problema quando a máquina apresenta alguma anormalidade;
- *trojan:* também chamado de cavalo-de-tróia, essa praga digital permite o acesso remoto ao computador após a infecção. Os *trojans* podem ter outras funcionalidades, como a captura de dados e a execução de instruções maliciosas presentes em *scripts*. Permitem acesso ao computador infectado porque usam portas *TCP* (*Transmission Control Protocol*) e informam ao seu criador a disponibilidade daquela máquina infectada;
- *backdoors:* ponto de entrada deixado numa aplicação que permite acesso ao sistema ou aplicação sem conhecimento da vítima;
- *spyware: software* que se instala em uma máquina e recolhe informações pessoais de um usuário sem o consentimento do mesmo. As informações que o *spyware* recolhe podem ser desde informações relativas a todos os sites que o usuário visitou, até informações importantes, como senhas;
- *rootkits:* tipo de *trojan* que busca se esconder de *softwares* de segurança e do usuário utilizando técnicas avançadas de programação. *Rootkits* escondem a sua presença no sistema, escondendo os seus processos na lista de processos ativos, além de retornar sempre erros de “arquivo inexistente” ao tentar acessar os arquivos do *trojan*;
- usuário: por incrível que pareça, a maior ameaça para qualquer sistema seguro é o próprio usuário. Por descuido, desinformação, ou até mesmo vingança, o

usuário pode comprometer a estrutura de um sistema com atitudes como: senhas simples, passagem de informação, recebimento de *e-mail* com vírus, acesso não autorizado à determinadas aplicações. O usuário é responsável pelo chamado *hacking from inside*, ou seja, a invasão feita de dentro da própria estrutura na qual trabalha.

2.3. O Alvo

Muitas pessoas possuem a ideia de que seus sistemas não serão atacados e que os atacantes nunca as encontrarão dentro da enorme rede que é a *Internet*. Acreditam que pelo fato de seus sistemas não possuírem valor, ninguém terá interesse em realizar o ataque. Ou talvez tenham a crença de que, pelo fato de estar utilizando *DHCP* (*Dynamic Host Configuration Protocol*), ninguém conseguirá encontrá-los. Todas essas ideias são erradas. A pessoa pode achar que o seu sistema executando um sistema operacional como *Windows XP* não tem valor, mas atacantes podem encontrar grande benefício nesse sistema, como usá-lo para atacar outro sistema, ou usar o *HD* (*Hard Disk*) para armazenar informações roubadas (MANOEL, SOARES e DA SILVA, 2004).

As estatísticas[3] a seguir, foram coletadas com o uso de *honeypots* (SPITZNER, 2002):

- no final do ano 2000, a expectativa de vida (tempo sem ataques) de um sistema rodando *Red Hat* 6.2 era de menos de 72 horas;
- o tempo mais rápido que um *honeypot* foi invadido é de 15 minutos. Isso significa que em 15 minutos o sistema foi encontrado, escaneado, atacado e invadido pelo atacante;
- no início de 2002, uma rede simples era escaneada em média por 31 sistemas diferentes por dia.

[3] Estatísticas mais recentes podem ser visualizadas em http://www.honeypots-alliance.org.br/stats/, esses dados são coletados por *honeypots* do Consórcio Brasileiro de *Honeypots*.

- entre o período de 2000 a 2001, houve um aumento de 100% em *scans* com o uso de ferramentas como o *Nmap*[4], e quase 900% de alertas gerados pelo *Snort*[5];

As estatísticas são assustadoras pelo fato dessas redes utilizadas pelos *honeypots* serem redes simples. Nada foi feito para atrair os atacantes. Estes *honeypots* podem ser representados pela maioria dos sistemas reais utilizados atualmente em empresas e corporações.

Infelizmente a situação piora a cada ano. Com o surgimento de ferramentas automatizadas para realizar ataques, qualquer pessoa com o mínimo de conhecimento poderá tentar algum ataque e este ser bem sucedido. Tudo isso nos mostra que todos são alvos.

2.4. Motivos dos Ataques

Reconhecer os motivos dos ataques pode ser uma ajuda considerável em entender as ameaças. Agências de inteligência norte-americanas criaram a sigla *MICE (Money, Ideology, Compromise, Ego)*[6]*.* Essa sigla serve para entender a razão pela qual um atacante tentará realizar um ataque a um sistema. A seguir são listados os motivos de ataques, comprovados por um *honeypot* (SPITZNER, 2002):

- *DoS*: o ataque de negação de serviço serve para tirar sistemas inteiros fora do ar e são feitos direcionando tráfego intenso à vítima (um servidor *Web,* por exemplo). Quanto maior esse tráfego, maior é o ataque. Muitos *blackhats* utilizam *DoS* (*Denial of Service*) para atrapalhar outros *blackhats* em seus ataques;

[4] http://nmap.org/

[5] http://www.snort.org/

[6] Dinheiro, Ideologia, Comprometer (no sentido de expor-se ao perigo), Ego.

- ganho financeiro: a obtenção ilegal de informações financeiras como contidas em cartões de crédito permitem que os *blackhats* roubem dinheiro, obtendo ganho financeiro;
- respeito e aceitação: muitos atacantes se dedicam aos ataques em busca de respeito e aceitação por parte da comunidade *blackhat*;
- espionagem corporativa: corporações que buscam obter vantagem sobre outras com base em roubo de informações confidenciais;
- aprendizado e desafio: alguns poucos atacantes buscam conhecimento e curiosidade sobre o valor do sistema atacado;
- ciclos de CPU: alguns *worms* são projetados para ganhar uso do processador da máquina infectada, possibilitando assim uso do poder computacional alheio.

2.5. Classificação dos Ataques

Esta seção mostra três tipos de ataques comumente utilizados pelos *blackhats*. São abordadas brevemente as ferramentas utilizadas nesses ataques.

2.5.1. Ataques de Exploração

São ataques cuja função é obter o maior número de informações possíveis sobre o alvo. O mais comum são os *scanners, softwares* que varrem redes em busca de vulnerabilidades, e portas de serviços abertas. Um dos *softwares scanners* mais utilizados é o *Nmap*, que possui uma série de recursos, entre os quais a identificação (*fingerprint*) de um sistema operacional (MARCELO e PITANGA, 2003). Na Figura 1 pode-se ver uma tela do *Nmap*.

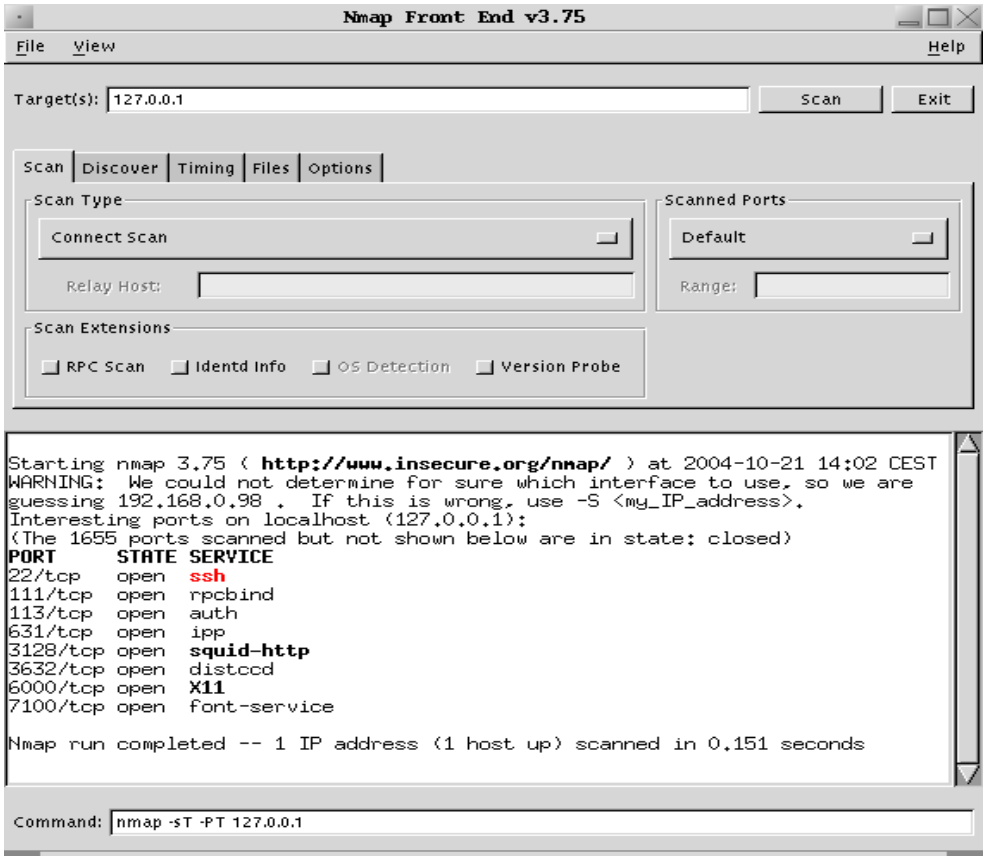


Figura 1 - Ferramenta *Nmap*
Fonte: (Nmap, 2009)

Outro scanner famoso é o *Nessus*. Executa uma série de testes contra um alvo, gerando um relatório *HTML* (*HyperText Markup Language*) sobre as vulnerabilidades encontradas. A Figura 2 apresenta um relatório feito pelo *software.*

2.5.2. Ataques de Paralisação

Este ataque tem como objetivo principal indisponibilizar temporariamente recursos de um sistema. É também conhecido como *DoS* ou negação de serviço (MARCELO e PITANGA, 2003).

Este ataque pode ser iniciado com ferramentas básicas como o *ping* e o *traceroute,* causando uma sobrecarga de conexões aos sistemas alvos, acarretando a negação de serviços. Alvos típicos desses ataques são servidores *Web*, na tentativa de indisponibilizar páginas hospedadas aos usuários.

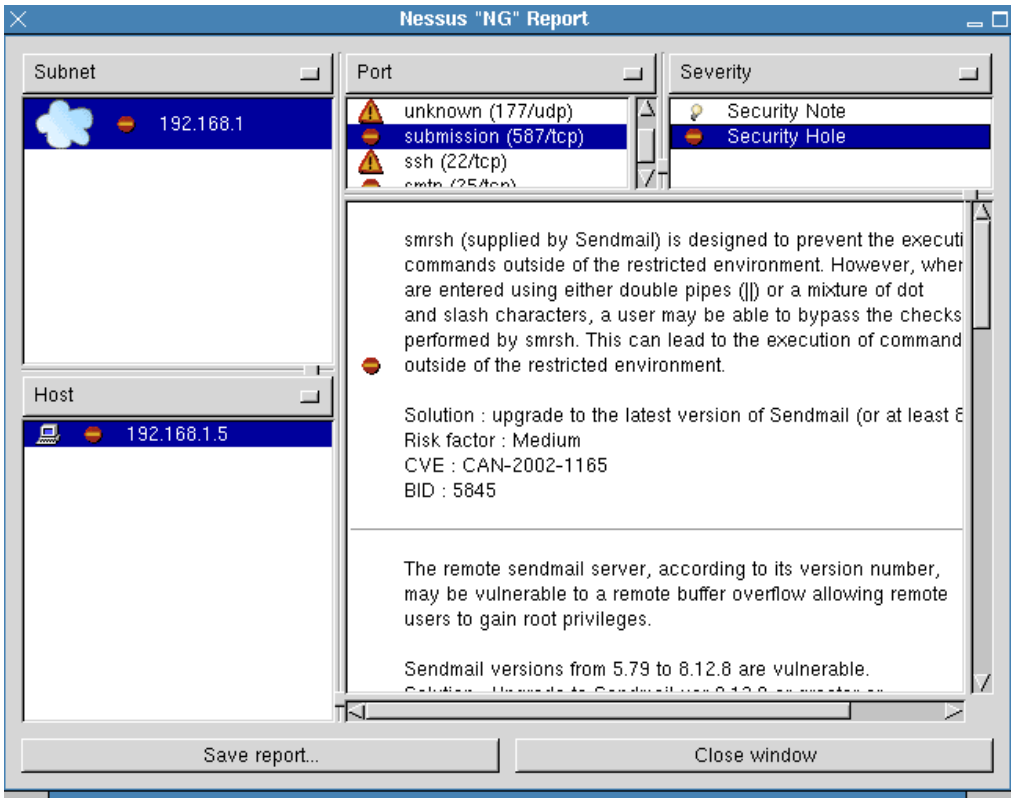


Figura 2 - Relatório Gerado Pelo Nessus
Fonte: (Nessus Report, 2009)

Pode-se definir então que o ataque de negação de serviço não se trata de uma invasão, mas sim na tentativa de invalidar um sistema através de sobrecarga de conexões ao alvo.

Atualmente esse tipo de ataque cedeu lugar a outro tipo de ataque ainda mais potente e destruidor, o *DDoS* (*Distributed Denial of Service*). Neste tipo de negação de serviço, o ataque é efetuado simultaneamente a partir de vários computadores previamente invadidos e apropriados, sem que o usuário tenha conhecimento, tornando o sistema atacado indisponível com maior eficiência. Ataques *DDoS* podem ser iniciados manualmente, ou através de *worms* contendo em seu código instruções de ataques programados. A Figura 3 ilustra esses tipos de ataques (MARCELO e PITANGA, 2003).

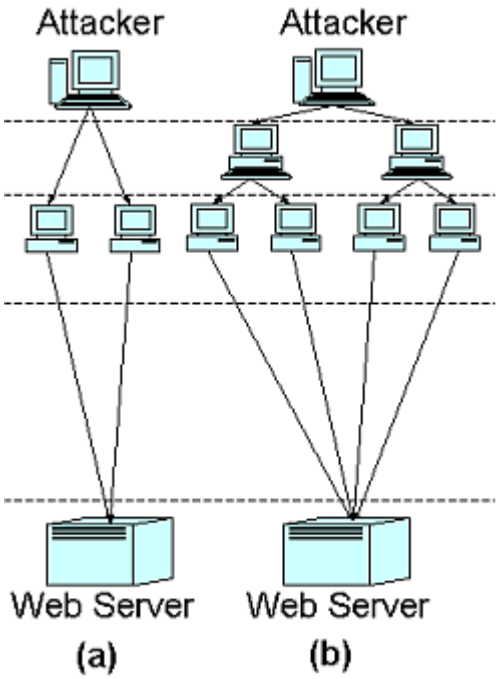


Figura 3 - Ataque de DoS(a) e ataque de DDoS(b)
Fonte: (DoS e DDoS, 2009)

2.5.3. Ataques de Comprometimento

Esse tipo de ataque tem a finalidade de comprometer o funcionamento dos dispositivos de uma rede através da desativação de serviços críticos em servidores, comprometimento ou destruição de informações do alvo, desperdício de recursos ou até mesmo comprometer fisicamente os recursos de um sistema (MARCELO e PITANGA, 2003).

Os ataques de comprometimento podem ser exemplificados por pichações de sites, destruição intencional de dados, suspensão de recursos do sistema como: processamento, memória.

2.6. Conclusão

Como foi discutido, há uma variedade de atacantes com domínio de técnicas de

invasão e comprometimento de um sistema computacional, bem como uma variedade de ferramentas e motivos rentáveis. Para ajudar no estudo destas técnicas e proteger uma organização destas ameaças, a seguir serão apresentados de uma forma abrangente os *honeypots.*

3. Introdução aos *Honeypots*

O termo *honeypot* vem do inglês pote de mel. O mel foi na antiguidade um elemento cobiçado por ser de doçura extrema e ser o alimento de nobres e reis. As pessoas acham o mel uma iguaria desejada e de grande sabor. O conceito de *honeypot* em uma rede é de ser um elemento atraente para o invasor, ou melhor, uma iguaria para o *blackhat* (MARCELO e PITANGA, 2003).

Existem diversas definições para o *honeypot.* Segundo Spitzner (2002), *honeypot* é um recurso de rede cuja função é de ser atacado e comprometido (invadido). Significa dizer que um *honeypot* poderá ser testado, atacado e invadido. Os *honeypots* não fazem nenhum tipo de prevenção, os mesmos fornecem informações adicionais de valor inestimável.

A definição acima quer dizer que um *honeypot* não é uma ferramenta de segurança, e sim de pesquisa de segurança. Outra definição de Marcelo e Pitanga (2003) é que *honeypot* é um sistema que possui falhas de segurança reais ou virtuais, colocadas de maneira proposital, a fim de que seja invadido e de que o fruto desta invasão possa ser estudado.

Honeypot é um atrativo que é colocado em uma rede para que possa ser comprometido e a partir daí sofrer uma análise minuciosa no que diz respeito à filosofia do atacante e das ferramentas que o mesmo utilizou. Obviamente que um *honeypot* pode se tornar, nas mãos de indivíduos despreparados, uma enorme brecha de segurança de qualquer sistema, ou seja, abrir caminho de maneira fácil para o invasor (PROVOS, 2007).

Contudo, os *honeypots* nas mãos de especialistas tornam-se um elemento de valor inestimável para a captura de informações, que dificilmente poderiam ser obtidas de outra maneira. Existem casos de capturas de ferramentas, *worms*, assinaturas de ataques, feitas por *honeypots* que foram revertidas para a comunidade de segurança, possibilitando a criação de novas ferramentas de defesa. (AZEVEDO, 2005)

3.1. História

Especialistas de segurança apontam como o início da história dos *honeypots* o clássico texto *The Cuckoo's Egg* (STOOL, 1990), escrito por um astrônomo do Laboratório *Lawrence*, localizado em *Berkeley,* Estados Unidos. Este texto aponta como, num período de 10 meses de 1986 a 1987, Stool localizou e encurralou em seus sistemas o *blackhat Hunter*. As técnicas utilizadas nessa operação para muitos autores são os precursores dos *honeypots* atuais.

Em outro artigo famoso, *An Evening With Berferd* (CHESWICK, 1991), Bill Cheswick relata como durante meses estudou as técnicas e criou armadilhas para o *blackhat Berferd,* que através de uma falha de segurança no *Sendmail*[7]*,* obtinha as senhas dos usuários do sistema. Este artigo é de um valor muito grande, já que a idéia por trás dos *honeypots* começou a ser desenhada ali, além de ter sido utilizado por muitos especialistas de segurança como metodologia para atrair e capturar invasores.

Em 1997, surgiu o primeiro *honeypot* real, pelas mãos de Fred Cohen, o *DTK* (*Deception Toolkit*). O *DTK* é um conjunto de *scripts* feitos nas linguagens de programação *Perl* e *C* que simulam vários servidores, inclusive com vulnerabilidades. Uma das grandes vantagens dessa ferramenta é possuir o código fonte livre, ainda sendo bastante utilizado nos dias atuais (AZEVEDO, 2005).

Em 1998, o primeiro produto comercial surgiu, o *Sting,* da extinta empresa *Cybercop,* adquirida pela *NAI* no final deste mesmo ano. Este *honeypot* possuía uma série de recursos e era feito não para ambientes *Unix,* e sim para *Windows NT.* O *Sting* simulava uma rede inteira, além de emitir respostas falsas para os atacantes, simulando diversos ambientes operacionais.

[7] Agente de transferência de correio de código aberto.

Ainda em 1998, Marty Roesch, criador do *IDS Snort*, desenvolveu um *honeypot* para o governo americano, este simulava uma rede classe C inteira, com sete diferentes sistemas operacionais simulados e uma série de serviços.

Outra data importante é o ano de 1999, com a criação do *Honeynet Project* por Lance Spitzner e mais 30 especialistas de segurança. Spitzner, um ex-militar e autor de vários artigos famosos, conseguiu desenvolver uma série de metodologias para a implementação de *honeypots*, tornando-se uma referência no assunto.

Em 2001, os *honeypots* foram amplamente utilizados por várias empresas para a captura de *worms*, que começaram a proliferar pela *Internet,* entre eles o *CodeRed II* e o W32/*LeavesWorm.* No ano 2002, os *honeypots* provaram seu valor capturando o primeiro *exploit* desconhecido para o *dtspcd* (um serviço de sistemas *Unix,* teoricamente um ataque remoto a esse serviço poderia ter como consequência o controle e acesso ao sistema invadido), vulnerabilidade que já havia sido reportada pela organização em pesquisa de segurança *CERT* (*Computer Emergency Response Team*) em 2001. Estava provado, portanto, que os *honeypots* poderiam obter informações sobre ataques desconhecidos da comunidade de segurança (MARCELO e PITANGA, 2003).

No mesmo ano de 2002 houve o lançamento do *honeyd*, de Niels Provos, um *honeypot* de código aberto que é amplamente utilizado nos dias atuais e será objeto de estudo deste trabalho.

3.2. Exemplos de *Honeypots*

Para melhor entendimento sobre a definição e conceito de *honeypots*, pode-se listar dois exemplos de implantações de *honeypots*. O propósito é demonstrar que *honeypots* podem ter vários tipos de implantações, e podem atingir objetivos diferentes. Contudo, todas as duas compartilham os mesmos conceitos e definições.

Para o primeiro exemplo, pode-se utilizar um *honeypot* em um sistema que não seja mais utilizado como produção. Este *honeypot* poderá se situar na *DMZ* (*Demilitarized Zone*, Figura 4, *Honeypot* A), e observar de perto todo o tráfego com destino ou origem a este sistema. A intenção é determinar se há alguma atividade não autorizada acontecendo dentro da *DMZ*.

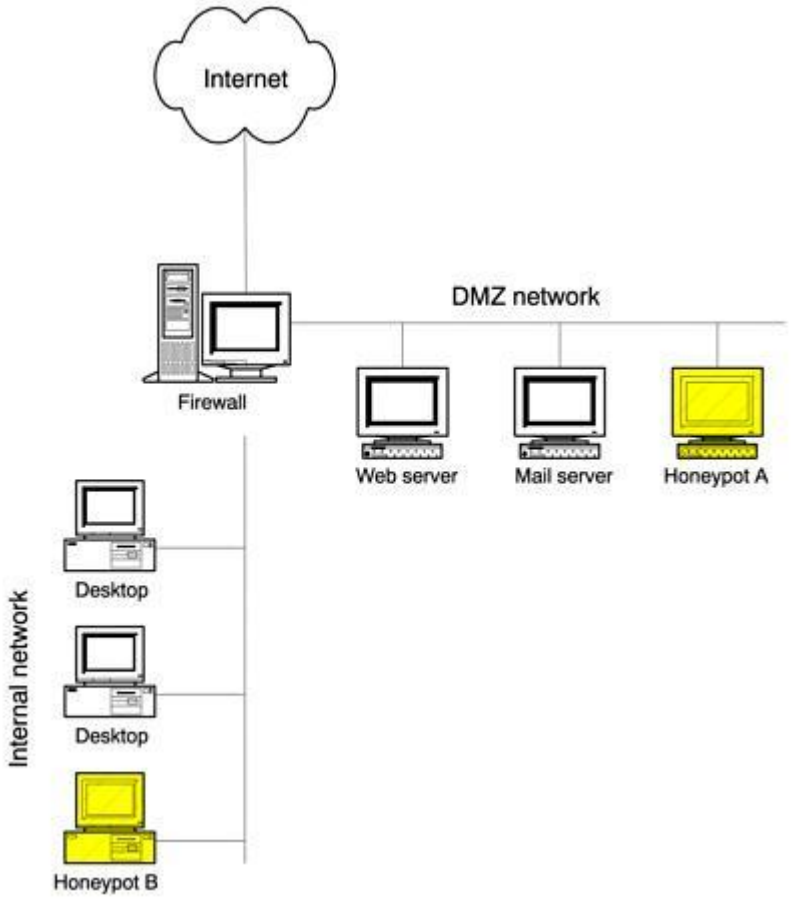


Figura 4 - Dois exemplos de Honeypot
Fonte: (SPITZNER, 2002)

O sistema não está em funcionamento, portanto ninguém deve se conectar a ele. Agora, se alguém da *Internet* se conecta ao *honeypot*, é mais provável que esteja sendo feita uma varredura ou potencialmente, um ataque ao sistema, talvez à procura de servidores vulneráveis. Se algum dos servidores de produção na *DMZ*, como um servidor de *e-mail* ou um servidor *Web*, fizer uma conexão com o *honeypot*, esses servidores podem ter sido comprometidos e, agora, o atacante pode estar rastreando outros sistemas vulneráveis, incluindo a nova *honeypot*. O sistema agora é um *honeypot*, por definição. Quando sondado

ou atacado, o *honeypot* é usado para detectar e alertar para qualquer atividade de rede não autorizada.

Outro exemplo pode ser um sistema projetado e construído desde o início como sendo um *honeypot*. *Worms* e ataques automatizados representam uma séria ameaça, especialmente se eles penetram no *firewall* e acessam a rede interna. Segundo Provos (2007), respostas forjadas e enviadas por um *honeypot* são criadas para manter conexões abertas com o atacante, abrandando ou mesmo parando ataques automatizados. Para este fim, é necessário implantar este exemplo de *honeypot* na rede interna (Figura 4, *Honeypot* B). Este *honeypot* é muito diferente do *honeypot* usado no exemplo que foi descrito anteriormente.

Nestes exemplos, foram vistos dois *honeypots* diferentes construídos de duas maneiras diferentes e para duas finalidades diferentes. Mas ambos compartilham o mesmo valor, sendo sondados, atacados ou comprometidos. E ambos demonstram o conceito de que pelo fato de não possuírem atividades de produção, qualquer coisa enviada até eles é mais provável como sendo uma atividade não autorizada.

3.3. Tipos de *Honeypots*

Os *honeypots* possuem uma categorização de acordo com a finalidade do seu uso. Esta seção apresenta os dois tipos de *honeypots* existentes.

3.3.1. *Honeypots* de Produção

Honeypots de produção agregam valor à segurança de uma organização e ajudam a mitigar o risco. A implantação desses *honeypots* dentro de uma organização é feita porque ajuda a proteger esse ambiente computacional de ataques. As organizações comerciais geralmente usam *honeypots* de produção para minimizar o risco de ataques. *Honeypots* de produção geralmente são mais fáceis de construir e implantar do que *honeypots* de

pesquisa, porque exigem menos funcionalidades. Devido à sua relativa simplicidade, geralmente os riscos também são menores. É muito mais difícil usar um *honeypot* de produção para atacar e prejudicar outros sistemas. No entanto, eles também geram menos informações sobre os ataques e os atacantes do que os *honeypots* de pesquisa. Pode-se aprender sobre os atacantes que os sistemas recebem ou o *exploit* que os mesmos utilizam, mas muito provavelmente não há como saber como se comunicam entre si ou como se desenvolvem as suas ferramentas (SPITZNER, 2002).

3.3.2. *Honeypots* de Pesquisa

Honeypots de pesquisa são destinados a obter informações sobre a comunidade *blackhat*. Estes *honeypots* não agregam valor direto para uma organização específica. Sua principal missão é investigar como as organizações podem enfrentar ameaças, quem são os atacantes, como eles são organizados, que tipo de ferramentas eles usam para atacar outros sistemas, e onde obtiveram essas ferramentas. Esta informação permite a compreensão das ameaças e como funcionam. Armado com este conhecimento, pode-se proteger melhor contra ataques (SPITZNER, 2002).

Honeypots de pesquisa são muito mais envolventes do que *honeypots* de produção. Para saber mais sobre os atacantes, é preciso dar-lhes sistemas operacionais e aplicações reais com as quais interagem. Isso gera muito mais informação do que simplesmente como as aplicações estão sendo sondadas. Pode-se eventualmente saber quem são os agressores, como se comunicam, ou como desenvolvem ou adquirem suas ferramentas. No entanto, este aumento de funcionalidade tem as suas desvantagens. *Honeypots* de pesquisa têm um maior fator de risco, e exigem mais tempo e esforço para administrar. Na verdade, *honeypots* de pesquisa poderiam, potencialmente, reduzir a segurança de uma organização, pois requerem enormes recursos e manutenção (PROVOS, 2007).

A distinção entre *honeypots* de produção e pesquisa não é absoluta, é apenas uma orientação para ajudar a identificar a finalidade de *honeypots*. Muitas vezes, o mesmo

honeypot pode ser uma solução de produção ou de pesquisa. Não depende tanto de como *o honeypot* é construído, mas a forma como é utilizado. Por exemplo, um *honeypot* pode ter capturado toda a atividade de um ataque. Se uma empresa está usando este *honeypot* como sendo de produção, é mais interessante a detecção do ataque, o bloqueio do atacante, e talvez até mesmo mover um processo aos os indivíduos envolvidos. Se a organização está usando o *honeypot* como sendo de pesquisa, é mais interessante o foco nas ferramentas que o atacante está usando, de onde o ataque está vindo, e suas atividades depois de ter comprometido o *honeypot*. É o mesmo *honeypot*, com as mesmas informações capturadas. A diferença está na finalidade do *honeypot*, uma vez que pode ser usado tanto como produção ou pesquisa (SPITZNER, 2002).

3.4. A importância dos *Honeypots*

Esta seção mostrará o valor e importância do uso de *honeypots*. Como mencionado anteriormente, ao contrário dos mecanismos de segurança, tais como *firewalls* e *IDSs*, um *honeypot* não resolve um problema específico. Pelo contrário, é uma ferramenta que contribui para a arquitetura de segurança de uma organização. O valor dos *honeypots* e os problemas que podem resolver dependem de como se constrói, se implanta e se usa os mesmos.

3.4.1. Vantagens dos *Honeypots*

Honeypots podem acrescentar grandes funcionalidades a uma organização, a seguir estão listadas as principais vantagens do uso de *honeypots* (SPITZNER, 2002):

- dados colhidos: um dos desafios que enfrenta a comunidade de segurança é conseguir extrair utilidade nos dados colhidos. Organizações coletam grandes quantidades de dados todos os dias, incluindo *logs* de *firewall*, *alertas* do *IDS* e *logs* gerais do sistema. A enorme quantidade de informação e dados torna

extremamente difícil obter qualquer utilização concreta a partir dos mesmos. *Honeypots*, por outro lado, recolhem muitos poucos dados, mas o que eles recolhem é de alto valor. Em vez de registrar *gigabytes* de dados a cada dia, a maioria dos *honeypots* coleta *megabytes* de dados por dia. Qualquer dado que é registrado é provavelmente uma varredura, uma sondagem, ou ataques ao sistema. Na Figura 5, observa-se uma tentativa de varredura contra uma rede de *honeypots*, onde há uma variedade de conexões *UDP* (*User Datagram Protocol*) feitas a partir de vários sistemas localizados na Alemanha, identificados a partir dos endereços *IP* (*Internet Protocol*);

Date	Time	Src-IP	Dst-IP	Proto	Src-P	Dst-P
02/02/15	23:50:15	213.68.213.135	10.1.1.101	udp	5298	18030
02/02/15	23:50:15	213.68.213.134	10.1.1.109	udp	18986	10903
02/02/15	23:50:15	213.68.213.133	10.1.1.108	udp	16932	16219
02/02/15	23:50:16	213.68.213.140	10.1.1.107	udp	1348	5274
02/02/15	23:50:16	213.68.213.130	10.1.1.105	udp	19841	15316
02/02/15	23:50:17	213.68.213.144	10.1.1.104	udp	17773	3327

Figura 5 - Levantamento das máquinas de uma determinada rede
Fonte: (SPITZNER, 2002)

- recursos: outro desafio que mecanismos de segurança enfrentam é a limitação de recursos, ou o esgotamento de recursos. Esgotamento de recursos é quando um recurso de segurança já não pode continuar a funcionar porque os seus recursos estão sobrecarregados. *Honeypots* normalmente não têm problemas de esgotamento de recursos. O *honeypot* só captura atividades dirigidas a si mesmo, por isso o sistema não é sobrecarregado pelo tráfego. Quando um *IDS* falha por causa do esgotamento de recursos, um *honeypot* não é suscetível a ter esse mesmo problema. Outro benefício de um *honeypot* é que não há necessidade de fazer um grande investimento em *hardware* para a implantação;
- simplicidade: não há algoritmos robustos para se desenvolver, não há bases de dados de assinaturas a se manter em *honeypots*. O único trabalho é instalar e configurar o *honeypot* desejado e esperar a invasão e comprometimento do

mesmo. No entanto, *honeypots* de pesquisa podem ser mais complexos de se implantar;

- retorno do investimento: quando *firewalls* conseguem evitar com sucesso os atacantes, tornam-se vítimas do seu próprio sucesso. A administração pode começar a questionar o retorno sobre investimentos, já que percebem que não há mais ameaças. A razão por nunca ser atacado é que o *firewall* ajudou a reduzir o risco. Os investimentos em outras tecnologias de segurança, como autenticação segura e criptografia, enfrentam o mesmo problema. Em contraste, *honeypots* ao capturar atividade não autorizada, podem ser utilizados para justificar não apenas seu próprio valor, mas os investimentos em recursos de segurança também. Quando a administração percebe que não há ameaças, *honeypots* podem provar que uma grande parcela de risco ainda existe.

3.4.2. Desvantagens dos *Honeypots*

Honeypots podem trazer alguns problemas para uma organização se forem programadas ou implantadas de maneira errada. A seguir estão listadas as principais desvantagens do uso de *honeypots* (SPITZNER, 2002):

- campo de visão limitado: a maior desvantagem dos *honeypots* é que têm um campo de visão limitado. Só identificam a atividade que é dirigida contra eles. Se um atacante invadir a rede, o *honeypot* não detectará a atividade maliciosa, ao menos que o *honeypot* seja atacado diretamente. Se o atacante identificar o *honeypot*, ele pode se infiltrar no sistema, com o *honeypot* sendo impossibilitado de detectar que o mesmo conseguiu a invasão;

- identificação: é quando o invasor pode identificar a verdadeira identidade de um *honeypot,* porque o mesmo tem determinadas características ou comportamentos esperados. Um *honeypot* implementado incorretamente pode gerar identificação.

Por exemplo, um *honeypot* pode ser projetado para emular um servidor *Web IIS*, mas o *honeypot* também tem certas características que o identificam como um servidor *Unix*. Essas identidades contraditórias atuam como uma assinatura do *honeypot*. Em *honeypots* de pesquisa a identificação é ainda mais problemática. O atacante pode fornecer informação errônea de forma proposital ao *honeypot*, levando a comunidade de segurança a tirar conclusões incorretas sobre a comunidade *blackhat*;

- risco: um *honeypot*, uma vez atacado, pode ser usado também para atacar, infiltrar ou prejudicar outros sistemas. *Honeypots* diferentes possuem diferentes níveis de risco. Alguns introduzem muito pouco risco, enquanto outros fornecem ao atacante plataformas completas para lançar novos ataques. Quanto mais simples o *honeypot*, menor o risco. Por exemplo, um *honeypot* que apenas emula alguns serviços é difícil de ser comprometido e usado para atacar outros sistemas. Em contraste, um *honeypot* que cria uma "jaula" para o atacante dá ao mesmo um sistema operacional real para interação. Um invasor pode ser capaz de sair da "jaula", e usar o *honeypot* para lançar ataques passivos ou ativos contra outros sistemas. O risco é variável, dependendo de como se constrói e implementa o *honeypot.*

Por causa de suas desvantagens, *honeypots* não podem substituir outros mecanismos de segurança, como *firewalls* e *IDSs*. Ao contrário, eles agregam valor ao trabalhar com mecanismos de segurança existentes.

3.5. O Papel dos *Honeypots* na Segurança

Agora que foram analisadas as vantagens e desvantagens dos honeypots, pode-se falar sobre a aplicação em segurança. Será apresentado como cada tipo de honeypot agrega valor à segurança e reduz o risco global de uma organização.

3.5.1. O Papel dos *Honeypots* de Produção

Será abordado neste tópico o valor de *honeypots* de produção para uma organização. Para identificar estes valores, pode-se quebrar a análise de segurança em três categorias e, em seguida, rever a forma como *honeypots* podem ou não agregar valor a cada um deles. Especificamente, pode-se dividir a segurança nas seguintes categorias: prevenção, detecção e resposta (SCHNEIER, 2000).

Spitzner (2002) detalha o papel de *honeypots* em cada categoria de segurança:

- prevenção: significa manter os atacantes fora do sistema. A comunidade de segurança utiliza ferramentas para prevenir atividades não autorizadas, como o *firewall. Honeypots* acrescentam pouco valor à prevenção, uma vez que não evitam que o inimigo acesse qualquer tipo de recurso do sistema. Se for implantado incorretamente, um *honeypot* pode introduzir riscos, fornecendo ao invasor uma vulnerabilidade em uma organização. Uma forma de prevenção que *honeypots* podem ser úteis é a dissuasão. Dissuasão pode funcionar para organizações que querem proteger recursos de alto valor. A intenção seria a de confundir os atacantes focados em alvos de escolha, ou seja, alvos programados para serem invadidos manualmente, e não com a ajuda de *rootkits* e *worms*. Esses atacantes selecionam cuidadosamente os seus objetivos e analisam as informações que recebem a partir deles. Nesse caso, *honeypots* poderiam ser usados para enganar ou intimidar o atacante, fornecendo informações erradas propositalmente;
- detecção: é ato de detectar e alertar atividades não autorizadas. A prevenção pode falhar, e o atacante entrará no sistema. A prevenção pode mitigar o risco, mas nunca eliminá-lo. *Honeypots* adicionam grande valor à detecção. Para muitas organizações, a detecção é extremamente difícil. Três desafios comuns de detecção são falsos positivos, falsos negativos, e agregação de dados. Falso

positivo é quando um *IDS* falsamente detecta atividades suspeitas ou maliciosas. Falso negativo é quando um *IDS* não consegue detectar um ataque. Agregação de dados é centralizar e recolher todos os dados utilizados para a detecção e transformar os dados em informações valiosas. *Honeypots* possuem uma resposta eficaz aos três desafios de detecção. *Honeypots* não tem tráfego de produção, motivo pelo qual há pouca ou nenhuma atividade para gerar falsos positivos. Também lidam bem com falsos negativos, porque não são facilmente contornados ou comprometidos por *exploits*. Na verdade, uma de suas principais vantagens é que podem detectar um novo ataque em virtude da atividade dirigida ao sistema, não pelas assinaturas dos mesmos. Por fim, *honeypots* resolvem o desafio de agregação de dados através da criação de muito pouca quantidade de dados. Isto torna extremamente fácil diagnosticar a informação útil de *honeypots*. Como exemplo um *honeypot* pode ser implantado dentro da *DMZ* (Figura 6) para ajudar a detectar ataques dentro de uma organização. Qualquer conexão da *Internet* para o *honeypot* indica que alguém está fazendo uma varredura nos servidores da *DMZ*. Se alguém se conectar à porta 25 do *honeypot*, por exemplo, isso indica que alguém provavelmente está explorando vulnerabilidades do servidor de *email*. Ainda mais revelador seria se o servidor de correio tivesse conexões iniciadas com o *honeypot*. Se o *honeypot* detectasse qualquer atividade desse sistema para si mesmo, isso indicaria que foi comprometido e estava sendo utilizado como *bots*[8] para invadir outros sistemas vulneráveis;

- resposta: assim que se detecta um ataque bem sucedido, deve-se ter a capacidade de responder. O desafio que as organizações enfrentam ao reagir a um incidente é a coleta de provas. Isto é essencial não apenas se uma organização quer mover um processo contra um atacante dentro dos limites da

[8] *Bots* podem ser utilizados para a coordenação e a operação de um ataque automatizado em um sistema computacional.

lei, mas também quando se trata de defesa contra um ataque. Uma vez comprometidas, as organizações devem determinar se o atacante invadiu outros

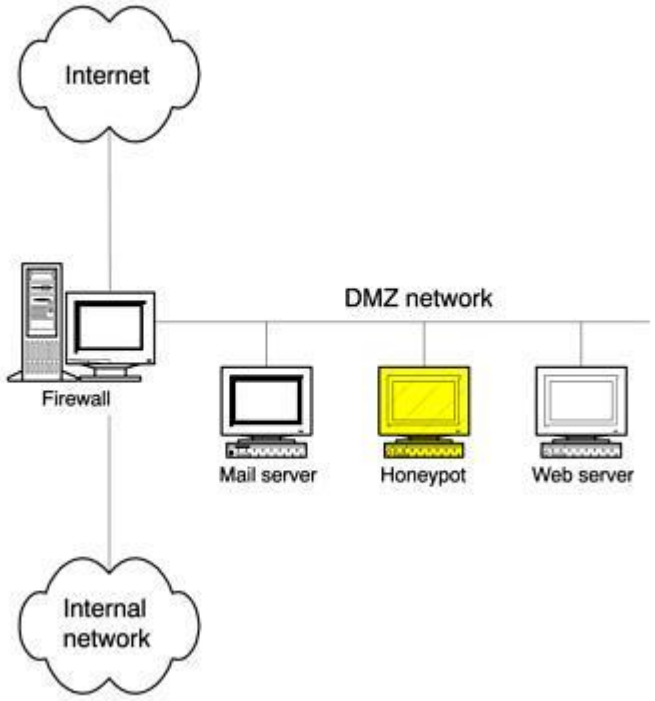


Figura 6 - *Honeypot* na *DMZ*
Fonte: (SPITZNER, 2002)

sistemas, deixou *backdoors*, entre outros. No entanto, essas provas podem ser adulteradas ou danificadas. Recuperar provas danificadas de um sistema comprometido é um dos maiores desafios enfrentados pelas equipes de resposta a incidentes. Um segundo desafio que organizações enfrentam após um incidente é que os sistemas comprometidos não podem ser deixados *offline*. *Honeypots* podem ajudar a enfrentar estes desafios de resposta. Como foi dito, *honeypots* não têm atividade de produção, assim já há uma solução para o problema da adulteração de dados. *Honeypots* também podem facilmente serem colocados *offline* para posterior análise. Como exemplo de valor à resposta a incidentes, considere uma organização com vários servidores *Web*. Tem-se novamente uma *DMZ*, todos os servidores *Web* escutando na porta 80, *HTTP* (*Hypertext Transfer Protocol*, Figura 7). No entanto, apenas os três servidores

Web têm entradas no *DNS* (*Domain Name System*), assim sendo são os únicos que irão receber um pedido de páginas *web*. Desde que o *honeypot* não esteja cadastrado no *DNS*, ele não terá qualquer pedido de páginas *Web*. O *honeypot*, no entanto, está executando as mesmas aplicações que os servidores. O *honeypot* da Figura 7 encontra-se no meio da rede. Se um atacante verifica cada servidor na *DMZ* por qualquer vulnerabilidade nos servidores *Web*, a varredura provavelmente também atingirá o *honeypot*. Se um dos servidores *Web* é atacado com sucesso, o *honeypot* também pode ser atacado. Se vários sistemas estão comprometidos, o atacante provavelmente tem usado as mesmas ferramentas e métodos em todos os sistemas. As organizações podem então se concentrar no *honeypot,* recolhendo dados para análise e, em seguida, aplicar as lições aprendidas nos outros sistemas comprometidos.

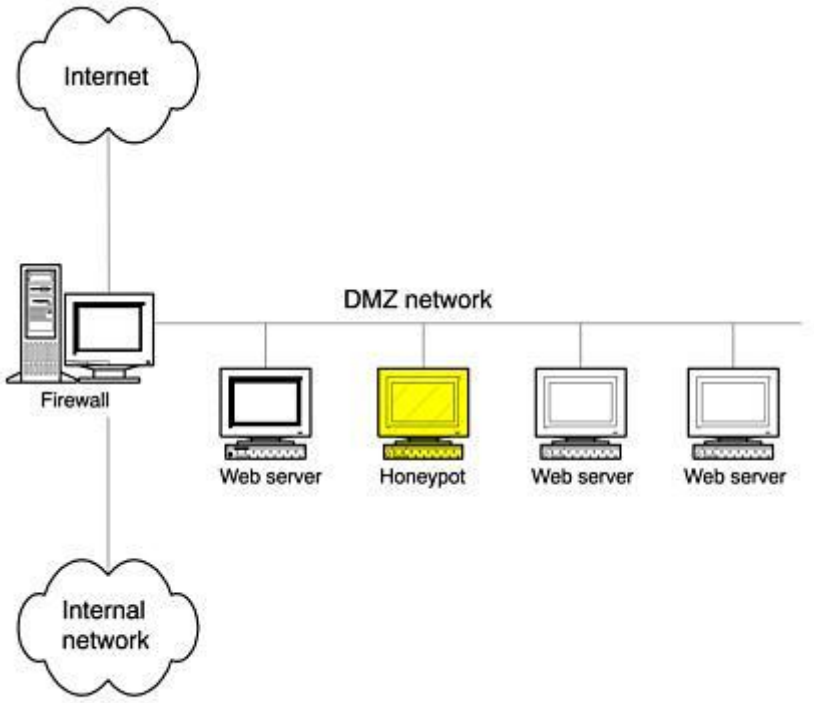


Figura 7 - Honeypot entre servidores Web
Fonte: (SPITZNER, 2002)

3.5.2. O Papel dos *Honeypots* de Pesquisa

Um dos maiores desafios que comunidades de segurança enfrentam é a falta de informação sobre o inimigo. Perguntas como quem é a ameaça, por que eles atacam, como atacam, quais são as suas ferramentas, e quando eles vão atacar de novo muitas vezes não podem ser respondidas. A comunidade de inteligência e contra-inteligência gasta bilhões de dólares recolhendo informação porque o conhecimento é algo difícil de ser conseguido nessa área. Para se defender contra uma ameaça, primeiro tem que ser informado sobre a mesma. No entanto, no mundo da segurança da informação, tem-se pouca informação sobre ameaças (SPITZNER, 2002).

O problema é a origem dos dados. Tradicionalmente, os profissionais de segurança têm aprendido sobre *blackhats* estudando as ferramentas que os mesmos usam. Quando um sistema é comprometido, administradores de segurança muitas vezes encontram as ferramentas que o invasor deixou no sistema atacado. Uma variedade de hipóteses é feita sobre os atacantes com base nestas ferramentas capturadas. Em vez de aprender apenas sobre as ferramentas que os atacantes usam, o ideal é identificar esses adversários virtuais, determinar o quão bem eles são organizados, e determinar os seus métodos (MARCELO e PITANGA, 2003).

O valor de tal informação é crucial, e oferece uma variedade de usos. *Honeypots* de pesquisa podem ser usados para (SPITZNER, 2002):

- capturar ameaças automatizadas, tais como *worms* ou *rootkits*. Capturando essas ferramentas e analisando sua carga maliciosa, as organizações podem melhor reagir e neutralizar a ameaça;
- como um mecanismo de alerta rápido, prevendo quando ataques futuros irão acontecer. Isso funciona através da implantação de múltiplos *honeypots* em diferentes locais e organizações. Os dados coletados a partir destes *honeypots* de pesquisa podem então ser utilizados para a modelagem estatística, prevendo

futuros ataques. Os ataques podem então ser identificados e impedidos antes que eles aconteçam;

- capturar ferramentas desconhecidas ou técnicas, como ficou demonstrado com o ataque *dtspcd*;
- entender melhor os motivos e a organização dos atacantes. Ao capturar sua atividade depois de invadir um sistema, tais como as comunicações entre si, é possível entender melhor quem é a ameaça e é como eles operam;
- obter informações sobre *blackhats* avançados.

Em geral, os *honeypots* de pesquisa não reduzem o risco para uma organização, mas as informações aprendidas podem ser aplicadas como forma de melhorar a prevenção, detecção e reação. No entanto, *honeypots* de pesquisa pouco contribuem para a segurança direta de uma organização. Se uma organização está tentando melhorar a segurança do seu ambiente de produção, ela pode querer considerar *honeypots* de produção, pois eles são fáceis de implantar e manter. Se organizações tais como universidades, governos ou grandes empresas estão interessadas em aprender mais sobre as ameaças, então *honeypots* de pesquisa são o mais indicado. O *Honeynet Project*[9] é um exemplo de uma organização utilizando *honeypots* de pesquisa para obter mais informações sobre a comunidade *blackhat*.

3.6. Conclusão

Honeypots são flexíveis e podem ser aplicados a uma variedade de situações. Têm vantagens específicas de acordo com o tipo utilizado. Honeypots têm vantagens como a coleta de dados de alto valor e através da detecção de atividades não autorizadas. No entanto, honeypots também possuem desvantagens, como o risco se forem implementados incorretamente. Dentro das três áreas de segurança (prevenção, detecção e resposta), o

[9] http://www.honeynet.org

valor principal de honeypots de produção é de detecção. Honeypots de pesquisa não ajudam a mitigar o risco, mas são usados para obter informações sobre as ameaças.

No capítulo seguinte são apresentados os diferentes níveis de interação que um *honeypot* pode atingir, traçando uma comparação entre estes níveis e mostrando exemplos de *honeypots* dentro de cada nível.

4. Classificando *Honeypots* Pelo Nível de Interação

O nível de interação fornece uma escala para medir e comparar *honeypots*. Quanto mais um *honeypot* pode fornecer recursos, mais um atacante pode fornecer informações para o *honeypot*. No entanto, pela mesma razão, o dano potencial ao sistema se o *honeypot* for mal configurado também é maior (SPITZNER, 2002).

A seguir são exploradas detalhadamente as características de cada nível de interação em separado e mostrado um comparativo entre os níveis de interação de um *honeypot*.

4.1. *Honeypots* de Baixa Interação

Honeypots de baixa interação normalmente são os mais fáceis de instalar, configurar, implantar e manter por causa de seu *design* simples e funcionalidade básica. Normalmente essas tecnologias simplesmente imitam uma variedade de serviços. O atacante está limitado a interagir com estes serviços pré-designados (MARCELO e PITANGA, 2003).

Um *honeypot* de baixa interação pode emular um servidor padrão *Unix* com diversos serviços em execução, como *telnet* e *FTP*. Um invasor poderia acessar o *honeypot* usando *telnet*, visualizar o *banner* que indica o sistema operacional, e talvez obter um *prompt* de *login*. O atacante pode então tentar fazer o *login* por ataque de força bruta. O *honeypot* capturaria e coletaria essas tentativas, mas não existe um sistema operacional real para o atacante. A interação do atacante é limitada a tentativas de *login*, conforme mostrado na Figura 8 (SPITZNER, 2002).

A Figura 8 mostra um exemplo de *honeypot* de baixa interação, especificamente o *BackOfficer Friendly*. Nesta janela vemos o *honeypot* detectando uma conexão do sistema com o *IP* 192.168.1.100. O atacante faz uma suposta conexão com a porta *HTTP*, tenta

fazer uma conexão *FTP* (*File Transfer Protocol*), e depois tentou um início de sessão *Telnet* com a conta de *root* e senha *r00t*. O atacante está limitado a quanto pode interagir com o *honeypot* por esses serviços emulados. O serviço permite que os atacantes apenas se conectem à porta e executem um único comando. Em seguida, o atacante será desconectado. O *FTP* só registra tentativas de conexões. Não existe serviço para o atacante realizar *login*. O serviço *Telnet* permite que os atacantes forneçam um *login* e uma senha, mas é somente isso. Não há nenhuma outra interação com estes três serviços. O que é mostrado na Figura 8 é a extensão dos serviços emulados e o grau de informação que pode ser obtido a partir deles (SPITZNER, 2002).

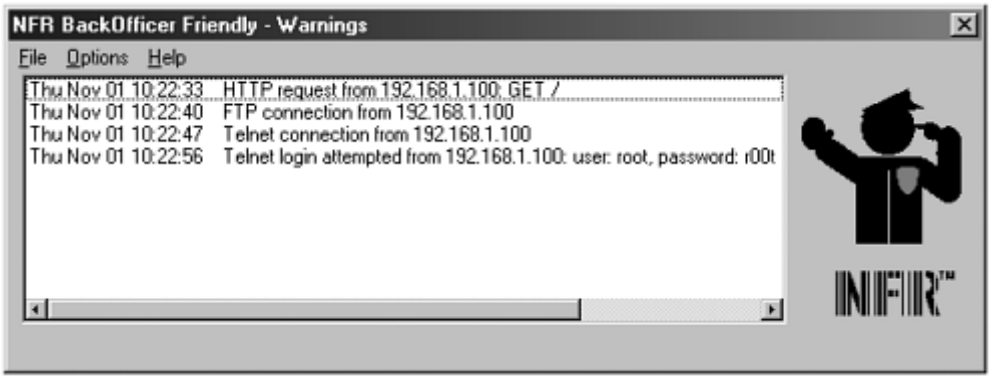


Figura 8 - Exemplo de *Honeypot* de Baixa interação
Fonte: (SPITZNER, 2002)

O principal valor de *honeypots* de baixa interação é a detecção, especificamente de varreduras ou tentativas de conexão não autorizadas. Pelo fato de oferecerem funcionalidades limitadas, a maioria destas funcionalidades pode ser emulada por um programa. O programa é simplesmente instalado em um sistema e configurado para oferecer os serviços que a administração deseja, e o *honeypot* está pronto. Isso faz com que a implantação e manutenção do *honeypot* sejam fáceis. Tudo o que administrador tem que fazer é manter o programa atualizado e monitorar quaisquer alertas gerados (SPITZNER, 2002).

No entanto, este nível de interação faz com que a quantidade de informação capturada sobre um atacante seja pouca. Soluções de baixa interação são limitadas às

informações transacionais e, possivelmente, algumas das interações do atacante com os serviços emulados. Informações transacionais são os dados recolhidos sobre as circunstâncias do ataque, mas não sobre o ataque em si (MARINHO, 2004).

Honeypots de baixa interação capturam os seguintes tipos de informação: hora e data do ataque; endereço *IP* de origem e porta de origem do ataque; endereço *IP* de destino e porta de destino do ataque. Se os serviços emulados permitem qualquer interação com o atacante, eles também são capazes de obter a atividade do invasor (MARINHO, 2004).

4.2. *Honeypots* de Média Interação

Honeypots de média interação oferecem mais capacidades de o atacante interagir com o sistema do que *honeypots* de baixa interação, mas menos funcionalidades que as soluções de alta interação. Podem esperar determinadas atividades e dar as respostas certas, muito além do que um *honeypot* de baixa interação daria (SPITZNER, 2002).

Honeypots de média interação são geralmente mais difíceis de instalar e configurar do que *honeypots* de baixa interação. Muitas vezes, essas soluções não são produtos comerciais feitos sob medida para atender a uma necessidade. Em vez disso, envolvem um alto nível de desenvolvimento e customização. Além disso, a maior funcionalidade, como emular um tipo específico de servidor *web*, requer mais modificações. É preciso maior quantidade de trabalho para construir um *honeypot* que emula um servidor *web IIS* da *Microsoft* e todas as suas funcionalidades do que um *honeypot* de baixo nível, que simplesmente realiza escutas na porta 80 ou porta 443 (SPITZNER, 2002).

Honeypots de média interação também têm uma maior complexidade, tanto na implantação como na manutenção e acaba aumentando o risco de algo dar errado. No entanto, podem reunir uma quantidade muito maior de informações. Ao contrário de simples varreduras de portas, pode-se capturar cargas de *worms* ou atividades do atacante, saber o que acontece depois que os mesmos ganham acesso ao sistema, e até mesmo capturar as

suas ferramentas. Este maior nível de interação vem com mais trabalho e maior risco, mas a recompensa vem com uma grande quantidade de informações (SPITZNER, 2002).

4.3. *Honeypots* de Alta Interação

*S*ão o extremo de tecnologias de *honeypot*. Fornecem uma vasta quantidade de informações sobre os atacantes, mas são extremamente complexos para construir e manter, e vêm com o maior nível de risco. O objetivo de um *honeypot* de alta interação é dar ao invasor o acesso a um sistema operacional real, onde nada é emulado ou restrito. As oportunidades de aprender com os atacantes nesse nível de interação são vastas. É possível descobrir novas ferramentas, identificar novas vulnerabilidades em sistemas operacionais ou aplicativos, e aprender como *blackhats* se comunicam entre eles. As possibilidades são quase ilimitadas, fazendo com que *honeypots* de alta interação seja uma arma extremamente poderosa (SPITZNER, 2002).

A Figura 9 mostra um exemplo de um *honeypot* de alta interação, mostrando um alto grau de coleta de informações. Neste caso, um atacante comprometeu um *honeypot* de alta interação com *Windows NT*, parte de uma *Honeynet*[10]. Uma vez que este é um *honeypot* de alta interação, o atacante teve acesso a um completo sistema operacional para trabalhar. O atacante comprometeu o *honeypot* para usar o serviço de *FTP*, e usava para o ataque especificamente um sistema *Windows 95*. O atacante tenta enviar ferramentas para o *honeypot* que acaba de *hackear*. Observa-se que o *honeypot* de alta interação pode capturar uma quantidade muito maior de informações em relação ao *honeypot* de baixa interação na Figura 8. Das teclas capturadas do invasor na Figura 9, percebe-se que o atacante realizou *login* em um sistema remoto e fez o *upload* pelo *FTP* das ferramentas nc.exe, pdump.exe e samdump.dll, comumente usados como utilitários de *cracking* (PROVOS, 2007).

[10] Múltiplos *Honeypots* formando um grande sistema computacional (MARCELO e PITANGA, 2003)

```
220-Serv-U FTP-Server v2.5h for WinSock ready...
220--------H-A-C-K T-H-E P-L-A-N-E-T--------
220-W3|_c0m3 T0 JohnA's 0d4y Ef-Tee-Pee S3rv3r.
220-Featuring 100% elite hax0r warez!@$#@
220-Im running win 95 (Release candidate 1),.
220 -------H-A-C-K  T-H-E  P-L-A-N-E-T--------
USER johna2k
331 User name okay, need password.
PASS haxedj00
230 User logged in, proceed.
PORT 172,16,1,106,12,71
200 PORT Command successful.
RETR nc.exe
150 Opening ASCII mode data connection for nc.exe (59392 bytes).
226 Transfer complete.
PORT 172,16,1,106,12,72
200 PORT Command successful.
RETR pdump.exe
150 Opening ASCII mode data connection for pdump.exe
(32768 bytes).
226 Transfer complete.
PORT 172,16,1,106,12,73
200 PORT Command successful.
RETR samdump.dll
150 Opening ASCII mode data connection for samdump.dll
(36864 bytes).
226 Transfer complete.
QUIT
221 Buh bye, you secksi hax0r j00 :]
```

Figura 9 - Exemplo de Honeypot de Alta Interação
Fonte: (PROVOS, 2007)

Devido ao seu risco, *honeypots* de alta interação são colocados dentro de um ambiente controlado, geralmente atrás de um *firewall*. A capacidade de controlar o atacante não vem do *honeypot* em si, mas a partir do dispositivo de controle de acesso à rede, no caso o *firewall*. O *firewall* permite que o invasor comprometa os *honeypots* situados dentro da rede interna, mas não deixa que o invasor use o *honeypot* para lançar ataques para fora da rede. Essa arquitetura é muito complexa para implantar e manter, especialmente se a intenção não seja que o atacante perceba que está sendo monitorado e controlado. Uma grande parte do trabalho vai para a construção de um *firewall* com regras adequadas (PROVOS, 2007).

4.4. Comparativo Entre os Níveis de Interação

A utilidade de um *honeypot* de baixa interação ou de alta interação depende do que se deseja alcançar. A Tabela 1 resume o comparativo entre os diferentes níveis de interação em quatro categorias. Essas categorias podem ajudar a determinar qual o nível de interação é o melhor para uma determinada organização.

A primeira categoria é a instalação e configuração, que define o tempo e esforço em instalar e configurar o *honeypot*. Em geral, quanto maior o nível de interação tem um *honeypot*, mais o trabalho necessário para instalar e configurar. Quanto mais funcionalidades se fornecem a um atacante, mais opções e serviços devem ser instalados e configurados.

A segunda categoria é a implantação e manutenção. Esta categoria define o tempo e o esforço envolvidos na implantação e manutenção do *honeypot* depois de ter configurado o

Tabela 1 - Comparativo entre níveis de interação
Fonte: (SPITZNER, 2002)

Nível de Interação	Instalação e Configuração	Manutenção e implantação	Captura de Informação	Nível de Risco
Baixa	Fácil	Fácil	Limitado	Baixo
Média	Médio	Médio	Variado	Médio
Alta	Difícil	Díficil	Extensivo	Alto

sistema. Mais uma vez, quanto maior o número de funcionalidades do *honeypot*, maior o trabalho necessário para implantar e manter.

A terceira categoria é a captura de informação, o quanto de dados o *honeypot* pode capturar de atacantes e suas atividades. Como mostram Figuras 8 e 9, *honeypots* de alta

interação podem coletar grandes quantidades de informação, enquanto que *honeypots* de baixa interação são muito limitados neste quesito.

Finalmente, os impactos sobre a quantidade de risco apresentada. A preocupação é com o risco de um *honeypot* ser usado para atacar, prejudicar ou infiltrar em outros sistemas ou organizações. Quanto maior o nível de interação, maior a funcionalidade fornecida para o atacante, e maior a complexidade. Combinados, esses elementos podem apresentar um grande risco. Por outro lado, *honeypots* de baixa interação são muito simples e oferecem pouca interação com os atacantes, criando uma solução de risco muito baixo. Nas seções seguintes, serão detalhados exemplos de *honeypots* de baixa, média e alta interação, e comparar suas vantagens e desvantagens.

4.5. Visão Geral de Seis *Honeypots*

O objetivo dessa seção é examinar brevemente seis *honeypots* específicos, todos com diferentes níveis de interação. O objetivo é fornecer uma seleção de *honeypots* e uma compreensão do valor que acrescentam e como funcionam. Estes seis foram selecionados porque representam o amplo espectro de *honeypots* diferentes e seus usos. Incluem-se as soluções comerciais, os *honeypots* feitos para atender um determinado uso (caseiros ou *homemade*), e as soluções *Open Source*. Essas soluções representam tanto *honeypots* de produção como *honeypots* de pesquisa e são encontradas em várias plataformas e com diferentes aplicações.

4.5.1. *BackOfficer Friendly*

BackOfficer Friendly, ou *BOF*, é uma solução simples de *honeypot* livre desenvolvido por Marcus Ranum e pela empresa *Network Flight Recorder*. *BOF* é um *honeypot* de baixa interação concebido para ser executado em praticamente qualquer sistema *Windows*. É uma excelente escolha para começar o uso de *honeypots*, uma vez que qualquer pessoa pode

instalá-lo em um sistema. A manutenção é extremamente simples de fazer, além de fácil configuração. No entanto, essa simplicidade tem um custo: as suas capacidades são severamente limitadas. Possui um pequeno conjunto de serviços que simplesmente escutam em portas específicas, como *FTP* e *SMTP* (*Simple Mail Transfer Protocol*). Uma tela do *BOF* pode ser vista na Figura 10 (BACKOFFICER FRIENDLY, 2009).

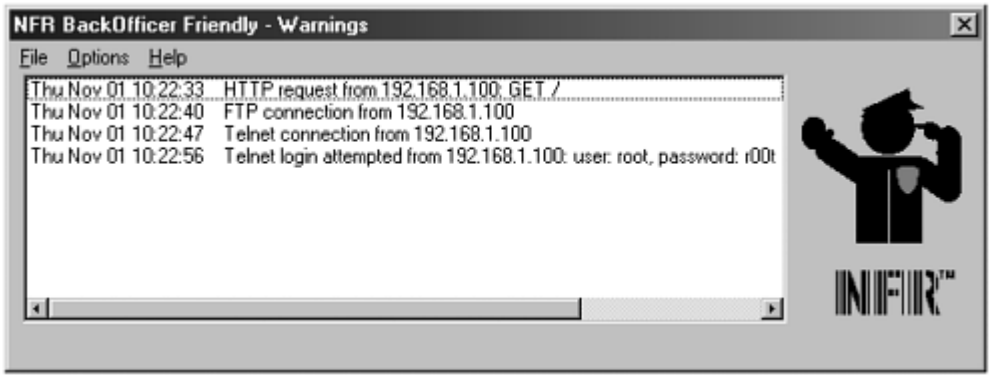


Figura 10 - Tela do BOF
Fonte: (SPITZNER, 2002)

4.5.2. *Specter*

Specter é um *honeypot* baseado em um *IDS*, é comercial com suporte e desenvolvimento pelo *Netsec*. Como o *BOF*, o *Specter* é um *honeypot* de baixa interação. No entanto, *Specter* possui funcionalidades maiores que o *BOF*. Simula um sistema vulnerável, proporcionando um alvo interessante para atrair *blackhats*. *Specter* oferece serviços de Internet comuns, tais como *SMTP*, *FTP*, *POP3* (*Post Office Protocol*), *HTTP* e *telnet* que parecem perfeitamente normais para os atacantes, mas na verdade são armadilhas para os mesmos explorarem as vulnerabilidades e deixarem vestígios. Estes vestígios são registrados e é notificado ao usuário do *honeypot*. Além disso, *Specter* automaticamente “investiga” os atacantes, enquanto ainda estão tentando invadir o sistema.

Specter fornece grandes quantidades de conteúdo de chamariz, incluindo imagens, arquivos *MP3*, mensagens de e-mail, arquivos de senhas, documentos e todos os tipos de *software*. Ele gera dinamicamente um programa chamariz que deixa “marcas” escondidas na máquina do atacante. Atualizações automáticas *on-line* do conteúdo do sistema ajudam o

banco de dados de vulnerabilidades do *honeypot* mudar constantemente sem interação do usuário. *Specter* é um *honeypot* de produção. Na Figura 11 é mostrada uma tela de exemplo do *Specter* (SPECTER, 2009).

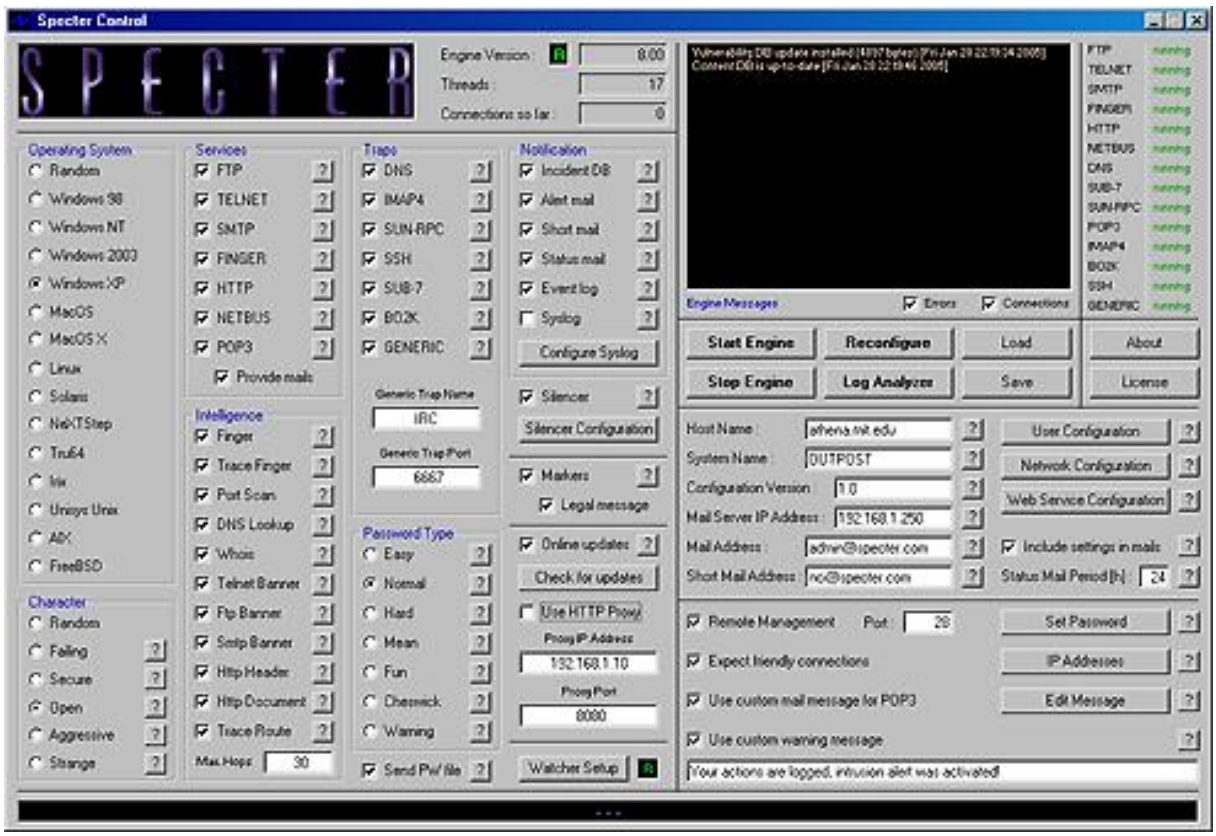


Figura 11 - Tela Do *Specter*
Fonte: (SPECTER, 2009)

4.5.3. *Honeyd*

O *honeypot OpenSource honeyd* é de baixa interação. Seu objetivo principal é detectar, capturar e alertar atividades suspeitas. Desenvolvido por Niels Provos, em Abril de 2002, o *honeyd* introduz vários novos conceitos para *honeypots*. Primeiro, ele não monitora um único endereço *IP* para atividades, em vez disso ele monitora redes com milhões de sistemas. Quando se detecta varreduras contra um sistema que não existe, ele dinamicamente assume a identidade da vítima e, em seguida, interage com o atacante, aumentando exponencialmente a capacidade de *honeypots* para detecção e captura de ataques. É possível emular centenas de sistemas operacionais, aplicações e a pilha de

protocolos *IP*. Como uma solução *OpenSource*, o *honeyd* é uma tecnologia livre, dando acesso total ao código fonte. Pode-se personalizar soluções próprias ou utilizar soluções desenvolvidas por outros membros da comunidade de segurança. Projetado para a plataforma *Unix*, o *honeyd* é relativamente fácil de instalar e configurar, baseando-se principalmente em uma interface de linha de comando (PROVOS, 2007).

4.5.4. Caseiros (*Homemade*)

Honeypots caseiros (*homemade*) são criados por indivíduos ou organizações para atender uma necessidade específica. Uma vez que não há dois iguais, é possível ter *honeypots* de baixa ou de alta interação. No entanto, em geral, *honeypots* caseiros tendem a ser de média interação. Podem ser projetados como produção ou pesquisa, dependendo de como são construídos, implantados e utilizados (SPITZNER, 2002).

4.5.5. Decoy Server

Decoy Server, antigamente chamado de *Mantrap*, é um *honeypot* comercial da *Symantec*. É um *honeypot* de alta interação. O *Decoy Server* é diferente porque simplesmente não emula serviços. Em vez disso, hospeda um sistema operacional inteiro e cria até quatro sistemas operacionais virtuais. Isto dá ao administrador o controle extensivo e capacidade de captura de dados sobre os sistemas operacionais virtuais. As empresas podem ainda instalar aplicativos de produção que se deseja testar, tais como *DNS*, servidores *web*, ou até mesmo um banco de dados. Esses sistemas operacionais virtuais têm quase exatamente a mesma interação e funcionalidade de sistemas de produção padrão. Assim, muita coisa pode ser aprendida com o atacante (DECOY SERVER, 2009).

Uma vez que é um produto comercial, o *Decoy Server* é relativamente fácil de implantar e manter. Também pode capturar uma quantidade incrível de informações. Não só detecta varreduras e conexões não autorizadas, mas pode capturar ataques desconhecidos

e até novas vulnerabilidades. No entanto, a sua versatilidade vem à custa de maior risco. Pelo fato de o *honeypot* ter um sistema operacional completo para o atacante poder explorar, o mesmo pode ser usado para atacar outros sistemas e executar atividades não autorizadas. O *Decoy Server* tem a flexibilidade de ser usado tanto como *honeypot* de produção como de pesquisa, embora seja mais usado para fins de produção (SPITZNER, 2002).

4.5.6. *Honeynets*

Honeynets representam o extremo de *honeypots* de alta interação. Não se limitam somente a fornecer ao atacante um sistema operacional completo para atacar e interagir, mas pode fornecer múltiplos *honeypots*. *Honeynets* nada mais são do que uma variedade de sistemas padrão implantado dentro de uma rede altamente controlada. Por sua natureza, estes sistemas se tornam *honeypots*, já que sua importância está em serem sondados, atacados ou comprometidos. A rede controlada captura toda a atividade que acontece dentro da *honeynet* e diminui o risco contendo a atividade do invasor (MARCELO e PITANGA, 2003).

A complexidade de uma *honeynet* não está na construção dos *honeypots* em si (que podem facilmente serem implantadas através de instalações padrão), mas sim na construção da rede que controla e capta todas as atividades que estão acontecendo para e através dos *honeypots*. Como tal, *honeynets* são alguns dos *honeypots* mais difíceis de implantar e manter. Essa complexidade também os torna uma das soluções de alto-risco. Sua vantagem reside no fato de que podem capturar o maior nível de informação sobre praticamente qualquer plataforma existente. *Honeynets* são utilizados principalmente para pesquisa. Devido à quantidade de trabalho envolvido, têm pouco valor como *honeypots* de produção (SPITZNER, 2002).

4.6. Conclusão

Este capítulo classificou *honeypots* baseados níveis de interação. Níveis de interação definem a complexidade, o risco e as funcionalidades que um *honeypot* pode oferecer. Quanto maior o nível de interação, maior a complexidade, o risco e a funcionalidade. Além disso, foram exemplificados seis *honeypots* comumente usadas pela comunidade de segurança. O capítulo a seguir mostra a implantação de um *honeypot* de baixa interação, o *honeyd*.

5. Implantação de um *honeypot*

Este capítulo tem como objetivo mostrar a implantação de um *honeypot* de baixa interação, a ferramenta *open source honeyd*. O capítulo está organizado em instalação, configuração, execução, interação e resultados obtidos.

5.1. Instalação do *Honeyd*

O *honeyd* é uma ferramenta designada para sistemas *Unix*, e sua instalação é relativamente simples, dependendo da distribuição a ser utilizada. Para esta implantação foi utilizada a distribuição do *Linux Debian Lenny* 5.0, e a instalação do *honeyd* foi feita da seguinte forma:

- *download* das bibliotecas necessárias para instalar o *honeyd*: *libevent*, *libdnet*, *libpcap*;
- utilizar o gerenciador de pacotes do *Debian apt* para instalar o *honeyd.* A instalação por esse método automaticamente instala outras ferramentas necessárias para se configurar e utilizar o *honeypot*, o *arpd* e o *honeydsum.* Esta instalação é feita pelo comando: ***apt-get install honeyd****;*

5.2. Configuração do *Honeyd*

Com a ferramenta instalada, o próximo passo é definir como será a estrutura do *honeypot*. Para isso é necessário conhecer as funções dos principais arquivos que acompanham o *honeyd*.

No diretório */etc/honeypot/* é possível visualizar todos os arquivos de configuração da ferramenta, mas os principais são os arquivos *honeyd.conf* e o *nmap.prints*. O primeiro arquivo contém todas as informações de configuração do *honeypot* e o segundo contém uma base de dados utilizada pela ferramenta de *scanner nmap* para identificar os sistemas

operacionais emulados pelo *honeyd.* No diretório */usr/share/honeyd* estão todos os *scripts* utilizados para emular os serviços que cada máquina vai possuir, tais como *telnet*, *SMTP e FTP*.

Para a implantação desse *honeypot* foi utilizado o esquema de máquina virtual para executar um distribuição *Linux* em cima de um sistema operacional hospedeiro *Windows*. A máquina hospedeira executava a versão *Windows XP Professional SP3,* enquanto que a máquina virtual executava a distribuição *Debian Lenny 5.0* (Figura 12).

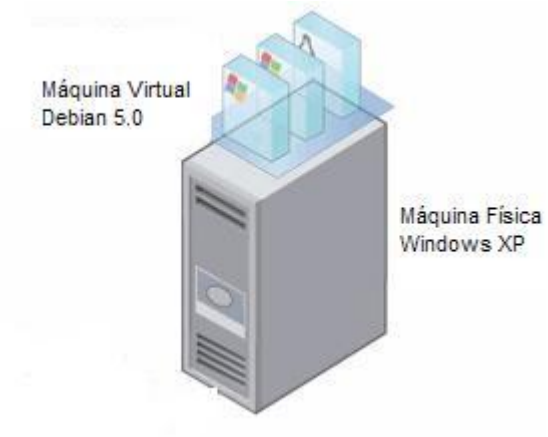


Figura 12 - Esquema de Máquina Virtual
Fonte: (PEOPLEWARE, 2009)

O primeiro passo é estruturar a rede que se deseja configurar. Esta rede pode ser feita utilizando a capacidade da ferramenta de emular uma rede completa, com roteadores, gateways e sub-redes. Esta rede foi projetada como sendo convencional, com um roteador funcionando como *gateway* para toda a rede interna, e cada sub-rede com seu próprio gateway e máquinas. Foi separado máquinas *Linux* e máquinas *Windows* em cada sub-rede a fim de facilitar a interação com o *honeypot*. A Figura 13 mostra a topologia da rede simulada pelo *honeyd*, assim como os respectivos *hosts.*

Estruturada a rede, o passo seguinte é configurar o *honeypot* através do arquivo *honeyd.conf.* No apêndice A é mostrado o arquivo de configuração montado para a *honeypot*.

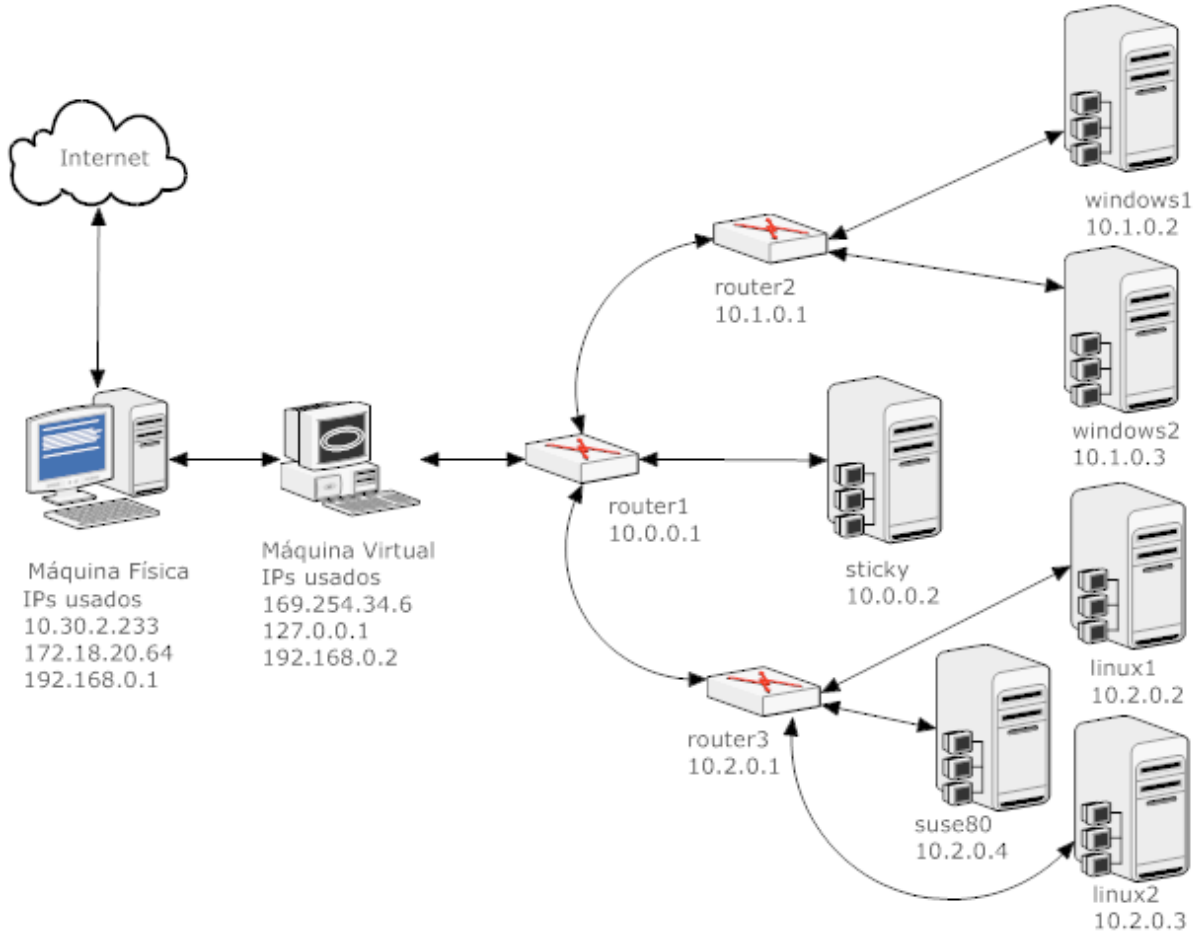


Figura 13 - Topologia da Rede Emulada

A seguir é exibida a explicação da sintaxe utilizada pelo *honeyd* para a criação dos *honeypots*:

- ***create windows1***: define a criação do *host* windows1;
- ***set windows1 personality "Microsoft Windows XP Professional SP1"*:** define a personalidade a ser usada para esse *host;*
- ***set windows1 default tcp action reset & set windows1default udp action reset & set windows1 default icmp action open***: definem a ação padrão das portas *TCP*, *UDP* e *ICMP*. Este parâmetro é essencial para enganar o *nmap;*
- ***add windows1 tcp port 80 "perl /usr/share/honeyd/scripts/win32/ iis-0.95/iisemul8.pl"*:** aqui é definido que o *honeyd* deverá simular o servidor *Web* IIS na porta 80.

- ***set windows1 uptime 3284460***: define qual vai ser o tempo que a máquina ficará ligada;

- ***bind 10.1.0.2 windows1***: relaciona o *host* criado ao *IP* desejado.

Além das configurações do *honeypot*, também é necessário fazer a configuração da rede virtual em que as máquinas estarão localizadas:

- ***route entry 10.0.0.1 network 10.0.0.0/8***: adiciona uma entrada na rede virtual 10.0.0.8/8, essa entrada será por meio do IP 10.0.0.1, que é um roteador simulado pelo *honeyd*;

- ***route 10.0.0.1 link 10.0.0.0/24***: adiciona uma rota do roteador 10.0.0.1 para a rede 10.0.0.0/24;

- ***route 10.0.0.1 add net 10.1.0.0/24 10.1.0.1***: adiciona uma rota do roteador 10.0.0.1 para a rede 10.1.0.0/24 através do roteador que funciona como *gateway* 10.1.0.1.

Com isso termina-se a configuração do *honeypot* em questão.

5.3. Execução do *Honeyd*

O próximo passo da implantação é a execução dos *honeypot* criado. Para isso é necessário executar comandos para o correto funcionamento do *honeyd*. Os seguintes passos são necessários:

- criar um roteamento para a rede dos *honeypots* e redirecioná-la para a interface de rede utilizada no *Linux,* com o comando ***route -n add -net 10.0.0.0/8 eth0***, onde 10.0.0.0/8 é a rede dos *honeypots* e eth0 é a interface de rede. Se a implantação for feita sem rede, é só trocar a opção eth0 pra lo (*loopback*);

- antes de inicializar o *honeyd* e começar a interagir, é preciso que o tráfego na rede seja redirecionado para o *honeypot*. Para isso, deve-se usar a ferramenta

arpd para o *honeypot* responder pelos endereços desocupados da rede. Para utilizar o *arpd*, só é necessário especificar a rede que ele vai ouvir. Quando ouvir alguma requisição *ARP* (*Address Resolution Protocol*) passando pela rede, checa se alguém tem o *IP*, se ninguém tiver, responderá como se fosse o *IP* requisitado. Os *hosts* virtuais que têm *IP* definido através do comando *bind*, responderão normalmente e o *host default* que não tem um *IP* definido, responderá por todos os endereços que não são utilizados na rede;

- iniciar o *honeyd* através do comando: ***honeyd -d -x xprobe2.conf -p nmap.prints -f honeyd.conf -l /var/log/ honeypot/honeyd -0 pf.os -i eth0 10.0.0.1-10.2.0.4***, onde –l é o diretório onde os *logs* gerados pelo *honeypot* serão armazenados. Os parâmetros -0, -i, -d são opcionais.

O *honeyd* é executado em *foreground* após esses passos, impedindo o usuário de utilizar aquela instância do console e digitar comandos. Isso é feito para ser possível visualizar o que acontece no *honeyd* quando alguma interação é feita. Para usar outros comandos, deve-se abrir outra instância do console.

5.4. Interação com o *Honeyd*

Para compreender o comportamento dos *honeypots* criados, é necessário algum nível de interação com os mesmos, a fim de comprovar a fidelidade dos sistemas emulados e com isso poder ludibriar os atacantes e colher informações relevantes para análise.

Foram utilizados para a interação com os sistemas a ferramenta de *scanner Nmap*, além de comandos nativos ao *Linux* para realizar conexões e acessos aos *hosts*. O *Nmap* foi executado tanto na máquina hospedeira quanto na máquina virtual, e os comandos foram executados na máquina virtual.

Para testar se o *host* da *honeypot* está ativo, foi dado o comando *ping* no roteador router2. A Figura 14 ilustra o resultado.

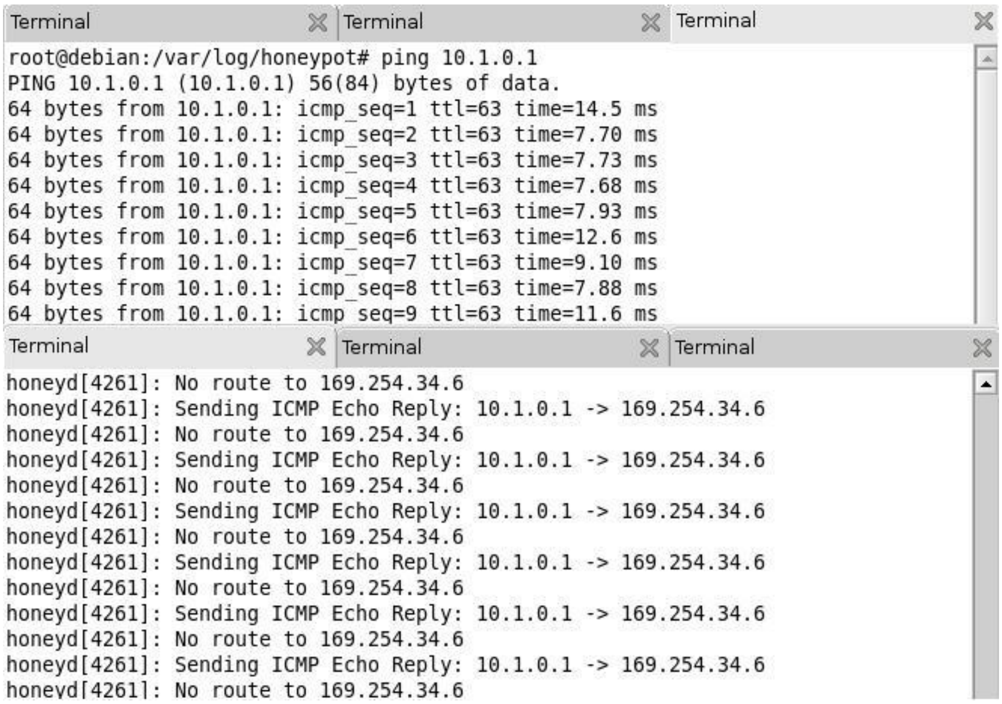


Figura 14 - Resultado do Comando *ping*

Como segundo exemplo, é utilizado o *wget*, comando que faz o *download* de páginas *Web* através de requisições *HTTP.* O comando foi direcionado ao *host* linux1*.* A figura 15 ilustra os resultados.

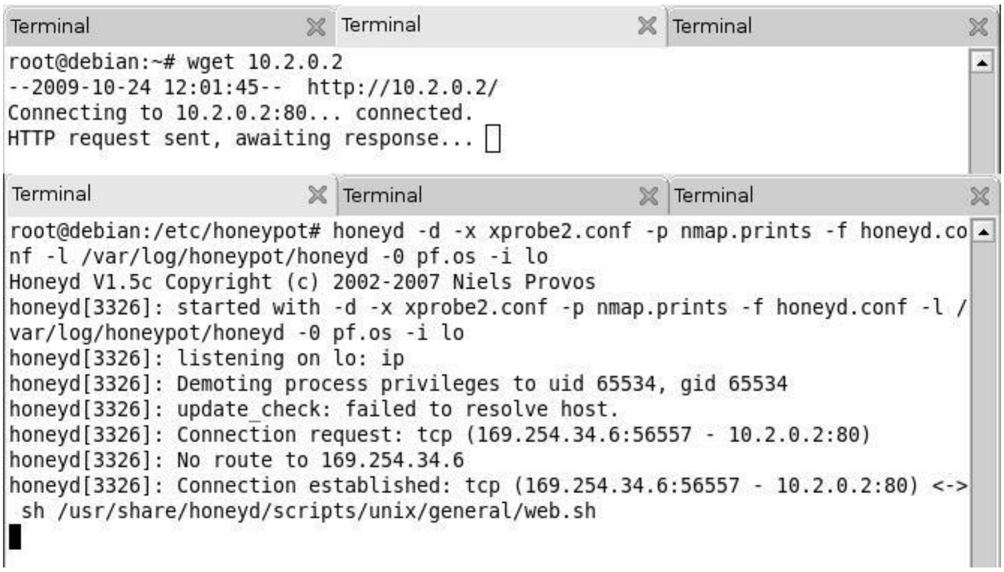


Figura 15 - Resultado do comando *wget*

Como se pode observar, realmente o *host* configurado se comporta como um servidor *web*, respondendo à requisição *HTTP*, e o *honeyd* registra a tentativa de conexão.

A seguir foi feita uma tentativa de conexão *SSH* (*Secure Shell*) com o usuário *root* no *host* linux2. A Figura 16 ilustra o resultado.

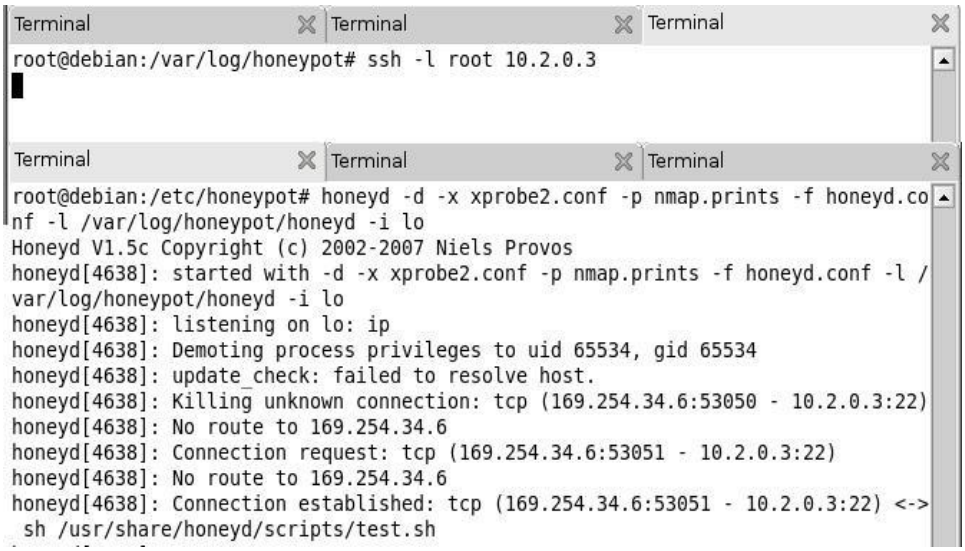


Figura 16 - Resultado do Comando *ssh*

O *honeyd* registra a conexão no seu *log*, mas ao usuário não é fornecido nenhum *prompt* para deixar a interação com o *SSH* mais real.

É importante ressaltar que para o atacante explorar qualquer serviço que o sistema possa oferecer, deve-se primeiro realizar uma varredura com o *Nmap* no sistema alvo para identificar todos os serviços e portas abertas. Foram feitas dezenas de varreduras em todos os *hosts* da *honeypot* a fim de comprovar a fidelidade dos sistemas emulados. Como há um volume muito grande de dados, só são mostrados alguns resultados gerados pelo *Nmap.*

O sistema colocado como exemplo é o *host* suse80. A Figura 17 ilustra o resultado da varredura realizada com o *Nmap*.

Como se pode observar pela figura, o *Nmap* achou e reconheceu praticamente todos os serviços emulados pelo *host suse80*. Os serviços que foram bem reconhecidos, inclusive com o nome do serviço e sua versão, foram *FTP*, *SSH*, *telnet, SMTP, Finger, Web, Pop3, Squid* (na porta 3128).

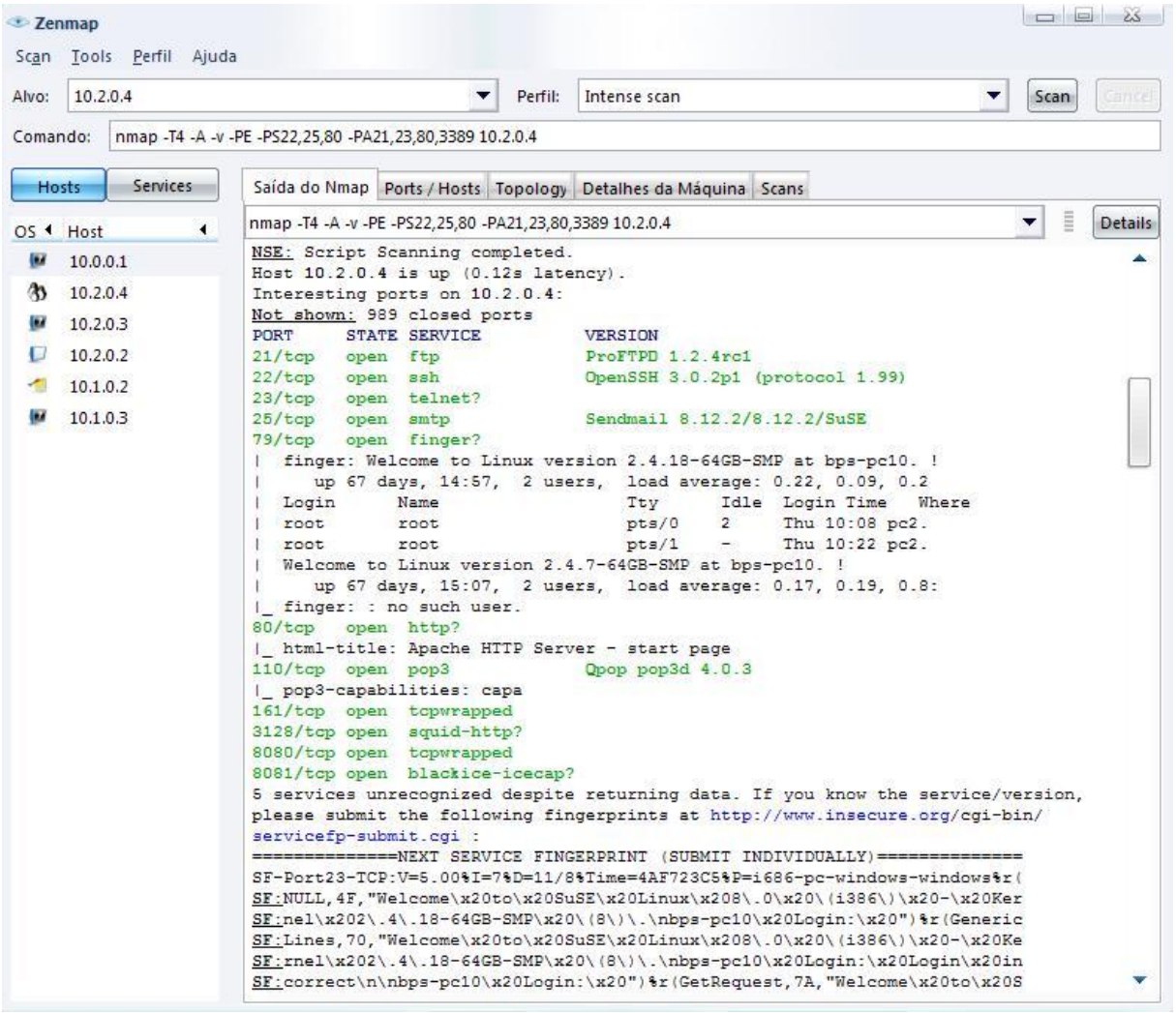


Figura 17 - Resultado do Nmap para suse80

O serviço de *syslog*, uma aplicação que serve para monitorar os *logs* de sistemas *Unix*, não foi identificado por causa de seu protocolo ser *UDP*. As outras duas portas do *Squid* (8080 e 8081) foram retratadas como serviços desconhecidos. Relativo ao sistema operacional, o *Nmap* identificou tratar-se de uma máquina rodando Linux, mas não foi possível especificar a versão do *kernel*, exibindo diversas opções de *kernels* disponíveis.

O problema do *Nmap* não reconhecer com eficácia o sistema operacional nativo ao *host* se deve ao fato de o arquivo *nmap.prints* do *honeyd* ser diferente ou desatualizado em relação ao arquivo de *fingerprints* do próprio *Nmap* (MARCELO e PITANGA, 2003).

Foi feita uma tentativa de atualizar o arquivo *nmap.prints*, mas sem sucesso. Ao executar o *honeyd* com a nova versão do arquivo, a ferramenta não conseguia iniciar com o mesmo e retornava diversas mensagens de erro de *fingerprint* não reconhecido.

Há um *honeypot* chamado *Labrea Tarpit*[11], uma solução que busca atrasar ou atrapalhar ataques automatizados (*autorooters*) e *worms* (SPITZNER, 2002). O *honeyd* oferece a opção de se emular um *honeypot* com essa característica através de um *host*. O *host* utilizado na configuração no *honeyd* para isso é o *sticky*. O *Nmap* foi usado para comprovar como isso pode ser feito. A Figura 18 ilustra essa varredura.

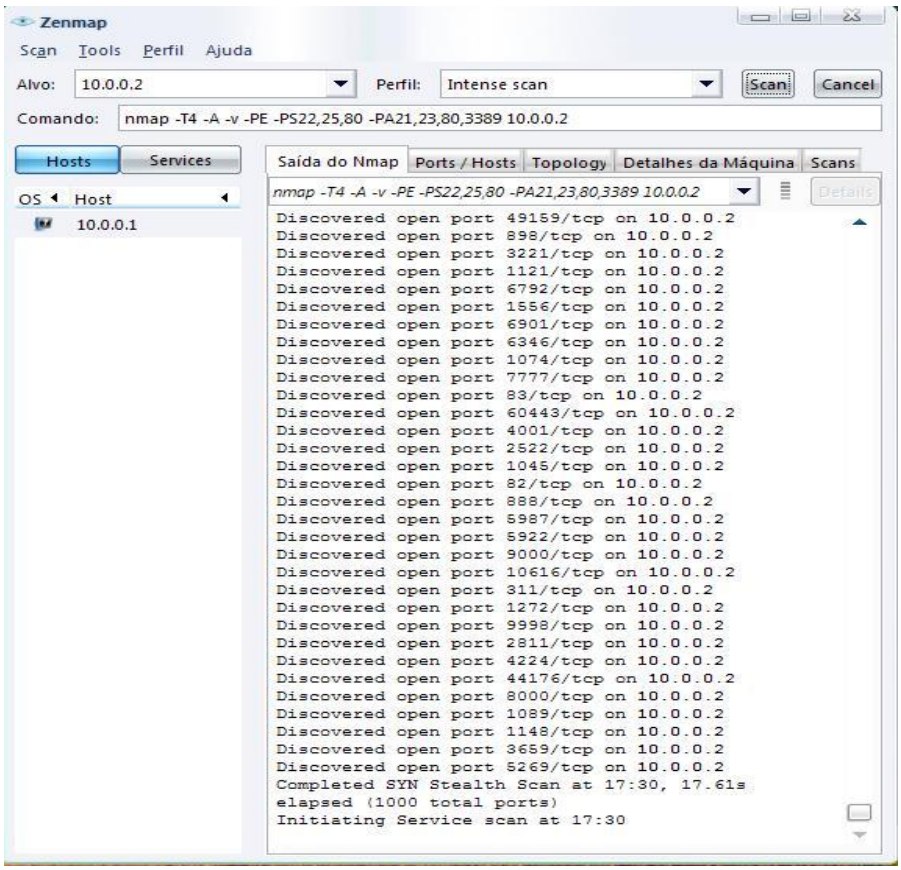


Figura 18 - Resultado do *Nmap* para sticky

Como se pode observar, o *Nmap* acha milhares de portas abertas, e chega até a travar o *software* em alguns momentos, gerando um tempo estimado de varredura de 4 horas e meia. Apesar de *honeypots* terem pouco valor em prevenção como foi dito

[11] http://labrea.sourceforge.net/

anteriormente, evidentemente isso é uma importante defesa contra ataques automatizados e *worms*.

Para finalizar os testes de interação, foi obtida a topologia de rede que o *Nmap* gera. Pode-se perceber pela Figura 19 que a topologia gerada representa fielmente a rede projetada inicialmente, com seus saltos entre roteadores e seus *hosts* presentes dentro da rede interna.

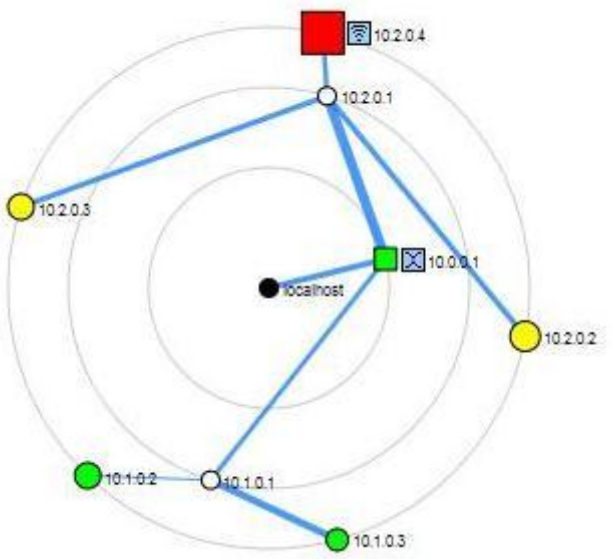


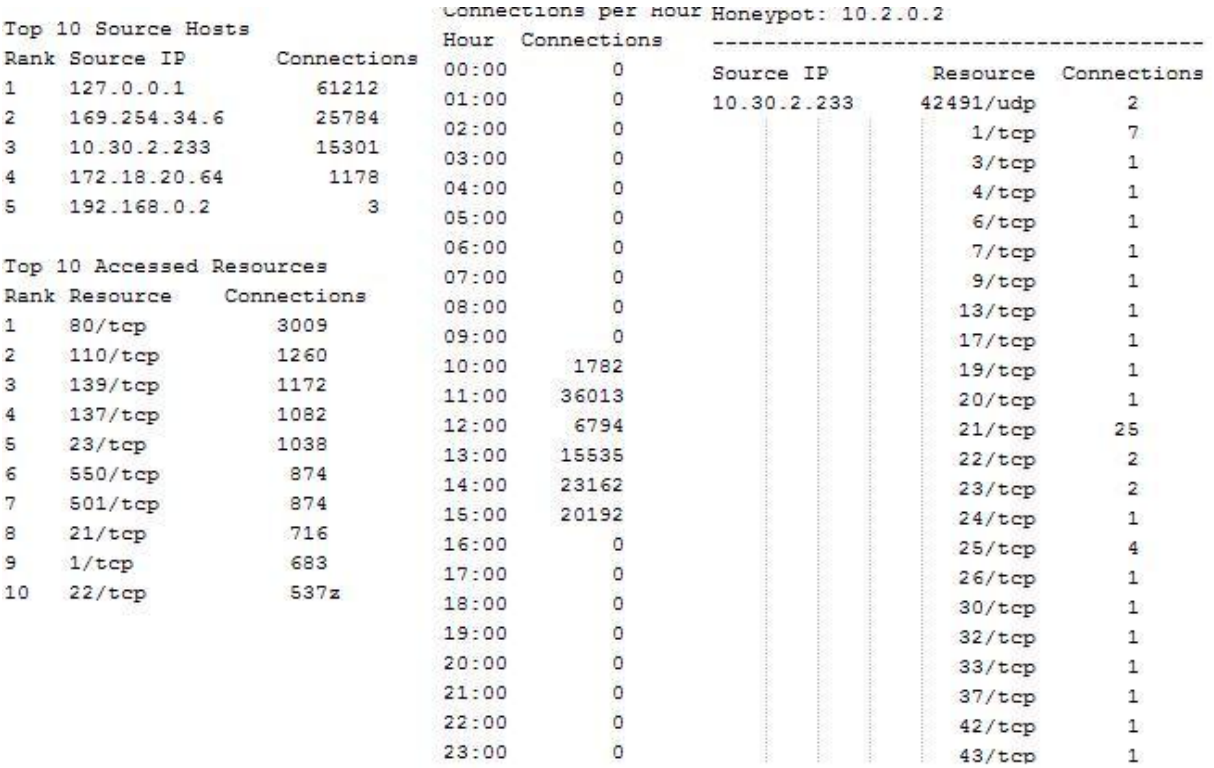
Figura 19 - Topologia de Rede Gerada Pelo *Nmap*

5.5. Resultados Obtidos

Apesar de não ter havido interação com atacantes reais, a interação feita e testada foi capaz de gerar bons resultados sobre a ferramenta. Em apenas uma semana de teste, foi gerado um arquivo de *log* de 3 *MB*, somente com o uso de alguns comandos e o *Nmap*. A Figura 20 ilustra trechos do arquivo de *log* gerado pelo *honeyd* formatado com a ferramenta *honeydsum*[12].

[12] http://www.honeynet.org.br/tools/

Observando a Figura 20, pode-se concluir que os *logs* são bem detalhados, mostrando os serviços mais acessados, os *IP*s de origem que mais se conectam nos *hosts*, além das horas que o *honeypot* recebe mais acessos. O *honeydsum* ainda gera um gráfico em uma página *HTML* (figura 21).



Top 10 Source Hosts

Rank	Source IP	Connections
1	127.0.0.1	61212
2	169.254.34.6	25784
3	10.30.2.233	15301
4	172.18.20.64	1178
5	192.168.0.2	3

Top 10 Accessed Resources

Rank	Resource	Connections
1	80/tcp	3009
2	110/tcp	1260
3	139/tcp	1172
4	137/tcp	1082
5	23/tcp	1038
6	550/tcp	874
7	501/tcp	874
8	21/tcp	716
9	1/tcp	683
10	22/tcp	537z

Connections per Hour

Hour	Connections
00:00	0
01:00	0
02:00	0
03:00	0
04:00	0
05:00	0
06:00	0
07:00	0
08:00	0
09:00	0
10:00	1782
11:00	36013
12:00	6794
13:00	15535
14:00	23162
15:00	20192
16:00	0
17:00	0
18:00	0
19:00	0
20:00	0
21:00	0
22:00	0
23:00	0

Honeypot: 10.2.0.2

Source IP	Resource	Connections
10.30.2.233	42491/udp	2
	1/tcp	7
	3/tcp	1
	4/tcp	1
	6/tcp	1
	7/tcp	1
	9/tcp	1
	13/tcp	1
	17/tcp	1
	19/tcp	1
	20/tcp	1
	21/tcp	25
	22/tcp	2
	23/tcp	2
	24/tcp	1
	25/tcp	4
	26/tcp	1
	30/tcp	1
	32/tcp	1
	33/tcp	1
	37/tcp	1
	42/tcp	1
	43/tcp	1

Figura 20 - *Log* gerado pelo honeyd

De posse desses *logs,* o analista de segurança pode ver qual recurso de sua rede é o mais visado, se há algum intruso usando alguma máquina como zumbi para acessar os *hosts* e realizar ataques automatizados, além de tentativas de varreduras com o *Nmap*.

A ferramenta *honeyd* tem muitos pontos positivos e alguns negativos. Como pontos positivos pode se ressaltar:

- facilidade na configuração e instalação;
- ampla gama de serviços e sistemas operacionais emulados;
- possibilidade de emular redes com *gateways* e roteadores;
- consegue enganar *scanners* como o *Nmap*;

- *logs* bem detalhados para análise;
- emula bem vulnerabilidades (*tarpit*) que enganam e atrasam ataques automatizados.

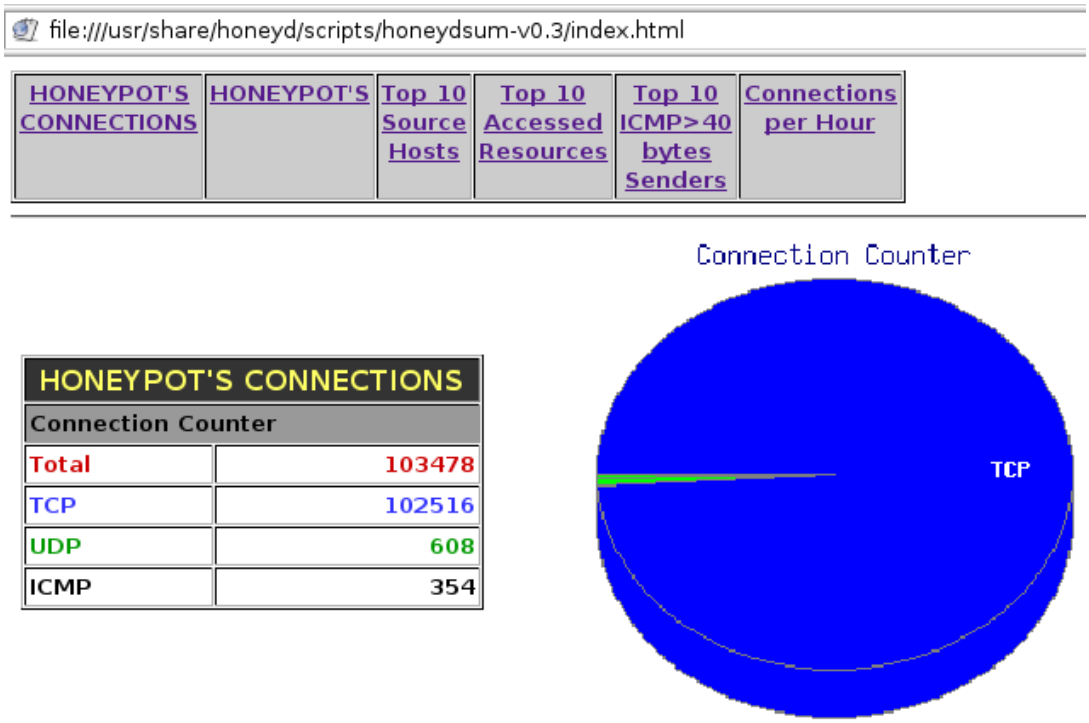


Figura 21 - Gráfico dos logs

No entanto, alguns pontos negativos podem ser observados:

- *bugs* em alguns *scripts* que emulam os serviços. Foi necessário corrigir esses *bugs* para os serviços rodarem corretamente. O apêndice B exibe trechos dos códigos alterados para garantir o funcionamento dos *scripts*;
- pouca documentação e ajuda escassa da comunidade de segurança;
- ineficácia no reconhecimento de sistemas operacionais pelo *Nmap*, gerado pela incompatibilidade dos arquivos de *fingerprints* do *honeyd* e do *Nmap.*

A implantação desse *honeypot* mostrou que a ferramenta *honeyd* pode se tornar uma importante aliada na implantação de segurança dentro de uma organização. Seus recursos são valiosos e representam fielmente o que os *honeypots* têm a oferecer de positivo no campo da segurança da informação.

6. Considerações Finais

Como foi visto, *honeypots* são uma ferramenta de gestão de segurança peculiar com o objetivo de ser sondado, atacado e comprometido. Direta ou indiretamente, ajudam a proteger sistemas de produção contra os invasores. *Honeypot* é uma tecnologia altamente flexível que pode ser aplicada a uma variedade de situações. Como ferramentas de segurança, têm vantagens específicas. Podem coletar pequenas quantidades de dados, mas a maioria dessa informação tem alto valor. Têm a capacidade de trabalhar eficazmente em ambientes de produção, ao mesmo tempo sendo dispositivos muito simples. Além disso, rapidamente demonstram seu valor através da detecção e captura de atividades não autorizadas.

Apesar dessas imensas qualidades, *honeypots* pouco fazem para afastar os atacantes. Apenas bons processos e procedimentos de segurança podem evitar que um determinado sistema seja comprometido. Procedimentos como a atualização dos antivírus, instalação de *patches* de atualização nos sistemas, e a desativação de serviços desnecessários podem impedir que os invasores explorem vulnerabilidades. São estas tarefas que tem prioridade para a segurança de qualquer organização. Com a adoção de tecnologias de *honeypot* crescente a cada ano, as organizações devem manter este pensamento em mente.

A impressão errônea sobre o que são *honeypots* e o seu valor têm levado ao atraso e à sua reprovação em organizações. Como as organizações tendem a compreender melhor o que *honeypots* podem ou não alcançar, tem ocorrido mais a aceitação destas tecnologias nos últimos anos (SPITZNER, 2002).

Honeypots se tornam mais fáceis de usar à medida que são incorporadas interfaces gráficas intuitivas. A maioria dos *honeypots* é mais específica na sua finalidade, principalmente implantados como de baixa ou de alta interação. *Honeypots* de média

interação são menos abordadas. Há um crescimento elevado do uso de *honeypots* de pesquisa, utilizadas para estudar e aprender as ameaças que existem no ciberespaço.

Os resultados da implantação de um *honeypot* nesse trabalho mostraram-se proveitosos, sendo possível observar as principais características de um *honeypot* de baixa interação, como a emulação de serviços e comportamentos que sistemas reais apresentariam. A geração dos *logs* pela ferramenta também comprovou outro valor, o da importância dos dados colhidos para análise. A ferramenta também foi útil ao emular um serviço capaz de atrasar ataques automatizados e *worms* (*tarpit*), contribuindo assim para a prevenção.

Uma proposta adicional a este trabalho seria a implantação de um *honeypot* dentro de uma rede real e com acesso à *Internet*. Com isso, as possibilidades de se perceber o comportamento de uma ferramenta de *honeypot* são maiores e mais reais. A forma como foi estudada e implantada um *honeypot* neste trabalho é somente para ilustrar os conceitos abordados no mesmo. Com certeza, a análise de tráfego real dirigido a um *honeypot* seria uma forma mais transparente de se aprender mais com esta importante ferramenta de segurança.

Com esta proposta concretizada, o próximo passo será integrar este trabalho ao Consórcio Brasileiro de *Honeypots*[13], um projeto de *honeypots* de baixa interação distribuídos que tem como objetivo aumentar a capacidade de detecção de incidentes, correlação de eventos e determinação de tendências de ataques na *Internet* brasileira. Este projeto conta com a participação de diversas instituições brasileiras, entre elas a UFRJ e a USP.

[13] Informações em http://www.honeypots-alliance.org.br/index-po.html

Referências Bibliográficas

Arpd and Honeyd Binaries. Disponível em: http://www.citi.umich.edu/u/provos/ honeyd/. Acesso em: 10/11/09.

Ataque de Negação de Serviço. Disponível em: http://www.infowester.com/ col091004. Acesso em: 15/07/09.

AZEVEDO, T. **Honeypots**: A segurança Através do Disfarce. 2005. COPPE/UFRJ, Rio de Janeiro – RJ, 2005.

BackOfficer Friendly. Disponível em: http://www.securityfocus.com/tools/2222. Acesso em: 18/11/09.

BARBATO, L.; MONTES, A. **Técnicas De Monitoração De Atividades Em Honeypots De Alta Interatividade.** Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos – SP.

Blackhats. Disponível em: http://www.blackhat.com/. Acesso em: 05/09/09.

Brazilian Honeypots Alliance. Disponível em: http://www.honeypots-alliance.org.br/stats/ Acesso em: 02/12/09.

CERT. Disponível em: http://www.cert.br/ Acesso em: 09/09/09.

CHESWICK**,** B**. An Evening With Berferd**. Disponível em: http://www. cheswick.com/ches/papers/berferd.pdf. Acesso em: 03/09/09.

Configurando e Instalando o Honeyd. Disponível em: http://ricardomartins.com.br/ 2009/04/10/configurando-e-instalando-o-honeyd/. Acesso em: 04/11/09.

Decoy Server. Disponível em: http://www.symantec.com. Acesso em: 18/11/09.

DoS e DDoS. Disponível em: http://www.acm.org/. Acesso em: 10/08/09.

Honeyd, Honeywall. Disponível em: https://www.os3.nl/2008-2009/students/ yanick_de_jong/assignment_week_4_-_honeyd_honeywall. Acesso em: 11/11/09.

Honeynets. Disponível em: http://www.honeynet.org/papers/honeynet. Acesso em: 11/11/09.

Installing Perl Modules. Disponível em: http://www.livejournal.com/ doc/server/lj.install.perl_setup.modules.html. Acesso em: 11/11/09.

Libdnet. Disponível em: http://libdnet.sourceforge.net. Acesso em: 10/11/09.

Libevent. Disponível em: http://www.monkey.org/~provos/libevent. Acesso em: 10/11/09.

Libpcap. Disponível em: http://www.tcpdump.org/. Acesso em: 10/11/09.

LUIZ, S. Projeto de honeypots divulga estatísticas de potenciais ataques. Disponível em: http://www.forumpcs.com.br/noticia.php?b=107045. Acesso em: 18/11/09.

MANOEL, SOARES e DA SILVA. **Honeypot**: Enganando e Conhecendo o Inimigo. 2004. Escola de Informática, UNIGRANRIO – Universidade do Grande Rio, Duque de Caxias – RJ, 2004.

MARCELO, A.; PITANGA, M. **Honeypots –** A Arte de Iludir Hackers. 1. Ed. Rio de Janeiro: Brasport, 2003.

MARINHO, R. **Honeypots**: Acompanhando Os Passos De Uma Invasão Em Tempo Real. 2004. Universidade de Fortaleza – UNIFOR, Fortaleza – CE, 2004.

Nessus. Disponível em: http://www.nessus.org. Acesso em: 15/07/09.

Nessus Report. Disponível em: http://www.sun.com. Acesso em: 10/08/09.

Nmap. Disponível em: http://nmap.org. Acesso em: 15/07/09.

PeopleWare. Disponível em: http://pplware.sapo.pt/**.** Acesso em: 26/11/09**.**

Project Honeynet. Disponível em: http://www.honeynet.org. Acesso em: 03/09/09.

PROVOS, N.; HOLZ, T. **Virtual Honeypots**: From Botnet Tracking to Intrusion Detection. 2. Ed. Addison Wesley Professional, 2007.

PROVOS, N. **Honeyd**. Disponível em: http://www.honeyd.org. Acesso em: 18/11/09.

Rootkits. Disponível em: http://www.linhadefensiva.org/2005/03/rootkit/. Acesso em: 05/09/09.

SCHNEIER, B. Disponível em: http://www.schneier.com. Acesso em: 11/11/09.

Snort. Disponível em: http://www.snort.org. Acesso em: 25/11/09.

Specter. Disponível em: http://www.specter.com. Acesso em: 18/11/09.

SPITZNER, L. **Honeypots**: Definitions And Values. Disponível em: http:// www.spitzner.net/honeypots.html. Acesso em: 07/07/09.

SPITZNER, L. **Honeypots:** Tracking Hackers. 1. Ed. Addison Wesley, 2002.

SPITZNER, L. **Honeypots**: Tracking Hackers. Disponível em: http://www.tracking-hackers.com. Acesso em: 07/07/09.

SPITZNER, L**. Open Source Honeypots**: Learning with Honeyd. Disponível em: http://www.securityfocus.com/infocus/1659. Acesso em: 07/07/09.

SPITZNER, L. **Open Source Honeypots, Part Two**: Deploying Honeyd in the Wild. Disponível em: http://www.securityfocus.com/infocus/1675. Acesso em: 07/07/09.

STOOL, C**. The Cuckoo's Egg**. Disponível em: http://www.ci.ulsa.mx/~elinos/ docencia/herseg/cuckoo_egg.pdf. Acesso em: 03/09/09.

Testing Honeyd Without A Network. Disponível em: http://aufwiedersehen 5.blogspot.com/2009/10/testing-honeyd-without-network.html. Acesso em: 04/11/09.

Tools Developed by Honeynet.BR. Disponível em: http://www.honeynet.org.br/ tools/. Acesso em: 07/07/09.

Apêndice A – Arquivo de Configuração do *Honeyd*

```
#Configuração do Honeyd
#Criação dos roteadores
create router1
set router1 personality "Cisco 7206 running IOS 11.1(24)"
set router1 default tcp action reset
add router1 tcp port 23 "perl /usr/share/honeyd/scripts/router-telnet.pl"
create router2
set router2 personality "Cisco 762 Non-IOS Software release 4.1(2) or 766 ISDN router"
set router2 default tcp action reset
add router2 tcp port 23 "perl /usr/share/honeyd/scripts/router-telnet.pl"
create router3
set router3 personality "Cisco IOS 11.3 - 12.0(11)"
set router3 default tcp action reset
set router3 default udp action reset
add router3 tcp port 23 "perl /usr/share/honeyd/scripts/router-telnet.pl"
# Template Default
create default
set default personality "FreeBSD 2.2.1-STABLE"
set default default tcp action reset
add default tcp port 80 "sh /usr/share/honeyd/scripts/web.sh"
add default tcp port 22 "sh /usr/share/honeyd/scripts/test.sh"
add default tcp port 113 reset
add default tcp port 1 reset
# Tarpit Template
# Máquina Mac destinada a diminuir a velocidade de ataques de rootkits e worms
create sticky
set sticky personality "Apple Mac OS X 10.2.6 (Jaguar)"
set sticky default tcp action tarpit open
set sticky default udp action block
# Máquinas Windows
create windows1
set windows1 personality "Microsoft Windows XP Professional SP1"
set windows1 default tcp action reset
set windows1 default udp action reset
set windows1 default icmp action open
add windows1 tcp port 80 "perl /usr/share/honeyd/scripts/win32/iis-0.95/iisemul8.pl"
```

```
add windows1 tcp port 139 open
add windows1 tcp port 137 open
add windows1 udp port 137 open
add windows1 udp port 135 open
set windows1 uptime 3284460
create windows2
set windows2 personality "Microsoft Windows XP Home Edition"
set windows2 default tcp action reset
set windows2 default udp action reset
set windows2 default icmp action open
add windows2 tcp port 1080 "perl /usr/share/honeyd/scripts/mydoom.pl -l
/var/logs/honeypot/#mydoom.log"
add windows2 tcp port 3127 "perl /usr/share/honeyd/scripts/mydoom.pl -l
/var/logs/honeypot/#mydoom.log"
set windows2 uptime 3284460
# Máquinas Linux
create linux1
set linux1 personality "Linux kernel 2.4.18 - 2.4.20 (X86)"
set linux1 default tcp action reset
set linux1 default udp action reset
add linux1 tcp port 110 "sh
/usr/share/honeyd/scripts/unix/general/pop/emulate-pop3.sh"
add linux1 tcp port 21 "sh /usr/share/honeyd/scripts/unix/general/ftp.sh"
add linux1 tcp port 25 "sh /usr/share/honeyd/scripts/unix/general/smtp.sh"
add linux1 tcp port 80 "sh /usr/share/honeyd/scripts/unix/general/web.sh"
set linux1 uptime 3284460
create linux2
set linux2 personality "Linux 2.4.16 - 2.4.18"
set linux2 default tcp action reset
set linux2 default udp action reset
add linux2 tcp port 23 "perl
/usr/share/honeyd/scripts/unix/general/telnet/faketelnet.pl"
add linux2 tcp port 22 "sh /usr/share/honeyd/scripts/test.sh"
add linux2 tcp port 161 "perl
/usr/share/honeyd/scripts/unix/general/snmp/fake-snmp.pl"
add linux2 tcp port 501 open
add linux2 tcp port 550 open
add linux2 udp port 601 open
set linux2 uptime 3284460
# Linux Suse 8.0
create suse80
set suse80 personality "Linux 2.4.7 (X86)"
```

```
set suse80 default tcp action reset

set suse80 default udp action block

set suse80 default icmp action open

add suse80 tcp port 21 "sh
/usr/share/honeyd/scripts/unix/linux/suse8.0/proftpd.sh"

add suse80 tcp port 22 "sh
/usr/share/honeyd/scripts/unix/linux/suse8.0/ssh.sh"

add suse80 tcp port 23 "sh
/usr/share/honeyd/scripts/unix/linux/suse8.0/telnetd.sh"

add suse80 tcp port 25 "sh
/usr/share/honeyd/scripts/unix/linux/suse8.0/sendmail.sh"

add suse80 tcp port 79 "sh
/usr/share/honeyd/scripts/unix/linux/suse8.0/fingerd.sh"

add suse80 tcp port 80 "sh
/usr/share/honeyd/scripts/unix/linux/suse8.0/apache.sh"

add suse80 tcp port 110 "sh
/usr/share/honeyd/scripts/unix/linux/suse8.0/qpop.sh"

add suse80 tcp port 3128 "sh
/usr/share/honeyd/scripts/unix/linux/suse8.0/squid.sh"

add suse80 tcp port 8080 "sh
/usr/share/honeyd/scripts/unix/linux/suse8.0/squid.sh"

add suse80 tcp port 8081 "sh
/usr/share/honeyd/scripts/unix/linux/suse8.0/squid.sh"

add suse80 udp port 161 "perl
/usr/share/honeyd/scripts/unix/general/snmp/fake-snmp.pl public private --
config=scripts/unix/general"

add suse80 udp port 514 "sh
/usr/share/honeyd/scripts/unix/linux/suse8.0/syslogd.sh"

set suse80 uptime 79239

#Configuração da Rede Virtual

route entry 10.0.0.1 network 10.0.0.0/8

route 10.0.0.1 link 10.0.0.0/24

route 10.0.0.1 add net 10.1.0.0/24 10.1.0.1

route 10.0.0.1 add net 10.2.0.0/24 10.2.0.1

route 10.1.0.1 link 10.1.0.0/24

route 10.2.0.1 link 10.2.0.0/24

bind 10.0.0.1 router1

bind 10.1.0.1 router2

bind 10.2.0.1 router3

bind 10.0.0.2 sticky

bind 10.1.0.2 windows1

bind 10.1.0.3 windows2

bind 10.2.0.2 linux1

bind 10.2.0.3 linux2

bind 10.2.0.4 suse80
```

Apêndice B – Alteração dos scripts de serviços do honeyd

Script web.sh, linhas 2 a 15:

```
REQUEST=""
while read name
do
      LINE=`echo "$name" | egrep -i "[a-z:]"`
      if [ -z "$LINE" ]
      then
            break
      fi
      echo "$name" >> /tmp/log
      NEWREQUEST=`echo   "$name"   |   grep   "GET   .scripts.*cmd.exe.*dir.*
HTTP/1.0"`
      if [ ! -z "$NEWREQUEST" ] ; then
            REQUEST=$NEWREQUEST
      fi
done
```

Script ftp.sh, linhas 21 a 31:

```
#set -x -v
DATE=`date`
host=`hostname`
domain=`dnsdomainname`
log=/tmp/honeyd/ftp-$1.log
AUTH="no"
PASS="no"
echo "$DATE: FTP started from $1 Port $2" >> $log
echo -e "220 $host.$domain FTP server (Version wu-2.6.0(5) $DATE) ready.\r"
while read incmd parm1 parm2 parm3 parm4 parm5
do
```