UNIVERSIDADE
ESTADUAL DE LONDRINA

Trabalho de Conclusão de Curso

Rafael de Paula Herrera

Um estudo sobre o uso de Honeypots

Londrina

2009

Rafael de Paula Herrera

Um estudo sobre o uso de Honeypots

Trabalho apresentado à Universidade Estadual de Londrina, como parte do requisito para a obtenção do título de Bacharel em Ciência da Computação, sob orientação do Prof. Dr. Mario Lemes Proença Jr.

Orientador:
Mario Lemes Proença Jr.

Universidade Estadual de Londrina

Londrina

2009

Rafael de Paula Herrera

Um estudo sobre o uso de Honeypots

Prof. Dr. Mario Lemes Proença Jr.
Universidade Estadual de Londrina

Prof. Dr. Alan Salvany Felinto
Universidade Estadual de Londrina

Prof. Dr. Jacques Duílio Brancher
Universidade Estadual de Londrina

Londrina, 16 de Novembro de 2009

Aos solos de guitarra tocados por Jimi Hendrix

Agradecimentos

Aos meus pais, Jonas e Maria Alice, pela oportunidade de passar por este planeta chamado, por nós seres humanos, Terra. Tenho plena consciência que sem seu apoio durante os momentos mais difíceis de minha vida, não teria a chance de buscar pelos meus sonhos, tão pouco teria condições de contar estruturalmente com bases sólidas para meu desenvolvimento sócio-cultural. A estes dois indivíduos dedico todo o meu amor, respeito e profunda admiração.

À minha irmã, Adriana, por ter sido, sempre, uma grande companheira ao longo de minha trajetória. Mesmo que desde o início de minha graduação nosso contato tenha sido frequentemente interrompido, busco em suas grandes obras artísticas a inspiração para que os dias não se façam rotineiros e que, sejam sempre repletos pela incessante busca do novo-criativo.

À memória de minha avó paterna, Eurides, que em vida sempre apoiou minhas decisões, sendo um verdadeiro exemplo de perseverança, garra e força de opinião, fato que me impulsiona até os dias de hoje, na busca pelos meus objetivos.

À memória de meu avô materno, Hélio, que em vida transmitiu sábios ensinamentos, tendo ajudado a moldar minha personalidade tal como ela é e, acima de tudo, por ser uma das maiores amizades que construí até hoje.

Aos meus amigos, por me acompanharem sempre almejando o melhor, trazendo muitos momentos de alegria, tendo sido escolhidos por mim, como membros genuínos de minha família.

Ao Bodão do passado, por me fazer orgulhoso e, não menos importante, ao Bodão do futuro, por realizar meus sonhos.

"I'm going to make him an offer he can't refuse."

Don Vito Corleone - The Godfather (1972)

Resumo

Honeypots são tecnologias empregadas na indústria de segurança de redes. Seus valores consistem, ao contrário de muitas ferramentas de segurança, em serem reconhecidos, atacados e, eventualmente, comprometidos. Em geral, são constituídos por computadores especialmente preparados para que tais ataques sejam efetivados contra os mesmos, funcionando como isca para diversos tipos de atacantes. (SPITZNER, 2002)

Ao se estudar os métodos empregados nas investidas contra os *Honeypots*, são geradas várias formas de conhecimento, benéficas no sentido de que podem ser utilizadas para aumentar a segurança dos ambientes de produção e gerar alertas sobre novas ameaças para a comunidade de especialistas em segurança no mundo todo.

O propósito deste documento é mostrar ao leitor os conceitos envolvidos na temática dos *Honeypots*. Serão abordadas as vantagens e desvantagens envolvidas em sua implantação, assim como os aspectos legais em sua utilização. Algumas soluções em *Honeypots* serão descritas, tendo suas características mais marcantes expostas, no intuito de incentivar a implantação daquelas que sejam interessantes em cenários compatíveis com seu nicho de atuação.

Palavras-chave: Honeypots, Segurança, Redes de Computadores, Hacking.

Abstract

Honeypots are technologies employed in the netword security industry. Its values consists, instead of many security tools, in being recognized, attacked and, eventually, compromised. In general, are constituted by computers specially prepared for such attacks to take effect against them, working as a bait to several kinds of attacks. (SPITZNER, 2002)

By studying the employed methods in the chargins against the *Honeypots*, there are many forms of knowledge generated, beneficals in the sense that it can be used to increase the security in production environments and generates alerts about new threats to the security expert's comunity in worldwide.

This document's purpose is to shows up to the reader the concepts involved in *Honeypots'* thematic. There are going to be addressed the advantages and disavantages involved in their deployment, as well as the legal aspects in its use. Some solutions in *Honeypots* are described, having their salient characteristics exposed, in the order to encourage the deployment of those wich are interesting in compatible scenarios with their performance niche.

Keywords: Honeypots, Security, Computer Networks, Hacking.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de abreviaturas e siglas

ABIN	Agência Brasileira de Inteligência
ABRANET	Associação Brasileira de Provedores de Serviços de Internet
ACK	Aknowledge
ARP	Address Resolution Protocol
CERT	Computer Emergency Response Team
CSS	Cascading Style Sheet
DDoS	Distributed Denial of Service
DMA	Direct Memory Access
DoS	Denial of Service
DSL	Digital Subscriber Line
FIFO	First In, First Out
FTP	File Transfer Protocol
GHH	Google Hack Honeypot
GNU	GNU Not Unix
HTML	Hypertext Markup Language
IAP	Internet Access Provider
ICMP	Internet Control Message Protocol
INPE	Instituto Nacional de Pesquisas Espaciais
INTERPOL	International Criminal Police Organization
IPv6	Internet Protocol Version 6
ISP	Internet Service Provider
NDoS	Network Denial of Service

PCRE	Perl Compatible Regular Expressions
RFC	Request for Comment
RST	Reset
SSH	Secure Shell
SYN	Synchronize
SYN-ACK	Sinchronize-Aknowledge
TCP	Transmission Control Protocol
TFN	Tribe Flood Network
UDP	User Datagram Protocol
UML	User-Mode-Linux
XHTML	Extensible Hypertext Markup Language
XML	Extensible Markup Language

Introdução

O uso da Internet teve um grande crescimento nos últimos anos (GROUP, 2009) e a maioria dos seus usuários são pessoas com baixo nível técnico computacional. É comum que tenham seus computadores infectados por pragas virtuais, que muitas vezes são programadas para lançar ataques em servidores alvo.

Podemos observar uma tendência de aumento na quantidade de dispositivos móveis interligados por meio de redes IP como: telefones celulares, smartphones e sensores (CHANG, 2007). Esse crescimento faz parte de um fenômeno que chamamos de Convergência. Isso nos remete imediatamente à necessidade de que sejam aperfeiçoadas as técnicas de prevenção e detecção de intrusões.

Neste cenário, os profissionais de segurança utilizam Honeypots (SADASIVAM *et al.*, 2005) como ferramentas de auxílio, sendo máquinas configuradas para coletarem dados sobre ataques sofridos. Seu intuito é atrair atacantes oferecendo um ambiente repleto de falhas de segurança, sem que eles saibam que tais ambientes são controlados e totalmente monitorados.

O sucesso nos ataques irá depender da efetividade das técnicas empregadas e da habilidade de quem os realiza, que por sua vez terá todos os seus passos registrados para uma análise posterior. Essa análise pode revelar um conjunto de informações valiosas que auxiliam a tarefa de manter as redes seguras. Os Honeypots registram atividades legitimamente maliciosas, pois suas identidades como ferramentas de segurança são mantidas sob sigilo, forjando-se comportamentos semelhantes aos apresentados por máquinas reais na rede monitorada.

Com um estudo sobre a natureza e a implantação de *Honeypots*, espera-se gerar um material capaz de agregar conhecimento diferenciado sobre segurança, servindo como base para que os administradores de rede possam estabelecer melhores políticas de segurança em suas

instituições. Este trabalho conta com uma descrição em detalhes de cada ítem contido em uma lista de 6 soluções em *Honeypots*. A partir desta iniciativa, espera-se que as tecnologias abordadas possam ser comparadas e o processo de escolha entre as mesmas seja melhor direcionado, visando o sucesso em sua implantação e na identificação/monitoramento de atividades ilegais.

A organização do trabalho obedece a seguinte estrutura:

- No capítulo 1 serão discutidas quais são, atualmente, as maiores formas de ameaças na *Internet*, mostrando em qual contexto cada uma encontra-se inserida.
- No capítulo 2 será apresentado o conceito de Honeypot, como forma de auxílio na melhora dos processos de segurança envolvidos nas instituições. Serão abordados os tipos de *Honeypots* encontrados no mercado, boas práticas de implantação, as principais organizações mundiais comprometidas no seu desenvolvimento e as implicações legais pertinentes a sua utilização.
- No capítulo 3 são apresentadas 6 tipos de soluções compreendidas pelas tecnologias em *Honeypots*. Suas características mais marcantes serão exploradas, no intuito de trazer um apanhado de informções que sejam capazes de guiar a escolha de quais tecnologias serão mais bem aceitas nos ambientes de pesquisa/produção, de acordo com seus propósitos.

1 Ameaças

As redes de computadores são alvos para inúmeras formas de ameaças. Será apresentado um conjunto compreendido por algumas das mais relevantes, levando em consideração sua capacidade de causar danos políticos, sociais e financeiros à sociedade, ainda que algumas delas tenham hoje em dia, somente sua importância histórica.

Antes de se atentar aos detalhes relacionados a tal seleção, é de grande valia que se saiba quais são os fatores que motivam os ataques, visto que são inúmeros os tipos de atacantes e alvos envolvidos no confronto.

Instituições de todos os gêneros e portes sofrem, frequentemente, com investidas advindas de diversos tipos de atacantes. Na maioria das vezes, são acionadas ferramentas completamente automatizadas (MCCARTY, 2003) e que não necessitam de interferência humana para cumprirem seus objetivos. Alternativamente, existem situações onde os ataques são direcionados, ou seja, são preparados e lançados por profissionais especializados em segurança e testes de penetração, sua bagagem técnina é grande, tendo envolvimento, em diversos níveis, com a comunidade *Hacker*.

Pelo seu comportamento, pode-se atribuir aos atacantes diversas classificações, que variam de acordo com sua capacidade técnica e motivos que os levam à exploração de falhas, sendo que em muitos casos, chegam a praticar delitos em prol de benefícios individuais desejados.

As ameaças abordadas neste trabalho, podem ser visualizadas na tabela 1, de acordo com sua classificação:

Tabela 1: Classificação dos tipos de ameaças

Tipo de Ameaça	**Classificação**
Cyber Warfare	Ameaça política
Redes de pedofilia	Ameaça social
Ataques de Negação de Serviço	Ameaça econômica e/ou política

1.1 Cyber Warfare

Historicamente, é comum que interesses de determinados grupos conduzam a humanidade a conflitos. Em geral, essas disputas envolvem diversos tipos de organizações, mediante os ideais que alimentam suas operações.

Nos dias de hoje, além dos conflitos armados entre as nações, são empregadas táticas de guerra nas redes de computadores, envolvendo grupos militares e paramilitares.

Em geral, as pessoas recrutadas para a participação neste tipo de atividade, recebem treinamento nas áreas de espionagem, segurança da informação, *hacking* e *Computação Forense*. Tais conhecimentos servem às organizações para espalhar caos e promover roubo de informações em instituições governamentais e privadas. Mesmo os governos e o setor privado, além de grupos criminosos, investem(OWENS *et al.*, 2009) em capacitação do capital humano para que o fluxo de atividades relacionadas esteja sempre em alta.

Os ataques promovidos, vão além das fronteiras físicas e políticas, podendo afetar a todos que estejam conectados à *Internet*. São alvos comuns: governos, instituições financeiras, grids científicos, *ISPs*, também conhecidos por *Internet Access Providers* (*IAP*) e até mesmo usuários domésticos. Existem casos mais críticos, onde o foco dos ataques são as redes de abastecimento de água e luz, infra-estruturas hospitalares, mecanismos de defesa militares e sistemas de robótica em organizações privadas.

É comum que grupos terroristas queiram desestabilizar governos, tentando promover possíveis quedas no abastecimento de serviços básicos, como foram constatados em ataques aos

sistemas que controlam a rede elétrica norte-americana. No meio corporativo temos a espionagem industrial, ocorrida entre empresas concorrentes. Também fazem parte da lista, os crimes financeiros à bolsa de valores, que podem causar quebras de bancos e grandes empresas, endividamento de países inteiros e consequente diminuição do poder aquisitivo da população, como possível reflexo desse quadro.

O *grid* de energia elétrica norte-americano, por exemplo, se conecta à várias redes, inclusive à *Internet*, e tem grande parte de sua operação automatizada. Seu governo federal admitiu que o *grid* é suscetível à guerra cibernética e levou a publico algumas notas onde são reportados ataques provenientes da China e Rússia(KREKEL, 2009) (WERMSKE, 2009) (ACOHIDO, 2009). Um ataque com sucesso, neste caso seria o suficiente para interromper o fornecimento de energia elétrica para várias partes da nação, o que poderia causar desde traumas na opinião pública, abrir brechas para ataques militares distribuídos enquanto mantém pontos de distração perante à guarda, até um grande impacto econômico do país.

A promoção de guerras cibernéticas é, portanto, um forte indício de que se efetivaram inúmeras mudanças na maneira como a sociedade se reorganizou em torno da era da informação. Sem que sejam implantadas boas políticas de segurança e com o sucesso em ataques direcionados, pode-se ocasionar colapsos em inúmeros setores, básicos até mesmo para a manutenção da ordem em nações inteiras.

1.2 Redes de pedofilia

O fácil acesso à *Internet*, trouxe às organizações criminosas espalhadas pelo mundo afora, a possibilidade da troca de informações de maneira ágil, no intuito de solidificarem e difundirem suas atividades ilícitas.

Atualmente, um dos grandes desafios dos profissionais de segurança, é auxiliar a justiça criminal no combate das redes de pedofilia, de modo que seja efetiva a perseguição aos infratores, devido seus agravantes contra a sociedade.

Uma prática comum entre este tipo de comunidade criminosa, é a busca por servidores

comprometidos, que possam fornecer espaço em disco e banda suficiente para a hospedagem e distribuição de conteúdo multimídia, como fotos e vídeos, envolvendo suas vítimas. Tais criminosos podem utilizar canais de *IRC* para comunicação instantânea, e hospedarem robôs especializados na distribuição de seus arquivos entre os interessados, de modo que os *downloads* sejam redirecionados para servidores que eventualmente tenham sido tomados, para que a troca dos dados seja mais eficaz.

Existem também, redes de distribuição de pornografia infantil que comercializam seus materiais, mediante transações financeiras, que em sua grande maioria são internacionais, assim como evidencia um documento (REIS; BECKER, 2004) produzido pelo Governo Brasileiro que, como colaboradores, teve diversos membros de organizações como a *Associação Brasileira de Provedores de Serviços de Internet* (*ABRANET*) e a *Agência Brasileira de Inteligência* (*ABIN*), em conjunto com a *International Criminal Police Organization* (*INTERPOL*).

Por esta razão, é importante que sejam empregadas tecnologias que possam monitorar as atividades nas redes de computadores, principalmente em instituições cujo poderio computacional é grande. Dessa maneira, esta categoria de criminosos, poderá ser rastreada com mais facilidade por autoridades e *Instituições de Pesquisa Forense Digital*, contribuindo em âmbito mundial, com a diminuição da quantidade de vítimas deste tipo de atividade ilegal.

1.3 Ataques de Negação de Serviço em Rede

Apesar de serem amplamente conhecidos, os ataques do tipo de Negação de Serviço em Rede, também conhecidos pela terminologia inglesa *Network Denial of Service* (*NDoS*), são vistos a partir de diferentes óticas pela comunidade de pesquisadores e especialistas envolvidos em estudos de segurança. Não existe uma definição formal e genérica para estes tipos de ataques, pois cada um deles é fruto da observação de casos reais e da extração das técnicas empregadas em suas execução.(SHIELDS, 2002)

Alguns autores os classificam como sendo tão somente o consumo, pelos atacantes, de recursos disponíveis na rede e que por consequência, fazem com que seu uso legítimo seja

comprometido. Outros autores dizem que esses ataques têm a característica de causarem mau-funcionamento em dispositivos necessários para a entrega de pacotes, como é o caso de quando são disparados contra os *roteadores*. Há ainda afirmações no sentido de que são fruto de qualquer ataque que resulte na indisponibilidade de informações, quando solicitadas, ou mesmo na corrupção destas. Todas essas abordagens mantém o foco no resultado gerado pelo ataque que, de uma maneira mais abrangente, resume-se na negação dos serviços solicitados.

Existem várias modalidades de ataques *DoS*, dentre os quais, destacam-se por sua popularidade, importância histórica e caráter inovador, o *Ping of Death*, *Ping Flooding*, *TCP SYN Flood* e *DoS Distribuído*, sendo que a partir destes, surgiram ao longo do tempo, inúmeras variantes, cada uma com particularidades únicas, que as tornam efetivas nos ambientes para os quais foram projetadas. Seus principais aspectos técnicos serão abordados sequencialmente, de maneira que forneçam a este estudo uma visão mais ampla dos motivos pelos quais são consideradas tão destrutivos em ambientes interconectados em redes.

1.3.1 Ping of Death

Consiste em enviar *pings* mal-formados (maliciosos), para um computador alvo. Um pacote *ping* tem, por padrão, um tamanho de 56 *bytes* (84 *bytes*, se o cabeçalho for considerado) e historicamente, os Sistemas Operacionais não eram capazes de lidar com pacotes *ping* que fossem maiores que o tamanho normal de um pacote *IP*, o que corresponde a 65.535 *bytes* (INSTITUTE, 1981). Seria incorreto enviar um pacote maior que o tamanho máximo definido, mas é possível realizar isto se utilizando da capacidade de fragmentação dos pacotes.

Quando os sistemas operacionais recebiam pacotes fragmentados maiores que 65.535 *bytes*, era frequente a ocorrência de comportamentos anormais, travamentos, e até mesmo *reboots*, devido a um *buffer overflow* que poderia ocorrer quando os pacotes maliciosos eram remontados. Esse exploit afetou muitos sistemas operacionais, como o *Unix*, *Linux*, *Mac*, *Windows* e até mesmo aqueles embarcados em impressoras e roteadores. No entanto, a grande maioria dos sistemas tiveram suas falhas corrigidas entre os anos de 1997 e 1998.

O problema em si não era relacionado ao protocolo utilizado pelo o utilitário *ping* (*ICMP*, e

sim ao processo de remontagem dos pacotes *IP*, que podem carregar qualquer tipo de protocolo, como por exemplo *TCP*, *UDP* ou *IGMP*. O processo de correção desta falha consistiu em adicionar checagens durante a remontagem dos pacotes, onde para cada fragmento de pacote recebido, avalia-se os campos de *deslocamento do fragmento* e *comprimento total*, presentes em seu cabeçalho *IP*, que somados não podem ultrapassar a marca de 65.535. Caso a soma seja maior, o pacote é invalidado e o fragmento *IP* é ignorado. Tal checagem é realizada frequentemente em *firewalls*, no intuito de proteger sistemas que não tenham essa correção implementada. Outra solução para este tipo de problema, embora não muito bem aceita por quebrar os padrões estabelecidos, é manter um *buffer* maior que 65.535 *bytes* para o processo de remontagem dos pacotes.

Atualmente outro tipo de ataque baseado em *pings* é empregado, sua denominação é *Ping Flood* e por definição, seu funcionamento é simples, baseando-se apenas na premissa de causar uma "inundação" de *pings* no alvo, comprometendo a quantidade de tráfego na rede a qual este pode interagir. Isso faz com que o sistema vitimado tenha problemas tanto para responder às requisições realizadas, quanto para realizar novas requisições.

1.3.2 Ping Flood

Um ataque deste tipo, caracteriza-se por um *DoS* onte o atacante sobrecarrega a vítima com pacotes *ICMP Echo Request*. Sendo bem sucedido somente, se o atacante possui mais largura de banda disponível que a vítima (por exemplo, um atacante que possui um link *DSL* e uma vítima que possui um modem *dial-up*). O atacante espera que a vítima responda com pacotes *ICMP Echo Response*, consumindo toda sua banda de saída quanto sua banda de entrada.

Para reduzir os efeitos de uma inundação de *pings*, pode-se utilizar *firewalls* para ou filtrar completamente os pacotes *ICMP Echo Request* de entrada, ou filtrar um grande número de requisições recebidas em um curto intervalo de tempo. Recusar a resposta produz dois benefícios: menos banda é desperdiçada quando não se responde às requisições e se dificulta ao atacante medir a efetividade de seu ataque. No entanto, tal atitude irá prevenir as medições

de latência de usuários legítimos, o que pode ser indesejado, além de infringir as definições do *RFC 1122* (BRADEN, 2009). Uma solução mais elegante, consiste em filtrar somente os pacotes *ICMP Echo Request* demasiadamente grandes ou então limitar a taxa com que o *firewall* deixa passar os pacotes *ICMP Echo Request*.

Não se pode confiar na procedência do endereço *IP* remetente, uma vez que este pode ter sido falsificado, fazendo com que os pacotes *ICMP Echo Request* pareçam vir de outros endereços, sejam eles específicos ou randômicos.

1.3.3 Smurf attack

Trata-se de um ataque *DoS*, onde é gerada uma quantidade abusiva de tráfego na rede da vítima. A inundação ocorre com pacotes de *ICMP Echo Response* e isso é possível pois, previamente, manipula-se maliciosamente os pacotes de *ICMP Echo Request*, colocando-se o endereço *IP* da vítima como remetente de todos os pacotes a serem enviados, que têm como destino vários endereços de *broadcast* nas redes alcançáveis.

Se um dispositivo de roteamento, que entrega tráfego ao endereço de *broadcast*, entregar como remetente dos pacotes *ICMP Echo Request*, o mesmo endereço *IP* de *broadcast*, a maioria das máquinas nessa rede *IP* responderão ao pacote recebido com um pacote *ICMP Echo Response*, fazendo com que o tráfego seja multiplicado pelo número de máquinas que responderem. Em redes de *broadcast* multi-acesso, isso faz com que centenas de máquinas possam responder a cada pacote.

Em meados de 1990, muitas redes *IP* estiveram suscetíveis como alvos de ataques *Smurf* (ou seja, responderem a *pings* destinados a endereços de *broadcast*). Devido à facilidade de se evitar este tipo de ataque e da conscientização da maioria dos administradores de rede, pouquíssimas redes se encontram vulneráveis nos dias de hoje.

Amplificadoras *Smurfs*, são redes que se permitem utilizar, devido sua má-configuração, por ataques *Smurf*. Em geral, agem amplificando o efeito deste tipo de ataque, pois geram quantidades enormes de pacotes *ICMP Echo Response* aos alvos, através da falsificação de seus endereços *IP*.

Para evitar o ataque, é preciso que sejam configurados, tanto roteadores quanto máquinas individuais, para que não respondam a *pings* destinados a endereços de *broadcast*. É preciso também, que se configure os roteadores para que não encaminhem pacotes diretamente a endereços *broadcast*. Até 1999, os padrões adotavam como uma das características básicas dos roteadores, a capacidade de encaminhamento de pacotes para o endereço de *broadcast*, o que foi abolido cerca de 1 ano mais tarde, mudando o os padrões da indústria e o comportamento dos roteadores, de modo que não mais encaminhassem esse tipo de pacotes.

Uma outra solução que pode ser empregada para evitar este ataque, é a chamada *Filtragem de Entrada* aliada à *Filtragem de Saída*, que irão avaliar quais são os endereços de *IP* válidos, que são caracterizados pelas redes interconectadas fisicamente. Obviamente não é aplicável à *Internet*, no entanto essa abordagem é extremamente pertinente quando utilizada entre *ISPs*, Universidades e grandes corporações, devido ao fato de todos esses exemplos possuirem, em geral, várias redes internas interconectadas e também por se conectarem às redes externas de mesmo gênero, além da *Internet*. Assim fica estabelecida uma "política de boa vizinhança" entre tais redes, onde só irão trafegar pacotes cuja fonte seja autorizada perante à topologia empregada.

1.3.4 Fraggle Attack

É um tipo de *DoS*, em que o atacante envia uma grande quantidade de tráfego *echo UDP* para endereços de *broadcast*, todos com seus remetentes falsos. Trata-se de uma reescrita de código do ataque *Smurf*. Ambos os ataques foram criados pelo mesmo autor, conhecido na comunidade por *TFreak*. As portas alvo deste tipo de ataque são a 7 (*echo*) e a 19 (*chargen*).

1.3.5 Christmas tree packet

Um pacote *Christmas tree* é determinado por possuir todas as opções possíveis para o protocolo que utiliza, marcadas como definidas. É também conhecido como pacote "Kamizake", *nastygram* e *lamp test segment*.

O termo deriva da analogia em que cada opção definida remete à lâmpadas com luzes

com colorações diferenciadas umas das outras, frequentemente fixadas às árvores de Natal e, sendo todas estas opções ativadas, têm se um contexto onde todas as respectivas lâmpadas estariam acesas na árvore. Quando utilizados para *scan*, as *flags* definidas são *FIN*, *URG* e *PSH*.

Pacotes *Christmas tree* são utilizados como método de exploração da natureza da pilha *TCP/IP*, através de seu envio e respectiva espera, ocorrendo assim uma análise das respostas obtidas. Muitos Sistemas Operacionais implementam seus próprios padrões para o protocolo *IP* (INSTITUTE, 1981), podendo apresentar implementações incompletas, de maneiras diferentes umas das outras. Pela observação de como um *host* responde a um pacote inválido, como é o caso do *Christmas tree*, pode-se tirar conclusões sobre o Sistema Operacional do alvo em questão. Versões do *Microsoft Windows*, *BSD/OS*, *HP-UX*, *Cisco IOS*, *MVS*, e *IRIX* mostram comportamentos que diferem do que é determinado pelo padrão estabelecido no *RFC 791*, quando consultados sobre tais pacotes.

Alguns *firewalls*, somente checam as políticas de segurança contra os pacotes que possuem a *flag SYN* definida, ou seja, que iniciaram a conexão de acordo com os padrões. Como os pacote *Christmas tree* dedicados à atividades de *scan* não possuem a flag *SYN* definida, acabam furando estes sistemas de segurança e conseguem alcançar seus alvos.

Um número extenso de pacotes *Christmas tree* podem inclusive, conduzir um ataque *DoS*, aproveitando-se do fato de que tais pacotes exigem muito mais processamento por parte de roteadores e de seus alvos finais, em comparação com pacotes "convencionais".

Pressupondo-se que os pacotes *Christmas tree* não são comumente encontrados nas redes (e tecnicamente, não seguem o padrão estabelecido pelo *RFC 791* (INSTITUTE, 1981)), podem ser facilmente detectados por Sistemas de Detecção de Intrusões e *firewalls* avançados. Da perspectiva de segurança nas redes, os pacotes *Christmas tree* são sempre suspeitos e indicam uma alta probabilidade de que estejam sendo realizas atividades de reconhecimento na rede em questão.

1.3.6 LAND

Trata-se de um ataque muito disseminado em meados de 1997, explorava falhas na implementação do protocolo *TCP/IP*, que na época, alguns Sistemas Operacionais e roteadores eram suscetíveis. Fora projetado inicialmente para afetar somente sistemas que rodassem *Windows95*. (THE..., 2009b)

Consistia bombardear o *host* alvo com pacotes *TCP* maliciosamente manipulados da seguinte maneira:

- Definia-se a flag *SYN* no pacote.
- As portas de entrada e saída eram definidas como sendo a mesma. No sistema para o qual foi projetado o ataque, era comum que se definisse uma das seguintes portas: *113* ou *139*, por fazerem parte das portas disponíveis em seu sistema de compartilhamento de arquivos em rede.
- Os endereços de *IP* do remetente/destinatário eram definidos com o mesmo endereço do alvo.

O efeito obtido era o mal-funcionamento do sistema, pois teoricamente, este cenário não deveria existir sob condições normais. Como consequência de não possuirem um tratamento adequado para tal caso, os sistemas entravam em um estado instável o suficiente para que o único meio de se retomar as atividades, fosse através de um *reboot* forçado nos computadores.

Com seu lançamento, esta falha trouxe à tona uma grande reviravolta nos meios de comunicação utilizados por profissionais de segurança e *hackers*, em especial nos canais de *IRC*, onde sempre foi comum a guerra entre os participantes, no intuito de se tomar o controle dos canais. Além disso, diversos fabricantes de roteadores e *switches*, como a *CISCO*, tiveram que tomar medidas par readequadar em seus produtos, no intuito de suprimir tal falha com sucesso. (TEENAGE..., 2009)

1.3.7 SYN flood

Quando um cliente tenta iniciar uma conexão com um servidor, ambos trocam uma série de mensagens, descritas normalmente como:

1. O cliente requisita uma conexão enviando uma mensagem *SYN* (*sinchronize*) para o servidor.
2. O servidor reconhece a requisição, enviando uma mensagem *SYN-ACK* (*sinchronize-aknowledge*) de volta para o cliente.
3. O cliente responde com uma mensagem *ACK* (*acknowledge*) e então, a conexão é estabelecida.

Este processo é chamado de *TCP three-way handshake*, sendo a base para toda conexão estabelecida ao se utilizar o protocolo *TCP*. A figura 1 mostra o funcionamento do processo descrito:

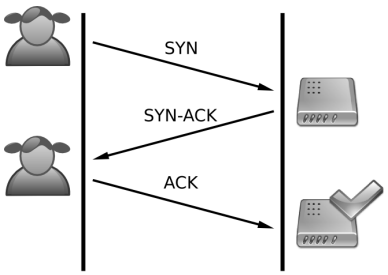

Figura 1: Usuário legítimo, se conectando ao servidor utilizando o *three-way-handshake*.

O *SYN flood* é um ataque bem conhecido e geralmente, não é efetivo contra redes modernas. Seu funcionamento depende do servidor alocar recursos depois do recebimento de um *SYN* e adicionalmente, antes que receba o *ACK*.

Existem dois métodos de ataque, ambos consistindo na premissa de que o servidor não receba o *ACK*:

- Um cliente malicioso pode evitar o envio do último *ACK*, fazendo com que o servidor fique sempre esperando pelo mesmo e tenha os recursos alocados comprometidos.

- O endereço de *IP* que originou o *SYN* enviado pode ser falsificado, fazendo com que o servidor envie o *SYN-ACK* para um *IP* falso, não recebendo então, o último *ACK*.

Em ambos os casos, o servidor irá esperar pelo último *ACK* durante algum tempo, tendo como efeito o mesmo que uma congestão na rede, de modo que este *ACK* restante não seja recebido no tempo certo. A figura 2 mostra o funcionamento do processo descrito:

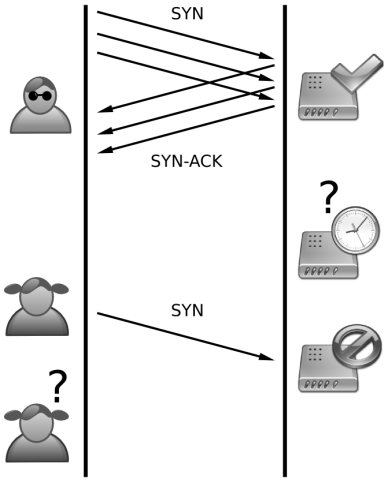

Figura 2: Ataque *SYN flood* é lançado, efetivando o *DoS*.

Se estas conexões incompletas ocuparem recursos no servidor, pode ser possível tomar todos os recursos disponíveis através de uma inundação de requisições *SYN*, daí sua definição. Uma vez que todos os recursos sejam reservados por estas conexões incompletas, sejam elas legítimas ou não, o resultado obtido é a negação de serviço para os demais clientes. Alguns sistemas podem falhar, através de um mal-funcionamento ocasionado por sua arquitetura, ou mesmo travarem, caso alguns recursos do Sistema Operacional estiverem correlacionados aos recursos que foram consumidos com o ataque.

Em meados de 1996, era comum a alocação de recursos para conexões incompletas *TCP*, através da utilização de uma fila, que era em geral, bastante curta, contando com aproximadamente 8 entradas, onde cada uma destas entradas era removida somente quando a conexão era completada ou então, quando expirava seu tempo de vida, que era de aproximadamente 3 minutos. Quando essa fila estava cheia, futuras conexões falhariam, sendo uma tarefa fácil a

de comprometer sistemas com este caráter, pois com apenas 8 pacotes enviados em sequência num intervalo de 3 minutos para cada uma, era possível se atingir uma negação de serviço dispondo do mínimo de recursos.

O *SYN flood* explora esse caráter falho na arquitetura das conexões *TCP* e por isso, para mitigar os riscos existem técnicas que fogem do domínio do protocolo, sendo apenas formas parciais de defesa, que devem ser utilizadas em conjunto para que sejam efetivas. Para que isso seja possível, pode-se empregar as seguintes abordagens:

- *SYN cookies*: O envio do *SYN-ACK* é realizado com números sequenciais cuidadosamente construídos, contendo um *hash* criptográfico, gerado a partir de informações tais como o endereço *IP* do cliente e o número da porta utilizada pela conexão. Quando o cliente responder com um *ACK* normal, essa sequência será inclusa no pacote, que será comparada pelo servidor, no intuito de verificar a autenticidade do mesmo. Adicionalmente, a alocação de memória é realizada pelo servidor durante o envio do terceiro pacote do *three-way-handshake*, e não durante o recebimento do primeiro pacote.

- *RST cookies*: É um método alternativo à utilização dos *SYN cookies*, onde o servidor envia aos clientes um *SYN-ACK* que contenha erros em sua descrição, desta maneira, se o cliente for autêntico, enviará uma mensagem de *RST* ao servidor, que verificará sua autenticidade, não tendo seu endereço *IP* considerado como falso, então o processo de *three-way-handshake* pode se repetir de maneira correta e a conexão é estabelecida com sucesso. Pode causar problemas com máquinas que utilizem o Sistema Operacional *Windows95* ou que estejam protegidas por *firewalls*. Pelo segundo motivo, não é uma técnica amplamente adotada pela comunidade.

- Alocação de micro-blocos: Consiste em não alocar registros completos para cada conexão, sendo que implementações modernas realizam a alocação de apenas 16 *bytes* como forma de prevenção do esgotamento de recursos de memória.

- Modificações na pilha *TCP/IP*: Reduz os riscos do *SYN flood*, através da redução do tempo limite de expiração dos registros alocados no início do *three-way-handshake*.

Uma abordagem comum envolve modificações baseadas em algoritmos que seletivamente descartam conexões de entrada específicas.

1.3.8 Stacheldraht

Traduzindo-se do alemão, significa "arame farpado". Foi escrito por *Random* e ganhou popularidade no verão norte-americano de 1999. Trata-se de um agente que atua em ataques *DoS* distribuídos, ou *DDoS*. Afeta sistemas *Linux* e *Solaris*, sendo capaz de realizar falsificações nos endereços *IP*. Utiliza outros ataques *DoS*, como: *UDP flood*, *ICMP flood*, *TCP SYN flood* e *Smurf attack*. Notavelmente, combina as redes *Trinoo* e *TFN* em seus ataques *DoS distribuídos*. Utiliza criptografia na comunicação entre atacante e "máquinas-escravas", além de possuir atualizações automáticas de seus agentes.

Ao analisar sua arquitetura (DITTRICH, 1999), nota-se que um atacante pode se conectar a vários gerenciadores intermediários. Cada gerenciador é capaz de controlar no máximo 1000 máquinas comprometidas, que por sua vez, irão receber ordens através de um canal seguro de comunicações. Existe um terminal interativo entre atacante e gerenciador, que se assemelha a um terminal de *telnet*, com opções que permitem o controle total do ataque, oferecendo meios para a adição de novos endereços *IP* de vítimas, iniciar ataques, parar ataques, verificar quais estações estão *offline*, entre outras. Para que seja possível o acesso ao terminal dos gerenciadores, uma senha é requisitada ao atacante, que por padrão é denominada "sicken". Essa característica foi determinada através de engenharia reversa. Seu método de criptografia utilizado para o armazenamento da senha é o *crypt()*, muito conhecido nos sistemas *UNIX*, que posteriormente é submetido ao método de cifragem simétrica *Blowfish*, cuja palavra-chave é "authentication", assim como também o são, todas as mensagens que são trocadas na comunicação cliente/servidor.

Assim como se faz com o *Trinoo* e com o *TFN*, o método de instalação é o mesmo que em qualquer sistema *UNIX* comprometido, com opções para que se possa esconder arquivos e programas (por exemplo, diretórios ocultos e *rootkits*). Uma característica que não é presente nem no *Trinoo* e nem no *TFN*, é sua auto-atualização, que se dá através do comando "rpc" da

família *Berkeley* (porta *514 TCP*), utilizando contas roubadas em servidores comprometidos. Então, todos os agentes e controladores são instruídos a buscarem versões mais recentes de si mesmos e substituirem suas versões antigas pelas mais novas que por ventura sejam obtidas.

Seu conjunto de ferramentas utiliza extensivamente pacotes *ICMP Echo Reply*, sendo assim, uma tarefa muito difícil consiguir bloquear seu funcionamento sem que os outros serviços de rede baseados nessa mesma tecnologia não sejam comprometidos. Em grandes redes é impraticável que se realize uma análise de pacotes *ICMP Echo Request* e *ICMP Echo Reply*, afim de se distinguir tráfego legítimo (como o do utilitário *ping*) do tráfego malicioso.

1.3.9 UDP flood attack

UDP é um protocolo que não requer que sejam mantidas conexões e não necessita que seja estabelecido um processo de inicialização para que as conexões ocorram. O *UDP Flood Attack* é um tipo de ataque *DoS*, onde o atacante envia pacotes *UDP* para portas aleatórias no sistema vitimado. Quando esse sistema recebe o pacote *UDP*, irá determinar que aplicação está a sua espera na porta informada. No momento que interpreta os dados e conclui que não existe tal aplicação esperando pelos dados, irá gerar um pacote *ICMP marcado como Destination Unrechacle*, que será enviado para o *host* cujos dados partiram, sendo este classificado como seu remetente. Se forem enviados pacotes *UDP* em uma quantidade suficiente para alvo, e estes forem entregues com sucesso nas portas especificadas, o sistema poderá sofrer uma queda.

Para que o *UDP Flood Attack* seja efetivamente reduzido, deve-se implantar *firewalls* em localizações críticas da rede, no intuito de filtrar os dados indesejados e provenientes de remetentes falsos. Adicionalmente, as ações seguintes podem ser tomadas:

- Desabilitar e filtrar os serviços *chargen* e *echo*, assim como os demais que funcionem com *UDP* e que não sejam utilizados.
- Utilização de *proxys* para prover serviços *UDP* que sejam essenciais, no intuito de serem protegidos contra o mal-uso.

- Monitorar a rede, para identificar os usuários do serviço, assim como o possíveis abusos.

1.4 Malwares

Malwares são definidos como qualquer software desenvolvido com propósito de causar dandos aos computadores, ou de utilizá-los como meio de realizar atividades ilegais. Podem ser classificados de várias maneiras, incluindo como parâmetro, o princípio utilizado para que sejam disseminados, como são executados e/ou as atividades que realizam. Os principais tipos de *malwares* incluem *worms*, *vírus*, *trojans*, *backdoors*, *spywares*, *rootkits* e *spams*.

Worms e *vírus*, são programas de computador que se replicam se a intervenção humana. A diferença básica entre ambos, é que os *vírus* se anexam aos executáveis, tornando-se parte dos mesmos, enquanto que os *worms* são auto-suficientes, não precisando se tornar parte de outro programa para que se repliquem. Adicionalmente, enquanto os *vírus* são projetados para causar problemas no sistema local e são passados através de setores de *boot* dos discos, anexos em *E-Mails* ou mídias removíveis, os *worms* são projetados para explorar o ambiente conectado em rede. Uma vez executado, um *worm* busca ativamente por outros computadores suscetíveis às mesmas falhas às que fora projetado para explorar, ao contrário dos *vírus*, que são programados para buscar por partes específicas do sistema local, mas que sejam vulneráveis à infecção, onde se replicam e mantém o máximo controle que puderem sobre sua execução.

Trojans, ou cavalos de tróia, são *softwares* distribuidos como sendo supostamente legítimos, com o objetivo de incitar os usuários a fazerem o *download* dos mesmos e instalá-los em seus sistemas. Em contraste com os *worms* e *vírus*, os *trojans* não são diretamente auto-replicáveis. São projetados para realizarem atividades perigosas, incluindo a corrupção de arquivos (muitas vezes de maneiras subitas), apagar dados e instalar outros tipos de *malwares*.

Backdoors são programas que fornecem acesso remoto aos sistemas de maneira infiltrada e escondida. Trabalham tipicamente permitindo que indivíduos ou mesmo outros sistemas, que saibam de sua presença, utilizem senhas especiais e/ou ações específicas que passem pelos métodos comuns de autenticação presentes em máquinas remotas, no intuito de obterem

acesso especial às contas administrativas. São projetados para se manterem escondidos, mesmo com inspeções cuidadosamente realizadas.

Spywares são *softwares* que são instalados nos computadores com o propósito de obterem informações sobre o mesmo, incluindo seus usuários e/ou outros computadores conectados na mesma rede aos quais as máquinas hospedeiras se comunicam. Os tipos de informação obtidas, em geral são nomes de usuários e suas senhas, hábitos de navegação na *Internet*, dados financeiros como contas bancárias e cartões de crédito, ou transações secretas. Uma aplicação comum dos *spywares* é mostrar *popups* por meio dos *browsers* utilizados pelos usuários, com base em seus hábitos de navegação na *Web*.

Rootkits são programas secretamente inseridos em computadores, que permitem que os intrusos ganhem acesso à contas administrativas dos sistemas, sendo assim capazes de controlar todos os aspectos dos computadores. Incluem, frequentemente, funções para que seus traços sejam escondidos depois de penetrarem no sistema, através da remoção dos arquivos de *log*, por exemplo. Tipicamente incluem *backdoors*, permitindo que o intruso obtenha posteriormente, acesso facilitado aos sistemas em questão, para que possa efetuar ataques em datas específicas.

Spam é um tipo de *E-Mail* indesejado, enviado em larga escala para os usuários. Apesar das pessoas receberem poucos *spams* por dia e sua grande maioria acabar sendo barrado por sistemas *anti-spam*, pode-se imaginar que este tipo de inconveniente não seja um grande problema a ser tratado. No entando, para os responsáveis pela manutenção da infra-estrutura que hospeda os serviços de *E-Mail*, este é um sério agravante, pois gera um grande volume de dados a serem armazenados, correspondendo, em geral, a mais da metade de todos os *E-Mails* autênticos armazenados, o que ocasiona uma grande carga nos sistemas responsáveis pelo gerenciamento dos mesmos. É comum que contenham certos tipos de *malwares* e que muito de seu conteúdo seja utilizado para efetivar fraudes. As organizações do mundo todo dedicam recursos consideráveis nas tarefas de filtragem e remoção dos *E-Mails* considerados *spam*, tomando os devidos cuidados para que não comprometam *E-Mails* legítimos de sua base de usuários.

Existem várias razões básicas pelas quais os *malwares* são criados. São empregados sen-

timentos de concretização de metas, desejo de mostrar capacidades técnicas, vontades que impulsionam a causar danos ou motivos financeiros. Este último, é provavelmente o mais importante de todos, pois existem grandes incentivos financeiros para que tais atividades sejam desenvolvidas, principalmente de/para organizações criminosas e em menor grau, a indivíduos especialistas em segurança de computadores, contratados para assistirem o desenvolvimento dos mesmos.

O prejuízo causado pelos *malwares* pode ser muito grande. Por exemplo, pode levar à indisponibilidade de computadores e redes inteiras, até que sejam reparados, o que pode ocasionar altos custos financeiros para seus responsáveis. Podem resultar na corrupção ou roubo de dados confidenciais, assim como no roubo de fundos. Adicionalmente, podem resultar em perdas temporárias e/ou danificação de equipamentos que dependem de computadores, que por ventura sejam vitimados.

Danos similares podem resultar advindos de *softwares* que sejam mal-projetados/escritos que, assim como os malwares, têm presença muito comum em meio às organizações. No entanto, a distinção entre ambos é dada subtamente. Enquanto os *malwares* são criados inteiramente, ou principalmente, no intuito de causarem injúrias ou beneficiarem seus criadores, às custas de outros indivíduos, os danos são consequência secundária no caso de *softwares* defeituosos, pois estes raramente são seu motivo de existir.

Existem inúmeros passos que os usuários de computadores podem tomar para que as chances de se infectarem por *malwares* sejam minimizadas. São incluidas como boas práticas, a utilização de *softwares* legítimos e seguros, a provisão de locais físicos que sejam seguros aos computadores e às redes, a adoção de políticas obrigatórias na definição de senhas seguras, implantação de *firewalls*, utilização de programas de detecção de *malwares*, evitar abrir *E-Mails* com anexos cujos remetentes sejam desconhecidos, evitar o *download* de programas dúbios e evitar o uso de contas administrativas para tarefas corriqueiras, sendo que estas devem ser utilizadas somente quando necessário.

Alguns tipos de Sistemas Operacionais e aplicações são muito mais resistentes aos *malwares* que outros, particularmente aqueles baseados no *Kernel* do *Linux* e outros Sistemas Ope-

racionais *Unix-like*. Esse fato se deve por terem sido projetados para evoluir ao longo do tempo, levando-se em consideração a segurança como fator essencial, ao invés de tomar uma abordagem na qual as tentativas de se adicionar camadas de segurança fossem implantadas posteriormente.

Também deve-se levar atentar ao fato de que são raras as tentativas de se projetar *malwares* para esses sistemas. Uma razão que justifica isso é que o número de computadores que os utiliza, ainda é a minoria, ocasionando em um menor número possíveis alvos para os criminosos, sendo menos interessante desprender esforços para atingir sua gama de usuários. O fato de serem melhor projetados quanto os requisitos de segurança, dificulta a produção de *malwares* que sejam eficientes e bem sucedidos nas investidas de ataque.

Devido sua importância, os *malwares* merecem um estudo minucioso, além do estabelecimento de sistemas capazes de detectar novas variantes destes, já que a manutenção do bom funcionamento dos computadores e de suas redes, é de suma importância para todas as organizações do mundo.

2 Honeypots

O primeiro passo para se entender o que são os *Honeypots*, através de sua definição. Diferente dos *firewalls* ou de Sistemas de Detecção de Intrusão, os *Honeypots* não resolvem um problema em especial. Ao invés disso, são ferramentas altamente flexíveis, com diferentes propósitos, complexidades envolvidas e cujos mecanismos de funcionamento se dão das mais variadas maneiras. Podem realizar atividades que vão desde detectar ataques criptografados à redes *IPv6* até capturar a mais recente fraude financeira envolvendo cartões de crédito. Essa é a flexibilidade que os torna ferramentas tão poderosas. (SPITZNER, 2001)

Honeypots são recursos de sistemas de informação, cujos valores residem no acesso não-autorizado ou mesmo o uso ilícito dos mesmos.

Esta definição geral cobre todas as diferentes manifestações de *Honeypots*. Sua natureza é exploratória, baseada em atividades realizadas por indivíduos mal-intencionados, com relação aos sistemas implantados. Conceitualmente, todos os Honeypots trabalham da mesma maneira, não tendo recursos disponíveis publicamente, sendo possível seu acesso, somente através do acesso não-autorizado, realizado através da quebra da segurança envolvida nos sistemas. (BAUMANN; PLATTNER, 2002)

Os *Honeypots* não têm valor real em meio aos ambientes de produção e por isso, não deveriam interagir com nenhuma outra forma de recurso, como outros sistemas ou usuários e, sendo assim, toda e qualquer atividade direcionada aos mesmos deve ser considerada suspeita.

Isso nos remete à situação em que qualquer tipo de interação com os *Honeypots* implantados, pode ser classificada como maliciosa e ilegal, pois toda tentativa de conexão muito provavelmente será relacionada à *scans*, sondagens, ataques ou ao comprometimento do sistema. A abordagem que este tipo de ferramenta toma é realmente simples, no entanto essa é

a chave para seu sucesso como grande ferramenta de valor para os profissionais de segurança.

Vantagens: são sistemas conceitualmente simples, o que os torna incrivelmente poderosos.

- Pequenos conjuntos de dados, de grande valor: Os *Honeypots* capturam pequenas quantidades de informações. Ao invés de registrar em logs, *Giga-bytes* de dados por dia, em geral, são capturados apenas poucos *Mega-bytes* de dados diários. É comum se encontrar sistemas em produção que geram milhares de alertas por dia, devido à quantidade de acessos que estes têm, provenientes de seus usuários, por estarem realmente expostos e interagindo com uma alta quantidade de outros sistemas. Já os *Honeypots* podem registrar poucas dezenas de alertas por dia, mas que representam muito mais valor que no caso anterior, já que todos estes alertas capturados representam atividades ilegais. Isso facilita o trabalho de interpretação dos dados, tanto em questões de recursos quanto financeiramente, uma vez que o universo de informações a serem analisadas é significantemente menor quando comparado àquele gerado por um sistema real, que se encontra em plena produção.

- Novas ferramentas e táticas: Os *Honeypots* são projetados para capturar toda e qualquer informação relacionada às atividades que envolvam interação com o mesmo, podendo revelar novas ferramentas e táticas empregadas pelos atacantes em suas investidas.

- Encriptação e *IPv6*: Diferente de outras tecnologias, tais como Sistemas de Detecção de Intrusão, pois trabalham bem em ambientes criptografados ou *IPv6*. Não importa o que seja lançado sobre os *Honeypots*, eles irão detectar e capturar tais informações.

- Informação: *Honeypots* coletam todas as informações que chegam até eles e poucas ferramentas podem registrar com tanta eficácia este tipo de captura, assim como os *Honeypots* fazem.

- Simplicidade: São conceitualmente simples. Não existem algorítmos complexos tanto no processo de desenvolvimento quanto no processo de implantação, não há a necessidade de se manter tabelas de estados ou assinaturas que devam ser atualizadas periodicamente. Quanto mais simples forem as tecnologias envolvidas, menos suscetíveis a falhas

ou má-configurações, estas serão.

Desvantagens: Assim como qualquer tecnologia, os *Honeypots* possuem seus pontos fracos. Isso porque eles não substituem as tecnologias atuais, mas trabalham em conjunto com as mesmas.

- Visão limitada: Os *Honeypots* podem rastrear e capturar atividades que interagem diretamente com eles e por isso, não irão capturar ataques direcionados a outros sistemas, a menos que, o atacante ou a ameaça, interaja também, de alguma maneira, com os *Honeypots*.

- Risco: Todas as tecnologias de segurança envolvem riscos. Assim como *firewalls* possuem o risco de serem penetrados, mecanismos de encriptação correm o risco de serem quebrados e Sistemas de Detecção de Intrusão podem falhar na detecção de ataques, os *Honeypots* não são infalíveis. Especificamente, os *Honeypots* estão propensos a serem tomados por atacantes que obtenham sucesso ao se penetrar nos sistemas hospedeiros, por eventuais falhas de configuração ou por conseguirem escalar de maneira adequada seus privilégios. Dependendo do tipo de *Honeypot* implantado, este pode oferecer tanto quanto, ou até menos risco, que um Sistema de Detecção de Intrusão falho, enquanto que existem determinados tipos de *Honeypots* que oferecem riscos reais às redes, uma vez que sejam comprometidos.

2.1 Classificação dos Honeypots

Os *Honeypots* são classificados de acordo com seus objetivos e o grau de interação que oferecem aos atacantes. Quanto aos objetivos, existem *Honeypots* de Produção e *Honeypots* de Pesquisa, onde o primeiro é utilizado para mitigar riscos nas redes ativas de organizações e o segundo é empregado na descoberta de novas estratégias de ataque e exploração de falhas de segurança. (JONES; ROMNEY, 2004)

Quanto ao grau de interação oferecido aos atacantes, temos *Honeypots* de baixa-interação e de alta-interação, onde o primeiro oferece níveis mínimos de interatividade com seus ata-

cantes, na maioria das vezes, emulando Sistemas Operacionais e serviços comprometidos, ao passo que o segundo é composto por um sistema operacional real, onde são introduzidas falhas reais e uma vez no controle de uma máquina comprometida, o atacante pode executar qualquer ação, como em um servidor qualquer, que por ventura tenha sido tomado. (SPITZNER, 2003a)

A tabela 2 mostra um comparativo de custo de implantação, entre as tecnologias em *Honeypots*, de acordo com seu tipo e nível de interação:

Tabela 2: Tecnologias em *Honeypots* comparadas por Custo de implantação

Tipo de Honeypot	**Nível de interação**	**Custo de implantação**
Pesquisa	Baixo	Baixo
Pesquisa	Alto	Alto
Produção	Baixo	Baixo
Produção	Alto	Alto

2.1.1 Honeypots de Produção

O principal intuito dos *Honeypots* de Produção é fornecer métricas que possam direcionar, de maneira mais efetiva, os esforços dos responsáveis pela administração de redes, afim de obterem níveis de segurança mais finos. Esse tipo de trabalho, é realizado através da coleta e análise dos dados de utilização, provenientes dos serviços oferecidos pelos *Honeypots* de produção. São relativamente fáceis de se utilizar e podem operar camuflados junto à topologia-padrão de rede implantada em uma organização.

Em geral, os principais dados capturados com este tipo de *Honeypot* são pertinentes aos tipos de ataques mais recorrentes que a rede em questão sofre, qual é a localização geográfica dos atacantes e quais são os meios utilizados para tal.

Pode-se utilizar esse tipo de *Honeypot* no intuito de se camuflar servidores de produção que têm grande valor nas organizações. Os atacantes têm em sua maioria, o intuito de vasculhar as redes em busca de máquinas que estejam suscetíveis a falhas de segurança conhecidas e,

uma vez encontradas, o processo de exploração dessas falhas permite que seu controle seja tomado sob circunstâncias de sucesso. Sendo assim, é comum que se introduza em meio às redes, máquinas contendo falhas propositais, no intuito de distrair os atacantes. Uma vez identificadas, os esforços para a tomada do controle será quase que unicamente direcionado às mesmas, resguardando as demais de possíveis tentativas de corrupção.

Mesmo que sejam implantados *Honeypots* de baixa interação, os atacantes terão muito tempo investido para que seu sucesso se resuma em pouco resultado, uma vez que o controle pleno de máquinas com esse caráter não é possível. Caso seja escolhida a implantação de um *Honeypot* de alta-interação, medidas de contenção devem ser tomadas, como a implantação de um *firewall* que seja capaz de controlar a saída, *outbound*, dos dados, evitando-se assim, que a partir de uma estação tomada, seja lançado, por exemplo, um ataque de *DoS* em alvos de interesse do atacante.

Uma vez identificados padrões de ataques sofridos pela rede, têm-se condições de elaborar planos de ação, visando mitigar os riscos. Estes planos podem envolver desde melhorias nos *firewalls*, políticas de acessos mais elaboradas, até em mudanças na utilização de senhas, por parte dos usuários, afinal, muitas vezes são identificados ataques por dicionário. Pode-se também incrementar a segurança de uma rede vulnerável com a adição de camadas adicionais de acesso a servidores com grande nível de importância nas organizações, atentando-se ao fato de realizar, progressivamente, filtragens nos dados até que estes cheguem em seus destinos.

2.1.2 Honeypots de Pesquisa

Os Honeypots de pesquisa, visam identificar tanto novos ataques automatizados quanto novas práticas de ataque empregadas pela comunidade *hacker*. Não têm o intuito, assim como os *Honeypots* de Produção, de identificar ataques corriqueiros, e sua finalidade é puramente voltada para a identificação de novos formatos em que possam se enquadrar ameaças desconhecidas ou que tenham sofrido alguma forma de mutação.

São muito empregados na busca por fontes de disseminação de *spam*, uma vez que é possível simular diferentes máquinas rodando diversos sistemas operacionais e serviços, sendo

assim, pode-se montar uma infra-estrutura virtual para interceptar este tipo de atividade e tráfego. Pode revelar inúmeros hosts infectados por *worms*, que são cada vez mais presentes nos mecanismos de disseminação de *spam*, o que pode ser definitivo, ao se tentar traçar a topologia total deste problema.

2.1.3 Honeypots de Baixa-interação

Honeypots de baixa-interação são compostos por softwares que emulam serviços e sistemas operacionais nas máquinas hospedeiras, assim como por exemplo, faz o *Honeyd*. Seu principal intuito é capturar informações dos atacantes, sem que existam grandes margens para que este se torne um intruso de sucesso, ou seja, consiga tomar controle do sistema hospedeiro.

Tal abordagem, é conseguida quando se têm, por exemplo, uma solução *Honeypot* que escuta todas as portas *TCP* e *UDP* por algum tipo de contato externo. Sendo assim, quando existe alguma comunicação, as informações são guardadas, mas sem que exista realmente, um serviço conhecido, como *FTP*, *Telnet* ou *SSH* na porta requisitada. Quando um serviço é emulado, por exemplo, um servidor de *FTP*, pode inclusive exibir *banners* nas tentativas de *login* e posteriormente, serem capturados tanto o *login* quanto a senha do atacante, assim como qualquer forma de interação que este tenha com o suposto serviço.

Pode-se escolher perfis de atacantes, através de serviços emulados, por exemplo, implantando-se um pseudo-*Webserver*. Desta maneira, os ataques mais frequentes capturados, serão aqueles destinados a explorar falhas pertinentes ao daemon escolhido. Assim, cria-se um público-alvo específico de estudo.

O filtro pode ocorrer também, com relação ao Sistema Operacional, uma vez que é possível emular comportamentos mediante a *fingerprints* extraídos por ferramentas de *scan* ativas, como o *Nmap* (NMAP..., 2009). Os tempos de resposta variam de acordo com o Sistema Operacional selecionado e para que a situação seja mais próxima da realidade de um servidor passível de ataque, escolhe-se serviços que funcionem bem com o ambiente que se deseja emular: um *Webserver* típico em ambientes de produção seria por exemplo, dotado de um *Kernel* do *Linux 2.6.x* e que tenha como *daemon* para servir páginas *Web*, o *Apache 2.2.x* (HTTP...,

2009). Dessa maneira é possível restringir cada vez mais o perfil de possíveis atacantes e se estudar as técnicas utilizadas em suas investidas. O tempo de resposta pode ser manipulado maliciosamente pelo *Honeypot* em questão, no intuito de enganar *fingerprints* ativos de Sistemas Operacionais, através de técnicas de emulação que trabalham como implementações específicas da Pilha *TCP/IP*, presentes nos referidos sistemas.

O nível de sofisticação envolvido nas respostas está intrinsicamente relacionado ao tipo de *Honeypot* que se utiliza e podem variar de entre os ambientes escolhidos, oferecendo taxas maiores ou menores de falsos-positivos para os atacantes.

2.1.4 Honeypots de Alta-interação

As *Honeynets* são o exemplo principal de *Honeypots* de alta-interação. Não são compostas por um produto em especial, tampouco por um software específico. Tratam-se de uma arquitetura especial, uma rede inteira de computadores especialmente projetados para sofrerem ataques. A ideia é que se tenha uma arquitetura que crie uma rede altamente controlável, onde todas as atividades sejam monitoradas. Nesta rede, são postas as vítimas propositais: computadores reais, rodando Sistemas Operacionais e aplicações reais.

Os atacantes encontram estas máquinas, lançam esforços para tomá-las e com sucesso, conseguem devido sua iniciativa. Quando se tornam instrusos, não imaginam estar sendo monitorados, por estarem em meio a uma *Honeynet*. Toda a atividade, desde sessões *SSH* encriptadas, *E-Mails* trocados/violados e uploads de arquivos, são capturados sem seu consentimento.

Isso é possível através da inserção de módulos do *Kernel*, nos sistemas vulneráveis, que capturam as ações tomadas pelos intrusos. Ao mesmo tempo, a *Honeynet* controla as atividades dos atacantes, sendo essa tarefa realizada pelo *gateway Honeywall*, que permite tráfego de entrada, *inbound*, para os sistemas-vítimas e ao mesmo tempo controla todo o tráfego de saída, *outbound*, usando tecnologias de prevenção de intrusão, tais como quando se emprega o serviço *Hogwash* (HOGWASH. . . , 2009). Isso permite ao atacante que tenha total flexibilidade ao interagir com o sistema comprometido, ao passo que está limitado no sentido de não

conseguir ameaçar diretamente outros computadores que não façam parte da *Honeynet*.

2.2 Boas práticas

Lidando com servidores, a implantação de processos não pode ser levada como uma tarefa banal, assim como é comumente feito em um ambiente de desenvolvimento. Se o objetivo for implantar *Honeypots*, em meio ao ambiente de produção, é importante que seja levada em consideração a segurança do sistema hospedeiro, por isso, existem práticas envolvidas no processo de *deploy* que o tornam mais seguro e consequentemente, menos suscetível a falhas.

Como uma das características principais de um *Honeypot* é atrair a atenção de atacantes em potencial, uma vez que este seja alvo de uma investida de sucesso, pode-se tomar o controle do sistema hospedeiro, caso se trate de um *Honeypot* de alta-interação. Para evitar que este mesmo sistema não seja utilizado para comprometer mais computadores, ou mesmo para disseminar atividades ilegais, deve-se apertar a segurança tanto quanto seja possível, e uma das maneiras mais eficientes consiste em limitar o escopo dos serviços, ou aplicações, que são postos em execução. Isso é possível graças à uma ferramenta desenvolvida por *Niels Provos*, um dos grandes estudiosos no ramo de segurança de redes e *Honeypots*. Uma breve visão sobre sua utilização será exposta e como fator opcional para o aumento da segurança em testes coleta de informações, utilizando-se *Honeypots*, recomenda-se que os ambientes sejam preparados com base neste utilitário chamado *Systrace*.

2.2.1 Protegendo serviços

Trata-se de uma ferramenta que força políticas de segurança relacionadas às chamadas de sistema(PROVOS, 2003). Isso faz com que um programa tenha seu acesso limitado, de acordo com o tipo de ação que pretende realizar no sistema. As operações não autorizadas disparam alarmes, permitindo ao usuário refinar a política de segurança implantada.

Ao lidar com aplicações complexas, é árdua a tarefa de se definir todas as chamadas de sistema que podem ser autorizadas. É impraticável o ato de prever todos os seus ciclos de

execução, aliados aos privilégios requeridos em cada instante. Para isso, o *Systrace* (PROVOS, 2009) realiza o monitoramento do executável, notificando o usuário sobre cada chamada de sistema cujos privilégios necessários sejam superiores aos que foram concedidos inicialmente. Depois de ser recebida a avaliação, é criada uma regra que será englobada, posteriormente, à sua política de segurança. No entanto, devido ao comportamento observado não ser totalmente previsível, é comum que esporadicamente sejam disparados alertas de segurança relacionados às chamadas de sistema, sendo que com o tempo, a tendência de uso determina que estes alertas se tornem cada vez mais raros, até que a política de segurança seja refinada o suficiente, de modo que, durante a execução do binário monitorado não sejam gerados mais alarmes durante todo o seu ciclo de vida.

As políticas de segurança podem ser aprendidas automaticamente, sendo estas aplicáveis a ambientes de testes, sem a necessidade mandatória de interferência ou pós-processamento manual. É desejável que se construa ambientes seguros em servidores, por exemplo, onde um serviço possa rodar isolado das demais, garantindo que qualquer comportamento anormal só afete a ele mesmo, não interferindo no sistema como um todo. Isso é obtido, empregando-se varias técnicas de *sandboxing*, e o *Systrace* pode ser utilizado como peça fundamental no processo de obtenção de um ambiente próximo do idealizado.

O encapsulamento de binários em ambientes *sandbox*, torna-se extremamente útil quando se deve lidar com exemplares cujo código-fonte não é disponibilizado, ou mesmo quando tratam-se de fragmentos advindos de grandes projetos de código-fonte aberto e que, devido sua grande extensão e complexidade, não têm todo o seu comportamento devidamente assegurado por técnicas que envolvam suites de testes unitários e cobertura de código efetivamente implantadas. As motivações para a utilização do *Systrace* extendem-se também aos casos onde é imprescindível a realização de auditorias nos sistemas, seja por fins de pesquisa ou pelo fato de que sua execução pode ser classificada como parte de um conjunto de ações críticas, perante o Sistema Operacional e à estrutura dependente deste, como serviços em execução e informações sensíveis hospedadas.

As chamadas de sistemas podem ser re-escritas dinamicamente, provendo ao Sistema

Operacional, um caráter semelhante ao encontrado em ambientes chroot virtuais, além de previnir race conditions na interpretação dos argumentos passados(R. *et al.*, 1999).

É possível monitorar máquinas remotas e gerar alertas em uma central, indicando operações que a política de segurança existentes não cobrem. Sendo assim, pode-se detectar intrusões e também realizar a prevenção das mesmas, mitigando riscos potenciais, em ambientes de produção, através de um processo contínuo de monitoramento e correções.

Aplicações ordinárias podem necessitar de privilégios maiores com relação aos que se deseja conceder. O *Systrace* pode interferir no modo em que executam, permitindo delimitar pontos onde os privilégios sejam concedidos, sem que seu comportamento seja alterado. De um modo geral, os privilégios são escalados somente no instante em que as chamadas de sistema associadas necessitam, desde que sejam denominadas confiáveis. Isso elimina a necessidade de se ter ciclos completos de execução marcados como privilegiados, por exemplo: através do uso das *flags setuid* e *setgid* (que poderão ser removidas completamente dos binários em questão) ou com a utilização de usuários privilegiados no sistema destinados exclusivamente para a execução de aplicações específicas.

2.3 Organizações envolvidas

A comunidade conta com várias organizações, responsáveis pelo desenvolvimento das tecnologias em *Honeypots* e identificação de ameaças e tendências de ataques globais. Para tal, as organizações contam com o apoio de instituições parceiras, que em geral compõe-se principalmente por Universidades e Empresas que reconhecem a importância deste tipo de trabalho.

São apresentadas a seguir, algumas das organizações mais influentes so setor mundial, no que compete à pesquisa e desenvolvimento de tecnologias em *Honeypots*. Existe também, a preocupação em apresentar duas iniciativas nacionais relacionadas, que também exercem um papel extremamente importante perante à comunidade mundial e são ótimos pontos de partida para organizações estabelecidas em território brasileiro e que se interessam em estabelecer

parceirias no combate às atividades ilegais, realizadas na *Internet*.

2.3.1 Honeynet Project

Fundada em 1999, o *Honeypot Project* é uma organização internacional, de pesquisa, dedicada à melhoria da segurança da *Internet*, sem custo algum para o público. Seus voluntários estão engajados com o movimento *Open Source* e visam trazer valor à comunidade por meio de medidas como o alerta sobre ameaças e vulnerabilidades presentes na *Internet* atualmente.

Muitas organizações e indivíduos não sabem que são alvos de ataques, tão pouco sabem quem os ataca, quais são seus motivos e como isso é realizado. Provendo este tipo de informação, é possível que se entenda essa dinâmica e então sejam tomadas medidas para que as ameaças sejam mitigadas. Essas informações são disponibilizadas através de uma série de *whitepapers* chamados *Know your Enemy*. (KNOW..., 2004) (KNOW..., 2002)

São compartilhados, com os especialistas em segurança, dados que ajudam no processo de defesa de seus recursos. Historicamente, as informações sobre atacantes eram limitadas às ferramentas utilizadas pelos mesmos. Como diferencial, são oferecidos também os motivos de ataque, a maneira como se comunicam, as datas de ataques e as ações tomadas quando os sistemas são comprometidos. Além da série *Know your Enemy*, existem desafios lançados chamados *Scan of the Month*, um agregador em conjunto ao blog oficial (HONEYNET..., 2009) e várias ferramentas (THE..., 2009a), responsáveis por muitas dessas informações adicionais.

2.3.2 Leurrecom

Trata-se de um projeto que visa a implantação de uma plataforma em *Honeypots* distribuída e de larga escala. São instalados sensores em diversas partes do mundo, onde intuito é a geração e análise posterior dos logs de atividades. (LEITA *et al.*, 2008)

Esse arranjo abrange hoje, em torno de 30 países distribuídos pelos 5 continentes, o que fornece uma medida muito próxima da realidade no que tange ao comportamento das atividades

ilegais ao redor do globo. Os dados são armazenados em bancos de dados sofisticados, com suporte à *SQL* e acompanham dados enriquecidos de informações contextuais, tais como localização geográfica dos atacantes, *fingerprint* de seus sistemas operacionais e busca reversa por seus endereços de *DNS*.

É oferecida aos seus parceiros, uma interface intuitiva de consulta aos dados, facilitando ainda mais as tarefas gerenciais através da possível elaboração de relatórios de alto-nível. Tais informações podem revelar qual é a frequência com que os ataques são lançados de diversos pontos, podendo ser comparados uns aos outros em termos de ferramentas utilizadas nos ataques, quem são especificamente os atacantes e quais foram as técnicas empregadas pelos mesmos em suas investidas.

Para se tornar parceira, uma instituição precisa somente concordar em hospedar em suas dependências de infra-estrutura, umas das plataformas oferecidas pelo projeto. Como consequência, é concedido o direito de acesso a um banco de dados que fora alimentado durante mais de 3 anos com atividades sendo registradas. É também requerido que se assine um termo de não-revelação pública dos nomes de outros parceiros e das identidades dos atacantes.

2.3.3 Honeynet.BR

Devido ao crescente interesse no desenvolvimento de tecnologias em *Honeytraps* por parte da comunidade e dos estudos dirigidos às *Honeynets*, liderados principalmente por *Lance Spitzner*, o grupo brasileiro *Honeynet.BR*, membro da *Honeynet Research Alliance* desde Junho de 2002, investiu esforços na criação de uma solução robusta em *Honeypots*, de modo que fosse efetiva a sua implantação em instituições de pesquisa e que pudesse revelar, em conjunto com muitas outras instituições, o comportamento das ameaças em território nacional.

Os dados são integrados com outras instituições credenciadas internacionalmente e com isso é possível observar as atividades em âmbito global, através de um projeto em comum. Seus idealizadores no Brasil pertencem ao Instituto Nacional de Pesquisas Espaciais (*INPE*), em conjunto com o *CERT.br* . (HOEPERS *et al.*, 2007)

Topologia

A arquitetura da rede *Honeynet.BR* consiste basicamente em uma *Honeynet GenII* com algumas modificações. Sua rede é composta de um segmento de rede administrativo e uma *Honeynet*.

A rede administrativa é composta por três componentes principais: um *Firewall*, um *Sistema de Detecção de Intrusões* e uma máquina para *Computação Forense*.

O *Firewall* consiste em um sistema *OpenBSD* operando em modo *bridge*, rodando *pf* (*OpenBSD Packet Filter*), logando todo o tráfego e limitando a banda de saída.

Um programa foi desenvolvido para limitar a atividade do intruso, interagindo com o *pf* para dinamicamente, mudar as regras de *firewall*. Esse programa, chamado *sessionlimit*, monitora o tráfego de saída, visando detectar e parar atividades maliciosas, tais como *portscans* provenientes da *honeynet*.

O Sistema de Detecção de Intrusão opera com as duas interfaces de rede, uma conectada à rede administrativa e a outra com a honeynet. A última, captura todo o tráfego e não possui endereço de *IP*. Essa máquina gera alertas e envia emails diários com sumários para todos os membros do projeto.

A máquina utilizada para computação forense, é utilizada para armazenar e analizar imagens das partições provenientes das máquinas que pertencem à *honeynet*, além das ferramentas deixadas pelos intrusos.

A figura 3 mostra esquematicamente, a topologia de rede que abordada.

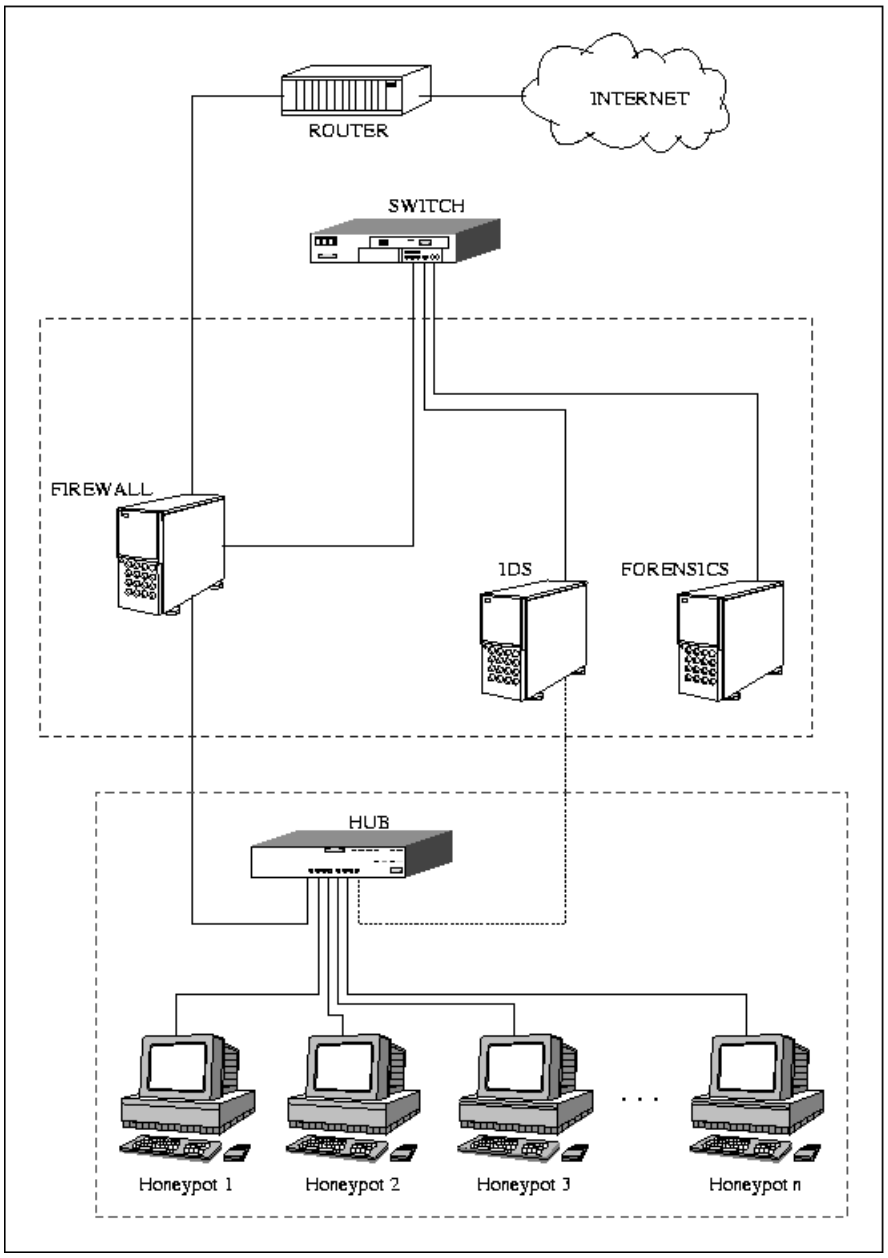

Figura 3: Topologia da *Honeynet*.

Honeynet

A *honeynet* é composta por várias máquinas rodando diferentes sistemas operacionais e serviços, são dotadas de utilitários para logar as atividades e o tráfego interno da *honeynet* é capturado pelo Sistema de Detecção de Intrusões.

As honeynets permitem que sejam observadas as ações e técnicas utilizadas pelos atacan-

tes, além de poderem capturar as ferramentas que estão sendo distribuídas pela comunidade blackhat. Em certas ocasiões, é possível também, monitorar eventuais conversas entre os atacantes, que podem guiar a um melhor entendimento sobre as suas motivações e perfis psicológicos.

Honeytraps

Os *honeytraps*, são em geral, divididos em duas categorias: passivos e ativos. Os *honeytraps* passivos são sistemas de baixa interatividade, ou seja, permitem somente uma baixa-interatividade e nenhum acesso aos sistemas hospedeiros. Exemplos de honeytraps são os da família *Back Officer*, *Deception Toolkit*, *Scpecter* e *Honeyd*. São extremamente úteis na detecção de tendências de ataques, atividades de *scan* e servem como agentes para o disparo de alertas prematuros ou como sistemas de detecção de atividades não-autorizadas em ambientes de produção.

Os *honeytraps* ativos permitem, em geral, interações completas entre os atacantes e os *hosts*, que podem estar conectados a uma *LAN*, que é o caso das *honeynets*, ou podem ser emulados em um único sistema, com uso de mecanismos de virtualização. No último caso, podem ser utilizadas aplicações como *VMWare*, *UML* (*User-Mode-Linux*) ou *Decoy Server* (formalmente conhecido por *ManTrap*).

2.3.4 Project Honey Pot

O *Project Honey Pot* é o primeiro e o único sistema distribuído para a identificação de *spammers* e os *spambots* que utilizam para obter endereços de *E-Mail* dos *websites*. Integrando-se ao *Project Honey Pot*, pode-se instalar endereços de *E-Mail* com identificadores especiais relacionados ao momento de acesso de cada visitante do site em que foi implantado, o que os relaciona com seu respectivo endereço de *IP*. Se algum destes endereços começar a receber *E-Mails*, não só é gerado um alerta sobre o spam recebido, como também é indicado o momento exato em que o endereço foi capturado e o *IP* referente ao infrator.

Para participarem do *Project Honey Pot*, os webmasters precisam somente instalar o

software provido pelo projeto em uma localização escolhida, em seus *websites*. O restante do processo é conduzido pela central responsável pela geração dos *E-Mails* a um baixo custo de banda, não sendo assim, nocivo em termos de geração de tráfego adicional em níveis prejudiciais.

Esses dados gerados pelo *website* são coletados, processados e compartilhados com seus responsáveis. Existem parcerias estabelecidas com promotorias de justiça responsáveis pelo enquadramento das atividades reportadas como sendo ilegais, assim como pela punição dos *spammers* identificados.

Adicionalmente, as mensagens de *E-Mail* geradas pelos spammers são capturadas, analisadas e compartilhadas com desenvolvedores e pesquisadores na área *anti-spam*. Os dados coletados pelo *Project Honey Pot* irão ajudar a construir uma nova geração de ferramentas em software *anti-spam*.

O *Project Honey Pot* foi criado por *Unspam Technologies, Inc*, uma companhia *anti-spam* com a missão singular de ajudar no projeto e aplicação de leis efetivas *anti-spam*.

2.3.5 Brazilian Honeypots Alliance

O objetivo deste projeto é aumentar a capacidade de detecção de incidentes, correlacionar eventos e analisar as tendências no espaço de Internet brasileiro. Para que isso seja possível, o projeto foca em definir uma rede de máquinas comprometidas com *Honeypots* de baixa-interação (*Honeyd*), cobrindo a maioria do espaço de endereços de IPs brasileiros, construir um sistema de análise de dados que permita o estudo das tendências de ataques e suas correlações e trabalhar com os times de Resposta a Incidentes de Segurança de Computadores, para que a informação seja disseminada.

A figura 4 mostra a distribuição dos *Honeypots* pelo território brasileiro, relacionando as instituições colaboradoras, que são enumeradas:

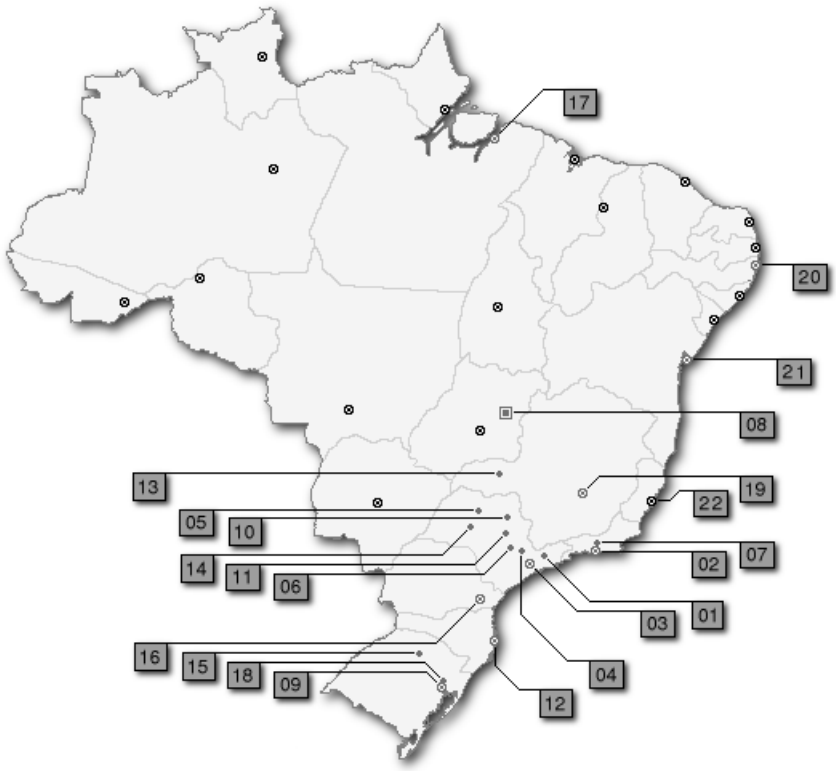

Figura 4: Distribuição dos Honeypots pelo território nacional.

A tabela 3 relaciona as instituições com sua respectiva enumeração, apresentada na figura anterior:

Tabela 3: Distribuição de *Honeypots* em instituições brasileiras.

#	**Cidade**	**Instituição**
01	São José dos Campos	INPE, ITA
02	Rio de Janeiro	BNDES, CBPF, Embratel, Fiocruz, Furnas, PUC-RIO, RedeRio
03	São Paulo	ANSP, Banco Real, CERT.br, Diveo, Durand, TIVIT, UNESP, UOL, USP
04	Campinas	CenPRA, ITAL, UNICAMP
05	São José do Rio Preto	UNESP
06	Piracicaba	USP
07	Petrópolis	LNCC
08	Brasília	Banco do Brasil, Brasil Telecom, CTIR Gov, Ministério da Justiça, TCU
09	Porto Alegre	CERT-RS
10	Ribeirão Preto	USP
11	São Carlos	USP
12	Florianópolis	UFSC DAS
13	Uberlândia	CTBC Telecom
14	Lins	FPTE
15	Passo Fundo	UPF
16	Curitiba	Onda, PoP-PR, PUCPR
17	Belém	UFPA
18	São Leopoldo	Unisinos
19	Belo Horizonte	CSIRT PoP-MG, Diveo
20	Recife	EMPREL
21	Salvador	UFBA
22	Vitória	PoP-ES

2.4 Implicações Legais

Os *Honeypots* foram utilizados peloa comunidade de especialistas em segurança por anos, até a presente dada. Foram aplicados com uma variedade imensa de propósitos e hoje em dia, seu uso comum é a obtenção e análise de informações relacionadas ao comportamento das redes. Desde o princípio de sua utilização, geraram discussões perante a comunidade sobre suas implicações legais. No entanto, ao longo dos anos esse debate aparentemente se extinguiu e a maioria das organizações o consideram como a menor das implicações. (SPITZNER, 2003b)

A maioria dos produtores de software voltados para segurança, como *Symantec*, *MCaffe*, *Websense* e *Sophs*, assim como muitos *CERTs* de vários países, implantam ativamente esses mecanismos para a melhoria da segurança da Internet. Os *Honeypots* não têm por natureza, incitar a invasão ou práticas ilegais, assim como não força os atacantes a cometerem delitos aos quais não estejam propensos naturalmente. São sistemas sem valor, quando analisados em conjunto aos demais presentes em um ambiente de produção, por isso, teoricamente nunca deveriam interagir com outros sistemas ou usuários. Adicionalmente, estes sistemas, a priori, não possuem contas abertas e por isso, a única maneira de se obter acesso aos mesmos, é quebrando sua segurança.

Uma visão não técnica sobre o assunto, seria descrita facilmente ao se comparar o conceito de invasão em *Honeypots* ao roubo de carros: Suponha que exista um estacionamento de carros, cuja lotação encontra-se em um nível máximo, ou mesmo perto disso, analogamente aos espaços de *IP* preenchidos por vários computadores. Cada qual é provido de trancas funcionais e alguns destes possuem câmeras em seus interiores (acionadas via sensores de movimento), representando os *firewalls* e *Honeypots* respectivamente. Ninguém, além dos legítimos donos dos carros, deveriam obter acesso ao interior dos mesmos, pois estes encontram-se seguros, através de suas trancas, e somente acessos autorizados são aceitos. No entanto, se alguém mal-intencionado decidir praticar um roubo, ou qualquer atividade ilegal, envolvendo o acesso não-autorizado ao interior do veículo, este terá que quebrar a segurança empregada no mesmo, arrombando tais trancas ou utilizando técnicas que não requerem as chaves do seu dono original para isso, como o estilhaçamento de seus vidros. Se efetivada com sucesso, essa invasão ao

veículo pode ser monitorada através da câmera instalada. O que se aplica totalmente aos mecanismos de captura de informações presentes nos *Honeypots*. Os dados gerados pelo intruso são armazenados a partir do momento em que este toma o interior do veículo, pois o acionamento se dá pelos sinais enviados pelos sensores.

Tendo em mente o que representam os Honeypots, um consenso que se tem sobre o assunto é que sua implantação é extremamente cabível para as organizações, desde que seus sistemas não sejam tratados com negligência. Deve-se atentar ao fato de que alguns tipos de *Honeypots* podem trazer riscos às redes de computadores, caso não sejam bem configurados e monitorados. Sendo assim, é da responsabilidade de seus administradores, que estes não sejam utilizados como meio de causar danos às demais organizações, pois caso isto aconteça, estarão passíveis de processos judiciais.

3 Soluções em Honeypots

Neste trabalho, serão abordadas as soluções em *Honeypots* compreendidas pela lista:

- Honeyd
- Google Hack Honeypot
- Honeypots caseiros
- Honeytrap
- Nephentes
- Argos

Suas principais características serão exploradas, levando em consideração eventuais notas que tenham relevância, no que compete a detalhes de implantação. A tabela 4 mostra um comparativo entre as soluções a serem abordadas, de acordo com seus propósito principais:

Tabela 4: Comparativo entre tecnologias em *Honeypots*

Tecnologia	**Propósito**
Honeyd	Monitoramento de pseudo-serviços
Google Hack Honeypot	Monitoramento em aplicações *Web*
Honeypots caseiros	Captura de *payloads*
Honeytrap	Captura e análise de *malwares*
Nephentes	Monitoramento de serviços e captura de *worms*
Argos	Identificação de novos ataques

3.1 Honeyd

Honeyd é um *framework* criado para de permitir a associação de centenas e até milhares de endereços *IP* a uma única máquina, no intuito de simular ambientes distintos, onde, em cada um deles é possível se atribuir o comportamento de um determinado Sistema Operacional e seus serviços *TCP* e *UDP*, com vulnerabilidades e características específicas, atraindo a atenção de atacantes em potencial e coletando dados sobre os ataques que sejam eventualmente lançados. (PROVOS; HOLZ, 2007)

A ideia de se ter múltiplos endereços de *IP* associados a uma única máquina, é justamente a de se configurar vários ambientes falsos e distintos em um único computador, ao mesmo tempo que estes servem como iscas a públicos variados e que possam representar diferentes níveis de risco às rede em que sejam implantados. Em geral, a escolha dos endereços de *IP* a serem atribuidos aos sistemas falsos, é dada através da identificação, na rede, daqueles que não são utilizados, com a possibilidade de se estabelecer políticas para tal, como por exemplo a inserção de uma máquina supostamente comprometida entre endereços de *IP* vizinhos completamente saudáveis em um ambiente de produção, causando aos olhos de um atacante, um embaralhamento entre computadores comprometidos e a topologia padrão previamente estabelecida, o que é extremamente comum e conveniente em meio a uma simulação, uma vez que a identificação automática dos alvos como sendo realmente falsos, torna-se impraticável, caso o processo de denfinição dos endereços de *IP* seja guiado totalmente ao acaso.

Uma configuração típica, que faz uso tanto do recurso de múltiplos endereços de *IP* e de variados Sistemas Operacionais pode ser obtida através do uso do honeyd em conjunção a uma implementação de *Proxy ARP*, que irá atuar como um chaveador, encaminhando as requisições feitas aos múltiplos endereços *IP* aos seus respectivos Sistemas Operacionais falsos, cada qual provido de uma implementação de sua pilha de rede (compatível com a que foi implementada na versão do sistema emulado) e de um conjunto de serviços falsos, mas capazes de responder a tentativas iniciais de comunicação, sendo assim, válidos perante os atacantes nos primeiros contatos, tendo até mesmo seus tempos de resposta ajustados para que se pareçam, ao máximo, com suas versões legítimas.

A possibilidade de customização de um ambiente que simula um Sistema Operacional e seus serviços, permite que se crie *Honeypots* com finalidades específicas. Um dos computadores virtuais pode responder, por exemplo, como hospedeiro de uma versão arbitrária do Sistema Operacional *Windows* e que seja suscetível a alguma falha de segurança, também arbitrária, mas que permita sua exploração por *malwares*. Devido às características de uso desse Sistema Operacional, este cenário é propício à identificação de varreduras automáticas realizadas por computadores usualmente também infectados pelo mesmo *malware*. Isso revela quais são as máquinas comprometidas e proporciona um avanço no grau de agilidade quanto às decisões a serem tomadas sobre manutenções urgentes, contribuindo assim com a eliminação do tráfego de pacotes maliciosos e com a diminuição de computadores infectados.

3.1.1 Requisitos de implantação

Para que seja possível sua implantação, o *honeyd* exige que sejam satisfeitas algumas dependências relacionadas às bibliotecas compartilhadas utilizadas em sua implementação, tais como:

- *libdnet* (LIBDNET, 2009): provê uma interface simplificada e portável para muitas rotinas de baixo-nível relacionadas à comunicação em redes, incluindo: manipulação de endereços de rede, *cache* de *arp* do *kernel*, consulta e manipulação da tabela de rotas, integração com *firewalls* (como *IP filter*, *ipfw*, *ipchains*, *pf* e *PktFilter*), consulta e manipulação da interface de rede, tunelamento *IP* (*tun* do *BSD*/*Linux*, dispositivo universal *TUN*/*TAP*) e transmissão de pacotes *IP* puros e de *frames Ethernet*. Suporta as linguagens *C*, *C++*, *Python*, *Perl* e *Ruby*. Roda oficialmente sob as plataformas *BSD* (*OpenBSD*, *FreeBSD*, *NetBSD*, *BSD/OS*), *Linux* (*Slackware*, *Redhat*, *Debian* e demais sabores), *MacOS X*, *Windows* (*NT*/*2000*/*XP*), *Solaris*, *IRIX*, *HP-UX* e *Tru64*.

- A *API libevent* (LIBEVENT, 2009) provê um mecanismo para executar funções de *callback* quando um evento específico ocorrer nos descritores de um arquivo, ou assim que um *timeout* seja alcançado. Além disso, suporta *callbacks* por sinais ou *timeouts* regulares.

- *libpcap* (TCPDUMP. . . , 2009) é uma interface independente para a captura de pacotes em espaço de usuário, provê um *framework* portável para monitoramento de redes em baixo-nível. É aplicável por exemplo, na coleta de dados estatísticos de rede, monitoramento de segurança ou mesmo, o *debug* de rede.

- A biblioteca *PCRE* (PCRE. . . , 2009) é um conjunto de funções que implementam padrões de identificação para expressões regulares utilizando a mesma semantica e sintaxe que *Perl 5*. *PCRE* tem sua *API* nativa, assim como um conjunto de funções *wrapper* que correspondem às expressões regulares *POSIX*.

3.1.2 Manipulação de múltiplos endereços de IP

A configuração mais simples que pode ser obtida, afim de utilizar múltiplos endereços de *IP* em um computador hospedeiro, direcionando o tráfego de entrada a cada um computador virtual criado pelo honeyd, respeitando os endereços de *IP* de destino, pode ser obtida através dos utilitários de rede disponíveis em sistemas *UNIX* (também conhecidos como *net-tools*).

Supondo que já exista um endereço *IP* associado a interface de rede local, pode-se verificar isso através do utilitário ifconfig:

```
bodacious@mothership:~$ /sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr 00:1d:92:bc:21:fc
          inet addr:10.90.10.93  Bcast:10.90.10.255  Mask:255.255.255.0
          inet6 addr: fe80::21d:92ff:febc:21fc/64 Scope:Link
          UP BROADCAST NOTRAILERS RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:22802 errors:0 dropped:0 overruns:0 frame:0
          TX packets:13166 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:8618694 (8.2 MiB)  TX bytes:1853636 (1.7 MiB)
          Interrupt:18 Base address:0x6000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:5599 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5599 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
```

```
          RX bytes:4320001 (4.1 MiB)  TX bytes:4320001 (4.1 MiB)
```

Para realizar a atribuição de um endereço de *IP* adicional à interface-física ethernet, identificada pelo endereço *MAC* "00:1d:92:bc:21:fc", executa-se, como usuário privilegiado, o seguinte procedimento:

```
root@mothership:~# ifconfig eth0:1 10.90.10.94 netmask 255.255.255.0 up
```

Uma nova verificação da saída do ifconfig, deve revelar que novo endereço foi atribuído com sucesso ao *alias* "eth0:1", estabelecido como identificador da interface virtual responsável pelo gerenciamento do novo endereço de *IP*:

```
bodacious@mothership:~$ /sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr 00:1d:92:bc:21:fc
          inet addr:10.90.10.93  Bcast:10.90.10.255  Mask:255.255.255.0
          inet6 addr: fe80::21d:92ff:febc:21fc/64 Scope:Link
          UP BROADCAST NOTRAILERS RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:50658 errors:0 dropped:0 overruns:0 frame:0
          TX packets:24479 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:13678318 (13.0 MiB)  TX bytes:2772362 (2.6 MiB)
          Interrupt:18 Base address:0x6000

eth0:1    Link encap:Ethernet  HWaddr 00:1d:92:bc:21:fc
          inet addr:10.90.10.94  Bcast:10.90.10.255  Mask:255.255.255.0
          UP BROADCAST NOTRAILERS RUNNING MULTICAST  MTU:1500  Metric:1
          Interrupt:18 Base address:0x6000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:6771 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6771 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:4794574 (4.5 MiB)  TX bytes:4794574 (4.5 MiB)
```

Verifica-se através da tabela de rotas do *Kernel*, que a rota-padrão de saída do sistema, é identificada por padrão, como sendo associada à interface *eth0*:

```
bodacious@mothership:~$ /sbin/route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.90.10.0      *               255.255.255.0   U     0      0        0 eth0
loopback        *               255.0.0.0       U     0      0        0 lo
default         10.90.10.1      0.0.0.0         UG    0      0        0 eth0
```

Portanto, para testar a nova interface atribuida, "eth0:1", deve-se indicar aos utilitários como o *ping*, qual o endereço de *IP* de onde os pacotes devem ser transmitidos:

```
bodacious@mothership:~$ ping -c 1 -I 10.90.10.94 dc.uel.br
PING dc.uel.br (189.90.67.67) from 10.90.10.94 : 56(84) bytes of data.
64 bytes from beta.dc.uel.br (189.90.67.67): icmp_seq=1 ttl=63 time=0.122 ms

--- dc.uel.br ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.122/0.122/0.122/0.000 ms
```

Com estes procedimentos, é possível se dar início à implantação do *honeyd*, pois ele utiliza espaços de *IP* não alocados na rede. Sendo capaz de associar vários endereço de *IP* à interface de rede, o primeiro passo para que se possa implantar com sucesso este serviço está garantido.

3.2 Google Hack Honeypot

Google Hack Honeypot (GOOGLE. . . , 2009), ou *GHH*, é uma ferramenta criada no intuito de se reagir a um novo tipo de ameaça: a utilização de mecanismos de busca para facilitar invasões. Foi projetado para reconhecer atacantes que utilizam os serviços de busca para encontrarem falhas de segurança em recursos desprotegidos.

Os alvos identificados, têm em comum o fato de possuírem aplicações Web vulneráveis e indexadas pelos robôs de busca, podendo ser facilmente identificadas, através de características específicas de cada falha que possuem, e com isso, acabam atraindo ataques aos seus sistemas hospedeiros.

O *GHH* emula uma aplicação Web vulnerável e indexada pelos mecanismos de busca. É escondido dos visitantes tradicionais, mas é encontrado através da utilização de web-crawlers

ou buscadores. Isso é possível através de um link transparente (não é detectado pela navegação convencional) estabelecido no momento que uma ferramenta indexa o site efetuando a leitura de seu código fonte (caracterizado pela presença de elementos *HTML*, *XHTML*, *XML*, *CSS* e *JavaScript*). O link transparente (quando bem configurado) reduz a quantidade de falsos-positivos e evita que seja possível a identificação, pelos atacantes, da "impressão digital" do *Honeypot*.

Todas as tentativas de varredura por falhas são registradas para análises posteriores. Dessa maneira, pode-se colocar em *black-lists* os endereços de *IP* (ou faixas inteiras destes) pertencentes a atacantes recorrentes e negar suas futuras tentativas de acesso, assim como notificar os provedores de que os ataques estão partindo de suas redes, ou então aplicar os dados obtidos na geração de estatísticas voltadas para a área de pesquisa.

Com essa ferramenta, é possível prover às organizações um nível adicional de segurança à sua presença na Web, visto que cada vez mais sites e aplicações estão sendo indexados pelos mecanismos de busca. O reflexo maior deste fenômeno, é o aumento da quantidade de sistemas que são frequentemente mal-configurados (como fóruns de mensagens e painéis administrativos), motivo pelo qual são agregados riscos enormes (e desnecessários) tanto à segurança quanto à integridade dos dados, além de possivelmente ocorrer o comprometimento da boa utilização dos recursos. *GHH* é uma ferramenta escrita em *PHP* e é licensiada sob a *GNU Public License 2* (GNU..., 2009).

3.3 Honeypots caseiros

É possível criar um *Honeypot* "caseiro" sem muito esforço. A técnica mais simples que pode ser empregada, consiste em acionar uma ou mais escutas em portas arbitrárias, sejam elas *TCP* ou *UDP*. Isso pode ser facilmente implementado pelo o uso do utilitário netcat, advindo do kit de ferramentas *Unix*. É amplamente conhecido como "o canivete suíço TCP/IP". Este tipo de Honeypot é classificado como Honeypot de Monitoramento de Portas, tendo como característica marcante o fato de ser utilizado na identificação de scans, tentativas de descoberta de "assinaturas" geradas pelos serviços disponibilizados pelo servidor e atividade-

des suspeitas/não-autorizadas, sejam elas manuais ou provenientes de ataques lançados por *malwares*. (KREIBICH; CROWCROFT, 2004)

Existe outra classe de honeypots caseiros, que é obtida através da criação de ambientes básicos para que sistemas-operacionais possam ser "enjaulados". Essa técnica pode ser obtida através de uma preparação manual desse ambiente, ou através do acionamento de um método de criação de um novo ambiente, a partir de um template previamente configurado.

3.3.1 Honeypot de Monitoramento de Portas

Sobre o netcat

O *netcat* lê e escreve e dados através de conexões de rede, provendo um conjunto de funcionalidades que proporcionam realizar tanto o *debug* quanto a exploração dos dados transmitidos. Tem a habilidade de trabalhar com os protocolos *TCP* ou *UDP*, operando de modo eficiente os *pipes* de redirecionamento de entrada e saída. Trata-se de uma ótima ferramenta a ser integrada em tarefas automatizadas, sendo compatível com uma grande variedade de linguagens de *scripting*, utilizadas amplamente no auxílio da administrações de servidores.

Supondo que se queira abrir uma conexão *TCP* para testes em um *host* servidor, a sintaxe padrão de para a execução do *netcat* é definida por:

```
nc -l -p porta
```

Onde os parâmetros tem o seguinte significado: -l: listening mode, ou seja, deve-se "escutar" por conexões, sendo isso o que caracteriza um "servidor", por mais que este exemplo seja trivial -p porta: especifica a porta em que a conexão deve ser disponibilizada

Para se conectar como cliente ao host servidor, a sintaxe utilizada é:

```
nc host porta
```

Onde "host" deve ser o nome de host em que se deseja conectar e "porta", a porta cuja conexão deve ser estabelecida

A conexão que pode ser aberta, utilizando a sintaxe e uso abordados até então será bidirecional, ou seja, têm-se a possibilidade de enviar e receber dados tanto no lado do cliente quanto no lado do servidor. Os dados transmitidos, são imprimidos na saída padrão do terminal onde o netcat foi acionado, mas pode-se redirecionar o fluxo de dados, utilizando pipes de redirecionamento de saída, de modo que as informações que chegarem terão como destino um arquivo específico, ou mesmo um *FIFO* alocado no sistema de arquivos.

Exemplificando o uso do netcat

Pode-se acionar o *netcat* para que os dados capturados pelo servidor sejam armazenados em um arquivo, para uma análise posterior com a sua sintaxe de acionamento modificada para:

```
nc -l -p porta 1>./captura.log 2>./erros.log
```

Onde os operadores *1* e *2*, significam respectivamente *saída padrão* e *saída de erros padrão* e ambos indicam que seus dados devem ser redirecionados para arquivos específicos.

O acionamento poderia ser dado, utilizando o mesmo princípio, acionando-se o netcat na forma:

```
nc -l -p porta &>./captura-geral.log
```

Onde o operador "&" indica que tanto a saída padrão quanto a saída de erros padrão devem ser redirecionados para um arquivo único.

Se necessário, pode-se criar um *FIFO* no sistema de arquivos e utilizá-lo por exemplo, com o propósito de acoplar alguma ferramenta de análise de logs qualquer, paralela a execução do *netcat*, de modo que identifique padrões de comportamento nos dados recebidos e, através disso, possa atuar como guia do processo de coleta de informações, através de uma maneira mais efetiva.

Para que isso seja possível, cria-se um *FIFO* chamado *fifo-teste*, com o comando mkfifo:

```
mkfifo fifo-teste
```

Em seguida, deve-se associar o netcat ao *FIFO* recém-criado (O caractere "&" é utilizado para enviar o processo para *background*):

```
nc -l -p porta > fifo-teste &
```

Para monitorar os dados, de maneira minimalista, e de modo que se possa extrair informações interessantes, pode-se utilizar uma ferramenta filtra o conteúdo do *FIFO* por informações específicas, como o *grep*.

3.4 Honeytrap

O Honeytrap (HONEYTRAP..., 2009) é um daemon *Honeypot* que observa ataques contra os serviços de rede. Em contraste com outros *Honeypots*, que em geral focam na coleta de *malwares*, o *Honeytrap* pretende capturar os *exploits* iniciais, processando futuramente os traços deixados pelos atacantes.

Ser apto a processar ataques desconhecidos significa que não se exige conhecimento algum sobre o protocolo ou vulnerabilidade explorada durante as investidas. No entanto, uma conexão de entrada pode ser gerenciada de maneiras diferenciadas. Implementa um conceito de servidor dinâmico, monitorando a stream de rede por sessões de entrada e inicia os serviços de escuta no mesmo instante. Cada serviço de escuta pode gerenciar múltiplas conexões e se auto-terminar após algum tempo de ociosidade.

Abaixo segue a saída com a inicialização do honeytrap:

```
honeytrap v1.1.0 - Initializing.
Loading plugin magicPE v0.0.1
Loading plugin ftpDownload v0.5.3
Loading plugin tftpDownload v0.4.1
Loading plugin b64Decode v0.3.1
Loading plugin vncDownload v0.3
Loading plugin SaveFile v0.2.1
Loading plugin submitPostgres v0.1.1
Servers will run as user honeytrap (1004).
Servers will run as group nogroup (65534).
Loading default responses.
```

```
Connections will be handled in mirror mode by default.
Logging to /opt/honeytrap/honeytrap.log.
Initialization complete.

honeytrap v1.1.0 Copyright (C) 2005-2008 Tillmann Werner
[2009-04-10 16:33:56] ---- Trapping attacks via NFQ. ----
[2009-04-10 16:34:11] ---- honeytrap stopped ----
```

Os dados de chegada são gerenciados por uma *stream* de *bytes* por cada sessão. Essa *stream* pode ser processada por diferentes módulos, por exemplo, para encontrar padrões nas mensagens recebidas, como decodificação de dados, *scan* de *worms* ou a execução de comandos para o *download* de *malwares*.

3.5 Nephentes

Nephentes (NEPENTHES. . . , 2009) é um Honeypot de Baixa-interação, emulando serviços e vulnerabilidades conhecidas, no intuito de capturar informações informações sobre ataques em potencial. Foi projetado para emular as vulnerabilidades utilizadas pelos *worms* para se espalharem, assim como, capturar os *worms* transmitidos.

Existem muitas maneiras que os worms podem utilizar para se espalharem, por isso o Nephentes é modular. Sua interfaces provêem:

- Resolução assíncrona de *DNS*
- Emulação de vulnerabilidades
- Download de arquivos
- Submissão de arquivos baixados
- Disparo de eventos
- Gerenciador de código *shell*

Os módulos de vulnerabilidades do *Nephentes* requerem que se possua o conhecimento sobre o funcionamento da vulnerabilidade a ser implantada. Deve-se realizar uma configuração

do *daemon*, de modo que imite fielmente seu diálogo com o *worm* e além de seu comportamento, que também deve ser definido, caso a vulnerabilidade em questão fosse explorada.

O *Nephentes* é muito útil na descoberda de novos exploits para velhas vulnerabilidades. Escrevendo-se diálogos em máquinas realmente vulneráveis, pode-se obter novas informações sobre as falhas a partir da análise dos logs gerados neste processo.

3.6 Argos

O projeto *Argos* (ARGOS..., 2009), trata-se de um emulador completo e seguro de Sistemas Operacionais para Honeypots. Se baseia no Qemu (QEMU..., 2009) (NAKAMOTO *et al.*, 2009), que utiliza tradução dinâmica de instruções para atingir bons resultados quanto à velocidade de emulação. (PORTOKALIDIS *et al.*, 2006)

Argos extende o *Qemu* de modo que habilita o mesmo a detectar tentativas de comprometimento remoto no Sistema Operacional hospedado. Utilizando análise dinâmica de defeitos gerados, faz o rastreamento dos dados provenientes da rede e que disparam chamadas de sistema, detectando qualquer tentativa maliciosa de interação remota. Quando o ataque é detectado, uma imagem da memória é registrada em *log*.

Historicamente, o *Argos* foi o primeiro passo para a criação de um *framework* que irá utilizar *Honeypots* da mais nova geração para identificar e produzir maneiras de remediar *worms 0-day* e outros ataques similares. A próxima geração de *Honeypots* não irá precisar que seus endereços de *IP* não sejam reconhecidos como pertencentes a tais, podendo inclusive, ser publicamente disponíveis para gerar tráfego ativamente. Em *Honeypots* atuais, isso não é possível, pois não há distinção entre o tráfego malicioso do tráfego legítimo, ao passo que o *Argos* consegue distinguir isso pois "assina" cada tentativa de possível exploração, tendo assim sua capacidade diferenciada de identificar o tráfego malicioso.

Características gerais:

- Não requer modificações no Sistema Operacional

- Suporta múltiplos Sistemas Operacionais, incluindo *Linux*, *Windows 2000* e *Windows XP* (além de todos aqueles também suportados pelo Qemu)
- Emula processadores *x86*, incluindo as extensões *MMX*, *SSE* e *SS2*
- Roda em diferentes Sistemas Operacionais (*Linux Unix*, *Windows*) e *CPUs* (*x86*, *x86_64*, *PowerPC*)

Análise dinâmica de defeitos:

- Detecta ataques que gerem fluxo arbitrário de dados
- Detecta ataques que realizam a execução de código arbitrário
- Gerencia *DMA*
- Gerencia os mapeamentos de espaço de usuário no *Kernel*

Conclusão

Neste trabalho, foram apresentadas algumas das principais ameaças que assolam a *Internet*, nos dias de hoje, enfatizando a importância dos ataques *Denial of Service*, devido seu alto grau de periculosidade. Foram também caracterizadas como ameaças relevantes, as redes de pedofilia, organizações criminosas que utilizam os recursos tecnológicos das redes de computadores para a troca de material ilegal entre seus componentes. Como grande problema aos usuários finais de computadores, foram abordados os *malwares*, cujos tipos e danos que podem causar variam, sendo um dos principais fenômenos digitais já presenciados em escala global atualmente.

Tendo motivação para se proteger das ameaças e, principalmente, para estudá-las, foram apresentados diversos conceitos relacionados às tecnologias em *Honeypots*. Foram discutidos os tipos de *Honeypots*, seus propósitos e aspectos legais envolvidos. Aos que se interessarem sobre a implantação de *Honeypots* em suas redes, foram referenciados alguns projetos com reconhecimento mundial, podendo estes serem tomados como ponto de partida.

Sua utilização é de suma importância em grandes organizações, como Universidades e Empresas. Com sua implantação nestes ambientes, pode-se realizar um levantamento extensivo sobre o comportamento das redes, sob condições normais e de risco.

Os *Honeypots* são ferramentas excepcionais para os especialistas em segurança. Revelam novos ataques, assim que estes são lançados, dessa maneira pode-se preparar soluções para os problemas que possam afetar massivamente os sistemas implantados no mundo todo, assim como aumentar o nível de proteção das instituições contra as investidas de atacantes. Avaliam quais são as ameaças recorrentes, auxiliando na melhoria da segurança nas redes e por consequência seus usuários são também protegidos com tais medidas. Inclusive, ajudam a combater diversas organizações criminosas no mundo todo. Sua implantação é recomendada

ao time de administradores das organizações, por se tratarem de valiosas ferramentas a serem empregadas na melhoria contínua de seus processos de segurança.

Como trabalho futuro, pode-se realizar um conjunto de instruções completas para a implantação de tecnologias em *Honeypots* em instituições interessadas, fornecendo diretrizes para que tanto a obtenção quanto a análise dos dados possam fazer parte das políticas de segurança empregadas. É de interesse comum, que se realize um estudo de caso no qual as análises realizadas sejam expostas, para que seja validada efetivamente a escolha das tecnologias de acordo com o ambiente em que podem ser inseridas.

Referências Bibliográficas

ACOHIDO, B. *Q&A on U.S. electrical grid infiltrated by Chinese, Russian cyberspies*. Abril 2009. http://lastwatchdog.com/chinese-russian-cyberspies-lurk-us-electrical-grid/.

ARGOS - An emulator for capturing zero-day attacks. Dezembro 2009. http://www.few.vu.nl/argos/?page=4. Disponível em: <http://www.few.vu.nl/argos-/?page=4>.

BAUMANN, R. *et al. White Paper: Honeypots*. Fevereiro 2002.

BRADEN, R. *RFC1122 - Requirements for Internet Hosts - Communication Layers*. Dezembro 2009. http://www.faqs.org/rfcs/rfc1122.html. Disponível em: <http://www.faqs-.org/rfcs/rfc1122.html>.

CHANG, A. *Wireless sensors extend reach of Internet into the real world*. Fevereiro 2007. http://www.usatoday.com/tech/wireless/data/2007-02-12-wireless-sensors_x.htm. Disponível em: <http://www.usatoday.com/tech/wireless/data/2007-02-12-wireless-sensors_x.htm>.

DITTRICH, D. *The s̈tacheldrahtd̈istributed denial of service attack tool*. Dezembro 1999. http://staff.washington.edu/dittrich/misc/stacheldraht.analysis.txt. Disponível em: <http://staff.washington.edu/dittrich/misc% -/stacheldraht.analysis.txt>.

GNU General Public License 2. Dezembro 2009. http://www.gnu.org/copyleft/gpl.html. Disponível em: <http://www.gnu.org/copyleft/gpl.html>.

GOOGLE Hack Honeypot. Dezembro 2009. http://ghh.sourceforge.net/. Disponível em: <http://ghh.sourceforge.net/>.

GROUP, M. M. *Internet growth statistics*. Dezembro 2009. http://www.internetworldstats.com/emarketing.htm. Disponível em: <http://www-.internetworldstats.com/emarketing.htm>.

HOEPERS, C. *et al. Honeypots e Honeynets: Definições e Aplicações*. Outubro 2007. http://www.cert.br/docs/whitepapers/honeypots-honeynets. Disponível em: <http://www.cert.br/docs/whitepapers/honeypots-honeynets>.

HOGWASH intrusion detection system and packet scrubber. Dezembro 2009. http://hogwash.sourceforge.net/. Disponível em: <http://hogwash.sourceforge.net/>.

HONEYNET Project Blog. Dezembro 2009. http://www.honeynet.org/aggregator. Disponível em: <http://www.honeynet.org/aggregator>.

HONEYTRAP – A Dynamic Meta-Honeypot Daemon. Dezembro 2009. `http://honeytrap.carnivore.it`. Disponível em: <http://honeytrap.carnivore.it>.

HTTP Server Project. Dezembro 2009. `http://httpd.apache.org/`. Disponível em: <http://httpd.apache.org/>.

INSTITUTE, I. S. *Internet Protocol*. [S.l.], 1981. Disponível em: <http://tools.ietf.org/html/rfc791>.

JONES, J. K. *et al.* Honeynets: an educational resource for it security. In: *CITC5 '04: Proceedings of the 5th conference on Information technology education*. New York, NY, USA: ACM, 2004. p. 24–28. ISBN 1-58113-936-5.

KNOW your enemy: revealing the security tools, tactics, and motives of the blackhat community. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 2002. ISBN 0-201-74613-1.

KNOW Your Enemy, Second Edition: Learning about Security Threats (2nd Edition). [S.l.]: Pearson Education, 2004. ISBN 0321166469.

KREIBICH, C. *et al.* Honeycomb: creating intrusion detection signatures using honeypots. *SIGCOMM Comput. Commun. Rev.*, ACM, New York, NY, USA, v. 34, n. 1, p. 51–56, 2004. ISSN 0146-4833.

KREKEL, B. Capability of the people's republic of china to conduct cyber warfare and computer network exploitation. *The US-China Economic and Security Review Commission*, Outubro 2009.

LEITA, C. *et al.* The leurre.com project: Collecting internet threats information using a worldwide distributed honeynet. In: *WISTDCS '08: Proceedings of the 2008 WOMBAT Workshop on Information Security Threats Data Collection and Sharing*. Washington, DC, USA: IEEE Computer Society, 2008. p. 40–57. ISBN 978-0-7695-3347-6.

LIBDNET. Dezembro 2009. `http://libdnet.sourceforge.net/`. Disponível em: <http://libdnet.sourceforge.net/>.

LIBEVENT. Dezembro 2009. `http://www.monkey.org/~provos/libevent/`. Disponível em: <http://www.monkey.org/~provos/libevent% -/>.

MCCARTY, B. Botnets: Big and bigger. *IEEE Security and Privacy*, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 1, n. 4, p. 87–90, 2003. ISSN 1540-7993.

NAKAMOTO, Y. *et al.* Proposing universal execution trace framework for embedded software using qemu. In: *STFSSD '09: Proceedings of the 2009 Software Technologies for Future Dependable Distributed Systems*. Washington, DC, USA: IEEE Computer Society, 2009. p. 173–178. ISBN 978-0-7695-3572-2.

NEPENTHES - finest collection -. Dezembro 2009. `http://nepenthes.carnivore.it/`. Disponível em: <http://nepenthes.carnivore.it/>.

NMAP - Free Security Scanner For Network Exploration & Security Audits. Dezembro 2009. `http://nmap.org/`. Disponível em: <http://nmap.org/>.

OWENS, W. A. *et al. Technology, Policy, Law, and Ethics Regarding U.S. Acquisition and Use of Cyberattack Capabilities*. [S.l.]: National Academies Press, 2009. ISBN 0309138507.

PCRE - Perl Compatible Regular Expressions. Dezembro 2009. `http://www.pcre.org/`. Disponível em: <http://www.pcre.org/>.

PORTOKALIDIS, G. *et al.* Argos: an emulator for fingerprinting zero-day attacks for advertised honeypots with automatic signature generation. In: *EuroSys '06: Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems 2006*. New York, NY, USA: ACM, 2006. p. 15–27. ISBN 1-59593-322-0.

PROVOS, N. Improving host security with system call policies. *12th USENIX Security Symposium*, 2003.

PROVOS, N. *Systrace - Interactive Policy Generation for System Calls*. Dezembro 2009. `http://www.citi.umich.edu/u/provos/systrace/`. Disponível em: <http://www.citi-.umich.edu/u/provos/systrace/>.

PROVOS, N. *et al. Virtual honeypots: from botnet tracking to intrusion detection*. [S.l.]: Addison-Wesley Professional, 2007. ISBN 9780321336323.

QEMU - open source processor emulator. Dezembro 2009. `http://www.qemu.org/`. Disponível em: <http://www.qemu.org/>.

R., S. *et al.* The flask security architecture: System support for diverse security policies. *Eighth USENIX Security Symposium*, p. 123–139, 1999.

REIS, A. V. dos *et al. Pesquisa sobre pornografia infantil na internet*. [S.l.], 2004.

SADASIVAM, K. *et al. Design of network security projects using Honeypots*. [S.l.]: University of Houston – Clear Lake, 2005.

SHIELDS, C. What do we mean by network denial of service? *Workshop on Information Assurance and Security*, Junho 2002.

SPITZNER, L. *Definitons and Value of Honeypots*. Outubro 2001. `http://www.enteract.com/~lspitz/honeypot.html`.

SPITZNER, L. *Honeypots: Tracking Hackers*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002. ISBN 0321108957.

SPITZNER, L. The honeynet project: Trapping the hackers. *IEEE Security and Privacy*, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 1, n. 2, p. 15–23, 2003. ISSN 1540-7993.

SPITZNER, L. *Honeypots: Are They Illegal?* Junho 2003. `http://www.securityfocus.com/infocus/1703`. Disponível em: <http://www.securityfocus.com/infocus/1703>.

TCPDUMP / libpcap. Dezembro 2009. `http://www.tcpdump.org/`. Disponível em: <http://www.tcpdump.org/>.

TEENAGE hacker tells his side of land attack story. Dezembro 2009. `http://jya.com/land-attack.txt`. Disponível em: <http://jya.com/land-attack.txt>.

THE Honeynet Projects. Dezembro 2009. `http://www.honeynet.org/project`. Disponível em: <http://www.honeynet.org/project>.

THE LAND attack (IP DOS). Dezembro 2009. `http://insecure.org/sploits/land.ip.DOS.html`. Disponível em: <http://insecure.org/sploits/land.ip.DOS% -.html>.

WERMSKE, R. *Experts: U.S. Military's Cyberwar Rules ill-formed¨*. Novembro 2009. `http://lastwatchdog.com/chinese-russian-cyberspies-lurk-us-electrical-grid/`.