HIJAZI, MAZZORANA E RAVANELLO

HONEYPOTS E ASPECTOS LEGAIS

Dissertação apresentada ao Programa de Pós-Graduação em Informática Aplicada da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de especialização em Redes e Sistemas Distribuídos.

Área de Concentração: *Segurança de sistemas, Honeypots e aspectos legais das ferramentas automatizadas de aquisição de provas de invasão de sistemas*

Orientador: Prof. Mestre Marcos Aurelio Pchek Laureano

CURITIBA

2004

Ravanello, Anderson Luiz; Hijazi, Houssan Ali; Mazzorana, Sidney Miguel
Honeypots e Aspectos Legais. Curitiba, 2004. 85 p.

Dissertação – Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática Aplicada.

1. Honeypots 2. Honeynets 3. Ferramentas de Segurança 4. Tecnologia e Aspectos Legias. I.Pontifícia Universidade Católica do Paraná. Centro de Ciências Exatas e de Tecnologia. Programa de Pós-Graduação em Informática Aplicada II-t

Dedicamos esta monografia às nossas famílias pelo tempo do qual abriram mão para que pudéssemos compor este trabalho.

Agradecimentos

Agradecemos aos professores que estimularam nossa curiosidade nos impondo novos desafios e ao Mestre Laureano pela ajuda e orientação para concretização deste trabalho.

Agradecemos aos nossos amigos e familiares por apoiarem, fazendo com que nos esforçássemos ao máximo.

Sumário

Lista de Figuras

Lista de Abreviaturas

IDS	*Intrusion Detection System*
IP	*Internet Protocol*
DoS	*Denial of Service*
DDoS	*Distributed Denial of Service*
LAN	*Local Area Network*
FTP	*File Transfer Protocol*
HTTP	*Hyper Text Transfer Protocol*
ICMP	*Internet Control Message Protocol*
DNS	*Domain Name Server*
UDP	*User Datagram Protocol*
DMZ	*De-Militarized Zone*
NIDS	*Network based Intrusion Detection Systems*
HIDS	*Host based Intrusion Detection Systems*
SIV	*System Integrity Verifier*
IIS	*Internet Information Services*
NIPS	*Network Intrusion Prevention System*
TTL	*Time To Live*
ToS	*Type of Service*
TELNET	*Network Terminal Protocol*

Resumo

Este trabalho apresenta uma análise dos conceitos de ataques digitais e ferramentas de segurança de sistemas, com o maior foco em *Honeypots* e *Honeynets* e como estes podem ser usados para fins legais.

Abstract

This paper introduces the concepts behind digital attacks and system security tools, with focus on honeypots and honeynets and how can they be deployed in order to provide legal value.

Keywords: Honeypots, Honeynets, Legal Value, Security.

Capítulo 1

Introdução

A questão da segurança de dados parece ser infinita, e até hoje especialistas nesta área reúnem esforços para obter uma garantia técnica da construção de redes invulneráveis e intransponíveis.

Até hoje, o único consenso a que se chegou foi o de que o único computador completamente seguro é aquele que se encontra desligado do mundo e sem energia elétrica, trancado em um cofre e com um par de guardas armados do lado de fora do cofre.[HON01b]

Existem então iniciativas variadas para tentar aprimorar a segurança dos sistemas, de modo que possa se ter um equilíbrio entre a conectividade das organizações e sua vulnerabilidade digital.

Contanto, não basta simplesmente que as ferramentas de segurança sejam aprimoradas para se evitar invasões; é necessário que se encontre uma maneira de punir os invasores digitais, utilizando o respaldo legal encontrado na legislação brasileira. Assim, com este trabalho se busca analisar ferramentas de segurança de sistemas que possam ser utilizadas como comprovação de ataque, apresentando subsídios pra o processo legal, com ênfase em *honeypots* e *honeynets*.

No capítulo 2, serão vistos os ataques digitais, as pessoas, que fazem tais ataques, e as motivações psicológicas, sociais e econômicas por trás do ato, expondo dois grandes grupos de invasores digitais.

No capítulo 3, vê-se as três principais classes de ferramentas de segurança de sistemas, suas vantagens e desvantagens.

No capítulo 4, são apresentadas *honeypots, honeynets* e suas funções principais, a captura de tráfego para estudo de ataques e proteção contra invasões digitais.

No capítulo 5, encontra-se embasamento jurídico para análise forense de ferramentas de segurança de sistemas e subsídios legais para o processo e conseqüente punição dos criminosos digitais.

Capítulo 2

Ataques, Atacantes e motivações

A origem dos problemas de segurança reside na certeza de que a todo o momento há muitas pessoas mal intencionadas, buscando obter vantagens financeiras ou comerciais ou apenas para causar danos e incômodos a sistemas digitais. Neste capítulo serão vistas as formas de ataque, as ferramentas utilizadas pelos invasores e o tipo de pessoa que pratica este tipo de ataque, com destaque para a figura do *script kiddie*.

2.1. O que é um ataque

Segundo [SHI00], um ataque é uma ação nociva à segurança de um sistema que deriva de uma ameaça inteligente, sendo essa ameaça uma tentativa deliberada (no sentido de método ou técnica) de evitar os serviços de segurança e violar a política de segurança de um sistema, podendo ser classificado, inicialmente quanto ao seu ao objetivo em passivo e ativo e também quanto à sua origem em interna e externa.

Em [CAM97a] e em [INN01] encontramos uma forma adicional de classificação de ataques baseada no grau de severidade do dano que o ataque pode causar, variando desde vandalismo virtual até negação de serviço e destruição de equipamentos.

2.2. Classificação de ataques

Para auxiliar na compreensão dos riscos de ataque aos quais os sistemas digitais estão expostos, é necessário classificar os ataques conforme objetivo, origem e severidade.

Conforme [SHI00, LAN03, CAM97a], existe uma fronteira virtual erguida pelas entidades na forma de sua Política de Segurança. Uma política de segurança é um conjunto de regras que visa regulamentar a produção, acesso e tráfego de informações e recursos

computacionais em uma organização e determinar formas de agir em caso de violação destas regras. Este conjunto de regras é usado como limitador e para determinar o escopo das técnicas e ferramentas de segurança de uma rede.

As políticas de segurança trazem as expressões "perímetros de segurança" e "domínios de segurança" como sinônimos, que serão usados a seguir.

2.2.1. Classificação de ataques conforme objetivo

- Ataque Passivo - Ataques passivos são aqueles que buscam obter informações de um sistema, evitando influir funcionamento do sistema afetado. Furtos de senhas, de endereços de e-mails, espionagem digital, fraude bancária e esquemas de desvio de dinheiro são exemplos de ataques do tipo passivo. As entidades que são mais vulneráveis a este tipo de invasão são as instituições financeiras (bancos, companhias de cartão de crédito), instituições privadas (empresas, sociedades) e departamentos governamentais. Estas instituições são mais visadas devido ao tipo de informação que trafegam, pois o mesmo representa ganhos imediatos para o atacante (por exemplo: desvio de fundos, espionagem industrial, hostilidades internacionais).

- Ataque ativo - Ataques ativos são os que buscam afetar o funcionamento dos dispositivos de uma rede, seja através da desativação de serviços críticos em servidores, comprometimento de informações do alvo, desperdício de recursos, destruição de informações e até comprometimento físico dos recursos de um sistema. Ataques ativos são exemplificados pela pichação de *sites*, destruição intencional de dados, desperdício de recursos do sistema (processamento, memória, documentos de impressão), suspensão dos serviços e até desativação por completo de um alvo, e, potencialmente, danos físicos ao equipamento envolvido.

2.2.2. Classificação dos ataques conforme a origem

- Ataque interno - Ataques internos são aqueles que são iniciados do lado de dentro do perímetro de segurança que é criado pelas políticas de segurança de uma organização. São considerados ataques internos todas as atividades que visam abusar ou fazem mal uso dos recursos computacionais aos quais teriam direitos de acesso regularmente. Funcionários

que utilizam os recursos da empresa para buscar informações sensíveis, vírus que contaminem máquinas de usuários para depois atacar servidores e ações de engenharia social, nas quais um indivíduo mal intencionado se vale da confiança a ele garantida para tentar comprometer informações são exemplos de ataques internos.

- Ataque externo - Ataques externos são todas as atividades nocivas ao funcionamento dos recursos computacionais que partam do perímetro externo ao domínio da política de segurança da entidade atacada. Esse perímetro diferencia os usuários internos dos externos e caracteriza os ataques externos como todos aqueles que são gerados por usuários não autorizados ou ilegítimos do sistema. Considera-se então o ambiente da Internet como a origem da maioria dos ataques externos a um sistema devido ao alto grau de conectividade dos sistemas à essa rede. No entanto, em uma rede corporativa é possível conceber-se diversos perímetros de segurança, e ataques vindos de outros setores, apesar de estarem partindo da mesma rede física, seriam considerados como ataques externos.

Exemplificando: um administrador de sistemas constrói um domínio de segurança chamado "diretoria", e outro chamado "funcionários". Se um usuário autorizado e autenticado no domínio "funcionários" efetua uma ação de ataque contra o domínio "diretoria", esse ataque seria considerado como externo do ponto de vista dos domínios de segurança, mas interno do ponto de vista da rede corporativa (ambos os sistemas se encontram na mesma rede, porém separados por políticas e regras de segurança diferentes). Os agentes de ataque externo potenciais são os amadores que pregam peças baseadas em ferramentas automatizadas de ataque, criminosos virtuais organizados, terroristas internacionais e até entidades governamentais hostis.

2.2.3. Classificação de ataques conforme a severidade

Outra forma de classificação dos ataques é a que abrange o dano causado quando o ataque obtém sucesso. A severidade é determinada de acordo com o tempo gasto na recuperação e prejuízo que o ataque consegue causar ao sistema afetado. O grau de severidade, no entanto, não é uma informação quantitativa, mas sim qualitativa, e diretamente ligada ao objetivo principal da entidade atacada. Um ataque de baixa severidade para uma entidade pode ser de severidade crítica para outra. Durante a construção da política de

segurança, o administrador de sistemas deve determinar quais são os tipos de ataques que devem se encaixar em quais categorias. Durante o processo de caracterização destes incidentes, o administrador deve responder a perguntas como:

- Qual o principal objetivo do sistema em relação ao negócio da entidade?
- Quanto tempo a entidade pode funcionar em caso de interrupção dos serviços?
- De todos os serviços disponibilizados, quais são os mais importantes perante os objetivos da entidade?

De posse destas informações é possível determinar quais são as prioridades no caso de falhas múltiplas e contabilizar os danos sofridos em caso de ataques.

- Baixa Severidade - Ataques de baixa severidade são todos aqueles cujo acontecimento não atrapalhem o funcionamento da empresa. Considera-se também de baixa severidade os ataques que podem ser rapidamente reparados, com pouco ou nenhum impacto para entidade. Um ataque que causasse a deleção de arquivos importantes, mas os mesmos pudessem ser rapidamente recuperados do conjunto de *backups* do dia anterior, e a brecha que permitiu seu acontecimento fechada, seria considerado de baixa severidade.

- Alta Severidade - Ataques de alta severidade são aqueles que, em geral, dificultariam o funcionamento da empresa ou que gastariam tempo e ou recursos para o reparo. Epidemias de vírus na rede interna, quedas de servidores de arquivos e interrupções no acesso à Internet são considerados eventos de alta severidade. Danos que incorram em re-instalação, reconfiguração ou perdas de dados sem backup e danos físicos com necessidade de substituição de equipamentos envolvidos também são inseridos nessa categoria.

- Ataques Críticos ou Incapacitantes - Ataques Críticos ou incapacitantes são todos aqueles ataques cujo acontecimento representaria grandes prejuízos ou causariam a finalização das atividades da entidade. Os ataques críticos são todos aqueles que afetam diretamente o negócio principal das entidades afetadas, e como tal, variam de cenário para cenário. Uma empresa financeira cujo cadastro de clientes furtado (com todas as informações pessoais desde nome completo até cartão de crédito, por exemplo), uma entidade de segurança nacional que tivesse seus servidores invadidos (posicionamento de tropas militares, por

exemplo) ou uma entidade formal que sofresse um ataque à sua reputação através de e-mails forjados (oferecendo ofertas duvidosas ou difamando outrem, por exemplo) são todos exemplos de ataques críticos.

Recentemente [TRE04, NOR04, MCA04] as empresas SCO Linux, Microsoft e Riaa sofreram ataques com este grau de severidade que foram contornados de maneiras próprias de cada entidade evitando assim a interrupção dos seus serviços e prejuízos para suas operações. Um exemplo de ataque incapacitante é o atentado de 11/09/2001, onde diversas empresas deixaram de existir com a queda das Torres Gêmeas de Nova York.

2.3. Formas de Ataque

Uma vez conhecendo os tipos de ataque é necessário saber como são feitos para poder finalmente proteger os sistemas contra os mesmos. Entender as formas de ataque e as ferramentas utilizadas são uma necessidade para se conseguir gerar ferramentas e técnicas de prevenção a novas ações.

Duas formas de ataque são caracterizadas: ataques manuais e ataques automatizados.

Ataques automatizados são mais comuns e responsáveis pela maioria das invasões e brechas em sistemas, enquanto ataques manuais são considerados potencialmente mais perigosos em seu escopo, devido à maneira de sua execução e à experiência necessária por parte do atacante para a execução de cada um [CAM97b, GER99, INN01, INN04].

2.3.1. Ataques automatizados

Ataques automatizados são aqueles que não demandam atenção humana para sua efetivação, podendo ocorrer apenas através da execução de scripts e softwares específicos para invasão. [MEI03]

Existem diversas formas de ataques automatizados: vírus, *worms*, cavalos de tróia e scripts de invasão.

- Vírus - são seções de código nocivo, que modificam programas originais através da inserção deste código no início dos arquivos afetados. Vírus podem se propagar através dos recursos computacionais pela execução de seus programas hospedeiros, afetando

assim novos alvos. São considerados ataques automatizados pois sua capacidade de replicação não depende da atividade do atacante, mas sim do atacado; quantos mais sistemas interagirem com o alvo infectado, maior será a ação dos vírus. Vírus são detectados através de suas assinaturas, particularidades de código conhecido, que programas específicos conseguem ler dentro dos arquivos afetados e efetuar a remoção apenas do código virótico, restaurando o arquivo afetado a sua condição normal. Vírus não podem ser executados e sua propagação está ligada à execução dos seus hospedeiros.

- *Worms* - diferem de vírus por serem programas completos, executáveis independentemente da existência de um hospedeiro. *worms* se propagam através de mensagens de correio eletrônico, conexões de rede e camuflados em arquivos aparentemente inocentes. A existência de *worms* também é endereçada pelos mesmos aplicativos que cuidam de infecções por vírus, porém para um *worm* não há correção, sendo a cura para a existência de *worms* em um sistema a deleção dos mesmos. São considerados ataques automatizados pela mesma razão que os vírus: capacidade de propagação e de causa de danos independente da interação do atacante.

- Cavalos de Tróia - são softwares aparentemente úteis e inofensivos, mas que em seu código contém seções nocivas que buscam burlar políticas e sistemas de segurança, gerando vulnerabilidades que possam ser exploradas posteriormente pelo atacante. Cavalos de tróia em geral não são detectados pela sua assinatura em arquivos contaminados (vírus) nem pela sua execução em sistemas contaminados (*worms*), mas pelos seus efeitos. Quando um sistema é contaminado por um cavalo de tróia, esta aplicação maliciosa abre uma porta que aceita conexões externas por onde o invasor irá efetuar seu ataque com sucesso, porta esta conhecida em jargão técnico com *BackDoor*. É através da monitoração destes efeitos que os programas antivírus conseguem encontrar a presença de cavalos de tróia em um sistema. A infecção por um cavalo de tróia se baseia em ataques de engenharia social onde o usuário pode ser exposto a estes riscos através de *sites* aparentemente idôneos, softwares e ferramentas condescendentes com pirataria de software e aplicativos de origem duvidosa. *Worms* podem conter em seu código um

componente de cavalo de tróia, permitindo que o atacante tenha a capacidade de infecção de um vírus com a abertura de brechas no sistema característica do cavalo de tróia.

- Scripts de invasão e Ferramentas de Exploração de falhas - são pacotes de softwares e instruções encadeadas para se fazer invasões a sistemas.

  Estes scripts são criados por indivíduos com alto grau de capacidade técnica para explorar amplas listas de fragilidades e falhas conhecidas em sistemas. Estas falhas e fragilidade em gral são expostas pelo próprio criador do software envolvido, e subsequentes remendos ou concertos são desenvolvidos para o software, com o objetivo de evitar a falha conhecida. Porém, nem todos os administradores têm tempo ou disposição para atualizar seus sistemas [CAM97b], fazendo com que vulnerabilidades passem despercebidas pelo administrador. Os criadores de scripts, então, criam ferramentas e receitas que visam exatamente atacar estas falhas conhecidas, buscando tomar o controle do sistema afetado.

  Uma vez que as ferramentas e scripts estão criados, os mesmos são disponibilizados na *internet*, em geral de maneira ruidosa para chamar a atenção de diversos possíveis atacantes que vêem nos scripts ferramentas para efetuar os ataques que eles próprios não conseguem, por incapacidade técnica.

  Em geral, uma ferramenta automática terá um ciclo de vida muito simples: 1-Escolha aleatória de um endereço; 2 - Sondagem do endereço; 3 - Descoberta de serviços ofertados pelo *host*; 4 - Comparação dos serviços ofertados pelo *host* com os serviços que podem ser explorados pela ferramenta; 5 - Exploração das falhas e tomada do controle.

  Considerando-se o tamanho da *internet* e o ciclo acima apresentado, poderia se dizer que as chances de sucesso de uma ferramenta automatizada são pequenas. Porém, o computador doméstico e a velocidade de conexão à Internet, somado com o método de dispersão das ferramentas automáticas, torna este ataque automatizado o mais perigoso de todos. Todos os dias equipamentos conectados à Internet apresentam centenas de sondagens, de diversas fontes diferentes. Esse grande número de tentativas de invasão e sondagens diárias dá-se por um motivo simples: o ciclo apresentado acima, em um computador doméstico moderno com *internet* de alta velocidade, demora apenas alguns segundos; logo, o computador e o usuário comum podem executar sondagens a um grande número de dispositivos na *internet* a cada dia. Como o método de invasão é totalmente

automático, o atacante pode apenas executar suas ferramentas de invasão em segundo plano, permitindo que as invasões aconteçam automaticamente.

Esses invasores que se valem de ferramentas automatizadas e de receitas prontas de invasão são conhecidos como *script kiddies*, de modo que o termo *script kiddie* acaba inferindo uma característica negativa, amadorística ao invasor. Dentro da comunidade que invade sistemas os que se valem das ferramentas automáticas são menos respeitados e reconhecidos que os atacantes que usam formas manuais de invasão.

O amadorismo dos invasores cria um paradoxo interessante: o conhecimento técnico necessário para utilizar uma ferramenta automática é de mediano a pequeno, mas a quantidade de invasões feitas automaticamente é bastante superior à das invasões manuais. O invasor automático não busca um sistema específico, mas qualquer sistema; para o invasor automático o objetivo é completar uma invasão com sucesso, sem critério de qual seja o sistema comprometido; se o sistema for interessante ao invasor, tanto melhor. Isso faz com que a grande quantidade de invasões automáticas represente muito menos prejuízo real que as invasões manuais [CAM97a, CAM97b, GER99].

2.3.3 Ataques manuais

São diferentes em motivação e em perfil do executante dos ataques automáticos. Um atacante manual escolhe cuidadosamente um alvo e um objetivo antes de selecionar a técnica de invasão. Os motivos por trás do ataque podem variar, desde a simples pichação ideológica até a mais sofisticada fraude eletrônica bancária.

Uma vez que o alvo tenha sido escolhido e o objetivo seja determinado, o atacante manual irá sondar a rede escolhida minuciosamente, testando todos os sistemas alcançáveis em busca de qualquer falha que não tenha sido remediada. Para o atacante manual, basta uma falha não atendida para que o ataque possa ser efetuado.

Os atacantes manuais eventualmente constroem suas próprias ferramentas, para automatizar seu trabalho. Provavelmente, uma vez que a ferramenta tenha sido utilizada, esta será disponibilizada na *internet*, aumentando os recursos dos amadores automatizados nas suas invasões.

Os atacantes manuais são mais nocivos exatamente por terem ao seu lado o conhecimento técnico que falta aos invasores automáticos, permitindo a eles a construção de

ferramentas para seus desígnios específicos, ferramentas estas que por vezes podem atacar fragilidades pouco conhecidas ou ainda inéditas, minando todos os esforços em assegurar um sistema.

2.4. Ferramentas de ataque

Com as ferramentas de ataque devidamente classificadas, pode-se enumerar os efeitos particulares das ferramentas, e citar alguns exemplos de ferramenta capaz de atingir os objetivos dos invasores.

2.4.1. Classificação de ferramentas de ataque por efeito

Um ataque pode ter diversos efeitos adversos em um sistema. Negação de serviço, obtenção de acesso indevido, aumento indevido de direitos e até controle total do sistema afetado. Serão enumeradas as ferramentas mais comuns para cada tipo de ataque e técnica utilizada pelos invasores.

- Negação de serviço - é um efeito que os ataques podem causar nos sistemas afetados. Por definição, negação de serviço é "um conjunto de ações que leva à indisponibilidade temporária de um determinado recurso computacional em um sistema" [HAC04].

  Os ataques de negação de serviço podem ser efetuados a partir de ferramentas simples, como o *ping* dos sistemas operacionais modernos, até vírus e *worms* que obrigam os sistemas afetados a tentarem numerosas conexões com alvos pré-determinados em janelas de tempo pré-determinadas [MEI03].

  O funcionamento de um ataque deste é bem simples: o atacante busca tomar tantos recursos quanto possível do atacado, usando desde conexões normais até redirecionamento de IP para que o atacado gaste recursos processando os pacotes IP, até que o atacado perca toda a capacidade de atender as requisições, sendo "afogado" em um volume muito grande de informações. Porém a capacidade de um atacante solitário conseguir afetar um grande servidor de maneira definitiva é pequena, causando uma evolução nesta técnica, que passou a ser conhecida como Negação de Serviço Distribuída (DDoS - *Distributed Denial of Service*). A nova técnica consiste em aumentar a quantidade de atacantes que tentam

afetar o mesmo alvo simultaneamente, fazendo com que bandas de dados e capacidade de processamento sejam sobrepujados mais rapidamente. Os DDoS´s então, podem ser iniciados manualmente por um grupo organizado de atacantes, ou através de vírus e *worms* que infectam um grande número de sistemas na Internet, contendo em seu código instruções para efetuar ataques dentro de uma janela de tempo específica contra sistemas específicos da Internet [CAM97b]. Ataques de negação de serviço, no entanto, também podem ser lançados internamente em um sistema, de modo que a inserção de código nocivo venha a prejudicar a capacidade de processamento do alvo, requisitando, por exemplo, serviços recursivamente a uma máquina, de modo que a mesma pare de responder.

Recentemente um *worm*, chamado de MYDOOM, varreu a *internet* infectando usuários do software de correio eletrônico *Ms-Outlook*. Esse *worm* continha instruções para efetuar um ataque de negação de serviços contra o *site* da empresa de software SCO Unix (www.sco.com) no espaço de tempo de uma semana. Esse ataque obrigou a SCO a mover todo o seu *site* para outro endereço, causando prejuízos e transtornos para o atacado. Este *worm* causou este prejuízo através da aplicação da técnica de DDoS [TRE04, NOR04, MCA04].

Outras ferramentas: *Portfuck, Smurf & Fraggle, Teardrop, Netdrop, Syndrop, b0nk!* .

- Enumeração de portas - são softwares auxiliares para o invasor. Uma vez apontados para um sistema cujo o IP seja conhecido, estes softwares tentam conexões em todas as suas portas, buscando identificar serviços e versões de software para reportar ao invasor. Uma destas ferramentas é o *Netcat*, que conta entre suas funcionalidades a capacidade de ser apontado para um bloco de endereços, automatizando o trabalho de se identificar sistemas e portas disponíveis em uma rede.

  Outras ferramentas: *Bindery, b1nd1ng, Epdump, l3gion, netviewX, nslist, snlist, userDump, Userinfo*.

- Obtenção de acesso - a obtenção de acesso também é um objetivo do invasor. Por obtenção de acesso pode-se entender que o atacante vai ter ao seu alcance um nome de usuário e uma senha válida no sistema, sendo que esse nome de usuário não obrigatoriamente é o nome de um usuário com privilégios de administrador do sistema,

pois muitos ataques podem ser efetuados para a elevação de privilégio. Ferramentas de obtenção de acesso são quebradores de senha, farejadores de rede e softwares que buscam conexões em sistemas cuja configuração de compartilhamento de dados não esteja segura. Um quebrador de senha, por exemplo, *L0phtcrack's* vai tentar efetuar *logons* em um servidor através de uma combinação de métodos de geração de senhas, tanto por força bruta, onde o software tentará "adivinhar" a senha, quando pela utilização de dicionários de palavras, para efetuar tentativas baseadas em comportamento humano, levando em conta a mentalidade do usuário ao cadastrar suas senhas.
Outras ferramentas: *Netbios Auditing Tool, Nwpcrack, SMBGrind, Sniffit*.

- Aumento de privilégio - aumento de privilégio é um passo das ações de um ataque. O atacante busca o aumento de privilégio a partir de uma conexão que já se encontra efetuada no sistema a ser comprometido, conexão esta feita através de um usuário legítimo do sistema (o usuário e senha podem ter sido obtidos anteriormente com o uso de ferramentas de obtenção de acesso). O aumento ou elevação de privilégio ocorrem por falhas em serviços ou softwares disponíveis no sistema, ou por problemas com a configuração de segurança do sistema. A ferramenta *getadmin2k* é um utilitário de linha de comando para Windows 2000 que permite, em se executando o utilitário, acessar diretamente as informações contidas no *active directory*, permitindo alterar qualquer senha facilmente. É uma ferramenta poderosa, apesar de deixar pistas claras da invasão do sistema.

- *Backdoor* - são softwares que uma vez instalados em sistemas, os deixam vulneráveis à conexão remota e controle total da máquina, através da abertura de portas de conexão. *Backdoors* costumam ser inseridos em ferramentas automáticas de invasão pelos invasores manuais e construtores de ferramentas de invasão que fazem isso com o intuito de tornar os sistemas utilizados pelos *script kiddies* susceptíveis às invasões dos atacantes experientes. Ferramentas como o *NETBus* permitem diversos graus de interação com o sistema afetado, desde acesso aos dispositivos físicos (por exemplo: imprimir remotamente uma mensagem ameaçadora na impressora local do atacado) até a desativação do sistema afetado.
  Outras ferramentas: *Jcmd, NTFSdos, Pandora, Revelation*.

- *Rootkits* - são as ferramentas prediletas do script *kiddie* [GER99]. São conjuntos de softwares que quando apontados a um *host*, trabalharão sozinhos até conseguirem acesso administrativo ao sistema ou esgotarem suas capacidades. Muitos *rootkits*, por serem feitos para serem facilmente usados, contém *backdoors*, que acabam deixando vulneráveis os sistemas dos invasores. Os rootkits não servem apenas para se obter acesso administrativo a um host. A maioria dos rootkits contém diversas ferramentas e aplicativos, para serem usadas em uma invasão.
  Outras ferramentas: *Cygwin32 , Xrootkit*

- Apagadores de Rastros - uma vez efetuado um ataque, o atacante precisa partir sem deixar rastros de que esteve ali, deixando apenas os efeitos da invasão. Existem ferramentas automatizadas, mas não 100% eficazes que podem modificar arquivos de *log*, alterar históricos de sistema operacional e até recalcular *checksum* de arquivos, evitando que os arquivos modificados pareçam modificados. *Wipe* e *zap* são utilitários de linha de comando que funcionam assim.

- Sondas (*Scanners*) - uma sonda é um software capaz de testar um *host* ou conjunto de *hosts* contra um grupo de vulnerabilidades conhecidas. Esta capacidade pode ser usada em duas maneiras antagônicas: ao passo que o administrador pode se utilizar desta ferramenta para sondar a sua rede por vulnerabilidades em seu sistema com o intuito de repará-las, o invasor efetuará a sondagem buscando um alvo em potencial. Há vários scanners, tanto comerciais quanto livres, mas todos funcionam sobe o mesmo princípio de sondar o IP, sondar a porta, verificar sistema em execução na porta e reportar.
  Outros *softwares*: *scan, Solarwinds, strobe, upscan*

2.5. Técnicas de prevenção

A melhor prevenção contra invasões é gerar e respeitar uma política de segurança que abranja atualização de softwares, controle de senhas, controle de recursos disponibilizados na rede e acesso físico ao sistema e pelo uso de ferramentas de proteção de sistemas como *Firewalls* e IDS´s;

2.6. Contra ataque

O contra-ataque para casos de invasões não pode ser dado com os mesmos métodos empregados pelo invasor. O contra-ataque deve ser feito de maneira legal, através de um processo civil utilizando técnicas forenses para extrair dos sistemas invadidos provas do ataque e do dano causado, evitando assim que o atacante comprometa novos sistemas. Será avaliada para os fins forenses a tecnologia de *Honeypots*.

2.7. Tipos de Atacante e Motivações

Já foram vistas as ferramentas utilizadas para se efetuar invasões. Por trás do uso destas ferramentas está uma pessoa, que em geral é um adolescente, do sexo masculino, com capacidade intelectual acima da média, que busca invadir sistemas ou para ganho próprio, através de fraudes bancárias, por exemplo, ou para ganho de notoriedade e satisfação pessoal, ainda que esta notoriedade seja apenas reconhecida por um seleto grupo[HON02a, MEI04, HAC04].

2.8. Conclusão

Uma vez vistas as formas de ataque e a facilidade com a qual os atacantes podem lançar ataques à esmo, juntamente com a falta de propósito na maioria dos ataques e a fragilidade dos sistemas que são pouco ou mal administrados.

Conhece-se então dois tipos de invasores, ambos igualmente perigosos: o *script kiddie* e o verdadeiro *hacker.* Ambos são perigosos, pois o primeiro se encontra com facilidade na *internet,* sendo muito numeroso e o segundo tem ao seu alcance ferramentas e experiência computacional.

Capítulo 3

Ferramentas de Segurança

Ferramentas de segurança são recursos utilizados para fornecer ao ambiente da rede maior segurança na geração, tráfego e administração dos dados.

Como visto no capítulo 2, a maioria dos sistemas computacionais está exposta a um grau variável de risco de ataques, sendo estes ataques cada vez mais complexos e com possibilidades de dano maiores. Um administrador pode dispor de 3 ferramentas básicas: *firewalls*, sistemas de detecção de intrusão e *honeypots* [SIM99], discutidos no próximo capítulo.

3.1. Firewalls

Firewalls foram criados para administrar e oferecer mais segurança ao tráfego que ocorre entre as redes. O nome *firewall* é usado em alusão às paredes corta-fogo que, em caso de incêndio, mantém ambientes isolados das chamas. Esta comparação é usada para demonstrar a implementação da *firewall*, dando idéia de que o ambiente externo a ele é inseguro (chamas) e o interno é seguro.

Todos os *firewalls* são implementados na transição entre redes pelas quais os pacotes precisam trafegar, analisando então a conformidade do pacote, conformidade esta que é determinada através de comparações internas do *firewall*. Estas comparações são baseadas em listas de regras que se encontram dentro do *firewall* e que determinam a validade ou não de um segmento específico de tráfego.

Por definição diz-se que *firewalls* são pontos de passagem entre redes que restringem tráfego de dados entre redes diferentes, criando um perímetro de proteção e oferecendo maior segurança para a rede dita "interna", enquanto protege os recursos da rede interna das ameaças de segurança da rede externa, tipicamente protegendo uma rede menor (LAN corporativa ou um *host* simples) de tráfego indevido da rede maior (Internet, por exemplo) e restringindo o acesso dos usuários da rede interna apenas aos recursos a eles autorizados na política de segurança da organização [SHI00], enquanto gera registros da atividade da rede e gera notificações quando condições determinadas de segurança são atingidas.

"Todo o *Firewall* é tão seguro quanto sua configuração permitir que seja"

"*Every Firewall is as safe as its configuration allows it to be*" [CHE97]

Os *firewall* podem ser classificados sob diversos aspectos: funcionamento, interação com o tráfego passante e arquitetura de implementação[GAR99, CHE97, FIR02].

3.1.1. Classificação de *Firewall* segundo o funcionamento

Tanto *firewalls* de software quanto de hardware precisam ter configurados em seus sistemas um conjunto de ações a serem tomadas quando acontecerem condições específicas no tráfego da rede. A forma de configuração destas ações é uma das características fundamentais dos *firewalls*

a) Funcionamento baseado em regras - *firewalls* cujo funcionamento se baseiam em regras são ferramentas que, para cada pacote que passa pelas interfaces de rede do *firewall*, um conjunto de testes é efetuado baseado em uma lista de regras configuradas pelo administrador como endereço de origem, endereço de destino, porta, tamanho do pacote, conteúdo do pacote, tamanho do frame, data e hora do tráfego. [CHA97, CHE97]

Firewalls baseados em regras oferecem maior capacidade de filtragem passiva, permitindo a um administrador maior controle sobre o tráfego, porém demandam muito mais esforço para implementação, configuração e administração. Este tipo de funcionamento também não protege a rede de tráfego cuja configuração não esteja feita corretamente.

O pacote IPFW é um exemplo de *firewall* baseado em regras muito usado atualmente. [CHA97].

b) Funcionamento baseado em aplicações - *firewalls* de funcionamento baseado em aplicações são construídos visando dar mais facilidade ao administrador do sistema, automatizando tarefas de criação de regras em conjuntos de aplicações. Aplicações são softwares específicos cujo tráfego possui uma assinatura conhecida pelo *firewall*, e a própria implementação do software interno do *firewall* cuidará de administrar as mudanças de estado, portas e tipos de dados para as aplicações autorizadas. Para que o *firewall* se mantenha seguro, o administrador necessita interagir com o *firewall* sempre que uma nova aplicação for trafegar na rede, e caso seja criada uma aplicação nova cujos dados não sejam conhecidos pela implementação do *firewall*, este tráfego estará desprotegido ou bloqueado.

O *Checkpoint Firewall-1* e o *Microsoft Internet Security and Aceleration* são exemplos de *firewalls* baseados em aplicações [CHE04a, MIC04].

3.1.2. Classificação de *Firewall* segundo o Grau de Interação com o tráfego

Quando implementa-se um *firewall* em uma rede, presume-se que o *firewall* será o único ponto de conexão entre a rede protegida e o meio externo. Logo considera-se que todo o tráfego que ocorre entre as redes passa obrigatoriamente pelo *firewall*. Como estes pacotes sofrem uma análise do software de *firewall*, podemos também classificar a ferramenta conforme o seu grau de interação com os pacotes. [GAR97, WIL03]

a) Filtros de Pacotes - são técnicas de *firewall* antigas e menos eficazes que as técnicas de *firewall* mais modernas. São implementados na camada de rede de um dispositivo mediador que impede, em nível de endereço e porta, o acesso aos dispositivos que estejam configurados em sua lista de regras. Esta lista de regras precisa ser criada e administrada manualmente, porém como não há monitoramento de estado da conexão e nem inspeção de pacotes, pacotes de dados destinados a endereços e portas liberados na rede, porém cujo conteúdo seja nocivo, alcançarão o endereço a ser atacado como se não houvessem ferramentas de segurança na rede.

Filtros de pacotes são implementados como a primeira camada de segurança de uma rede, geralmente diretamente no sistema operacional do roteador de conexão externo, permitindo que apenas as portas autorizadas explicitamente trafeguem na rede (para

configurações restritivas) ou negando o acesso a portas específicas (para configurações permissivas).

Por oferecerem regras simples do tipo "autoriza ou nega o acesso", filtros de pacotes tem boa performance, com pouco impacto na velocidade da rede.

Um exemplo genérico de regra de filtro de pacotes seria:

"*External: Deny-Al*"

"*External: Allow port* 80 *to host* 192.168.7.33"

"*External: Allow port* 21 *to host* 192.168.7.33"

Esta regra negaria primariamente todas as conexões entrantes (configuração restritiva) na rede, permitindo apenas que a porta 80 (servidor WEB) e a porta 21 (servidor FTP) fossem acessados, porém a rede estaria desprotegida contra ataques em nível de aplicação contra estes 2 serviços.

b) *Gateway* de Camada de Aplicação - os *gateways* de camada de aplicação, quando comparados a filtros de pacotes, aumentam a segurança dos *firewalls* através da análise de todas as camadas da aplicação, trazendo o aspecto de análise de informações ao tráfego de redes.

Este objetivo é atingido através da utilização de softwares intermediários, chamados de *Proxy* de Aplicação. *Proxys* de aplicação são servidores que são executados no *host firewall*, servindo para receber as conexões internas em uma das suas interfaces e efetuando conexões externas em uma outra interface. Esses softwares intermediários quebram a arquitetura cliente servidor, efetuando conexões seguras entre clientes locais de um lado e servidores remotos no outro lado. Há dois problemas fundamentais nesta arquitetura: primariamente a escalabilidade é limitada, pois todas as conexões efetuadas pelos clientes internos requisitam recursos do servidor de *firewall*, facilmente causando uma situação de concorrência no recurso do *firewall*; adicionalmente, todas as aplicações acessadas pelo cliente necessitam de um servidor intermediário apropriado, fazendo com que o suporte a novas aplicações e serviços seja limitado.

As vantagens desta implementação são contrapostas pela concorrência de recursos no firewall e pelo suporte limitado a novas aplicações, causando desde lentidão no tráfego até impossibilidade de acesso a serviços externos da rede. [CIS03]

c) Inspeção *Statefull* - é considerada uma evolução dos filtros de pacotes e não dos *Gateways* de camada de aplicação. A inspeção *statefull* ocorre também na camada de redes, onde cada pacote trafegado será analisado pelo seu conteúdo, anotando o tráfego em tabelas de estado que contém informações como horário original, IP de abertura de conexão, IP de destino, porta de destino e conteúdo de alguns pacotes. A análise destas tabelas de estado então, combinada com os filtros de pacotes, aumenta a segurança da rede. Se com filtros de pacotes havia a autorização de tráfego apenas para as portas liberadas, com a inspeção de pacotes temos a análise do tráfego em busca de conteúdo que seja inseguro ou nocivo. Inspeção *Statefull* também é a maneira mais apropriada de se ter o funcionamento baseado em aplicações, visto que várias aplicações não tem regras facilmente configuráveis, por exemplo, serviços baseados em RPC cuja alocação de portas é dinâmica, causando transtornos para a administração de redes.

A inspeção *statefull* não quebra a arquitetura cliente servidor, e o custo de processamento causado pelo tráfego é menor do que o dos *gateways* de camada de aplicação, porém o funcionamento correto da inspeção requer que o software de análise das tabelas de estado esteja sempre atualizado, para que novas e mais sofisticadas tentativas de ataque não consigam sucesso em afetar os sistemas da rede protegida, requerendo assim um grau de administração constante, assim como os serviços de anti-vírus [CHE04c, GHU02].

d) *Application Inteligence* (Aplicações Inteligentes) - é um conjunto de regras avançadas, ainda em fase de pesquisa e desenvolvimento que busca aprimorar os *firewalls* baseados em aplicações com inspeção *Statefull*, através da aplicação de medidas de segurança criadas especificamente para bloquear ataques sofisticados contra aplicações. *Application Inteligence* também funciona na camada de redes sem quebrar a arquitetura cliente-servidor, porém possui capacidades relativas a várias camadas da pilha OSI, aprimorando a segurança da rede toda, visando fornecer o recurso de *firewalls* também para redes convergentes.

As camadas cobertas pelo *aplication inteligence* são aplicação e apresentação, sessão, transporte e rede.

Na camada de aplicação e apresentação, um *firewall* com recursos de *application inteligence* pode monitorar e bloquear ações nocivas como cabeçalhos corrompidos em URLs, restringir comandos HTTP inseguros, efetuar busca de DNS reverso, restringir e reforçar a formatação correta de comandos do tipo MAIL e RCPT, bloqueio de múltiplas requisições de

cabeçalhos e comandos do tipo "*content-type*", restringir transferência entre zonas de DNS, reforçar os parâmetros mandatórios do protocolo H.323 (Protocolo de Definição do DNS) [SHI00].

Na camada de sessão, busca-se prevenir mapeamentos de portas RPC, validação de validade de certificados digitais trocados, e a prevenção de vulnerabilidades em segredos pré-compartilhados.

Na camada de transporte os fragmentos de pacotes como FIN sem ACK e SYN são bloqueados, assim como verificação de campos de tamanho de pacotes UDP, verificação de requisições e respostas UDP, reforçar o *3-way-handshake*, bloqueio de *Fingerprint* de sistemas operacionais.

Na camada de rede o *application inteligence* bloqueia pacotes ICMP com tamanho excessivo, restringe fragmentação IP-UDP, avalia os pacotes para que seu tamanho seja realmente o valor postado no campo *Header* do pacote.

Application inteligence encontra-se ainda em pesquisa e em fase de aprimoração, logo hoje ainda há poucos softwares com esta capacidade e o custo em processamento de todas estas características ainda é muito alto [FIR02, CHE04b].

3.1.3. Classificação de *Firewall* segundo a Arquitetura

A arquitetura de implementação do *firewall* também é um fator de análise importante. Por arquitetura de implementação entende-se a posição relativa que o *firewall* possui em relação às redes protegidas, aos recursos de conexão e às redes externas.

A análise da arquitetura de implementação de *firewalls* traz os conceito de *Bastion Host* e DMZ, que precisam ser explicados: *Bastion Hosts* são servidores cuidadosamente implementados e de alta segurança que mantém contato com a rede externa, conseqüentemente estando expostos aos riscos de ataques. DMZ´s são áreas intermediárias entre a rede interna e externa onde os servidores que recebem tráfego externo estão hospedados de maneira separada da rede interna de uma corporação.

a) *Dual Homed Host* - são *Bastion hosts* nos quais o *firewall* é implementado, expondo sua interface externa e desativando o roteamento entre as interfaces externas e as interfaces

internas. Assim, conexões externas chegariam até o *Bastion host* apenas, e conexões internas teriam que obrigatoriamente passar pelo *Bastion Host*.

Como mostra a figura 1, entre a *internet* e um *Dual Homed host*, poderia ser implementado um conjunto de filtros de pacotes, no roteador mais próximo, por exemplo, para diminuir as possibilidades de ataques, diminuindo a necessidade de filtros de pacotes no próprio *bastion host*.

Esta arquitetura gera um ponto focal na rede, com vantagens e desvantagens: O tráfego centralizado permite uma administração focalizada em um único *host*, porém o excesso de tráfego em um único *host* pode causar condições de concorrência e caso o *bastion host* seja tomado por um invasor, toda a rede estará vulnerável à ataques.

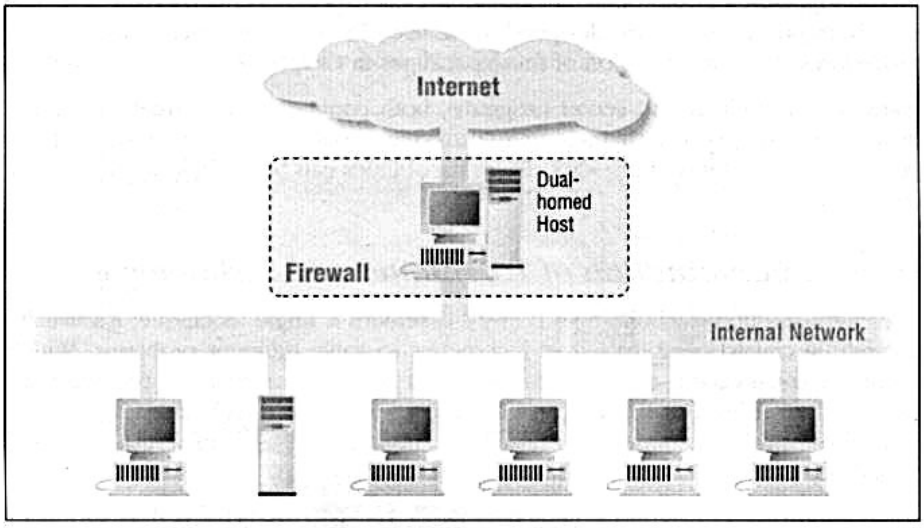

Figura 1 - Arquitetura Dual Homed Host

b) *Screened Host* – conforme a figura 2, essa arquitetura apresenta uma conexão externa com ligação apenas a um *host* interno, que é um *bastion host*. Este *host* estaria conectado à rede interna, e não entre as redes, e o *firewall* teria que ser implementado no roteador, porém haveria pacotes externos entrando na rede local. Apesar de ser aparentemente menos seguro do que os *dual homed hosts*, este tipo de arquitetura permite o acesso aos serviços do *bastion host* sem causar condições de concorrência na rede, uma vez que todo o trabalho de filtragem e análise de tráfego ocorre no roteador.

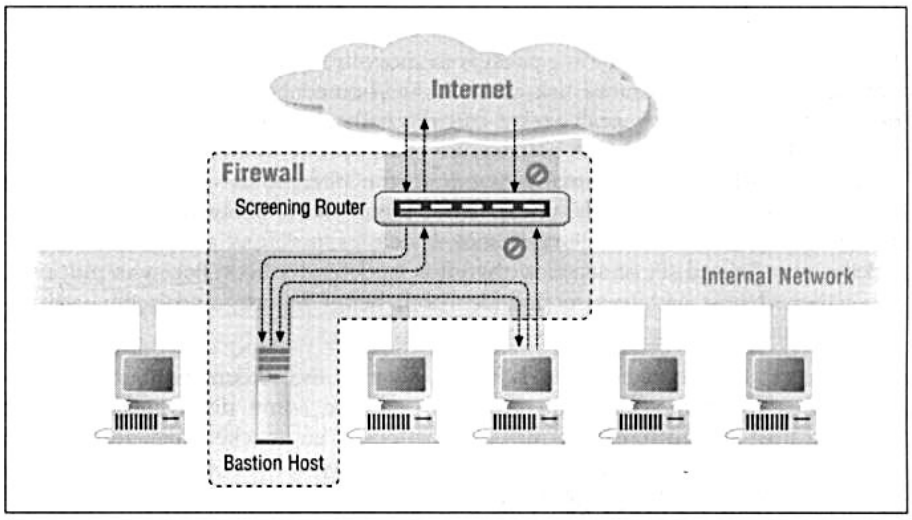

Figura 2 - Arquitetura Screened Host

c) *Screened Subnet* - adicionam mais uma camada à segurança de redes através da utilização de DMZ para hospedagem dos *bastion hosts* e de um *firewall* adicional que separa a rede interna da DMZ. Em uma *screened subnet* um *firewall* separa a rede externa da DMZ que hospeda os serviços que podem ser acessados pela rede externa, como por exemplo, um servidor de correio corporativo. Dentro da DMZ está também o *bastion host* que contém o *firewall* que dá acesso da rede interna à DMZ e roteia as requisições da rede interna para o roteador da DMZ.

Esta arquitetura de *screened subnets* permite maior segurança a uma rede, uma vez que as tentativas de invasão serão efetuadas contra *bastion hosts* que não possuem acesso à rede interna, e que o *bastion host* de saída da rede é protegido por filtros de entrada no roteador externo.

3.1.4. Problemas inerentes da ferramenta

Todas as implementações de *firewall* apresentam fragilidades específicas; filtros de pacotes não conseguem proteger portas autorizadas, *firewall* de aplicação requer muito recurso computacional, inspeção *statefull* precisa de bases de dados de assinaturas atualizadas diariamente e *application inteligence* ainda se encontra em fase de pesquisa e requer muito poder de processamento.

3.2. IDS - Sistema de Detecção de intrusão

IDS´s são ferramentas de segurança automatizadas que visam proteger a rede através da monitoração do tráfego de dados, gerando alarmes e informando sobre ações que violem a política de segurança de uma organização [INN01].

A tecnologia dos IDS difere em escopo e em objetivo do *firewall* e serve como complemento à essa ferramenta. Ao passo que os *firewalls* situam-se às margens das redes, nos pontos de transição entre uma rede e outra, os IDS precisam estar imersos na rede, com grande visibilidade para todos os pacotes que trafegam no meio. Onde os *Firewalls* buscam evitar ataques, IDS são criados tomando por base o fato de que, mais cedo ou mais tarde, haverá um ataque que será capaz de burlar o *firewall* e penetrar na rede interna, onde o IDS deve detectar esta intrusão e agir de acordo para emitir alarmes [SNO04].

Se, análogamente, um *firewall* é a porta de um cofre, o IDS é o sensor de movimento que monitora a sala do cofre. A porta do cofre protege seu interior do meio externo, mas o sensor de movimento, ao detectar uma presença na sala do cofre, dispara os alarmes apropriados [ROB98].

Um IDS então pode ser resumido como "um sistema digital capaz de monitorar o tráfego de um segmento de rede e classificá-lo como seguro e inseguro, e, em caso de apontamento de tráfego inseguro, determinar a insegurança do mesmo e alertar a administração da rede" [IDS04].

3.2.1. Características desejáveis em IDS´s

Para que se possa considerar a implementação de IDS´s como confiável, é necessário exigir da implementação um conjunto de características desejáveis, propriedades que o IDS, como sistema automatizado e de alarmes, deva possuir [CDI04]:

- Simplicidade de configuração - os IDS´s, preferencialmente devem ser simples e rápidos de se configurar, demandando pouco tempo para sua implantação numa rede, evitando que se desperdice recursos importantes na instalação de uma ferramenta auxiliar de segurança.

- Simplicidade de administração - estas ferramentas devem gerar alarmes que sejam facilmente interpretados pelo administrador do sistema, evitando complicações para a parte humana do processo.
- Independência de operação - a ferramenta não deve demandar operação manual, e salvo atualizações e intervenções administrativas, deve ser capaz de atender ao seu ciclo de vida sem interação humana (por ciclo de vida temos o conjunto de ações "captura de dados, análise, decisão, ação apropriada").
- Tolerância a falhas - como é uma ferramenta automatizada e não assistida, espera-se que a ferramenta não tenha travamentos nem chegue em condições de parada; caso estas condições aconteçam, a ferramenta deve ignorar o erro, assumir a falha encontrada e resumir sua operação normal. Nesta característica também inclui-se a capacidade de retorno das operações com uma reinicialização do sistema.
- Segurança – o sistema deve ser seguro, de modo que tentativas de subversão ao seu sistema de análise de invasões sejam infrutíferas; Também não devem estar hospedados na mesma máquina quaisquer serviços que possam vir a ser explorados por um ataque direto ao sistema de detecção de intrusão. Por estar inserido na rede, os sistemas de detecção de intrusão estão muito próximos dos *hosts* em produção e em caso de comprometimento do mesmo, o grau de risco é muito grande para os sistemas de produção, pois o *host* que executa o IDS está diretamente conectado à rede, fazendo com que o comprometimento do mesmo exponha toda a rede ao invasor.
- Baixo impacto no funcionamento do sistema - os sistemas de detecção de intrusão devem funcionar como monitores de tráfego, e não devem gerar tráfego na rede que venha a prejudicar a operação normal dos sistemas, e nem estar instalados em máquinas de produção pois a análise de tráfego demanda grande carga de processamento.
- Analisar padrões - seja através de bases de dados de assinaturas com informações sobre intrusões ou através da configuração meticulosa, a ferramenta de detecção de intrusões deve ser capaz de separar, entre todo o volume de tráfego de um segmento específico de rede, aquelas informações que podem constituir riscos para a segurança da organização.

- Discrição - um sistema de detecção de intrusão deve se manter em modo promíscuo em uma rede, ouvindo requisições e respondendo ao menor número possível de pacotes, de modo que um possível invasor não seja alertado da presença do sistema automatizado. Assim como as câmeras de vigilâncias são pequenas e difíceis de se distinguir, os IDS devem ser discretos e sua operação na rede não deve deixar pistas de que os há monitoração nos segmentos de rede.
- Resistente a erros de monitoração - um sistema automatizado de detecção de intrusão é suscetível a três tipos de erro: Falsos Positivos, Falsos Negativos e erros de subversão. Falsos Positivos são ocorrências de tráfego que a ferramenta classifica como ataques, mas não são; Falsos negativos são ocorrências em que o tráfego é uma ameaça, mas a ferramenta não foi capaz de captar ou interpretar corretamente e erros de subversão são ataques diretos à ferramenta que geram falsos negativos ou que fazem com que a ferramenta perca a capacidade de interpretar tentativas de invasão corretamente.

3.2.2. Classificação de IDS

Classificamos as ferramentas de detecção de intrusão com relação à sua posição perante a rede (NIDS e HIDS) e com relação ao seu escopo e tempo relativo de funcionamento (SIV e LFM).

3.2.2.1. NIDS

Sistemas de detecção de intrusão baseados em rede (*Network based Intrusion Detection Systems*) são ferramentas IDS que ficam hospedadas em máquinas de um segmento específico da rede com o objetivo de monitorar todo o tráfego de rede, analisar o conteúdo do tráfego e emitir alarmes caso o tráfego apresente risco, expondo origem, destino e conteúdo do pacote, de maneira a simplificar o trabalho de análise do administrador.

Um NIDS funciona como um *sniffer* de redes, em modo promíscuo, e possui uma série de requisitos para poder ser eficiente; o NIDS precisa ter visibilidade para todo o segmento alvo, o hardware disponibilizado para a ferramenta precisa ter capacidade de processamento equivalente ao tráfego do segmento, o sistema de detecção precisa ter sua base de

conhecimentos atualizada e finalmente um lugar seguro onde armazenar os históricos de registros do tráfego marcado como ataque.

A ferramenta *SNORT* é um exemplo bem explorado de NIDS [SNO04]

3.2.2.2. HIDS

Sistemas de detecção baseados em *hosts* tem um escopo mais definido e tendem a ser utilizados em ambientes controlados, geralmente sendo aplicados em cenários onde o tráfego gerado pelos usuários não é tão importante quanto o tráfego em servidores de missão crítica.

Sistemas de detecção de intrusão baseados em *hosts* então protegeriam apenas o *host* onde estão implementados, e monitorariam todo o tráfego gerado neste servidor.

Existem 2 características negativas em um HIDS: a coleta e análise de dados requerem tempo e poder de processamento, tempo esse que vem da concorrência contra os processos de missão crítica da organização; os históricos de registro de tráfego nocivo precisam ser armazenados em um local separado do HIDS, para evitar que um eventual invasor bem sucedido possa tomar o controle do *host* do IDS e ainda alterar as provas da sua invasão.

A ferramenta *SEBEK* é um exemplo de HIDS. [SEB03]

3.2.2.3. SIV

Existem ferramentas que podem ser executadas em hosts para se verificar se, em caso de invasão, o invasor efetuou alguma modificação em comandos críticos do sistema, deixando para trás marcas da invasão. Um SIV (*System Integrity Verifier*) é uma ferramenta capaz de verificar a integridade do sistema invadido através de um banco de assinaturas e da comparação desse banco de assinaturas com os arquivos monitorados do sistema afetado. Ferramentas SIV podem funcionar em tempo real, verificando os arquivos monitorados em tempos pré-determinados ou ainda funcionar conforme a demanda do administrador do sistema.

O software *Tripwire* é um exemplo de SIV, funcionando tanto baseado em bancos de assinaturas locais quanto conectando em servidores remotos de assinaturas. [ROB02]

3.2.2.4. LFM

Assim como os NIDS que monitoram o tráfego de rede, ferramentas automáticas de monitoração de registros históricos podem ser aplicadas para se verificar de sistemas bem conhecidos da rede, buscando informações que caracterizem mal uso dos sistemas. Monitores de *logs* podem incorrer no problema de apresentarem demora para analisar todo o tráfego, por vezes deixando de emitir os alarmes apropriados em tempo de se evitar danos aos sistemas de rede.

O software "*swatch*" é um exemplo de um LFM [ROB02].

3.2.3. Métodos de detecção de intrusão

Uma vez conhecidas as características da ferramenta de detecção de intrusão, é possível se classificar a ferramenta de acordo com o método utilizado pela mesma para caracterizar o grau de periculosidade de um fragmento de tráfego específico.

Existem duas maneiras de se classificar os mecanismos de tomada de decisão de um sistema de detecção de intrusão: detecção de intrusão baseada em anomalia e detecção de intrusão baseada em conhecimento [ROB98, IDS04, RNP99].

3.2.3.1. Detecção de intrusão baseada em assinatura

Considera-se como o conhecimento de uma ferramenta automatizada de detecção de intrusão todas as informações às quais a ferramenta tem disponibilidade para comparação com o tráfego de rede, análise de criticidade e tomada de decisão.

O conhecimento em geral é proveniente de uma base de dados, em geral atualizada pelo mantenedor do software de IDS utilizado pela organização e demanda constante atualização para conseguir apurar todas as invasões mais modernas, de maneira análoga aos servidores de anti-vírus. Este conhecimento é a informação que a ferramenta de segurança baseada em assinaturas usa para analisar uma seção específica de tráfego de rede e determinar se a seção é nociva ou faz parte de um código nocivo.

Como a ferramenta considera que todo o tráfego é inofensivo a não ser que seja explicitamente caracterizado como inseguro ou nocivo, considera-se que a ferramenta tem um

alto grau de precisão sem gerar falsos alarmes, porém sua capacidade de entender e de reportar todos os eventos inseguros depende do grau de compleição da base de dados de assinaturas, vulnerabilidades e exploração de falhas.

Ferramentas baseadas em conhecimento, no entanto, não são eficientes na identificação de mau uso ou abuso de recursos; Podemos exemplificar da seguinte maneira: Um usuário acessa uma planilha de cargos e salários da empresa, e nela faz uma modificação; esta ocorrência é normal para um dia comum de trabalho, mas se o mesmo evento ocorresse às 3 da manhã entre um sábado e um domingo, por exemplo, um IDS baseado em assinaturas não caracterizaria a ação como suspeita [IDS04].

3.2.3.2. Detecção de intrusão baseado em anomalia

Ferramentas baseadas em anomalias consideram que é possível mapear as atividades e o tráfego de uma rede de modo a determinar tudo o que é padrão e apontar todo o tráfego que é anômalo. Este método capta as informações que compõe os padrões do funcionamento da rede através de métodos como *sniffers* e análise de registros históricos, e uma vez que o modelo de tráfego esteja pronto, todo o tráfego é comparado ao modelo. Com isso, diz-se que o método de detecção baseado em anomalia vai ter um grau de precisão maior que os métodos baseados em conhecimentos, servindo inclusive para detectar novas formas de ataque, porém ao custo do aumento da quantidade de alarmes falsos.

As ferramentas então acabam sofrendo perante o grande volume de informações incorretas (falsos positivos), à necessidade que as redes apresentam de mudar de característica de tráfego, serviços e equipamentos e devido ao fato de que a rede pode ser atacada exatamente no momento da criação da base de comportamentos, então o ataque seria mascarado como funcionamento comum, e não como anomalia. [ROB02]

3.2.4. Limitações do IDS

Existem características limitantes nas ferramentas de detecção de intrusão. Segmentação da rede, limitação de recursos, ataques desferidos contra o próprio IDS e técnicas específicas de evasão.

O modelo CIDF possui falhas ao considerar que pacotes de dados contém informações suficientes para se interpretar e atribuir um grau de periculosidade ao tráfego analisado e também ao considerar que o computador de destino irá interpretar os pacotes enviados da mesma maneira que o IDS irá. Estas limitações podem ser exploradas e a ferramenta pode ser fragilizada, suspendendo assim sua capacidade de interpretar o tráfego corretamente. [RNP99, IDS04, ROB98]

- Segmentação de rede - é uma necessidade moderna. Devido ao crescimento das redes *ethernet*, a aplicação dos *switches* em redes é cada vez mais comum. Em uma rede segmentada, um NIDS tem alcance limitado ao domínio de broadcast que é atingível.
  Esta fragmentação afeta os NIDS, e pode ser evitada através do planejamento da implementação do IDS, garantindo o seu posicionamento em local de alta visibilidade da rede, por exemplo em um *Bastion Host* posicionado diretamente na DMZ.
- Limitação de recursos - um IDS também possui recursos de processamento, armazenamento e memória que são utilizados para se analisar o tráfego. Um atacante que possua recursos suficientes pode ser capaz de sobrepujar a capacidade de processamento da ferramenta. Um motivo adicional para a limitação de recursos ser um risco real é que comparativamente, se o atacante gasta aproximadamente 5 ciclos de processamento para gerar um pacote aleatório, a ferramenta de IDS gastará 7 ciclos de processamento para interpretar este mesmo pacote [ROB98].
- Ataques contra o IDS - as ferramentas IDS também são frágeis e suscetíveis a ataques. Uma ferramenta IDS que venha a ser comprometida em uma rede apresenta um grau de risco ainda maior do que um ataque a um *host* qualquer da rede, visto que as ferramentas de IDS são projetadas para possuírem alto grau de visibilidade na rede e as ferramentas, por serem automatizadas, recebem menos esforço administrativo, facilitando os ataques[IDS04, RNP99].
- Evasões simples - a evolução das redes está trazendo ferramentas melhores para evasão de IDS. Cifragem de dados e o uso de VPNS são técnicas bastante exploradas para se evadir do controle de um IDS. A mais nova tecnologia que possui o efeito colateral de permitir melhor ocultação do tráfego dos IDS é o padrão IPv6, que transforma o IPSEC em característica mandatória da rede, aumentando a segurança da rede, mas oferecendo maior capacidade de se inserir tráfego malicioso em pacotes cifrados [ROB98].

- Evasões complexas - existem 2 formas complexas de se evadir um IDS: inserção e evasão [ROB98].

  O método de inserção baseia-se no conceito de detecção de invasão através de base de conhecimentos; se o invasor sabe o que o IDS está buscando na rede, forja-se uma larga quantidade de pacotes com o conteúdo que o IDS busca, forçando a geração de falsos positivos que por sua vez demandarão bastante esforço do administrador de sistemas para identificar o verdadeiro pacote de invasão entre uma quantidade grande de dados.

  Evasão ataca diretamente uma fragilidade do modelo CDIF que presume que tanto o IDS quanto o *host* de destino irão reagir da mesma maneira perante um pacote. Por definição a pilha TCP exige que todos os *hosts* reajam da mesma forma perante um pacote porém é sabido que implementações incorretas, versões de software e de *drivers* de rede podem influenciar nesta interpretação de dados; além disso, o tráfego é diferente em pontos diferentes da rede: se um IDS recebe um pacote em um momento, nada garante que o *host* destino irá receber o mesmo pacote pois entre o IDS e o *host* destino sempre há a rede local e o tráfego normal da rede, com isso não é possível garantir que ocorrência de atraso possa influenciar tanto na leitura do pacote pelo IDS quanto na recepção do pacote pelo *host* de destino.

  Na prática é bastante difícil de se inserir código arbitrário em pacotes, porém não é impossível e já há ferramentas disponíveis para testes de IDS que suportam inserção, como o CASL e o IPSEND.

3.3. Conclusão

Firewalls e IDS´s são a escolha mais comum para aumento de segurança em sistemas digitais; no entanto, foram apontadas fragilidades em seu funcionamento que podem ser exploradas e cuja existência não pode ser contornada, em alguns casos por serem falhas oriundas da própria arquitetura da solução.

Para auxiliar na diminuição destas falhas, podem ser usadas um conjunto de novas tecnologias e ferramentas, que visam atuar nas lacunas deixadas pelos *firewalls* e os IDS´s . Estas técnicas e ferramentas são os *honeypots*.

Capítulo 4

Honeypots, Honeynets e Honeytokens

Com as lacunas existentes nos sistemas de segurança mais comumente usados na *internet*, é necessário utilizar ferramentas capazes de interagir com os criminosos digitais e de fornecer subsídios para que os ataques, ferramentas e vírus sejam analisados, de modo que seja possível se prevenir contra as ações dos atacantes digitais. Os *honeypots* e as *honeynets* permitem que os administradores de rede diminuam o risco aos quais seus sistemas estão expostos e ajudam a entender a maneira como o ataque foi feito.

4.1. Honeypots

A principal arma que podemos usar contra o inimigo é poder conhecer suas atitudes e conseguir prever o que ele pode fazer contra nós. No mundo da informática existe muito pouca informação de como são efetuados os ataques e como eles acontecem. Se não for possível conseguir prever o que o inimigo pode fazer contra o sistema, não tem como implementar meios para se proteger.

Hoje existem algumas ferramentas para analisar o comportamento dos inimigos virtuais e analisar suas atitudes. Com essas ferramentas podemos conhecer o que ocorre após a invasão de um sistema e qual a atitude de um invasor depois que ele consegue comprometer o sistema. Saber o que ele deseja, como ele age e quais são seus objetivos.

Uma dessas ferramentas é o *Honeypot* (Pote de Mel). Esta ferramenta é instalada em um computador com o objetivo de ser atacado para analisar todos os dados que entram e saem do sistema e como ocorre uma invasão. Segundo [HON02] os *honeypots* são sistemas criados

para atraírem e serem comprometidos por um atacante, gerando um registro histórico de todas as ações feitas neste ataque. Após o comprometimento dos sistemas podemos extrair informações que auxiliem no mecanismo de defesa ou mesmo sejam utilizados como alertas de invasão. As máquinas com um *Honeypot* instalado executam serviços falsos, que respondem como seus originais, mas na verdade estão fazendo outras operações totalmente diferentes. Estes programas que executam este tipo de atividade são chamados de *fake servers* [HBR00]. O objetivo de um *Honeypot* é utilizar recursos para capturar a sessão de um invasor e monitorar seus passos para saber como ele fez o ataque e o que pretende fazer depois que invadir o sistema, ou seja, criar um ambiente para que o invasor pense estar no domínio da situação, o que não é verdade, e assim registrar todos os seus passos dentro do sistema [FON01].

4.1.2. A História do Honeypot

O *Honeypot* começou em 1991 com a publicação dos artigos "*The Cucko's Egg*" de Clifford Stoll que durante 10 meses (1986/87) e localizou e encurralou o *hacker* Hunter e "*An Evening with Berferd*" de Bill Cheswicks que durante meses estudou as técnicas e criou armadilhas para o *hacker* Berferd, onde foram lançadas as bases do projeto e a primeira análise de um ataque. Em 1992 o especialista Bill Cheswick explicou no artigo "*An Evening with Berferd In Which a Cracker is Lured, Endured, and Studied*" os resultados do acompanhamento de invasões em um dos sistemas da AT&T, projetado especialmente para este fim [NSO03]. Em 1997 Fred Cohen's lançou o DTK (descrito a seguir) o primeiro *honeypot*, que era aberto e gratuito. Em 1999 surgiu o *Honeynet Project*, criado por Lance Spitzner em uma entidade formada por cerca de 50 especialistas de segurança criar ferramentas de defesa contra ataques. Um dos resultados foi lançamento do *Honeyd*, uma ferramenta com solução em software livre de *Honeypot* [HBR00]. Este foi o grande passo que ganhou repercussão mundial ao demonstrar a importância do estudo do comportamento dos invasores de uma rede para o desenvolvimento de novas ferramentas e sistemas de defesa.

4.1.3. O Honeypot no Brasil

No Brasil o existe o projeto HoneynetBR (www.honeynet.org.br), criado e mantido em parceria por especialistas do Instituto Nacional de Pesquisas Espaciais (INPE) e do grupo brasileiro de resposta a incidentes de segurança (NBSO). O projeto utiliza *Honeypots* de alta

interação com sistemas reais, porém com algumas modificações que permitem a captura de todos os dados, inclusive os criptografados. De certo modo, como descrito na literatura, a própria *Honeynet* pode ser considerada como um único *Honeypot* composto por diversos sistemas, porém o projeto brasileiro refere-se a cada sistema individual dentro da *Honeynet* como um *Honeypot* individual [NSO03].

Três meses após o lançamento, o projeto recebeu um importante reconhecimento, tornou-se membro da *Honeynet Research Alliance* (www.honeynet.org), que reúne diversos grupos de várias partes do mundo, todos empenhados em desenvolver a tecnologia de *honeynets*.

Visando a redução de custo, o projeto brasileiro iniciou com um *Hub* e cinco computadores, sendo uma máquina de gerenciamento e o restante pertencente à rede da *Honeynet*. Os sistemas operacionais utilizados foram Linux e OpenBSD, e todos os softwares são Open Source, dentre eles o *snort, snort-inline, ettercap*, etc.

4.1.4. Vantagens de uma Honeypot

Os *Honeypots* podem ser vistos como um recurso que não tem valor de produção, afinal não existe razão legítima para alguém fora da rede interagir com ele. Assim, qualquer tentativa de se comunicar com o sistema é uma sondagem, varredura ou ataque de estranhos. Inversamente, se o sistema inicia conexões fora de sua faixa de atuação, é bem provável que o sistema esteja comprometido. O *Honeypot* aprimora a detecção ao reduzir o número de falsos positivos, pois as tecnologias tradicionais sobrecarregam o trabalho dos administradores gerando muita informação falsa, ou seja, os analistas gastam tempo valioso analisando informação que pode ser um simples trafego normal da rede. Um IDS, por exemplo, pode gerar mais de 10 mil alertas por dia. Além das informações normais como captura de informação de cabeçalho IP, o que ajuda a determinar de onde o ataque está vindo, quais portas foram usadas e quando a atividade ocorreu, os *Honeypots* também coletam informações sobre a identidade dos atacantes, a língua que eles falam, quais ferramentas utilizaram, como as desenvolveram e a motivação para o ataque [RLW03]. Outra grande vantagem do *Honeypot* é que, como ele não é um sistema que faz parte da produção, pode ser retirado da rede a qualquer momento para análise dos dados, ou por motivo de ter sido comprometido, e ser recolocado sem que haja prejuízo para os sistemas de produção.

4.1.5. Desvantagens de um Honeypot

Como ele é um sistema isolado, instalado em um computador, ele tem sua capacidade relativamente limitada, pois não consegue captura dados de um ataque ao seu servidor Web, por exemplo, nem detectam se existem funcionários que estão roubando arquivos de sistemas confidenciais, enxergando apenas o que cruza o seu caminho. O *Honeypot* é uma ferramenta com a qual os atacantes podem interagir, o que é algo extremamente arriscado se mal implementado, por isso ele deve atuar como um complemento a outras soluções existentes, e não como uma ferramenta substituta [RLW03].

4.1.6. Como funciona uma Honeypot

Podemos exemplificar o funcionamento da seguinte maneira: um *hacker* executa um *telnet* para uma determinada máquina, o *fake Server* desta máquina emula o *telnet* e responde ao comando capturando as informações do atacante [HBR00].

4.1.7. Tipos e Níveis de Honeypots

Existem *Honeypots* de pesquisa, mais utilizadas universidades, agências do governo e o setor militar, e as de produção, que são mais utilizadas por organizações comerciais.

a) Quanto aos tipos de *Honeypots*, podemos citar:

- *Honeypots* de pesquisa: são programadas para acumular o máximo de informações dos atacantes e suas ferramentas. Eles têm um grau alto de comprometimento e trabalham em redes externas ou sem ligação com a rede principal. São utilizados para coletar informações sobre atacantes, fornecendo um quadro mais preciso dos tipos de ameaça que existem e como combatê-las.

- *Honeypots* de produção: tem como objetivo diminuir os riscos da rede principal, pois o principal elemento é a distração ou dispersão. Eles são usados para proteger a organização evitando, detectando ou auxiliando na resposta a um ataque.

b) Quanto aos níveis de *Honeypots*: basicamente temos três, os de baixa, média e alta interação [NSO03, SAR01]:

- *Honeypots* de baixa interação (*Low-interaction Honeypots*) normalmente emulam serviços e sistemas operacionais, não permitindo que o atacante interaja com o sistema.

- *Honeypots* de média interação: utilizam um ambiente falso e cria para o atacante uma ilusão de domínio da máquina. É melhor utilizada para estudo, pois dá a impressão de estar sendo invadida realmente.

- *Honeypots* de alta interação (*High-Interaction Honeypots*) são compostos por sistemas operacionais e serviços reais e permitem que o atacante interaja com o sistema.

4.1.8. Ferramentas para criar um Honeypot

Existem muitas ferramentas para se implementar um *Honeypot*, cada administrador deve buscar uma de acordo com a necessidade da sua rede, verificando quais são as possíveis vulnerabilidades para invasão e instalar uma configuração que traga informações voltadas para o seu ambiente de produção. Como exemplo de *Honeypot* podemos citar o *Deception Toolkit*, o *CyberCop Sting* e o *Mantrap*, descritas a seguir.

a) O *Deception ToolKit* (DTK) criado por Fred Cohen (1997) utiliza uma coleção de *scriptis* em *Perl* e C que simulam vários servidores e emulam diversas vulnerabilidades conhecidas em sistemas com o objetivo de enganar os atacantes. Como por exemplo, o Sendmail, que passa um arquivo de senhas falso e, em seguida, esses *scripts* são executados em um sistema hospedeiro. O atacante é atraído para utilizar este arquivo de senhas que não são reais e com isso perder tempo valioso para descobrir tais senhas. Com o DTK é possível aprender formas de ataque sobre vulnerabilidades conhecidas.

b) Cybercop Sting é uma máquina Windows NT que faz emulação de uma rede inteira replicando as pilhas IP (*Internet Protocol*) e o *inetd* dos sistemas operacionais. Pode-se simular a implementação de diversos sistemas dentro de uma máquina de *Honeypot* física. A vantagem é que é possível simular toda uma rede com diversos sistemas de

forma rápida e simples, contudo só é possível emular ambientes conhecidos como *login* de *Telnet* ou um *banner* SMTP (*Simple Mail Transfer Protocol*).

c) O *Recourse Mantrap* ao invés de simular um sistema operacional, ele encapsula vários ambientes, pois ele executa uma imagem de um sistema dentro do outro. A grande vantagem é que temos um sistema operacional real sendo executado e o atacante estará manipulando um sistema virtual completo com o qual pode interagir depois do sistema comprometido. A desvantagem é que ficamos presos aos sistemas operacionais que o fabricante pode oferecer. Além disso, o atacante pode utilizar o seu sistema como trampolim para atacar outros sistemas, pois não temos como conter as atividades do intruso, apenas analisar suas atitudes.

   Recentemente o projeto *Honeypot* do Brasil anunciou o *Rootcheck*, a nova ferramenta desenvolvida por Daniel B. Cid e já foi testada em vários ambientes de produção. Esta ferramenta trabalha como um auditor e localizador de *rootkits* desenvolvido para ambientes Linux, ele é capaz de detectar *rootkits* como o *suckit*, *adore*, etc [BLO03]. O *rootcheck* tem uma ferramenta de detecção de *rootkits*, ele checa quais as portas do sistema estão em uso e compara com o resultado do *netstat*.

4.1.9. Riscos do Honeypot

O ganho obtido com o uso de *honeypots* representa uma melhoria na qualidade dos alarmes de invasão e o aprendizado das novas técnicas sendo utilizadas pelos atacantes, mas, temos que lembrar que existe um risco implícito no que pode acontecer caso a máquina seja invadida e se perca o controle do que está acontecendo. A famosa história de feitiço que se vira contra o feiticeiro pode ser uma realidade brutal, e aquilo que era uma ferramenta de estudo pode se tornar uma ameaça real [FON01]. O *Honeypot* pode ser utilizado como trampolim para o *hacker* invadir outros sistemas, neste caso as ações legais voltam-se contra o responsável pela *Honeypot*.

4.1.10. Classificação de Honeypots baseados na implementação

Os *honeypots* podem ser divididos quanto à implementação em reais e virtuais.

a) *Honeypots* virtuais são softwares que emulam serviços, servidores e tráfego de rede. Estes softwares são configurados para abrirem portas de conexão e responderem a requisições externas. O grau de resposta varia de software para software. O *Honeyd* é uma ferramenta de implementação de *honeypots* virtuais.
b) *Honeypots* reais são implementações de sistemas operacionais e serviços válidos, sobre os quais são implementadas modificações que aprimoram o nível de geração de alertas e históricos de registros do sistema.

Ambas implementações servem ao propósito de atrair e enganar um invasor, sendo que sua eficiência contra os invasores só pode ser medida pelo grau de interação.

4.1.11. Implementação de Honeypots

Para implementar *honeypots* não pode-se usar um manual ou um livro de melhores práticas, pois cada organização deve analisar cada finalidade desejada. É importante ressaltar que as máquinas de *honeypots* devem ter regras que as permitam receber conexões, porém não iniciar conexões (assim protegendo o restante dos *hosts* da rede em caso de comprometimento da máquina do *honeypot*). O administrador deve saber quanto risco a sua organização é capaz de tolerar e implantar as regras apropriadas para minimizar os riscos.

Existem basicamente dois tipos de soluções que podem ser implementadas: as de monitoramento de portas e a de ambientes fechados. As soluções de monitoramento de porta podem basear-se em programas que ‘escutam’ a porta ou aplicações customizadas de monitoramento de porta, são projetadas para executar tarefas muito específicas, como detectar ataques ou capturar ferramentas automatizadas, estes sistemas são mais fáceis de criar, implementar e manter, além de menos arriscados, porém são mais limitados no escopo de informação que fornecem. Já em uma solução de ambiente fechado é preciso criar um ambiente virtual que, para um atacante, parece fazer parte de um sistema em pleno funcionamento. O principal valor está na forma de prevenção ou de fornecimento de informação sobre como reagir a um ataque. Estes sistemas são mais complexos e envolvem mais riscos, mas obtêm uma gama maior de informação.

4.2. Honeynets

Honeynet não é um sistema único, conforme figura 3, mas sim um conjunto de *honeypots*, podendo ser vários sistemas diferentes, tal como um servidor web IIS Windows NT, servidor DNS Linux ou um FTP em Solaris, com isso aprende-se mais sobre ferramentas e táticas usadas, é uma rede composta por sistemas de produção e aplicativos padrão, onde nada é feito ou emulado para torná-la menos segura. Uma *Honeynet* bem sucedida depende do Controle de Dados e da Captura de Dados.

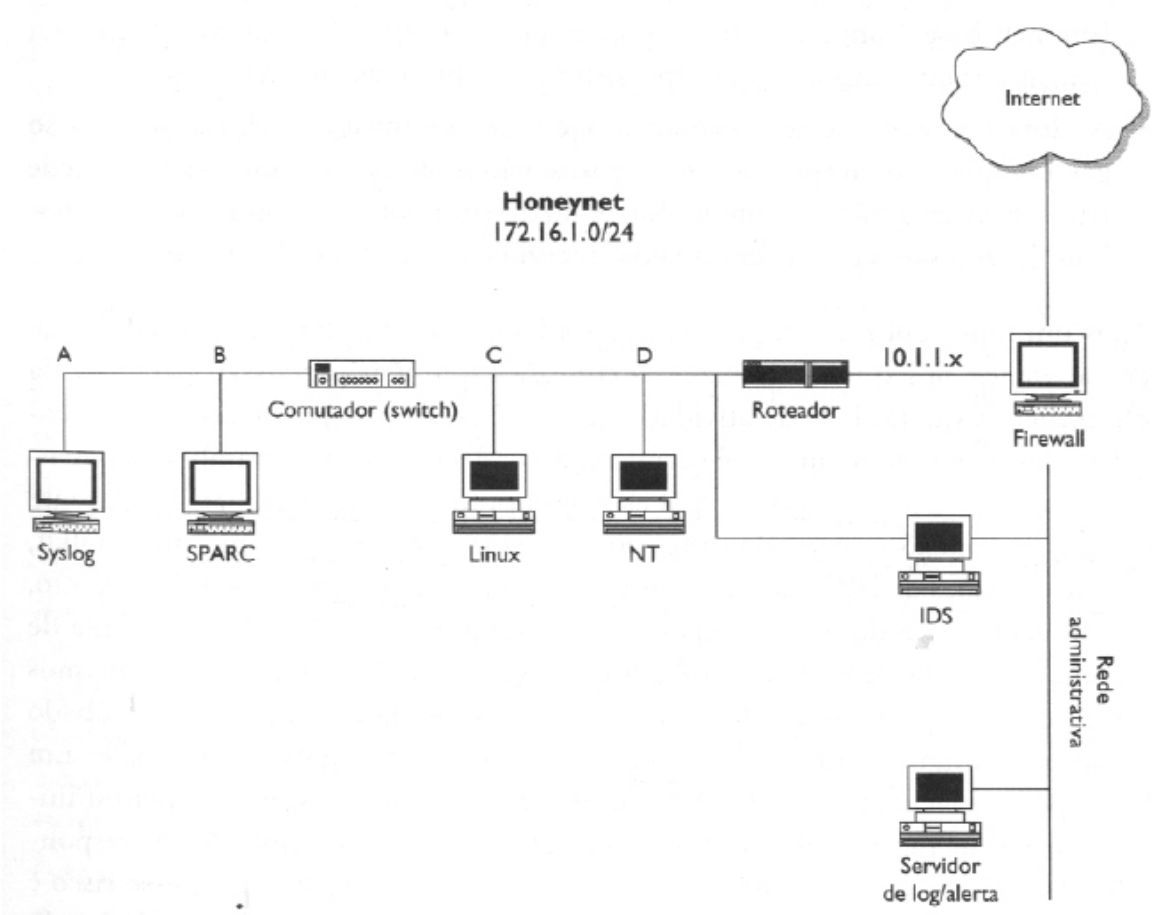

Figura 3 - Honeynet

4.2.1. Controle de Dados

Após um *honeypot* ser comprometido, o fluxo dos dados deve ser controlado sem que o atacante perceba, e garantir que o sistema não seja usado para atacar outras redes fora da *honeynet*.

Um *firewall* pode ser usado para controlar o acesso, fazendo com que os dados que entram e saem da *honeynet* passem pelo mesmo.

Numa *honeynet* há três redes distintas, a rede externa, a rede com os *honeypots* e uma terceira rede que é a administrativa, essas três redes são separadas pelo *firewall*.

A rede externa seria a Internet, que é de onde vêm os ataques, a *honeynet* possui os sistemas a serem comprometidos e a administrativa que é uma rede confiável e serve para coletar os dados remotamente.

Segundo [HON02], são definidas três regras no *firewall*:

Primeira regra – Qualquer conexão para a rede dos *honeypots* é autorizada, deste modo qualquer pessoa pode varrer os sistemas da *honeynet*.

Segunda regra – Conexões da rede dos *honeypots* para rede externa são controladas pelo *firewall*, evitando com que os sistemas da *honeynet* sejam usados para ataque a sistemas da rede externa.

Terceira regra – Conexões da *honeynet* para a rede administrativa não são autorizadas, evitando que o atacante, após comprometer um *honeypot,* tente invadir a rede administrativa.

Essas regras podem mudar e representar a filtragem do sistema de produção de uma organização.

A quantidade de conexões a partir de um *honeypot* deve ser controlada, quanto maior o número de conexões maior o risco. Caso seja ilimitada, o sistema comprometido pode ser usado para atacar outros sistemas, por outro lado, caso a quantidade for limitada, o atacante pode ficar desconfiado e com isso o *honeypot* comprometido deixará de ser útil. Segundo [HON02], um número de cinco conexões em 24 horas é um número adequado, dando ao hacker flexibilidade para *downloads* de suas ferramentas, enviar e-mail, usar o IRC (*Internet Relay Chat*) para comunicação ou qualquer outro serviço.

Meios automatizados devem ser implementados para mandar alertas, controlar e bloquear quando o número de conexões for excedido. O ataque pode ocorrer a qualquer hora do dia, e uma intervenção humana pode ser um risco, caso o *firewall* esteja configurado para mandar alertas via e-mail quando o número de conexões permitidas for excedido, estes e-mails podem não ser enviados por alguma falha de DNS (*Domain Name Server*) ou não serem lidos.

4.2.2. Captura de Dados

A captura de dados é a coleta de todas as atividades dentro da *Honeynet* comprometida para análise posterior, fazer isso com uma única camada é muito arriscado e podem ocorrer falhas, então é usado uma junção de vários sistemas de captura, sendo *logs* de *firewall/*roteador, registros de histórico de rede e os próprios sistemas, com isso melhorando o conteúdo das informações.

Esta captura não pode ser armazenada só localmente no sistema comprometido e sim num sistema confiável e seguro sem que o atacante tenha acesso. Caso o atacante tenha acesso a esses registros de arquivo, ele pode destruí-los ou modificá-los.

As camadas da captura de dados podem ser:

- Camada de controle de acesso - é a primeira camada da captura de dados, podendo ser um *firewall* ou um roteador. Como dito anteriormente, ele manda alertas para e-mail do administrador de tudo que entra e sai da *honeynet*.Todo trafego que entra e sai da rede *honeynet* é suspeito, é controlado e registrado pelos dispositivos de controle de acesso. Esses alertas são importantes para comparações futuras com os do IDS (Camada de Rede). O problema desta camada é que não registra a atividade dentro do *honeynet*, apenas o trafego que passa pelo dispositivo de controle de acesso.
- Camada de Rede - serve para capturar e analisar os pacotes que trafegam pela rede. Um IDS é usado para alertar sobre as atividades suspeitas, usando análise de assinatura dos pacotes ou detecção de anomalias e identificar e capturar a carga útil do pacote. Essas informações dos ataques, dos pacotes e do pressionamento de teclas devem ser armazenadas de tal maneira que seja fácil a análise delas posteriormente.
- Camada do Sistema - Caso o atacante use comunicação criptografada, a captura do pressionamento das teclas, por exemplo, torna-se difícil. Os registros de histórico no sistema comprometido não são armazenados só localmente, mas também num sistema remoto confiável e seguro, portanto também são capturados pelo IDS por estarem passando na rede. As formas encontradas de capturar os pressionamentos das teclas são: modificando o *bash* (/bin/bash) ou usando um *patch* no *kernel*, fazendo o armazenamento no servidor de *syslog* remoto.
- Camada *Off-Line* – Esta camada trabalha fazendo uma imagem do seu *honeypot* antes de ser colocado em atividade. Com isso, quando o sistema for comprometido, basta fazer uma comparação da imagem anterior com a atual para identificar as modificações que

ocorreram. Segundo [HON02], uma ferramenta usada para tal fim é o *Tripwire*, que faz a imagem do sistema antes de ser comprometido e a compara com o estado do após a invasão, que pode identificar os binários do sistema e os arquivos de configuração que foram modificados através do seu banco de dados.

4.2.3. Análise de Dados

As *honeynets* são ótimas para controlar e capturar os dados de um atacante, mas estas informações não serão úteis sem transformá-las em dados relevantes e de fácil entendimento. Para isso os dados são capturados de forma inteligente, automatizando os processos que os coletam, como um *e-mail* de alerta. Serão examinados os *logs* do *firewall*, alertas de IDS e o tráfego capturado, registros do sistema e os pressionamentos de teclas.
Os tipos de análise podem ser:

- *Logs* de Firewall – a análise de registros de um *firewall* geralmente é um processo muito trabalhoso, pois há grande quantidade de informações. No caso da *honeynet* o tráfego não é muito grande, pois não é um sistema de produção. Nela todo tráfego é suspeito, logo todo dado capturado é útil. Caso o firewall seja configurado para enviar alertas para o e-mail do administrador quando há tentativas de entrada e saída de conexões, o processo de análise torna-se menos trabalhoso.

- Análise do IDS - o IDS captura três fontes de informações, a primeira são os alertas gerados pelo próprio IDS, a segunda é a captura de tráfego da rede armazenando num arquivo binário e a terceira são registros de histórico da sessão ASCII detectados na carga útil do pacote, como o pressionamento das teclas. Parece redundância enviar um alerta para o administrador, mas enquanto os alertas do *firewall* mostram apenas as tentativas de conexão, os alertas de IDS podem informar o que um atacante está executando, isso é uma vantagem da captura ser feita em camadas.

- Registros Históricos do Sistema - Em um sistema comprometido, o atacante tentará apagar ou modificar os arquivos de registro do sistema, por isso eles devem ser armazenados em um servidor remoto. Quando são mandados do *honeypot* comprometido para o servidor de registro de histórico remoto, esses dados são capturados pelo IDS através da rede. Com

isso, os *logs* de sistemas estão em 3 lugares: no próprio sistema, no servidor de arquivos de registro remoto e na captura feita pelo IDS. Caso o atacante modifique os registros de histórico do sistema, basta fazer uma comparação com os dados capturados pelo IDS ou armazenados no servidor de arquivos de registro remoto.

4.2.4. Análise de Dados Avançada

Muitas vezes os *logs* de um *firewall*, os alertas de IDS e os arquivos de registro do sistema podem não informar o dado procurado, então métodos mais avançados precisam ser usados. Há dois tipos de análise avançada de dados, a primeira é a obtenção passiva de impressões digitais (*fingerprinting*) e a outra é a argumentação do sistema. A obtenção passiva de impressões digitais serve para saber o sistema operacional utilizado, os serviços e até o aplicativo usado pelo atacante. Em [HON02] diz, que algumas particularidades da pilha IP, como TTL (*Time to Live*), tamanho da janela, o DF (*Don´t fragment*) e o TOS (*Type of Service*) dos sistemas operacionais e do aplicativos são diferentes, e isso pode ser usado para obtenção dos dados procurados. Esta obtenção é feita através dos rastros de um *sniffer* do trafego do sistema remoto, ela não é totalmente perfeita, pois alguns aplicativos criam seus próprios pacotes e podem não produzir assinaturas iguais as do sistema operacional, portanto alguns valores podem ser modificados.

4.3. Honeynets GENII (segunda geração)

Conforme [GEN03], a h*oneynet* não é um produto que você simplesmente instala o software de um CDROM e o deixa de lado. A *honeynet* é uma arquitetura, uma rede controlada e usada para conter, analisar as ferramentas, e os motivos dos atacantes. A segunda geração das ferramentas de pesquisa, *honeynet* (GENII) é baseada e combinada com velhas e novas tecnologias, acrescenta flexibilidade, comodidade e segurança as *honeynets*, possui algumas particularidades em relação à primeira geração, entre elas o conceito *honeywall,* que é um mecanismo de administração com o foco no gerenciamento de rede que trabalha na segunda camada de rede.

4.3.1. Honeywall

Honeywall é um conceito novo usado nas *honeynets* GENII, e como na primeira geração é o elemento chave, pois separa os sistemas a serem comprometidos da rede externa, tudo que entra e sai da *honeynet* deve passar pelo *honeywall.* Em [GEN03], é implementado um *honeywall* de segunda camada de rede, por ser de mais difícil detecção e com três interfaces. O *gateway* separa a rede de produção da rede *honeynet*, a primeira interface é ligada a rede de produção, a segunda é conectada a rede a ser comprometida e a terceira interface para administração remota do *gateway*, incluindo o ato de mover registros de histórico e dados capturados para um lugar centralizado. Na primeira e na segunda interface não se designa um endereço IP a elas, pois estão *bridging*, já a terceira possui um IP e é separada das outras redes com o propósito de administração. As vantagens dessa arquitetura, por estar usando um *honeywall* em modo *brigde*, é que não há roteamento entre as redes, logo os pacotes não são modificados com a inserção do MAC do roteador e nem há diminuição do TTL (*Time To Live*) entre *hops* de rede, tornando assim o *honeywall* de difícil detecção e permitindo que o mesmo analise o tráfego entre as duas redes.

4.3.2. Captura e Controle de Dados

O Controle de Dados é implementado dentro do *gateway* de camada 2, pois todo o tráfego pode entrar e sair através do mesmo. Como falado anteriormente, o propósito do controle dos dados é prevenir que o atacante use o sistema comprometido para atacar outros sistemas fora da *honeynet,* através da contagem das conexões obedecendo a um limite pré-definido. A quantidade de conexões permitida depende do risco que estamos dispostos a encarar, pois temos que fazer isso sem que o atacante perceba. Além da contagem de conexões para fora da rede *honeynet,* existe o NIPS (*Network Intrusion Prevention System*), que pode identificar e bloquear ataques conhecidos. O NIPS inspeciona cada pacote que viaja através do *gateway,* se o pacote atende a alguma regra do IDS, não é gerado um alerta somente, mas também o pacote pode ser descartado ou modificado. Em [GEN03], o Honeynet Project usa o *snort_inline*, uma modificação do IDS *snort* que pode descartar ou modificar pacotes. A finalidade da captura de dados, segundo [HON02], é registrar todas as atividades que ocorrem dentro da *honeynet*, incluindo níveis de rede e sistema, isso é feito em camadas para dar mais segurança. As três camadas identificadas pelo *Honeynet Project* seriam: os *logs*

de *firewall*, análise do IDS e os arquivos de registro do sistema. Para ajudar na coleta dos registros históricos do sistema, foi criado um *patch* escondido do *kernel* chamado *sebek* capaz de registrar a atividade do atacante. A informação coletada pelo s*ebek*, não fica no *honeypot* comprometido, ela é transmitida via UDP para uma maquina *sniffing*. O atacante não pode ver essa informação sendo transmitida, isso é feito modificando o *honeypot*, fazendo com que ele não veja qualquer pacote em tal porta UDP.

4.4. Honeytokens

O conceito de *honeytokens* foi introduzido por André Paes de Barros, na lista de discussão do projeto *honeynets*, aprimorado e estudado por Lance Spitzner, trazendo mais um conceito interessante para a segurança de sistemas.

4.4.1. Definição de Honeytokens

Ao considerarmos que "*Honeypots* são sistemas computacionais cujo principal objetivo é ser sondado, atacado e comprometido", expomos uma das fragilidades do sistema já estudado, que é o fato de que há um hardware, um equipamento que pode ser explorado e que pode ser utilizado para se fazer mais ataques contra o resto da rede; *honeytokens* então transcendem esta definição por não serem equipamentos, mas pedaços de informação caracterizada conforme padrões complexos criados pelo administrador.

Uma das muitas vantagens já conhecidas da utilização de um *honeypot* reside no fato de que o tráfego a ser analisado em um *honeypot* é consideravelmente menor do que aquele que há em IDS´s e *Firewalls*, visto que um *honeypot* gera menos alarmes falsos. *Honeytokens* também geram menos alarmes falsos através da inserção de informações características que podem ser analisadas pelo conteúdo, em contrapartida às análises baseadas em comportamento e assinatura, demonstrando acessos indevidos a informações que nunca seriam acessadas através do comportamento padrão do uso do serviço.

Honeytokens então são pequenos conjuntos de informações que são inseridos em bancos de dados ou massas de informações, cujo sentido específico não é válido, mas o tráfego na rede representaria um acesso indevido. Com estas características, podemos exemplificar um *honeytokens*, como algumas da seguinte forma:

a) Uma conta específica: Com a criação de uma conta específica de usuário com poucos direitos, mas com uma configuração de segurança fraca, inserido a exemplo em um sistema Linux, uma conta chamada “Administrador” com a senha “Administrador”, cujo acesso seria monitorado.

b) Uma informação específica dentro de um arquivo: Através da inserção em um arquivo de uma seqüência específica de dados, como um número de cartão de crédito falso (“123443211199”) em uma base de dados, cujo tráfego pela rede seria monitorado, gerando alarmes caso tal conjunto de dados fosse detectado;

c) Um *link* oculto em um documento: Na criação dos documentos confidenciais de uma empresa, pode se ocultar um pequeno link cujo destino resida em um *site* externo, se possível de modo a representar qual documento está sendo acessado; esta técnica permitiria que se fizesse um rastreamento de todos os acessos a um documento controlado.

4.4.2. Implementação de Honeytokens perante massas de dados reais ou forjadas

A implementação de *honeytokens* é diretamente dependente de análise dos dados que são críticos para uma organização. Como sua forma pode ser dada por qualquer uma das implementações descritas acima, o grau de funcionalidade de cada uma das implementações é relacionado à relevância das informações que são forjadas dentro de uma massa de dados.

Em um ambiente corporativo, pode-se considerar que uma conta de administração é uma informação de grande relevância, pois tal informação permitiria aos atacantes uma grande parcela de controle sobre o sistema, mas quando se trata de utilizar um *honeytoken* para controlar o acesso à informação, é necessário uma análise do cenário e do tipo de informação que a organização trafega, dificultando a criação de uma lista de melhores práticas para implementação dos mesmos.

Os *honeytokens* são classificados em sua abrangência dentro do assunto de *honeypots* e *honeynets* como utilitários de detecção de acesso às informações.

4.4.3. Vantagens do uso de Honeytokens

Honeytokens aprimoram ainda mais uma das principais vantagens da tecnologia de *honeypots* como um todo, que é a diminuição de alarmes falsos. Através da inserção de uma informação inválida na massa de dados, pode-se observar o mau-uso de informação, acesso indevido e até rotinas de software que estejam incorretas. O tráfego de um *honeytoken* pela rede é característica de que algo incomum está acontecendo.

Honeytokens também podem ser utilizados para se construir padrões de comportamento do uso de um sistema, expondo a rotina do tráfego de informações e auxiliando na construção dos padrões que podem ser utilizados para análise pelos IDS baseados em comportamento.

4.4.4. Desvantagens do uso de Honeytokens

Além de ser um conceito experimental, com pouco mais de um ano de criação, e para o qual as primeiras RFC´s ainda estão sendo submetidas, os *honeytokens* quando aplicados sozinhos são ainda mais suscetíveis às fragilidades dos IDS. Se já é difícil para um IDS identificar um tráfego anômalo que acontece várias vezes (por exemplo, diversas tentativas de intrusão), quando apenas uma informação vital capaz de comprovar toda a invasão trafega em meio a um grande conjunto de informações, há uma possibilidade maior de que esta pequena informação seja ignorada. Quando se aplica *honeytokens* em *honeypots*, no entanto, retira-se o elemento de dúvida da tentativa de invasão, comprovando que o invasor realmente usou uma informação indevida para efetuar o ataque.

4.4.5. Descrição de um caso de uso de Honeytokens

Recentemente uma grande empresa de RH suspeitava que seu banco de currículos na Internet estava sendo acessado pelos concorrentes de maneira que muitos dos seus clientes passaram a ser contatadas pela concorrência.

Como a empresa de RH não poderia acusar formalmente a concorrência de furto de propriedade intelectual, foi criada uma série de currículos cujas informações nunca apareceriam em nenhuma das pesquisas que fizessem sentido, a não ser que se fizesse uma pesquisa por todos os currículos do *site* (por exemplo, procurando a letra A) .

Uma vez que estes currículos foram inseridos, a empresa fez uma pesquisa em todos os *sites* dos seus concorrentes por uma expressão-chave muito específica (“criação de *escargots* verdes”, por exemplo), e encontrou o currículo fictício que tinha sido forjado e plantado em sua base de dados.

Quando confrontada judicialmente, a concorrente acabou cedendo e dizendo que “alguns de seus consultores faziam pesquisas em outros *sites* para procurar profissionais”; o processo ainda se desenrola na justiça.

Estes currículos forjados são os *Honeytokens*, informações forjadas, inseridas em um sistema com o objetivo de qualificar e dar certeza da invasão.

4.5. Conclusão

Honeypots e *Honeynets* agem com grande eficiência na diminuição do risco a sistemas digitais, porém sua implementação não é simples e permite que um administrador despreparado exponha seu sistema a riscos desnecessários; no entanto, uma vez que um *honeypot* esteja em funcionamento, as vantagens para o administrador do sistema vão desde a aquisição de provas digitais e da análise dos ataques até o aprendizado de novas ferramentas e formas de ataque.

Capítulo 5

Crime Digital, Análise Forense e o Aspecto Legal dos Honeypots

O uso de *honeypots* e *honeynets* permitem que o administrador do sistema gere um histórico, um registro histórico preciso de todos os passos do invasor enquanto este abusava do sistema invadido. É possível, então, retaliar legalmente de modo que este invasor não possa mais invadir outros sistemas, fazendo-o pagar pelos danos causados.

Uma vez conhecidas as ferramentas de segurança, suas falhas tecnológicas e apresentados os *honeypots* como novas tecnologias úteis para preencher as lacunas de segurança entre os IDS e os *Firewalls*, pode-se adicionar mais uma funcionalidade aos *honeypots*, esta sendo a sua capacidade de servir como prova em processos legais contra os invasores.

É este aspecto que será analisado, agora sob o ponto de vista de diversos "operadores do direito".

5.1. Crime digital

Segundo [PAT01, REN04a, ALE04b, DEM03b], o crime digital em todas as suas formas é um Crime de Meio, um crime corriqueiro que é cometido através do uso do computador, e não uma nova modalidade de crime nunca visto antes. Logo, a questão de se punir os criminosos digitais não é tanto pela falta de leis que o permitam, mas também pelo despreparo do poder de polícia em lidar contra os atos ilegais com as ferramentas que se encontram disponíveis na jurisdição brasileira.

É um fato conhecido de que a justiça brasileira é lenta tanto em processar quanto legislar [REN00], porém com a existência da “tipificação dos crimes” já na legislação, e apenas a necessidade de se utilizar os ditos tipos de crimes no âmbito da informática ajuda a agilizar eventuais processos contra criminosos digitais.

Os crimes que podemos analisar então são aqueles cujo fim está coberto pelo âmbito da legislação já vigente, divididos entre [TUL02] crimes contra a pessoa, crimes contra o patrimônio, crimes contra a propriedade imaterial, crime contra os costumes, crimes contra a incolumidade pública, crimes contra a paz pública e outros crimes menos comuns. Será exemplificado, a seguir, formas digitais da ocorrência destes crimes.

5.1.1. Crimes contra a pessoa

Crimes que visam a vida e a integridade física dos seres humanos:

- Homicídio: Ticio invade um sistema de controle de semáforos, deixando o sinal no estado "verde" tanto para o pedestre quanto para o veículo que vem no sentido contrário, causando o atropelamento do pedestre; neste caso, respondem tanto o motorista e o invasor por homicídio, o motorista responde por homicídio culposo por não ter pretendido matar um pedestre e o invasor por homicídio doloso, por ter deliberadamente causado um evento capaz de tirar vidas.
- Crimes contra a honra: Uma pessoa mal intencionada publica em uma página web informações de cunho calunioso ou informações que não se pode provar; Uma pessoa envia para mais de uma pessoa um e-mail expondo a sua opinião negativa acerca de outra pessoa de maneira caluniosa, com informações acerca da pessoa que quem enviou o e-mail não consegue provar a veracidade.
- Indução, Estímulo ou Auxílio ao Suicídio: Ticio encontra Mévio em uma sala de bate-papo, onde Mévio revela à Ticio seu desejo de extinguir a sua vida. Então Ticio passa a estimular Mévio a cometer o suicídio, e caso Mévio venha a ter sucesso neste ato, a prova material da influência de Ticio na morte de Mévio é exatamente o computador e os eventuais *logs* de conversas entre Ticio e Mévio. A título de exemplo, na data de 18/03/1999, o jornal "O Tempo", de Belo Horizonte, publicou o endereço de um *site* americano que encorajava o suicídio como solução final dos

problemas, e pedia que os suicidas publicassem suas cartas de despedida no *site*. Pelo menos 3 pessoas das que postaram suas cartas de despedida no *site* foram encontradas mortas e uma quarta não teve sucesso na tentativa de suicídio e foi internada para tratamento psicológico;

5.1.2. Crimes contra o patrimônio

- Furto: Ticio entra em um *site* de algum operador financeiro e passa a manipular os centavos de diversas contas, transferindo-os para uma conta própria.
- Estelionato: Ticio envia e-mails fazendo correntes e pedindo que sejam efetuados depósitos monetários em uma conta corrente específica; ou ainda: a mesma pessoa utiliza um software criado para gerar números falsos de cartão de crédito e de CPF , fazendo compras então com estes números falsos. O crime de estelionato é o mais comum pela Internet, e é o que mais gera processos criminais.

5.1.3. Crimes contra a propriedade imaterial

- Violação de Direito Autoral: Ticio cria um *site* que permite que outras pessoas façam *download* de programas completos ou músicas sem pagar nada por isso
- Concorrência Desleal : o dono da Empresa MevioTronic publica em um *site* que o produto produzido pela sua concorrente, a empresa TicioTronic, é altamente nocivo para a saúde, segundo uma pesquisa americana.
- Usurpação de nome ou pseudônimo alheio: Ticio invade o *site* de um famoso escritor Mévio e lá publica um conto de sua autoria, e um comentário assinado pelo escritor dizendo que nele baseou alguma obra qualquer.

5.1.4. Crimes contra os costumes

- Pedofilia: Publicar, gerar, transmitir ou acessar imagens de crianças e adolescentes mantendo relações sexuais.
- Favorecimento à prostituição: Ticio constrói um *site* que contém *links* para fotos e números de telefone de prostitutas, ou ainda uma Mévio que envia mensagens (e-

mail, celular, torpedo), convidando mulheres a se cadastrarem e oferecerem seus serviços em dado *site.*

- Rufianismo: no mesmo *site* acima, uma opção para se contratar as mulheres e se pagar com o cartão de crédito diretamente no *site*.

5.1.5. Crimes contra a incolumidade pública

- Tráfico de Drogas e de Armas com ou sem Associação para: Ticio anuncia em um leilão pela Internet drogas ou armas com entrega em domicílio.

5.1.6. Crimes contra a paz pública

- Incitação ao Crime: Ticio, preconceituosa faz um *site* com comentários racistas e com a possibilidade de outras pessoas também "expressarem sua opinião".
- Formação de Quadrilha ou bando: Ticio, Mévio e Licio combinam, em um *chat* pela Internet a invasão de um *site* de um grande banco, com o objetivo de dividir os espólios entre suas contas.

5.1.7. Outros crimes menos comuns

- Ultraje a culto ou prática religiosa: Ticio constrói um *site* apenas para maldizer uma prática religiosa e todos os seus seguidores.
- Crime eleitoral: Ex-candidato Ticio, desprovido de direitos eleitorais por ter sido caçado anteriormente, envia mensagens às pessoas, pedindo votos para seu aliado, o candidato Mévio.

5.2. Legislação específica para o meio digital

A lei 9.296/96 é a primeira lei específica para o meio digital e trata, basicamente do sigilo das transmissões de dados, segundo a qual é vedado a qualquer pessoa ou entidade o direito de interceptação de mensagens digitais ou telefônicas, bem como quaisquer comunicações entre 2 computadores por meios telefônicos, telemáticos ou digitais.

A interpretação mais recente desta lei observa exatamente esta ultima parte da lei, "comunicação entre 2 computadores" e a aplica a furto de dados de bancos de dados, invasão e espionagem ou *sniffing* da rede e outros delitos que envolvam a manipulação de um terceiro à um conjunto de dados pertencente a outros computadores.

Ainda a lei 9.983/00 prevê como crime a ação de divulgação de segredo, inclusive por meio da Internet tanto a sua transmissão quanto sua descoberta, sendo considerado como segredo, para efeitos da lei, senhas, dados de clientes ou quaisquer outras informações que não possam ser obtidas senão através da invasão do *site*. Esta lei também inclui como crime ações que englobam mas não se limitam à inserção proposital de dados inválidos em bancos de dados e da construção e modificação de sistemas sem a autorização do proprietário.

Ainda circula pela câmara dos deputados um Projeto de lei, de número 89/04 que prevê condutas tipicamente do meio digital, como disseminação de vírus, invasão e pichação de *sites*, entre outros.

5.3. Prova de autoria e dificuldades técnicas que atrapalham a captura de criminosos virtuais

Visto que há legislação capaz de atender muitas das ocorrências de crimes digitais, podemos afirmar que não é apenas a incapacidade de se processar um crime digital que impede que o Brasil tenha um número tão grande de ocorrências de invasões sem punição.

A principal dificuldade encontra-se em efetuar a “prova da autoria” de um crime digital, prova esta que é a evidência irrefutável de que uma pessoa utilizou um computador para efetuar um crime.

A legislação brasileira compara o computador, nos casos de crime digital, à uma ferramenta, à arma do crime. Se em um homicídio nós temos a figura da vítima, o ato (perfurações por arma de fogo) e a arma que foi usada, em um crime digital existe a mesma estrutura, que é a vítima que teve sua perda moral ou material, o ato (modificação de dados, exclusão, cópia indevida) e a arma que foi usada para tal, no caso o computador.

Em um crime real, no entanto, existe a necessidade em se ligar a arma de um crime a uma pessoa que o tenha cometido. Esta ligação pode ser efetuada por testemunhos, por análises forenses laboratoriais ou por provas materiais como fotos e filmagens, por exemplo.

No mundo digital, no entanto, há uma complicação a isso: como garantir que uma pessoa realmente utilizou tal computador para efetuar um crime? As técnicas forenses são utilizadas para determinar com exatidão qual computador foi utilizado e quais as ações do criminoso digital, porém é difícil de se ligar uma pessoa ao ato criminoso. Esta é exatamente a maior dificuldade em se reprimir o crime digital. Para poder autuar um criminoso digital, é necessário um conjunto muito grande de provas circunstanciais ou então de uma autuação em flagrante delito; dadas às dimensões da Internet, onde o crime pode ser cometido em qualquer lugar do mundo e a partir de qualquer outro lugar do mundo, o flagrante instantâneo de mostra difícil de se obter, logo é necessária sempre uma investigação profunda na qual se permite que o delito seja praticado às vezes até mais de uma vez, para que se possa obter uma autuação em flagrante.

5.4. Análise Forense

"A análise forense é um conjunto de técnicas que, quando utilizadas pelo poder de polícia, confere ao microcomputador, às informações nele contidas e por ele transmitidas à característica jurídica de prova" [MAR03].

Após efetuada a autuação de um suspeito, é comum que se lacre o computador e que se leve o mesmo para análise. Esta análise consiste na geração de cópia bit a bit de todas as informações, de modo que o sistema de arquivos do computador sendo analisado (documentos, diretórios, arquivos existentes ou apagados) possa ser reconstruído garantindo que todas as informações foram mantidas e são garantidamente verdadeiras.

A legislação brasileira não possui regulamentação para a análise forense digital, mas historicamente todas as apresentações de provas digitais tem sido bem aceitas nos tribunais, comprovando que em maneira geral, os analistas forenses já se encontram preparados para enfrentar os crimes digitais.

Em vias gerais, um procedimento forense se dá através da cópia física dos dados do disco, da análise e anotação das informações da BIOS, quando relevantes, da montagem da imagem feita em um servidor de analise forense (*autopsy forensic browser*, por exemplo) e então a análise e geração de um parecer forense.

Poderíamos resumir a parte da aquisição dos dados de um disco apreendido em 1 linha de comando, que funcionaria na maioria das máquinas Linux:

dd if=/dev/hdc | tee disco.img | md5sum > disco.md5

Este comando baixaria a imagem física do terceiro HD ide de uma máquina e dela geraria um *checksum*, garantindo a veracidade e a segurança dos dados adquiridos [PRO04].

5.4.1. Ferramentas Forenses de análise digital

A análise de uma invasão depende de um grande esforço computacional, de uma grande experiência com informática e principalmente do uso correto das poucas ferramentas existentes. A polícia técnica brasileira faz uso de duas ferramentas que auxiliam no processo de análise forense , o *Kit Sleuth* e o *Autopsy Forensic Browser* [SLE04, AUT04]

5.4.1.1. Kit de Ferramentas Sleuth

O Sleuth é um kit de ferramentas baseadas em Unix capaz de navegar por imagens de dados geradas com o comando dd e analisar discos com partições de diversos tipos, inclusive, porém não limitado à NTFS, FAT32, EXT2, Solaris Disk Slices, entre outros.
A Leitura dos dados pode ser feita de duas maneiras, ou em sistemas "mortos" ou em sistemas "vivos".

Em sistemas "mortos", considera-se que se possui uma imagem gerada a partir do disco original da máquina que está sendo analisada, inclusive sem a interação do sistema operacional da própria máquina para que não hajam modificações no sistema de arquivos no momento da inicialização do sistema. Para que se obtenha um sistema "morto", considera-se que o especialista remove o disco da máquina apreendida, e conecta este disco em modo escravo em outro sistema, gerando a imagem (executando o comando dd, com a linha de comando mostrada acima) a partir do sistema mestre. Considera-se a análise de um sistema "morto" como uma técnica não intrusiva e de maneira geral, mais confiável. Os dados acessados são analisados em formato ASCII, ou hexadecimal e uma parte importante do sistema é um analisador de HASH, que permite se verificar a ordem de modificação de

arquivos. Finalmente, é uma ferramenta que apresenta um navegador de tipos de arquivo, onde se pode separar os arquivos por tipo, organizando-se, por exemplo, todos os documentos, todas as imagens, etc.

A ferramenta permite também que se rode a análise em sistemas "vivos", tipicamente a partir de um CD que irá analisar as modificações do sistema de arquivos enquanto o sistema está no ar. A análise viva tem uma menor funcionalidade para fins forenses, onde se busca um histórico do que aconteceu em um sistema, mas é apropriada para se mapear, por exemplo, atividades de vírus ou ataques à sistemas enquanto estes estão acontecendo.

5.4.1.2. Authopsy Forensic Browser

O *Authopsy Forensic Browser* (AFB) é uma evolução do kit Sleuth que usa as mesmas ferramentas e a forma de análise do kit, porém com um diferencial: por ser gráfico e apresentar uma aparência amigável de "gerenciador de arquivos", permite que o trabalho de extração e análise de dados seja feito muito mais rapidamente.

5.5. Honeypots como ferramenta Forense

Já foram vistas todas as vantagens tecnológicas do uso de um *honeypot* em uma rede, porém ainda há vantagens jurídicas em se utilizar um *honeypot*.

Inicialmente, no entanto, é necessário (re)citar que os *honeypots* são uma ferramenta de segurança, e como tal, visam impedir a invasão de um sistema em produção, e ao impedir que a invasão seja cometida com sucesso, os *honeypots* automaticamente modificam o crime cometido pelo invasor. Se o invasor iniciou suas atividades com o objetivo de invadir um *site* X (e conseqüentemente causar um dano que pode ser medido pelo prejuízo sofrido pela vítima), quando os *honeypots* fazem com que o invasor ataque e afete um outro *site*, que não está em produção, o dano é menor, e, conseqüentemente a gravidade do crime também. Na cena jurídica chama-se o ato de levar um criminoso a não cometer um crime conforme o planejado de *Aberratio Ictus* (erro de execução).

Podemos citar um exemplo bastante esclarecedor de como a Lei vê uma ferramenta que leva um criminoso a cometer um crime diferentemente de como planejou, com implicações legais diferentes para cada caso:

Cenário do exemplo:

Há 3 pessoas, Ticio, Mévio e Licio. Tomemos Ticio e Mévio por irmãos e Licio como filho de Ticio, logo sobrinho de Mévio.

Imaginando-se que Ticio tenha um seguro de vida, do qual Licio é beneficiário, e Licio, encontrando-se em dificuldades financeiras, decide por assassinar o próprio pai e assim obter uma vantagem financeira, onde o planejamento prévio do crime torna-o doloso, e no caso deste exemplo, com o agravante de ser por "motivo torpe ou vingança", o motivo torpe sendo a razão monetária do crime. Certo dia os irmãos Ticio e Mévio se encontram em um bar à noite, e Licio, sabendo que ambos estarão cansados ou sob o efeito do álcool na volta da visita ao bar, mune-se de uma arma de fogo e se esconde atrás de um objeto pelo qual Ticio e Mévio certamente vão passar.

No momento em que Ticio e Mévio passam, Licio dispara sua arma, mas pela posição em que vinham Ticio e Mévio, ou por causa da falta de iluminação, ou qualquer outro motivo, Licio assassina Mévio por engano. Neste momento, perder-se-ia os agravantes e o dolo do crime, afinal Licio não pretendia matar Mévio, mas sim Ticio.

Porém a legislação brasileira prevê este caso e confere ao assassinato de Mévio as mesmas características dolosas e torpes de como se o crime tivesse sido cometido com sucesso contra Ticio.

Porém, se Ticio tivesse se precavido anteriormente e adquirido um boneco que fosse altamente confundível consigo, e Licio disparasse sua arma contra o boneco, errando a pessoa Ticio e "matando" a isca, o crime (que era o assassinato de Ticio), se descaracterizaria, se não fosse por um outro artefato legal que permite que o crime se transforme, nesse caso do "homicídio de Mévio" (no primeiro caso) para a "tentativa de homicídio de Ticio, com agravantes". Neste caso do boneco, a condenação de Licio é muito mais branda do que no caso em que Mévio foi assassinado erroneamente, mas o dano causado foi incomparável, quando se pesa o valor de se reparar um boneco com o valor de uma vida (a vida de Mévio, assassinado erroneamente no primeiro caso, ou a vida de Ticio, caso não houvesse nem Mévio nem o boneco para receberem os disparos).

Juridicamente, em um crime digital, os *honeypots* tem mesma a função que o boneco de Ticio no exemplo. Com a invasão de um *honeypot*, diminui a parte danosa do processo cível, porém a premeditação do ato criminoso e sua execução se mantém, visto que o criminoso digital teria efetivamente "assinado Ticio", através do ataque a um servidor

qualquer, porém ao fazê-lo contra um *honeypot*, o criminoso afetou apenas um sistema menos importante.

Uma grande vantagem legal do uso de um *honeypot* é que ele se torna uma prova altamente confiável do crime, primeiro por armazenar em várias camadas os *logs* de acesso (*honeypot* Gen II) - fazendo com que os registros históricos não sejam facilmente corrompidos, também por permitir alto grau de interação com o criminoso, e também por ser facilmente removido do ar para análise da invasão. Assim, análogamente se permite fazer uma análise forense com as ferramentas de análise que possui o mesmo valor de um exame de "corpo de delito" legal, inclusive com os registros históricos do sistema que demonstram todas as ações do invasor.

Com as lacunas existentes nos sistemas de segurança mais comumente usados na *internet*, é necessário utilizar ferramentas capazes de interagir com os criminosos digitais e de fornecer subsídios para que os ataques, ferramentas e vírus sejam analisados, de modo que seja possível se prevenir contra as ações dos atacantes digitais. Os *honeypots* e as *honeynets* permitem que os administradores de rede diminuam o risco aos quais seus sistemas estão expostos e ajudam a entender a maneira como o ataque foi feito.

5.6. Conlusão

A legislação brasileira se encontra moderadamente preparada para processar tanto civil quanto criminalmente, alguns dos crimes virtuais mais comuns. Certas modalidades de crimes, no entanto, carecem de legislação mais apropriada para que se puna apropriadamente o transgressor. Logo, tudo o que falta para que mais criminosos virtuais sejam capturados é um melhor preparo das polícias técnicas e uma melhoria na preparação da magistratura brasileira, de modo que esta estivesse apta a julgar casos de crimes digitais.

Conclusão

Neste trabalho foi apresentada a tecnologia de *honeypots* como uma ferramenta válida para se preencher a lacuna de segurança que existe entre os *Firewalls* e os IDS´s.

Após estudados os principais meios de ataque e os atacantes mais comuns, os *firewalls* e suas fragilidades, os IDS´s e seus problemas, os *honeypots* se apresentam como ferramentas úteis na prevenção de invasões a sistemas de produção e no estudo de novas técnicas e ferramentas de invasão usadas pelo submundo digital.

Perante a lei brasileira, o uso das ferramentas digitais aparece surpreendentemente bem regulamentado, em contrapartida ao que se acreditava anteriormente: muitos crimes digitais já se encontram cobertos pela legislação, e há projetos de lei já em fase final de aprovação.

Juridicamente, vê-se que as ferramentas de segurança servem no processo criminal como "testemunhas" ou como "provas materiais", e que o uso de *honeypots* quando estes conseguem evitar o dano aos sistemas de produção acaba por mitigar também a gravidade do crime cometido, logo amenizando quaisquer punições legais que possam ser aplicadas ao invasor.

Referências Bibliográficas

[CAM97a] CAMPANA, C. Ferramentas de Segurança - http://www.rnp.br/newsgen/9711/seguranca.html , acessado em março/2004;

[FRA99] NED, F. Introdução a IDS - http://www.rnp.br/newsgen/9909/ids.html#p5, acessado em março/2004

[CAM97b] Rede Nacional de Ensino e Pesquisa Segurança. Você se preocupa com isso? - http://www.rnp.br/newsgen/9705/n1-3.html, acessado em março/2004

[LAN03] SPITZNER, L. Honeypots, Values and Virtues LanceSspitzner An Introduction to Intrusion Detection Systems http://www.tracking-hackers.com – acessado em março/2004

[INN01] INNELLA, P. An Introduction to Intrusion Detection Systems - http://www.securityfocus.com/infocus/1520 , acessado em março/2004

[INN04] INNELLA, P. Infocus: Yearly Statistics Paul Innella and Oba McMillan, Tetrad Digital Integrity, LLC - http://www.securityfocus.com/infocus/2720, acessado em março/2004

[SHI00] Shirey Internet Security Glossary - http://www.ietf.org/rfc/rfc2828.txt, acessado em março/2004

[TRE04] Trend Micro Inc Trend Alerts - http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?Vname=WORM_MYDOOM.A, acessado em março/2004

[NOR2004] Symantec Inc Virus reports - http://securityresponse.symantec.com/avcenter/venc/data/w32.mydoom.f@mm.html acessado em março/2004

[MCA04] McAfee Inc McAfee Security - http://us.mcafee.com/virusInfo/default.asp?id=description&virus_k=100983, acessado em março/2004

[GER99] GERLARCH C. O ataque do script Kiddie http://www.rnp.br/newsgen/9905/kiddie.html, acessado em março/2004

[MEI2003] MEINELL C. The Überhacker II: More ways to break into a computer http://www.happyhacker.org/tuh.shtml, acessado em março/2004

[HAC04] Hacking Exposed - http://www.hackingexposed.com/tools/tools.html, acessado março/2004

[CHE04a] Check Point's Foundation Technologies - http://www.checkpoint.com/products/solutions/technologies.html, acessado em abril/2004

[FIR02] Firewall.net - http://www.firewall-net.com/en/, acessado em abril/2004

[CHE04b] Check Point Application Intelligence - http://www.checkpoint.com/products/downloads/applicationintelligence_whitepaper .pdf, acessado em abril/2004

[CHE04c] Statefull Inspection Technology - http://www.checkpoint.com/products/downloads/Stateful_Inspection.pdf, acessado em abril/2004

[WIL03] Firewall - http://www.wilders.org/firewalls.htm, acessado em abril/2004

[LOP97] LOPES, R. http://www.rnp.br/newsgen/9708/n3-1.html, acessado em abril/2004

[GHU02] GURKHA, G. www.madison-gurkha.com/publications/ nordu2002/nordu.htm, acessado em abril/2004

[SHI00] SHIRLEY R. http://www.ietf.org/rfc/rfc2828.txt, acessado em abril/2004

[CHA97] CHAPMAN B.; ZWICKY E. Building Internet Firewalls

[CHE97] CHESWICK B.; BELLOVIN S. Firewalls And Internet Security

[GAR99] GARFINKEL S.; SPAFFORD G. Company Practical UNIX Security

[MIC04] www.microsoft.com/products/isaserver/default.asp, acessado em abril/2004

[CIS03] Strategies to Protect Against Distributed Denial of Service (DDoS) Attacks - http://www.cisco.com/en/US/tech/tk583/tk385/technologies_white_paper09186a0080174a5b.shtml, acessado em abril/2004

[BLO03] BR-LINUX.ORG – http://brlinux.linuxsecurity.com.br/noticias/001083.htm, acessado em abril/2004.

[HON02] – O PROJETO *HONEYNET*, Conheça o seu Inimigo, tradução: Kátia Aparecida Roque. Editora Makron Books, São Paulo, 2002.

[HBR00] – *HONEYPOTBR*, Projeto *Honeypot* Brasil FAQ (Perguntas freqüentes), http://www.honeypot.com.br, acessado em abril/2004.

[NSO03] – *NIC BR Security Office, Brazilian Computer Emergency Response Team*, http://www.nbso.nic.br/docs/reportagens/2003/2003-06-24.html, acessado em abril/2004.

[FON01] – FONSECA, ANTONIO MARCELO FERREIRA DA. Projeto Pliscka – O Nosso Primeiro *Honeypot* na Internet

[SAR01] SARTINI, HUMBERTO. Ferramentas de estudo de segurança – Projeto *HoneyPotBR*, http://web.onda.com.br/humberto, acessado em abril/2004.

[RLW03] *RESELLERWEB*, Armadilha para os *hackers* – http://www.resellerweb.com.br/shared/print_story.asp?id=42776, acessado em abril/2004

[VAB04]VERDADE@ABSOLUTA. Detecção de Sistema Operacional Remotamente via o *FingerPrinting* da Pilha TCP/IP, http://www.absoluta.org/absoluta/seguranca/seg_detect_os.html, acessado em 07/04/2004

[GEN03] THE *HONEYNET* PROJECT. *Know Your Enemy*: GenII *Honeynets*, http://www.honeynet.org/papers/gen2/, acessado em 13/04/2004

[PFI02] *THE HONEYNET PROJECT. Passive Fingerprinting*, http://www.honeynet.org/papers/finger, acessado em 01/04/2004.

[LAN03] *HONEYPOT BY* LANCE SPITZNER. *Honeytokens*: *The Other*, ww.tracking-hackers.com *Last updated* 21 July, 2003, http://www.securityfocus.com/infocus/1713, acessado em abril/04

[NIC03] THOMPSON, NICHOLAS FELLOW. *The New York Times*, April 28, 2003. http://www.newamerica.net/index.cfm?pg=article&pubID=1205, acessado em abril/2004.

[HNK03] *HONEYTOKENS*. O Próximo Nível dos *Honeypots*. http://www.paesdebarros.com.br/artigos.html, Domingo, Fevereiro 23, 2003, acessado em abril/2004

[PRO04] Prof. LYCIO P. http://www.opencs.com.br/infocenter/download/arquivo/Palestras%20CIAB%202003/Fraudes_Eletronicas/Provas_fraudes_eletronicas.pdf, acessado em 19/04/2004

[REN04a] OPICE R.; CANHA J. Os crimes eletrônicos e seus enquadramentos legais (The Electronic Crimes and their Legal Status) Caderno LegalJornal Gazeta Mercantil, acessado 15/04/2004

[ALE04a] – COELHO A. Roubo de Identidade via Internet - Cartões de Crédito e Informações Bancárias/Brasil - Estados Unidos (*Identity Theft via Internet - Credit Cards and Accounting Information/Brazil - The United States*) - 2004

[CAR04a] CHAVES C. Identificação da Autoria nos Cibercrimes (Identification of the Perpetrator in Cybercrimes) - 2004

[OMA04a] KAMINSKI O. Brazil: Os vírus de computador e a legislação penal brasileira (Brazil: the computer virus and the Brazilian Criminal Law) - 2004

[ALE04b] COELHO A. Crime e Internet no Brasil - A Lei 9.296/96 como solução para a lacuna Jurídica atual no que tange às invasões de bancos de dados, servidores e computadores pessoais via Internet (Crime and Internet in Brazil - The Law No. 9.296/96 - 2004

[VIN03] CARNEIRO V. Dos Crimes de Pirataria pela Internet e sua Representação (The Piracy Crimes throught the Internet and their Representations) - 2003

[DEM03] REINALDO FILHO D. Crime de Divulgação de Pornografia Infantil pela Internet - Breves Comentários à Lei 10.764/03 (The Disclosure Crime of Child Pornography through the Internet - Comments on Brazilian Law No. 10.764/03) - 2003

[DEM03b] REINALDO FILHO D. Questões Técnicas Dificultam Condenações por Crimes cometidos pela Internet - 2003

[REM03a] OPICE R.; GOMES M. O Novo Código Civil e as Relações Jurídicas Virtuais (The New Brazilian Civil Code and the Virtual Legal Transactions) - 2003

[AMA03] VIDONHO A. A Nova Responsabilidade Civil do Incapaz pelos Atos Praticados pela Internet - 2003

[AIR03] ROVER A.; OLSEN L. Validade Jurídica de Documentos Eletrônicos assinados com Infra-Estrutura diferentes a ICP-Brasil - 2003

[MAR03] KNOOP M. Análise forense: um mundo sem fronteiras - 2003

[JAR03] BEZERRA J. A Interceptação do Conteúdo dos E-mails e o Cabimento como Meio de Prova - 2003

[MAR03] PAIVA, M. A utilização da lei do fac-símile para o e-mail - .2003

[THA03] GUARINO, T. O Software e o Regime Jurídico Aplicável (The Software and the Applicable Legislation) - 2003

[MAR03] PAIVA, M. A autenticação de documentos no novo Código Civil (The certification of documents according to the New Brazilian Civil Code) - 2003

[MAR03b] PAIVA, M. O impacto do novo Código Civil nas relações virtuais (The impact of the new Brazilian Civil Code on virtual relations) (article in Portuguese)- 2003

[UKL03] COMISSION, L. Difamation and the Internet - A Preliminary Investigation (Difamação e a Internet - Uma Investigação Preliminar) Report prepared by the Law Comission - UK (Dec/2002). Relatório preparado pela Comissão Legal do Reino Unido (Dez/2002).Law Comission – UK - 2003

[REN03c] BLUM, R. O novo Código Civil e a internet (The New Civil Code and the Internet) (Article in Portuguese)Gazeta Mercantil - 2003

[REN02] BLUM, R e ABRUSIO, J.. Crimes Eletrônicos (Electronic Crimes) (article in Portuguese) CBEJI - 2002

[FLA02] SIQUEIRA, F. Furto, Supressão de Dados Sigilosos Consignados em Sites na Internet de Acesso Restrito e o Estelionato Virtual - 2002

[ALE02] DAUON, A. Adultério Virtual (Virtual Adultery) - 2002

[LUC02] BARROS, L. O Crime na Era da Inform@ção - 2002

[SAN02] NOGUEIRA, S. O Princípio da Intervenção Mínima e a Lei Penal Especial para os Crimes de Informática - 2002

[DAV02] BRASIL, D. Meios Eletrônicos de Prova (Legais ou Não?) -2002

[TUl02] VIANNA, T. Hackers: Um Estudo Criminológico da Subcultara Cyberpunk - 2002

[TUL02b] VIANNA T. Dos Crimes pela Internet - 2002

[EME02] PASSOS, E. Internet e Legislação - 2002

[PED01] REZENDO, P. Onde estão os verdadeiros crimes de informática? - 2001

[GIB01] BRUNO, G. O Sigilo de Dados e a Privacidade On Line - Anteprojeto de Lei do Comércio Eletrônico - 2001

[ANG01] BRASIL, A. Incoerência: Lei Anti-Spam americana o torna legal - Angela Bittencourt Brasil - 2001

[TUL01] VIANNA, T. Dos Crimes por Computador - 2001

[REN00] BLUM, R. A Internet e os Tribunais - 2000

[ALE00] DAOUN, A. Os Novos Crimes de Informática - 2000

[PAT01] PECK, P. Direito Digital

[AUT04] Autopsy Forensic Tools – acessado em abril/2004
http://www.sleuthkit.org/autopsy/desc.php

[SLE04] Sleuth Forensic ToolKit – acessado em abril/2004
http://www.sleuthkit.org/sleuthkit/desc.php