unesp UNIVERSIDADE ESTADUAL PAULISTA

Administração de Redes TCP/IP

Tópico:
Conceitos de Segurança
Parte 2 - As Seqüelas: *Backdoors*

Prof. Dr. Adriano Mauro Cansian
adriano@dcce.ibilce.unesp.br
UNESP - IBILCE - São José do Rio Preto

Conceitos de Segurança[1]

Parte 2 - As Seqüelas: *Backdoors*

Prof. Dr. Adriano Mauro Cansian
adriano@dcce.ibilce.unesp.br
UNESP - IBILCE - São José do Rio Preto

1. As seqüelas: ***backdoors***[2]

1.1. Introdução

Desde os primeiros dias que os invasores de computadores após uma invasão bem sucedida tentam desenvolver técnicas ou *backdoors* que **permitam-lhes retornar à máquina invadida posteriormente**.

Este texto concentra-se nos *backdoors* mais comumente empregados e em possíveis maneiras de identifica-los.

O foco principal será em *backdoors* para UNIX com alguma discussão no desenvolvimento futuro de *backdoors* para Windows NT. Será descrita a complexidade das questões envolvidas nas tentativas de determinar os métodos utilizados pelos invasores e as bases para o entendimento, por parte dos administradores, de como eles devem proceder para barrar os invasores de voltarem a acessar os seus sistemas.

Quando um administrador compreende o quão difícil é barrar um invasor, depois que ele conseguiu acesso ao sistema, torna-se mais fácil para ele mudar para uma atitude pró-ativa no bloqueio de acesso a invasões.

Pretende-se abordar os métodos mais populares utilizados por invasores principiantes e avançados.

Não é objetivo deste trabalho abranger todas as formas possíveis de criar e instalar um *backdoor* uma vez que não há limite para todas as possibilidades de faze-lo.

---

[1] Créditos: este material foi produzido a partir de **"Tópicos em Segurança de Redes"** - Volume I, V.0.5.1, produzido e/ou traduzido por Pedro A. M. Vazquez <*vazquez@iqm.unicamp.br*> e outros (27/julho/98). Reproduzido sob permissão.

[2] Original de Christopher Klaus <cklaus@ISS.NET>; publicado originalmente em *Bugtraq* (http://www.geek-girl.com/bugtraq/) de 16 de Agosto de 1997. Tradução livre de Pedro A.M. Vazquez <*vazquez@iqm.unicamp.br*>. Reproduzido sob permissão.

Para a maioria dos invasores um *backdoor* deve prover as seguintes funções:

- Permitir o acesso à máquina mesmo se o administrador tentar torna-la segura mudando senhas;

- Permitir o acesso à máquina com a menor visibilidade possível, a maioria dos *backdoors* provê alguma forma de evitar que sejam gerados logs do acesso e em muitos casos a máquina não mostra se há alguém logado nela durante o acesso do invasor.

- Permitir o acesso à máquina da forma mais rápida possível. A maioria dos invasores deseja voltar a acessar o computador sem precisar repetir o trabalho de explorar furos de segurança existentes nela.

Em alguns casos, se o invasor imaginar que o administrador possa detectar os *backdoors* instalados, ele recorrerá ao uso repetido da(s) vulnerabilidade(s) da máquina como forma de *backdoor*.

Desta forma, não alterando nada, poderá confundir o administrador e as vulnerabilidades do sistema permanecerão na forma de *backdoor* desconhecido.

1.2. Password Cracking

Um dos primeiros e mais antigos métodos empregados pelos invasores para obter acesso a um sistema UNIX consiste em rodar um ***password cracker*** no arquivo de senhas. Isto revela contas com senhas fracas.

E todas estas novas contas terminam tornando-se *backdoors* mesmo que o administrador bloqueie a conta originalmente utilizada pelo invasor. **Muitas vezes o invasor poderá procurar entre essas contas aquelas abandonadas ou raramente utilizadas e trocar a senha para algo difícil**.

Com isto o administrador, mesmo rodando um *password cracker*, não descobrirá facilmente quais contas bloquear no seu sistema.

1.3. *Rhosts* + +

Máquinas UNIX em rede costumam utilizar serviços como rsh e rlogin que utilizam um **mecanismo simples de autenticação baseado em hostnames listados em** .rhosts**.**

Um usuário pode facilmente configurar quais máquinas não vão exigir uma senha para acesso.

Um invasor que tenha obtido acesso ao arquivo .rhosts de alguém pode colocar um + + nele permitindo a qualquer um na Internet logar nessa conta sem precisar de uma senha.

Muitos invasores utilizam este método especialmente quando os diretórios home das contas são incorretamente exportados para o mundo. Estas contas tornam-se *backdoors* para os invasores retornarem ao sistema a qualquer momento.

Muitos invasores preferem utilizar rsh no lugar do rlogin pois, em muitos casos, ele não possui nenhuma forma de registro do acesso nos logs do sistema.

Muitos administradores costumam checar as áreas de usuários a procura de + + nos .rhosts o que leva o invasor a colocar neles um hostname e um username de outra conta comprometida na rede tornando mais difícil e menos óbvia a descoberta.

1.4. *Checksum* e *Timestamp*

Há tempos os invasores substituem os executáveis originais do sistema pelas suas próprias versões com cavalos de Tróia.

Muitos administradores confiaram no uso das datas de acesso e do *checksum* dos arquivos (ex. o programa sum do UNIX) como forma segura de determinar quando um arquivo binário foi modificado.

Os invasores desenvolveram tecnologias que permitem recriar a mesma data de acesso do arquivo original.

O programa sum depende de um *checksum* baseado em CRC e pode ser facilmente enganado.

Os invasores **desenvolveram programas que podem modificar um binário com um cavalo de Tróia para que ele gere o mesmo *checksum* do programa original** enganando o administrador.

O uso do MD5 para o cálculo de *checksums* é a maneira recomendada hoje em dia pela maioria dos fabricantes. O MD5 é baseado em um algoritmo que ninguém, ainda, conseguiu provar que pode ser fraudado.

1.5. Login

No UNIX, normalmente o programa **login** é o responsável pela autenticação via senhas quando alguém realiza um telnet para a máquina.

Os invasores **modificaram o programa login.c original** de forma que este compare a senha fornecida não somente com as senhas armazenadas no sistema mas também com senhas *backdoor* compiladas nele.

Se o usuário digitar esta senha ele permitirá o acesso mesmo que o administrador tenha modificado a senha desta conta no sistema.

Isto permite que o invasor tenha acesso a qualquer conta no sistema, inclusive a conta root.

Este *backdoor* também **permite o acesso sem que seja registrado nos arquivos do sistema** (wtmp, utmp eoutros) a sessão do invasor.

Desta forma um invasor pode estar logado no sistema e com acesso ***shell* sem que ninguém o detecte**.

O uso de *checksums* MD5 permite verificar e detectar alterações no programa login.

1.6. Telnetd

Quando um usuário realiza um telnet para uma máquina o **inetd** recebe a conexão e a passa para o in.telnetd.

Este por sua vez executa o programa **login**. Alguns invasores sabem que os administradores verificam o programa login freqüentemente e **optam por modificar o in.telnetd**.

Internamente o in.telnetd realiza uma série de procedimentos durante o login entre eles a definição dos parâmetros de ambiente como o tipo de terminal.

Tipicamente isto é um valor como Xterm ou VT100. **Um invasor pode modificar o in.telnetd para fornecer-lhe um *shell* quando o terminal for igual a letmein**.

Alguns invasores modificaram o in.telnetd para fornecer acesso a qualquer conexão originada numa porta específica no computador remoto.

1.7. Serviços de Rede

Praticamente **qualquer serviço de rede** provido por um computador **pode ser alvo de um *backdoor***.

Versões do *finger, rsh, rexec, rlogin, ftp* e mesmo do inetd contendo um *backdoor* estão disponíveis na rede há um bom tempo.

Existem programas que são apenas um *shell* conectado a uma porta TCP contendo uma senha *backdoor* para o acesso.

Estes programas muitas vezes **substituem um serviço** como uucp que não é utilizado em uma dada máquina mas está habilitado ou podem **ter sido adicionados ao inetd.conf como um serviço novo**.

Os administradores precisam ficar muito atentos a quais serviços estão sendo disponibilizados no seu equipamento e utilizar o *checksum* MD5 dos programas originais para verificar a integridade dos mesmos.

1.8. Cronjob

O cron no UNIX é utilizado para executar programas em horários pré-definidos.

Um invasor pode adicionar um *shell backdoor* no cron para ser executado entre 1 e 2 da madrugada. Desta forma, durante uma hora a cada noite o invasor terá acesso ao sistema.

Os invasores também têm instalado *backdoors* em programas que legitimamente são executados de forma rotineira via cron.

1.9. *Backdoors* em Bibliotecas

Praticamente todo sistema UNIX utiliza **shared libs**. Estas bibliotecas **visam reduzir o tamanho dos programas através do compartilhamento de funções comuns a eles**.

Alguns invasores têm introduzido *backdoors* em algumas rotinas dessas bibliotecas como a **crypt.c** e **a _crypt.c**.

Programas como o login utilizam a função crypt() e, se um *backdoor* estiver contido nela, ela pode fornecer um *shell* para o invasor.

Desta forma, **mesmo que o administrador utilize o MD5** para conferir a integridade **do programa login** ele estará corrompido.

Muitos administradores não verificam com MD5 a integridade de suas shared libs.

Um problema para os invasores surgiu quando alguns administradores iniciaram a utilização do MD5 em todos os arquivos do sistema.

Uma das formas encontradas pelos intrusos para contornar isto foi instalar *backdoors* nas funções de acesso a arquivos e na função open().

Estes *backdoors* foram configurados para para ler os arquivos originais mas executar cavalos de Tróia.

Desta forma o programa de MD5 **sempre obtém os *checksums* esperados**. Uma forma de contornar este problema consiste no administrador criar um programa de cálculo de *checksum* MD5 estático.

1.10. *Kernel*

O *kernel* é a base de todas as operações do UNIX.

O mesmo método utilizado com as bibliotecas pode ser empregado com ele. A diferença agora é que mesmo um programa estático pode ser enganado.

Um *kernel* contendo um *backdoor* bem feito é possivelmente uma das coisas mais difíceis de serem detectadas.

Felizmente **não existem scripts de modificação e introdução de *backdoors* em *kernel* amplamente disponíveis** e ninguém sabe exatamente o quão espalhada é a sua utilização.

1.11. Filesystem

Um invasor pode desejar armazenar seus dados e arquivos em um servidor sem que o administrador os descubra.

Estes arquivos podem consistir, por exemplo, no seu kit de invasão, logs de *sniffers*, etc.

Para ocultar estes arquivos, usualmente grandes, do administrador **um invasor pode modificar programas do sistema como ls, du e fsck para que não mostrem a existência de certos arquivos e diretórios**.

Em um nível bem mais baixo um invasor modificou uma seção de um disco rígido com um formato próprio que aparecia como *bad sectors* no disco.

Desta forma, utilizando ferramentas especiais, o invasor pode acessar os seus arquivos mas para o administrador torna-se muito difícil descobrir que os setores marcados como *bad* na verdade contém arquivos escondidos.

1.12. Ocultação de Processos em Execução

Um invasor muitas vezes **deseja ocultar os programas que ele está executando**. Estes programas, usualmente, são *password crackers* ou *sniffers* de rede.

Existem alguns métodos utilizados para isto, os mais comuns são:

- Ele pode escrever o programa de forma que este modifique o valor de argv[] e aparecer com o nome de outro processo;

- Ele pode renomear o comando para o nome de um programa válido como syslogd. **A saída do comando ps mostraria um nome padrão de comando do sistema em execução**;

- Ele pode alterar as bibliotecas do sistema **para que o "ps" não mostre todos os processos**;

- Ele pode instalar um *patch* ou *backdoor* em uma função controlada por interrupção de forma que ela não apareça na tabela de processos.

Um exemplo de *backdoor* deste tipo é o amod.tar.gz disponível em:

http://star.niimm.spb.su/~maillist/bugtraq.1/0777.html

- O invasor pode modificar o *kernel* para ocultar certos processos.

1.13. Rootkit

Um dos pacotes mais populares para instalação de *backdoors* é o ***rootkit***.

Ele pode ser localizado utilizando serviços de busca no WWW.

A seguir os arquivos que são instalado conforme descrito no README do rootkit:

z2 → remove entradas do utmp, wtmp e lastlog;
es → sniffer Ethernet de autoria de rokstar para kernels baseados em sun4;
fix → tenta falsificar *checksums* e datas/permissões;
sl → acesso root via senha mágica fornecida ao login;
ic → ifconfig modificado para não mostrar interfaces em modo promíscuo;
ps → oculta processos específicos;
ns → netstat modificado para não mostrar conexões para certos computadores;
ls → oculta diretórios e arquivos;
du5 → oculta a quantidade de espaço em disco utilizada;
ls5 → oculta diretórios e arquivos;

1.14. Tráfego de Rede

Os invasores desejam não apenas **ocultar suas pegadas** nos computadores mas também **o seu tráfego** na rede o máximo possível.

Os *backdoors* de tráfego de rede algumas vezes permitem ao invasor **obter acesso através de um firewall**.

Existem muitos *backdoors* de rede que permitem um invasor montar um esquema onde uma certa porta em uma dada máquina permitirá acesso sem ao menos passar pelos serviços normais.

Como o tráfego de rede destina-se a uma porta não padrão o administrador pode falhar em detecta-lo.

Estes *backdoors* de tráfego de rede tipicamente utilizam TCP, UDP e ICMP, mas podem ser utilizar outros protocolos.

1.15. *Shell* TCP

O invasor pode **configurar um *backdoor* baseado em um *shell* TCP em algum número de porta bem alto onde o firewall não realiza bloqueio**.

Muitas vezes estes *backdoors* não estão protegidos por senha de forma que um administrador que conecte neles não perceberá de imediato que se trata de um acesso *shell*.

Um administrador pode procurar por estas conexões com o netstat verificando quais portas estão escutando a rede, de onde para onde as conexões existentes estão sendo feitas.

Muitas vezes estes *backdoors* permitem ao invasor passar por cima dos TCP Wrappers.

Estes *backdoors* pode ser executados na porta SMTP que é liberada na maioria dos firewalls.

1.16. *Shell* UDP

Como muitas vezes os administradores podem detectar as irregularidades nas conexões TCP o uso de *backdoors shell* baseados em UDP evita que o netstat mostre um invasor acessando o computador.

Muitos firewalls foram configurados para deixarem passar serviços baseados em UDP como o DNS.

Em alguns casos o invasor pode colocar o *backdoor* com *shell* UDP nesta porta e conseguirá atravessar o firewall.

1.17. *Shell* ICMP

O ping é a forma mais comum de descobrir se um computador está no ar através do envio de pacotes ICMP.

Muitos firewalls permitem máquinas externas enviarem estes pacotes para máquinas internas.

Um invasor pode inserir dados nos pacotes ICMP do ping e "tunelar" um *shell* entre ambas as máquinas.

Um administrador pode perceber um excesso de pacotes ping mas, a menos que ele olhe o conteúdo desses pacotes, um invasor pode passar desapercebido.

1.18. Link Criptografado

Um administrador pode fazer uso de um *sniffer* para tentar acessar o conteúdo de pacotes que parecem ser de alguém fazendo um acesso *shell*, mas **um invasor pode adicionar criptografia ao *backdoor* de tráfego de rede tornando praticamente impossível determinar o conteúdo do que está sendo transmitido entre ambas as máquinas**.

1.19. Windows NT

Como o Windows NT não permite, de forma fácil, múltiplos usuários em uma mesma máquina e acesso remoto da mesma forma que o UNIX, **torna-se mais difícil para o invasor quebrar sistemas baseados no NT, instalar um *backdoor* e lançar um ataque a partir deles**.

Por isto é muito mais frequente ataques realizados a partir de um computador UNIX do que de um computador NT.

Entretanto, sistemas NT são muito vulneráveis a ataques DoS.

A medida que o Windows NT avançar na tecnologia de acesso multiusuário a frequência de invasores que utilizam NT para seus ataques deverá aumentar.

E se isto ocorrer, muitos dos conceitos oriundos dos *backdoors* UNIX podem ser transferidos para o NT e os administradores devem estar preparados para o invasor. Por exemplo, hoje já existem servidores telnetd para NT.

2. Soluções

Conforme a tecnologia de *backdoors* avança torna-se cada vez mais difícil para os administradores determinarem se um invasor conseguiu penetrar no seu sistema ou se ele foi expulso de forma bem sucedida.

2.1. Determinação do Nível de Vulnerabilidade

Uma das primeiras etapas de um comportamento pró-ativo visando segurança é **determinar o nível de vulnerabilidade da sua rede e assim definir que furos existem que necessitam ser corrigidos**.

Existem várias ferramentas comerciais que auxiliam na varredura e auditoria de sistemas buscando vulnerabilidades.

Muitas empresas melhorariam de forma dramática a sua segurança se simplesmente instalassem os *patches* de segurança disponibilizados gratuitamente pelos vendedores.

2.2. Linha de Base MD5

Um componente obrigatório de um scanner de sistemas são as **linhas de base MD5**.

Estas linhas de base **devem ser criadas ANTES de um ataque** e utilizando o sistema original.

Uma vez que o invasor obteve acesso ao equipamento e instalou *backdoors* **qualquer tentativa de criar uma linha de base pode incorporar a ela os *checksums* dos *backdoors***.

Muitas empresas foram invadidas e tiveram *backdoors* instalados nos seus sistemas por vários meses, depois de algum tempo todos os backups dos sistemas continham os *backdoors*.

Quando algumas dessas empresas descobriram que haviam sido invadidas elas recuperaram o backup na esperança de remover os *backdoors*. O esforço foi fútil, uma vez que elas estavam recuperando todos os arquivos, inclusive aqueles com *backdoors*.

A comparação de linha de base e sua geração deve ser feita antes que um ataque aconteça.

2.3. Detecção de Invasão

A detecção de invasão têm se tornado mais importante na medida que as organizações estão se conectando à Internet e permitindo conexões para alguns de seus computadores.

A maioria dos sistemas antigos de detecção de invasão (IDS) era baseada na análise de logs.

Os sistemas IDS mais modernos utilizam tecnologias baseadas em *sniffing* em tempo real e análise de tráfego de rede.

Estes IDS podem olhar o conteúdo de pacotes UDP DNS e determinar se eles conferem com as requisições do protocolo DNS. Se isto não ocorre um alerta pode ser disparado e os pacotes capturados para análise posterior.

O mesmo princípio pode ser aplicado à análise de dados contidos em pacotes ICMP para determinar se trata-se de um pacote ping normal ou uma sessão *shell* criptografada.

2.4. Vigilância

Uma vez que a área de segurança modifica-se de forma extremamente rápida, com novas vulnerabilidades sendo descobertas e os invasores constantemente projetando novos ataques e técnicas de *backdoors* não há tecnologia de segurança efetiva sem vigilância constante.

Tenha em mente que nenhuma defesa é completamente invulnerável e que **não há substituto para a inquisitividade e a atenção detalhada**.

3. Obtendo Auxílio

Obtendo auxílio na Internet-Br

3.1. Introdução

A Internet/Br ainda não possui mecanismos totalmente consolidados de reação a ataques e invasões.

Apesar de quase uma década de existência da rede somente em julho de 1996, por iniciativa do Comitê Gestor [1] através do GTER [2], que foi criado o Subgrupo de Segurança de Redes (SGTS) voltado para a organização de mecanismos de reação e atuação em situações de emergência.

Os trabalhos deste subgrupo até o momento consistiram em tentar reunir os indivíduos e instituições com interesse na área, diagnosticar os principais problemas relacionados com

segurança de redes no âmbito da Internet/Br e organizar ou tentar propor formas de organização que permitam tornar um pouco mais segura a rede brasileira.

Paralelamente, mas em harmonia com o SGTS, vários indivíduos e instituições tem criado e mantido uma série de mecanismos de auxílio no que diz respeito à segurança.

3.2. Listas Eletrônicas

- **seguranca@pangeia.com.br**

Localizada na Pangéia Informática, é uma lista moderada de inscrição restrita a administradores e responsáveis por redes.

Esta lista visa a divulgação de vulnerabilidades, mecanismos de proteção, discussões sobre segurança de redes e assuntos relacionados. Esta lista é moderada por Nelson Murilo (nelson@pangeia.com.br).

3.3. Centros de Atendimento a Emergências

Algumas instituições e provedores de backbone organizaram centros de atendimentos a emergência relacionadas a segurança no seu âmbito de atuação.

- CAE/RNP → atende o backbone da Rede Nacional de Pesquisa e seus clientes[3];
- CERT-RS → atende o backbone da Rede Tche no Rio Grande do Sul e seus clientes [4];
- GAE-UNICAMP → atende incidentes de segurança com origem ou relacionados à Universidade Estadual de Campinas [5];
- CERT-UNESP → atende incidentes de segurança com origem ou relacionados à UNESP - Universidade Estadual Paulista (cert-unesp@unesp.br)

Adicionalmente, por iniciativa do CG, foi criado em julho de 1997 sob o NIC/Br um ponto de contato e coordenação denominado NBSO (NIC/Br Security Office) cuja finalidade e objetivo é servir de ponto focal de contato em incidentes no
exterior envolvendo a Internet/Br [6].

4. Referências

1. Comitê Gestor, http://www.cg.org.br
2. GTER: Grupo de Trabalho de Engenharia de Redes http://www.cg.org.br/gter
3. CAE/RNP, http://www.cae.rnp.br, carlos@na-df.rnp.br
4. CERT-RS, berthold@pop-rs.rnp.br
5. GAE UNICAMP, guina@obelix.unicamp.br
6. NBSO, nbso@nic.br, nbso@cert.nic.br
7. gwhois, http://www.pangeia.com.br/sm/

O tópico a seguir é relativo a leitura e estudo individual, como exercício complementar. É parte integrante do curso.

5. MAIL e SPAM[1]

5.1. Introdução

Remeter correio eletrônico em demasia a um vasto número de pessoas com propaganda, correntes, esquema de ganhe dinheiro fácil ou fique rico de modo rápido e boatos, é abuso. A esta prática anti-ética, dá-se o nome de SPAM.

Etimologicamente, o termo SPAM vem de um tipo de presunto pastoso industrializado e comercializado nos Estados Unidos e que foi ridiculamente utilizado numa cena de um dos filmes da série Monty Pyton. A cena se passa num bar quando comicamente vikings sentam-se à mesa e ficam repetindo `SPAM! SPAM! SPAM!' sem parar até se tornar insuportavelmente ridículo. Por analogia, seria a repetição de correio para diversos usuários tornando-se ridículo sobremaneira.

Diferentemente do que seja mailbomb, o SPAM visa atingir um número maior de usuários destinatários enquanto que o primeiro visa de modo mais danoso prejudicar a caixa-postal de um específico usuário ou todo espaço em disco do provedor enviando para uma ou mais pessoas correios repetidos em grande quantidade, muitas vezes acompanhado de pornografia ou documentos infectados por vírus.

O SPAM também ocorre em demasia na USENET, dificultando a leitura das mensagens e irritando os moderadores dos grupos por serem repetitivas e principalmente por não respeitarem o tópico a que se refere o grupo. Contudo, não é relevante no contexto brasileiro pois a USENET não é um serviço tão utilizado a nível nacional. A difusão ocorre em sua maioria no desejo inconsequente de atingir um grande número de pessoas para propaganda comercial através de correio. Um modo barato para quem envia e caro para quem recebe.

Em um ataque a AOL, nos EUA, a empresa CyberPromotions enviou 8 milhões de mensagens por dia aos usuários da America Online. Considerando que um usuário comum leva 10 segundos para identificar e descartar a mensagem foram gastas aproximadamente 5.000 horas por dia de tempo de conexão para remoção de SPAM apenas neste caso.

Outra questão relevante é que o SPAM não dá opção de recebimento ao remetente. Simplesmente recebe-se tal lixo, contendo anúncios, na caixa postal pessoal. Muitos deles subscrevem aleatoriamente o usuário na lista de anúncio de determinado produto

[1] Por Isamar Villas Boas Perrelli Maia <isamar@magiclink.com.br>; parte integrante do material **"Tópicos em Segurança de Redes"**. Disponível em http://www.nic.br/book. Reproduzido sob permissão de Pedro A.M.Vazquez <*vazquez@iqm.unicamp.br*>.

acompanhado de uma introdução do tipo: Caso não tenha interesse de continuar na lista, favor remeter correio para ..... solicitando o cancelamento de sua inscrição. Considerando o tamanho e diversidade da rede com seu incontável número de listas de discussão, se houvesse, via de regra, assinatura sem autorização do usuário, em pouco
tempo cada caixa-postal pessoal alcançaria o mínimo de 5 mil mensagens por dia, numa estimativa mais do que otimista.

5.2. Filtros

Esta invasão de privacidade já está sendo combatida em toda a rede e, no Brasil, o número de casos vem aumentando sensivelmente nos últimos meses. Todo e qualquer provedor é passível de ser uma das vítimas diárias.

Uma forma de tentar reduzir e evitar este tipo de prática é a aplicação do recurso de filtro.

Bloquear o recebimento de correio eletrônico de determinados domínios e servidores coniventes com o SPAM e que não respondem de forma alguma a mensagens com reclamações e queixas das práticas indevidas de seus usuários.

5.3. Filtros por Software

Além dos filtros nos roteadores também pode-se utilizar software de transporte adaptado ao não recebimento de SPAM ou software filtro como o ***Procmail*** que utiliza critérios flexíveis de palavra-chave ou limitação de quantidade recebida de uma só
mensagem auxiliando consideravelmente o bloqueio.

Os softwares de transporte de correio eletrônicos comumente utilizados já estão se adaptando a esta nova ameaça, como por exemplo o sendmail e o qmail em plataformas UNIX. Para os usuários de sendmail, algumas regras foram criadas para as versões recentes (8.8.X).

O qmail é uma nova proposta. Seguro, confiável, eficiente e simples. Suporta as melhorias do sendmail, como domínios virtuais, host e user masquerading além de controle de relay, detecção e bloqueio de loops, gerenciamento nativo de listas de discussão, roteamento por domínio, UUCP, mailertable, pop3 nativo com toda a segurança hoje necessária para prevenção de mailbomb, *backdoors* e SPAM.

5.4. Conclusões sobre Mail e Spam

Em suma, judicialmente nos EUA, principalmente no estado de Nevada, existem ensaios de legislação em adaptações ao SPAM em sua versão por FAX.

A Compuserve contra o SPAM provocado pela CyberPromotions é sempre enfocada em discussões do gênero. Recentemente, judicialmente foi estabelecido que a CyberPromotions pagaria 65 mil dólares de advogados e a Compuserve daria 30 mil em retorno em propaganda em seu Web Site.

Urge, no Brasil, a conscientização dos usuários por parte dos provedores estabelecendo Políticas Aceitáveis de Utilização(AUP) e auxiliando vítimas desta constante e desprezível falta de ética.

5.5. Referências sobre Mail e Spam

Fontes:

- http://www.magiclink.com.br/spambr/spam
- http://www.magiclink.com.br/spambr
- http://www-fofa.concordia.ca/spam
- http://ftp.lsoft.com/spamorama.html
- http://www.leg.state.nv.us/97bills/SB/SB13.HTM
- http://www.sendmail.org
- http://www.qmail.org
- http://www.mids.org/nospam

Listas:

- sendm@dl.ac.uk em majordomo@dl.ac.uk Gateway do USENET Group comp.mail.sendmail
- djb-qmail-request@koobera.math.uiuc.edu
- Procmail em procmail@informatik.uith-aachen.de
- Lista Nacional Anti-SPAM spambr-l@magiclink.com.br em majordomo@magiclink.com.br
- Lista Internacional Anti-SPAM spam-l@peach.ease.lsoft.com em listproc@peach.ease.lsoft.com

Exercícios:

Pesquisar e estudar os seguintes assuntos, como parte integrante do curso:

1. O que é, e como funciona, o ***identd***.

2. Procure por ferramentas para fazer scanner para obter MD5 (Dica: Rivest, R. **RFC-1321** *The MD5 Message-Digest Algorithm*, 1992 ou procure na Internet)

3. O que é *ssh* ? (Dica: veja os seguintes URLs: http://public.srce.hr/cigaly/ssh ou ftp://ftp.ietf.org/internet-drafts/draft-ietf-tls-ssh-00.txt ou procure outros documentos na Internet). Procure e instale um cliente ssh para ser usado sobre windows.

Anotações:

Arquivo: **net-sec2.pdf**
Última atualização: quinta-feira, 14 de novembro de 2002 19:12:04