Noções Básicas de SEGURANÇA

by CAT

Marita Maestrelli marita@cbpf.br
Fernanda Santoro Jannuzzi fernanda@cbpf.br
Fernando Spencer spencer@cbpf.br

Dezembro/2003

PREFÀCIO

Esta Nota Técnica foi desenvolvida com o objetivo de servir como fonte de consulta para pessoas que utilizam a Internet e necessitam obter noções sobre segurança.

Informações de terminologia utilizada em segurança de redes são introduzidas com a principal finalidade de orientar os novos alunos de vocação científica e de iniciação científica que ingressam no CBPF.

1. Introdução

O crescimento da Internet trouxe também insegurança para os usuários. A diversidade de sistemas operacionais, hardwares, fabricantes e softwares trazem uma incerteza do que está realmente funcionando em nossos computadores. De acordo com as estatísticas do CERT ("*Computer Emergency Response Team*", órgão que atua na prevenção de incidentes de segurança na rede) foram notificados mais de cem mil incidentes de segurança no ano 2003 contra oitenta mil no ano anterior. A segurança da rede passou a ser uma preocupação de todos, não apenas dos administradores de redes, mas também dos usuários que recebem correspondências eletrônicas infectadas com vírus e *worms*. Diversos especialistas têm se preocupado em construir ferramentas que permitam aos gerentes de sistemas computacionais verificar a segurança e a integridade de suas redes, equipamentos e dados, tais como: nmap, nagios, nessus etc. No entanto, essa é uma tarefa complexa, pois exige uma constante vigilância de serviços disponibilizados pelos recursos computacionais e pelos usuários.

O CBPF (Centro Brasileiro de Pesquisas Físicas) é um instituto voltado para a pesquisa em física, e devido a sua necessidade de comunicação (nacional e internacional), se transformou em um importante ponto de troca de tráfego nacional, regional e metropolitano. Desta forma, está situado hoje no CBPF os pontos de troca de tráfego das redes acadêmicas brasileiras (Rede Nacional de Ensino e Pesquisa - RNP) e metropolitana (Rede-Rio de Computadores).

2. Hacker

O que é?

Primeiramente, vamos identificar o termo: Hacker, originalmente, designava qualquer pessoa que fosse extremamente especializada em uma determinada área. Qualquer fera em qualquer assunto poderia ser considerada um hacker, seguindo esta linha qualquer profissional tinha na sua área pelo menos um hacker. Segundo alguns dicionários uma das definições do termo hack é invadir algo ilegalmente e hacking é o ato de invadir ilegalmente.
Através do cinema americano, o termo *Hacker de Computador* passou a ser utilizado largamente, mas nem por isso perdeu sua identidade.

O filme Jogos de Guerra (Wargames) mostrou Mathew Broderick como um hacker típico: tem um computador no quarto, um problema de comportamento e gosta de fast food. Entra no computador da escola para mudar uma nota que o reprovaria e acessa por acidente o NORAD, sistema de defesa estratégica das forças armadas norte americanas, e começa um "jogo"

chamado Guerra Termonuclear Global. Não sabia que havia iniciado a contagem regressiva para a guerra e que só ele mesmo poderia interrompe-la.

Este filme não foi a base de lançamento de atitudes hacker por todo o mundo, mas foi um dos responsáveis pela dilatação desses pensamentos. O mercado americano lotou as prateleiras de livros como *Cyberpunk*, e mais tarde, qualquer nota sobre invasão de sistemas ou crimes relacionados a computadores ganhava um espaço cada vez maior na mídia. Passamos, então a ter um mundo dividindo o conceito de hacker. Pessoas que trabalhavam em projetos de computadores e técnicos altamente especializados e garotos, que após descobrirem que invadir um sistema ou lançar um míssil não era tão fácil quanto ver um filme ou ler um livro, insistiram e estudaram muito (as maiores virtudes dos hackers são a força de vontade e a dedicação aos estudos), conseguiram muitas proezas e hoje, grande parte trabalha na área de segurança de computadores.

A grande maioria dos hackers é jovem. Dizem que é uma fase da vida de cada "micreiro". E além do mais o jovem tem muito mais tempo para estudar e aprender. Depois que cresce, precisa se preocupar com a vida de verdade e passa a trabalhar (geralmente com computadores), deixando de invadir sistemas ou fazer coisas piores. Os poucos que continuam a praticar atos de hacker são espiões industriais ou especialistas em segurança, e passam a fazer um jogo de gente grande, onde a pessoa vai precisar deter de verdade os invasores perigosos (os espiões), e estes se protegerem do risco de invadir sistemas (e da polícia).

2.1 A Divisão Hierárquica

A hierarquia do mundo underground é muito simples: ou a pessoa é um hacker, ou não. Simples assim: se a pessoa tem conhecimentos aprofundados em qualquer assunto, a pessoa pode se considerar um hacker; caso contrário, se a pessoa não tem nenhuma novidade em nenhum campo da computação ou correlatos, e apenas utiliza o conhecimento dos hackers para fazer suas investidas, ela precisa se superar e passar por várias etapas, até chegar a este nível.

Dentro do fechado e pequeno grupo dos verdadeiros gênios dos computadores, podem-se distinguir três sub-grupos principais:

- **Hacker**: É aquela pessoa que possui uma grande facilidade de análise, assimilação, compreensão e capacidades surpreendentes de conseguir fazer o que quiser com um computador. Ela sabe perfeitamente que nenhum sistema é completamente livre de falhas, e sabe onde procura-las, utilizando técnicas variadas e quanto mais complexa for a descoberta mais valor tem o hacker.

- **Cracker**: Possui tanto conhecimento quanto os hackers, mas com a diferença de que, para ele, não basta entrar em sistemas, quebrar senhas, e descobrir falhas. Ele precisa deixar um aviso de que esteve lá, geralmente com recados malcriados, algumas vezes destruindo partes do sistema, e até aniquilando com tudo o que vê pela frente. Também são atribuídos aos crackers programas que retiram travas em softwares, bem como os que alteram suas características, adicionando ou modificando opções, muitas vezes relacionadas à pirataria.

- **Phreaker**: É especializado em telefonia. Faz parte de suas principais atividades as ligações gratuitas (tanto local como interurbano e internacional), reprogramação de centrais telefônicas, instalação de escutas (não aquelas colocadas em postes telefônicos, mas imagine algo no sentido de, a cada vez que seu telefone tocar, o dele também o fará, e ele poderá ouvir sua conversa), etc. O conhecimento de um phreaker é essencial para se buscar informações que seriam muito úteis nas mãos de mal-intencionados. Além de permitir que um possível ataque a um sistema tenha como ponto de partida provedores de acessos em outros países, suas técnicas permitem não somente ficar invisível diante de um provável rastreamento, como também forjar o culpado da ligação fraudulenta, fazendo com que alguém pague a conta.

Agora, fora destes grupos, temos inúmeras categorias de "não-hackers", onde se enquadra a maioria dos pretendentes a hacker, e a cada dia, surgem novos termos para designá-los. São os principais:

- **Lamers**: Pessoa que quer aprender sobre hackers, e sai perguntando para todo mundo. Os hackers, ou qualquer outra categoria, não gostam disso, e passam a lhe insultar, chamando-o de lamer. Ou seja, novato.

- **Wannabe**: É o principiante que aprendeu a usar algumas receitas de bolo (programas já prontos para descobrir senhas ou invadir sistemas), entrou em sistema simples e acha que poderá utilizar estes recursos para invadir redes com grande estrutura.

Um outro detalhe que vale lembrar é que: os "pseudo-hackers" fazem questão de escrever de forma absolutamente ilegível, trocando letras por caracteres especiais que, segundo eles, se parecem. Além disso, muitas palavras podem ser substituídas por outras com grafia diferentes. Lamers, por exemplo, pode perfeitamente virar Lamerz, Lammerz, Lamah, e etc.

Por incrível que pareça, a maioria das pessoas que acham que são hackers, não são. E uma minoria, que obviamente jura não ter nenhum envolvimento com o underground da computação, são hackers muito experientes mas raramente perigosos.

2.2 O Underground

"Não há poder sem informação" - É com essa frase que esta comunidade se mantém: a eterna busca de mais dados, mais informações, mais sabedoria. O entendimento completo de uma tecnologia é fundamental. O conhecimento completo de todas é o impossível.

O mundo virtual em que habitam os hackers é extremamente complexo. Existem os mais graduados, que são respeitados única e exclusivamente pelo grau de conhecimento que possuem; existem os intermediários, que estão sempre contribuindo com descobertas de pequenas falhas; e não existe hacker "ouvinte". Um hacker precisa de informação exclusiva e secreta, com a qual poderá trocar ou barganhar com outros hackers. Os hackers mais avançados se empenham em desenvolver ataques muito sofisticados, pesquisas de grande porte, e por isso passam um bom tempo recebendo informação de outros piratas da rede e montando gigantescos quebra-cabeças, sem que para isso precisem compartilhar uma informação que não tenha sido exaustivamente testada, não pondo em risco seu projeto.

Estes grandes ataques, quando ocorrem, movimentam toda a comunidade hacker, e principalmente os responsáveis pela segurança em sistemas iguais ao que sofreu o ataque. A circulação da informação de como o ataque foi feito pode levar dias (contando com o levantamento procedimental por parte da vítima) e até que todos os sistemas semelhantes corrijam as falhas que permitiram o ataque, muitos outros sistemas ficam vulneráveis durante horas, pois a informação técnica do ataque no underground hacker é repassada de uma maneira muito mais veloz, já que o procedimento não precisa ser descoberto, apenas refeito.

Podemos perceber, então, que uma informação realmente séria sobre falha de segurança é geralmente válida por muito pouco tempo. Empresas sérias com sistemas importantes corrigem suas falhas de forma quase tão rápida quanto o ataque. E dificilmente um sistema em que a segurança é completamente vital utiliza processos comuns. Geralmente são processos personalizados, de tecnologia própria, o que comunitariamente é muito bom, pois um ataque efetuado naquele sistema não funcionará em nenhum outro, mas por outro lado, apenas os responsáveis por este sistema é que estarão encarregados de detectar e corrigir o problema, coisa que é conseguida de forma absurdamente mais rápida quando muitas pessoas temem sofrer o mesmo ataque.

A tendência, hoje, é a de se formarem grupos, onde há somente uma troca de informação interna, entre os componentes, e está cada vez mais rara a troca de informações entre os grupos. Talvez estejamos presenciando uma corrida para o aperfeiçoamento das técnicas. Um grupo parece estar sempre preocupado em superar o outro.

2.3 As Motivações

O motivo pelo qual os jovens ganham destaque na mídia é a sua captura, pois eles não possuem conhecimento suficiente para que se mantenham ocultos por muito tempo. Deixam rastros por onde passam, pelo descuido e inconseqüência, ou porque simplesmente não têm motivos para se esconderem.

Do outro lado, estão os hackers profissionais, extremamente cuidadosos com suas investidas, e são muito mais difíceis de se detectar e capturar. Afinal, estes não estão mais brincando.

Independente do tipo de hacker, as motivações para seus ataques são bastante variadas, e podemos dividir suas ações em algumas categorias distintas:

- ✓ **Espionagem Industrial**: Pode ocorrer de uma empresa contratar um hacker para que este invada o sistema da concorrência e descubra seus planos, roube seus programas ou até mesmo suas políticas de parcerias e de investimento (geralmente praticadas por hackers profissionais).

- ✓ **Proveito Próprio**: O hacker pode invadir um sistema para roubar dinheiro, transferir bens, cancelar dívidas ou até mesmo ganhar concursos. Qualquer ação em que ele seja diretamente beneficiado.

- ✓ **Inexperiência**: Há também o caso de uma invasão ocorrer por ignorância. Por exemplo, um funcionário que acessa sua conta da empresa através do seu micro em casa. Dependendo da política de segurança da empresa, isto pode ser considerado uma invasão, mesmo que o usuário não tenha conhecimento do problema que pode causar.

- ✓ **Vingança**: Um ex-funcionário, tendo conhecimento do sistema, pode causar vários problemas, se o gerente de segurança da empresa não "cortar" seu acesso imediatamente após sua saída da empresa. Ou, um parceiro de pesquisas pode acessar "mais do que deve" após a quebra de um contrato, trazendo complicações e prejuízos à empresa.

- ✓ **Status ou Necessidade de Aceitação**: Uma invasão difícil pode fazer com que o invasor ganhe um certo status junto aos seus colegas. Isso pode acarretar uma competição, ou uma verdadeira "gincana" na sua empresa. Dentro de grupos, é constante a necessidade de mostrar sua superioridade. Este é um fato natural, seja entre humanos, animais selvagens ou hackers.

- ✓ **Curiosidade e Aprendizado**: Muitos hackers alegam invadir sistemas apenas para aprender como eles funcionam. Alguns fazem questão de testar o esquema de segurança, buscando brechas e aprendendo sobre

novos mecanismos. Este tipo de ataque raramente causa um dano maior ou compromete os serviços atacados.

- ✓ **Busca de Aventuras**: O ataque a sistemas importantes, onde os esquemas de segurança são muito avançados, podem fazer com que o hacker se sinta motivado pelo desafio e pelo perigo de ser pego, assim como alpinistas sobem montanhas, mesmo sabendo do risco de caírem.

- ✓ **Maldade**: Algumas pessoas sentem prazer na destruição. Invadem e destroem, pelo puro prazer de causar o mal. Raramente são pegos e se vangloriam dos seus atos.

Seja o hacker quem for e faça ele o que fizer, é importante que ele seja neutralizado, pelo menos temporariamente, até que seu esquema de segurança seja revisto e atualizado.

Essa atualização precisa ser constante, pois os hackers estão na "cristas da onda" no que dizem respeito às falhas de segurança e, muitas vezes, eles não fazem nada além de invadir sistemas. É extremamente necessário que haja alguém dedicado a este assunto, pelo menos o mesmo tempo gasto por dia pelos hackers nas tentativas de invasão.

2.4 As Ameaças

O simples fato de estar conectado à Internet implica em uma série de possibilidades de atentado à sua segurança. Entre as principais ameaças estão:

- ❖ **Redes Corporativas**: Disponibilizar serviços Internet em uma rede corporativa pode abrir diversos furos de segurança, permitindo que recursos ou informações da empresa sejam acessados de forma indevida por estranhos.

- ❖ **Servidores**: A ameaça de alteração nas informações de servidores pode ser mortalmente prejudicial para uma empresa. Por exemplo, a modificação da especificação de algum produto em um servidor da web pode fazer com que inúmeros negócios sejam perdidos.

- ❖ **Transmissão**: Às vezes pode não ser necessário uma invasão de servidores ou redes. É possível violar a transmissão da informação pela Internet, interceptando-se mensagens, arquivos, senhas etc.

- ❖ **Interrompimento**: Um serviço pode ser atacado de uma maneira menos sutil, mas nem por isso menos perigosa. Pode-se, simplesmente, fazer com que o serviço deixe de funcionar, e que todos os seus usuários legítimos fiquem inacessíveis.

- **Negação**: Em transações pela Internet, ainda temos um outro problema a solucionar. Negar a participação em uma negociação digital também pode ser considerada um ataque ao seu correto funcionamento.

2.5 Hackers Famosos

Kevin David Mitnick (EUA)

O mais famoso hacker do mundo. Atualmente preso, condenado por fraudes no sistema de telefonia, roubo de informações e invasão de sistemas. Os danos materiais são incalculáveis.

Kevin Poulsen (EUA)

Amigo de Mitnick, também especializado em telefonia, ganhava concursos em rádios. Ganhou um Porsche por ser o 102º ouvinte a ligar, mas na verdade ele tinha invadido a central telefônica, e isso foi fácil demais.

Mark Abene (EUA)

Inspirou toda uma geração a fuçar os sistemas públicos de comunicação – mais uma vez, a telefonia - e sua popularidade chegou ao nível de ser considerado uma das 100 pessoas mais "espertas" de New York. Trabalha atualmente como consultor em segurança de sistema.

John Draper (EUA)

Praticamente um ídolo dos três acima, introduziu o conceito de Phreaker, ao conseguir fazer ligações gratuitas utilizando um apito de plástico que vinha de brinde em uma caixa de cereais. Obrigou os EUA a trocar de sinalização de controle nos seus sistemas de telefonia.

Johan Helsingius (Finlândia)

Responsável por um dos mais famosos servidores de email anônimo. Foi preso após se recusar a fornecer dados de um acesso que publicou documentos secretos da Church of Scientology na Internet.

Tinha para isso um 486 com HD de 200Mb, e nunca precisou usar seu próprio servidor.

Vladimir Levin (Rússia)

Preso pela Interpol após meses de investigação, nos quais ele conseguiu transferir 10 milhões de dólares de contas bancárias do Citibank. Insiste na idéia de que um dos advogados contratados para defendê-lo é, na verdade, um agente do FBI.

Robert Morris (EUA)

Espalhou "acidentalmente" um worm que infectou milhões de computadores e fez boa parte da Internet parar em 1988. Ele é filho de um cientista chefe do National Computer Security Center, parte da Agência Nacional de Segurança.

3. Principais Formas de Ataques

Existem inúmeras categorias de ataque a serviços Internet, explorando falhas de software, hardware e principalmente burlando esquemas de segurança ineficazes em nossas políticas operacionais. Algumas categorias mais comuns são:

- **Sniffer**: Captura informações que trafegam pela rede, como logins e senhas em texto não-criptografado, podendo ser utilizadas futuramente por um invasor.

- **Spoof**: Baseia-se na confiança da negociação entre servidores, que acreditam na veracidade do endereço de origem daquela ordem ou informação, e podem sofrer um ataque por mensagens na qual a origem é "disfarçada" como sendo de alguém de confiança.

- **Mudanças de rota**: Um ataque desta natureza pode fazer com que toda a informação de um dado servidor seja obrigada a passar por um espião antes de seguir seu caminho, ou simplesmente sejam redirecionadas para lugar nenhum, causando a "queda" do serviço.

- **Trojan Horse**: Um programa pode ser inadvertidamente instalado em um servidor, o qual permitiria uma invasão por alguma porta ou "brecha" propositadamente implantada neste programa.

- **Replay**: Alguma ação, comandos ou seqüência de eventos podem ser observados durante um processo de autenticação, e repetidas, posteriormente, por um invasor para obter acesso a estes sistemas.

- **Vírus**: À primeira vista pode parecer um simples problema de usuários domésticos, mas os vírus, se não forem devidamente eliminados e controlados, podem causar, direta e indiretamente, vários problemas em uma rede, desde a impossibilidade de comunicação interna, até a interrupção dos serviços vitais(será tratado mais a frente neste estudo).

- **Adulteração**: A falta de um controle de conteúdo eficiente pode fazer com que uma informação seja adulterada durante sua transmissão, pondo em risco a comunicação entre dois sistemas.

Alguns produtos, tais como servidores ou até mesmo sistemas operacionais completos, vêm com a sua configuração padrão de uma forma muito pouco segura, e é necessário conhecer todos os aspectos da nova ferramenta instalada no seu sistema para que ela seja corretamente utilizada.

Programas como servidores de mail, ftp, web ou configurações do sistema, como permissões de arquivos, aplicativos desnecessários (tal como um lpd em um servidor exclusivo de web) devem ser completamente vistoriados antes da sua instalação, procurando se informar sobre suas capacidades de auditoria e sobre seus recursos de segurança.

Pode-se ainda, caso o sistema em questão seja inseguro e necessário, cogitar produtos que possam implementar esta segurança à parte, e dar a mesma importância a eles, instalando-os simultaneamente, e nunca "deixar para depois", pois se alguém detectar uma falha no seu sistema, ela será explorada imediatamente.

E o mais importante: estas configurações padrões, ou os erros mais comuns durante as instalações, são os primeiros alvos dos hackers e, em pouquíssimo tempo seu sistema poderá ser completamente comprometido por pura falta de atenção.

3.1 Vírus

Vírus são programas. No tipo mais comum de vírus eles são programas muito pequenos e invisíveis. O computador (ou melhor dizendo, o sistema operacional), por si só, não tem como detectar a existência do mesmo. Ele não é referenciado em nenhuma parte dos seus arquivos, ninguém sabe dele, e ele não costuma se mostrar antes do ataque fatal. Em linhas gerais, um vírus chega até a memória do computador de duas formas.

A primeira e a mais simples é a seguinte: em qualquer disco (tanto diskette quanto HD) existe um setor que é lido primeiro pelo sistema operacional quando o computador o acessa. Este setor identifica o disco e informa como o sistema operacional (SO) deve agir. O vírus se aloja exatamente neste setor, e espera que o computador o acesse.

A partir daí ele passa para a memória do computador e entra na segunda fase da infecção. Mas antes de falarmos da segunda fase, vamos analisar o segundo método: o vírus se agrega a um arquivo executável (fica pendurado mesmo nesse arquivo). Acessar o disco onde este arquivo está não é o suficiente para se contaminar. É preciso executar o arquivo contaminado. O vírus se anexa, geralmente, em uma parte do arquivo onde não interfira no seu funcionamento (do arquivo), pois assim o usuário não vai perceber nenhuma alteração e vai continuar usando o programa infectado.

O vírus, após ter sido executado, fica escondido agora na memória do computador, e imediatamente infecta todos os discos que estão ligados ao computador, colocando uma cópia de si mesmo no setor que é lido primeiro (boot), e quando o disco for transferido para outro computador, este ao acessar o disco contaminado (neste setor acima citado), executará o vírus e o alocará na sua memória, o que por sua vez irá infectar todos os discos utilizados neste computador, e assim o vírus vai se alastrando.

Os vírus que se anexam a arquivos infectam também todos os arquivos que estão sendo ou que serão executados. Alguns às vezes re-contaminam o mesmo arquivo tantas vezes e ele fica tão grande que passa a ocupar um espaço considerável em seu disco. Outros, mais inteligentes, se escondem entre os espaços do programa original, tornando-se imperceptíveis.

Cada vírus possui um critério para começar o ataque propriamente dito. Os arquivos começam a ser apagados, o micro começa a travar, documentos que não são salvos e várias outras tragédias. Alguns apenas mostram mensagens chatas, outros mais elaborados fazem estragos muitos grandes. Geralmente atacam por data. O mais famoso é o Sexta-feira 13. Mas existem outras datas, tal como o Chernobil (Win.CIH), dia 26 de todos os meses, dia 26 de abril, e etc.

A maior parte dos softwares anti-vírus possui um catálogo detalhado de cada vírus e suas variações, com seus métodos de contaminação, estragos que provocam, etc.

Uma outra espécie de vírus, chamados de "Vírus de Macro", não contaminam arquivos executáveis, mas sim documentos de aplicativos que possuem linguagem de programação, como o Word, da Microsoft. Nestes aplicativos, é possível escrever linhas de código para automatizar uma série de tarefas, e os vírus consistem em programas que alteram as características do aplicativo, retirando opções dos menus, salvando arquivos vazios ou com nomes errados, etc. Mesmo estes vírus, que se escondem dentro de documentos, precisam ser abertos (ou lidos) no aplicativo que o criou para que a contaminação seja feita.

Após contaminado, o aplicativo faz uma cópia do vírus para todos os documentos abertos após a contaminação, alastrando ainda mais o problema. Dependendo do grau de "liberdade" que a linguagem macro contida nestes aplicativos possui, um vírus de macro pode até mesmo apagar todos os seus

arquivos. Os anti-vírus mais novos detectam e corrigem arquivos com estes vírus.

3.2 Mailbombs

O problema do mail-bomb geralmente não é seu, mas sim do seu provedor. Funciona da seguinte forma: cada usuário de um provedor de Internet possui um espaço limitado na caixa postal para receber mensagens. O bomb se refere à esgotar o espaço que lhe destinam, assim, o provedor, com sua caixa postal cheia de mensagens, começa a rejeitar mensagens novas destinadas a você, até que você retire suas mensagens e esvazie sua caixa postal.

O processo de mail-bomb é exatamente isso: várias mensagens repetidas, geralmente contendo um texto malcriado, que lotam seu e-mail e impede que você receba mais mensagens, o que pode ser muito prejudicial para algumas pessoas com negócios importantes que dependem de correio eletrônico.

Os bons programas leitores de e-mail possuem a capacidade de lhe mostrar apenas o Subject, o tamanho e o remetente das mensagens que estão lhe esperando no provedor. A partir daí você pode escolher qual você quer trazer para seu computador, qual você quer apagar do servidor, ou ambos. Isto é muito útil para se prevenir de "baixar" milhares de mensagens que só serviriam para ocupar seu tempo.

Alguns provedores possuem mecanismos "anti-mail-bombs", assim como os serviços de caixa-postal virtual, por exemplo o Pobox.

3.3 Os Cavalos de Tróia Digitais

Baseado na história sobre o presente de grego, um belo cavalo de madeira, que escondia centenas de soldados prontos para acabar com a paz dos troianos, os chamados "programas Cavalo-de-Tróia" são, aparentemente, programas comuns, muitas vezes conhecidos, algumas vezes novos, mas que sempre fazem algo mais do que anunciam.

Seja capturar senhas, causar estragos ou tirar proveito do usuário de alguma forma não documentada. Algumas pessoas têm confundido Cavalos de Tróia com Backdoors. Os conceitos são parecidos, mas existem diferenças fundamentais. Por exemplo: o Back Orifice e o NetBus são Cavalos de Tróia, pois não possuem nenhuma finalidade prática para a vítima. Já o ICKiller, que derruba usuários do ICQ possui uma backdoor, pois há uma porta dos fundos que permite certas invasões.

3.4 Worms

Os "worms" são um tipo de vírus que enviam cópias de si mesmos para outros computadores, agravando o quadro de ataques em massa. O ILoveYou é um exemplo deste tipo de vírus e de worm, que gerou milhões de dólares em prejuízo às empresas. Os fabricantes de anti-vírus alertam para o fato de que seus usuários devem manter seus programas sempre atualizados para evitar surpresas desagradáveis.

Em países como o Brasil, onde a segurança é quase inexistente, os ataques deste vírus são em grande escala, através da lista de contatos do programa de email da vítima.

Ex: o "worm" Hybris atinge somente os sistemas Windows, infectando a biblioteca WSOCK32.DLL e as funções do sistema que estabilizam as conexões de rede, incluindo a internet. Hybris faz uma busca por endereços de email e envia uma cópia de si mesmo a todos que encontrar.

3.5 Backdoors

Antigamente, todo sistema tinha uma backdoor (ou "porta dos fundos"), por onde o criador do sistema poderia entrar a qualquer hora, independente de quem fosse o responsável pelo sistema, desta forma, o autor garantiria sua entrada por esta porta onde somente ele saberia o local exato.

Até algum tempo atrás, programadores deixavam uma porta ou passagem. Hoje isso é mais difícil, pois existem muito mais pessoas interessadas estes espaços.

De qualquer forma, quando você ouvir falar em "encontrar uma backdoor" ou "deixar uma backdoor em algum sistema" lembre-se de que, geralmente, estão se referindo a uma característica secreta e obviamente não documentada que permite o acesso de quem implantou esta backdoor.

Lembre-se de que é perfeitamente possível existir backdoors em scripts de IRC, programas de FTP, ou qualquer coisa que você encontrar pela Internet.

Algumas versões de Cavalos de Tróia possuem a capacidade de se anexar a outros programas, fazendo com que este se transforme em uma backdoor, sem que você perceba nada de diferente em seu sistema. Pelo menos até alguém entrar tentar alguma coisa.

3.6 E-mails Anônimos

A Internet, por concepção, facilita muito a comunicação entre computadores e pessoas. Acontece que, em prol da facilidade, deixamos de ter segurança. Um bom exemplo disso é o e-mail. Ele foi feito para nos comunicarmos uns com os outros, inicialmente para troca de informações

acadêmicas, e hoje, é muito fácil ludibriar o protocolo usado pela correspondência eletrônica para que ele assuma como remetente da mensagem quem quisermos que seja. Ele não verifica o endereço de quem enviou a mensagem, apenas pergunta qual é.

Nos programas de e-mail, além de preencher todos os campos normais de uma mensagem, eles permitem que você preencha também o campo "from:", ou seja, ele deixa você se passar por quem você quiser. E também em que servidor de e-mail você deseja mandar a mensagem. O único método de autenticarmos uma mensagem como sendo de quem diz ser é através da criptografia e assinatura eletrônica.

Lembre-se que é sempre possível identificar quem enviou a mensagem. Nos programas de e-mail normais, muita informação sobre a mensagem – o cabeçalho - não aparece (para não fazer confusão), mas este cabeçalho contém todo o caminho percorrido pela mensagem.

Tornando o nosso exemplo mais simples, a mensagem passou por apenas dois servidores, o que recebeu a mensagem, e o seu. Você pode consultar no seu servidor (pelo número da mensagem, ou horário), qual foi o servidor que colocou esta mensagem lá.

Descobrindo isso, partimos para o segundo servidor (de smtp) e descobrimos o IP da pessoa que colocou a mensagem naquele servidor de e-mail (ou seja, o número IP da conexão aberta no servidor naquele horário específico).

Provavelmente seria o IP de um terceiro provedor (de acesso). Bastaria descobrirmos qual o usuário que estava conectado no modem deste terceiro provedor (cada modem do provedor possui um IP) no horário em que a mensagem foi posta do servidor de e-mail e pronto. Teríamos o IP do autor da mensagem.

Obs.: Em alguns casos, servidores SMTP erroneamente configurados permitem a postagem de mensagens para fora do seu domínio, um fato conhecido como "relay". Estes servidores ajudam a manter o anonimato, e devem ser desativados ou corrigidos.

3.7 Ataque IP Spoofing

O IP Spoofing ficou famoso após ter sido a atração principal do ataque à rede de Tsutomu Shimomura, um dos maiores especialistas de segurança dos Estados Unidos, quando através dele, na noite de natal de 1994, o mais famoso e procurado hacker americano, Kevin Mitnick, invadiu a sua rede particular e roubou alguns dos seus programas, dentre eles, um software de programação dos celulares Oki, que lhe permitiria ter um controle total sobre suas ligações clandestinas e escutas de celulares.

Esta história, é claro, foi desmentida por Kevin, que apesar do sabido interesse no referido telefone, não demonstrava aptidão suficiente para realizar um ataque desta grandeza. Apesar deste tipo de ataque ter sido teorizado há muitos anos, somente na metade desta década ele ficou mundialmente conhecido e, em 1997 já era o 4° método de invasão mais utilizado. Hoje encontram-se ferramentas que automatizam esta fatigante tarefa, e tem-se conhecimento de ataques de spoofing concretizados em menos de meio minuto.

Disfarce. É isto que este ataque faz. Muitas comunicações entre computadores na Internet se baseiam em "parceiros" confiáveis. Um computador X pode manter uma comunicação com um computador Y de forma que não seja necessária a constante verificação de autenticidade entre eles. O hacker, então, se disfarça, dizendo para o computador X que "ele" é o computador Y. Desta forma o computador X vai aceitar seus comandos tranqüilamente, enquanto ele faz uso indevido do sistema.

Um pouco mais de detalhes: os pacotes IP possuem um endereço destino e um endereço origem.

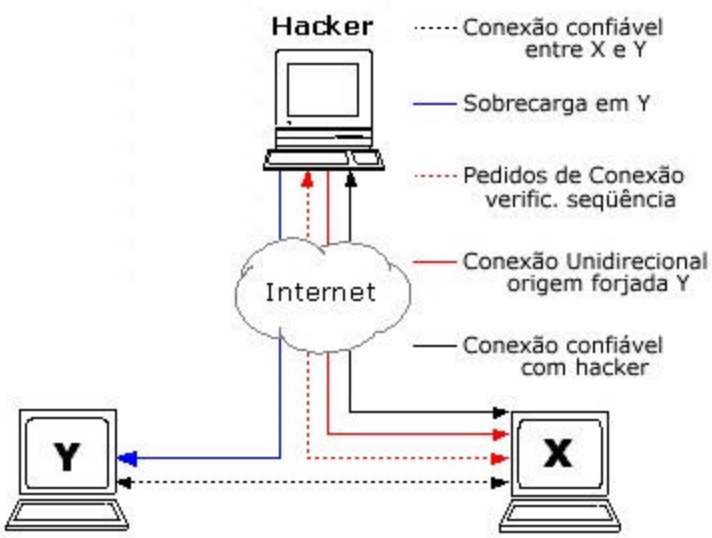
Normalmente o endereço origem reflete a realidade, mas nada impede que um hacker altere este pacote para que ele pareça ter vindo de outro lugar. Além de enganar o destino, neste caso o computador X, é necessário que se sobrecarregue o computador Y, para que ele não responda às mensagens de X, cancelando assim sua conexão.

Faz-se necessária, ainda, uma "previsão" do número de seqüência mandado por X. Este número é enviado por X ao originador da conexão (supostamente o Y). Mas Y não irá responder, devido à sobrecarga explicada anteriormente. Então o hacker deve prever o número de seqüência mandado por X para que seja enviado um novo pacote com estes números de seqüência, fingindo novamente ter sido enviado por Y, e forjando a autenticação.

A previsão deste número de seqüência é um processo demorado e criativo. Durante a negociação da conexão, os computadores trocam informações para efetuarem o "handshake" ou "aperto de mão". Dentre as informações trocadas estão os números de seqüência, que devem ser repetidos para o destino, para que este se certifique da autenticidade da conexão. O que o hacker pode fazer é o seguinte: enviar, através de um pacote legítimo, com o endereço de origem verdadeiro, vários pedidos de conexão à X. Este responde com um número de seqüência para que o hacker o repita e efetue a conexão, mas a máquina de origem (o hacker) não tem privilégios e não lhe interessa fechar esta conexão. Então ele não responde a estes pacotes de X, apenas os guarda e verifica seu número de seqüência.

Após vários pedidos de conexão com X, o hacker pode "aprender" como X gera seus números e então mandar um pedido de conexão, desta vez com o endereço de origem sendo Y (o computador confiável).

O hacker não vai receber os pacotes de X com os números de seqüência, pois estes irão para o endereço de origem (computador Y, que, a esta altura, precisa estar sobrecarregado para não respondê-los), mas, com base nos cálculos anteriores, o hacker prevê e manda o número de seqüência correto para o computador X, fechando a conexão.



Esta conexão é unidirecional, pois todas as respostas de X serão destinadas ao computador Y, e não ao computador do hacker. Então o hacker age "às cegas", pois não recebe nenhum tipo de retorno de X, pelo menos enquanto configura X para aceitar conexões do seu próprio computador. Ao terminar, o hacker tira o disfarce: desfaz a conexão falsa com X, e faz uma legítima, agora que X pode aceitar conexões confiáveis através do computador do hacker, com todos os privilégios possíveis e imagináveis.

Há dois inconvenientes neste ataque: o trabalho de achar um padrão nos números de seqüência e a falta de retorno do computador invadido. Entretanto, existe ainda uma outra categoria de spoofing, que, apesar das condições pouco comuns, é muito mais eficiente. Neste tipo de ataque, um computador de pouca importância em uma rede é invadido por um método simples qualquer e, estando localizado no mesmo meio físico onde ocorre uma comunicação entre duas máquinas importantes, ele oderá "captar" todo o tráfego deste meio físico. Com isso ele saberá os números de seqüência de ambos os lados, podendo interceptar este diálogo, pondo-se no lugar de um dos computadores da conexão confiável, tendo total acesso aos dados mais sigilosos.

Este tipo de ataque só pode ser eficazmente prevenido através de um controle apurado de TODOS os computadores da rede local, que compartilham o mesmo meio físico, ou através de uma comunicação confiável baseada em aplicativos de criptografia.

3.8 Login e Senhas

Alguns usuários possuem níveis maiores, onde podem fazer muitas coisas; outros, níveis menores, onde não podem fazer quase nada.

O responsável por um sistema desses, jamais daria um tipo de acesso "livre" para quem não tem responsabilidades para com o sistema, mas todos eles possuem acesso ao arquivo de senhas, que é um arquivo geral que contém as senhas de todos os usuários. Então, alguém pode pegar ilegalmente este arquivo de senhas.

A senha, evidentemente, é codificada e não pode ser lida de forma alguma depois da codificação. Não existe maneira de se recuperar a senha original apenas com o resultado de sua codificação no arquivo de senhas. Então, como o sistema faz para saber se a senha que o usuário digitou está correta? Toda vez em que o usuário tentar entrar no sistema, a senha que ele digitar é codificada, e o resultado comparado com o que consta no arquivo de senhas. Se for igual (o cálculo de codificação é sempre o mesmo) então o acesso àquele usuário é liberado.

O usuário mal-intencionado pode codificar uma palavra, se o resultado for igual ao código constante do seu arquivo de senhas, ele sabe, por referência, qual a palavra (senha) que deve ser utilizada. Então ele tenta várias palavras até encontrar alguma que se encaixe na codificação.

Imagine se todas as senhas fossem numéricas, de apenas 4 dígitos? Seria muito simples criar um arquivo que contivesse todas as combinações possíveis dos números em 4 posições diferentes. Pronto o arquivo, codifica-se cada uma delas, depois é só comparar para ver qual a senha de cada usuário.

Acontece que nenhum sistema sério utiliza regras tão falhas para as senhas. Muitos, inclusive, proíbem até o uso de palavras que existam no dicionário. Assim se evita que alguém codifique um arquivo contendo palavras do dicionário e descubra as senhas. Quanto maior em comprimento e mais variados forem os caracteres da senha, melhor, ou seja, sua senha deve ser absurdamente complexa, evitando palavras evidentes, tais como: login, a placa do seu carro, data de nascimento, pois ela não pode ser lembrada (pelos outros).

Importante - De vez em quando encontramos pela Internet uns formulários que geralmente são chamados de Zero Hora, ou algo do gênero. Segundo os autores das páginas, ele é um script que explora uma tal falha nos servidores (em todos) que fazem com que você possa zerar sua utilização no provedor. Eles pedem seu login, o endereço do seu provedor e, é claro, sua senha.

4. Comunicação na Rede

4.1 O Protocolo TCP/IP

Qualquer comunicação precisa de padrões para que as partes comunicantes se entendam. Com a rede e os computadores acontece da mesma forma: ao se comunicarem, os computadores precisam trocar dados dentro de uma padronização conhecida por ambos.

Existem várias padrões escolhido para essa comunicação entre os nós da internet, que receberam o nome de protocolo, cada um com suas peculiaridades. O protocolo TCP/IP, que é padrão de comunicação para Internet.

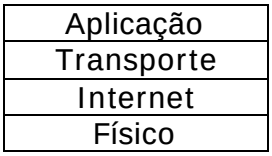
Devemos ter em mente "como" estes computadores vão se interligar para trocar informações. Existem dois modos básicos para isto: a comutação de circuitos e a comutação de pacotes. Na comutação de circuitos, os computadores se ligam diretamente para a troca de informações, e na comutação de pacotes, os computadores mandam a informação para pontos intermediários até alcançarem seu destino.

Enquanto na comutação de circuitos os computadores podem trocar grandes volumes de informações, de forma direta e contínua, na comutação de pacotes, os computadores mandam pequenos pedaços (pacotes) de informações que, por não irem diretamente ao destino, precisam conter - em cada pacote - o endereço de origem e destino completos.

O protocolo TCP/IP foi criado para atender algumas necessidades computacionais básicas. Primeiro, as redes a serem construídas não poderiam parar, mesmo que alguns dos computadores desta rede fossem destruídos; Segundo, os computadores não poderiam estar, todos, fisicamente ligados (por motivos de custo); Terceiro, esta rede possuiria vários tipos de computadores diferentes, e eles precisariam se comunicar.

Não foi preciso muito esforço para perceber que uma comutação de circuitos seria cara e frágil, principalmente sob ataques nucleares, por causa da falta de ligações redundantes, o que seria extremamente útil e fácil de implementar em uma comutação de pacotes. Portanto, foi criado um protocolo que se comunica por pacotes: o TCP/IP. E que possui um alto grau de interoperabilidade.

Este protocolo foi criado em "camadas", ou seja, dentro do próprio protocolo, existem processos bem definidos, cada um fazendo sua tarefa na comunicação, e cada processo se comunicando com o seguinte através dos níveis que os separam.



O modelo de camadas tem uma vantagem notória: se os processos de comunicação estão bem definidos e separados em cada camada, qualquer alteração necessária em um destes processos poderia ser feita isoladamente, não precisando, portanto, rescrever todo o protocolo.

- **Físico** - O nível físico, na verdade, é de responsabilidade da placa de rede que, dependendo do meio ao qual está ligada, enviará um pacote diferente. Por exemplo, se for uma placa Ethernet, enviará os quadros padrão IEEE802.3, se for ATM, Frame Relay, Token Ring, enviará seus quadros específicos. O importante nesta camada, para o funcionamento do TCP/IP, é a maneira com que a camada superior se comunica com ela.

- **Internet** - Também chamada de "Interredes", esta camada é a responsável pelo direcionamento do tráfego dos pacotes. É nela que são identificados os endereços IP. A propósito, IP quer dizer "Internet Protocol", que serve para as comunicações intermediárias entre um computador e outro. Esta camada possui também um protocolo para envio de mensagens de controle e de erro, que é enviado para as camadas "Interredes" de outros computadores, chamado ICMP (Internet Control Message Protocol). O ICMP tem a função de, por exemplo, checar a existência de um endereço. Note: os pacotes ICMP são enviados dentro dos pacotes IP, assim como os da camada de transporte, mas o ICMP é gerado dentro na própria camada "internet".

- **Transporte** - No nível de transporte, não interessa mais como o pacote vai chegar ao seu destino. O que importa para este nível é a comunicação fim-a-fim, host-to-host. Esta camada é responsável pelo controle de fluxo, impedindo que o protocolo se afogue no meio de tantos pacotes. Ela também organiza os pacotes que chegam (enviados a ela pela camada Internet), pois eles podem chegar fora de ordem. Após colocar tudo em ordem, se a camada Transporte der falta de algum pacote, é ela que pede o seu reenvio. Na hora de enviar pacotes, é esta camada que quebra as informações em pequenos pedaços, antes de repassá-la. Ela possui dois tipos de pacotes, o TCP e o UDP.

- **Aplicação** - Os responsáveis por esta camada são os programas que nós utilizamos na Internet. Por exemplo, o navegador, possui a camada de Aplicação do protocolo TCP/IP. Ele está utilizando o "protocolo de aplicação HTTP", assim como o seu programa de FTP utiliza o "protocolo de aplicação FTP" e seu programa de e-mail o "protocolo de aplicação SMTP e POP".

4.2 Encapsulamento

Cada pacote de nível no TCP/IP possui uma área de controle do pacote e uma área de dados. A área de dados carrega a informação relevante para nós (como aquele GET /etc/passwd), e a área de controle, carrega informações relevantes para a comunicação entre os computadores.

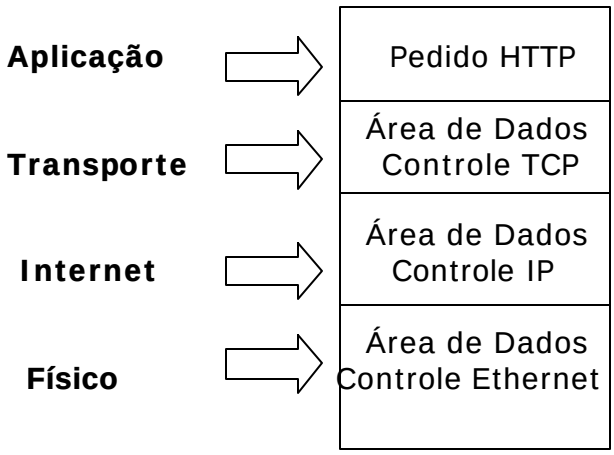
A área de controle do pacote IP possui um campo chamado TTL (Time To Live) que indica aos outros computadores o quanto aquele pacote deve caminhar até ser descartado. Em cada host de nível Internet pelo qual passa este pacote, o número do campo TTL é subtraído de 1 e, quando chegar a zero, ele deixa de ser retransmitido. O número padrão deste campo é 30, e serve para que o pacote não fique eternamente perambulando pela rede, caso não encontre o caminho certo.

Os pacotes dos níveis altos são "encapsulados" da área de dados do pacote imediatamente inferior, que por sua vez é encapsulado na área de dados no nível abaixo, até chegar ao nível físico, onde então é enviado pela rede.

O envio de uma informação passa por 4 processos de empacotamento, cada um responsável por um aspecto da comunicação, acrescentando sempre as informações necessárias para que a informação original chegue até seu destino, não se importando com o que está sendo "carregado" na área de dados. A área de dados não é de responsabilidade do nível corrente e sim do nível superior, portanto, o que se deve fazer é apenas levar aquela área de dados até o outro lado da comunicação.

Nível Físico
Controle Ethernet
Nível Internet
Controle IP
Nível Transporte
Controle TCP
Nível Aplicação
Controle HTTP

Veja esta outra tabela:



Quando você digita um endereço no seu programa de internet, vamos supor, no browser, ele monta uma instrução do tipo "Protocolo de aplicação HTTP". Esta instrução é referente a um pedido de arquivo feito por você a algum servidor Web em algum lugar do planeta. Então, seu navegador passa esta instrução aos seus drivers de rede, mais precisamente à camada de Transporte. A camada de transporte aloja esta instrução HTTP em sua área de dados, e monta um cabeçalho com informações relevantes sobre o tipo de conexão que será mantida entre os dois computadores para a transmissão deste arquivo. Após especificar de que forma os computadores irão trocar esta informação, ele pega sua área de dados (aquela instrução HTTP), acrescida destas informações sobre a conexão, e a repassa para a camada inferior, internet, que alocará esta informação em sua área de dados. Esta camada acrescentará informações referentes ao endereço de destino (o servidor Web) que receberá estes dados, assim como preferências de rotas e dados de controle do pacote enquanto ele estiver a caminho do seu destino. Estando este cabeçalho pronto e anexado a área de dados, esta informação é agora passada ao driver da placa de rede, que montará um quadro de envio compatível com o meio físico ao qual está ligado.

Chegando ao destino, independente do tipo de pacote do nível físico (o que refletirá o tipo de rede utilizada na instalação do servidor Web), a placa de rede instalada neste servidor abrirá este pacote e repassará sua área de dados ao nível superior. A área de dados do nível físico, como sabemos, é o pacote do nível internet, que será repassado justamente ao nível internet. Este nível, por sua vez, checa todos os dados de controle contidos no cabeçalho, como endereço destino e origem, e repassa sua área de dados (o pacote do nível Transporte) ao nível superior que é o Nível de Transporte. Este nível de transporte, ao receber seu pacote, checa os dados do cabeçalho quanto aos tipos de conexão, checksum, etc. e, estando tudo em ordem, repassa sua área de dados (o comando HTTP) ao servidor Web, que responde pelo Nível de Aplicação. Ao receber o comando, o servidor Web busca o arquivo pedido pelo seu browser e o envia, fazendo novamente todo o processo de empacotamento e transmissão, até chegar ao seu computador, começando o mesmo processo.

Ao contrário do que pode parecer, isto é feito em pouquíssimo tempo - desde que tudo esteja funcionando pelo caminho. A maior parte dos "gargalos" nestas comunicações estão nas extremidades, seja pela velocidade do seu modem (link do seu provedor), ou pela sobrecarga no servidor requerido. Mas há a possibilidade de acontecer algum problema enquanto seu pacote de informação estiver a caminho, e ele precisa saber resolver caso encontre um destes problemas. Se ainda assim, seu pacote não chegar ao seu destino, precisamos saber onde e porque isto ocorreu, e este é o principal motivo para tantas informações de controle nos cabeçalhos, pois os pacotes não possuem nenhuma técnica que possibilite-o seguir adiante. Tudo fica a cargo dos GATEWAYS, não importando o que aconteça.

4.3 Gateways

Os gateways são equipamentos computacionais que ligam outros equipamentos computacionais para que estes troquem informações entre si. Tem sua forma retangular, e possui leds que indicam o estado da conexão com aquele determinado nó, e vários cabos de rede conectados que os conectam. Algumas vezes também, os gateways podem ser computadores, rodando programas de comunicação.

Gateway é um nome genérico dados a estes equipamentos, apesar de muitos deles terem seus próprios nomes. Isso faz com que os gateways recebam diferentes nomes é a capacidade que eles possuem em se aprofundar na análise do tráfego de informação que transporta. Ex: Um HUB tem a função de repassar o sinal elétrico recebido em uma de suas portas para todas as outras portas. Ele não se importa com o que está sendo transmitido, apenas "escuta" de um lado e "fala" do outro. Portanto, como não "entende" os pacotes que transfere, ele não é considerado um gateway. Ele não analisa o tráfego e não toma nenhuma decisão.

Já as Bridges, quando recebem o pacote de informação, verificam "para onde" este pacote vai e o encaminha para a saída correta. A Bridge faz este endereçamento baseado no endereço FÍSICO das placas de rede as quais ela tem acesso (algo como 30:A5:F3:00:21:6C:E0:FF), portanto, basta que se entenda o pacote do nível físico para fazer o encaminhamento. Ela então é considerada um "Gateway de Nível Físico".

Os Roteadores (ou Routers) vão além: eles abrem o pacote do nível físico e checam o pacote do nível Internet. Neste segundo pacote, eles descobrem o endereço IP do destino, e encaminham o pacote devidamente. Com isto, os roteadores podem ser usados para interligar redes físicas diferentes, bastando que, para isto, ele "saiba ouvir e falar" quadros de níveis físicos distintos. O fato de "entenderem" os pacotes do nível Internet os fazem receber o nome de "Gateway de Nível Internet".

Gateways de Nível de Transporte são capazes de analisar o tipo de pacote endereçado à máquina-destino da comunicação (muitos roteadores fazem isto, apesar de não ser sua função principal) e Gateways de Nível de Aplicação abrem todos os níveis de pacotes até chegar à informação original pedida pelo remetente.

Um exemplo deste gateway são os servidores Proxy Http, que mantém um estoque dos arquivos mais pedidos pelos usuários e, analisando o novo pedido que chega, sequer deixa o pacote sair da rede, respondendo com a cópia local deste arquivo. Para isto, o Proxy precisa "enxergar" o pacote do nível de aplicação.

4.4 Criptografia

A criptografia é uma arte: a arte de escrever ocultamente. Talvez tão antiga quanto a própria escrita, hoje é um dos métodos mais eficientes de se transferir informações, sem que haja a possibilidade de comprometimento do sigilo.

Baseada em chaves, uma informação pode ser codificada através de algum algorítimo de criptografia, de modo que, tendo conhecimento do algorítimo utilizado e da chave utilizada, é possível recuperar a informação original fazendo o percurso contrário da encriptação, a decriptação.

Com o aumento da capacidade computacional, podemos hoje utilizar complexos esquemas criptográficos, que antes eram impraticáveis pela demora com os quais eram codificadas pequenas informações. E além da capacidade técnica, possuímos algumas características na criptografia moderna que a faz se subdividir em dois grandes grupos: Criptografia de Chave Simétrica e Criptografia de Chave Assimétrica.

Criptografia de Chave Simétrica

Esta é a criptografia tradicional, onde a mesma chave utilizada na codificação deve ser utilizada da decodificação. Alguns algorítimos de criptografia de chave simétrica: IDEA (International Data Encryption Algorithm), DES (Data Encryption Standard) da IBM e o RC2/4, da RSA Data Security.

O problema dessa simetria é: como vamos informar ao destinatário a chave para a decriptação de forma segura. Se encontrarmos um modo seguro de lhe contar a chave, não poderíamos utilizar este modo para lhe passar a informação de uma vez. Este não é o melhor método para trocarmos dados importantes e/ou sigilosos pela rede.

No entanto, a criptografia simétrica é bastante eficiente em conexões seguras na Internet, onde processos computacionais trocam senhas temporárias para algumas transmissões críticas , por exemplo: navegar pela Internet e visitar sites , onde geralmente são preenchidos dados sigilosos, que está utilizando o SSL (Secure Sockets Layer) que funciona à base de criptografia simétrica, muito provavelmente um DES ou algo da RSA.

Criptografia de Chave Assimétrica

Estudos realizados há 20 anos tornaram possíveis algorítimos de criptografia utilizando duas chaves. Criptografando-se com a chave A, só seria possível a decriptação com a chave B, sendo a recíproca verdadeira.

Chave Pública e Chave Privada

Esta assimetria dá uma outra abordagem, a de chave pública e chave privada. Com duas chaves, não precisamos ficar presos a uma "troca" para o processo de decodificação e codificação. Cada um poderá possuir sua chave pública e sua chave privada. Como o próprio nome já diz, a chave privada é de conhecimento único e exclusivo seu. Já a pública deve estar disponível a quem quiser lhe enviar informações encriptadas.

Como a encriptação/decriptação depende das duas chaves, se você quiser, por exemplo, nos mandar uma mensagem criptografada, deve encriptá-la com a nossa chave pública. Como dito anteriormente, a única chave que decripta esta mensagem é o par da chave pública, ou seja, nossa chave privada.

Somente nós conseguiremos ler a mensagem (desde que nossa chave privada seja mantida em um lugar seguro).

Continuando o exemplo, caso nós queiramos mandar uma mensagem criptografada para você, nós primeiro conseguimos uma cópia da SUA chave pública e a uso na encriptação. Somente sua chave privada poderá decriptar

esta mensagem, e mesmo que a mensagem tenha sido interceptada, não passará de um conjunto de caracteres estranhos.

PS: Lembre-se que uma mensagem comum, em texto simples, pode ser lida em qualquer parte da Internet, em especial no seu provedor de acesso. Apesar da política na maioria dos serviços proibir a leitura de mensagem dos usuários, ela é praticamente possível. Se você tem assuntos secretos a tratar por email, faça uso da criptografia. (existem sistemas de criptografia por telefone, mas ambos os lados precisam possuir o mesmo sistema, e a criptografia é por chave simétrica)

Assinatura Digital

Realmente, todos os que possuírem nossa chave pública poderão ler nossa mensagem, mas também é verdadeiro que, se foi possível decriptar com nossa chave pública, é porque ela foi encriptada com nossa chave privada. Como as únicas pessoas que sabem a nossa chave privada somos nós, está assegurada a nossa identidade como autores daquela mensagem.

Criptografia + Assinatura Digital

Pode-se ainda, com o uso de quatro chaves, criptografar e autenticar a mensagem. Primeiro nós criptografamos nossa mensagem com a NOSSA chave PRIVADA. Depois criptografamos novamente a mensagem, desta vez com a SUA chave PÚBLICA. Ao receber, você deverá usar sua chave privada para decriptar a mensagem e, após, usar a nossa chave pública, para decriptá-la novamente e se assegurar de que fomos realmente nós que lhe escrevemos esta mensagem.

Neste caso, a ordem com que usamos a nossa chave privada e a sua chave pública faz diferença. Veja bem: no exemplo anterior, você não tem como provar que fomos realmente nós que enviamos esta mensagem até que você a decripte com sua chave privada. Se nós tivéssemos invertido a ordem, primeiro encriptando a mensagem com sua chave pública e depois assinando com nossa chave privada, todos saberiam que fomos realmente nós que escrevemos aquela mensagem, apesar de só você conseguir ler o seu conteúdo.

Simétrica, Assimétrica

Utilizando criptografia com chave assimétrica encontramos um problema: ela é muito lenta. É preciso um bom computador para que o tempo de criptografia se torne viável, pois um texto grande pode levar de alguns minutos a várias horas. Já a simétrica é rápida, mas possui o problema da chave única.

Para resolvermos o problema podemos criptografamos uma mensagem com uma chave qualquer simétrica. A mensagem é grande, mas a simétrica é rápida. Então pegamos essa chave, que é pequena, e criptografamos

assimetricamente com a SUA chave pública. Só você saberá qual é a chave (decriptando-a com sua chave privada), e poderá decriptar a mensagem com a "nossa" chave simétrica.

Message Digest

Existe uma maneira de se criar um código a partir de uma mensagem, que reflita o seu conteúdo em um pequeno conjunto de caracteres. Aplica-se um cálculo na mensagem e este cálculo nos retorna um "message digest", como se fosse uma impressão digital da mensagem. Qualquer simples troca e o message digest será diferente.

Alguns algorítimos que fazem a "extração" do message digest mais utilizados no mercado: MD4/5 e o SHA (Secure Hash Algorithm).

O cálculo do message digest possui duas características fundamentais: não pode ser possível inverter o cálculo sobre o message digest para recuperar a mensagem original; e o message digest deve ser único por mensagem, ou seja, não pode existir um mesmo message digest para duas mensagens diferentes.

Com essa nova ferramenta, pode-se assegurar a autenticidade com criptografia assimétrica, mas nos poupando tempo. Ex: uma mensagem "não-secreta" para lhe enviar: Primeiro calculamos o message digest da nossa mensagem. Encriptamos o message digest com a NOSSA chave PRIVADA, e lhe mandamos a mensagem junto com o message digest encriptado. Ao receber a mensagem, você calcula o message digest da nossa mensagem, e compara com o message digest encriptado (utilizando nossa chave pública para decriptá-lo). Sendo igual, você terá certeza de que fomos nós que escrevemos aquela mensagem.

Mas a mensagem em si foi enviada sem criptografia. Qualquer um poderia ter lido.

Geral

Primeiro criamos um message digest da nossa mensagem e o encriptamos com a NOSSA chave PRIVADA.

Depois, escolhemos uma chave SIMÉTRICA qualquer e encriptamos a mensagem inteira. Então nós encriptamos a chave simétrica com a SUA chave PÚBLICA. Agora enviamos tudo para você: o message digest da nossa mensagem encriptado com nossa chave privada, a mensagem criptografada com a chave simétrica, e a chave simétrica encriptada com a sua chave pública.

Ao receber a mensagem, você deverá fazer o seguinte: decriptar a chave simétrica com a sua chave privada. Com a chave simétrica decriptada, decripte a mensagem. Agora decripte nosso message digest com nossa chave pública e calcule o message digest da mensagem original. Finalmente,

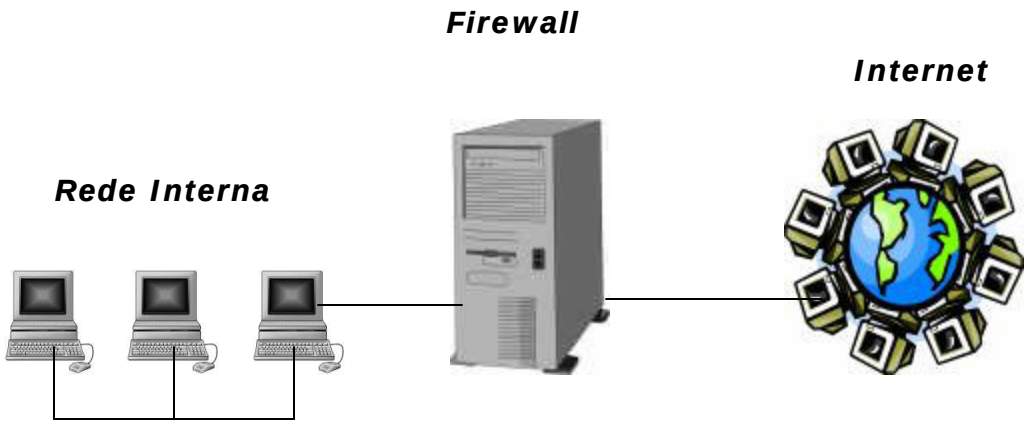
compare os dois message digest: o que lhe mandamos e o que você calculou. Se forem iguais, você terá certeza de que fomos nós que enviamos a mensagem e que a mensagem não sofreu nenhuma alteração.

5. Firewalls

Uma das grandes preocupações na área de segurança de redes é a vulnerabilidade de um computador que pode comprometer as transmissões pelo meio físico da rede na qual está ligado. Muito se tem feito para que o host (equipamento computacional) esteja seguro isoladamente, impedindo o acesso indevido a seus dados e monitorando qualquer tentativa de invasão. Entretanto, um outro método tem se mostrado bastante eficiente: impedir que informações indesejadas entrem na rede como um todo.

Não é um método substituto à segurança do host, mas complementar, e consiste no seguinte: na ligação da rede interna com a Internet, instala-se um equipamento que permitirá, ou não, a entrada e saída de informação, baseado em uma lista de restrições e permissões, devidamente configurada para suprir as necessidades básicas de comunicação da rede interna com a Internet e vice-versa. Nem mais, nem menos. Esta configuração é a chave do sucesso ou fracasso de um firewall.

É importante lembrar que o firewall deve estar presente em todas as conexões da rede interna com a Internet. Não adianta nada colocar um firewall super sofisticado na ligação do backbone se, dentro da rede interna, existir um micro com um modem conectado em outra rede.



Nota: Firewall não é "Parede contra fogo", como muitos dizem, e sim "Parede de fogo", feita para impedir a passagem de alguém (ou de algo).

Aviso: É extremamente recomendado que se tenha conhecimentos de TCP/IP, endereçamento IP, portas de conexão e pacotes TCP/UDP/ICMP, antes de prosseguir na leitura deste tópico.

5.1 A Filtragem de Pacotes

Essa é a maneira mais simples de se construir um firewall. Geralmente utilizadas em roteadores, as listas de acesso têm uma ótima relação custo x benefício: os roteadores já possuem estas facilidades, basta sentar e aprender a configurá-los; a filtragem é bem eficiente, invisível e rápida (se o roteador for de boa qualidade).

Os roteadores (que toda rede com conexão à internet possui) têm um papel simples: interligam duas redes e fazem o transporte de pacotes de informação de uma rede para outra, conforme sua necessidade. Mas muitos destes roteadores, além de identificar o destino do pacote e encaminhá-lo na direção certa, eles checam ainda: a direção dos pacotes; de onde veio e para onde vai (rede interna e Internet); endereço de origem e destino; tipo de pacote; portas de conexão; e flags do pacote.

Estes pontos de conexão da Internet com a rede interna podem receber uma série de regras para avaliar a informação corrente. São as listas de acesso que definem o que deve e o que não deve passar por este ponto de conexão. Elas são mais ou menos assim:

# regra	s/ñ	protocolo	origem	destino	opções

A opção "sim/não" eqüivale a "permitir a passagem do pacote/negar a passagem do pacote" e em "opções" definiremos os flags do pacote e portas de destino. Claro, esta é uma generalização das sintaxes mais comuns, o que nos permite ter uma idéia geral de como isto pode ser configurado. Consulte o manual do seu roteador para checar a sintaxe correta das listas de acesso. Algumas coisas podem estar fora de ordem, portanto, é melhor entender o conceito.

Vamos supor que nós queiramos impedir o acesso à nossa rede interna para que não seja possível uma conexão Telnet (pacotes TCPs da porta 23) em nossa rede (pelo menos as conexões vindas da Internet). Este filtro, logicamente, deveria ser colocado na interface de entrada dos pacotes externos na rede interna. Pela sintaxe do exemplo, ficaria assim:

# regra	s/ñ	protocolo	origem	destino	opções
1	ñ	TCP	qualquer	interno	porta 23

Ou seja, negar todos os pacotes TCP para a porta 23 vindos da Internet em direção à qualquer máquina da rede interna. Isto funciona muito bem. Mas vamos supor que, em alguma parte da internet, exista um funcionário da minha empresa que precise abrir sessões Telnet.

Então poderíamos fazer o seguinte: procurar saber *exatamente* de onde ele está tentando se conectar e permitir a entrada dos seus pacotes.

# regra	s/ñ	protocolo	origem	destino	opções
#1	ñ	TCP	qualquer	interno	porta 23
#2	s	TCP	255.01.1.8	interno	porta 23

Dessa forma, os pacotes vindo de 255.01.1.8 poderiam passar pelo roteador.

Se duas restrições não forem excludentes (o que NÃO é o caso do exemplo acima) elas podem fazer parte da mesma regra (i.e., terem o mesmo número). Mas, mais uma vez, é melhor consultar o manual do seu roteador para certificar-se de "como" ele dá prioridade às regras.

Outro detalhe: baseado nesta ordem, podemos definir uma política para a segurança da rede. A primeira opção seria liberar tudo e negar os serviços perigosos; a segunda seria negar tudo e liberar os serviços necessários. Sem sombra de dúvidas, a segunda é mais segura, entretanto, os funcionários da rede interna podem precisar acessar livremente à Internet como forma de trabalho, e a manutenção desta lista seria tão trabalhosa, visto a proliferação de programas Internet que, em pouco tempo estaríamos completamente perdidos em um emaranhado de regras sem sentido.

Então vamos pelo caminho mais simples, negar apenas os serviços perigosos. Não entraremos no mérito do "por quê" um serviço é considerado perigoso. Seria uma discussão sem fim. Então vamos nos prender a um pressuposto simples: os serviços disponibilizados em portas menores que 1024 são perigosos. Caso não saiba, estas portas são destinadas a serviços que precisam ser executados com privilégios de root , e o comprometimento de um serviço nesta porta estaria dando todos o poder nas mão de um hacker.

Obs: os serviços NFS, porta 2049 e X-window, porta 6000, apesar de não estarem rodando nas portas privilegiadas, são considerados perigosos demais e também serão bloqueados.

# regra	s/ñ	protocolo	origem	destino	opções
#1	ñ	UDP	qualquer	interno	porta 2029
#1	ñ	TCP	qualquer	interno	porta 6000

#1	ñ	UDP/TCP	qualquer	interno	porta <1024

A liberação de todas as portas altas é necessária porque os clientes dos usuários da rede interna (browsers, mails, telnets etc.) utilizam uma infinidade delas. Seria impossível (ou quase) liberar apenas as necessárias. Aparentemente estas regras resolveriam o problema, não fosse o seguinte fato: os servidores "geralmente" são executados em portas baixas (<1024), mas nada impede que um funcionário da rede execute um servidor de Telnet (que geralmente fica na porta 23), por exemplo, na porta 3456. Isto não é muito difícil de acontecer. Muita gente se utiliza deste método para escapar justamente das limitações impostas pelo firewall da sua empresa. Para eles é ótimo, pois podem navegar pela rede da empresa sem estar fisicamente lá, baixar arquivos, tirar relatórios, modificar configurações, muito bom, inclusive para os hackers.

Mas, como dissemos anteriormente, os roteadores verificam também um tal de "flags do pacote" (uma flag é uma sinalização entre os computadores intercomunicantes que identificam alguns estados em que o pacote se encontra. Analisaremos um estado bem interessante: o flag ACK! O ACK (de 'acknowledgment') identifica que este pacote é uma resposta a um pedido, ou seja, que pertence a uma conexão já existente. Os pedidos de conexão originais não possuem o flag ACK. Aí está a chave para o bloqueio da conexão nos servidores dos funcionários irresponsáveis.)

# regra	s/ñ	protocolo	origem	destino	opções
#1	s	IP	interno	qualquer	
#2	s	TCP	qualquer	interno	flag ACK

Desta forma, todos os pacotes com o flag ACK ativado poderiam entrar na rede interna, mas os outros (que muito provavelmente estariam fazendo uma tentativa de conexão) seriam bloqueados. É importante lembrar que estas regras se aplicam a cada interface do roteador, ou seja, a regra 1 seria aplicada na interface do roteador com a rede interna e, a número 2, na interface do roteador com a Internet. O sentido em que a informação passa pelo roteador deve ser levado (e muito) em consideração.

No ACK só podemos impedir a passagem do pacote TCP. O UDP não possui este flag.

O segundo motivo para a preocupação é o FTP. Isto é problemático para os firewalls. Os clientes de FTP (assim como todos os outros clientes) iniciam a comunicação com uma porta "local" alta, vamos supor, 2030, direcionada à porta 21 do servidor de FTP do qual se deseja baixar o arquivo. Até aí tudo bem, estamos dentro da última regra imposta pela filtragem de pacotes. O problema é o seguinte: a conexão FTP usa duas portas. Uma para controle e

outra para a transmissão dos arquivos. Você, ao se conectar no servidor de FTP (porta local 2030, porta remota 21), os computadores negociam, através das portas 2030/21, uma porta local para a transferência do arquivo, vamos supor 2055, que receberá o arquivo do servidor FTP originado na porta 20. Só que, quem faz o pedido de conexão na porta 2055 é o servidor FTP, portanto, o pacote tentará entrar na rede em uma porta alta sem o flag ACK acionado. Seria negado no ato.

O fator principal para esta desordem é a aleatoriedade com que os programas escolhem uma porta local para a transmissão do arquivo. O único jeito de se conseguir baixar os arquivos por FTP seria liberando novamente as portas altas, mas como vimos, esta não é uma atitude aceitável.

Há uma outra maneira bastante interessante de se contornar este problema, mas os recursos necessários para isto são um pouco mais difíceis de se encontrar: podemos filtrar pacotes baseados na porta de *origem* em que foram criados. A regra ficaria mais ou menos assim:

# regra	s/ñ	protocolo	Origem	destino	opções
#1	s	IP	Interno	qualquer	
#1	s	TCP	Qualquer	interno	flag ACK
#1	s	TCP	qualquer porta = 20	qualquer porta >1023	

É lógico que isto somente reduziria um pouco os problemas, até porque nada impede que o hacker utilize justamente a porta 20 para passar pelo firewall. O único inconveniente é que ele precisaria ter privilégios de root no sistema originador do ataque para poder usar estas portas de número baixo. Mas só se estiver usando algum sabor de Unix, porque se estiver em DOS não há maiores problemas (para o hacker). O mais importante nestas regras é o fato de que, com elas, o número de pessoas que vai desistir da invasão sem ao menos tentar entender o motivo da falha é de 90%. A grande maioria dos supostos hackers utiliza "receitinhas de bolo" para invasões. Se acontecer qualquer coisa diferente do planejado, ela ficará perdida. E o firewall é uma excelente arma contra estas pessoas.

O autor do livro "Firewall - Repelling the wily hacker", Steve Bellovin, propôs a utilização de um recurso no FTP chamado "FTP Passivo" que resolveria este problema da filtragem de pacotes. Com este método, os clientes e servidores de FTP trocariam normalmente as informações mas, no momento de transferir o arquivo, quem faria o pedido de envio pela porta 20 seria o cliente. Com isto os pacotes do arquivo chegariam com o flag ACK acionado (já que agora se trata de uma conexão feita, e não mais de uma tentativa de estabelecê-la), e assim poderíamos impedir a conexão com todas as portas altas da sua rede

Interna. Mas infelizmente nem todos os servidores e clientes de FTP já implementaram esta
característica (que, diga-se de passagem, já existe no protocolo FTP).

5.2 Os Filtros Inteligentes

Pensando nas dificuldades de configuração e falta de recursos dos roteadores para a implementação dos filtros de pacotes, muitos fabricantes criaram ferramentas para fazer este tipo de filtragem, desta vez baseada em um host (computador) específico para esta tarefa, localizado nos pontos de conexão da rede interna com a Internet.

Os chamados filtros inteligentes são aplicações executadas em, por exemplo, computadores ligados ao roteador e à rede interna. O tráfego de um lado para outro se dá (ou não) conforme as regras estabelecidas nas aplicações. Apesar desta solução requerer um equipamento extra, ela nos dá uma série de vantagens sobre os filtros baseados em roteador, principalmente no que diz respeito à monitoração de acesso.

Roteadores, quando possuem algum tipo de log, não guardam informações muito precisas sobre as tentativas de conexão na (ou da) rede interna, enquanto os filtros inteligentes possuem vários níveis de logs, nos quais é possível (e bastante recomendável) perceber os tipos de tentativa de acesso, e até definir certas ações caso um evento em especial relacionado à segurança aconteça.

Uma outra característica interessante dos filtros inteligentes é a tentativa de implementar um controle de pacotes UDP, guardando informações sobre eles e tentando "improvisar" o flag ACK. Montando-se uma tabela de pacotes UDP que passam, pode-se comparar os pacotes UDP que retornam e verificar se eles são uma resposta ou se são uma tentativa de novo contato.

Mais uma vez, voltamos a avisar: nem mesmo os filtros inteligentes são substitutos para a segurança dos computadores internos. Fica fácil visualizar a quantidade de problemas se seus computadores da rede interna não apresentarem nenhum nível de segurança e o firewall for comprometido.

6.0 Proxy

Neste sistema, temos um gateway (computador que faz uma ligação) entre o nosso host e o host que desejamos acessar. Esse gateway possui uma ligação com a rede externa e outra com a rede interna. Tudo o que passa de uma para outra deve, obrigatoriamente, passar pelo gateway. O fato de terem duas ligações lhe confere o nome de "Gateway de Base Dupla".

O seu funcionamento é simples, mas de funcionalidade trabalhosa: Vamos supor que eu queira acessar um servidor de FTP em uma rede que possua este gateway de base dupla. Primeiro, nós precisamos nos conectar ao Proxy (gateway), e dele nos conectar ao servidor FTP. Ao baixar o arquivo,

este não chegará direto até nós, e sim até o proxy. Após encerrada a transferência do arquivo até o proxy, transferimos-o, do proxy, até nosso computador. O conceito é relativamente simples, mas isto implica em alguns problemas.

O problema é em termos de capacidade de administração, já que todos os usuários da rede precisariam ter uma conta no servidor proxy, o que seria muito ruim, principalmente pelo fato mais do que sabido de que o mais grave em uma política de segurança é justamente a falta de cuidado dos próprios usuários do sistema. E se todos os usuários tiverem a mesma falta de cuidado em uma única máquina, em pouco tempo alguma conta seria violada e o servidor proxy estaria comprometido.

Há um outro tipo de proxy que facilita um pouco as coisas: os proxies de aplicação. Estes, ao invés de fazerem com o que usuário se conecte com o destino através do proxy, permitem ao usuário a conexão através de seu próprio computador para que transfira as informações diretamente, desde que passe nas regras estabelecidas no proxy de aplicação. Ao se conectar no proxy de aplicação, este oferece as opções que o usuário tem disponível na rede, nada mais, como acontece nos gateways simples de base dupla, em que o usuário tem acesso ao sistema para se conectar a máquinas internas da rede.

Entretanto, uma das principais vantagens destes proxies é a capacidade de "esconder" a verdadeira estrutura interna da rede. Vejamos como: ao tentarmos nos conectar em uma rede que possui este tipo de proxy, não precisamos saber nada além do número IP deste servidor. De acordo com o tipo de serviço pedido pela nossa conexão, o servidor de proxy a encaminha para o servidor que trata deste serviço, que retorna para o proxy, que nos responde o serviço.

Referências:

I- www.anti-hackers.com.br

II- www.hackers.com.br

III- www.cg.org.br/acoes/cartilha.htm

IV- www.securityfocus.com

V- Nota Técnica CBPF-NT-008/00- Segurança na Web

VI- Nota Técnica CBPF-NT-005/02 - Segurança no Portal UNIX