FACULDADES DE VALINHOS

POS GRADUAÇÃO EM GESTÃO DE TECNOLOGIA DA INFORMAÇÃO

MONOGRAFIA

SEGURANÇA DA INFORMAÇÃO EM REDES IP

Luis Fernando Brocanelli Braghetto
RA 0405009
– Valinhos – 2005 –

FACULDADES DE VALINHOS

POS GRADUAÇÃO EM GESTÃO DE TECNOLOGIA DA INFORMAÇÃO

Monografia realizada sobre "Segurança da Informação em Redes IP" como requisito parcial para a obtenção do Grau de Pós Graduado LatuSensu no Curso de Gestão em Tecnologia da Informação das Faculdades de Valinhos, sob orientação da Professora Laurimar G. Vendrusculo.

Luis Fernando Brocanelli Braghetto
RA 0405009
– Valinhos – 2005 –

"*Porque atacarão à mim se existem tantos alvos maiores por aí?*"
[MIC-05]

Índice

Índice de Figuras

Capitulo 1 – Definições e Objetivos

1.1 - Introdução

Os computadores estão cada vez mais presentes no nosso dia-a-dia, e utilizamos-o cada vez mais para assuntos variados: informação (através de notícias), trabalho, estudos (incluindo o *e-learning*), compartilhamento de informações e armazenamento de dados.

Desde o momento que os computadores puderam armazenar tantas informações de maneira fácil, e que podem recuperar com uma velocidade espantosa, ficamos mais e mais dependentes dele.

A maioria das aplicações de alguma maneira está ligada à Internet. A Internet pode ser considerado um marco importante que reavalia a maneira de utilizarmos o computador.

Porém, se por um lado a Internet é uma ferramenta mais indispensável a cada dia, e um número crescente de pessoas estão conectadas, por um tempo mais longo, isto leva inevitavelmente à algumas pessoas com más intenções a se aproveitarem da situação, e com o aumento dos números de aplicações e complexidade das redes, as pessoas estão mais e mais susceptíveis a ataques.

Porém a segurança abrange desde fechadura na porta da sala de computadores até o uso de ferramentas sofisticadas de criptografia, firewall, autenticação e autorização de acesso, etc.

Por isso a segurança da informação é tão importante.

Este trabalho tem como objetivo dar uma visão gerencial de quão perigoso é um ataque, e o quão desprecavidos somos no dia-a-dia, não competindo a este ainda saber sobre a índole das pessoas más a ação que elas praticam relacionadas à quebra de segurança.

1.2 – Definições

1.2.1 - Informação

Um dos termos que se tornam mais obsoletos hoje em dia é a chamada "segurança de rede". Este vem sendo rapidamente substituído pela "segurança da informação".

A rede por si só não agrega nada, é apenas um meio de se obter informação.

Mas o que é informação?

Na opinião deste estudo informação representa dados, cadastros, know how, conhecimentos, dados pessoais, senhas, chaves criptográficas, conversas, correspondências, hábitos.

Segundo o Aurélio [AUR 89] informação é dados acerca de alguém ou algo. Conhecimento, participação, instrução, direção.

Outra definição interessante é dado por [CAOR 05] Esclarecimento; explicação; instrução; aviso; comunicação; fornecimento de dados; notas; argumentos, etc. É todo e qualquer capital intelectual.

Sempre acho importante diferenciarmos dado e informação. Dado é apenas um conjunto de bytes que podem ou não fazer sentido para nós. Mas informação é algo importante. Informação é o conhecimento gerado a partir de dados.

1.2.1 – Quanto custa a informação?

Uma informação é algo valioso. Acho que podemos refletir sobre o assunto valor de uma informação, pensando rapidamente em alguns exemplos (talvez até utópicos) para facilitar o entendimento que quantificar um valor, seja ele em moeda ou outra medida, pode não ser uma tarefa fácil.

Primeiramente, o quanto custa a informação de como é produzida o refrigerante Coca-Cola? Muitos de nós podemos pensar que é um valor alto, que pode ser medido em dólares ou reais. Se essa fórmula (informação) cair em mãos erradas, o possível prejuízo anual da venda de refrigerantes desta companhia pode ser catastrófica.

Talvez um pouco mais difícil seja saber quanto vale a informação, por exemplo, do saldo bancário de correntistas de um banco? Se esta informação cair em mãos erradas, descobrir o nome de um milionário pode causar um sério risco a segurança pessoal dessa pessoa ou da sua família, pois estariam sujeitas a um assalto ou seqüestro.

E o quanto vale a informação de um relatório de vendas de uma empresa? O valor dela para o concorrente seria altíssimo, pois talvez esse concorrente possa usar estas informações para "roubar" parte do mercado da primeira empresa.

Outra reflexão sobre o valor da informação, talvez um pouco mais árduo de mensurar fosse pensar: Qual o valor de um e-mail com um diário de uma mulher, que está guardado em um serviço de "disco-virtual" de um provedor? Talvez para o administrador de rede o valor seja nenhum. Mas para o dono da informação (o diário), é um valor impossível de quantificar, pois não pode ser pago em moeda, e sim em sentimentos desta pessoa.

Portanto, uma empresa que guarda informações, seja quais forem, devem admitir de imediato que informação não tem preço, e que deve ser protegida à qualquer custo!

E se não parasse por aí, ainda temos que pensar que ao sofrer um ataque, muitas vezes teremos o serviço paralisado devido a estragos causados por hackers. Um site de comércio via Internet sem funcionar deixa de ganhar dinheiro.

Pior ainda é se fosse noticiado para seus clientes sobre o ataque feito na sua loja. Talvez a credibilidade dela fosse altamente afetada, pois os clientes tendem a perder a confiança em locais que já foram atacados. Por isso mesmo, muitas vezes estes ataques nunca serão conhecidos pelo público. Portanto, um ataque custa dinheiro. Informação custa dinheiro.

A única saída é proteger-se com várias técnicas de segurança da informação.

1.2.2 – O que é segurança?

Segundo [AUR 89], Segurança é o estado, qualidade ou condição de seguro. Condição básica daquele que se pode confiar. Segundo [AUR 89], seguro é algo livre de perigo. Livre de riscos, protegido acautelado, garantido.

De acordo com [KAT 77], segurança de dados pode ser definida como a proteção de dados contra a revelação acidental ou intencional a pessoas não autorizados e contra alterações não autorizadas.

Portanto, fácil dizer o que é segurança.

Segurança da informação mais aplicadamente à nosso assunto, vem da idéia de que informações ou conhecimentos estejam seguros e protegidos contra pessoas que não

precisam (ou não devem) ter acesso àquela informação. Podemos também pensar que segurança da informação é um conjunto de medidas que constitui basicamente de controles e de políticas de seguranças, tendo como objetivo a proteção das informações, controlando o risco de revelação ou alteração por pessoas não autorizadas. [BRA 05].

1.2.3 – O que é um hacker?

A principio quase todos temos uma idéia do que é um hacker. Um hacker na denominação formal é aquele que utiliza dos seus conhecimentos para invadir sistemas, não com o intuito de causar danos, mas como um auto-desafio às suas habilidades.

Um cracker é um hacker que tem um intuito em roubar a informação ou causar dano. Entretanto, nos dias de hoje, hacker é um termo utilizado para definir qualquer pessoa que causou um incidente de segurança [CAOR 05].

Capitulo 2 – Ataques

2.1 – Porque acontecem ataques?

De alguma maneira, ataques acontecem porque existe:

a) Uma motivação de um *hacker* (que aqui chamaremos de atacante);
b) Uma falha na estrutura de segurança do alvo atacado;
c) Um desconhecimento do valor da informação.

Podemos então imaginar isto como uma balança: Quanto mais vulnerável o sistema de segurança de uma empresa, menos motivação precisa o hacker, pois é mais fácil efetuar o ataque.

Se o sistema for muito bem protegido, entende-se que o hacker precisará de mais conhecimento, maior motivação e mais tempo para atacar.

Administradores de redes cada vez mais despreparados para suportar essa crescente onda de ataques e acúmulos de funções, gerando cada vez menos tempo para se preocupar com a segurança, adicionado à idéia de que a pessoa nunca será alvo de um ataque é mais um fator a ser levada em consideração.

Entretanto, com o aumento exponencial de ataques e vulnerabilidades dos sistemas, muitas vezes causados pelo esquema de redes excessivamente complexo, os profissionais de informática costumam saber que segurança é algo importante, porém os proprietários de empresas muitas vezes não acreditam que podem ser atacados. “Porque atacarão à mim se existem tantos alvos maiores por aí?” [MIC 05]. Embora as pequenas empresas costumem ter este pensamento, e o índice de ataques em grandes empresas sejam significativamente maiores em grandes corporações do que pequenos escritórios, a falha de segurança é cada vez mais comum em nossas vidas. Tenho certeza que já fomos atacados, ou que pelo menos conhecemos pessoas ou empresas que o foram.

E uma vez que você foi atacado, a recuperação costuma ser algo desesperador ou traumático para a maioria das pessoas que não estão preparadas para lidar com essas situações. Imagine o quanto custa a indisponibilidade da sua rede por um dia devido a esses ataques? Muitas vezes é mais fácil fazer uma analogia com a vida real! Imagine que você mora em um lugar tranqüilo, uma casa de classe média. Você deixaria sua casa aberta enquanto está viajando?

Resumidamente, ataques acontecem porque em algum momento alguém não tomou as precauções necessárias para evitá-lo. No decorrer deste estudo, poderemos entender o que podemos fazer para evitar.

2.1.2 – Tipos de atacantes

Como já dissemos, os atacantes são normalmente conhecidos como hackers. Entretanto, entre eles existe uma denominação específica para cada tipo de atacante [CAOR 05]:

- *Hackers*: são aqueles que utilizam seus conhecimentos para atacar algum computador ou serviço para provar a si mesmo que ele é capaz de acessar uma informação proibida. Normalmente ele não conta para ninguém sobre o ataque, não precisa de reconhecimento e não destrói nada;
- *Crackers*: são aqueles que utilizam seus conhecimentos para danificar informações, roubar e talvez ganhar dinheiro com isso;

- Script Kiddies: Essa modalidade está cada vez mais comum na Internet. Este tipo de atacante é normalmente leigo em informática e acha que é um hacker, porém utiliza-se de ferramentas prontas (programas ou scripts). Como esses scripts são grandemente difundidos, a proteção contra eles é na maioria das vezes bem fácil, pois eles não sabem muito. Normalmente eles contam para os amigos que são bons *hackers* e são facilmente pegos;
- *Insiders*: São Crackers que já trabalham dentro da empresa ao qual pretendem fazer o ataque;
- Pichadores: São Crackers que não que normalmente fazem *defeacement*[1] em sites. Normalmente eles gostam de fazer propaganda dos seus ataques;
- CyberTerroristas: São pichadores de sites com o intuito de exprimir sua opinião sobre aquela corporação. Por exemplo, um atacante que faz um *defeacement* em uma página do governo, colocando outra no lugar para demonstrar seu descontentamento com algum ato do mesmo, pode ser considerado um CyberTerrorista;
- WhiteHats: São consultores que sabem fazer ataques (*pentest*) com o intuito de testar a segurança dos seus sistemas ou sistemas de seus clientes;
- *BlackHats*: São os crackers.

2.2 – Identificando ataques

Muitas vezes, seu computador, rede ou servidor já foi atacado e você talvez nem saiba. E hoje em dia é comum que seu sistema operacional do servidor falhe e alguém já diz: "Atacaram o servidor!".

É uma questão de conhecimento para o profissional de segurança da informação saber se sua rede foi ou não foi atacada. É uma tarefa que normalmente consome tempo, além de exigir na maioria das vezes de ferramentas específicas para tal ação.

Entre as maneiras de detectar os ataques estão o IDS (*Intrusion Detection System*), análise de *log* ou análise forense. Estes serão detalhados posteriormente neste documento.

2.3 – Alvos dos Ataques

2.3.1 – Alvos mais Comuns

É difícil dizer o alvo mais comum! O que se pode dizer é que a cada dia existe mais invasões ocorrem no mundo, e a possibilidade do próximo ser você é grande.

Como vimos, o alvo tem a ver com a motivação do atacante. Um ex-funcionário recém demitido pode ser uma ameaça iminente de segurança em uma rede se houver alguma falha de segurança. Ele tem a motivação, o conhecimento técnico e conhece os procedimentos internos da empresa (incluindo topologia, equipamentos, etc.). Muitas vezes ele pode usar indevidamente senhas de ex-colegas de trabalho que ele aprendeu durante o tempo que ficou na empresa para facilitar a entrada.

[1] *Defeacement* é um tipo de ataque em um servidor web para alterar o conteúdo da página inicial.

Se a empresa, por outro lado, tiver uma boa estratégica de segurança da informação implantada, o atacante pode pensar duas vezes antes de fazer isso.

Este é apenas um dos exemplos mais clássicos. Todos os dias pessoas são atacadas, pois existe uma enorme diversidade de ataques possíveis.

Tenha certeza que se não houver uma proteção adequada, você poderá ser alvo até dos *Script Kiddies* com suas poderosas ferramentas, e que apesar de não ter conhecimento técnico apropriado ou motivação, ainda assim pode causar um desastre.

2.3.2 – Porque investimos pouco em segurança?

Raramente o diretor de uma instituição é técnico, então talvez não consiga compreender o impacto que esta falha de segurança representa no dia-a-dia dos negócios. Felizmente esse cenário está mudando rapidamente.

Como já explanado, eles podem pensar que não são alvos de ataques, e como muitas vezes não conseguimos quantificar o valor da informação, o investimento necessário para proteção, que muitas vezes chamamos erroneamente de custo (um gasto financeiro em tecnologia, políticas, treinamento e pessoal), é relativamente alto e que se bem implantado passe despercebido. Um investimento em segurança bem feito previne os ataques (e fica difícil dizer se o investimento foi bem feito, pois não há incidentes de segurança, ou se se investiu além do necessário). Neste ponto o investimento em segurança ou pode ser comparado como o atual para a mesma classe de segurança pessoal ou em favor do patrimônio. Trabalha se com a probabilidade pequena de se utilizar este recurso, mas também deste recurso ter sido calculado de forma aquém daquela esperada. Como é muito difícil mensurar isto, devemos sempre pecar pelo excesso, pois um incidente pode ter um custo muito maior.

Porém é exatamente na hora de uma tragédia que as pessoas tendem a dar valor a essa proteção.

Tenho convicção que muitas vezes o custo de *disaster-recovery*[2] pode ser menor que o valor necessário para implantar um esquema de segurança apropriado.

Talvez isso explique uma possível dificuldade em implantar segurança em um Banco via Internet ou cartão de crédito. Muitas pessoas deixam de utilizar seus cartões de crédito se precisarem digitar uma senha, que esquecemos regularmente, preferindo usar outro cartão ou talvez um cheque. Talvez seja mais barato manter uma estrutura tecnológica de *Bankline* e pagar pelas fraudes que ocorrem algumas vezes do que contratar dezenas de milhares de caixas para atender aos clientes, isto sem contar a falta de praticidade resultante a deslocar os clientes até a agência mais próxima.

Seguindo esse esquema, hoje em dia os bancos oferecem soluções inteligentes e seguras, rápidas e fáceis para tentar dificultar a vida dos hackers. Exemplos disso são o uso de requisição de datas de nascimento, cartões de segurança, teclados virtuais, etc.

Talvez essas atitudes venham de uma estatística (survey) que aproximadamente 500 companhias responderam: [CSI 02]

- 90% detectaram brechas de segurança;
- 80% perderam dinheiro com ataques;
- Somente 44% souberam quantificar suas perdas financeiras. Juntas foram cerca de 450 milhões de prejuízo;

[2] *disaster-recovery* é o procedimento emergencial técnico de reparo após um acidente (perda de informações, incendio, inundação, queima de equipamento, falha de segurança)

- 74% citaram que os ataques vieram da Internet, enquanto 33% citaram que houve ataques internos;
- 40% sofreram ataques de negação de serviço causando indisponibilidade do mesmo (DoS).

2.3.3 – Origem dos ataques

A origem dos ataques podem ser as mais diversas possíveis. Hoje em dia, mais e mais empresas estão sofrendo ataques de *insiders*[3] portanto precisamos tomar cuidado por todos os lados.

Na internet de modo geral, [CER 04] relata em uma estatística realizada no quarto trimestre do ano de 2004 sobre os atacantes. Nesta pesquisa levaram em consideração o *country-code*[4] da origem dos ataques detectados e reportados ao CERT.br. Como resultado desta, os Brasileiros continuam em primeira colocação sendo responsáveis por 29,7% dos ataques registrados, seguidos respectivamente por Estados Unidos com 21,3%, Coréia com 15,3% e China com 7,2%. Já em uma pesquisa semelhante realizada em 2005, os EUA passaram para primeira colocação com 27,6% e os Brasileiros com 26,1%. [CER 05]

Segundo a pesquisa realizada por [DEL 04] com 64 empresas – incluindo bancos, seguradoras e instituições financeiras entre as 250 maiores organizações do mundo –, o estudo concluiu que 83% das empresas foram alvos de ataques pelo menos uma vez ao longo dos doze meses anteriores a pesquisa. A origem desses ataques é variável. Para 21% das empresas, a fonte era externa; para 13% a origem era interna; e 49% das organizações registraram ataques tanto de proveniência externa quanto interna.

Estudos citados anteriormente reforçam a importância que a segurança deve ser tratada com seriedade e profissionalismo.

2.3.4 – Perfil dos Atacantes

A literatura nesta temática define diferentes perfis para os atacantes (*hackers)*. Antigamente, tinha-se em mente que o hacker tinha um perfil muito bem definido.

Segundo [LIC 03, CAOR 05] o hacker é um individuo obsessivo, de classe média, cor branca, sexo masculino, entre 12 e 28 anos, pouca habilidade social e possível histórico de abuso físico ou social.

Patrick Gray [PAT 04], um policial aposentado do FBI especializado em crime digital, diz que a lenda de que o hacker costumava ter um perfil do tipo homem, branco, entre 25 e 35 anos, sofreu abusos na infância, coisas assim, simplesmente não existe mais. Na verdade hoje em dia são pessoas que trabalham em equipe para o crime organizado. No início eram jovens testando as suas habilidades. Hoje estão mais sofisticados. São pessoas à procura de coisas de valor que possam ser roubadas e que muitas vezes são contratadas para fins específicos, como descobrir segredos dos concorrentes de uma empresa.

Em [HON 02], os psicólogos da HoneyNet Project, que veremos nos próximos capítulos, ouviram por algumas semanas, alguns supostos *hackers* através dos CHATs IRC[5]. Naquela oportunidade, eles estavam verificando um grupo denominado k1ddl3,

[3] *insiders* é o atacante que trabalha dentro da empresa.
[4] *country-code* é o resultado da resolução de DNS reversa (tais como usuario.com.br, o *country-code* é "br")
[5] IRC (*Internet Relay Chat*) é um protocolo. As aplicações IRC servem para que usuários possam se comunicar através de conversas (*chat*) instantâneas

liderado por um adolescente *CyberTerroristas* que se autodenominava D1ck. Aparentemente este individuo está motivado para atacar por causa da disputa pela Caxemira[6]. Aparentemente apenas o líder parece ter essa motivação, enquanto os seguidores se juntaram apenas pela causa comum do *hacking*. Aparentemente ele não possui supervisão dos pais ao usar o computador por tantas horas seguidas, pois normalmente fica conectado da 1 às 6 da madrugada por cinco dias seguidos. D1ck tem habilidades básicas em linguagem de programação C, e conhece os comandos Unix/Linux, embora use na maior parte scripts prontos. D1ck não pode fazer hacking dos sistemas Windows, pois tem pouco acesso ao sistema devido sua família tem pouco dinheiro e ele usa um computador antigo. D1ck também deixou escapar que mora em Karachi no Paquistão, tem 17 anos e tem cerca de 300 libras de peso (140 kg aproximadamente).

Max Kilger, psicólogo da HoneyNet, faz comentários sobre esse grupo: "*além do perfil dos indivíduos deste grupo, é importante entender suas ações no contexto maior da estrutura social da própria comunidade blackhats. A compreensão de como a comunidade determina as ações dos indivíduos é uma etapa importante para compreender as suas motivações, além de ser útil na criação dos métodos de orientar as suas ações na direção de um objeto [incidente] de segurança específico*". Podemos notar também que eles são mais "importantes" dentro da comunidade *underground* se ele é mais habilidoso e tirou mais hosts do ar por mais tempo. Nesta conversa uma mulher que usa m4ry no seu *nickname* sugere o desligamento do pessoal menos técnico ou menos qualificados.

Além disso, os criadores de vírus atualmente estão sendo cassados devido ao enorme prejuízo e incidentes de segurança causados pelos seus programas. Recentemente foram presos Farid Essebar, 18 anos, um russo que morava em Marrocos e usava o *nickname* "Diabl0" e a Turquia, as autoridades prenderam Atilla Ekici, ou "Coder", 21 anos. Juntos criaram o vírus Zotob e outros 20 vírus, incluindo o MyDoom-BG [VUN 05, TER 05]

Em uma rápida pesquisa na internet foi possível verificar um perfil de *hackers* que fazem fraudes em bancos. Todos relativamente parecidos. Incluindo o brasileiro citado em [FOL 05] com ataques que contabilizam prejuízos na ordem superior a 6 milhões de reais. Neste caso específico, o atacante era homem, 19 anos.

Tipicamente os outros atacantes tinham entre 17 e 30 anos, geralmente não trabalhavam sozinhos e contavam com ajuda das namoradas ou amigos. Eles usavam as técnicas de *phishing* ou *DNS Tampering* para o ataque e pegavam os clientes mais leigos que forneciam todos os dados que os hackers pediam em suas páginas falsas.

Mesmo assim, isso é apenas uma fração das comunidades que existem no mundo a fora, e nem todas pensam da mesma maneira. O objetivo disto é mostrar quantos *hackers* existem, e o quão inseguro estão os seus sistemas.

2.3.4 – Mitos de Segurança

Podemos agora ver alguns mitos da segurança e já podemos entender o porquê estes são mitos [LIC 03]:

- Isto nunca acontecerá conosco;
- Nunca fomos atacados, não precisamos de mais segurança;
- Já estamos seguros com um firewall;

[6] Caxemira é um Estado, na parte norte da Índia, a oeste do Nepal. Vivem lá entre 5 a 7 milhões, na sua maioria mulçumanos.

- Utilizamos os melhores (mais caros) sistemas de segurança, então eles devem ser seguros;
- Não podemos gastar com segurança agora, deixa assim mesmo;
- Ninguém vai descobrir essa "falhinha" de segurança;
- Tomamos todas as atitudes e precauções. Testes não são necessários;
- Nosso parceiro é confiável, pode liberar acesso para ele.

Um bom profissional da área deve ser capaz de identificar os pontos vulneráveis e nunca ter esses "mitos" em mente.

Possuir bons argumentos para derrubar estes mitos significa conhecer bem os riscos que a organização está correndo, levando em consideração toda a diversidade e complexidade do ambiente.

Capitulo 3 – Política de Segurança

3.1 – O que é?

Política de segurança é algo que a primeira vista pode ser algo complicado e que aparentemente não pode ser implantado em empresas pequenas. Entretanto esta abordagem deve ser o primeiro passo em busca de uma implantação da proteção ou segurança em uma organização.

Política de segurança não está unicamente ligado à tecnologia, não define apenas como deve ser configurado o *firewall* da empresa.

Política de segurança é um documento que contém a estratégia de segurança de uma organização, incluindo aspectos tecnológicos, humanos e culturais. A política de segurança atribui direitos e responsabilidades às pessoas que lidam com os recursos computacionais de uma instituição e com as informações neles armazenados. Este documento também define as atribuições de cada um em relação à segurança dos recursos com os quais trabalham. [NIC 05]

A política de segurança também deve prever o que pode ser feito na rede da instituição e o que será considerado inaceitável. Tudo o que descumprir a política de segurança pode ser considerado um incidente de segurança e nesta política deverá ser definida a penalidade àqueles que não o cumprirem.

3.1.1 – Visão geral sobre política de segurança

Criar a política de segurança deve ser feita com uma visão ampla, levando em consideração desde um simples incidente de segurança, até o que fazer em casos mais graves. Nesta fase de definição das políticas, devemos investir um tempo considerável para a sua elaboração.

O planejamento da política deve ter como diretriz a abordagem geral de todos os pontos, incluindo as ACLs[7] e ainda deve servir de guia para a implantação na vida real da segurança de uma organização.

Podemos dividir a tarefa em três itens: "Política", "Normas" e "Procedimentos". Estes itens podem ser desenvolvidos com base em padrões de referencias como o ISO 17799 e o BS 7799.

O BSI (British Standard Institute) foi responsável pela criação da norma BS 7799, considerada o mais completo padrão para o gerenciamento da segurança da informação no mundo. Com ela é possível implementar um sistema de gestão de segurança baseado em controles e práticas definidos por uma norma e práticas internacionais. Em dezembro de 2000, a Parte 1 da BS 7799 se tornou norma oficial da ISO sob o código ISO/IEC 17799. Em agosto de 2001 o Brasil adotou esta norma ISO como seu padrão, através da ABNT, sob o código NBR-ISO/IEC 17799. [MOD 05]

O objetivo da BS 7799 é fornecer recomendações para gestão da segurança da informação para uso dos responsáveis pela introdução, implementação ou manutenção da segurança em suas empresas. Também se destina a prover uma base comum para o desenvolvimento de normas de segurança organizacional e das práticas efetivas da gestão da segurança, e a prover confiança nos relacionamentos entre as organizações.

[7] ACL significa "*Access Control Lists*" que é uma lista de quais recursos podem ser acessados por um determinado usuário.

3.1.2 – Níveis da Política de Segurança

Sendo a política de segurança o guia de como deve ser a proteção contra incidentes em uma organização, esta deve ser ampla. Deverá contemplar os três grupos de pessoas em uma instituição:

- Genérico: Para o entendimento dos membros da corporação do que está sendo definido. Deve dar uma visão geral da segurança da organização.
- Usuário: Para que o usuário tenha consciência do seu papel para a manutenção da segurança da organização. Entende por usuário aqueles que utilizam os recursos tecnológicos da empresa.
- Técnico: Procedimento específico para os profissionais de TI, incluindo os profissionais de segurança. Neste nível, deve ser especificado em detalhes as definições das expectativas em relação a segurança, deve conter o *guideline* das implementações das regras do firewall, *ACLs*, etc.

3.2 – Introdução à Engenharia Social

A engenharia social é um dos meios mais utilizados para obtenção de informações sigilosas e importantes. Isso porque explora com muita sofisticação as "falhas de segurança dos humanos". As organizações investem em tecnologia de segurança (tais como IDS, Firewall, segurança física, etc.), mas não estão desguarnecidas de métodos eficientes que protegem seus funcionários das armadilhas de engenharia social.

Poderíamos definir engenharia social como qualquer método usado para exploração da confiança das pessoas com fins de obter de informações sigilosas, importantes ou valiosas de qual não poderiam obter legalmente. Para isso, o atacante pode se passar por outra pessoa, assumir outra personalidade, fingir que é um profissional de determinada área, etc. [INF 04].

Temos que ter em mente que engenharia social pode não ser apenas o ataque (roubo de informações), mas a engenharia social pode ser um passo inicial para aprendermos mais sobre a organização, a fim de facilitar um ataque futuro utilizando outros métodos. Engenharia social então, também está ligado à coletar informações sobre uma rede, organização ou pessoa, utilizando alguma técnica para obter dados que são confidenciais, ou que possam facilitar um ataque "tradicional" no futuro.

Para procedermos com um ataque inicial de engenharia social, poderemos ter mais sucesso se pudermos pesquisar informações sobre o alvo, conhecer as regras e procedimentos internos, acesso à algum recurso interno (fingindo ser fornecedor, parceiro, autoridade policial, etc.) ou coletar informações que a vitima possa achar que é inofensiva.

Os tipos mais comuns de ataque de engenharia social são: [CAOR 05]

- Simplesmente pedir a informação: As pessoas podem acabar te dando a informação através de um telefonema para tirar dúvidas, ou apenas por curiosidade;
- Usando simpatia, culpa e a intimidação: Podemos ser simpáticos com a vítima para facilitar o vazamento de informações;

- Entrando nas instalações e fazendo se passar por fornecedor, parceiro, cliente, etc. e olhar sobre o movimento de pessoas, localização, equipamentos utilizados, etc.;
- Investigando lixo, sucata e papeis para reciclagem;
- Ataque a empregados iniciantes, pois os empregados novatos tendem a contar o que aprenderam por diversos motivos, como por exemplo para se sentir útil, para demonstrar que agregou conhecimentos, por falta de conhecimento das políticas de funcionários;
- Alteração de cargo: podemos simplesmente entrar na organização, pedir informação à outros funcionários e dizer que somos "novos na empresa". Isto pode ser uma maneira fácil de explicar a falta de um crachá de identificação, uniforme ou ser desconhecido na organização;
- Ataque a empregados ou ex-empregados: Empregados insatisfeitos ou ex-funcionários podem dar muitas informações, pois em suas opiniões eles "não tem o que perder";
- Serviços públicos: Servidores de e-mail, DNS (INFO, TXT, SOA, etc.), website da empresa.

3.3 – Melhor segurança utilizando políticas de segurança

Agora, podemos ter idéia de como uma política de segurança bem feita pode evitar os ataques de engenharia social.

Entretanto, não basta apenas escrevermos um documento e achar que apenas isto pode resolver o problema. A política deve ter em mente os seguintes elementos [HUR 99, CAOR 05]:

Estratégia:

Criatividade quanto as definições da política e do plano de defesa contra intrusões, além de adaptabilidade a mudanças no ambiente, e até mesmo os incidentes não previstos. Deve-se fazer uma avaliação dos aspectos de produtividade dos usuários, de forma que medidas de segurança não influenciem negativamente no andamento dos negócios da organização.

Vigilância:

Todos devem entender a importância da segurança da informação na organização como um todo. Todos devem estar empenhados em agir de maneira consciente para evitar um possível problema. Quanto ao aspecto técnico, vigilância significa o regular e consistente monitoramento da rede e dos sistemas, incluindo alarmes, logs, alterações do perfil normal do dia-a-dia, etc.

Atitude

Atitude é a postura e conduta quanto à segurança. Estas atitudes podem ser obtidas por meio de treinamento e orientação adequada de toda a comunidade envolvida direta ou indiretamente com sistemas de informação. A segurança deve ser incorporado ao cotidiano da organização

Tecnologia:

A solução tecnologia (hardware e software) não deve ser o único ato de confiança. Não basta apenas comprar o produto de um fabricante famoso, pois isto causa uma falsa sensação de segurança. Nisto, o mais importante não é ter um produto,

mas sim uma política de segurança que abraça múltiplas práticas e tecnologias de segurança.

Esta solução deve ser flexível, adaptativa e completa, a fim de suprir os requerimentos estratégicos da organização.

De acordo com a ISO17799, uma política de segurança deve conter:

- Definição de segurança da informação, resumo das metas, do escopo e a importância da segurança para a organização, enfatizando seu papel estratégico;
- Declaração do comprometimento do corpo executivo;
- Breve explanação das políticas, princípios, padrões e requisitos de conformidade de segurança no contexto específico da organização, por exemplo:
    - Conformidade com a legislação;
    - Requisitos na educação e treinamento em segurança;
    - Gerenciamento de continuidade do negócio;
    - Conseqüências da violação na política de segurança;
    - Definição das responsabilidades gerais e específicas na gestão da segurança de informação, incluindo registro de incidentes;
    - Referências que possam apoiar a política.

3.3.1 – Exemplo de itens cobertos por uma boa política de segurança

Discutimos que a política deve conter a visão da organização na questão da segurança. Mas o que devemos cobrir em uma boa política de segurança? [LIC 03, JUL 04]

Estrutura de Política de Segurança

1. Introdução
1.1. Política de Segurança
    1.1.1. Informações Gerais
    1.1.2. Objetivos
1.2. Estrutura de Responsabilidade Organizacional
    1.2.1 Serviços de Informações Corporativos
    1.2.2 Serviços de Informações de Unidades de Negócio
    1.2.3 Organizações Internacionais
1.3. Padrões de Segurança
    1.3.1 Confidencialidade
    1.3.2 Integridade
    1.3.3 Autorização
    1.3.4 Acesso
    1.3.5 Uso Apropriado
    1.3.6 Privacidade dos Funcionários

2. Descrição do Sistema
2.1 Papel do Sistema
    2.1.1 Tipo de informação manipulada pelo Sistema
    2.1.2 Tipos de usuários (administração, usuário normal, etc.)

2.1.3 Número de usuários
2.1.4 Classificação dos dados
2.1.5 Quantidade de dados (número de bytes)
2.1.6 Configuração do Sistema
2.1.6.1 Número de terminais
2.1.6.2 Número de console de controle
2.1.6.3 Número e tipos de terminais (inteligente, burro, de impressão, etc.)
2.1.6.4 Arranjos para carregamento de mídia
2.1.6.5 Software (sistema operacional e versão)
2.1.6.6 Interconexões (LAN e WAN)

3. Requisitos de segurança e medidas
3.1 Ameaças à confidencialidade, integridade e disponibilidade de dados
3.2 Natureza dos recursos de possíveis atacantes e atratividade do sistema e dos dados
com alvo
3.3 Impactos do comprometimento acidental dos dados

4. Plano de resposta a incidentes de segurança
4.1 Preparação e planejamento da resposta a incidentes
4.2 Notificação e pontos de contato
4.3 Identificação de um incidente
4.4 Resposta a um incidente
4.5 Conseqüências de um incidente
4.6 Forense computacional e implicações legais
4.7 Contatos de relações públicas
4.8 Passos-chave
4.8.1 Contenção
4.8.2 Erradicação
4.8.3 Recuperação
4.8.4 Acompanhamento
4.8.5 Conseqüências/Lições apreendidas
4.9 Responsabilidades

5. Contatos e outros recursos
6. Referências

3.4 – Política de Privacidade

Muitos de nós já fizemos algum tipo de cadastro em algum website na Internet.

Porém, você já pensou o que é feito com essas informações?

Entretanto, os sites coletam muito mais informação do que você preenche. Durante um simples acesso à um site, o servidor já está coletando informações, tais como endereço IP, tipo e versão do navegador, quais foram as páginas acessadas, etc. Tudo isso pode ajudar à traçar o perfil dos dos internautas.

Sites de comércio eletrônico já usam essa tecnologia. Eles verificam os produtos que o usuário mais visita, produtos procurados no site em geral, oferecendo uma

publicidade dirigida. Se por um lado isto tem um motivo aparentemente “nobre” de vender mais, tudo que pode ser usado para o bem será usado para o mal pelos atacantes.

A política de privacidade dos websites, deve especificar o que estão coletando, para quem repassarão (terceiros) e qual o objetivo a que se destina.

Infelizmente poucas pessoas lêem estes documentos.

Hoje em dia é comum que pessoas nem saibam que estão sendo vigiadas enquanto navegam na Internet.

Capítulo 4 – Tipos de ataques

Depois das recomendações citadas anteriormente, é sempre importante que saibamos como atacar para nos prevenir. No mínimo, devemos saber um pouco mais sobre os ataques, porque sem essas informações, muitas vezes nem sabemos como evitar os problemas de segurança.

Conforme dito, segurança não é simples, e requer, além de políticas de segurança suportadas pela diretoria da organização, uma proteção tecnológica implantada por profissionais experientes.

Aqui poderemos ver alguns tipos de ataques mais comuns, e entender como eles podem comprometer a segurança da organização.

Fica-se registrado, que muitas pessoas seguem a literatura (*play-by-the-book*), e o ataque é nada mais do que fazer algo inesperado para conseguirmos um resultado esperado. Para atacarmos precisamos ter a mente aberta a possibilidades e a que qualquer coisa pode ser um passo em direção ao sucesso do ataque.

Por fim, entenderemos o principio básico de um ataque: Um ataque normalmente é complicado para fazer algo simples e abrir um espaço onde poderemos efetivamente roubar a informação de maneira "fácil".

Durante este capítulo, teremos em mente que esta é a visão do atacante para nós (*WhiteHats*) podermos aprender a melhorar nossa segurança. É importante ter um bom conhecimento de redes (modelo OSI) para acompanhar em detalhes o que poderemos fazer.

4.1 – Scanning

Scanning é o ato de fazer algum tipo de pesquisa para que possamos conhecer algo mais do alvo. Saber uma informação adicional pode facilitar imensamente o ataque, pois um ataque *"no escuro"* tem menor probabilidade de ser bem sucesso.

4.1.1 – Network Scanning

Network Scanning é um procedimento para identificar os *hosts* ativos em uma rede, para o propósito de atacar. O objetivo, dado um determinado segmento de rede, é que possamos descobrir onde temos equipamentos de redes ativos que podem ser atacados. Várias técnicas são possíveis, tais como o *ping sweep*[8] ou envio de pacotes aleatórios para um endereço.

Muitos técnicos em segurança acreditam que o bloqueio do ping é o suficiente para evitar esse scanning, porém essas estão erradas. Podemos enviar um pacote TCP qualquer para um conjunto de hosts. Se voltar um pacote TCP com a *flag RST*[9] ativa por exemplo, confirma-se que há um equipamento de rede ativo.

4.1.2 – Port Scanning

Uma vez que sabemos que um host está conectado a Internet, precisamos descobrir qual aplicação está rodando nela. Uma das maneiras mais fáceis é o *Port Scanning*.

[8] ping sweep é o ato de enviar um pacote de ping (ICMP Echo-Request) e aguardar uma confirmação.
[9] *RST* (*reset*) é um bit da parte do cabeçalho do protocolo de transporte TCP, e normalmente indica um erro na transmissão.

Todo serviço do TCP/IP depende de portas para escutar ou enviar dados. Essas portas podem ser usadas em TCP ou UDP.

O *port scanning* é o método que faz uma varredura em um endereço IP para descobrir as portas TCP e UDP que estão em estado "LISTEN". Se uma porta está em estado "LISTEN", provavelmente há uma aplicação servidor nela rodando (tais como servidor HTTP, E-mail, FTP, DNS, etc.).

Os serviços importantes, tais como HTTP, POP3, SMTP, etc., possuem portas conhecidas (80, 110, 25, respectivamente), e é de nosso interesse que todos saibam para que possam acessar essas aplicações.

Entretanto, é possível que exista alguma aplicação na máquina que não esteja em uso, porém instalada, ou que esteja sem atualização, etc. Os atacantes procuram normalmente essas aplicações para explorar vulnerabilidades. Este fato é descoberto através de um processo de *portscanning*.

O mais famoso aplicativo de *portscanning* que existe é o nmap.

Os firewalls e aplicativos normalmente querem esconder-se destes aplicativos de *scanning*, porém eles normalmente possuem vários métodos para descobrir se uma porta está em estado "*LISTEN*". Os mais usados pelo nmap são:

- TCP Connect(): O método mais básico de tentar descobrir as portas abertas é simplesmente conectando-se nelas. O uso da API *connect()* do TCP espera a conexão se estabelecer e se isto ocorrer, a porta está aberta. Neste tipo de conexão, os três passos do *hand-shake* são feitos (SYN, SYN+ACK, ACK).
- TCP SYN (*half-open connections*): Este método é semelhante ao anterior, mas não abre uma conexão completa, fazendo que alguns IDSs mais simples não possam detectá-los. Neste tipo de conexão só um passos é feito (SYN). Se ele receber o SYN+ACK em retorno, significa que a porta está aberta. Se ele receber um TCP RST em retorno, significa que a porta está fechada.
- UDP: Este método envia um pacote de 0 bytes de payload para uma porta UDP. Se não receber resposta, a porta remota está aberta (pois UDP não tem confirmação de entrega). Se receber um UDP RST em retorno, significa que a porta está fechada.
- FIN Scan: Quando uma conexão TCP é fechada, o host deve enviar um pacote do tipo TCP FIN. O host remoto deveria responder com ACK. Entretanto, para que encerrar uma conexão que não existe? Claro que é para saber se a porta está em LISTEN sem o firewall descobrir. O atacante envia um FIN. Se receber um RST significa que a porta está fechada, porém se não receber uma resposta, significa que o host remoto está com a porta em estado LISTEN. Os Windows não são susceptíveis a esse tipo de scan (não funciona).
- Xmas: Portas fechadas enviam um RST como resposta a pacotes FIN em portas fechadas. Se enviarmos um pacote TCP FIN+URG+PUSH podemos criar um pacote não detectável pelo firewall e ainda assim se receber um RST significa que a porta está fechada, e se não receber uma resposta, significa que o host remoto está com a porta em estado LISTEN. Os Windows não são susceptíveis a esse tipo de scan (não funciona).
- NULL Scan: Semelhante ao FIN SCAN, entretanto nenhum flag do TCP é habilitado (o que seria impossível em um pacote normal de inicio de conexão). Se receber um RST significa que a porta está fechada, porém

se não receber uma resposta, significa que o host remoto está com a porta em estado LISTEN. Os Windows não são susceptíveis a esse tipo de scan (não funciona)

- Random SCAN: Na verdade é uma técnica que é usada em conjunto com outra das técnicas citadas acima. Ao fazer o *scanning* todas as portas de modo seqüencial (1, 2, 3, 4, 5, 6, etc.), os IDSs mais inteligentes podem detectar essa atividade anormal. Por isso o *scanner* pode fazer o *scan* de forma um pouco mais aleatória, evitando ser descoberto pelo firewall.

4.1.2.1 - Como evitar?

Evitar o acesso à portas indevidas, é feito através de firewalls, que bloqueiam ou permitem o acesso à determinadas portas pré-estabelecidas.

Porém o aplicativo de *PortScanning* passa por uma aplicação "teoricamente" legitima, tentando abrir conexões de maneira oficial *(Three-way-handshaking*) ou através de outros tipos de *scanning* (TCP Syn Scan, Stealth FIN, Xmas Tree, Null Scan, etc.).

Uma maneira de evitar os *SynScan, Stealth FIN, Xmas*, etc., é utilizando regras de firewall com filtro de estados que podem ficar verificando as flags do TCP. O TCPConnect só pode ser evitado com um filtro limitando o número de conexões abertas por segundo ou minuto (pois um nmap chega a 1700 conexões TCP-half-open para um scan).

Os IDSs (*Intrusion Detection Systems*) podem detectar esses scans que são uma anomalia do tráfego normal para futura análise. Alguns Firewalls com IDS integrados podem detectar e bloquear uma parte do ataque.

4.1.3 – Firewalking

Firewalking pode ser definido como "uma técnica que permite a obtenção de informações sobre uma rede remota protegida por um Firewall. Obtém-se informações sobre as regras de filtragem dos Firewalls e um mapa da topologia da rede. Esta técnica permite que pacotes passem por portas em um gateway, além de determinar se um pacote com várias informações de controle pode passar por um gateway." [FIR 98]

Firewalking usa um analisador de pacotes do tipo *traceroute-ip* para determinar quais são e quais não são os pacotes liberados pelo firewall. Esta técnica é usada para verificar quais portas estão abertas ou "com bypass" no gateway, além de prover um mapa dos roteadores atrás de um firewall.

Isso pode ser feito com um traceroute, ou com uma ferramenta mais elaborada para dificultar a detecção de rede.

4.1.3.1 - A ferramenta firewalk

Firewalk é uma ferramenta de auditoria de rede que emprega as técnicas semelhantes à traceroute. Ele executa um traceroute utilizando as técnicas padrões, porém firewalk pode fazer scan enviando pacotes TCP, UDP ou ICMP com um TTL maior que o do gateway. Se o gateway permitir o tráfego, o pacote será injetado na rede local e irá expirar e enviar uma mensagem ICMP TTL *Exceeded in transit*.

Porém, enviando pacotes com TTLs específicos, em ordem predefinida, podemos ir gravando as respostas e determinar os filtros ativos no firewall.

4.1.3.2 – Como evitar?

Evitar isso é uma tarefa que envolve uma violação da RFC. Simplesmente o firewall deverá impedir que pacotes *ICMP TTL Exceeded in transit* saiam da rede.

Isso pode comprometer algumas tarefas ou alguns detalhes no funcionamento da sua rede local, porém é um preço que se deve verificar se podemos pagar em função da segurança adicional.

4.1.4 – DNS Querying, WINS, Reverse DNS

Os servidores de DNS são aplicações legítimas que transformam endereços lógicos (*FQDN*[10]) em endereços IPs, e endereços IPs em endereços lógicos (*FQDN*). No DNS podemos encontrar algumas informações sobre a rede.

Na questão de descobrir o que um servidor faz, podemos consultar o DNS de forma reversa. Por exemplo, queremos saber o nome da máquina 150.20.20.5 e se ela nos retornar um mail.consultores.com.br, podemos inferir que ela é um servidor de e-mail. Além disso, poderíamos ter encontrado algo como firewall.consultores.com.br que nos diz quem seria o firewall. Esta técnica de conseguir informações por DNS não é intrusiva e não pode ser detectada por firewall.

Além disso, os DNS provêem alguns tipos de registros adicionais, tais como SOA, TXT (com um texto como se fosse um comentário do administrador). Conseguiremos as informações do e-mail do responsável, que eventualmente pode sofrer *phishing*, os endereços dos servidores de DNS, entre outras informações.
O WINS é um protocolo criado para usar-se com redes Microsoft, e integram o serviço de DNS com nomes NetBIOS. Podemos com um pouco de sorte achar um registro do tipo *diretor.rede.compania.com.br* e este possuir um IP. Temos eventualmente como fazer uma idéia da topologia de rede da organização alvo.

4.1.5 – WHOIS

WHOIS é uma ferramenta normalmente utilizada com fins lícitos para procurarmos informações sobre a propriedade de algo na Internet (como um domínio, um *Autonomous System* (AS) ou classe de endereços IPs). Hoje os servidores de WHOIS disponibilizam várias outras informações. É bem comum acharmos formas de contato "mais diretas" ao administrador de uma rede.

Qualquer informação adicional pode ser usada em um ataque de engenharia social.

4.1.6 – Finger

Finger é um serviço que diz quais pessoas estão conectadas à um equipamento ou servidor na Internet. Podemos eventualmente descobrir informações de nomes de usuários do administrador ou de algum usuário para ser utilizado em um processo de *password guessing*[11].

[10] FQDN (*Fully Qualified Domain Name*) é o nome completo de um host no espaço de DNS da Internet, incluíndo nome do servidor, da rede e do dominio. Exemplo "solimoes.dcc.unicamp.br" (sem aspas).
[11] Verificação de senhas fáceis.

4.1.7 – Google e Sites da Companhia

Muitas empresas criam sites com um grande conteúdo. Esse conteúdo pode ser utilizado para vários fins, incluindo os ilícitos. Podemos pesquisar sobre a vida de um administrador para facilitar a descoberta das suas senhas. Podemos pesquisar o nome de um diretor de alto escalão para fazermos um ataque de engenharia social.

E além disso, muitas empresas criam "*backdoors*" no próprio site. Criam páginas que ao acessar podem eventualmente levar à uma "área de administração". Como as ferramentas de busca visitam todos os links, seria possível encontrar o endereço FQDN da página "*backdoor*" e causar algum tipo de estrago.

4.1.8 – Version Determination

Os softwares normalmente exibem a versão que estão executando no momento para fins informativos e/ou de compatibilidade. O que a primeira vista pode ser inofensivo, pode trazer um sério prejuízo.

Se um hacker descobre, por exemplo, a versão do software de SSH que está ativo no servidor, este poderá ir até o site do fabricante e saber quais foram os bugs de segurança que foram descobertos nesta versão até agora. Isto pode ser usado para um ataque do tipo *exploit* (será visto mais adiante neste capítulo).

Vejamos um simples teste:

```
telnet mail.dominio.com 110
+OK Microsoft Exchange 2000 POP3 server version 6.0.6603.0 (mail.dominio.com) ready.
```

Podemos facilmente entender qual a versão. Basta pesquisarmos um pouco na Internet para descobrir formas de atacar este servidor.

4.1.9 – OS Version Determination

Ferramentas hoje em dia podem detectar a versão do sistema operacional, e diferenciar entre Windows, Linux, Solaris e outros.

Isto é possível porque cada sistema operacional, apesar de usar o mesmo padrão para a pilha de protocolo TCP/IP, possui uma implementação diferente. Cada leve interpretação diferente do RFC, pode dar uma dica de qual é o padrão ou assinatura das respostas.

Baseado nessas assinaturas, os aplicativos contam com uma grande base de dados, que permite que seja detectada com grande precisão a versão do sistema operacional. Complemento a essa técnica, os mesmos já se utilizam de técnicas de *Software Version Determination* e conseguem resumir em um relatório bastante simples [INS 05]:

(deixado intencionalmente em branco)

```
# nmap -A -T4 -F www.insecure.org
Starting nmap 3.40PVT16 ( http://www.insecure.org/nmap/ ) at 2003-09-06 19:49 PDT
Interesting ports on www.insecure.org (205.217.153.53):
(The 1206 ports scanned but not shown below are in state: filtered)
PORT    STATE  SERVICE VERSION
22/tcp  open   ssh     OpenSSH 3.1p1 (protocol 1.99)
25/tcp  open   smtp    Qmail smtpd
53/tcp  open   domain  ISC Bind 9.2.1
80/tcp open  http   Apache httpd 2.0.39 ((Unix) mod_perl/1.99_07-dev Perl/v5.6.1)
113/tcp closed auth
Device type: general purpose
Running: Linux 2.4.X|2.5.X
OS details: Linux Kernel 2.4.0 - 2.5.20
Uptime 108.307 days (since Wed May 21 12:27:44 2003)
```

Figura 1: Relatório oferecido pela ferramenta NMAP

Algo interessante é saber o "*uptime*" de um servidor, ou seja, o tempo desde o último reset. Essa informação é descoberta baseada na análise dos números de reconhecimento da pilha TCP (TCP SYN, SYN+ACK).

Isto é especialmente útil para um atacante que esteja rastreando 500.000 máquinas para descobrir qual sistema roda e quais portas estão abertas. Se houver uma versão de software sendo executada com uma versão antiga e sem *patches*, poderemos utilizar um ataque específico para a versão de software, de modo que sabemos que poderemos entrar através de ferramentas de *exploits*.

4.1.10 – Scanning de vulnerabilidades

Existem aplicativos que procuram vulnerabilidades específicas para cada serviço em uma rede. Eles podem ser utilizados em auditorias, mas também com intuito de coletar informações para ataques. O mapeamento inicial é feito por *port scanning* e logo após é feito um *version determination*. Alguns riscos que estes podem checar [LIC 03]:

- Compartilhamento de arquivos não protegidos por senha;
- Configuração incorreta;
- Software desatualizado;
- Exploits possíveis;
- Falhas no nível de rede do protocolo;
- Configurações de roteadores potencialmente perigosos;
- Checagem de cavalos-de-tróia, tais como BackOrifice ou Netbus;
- Checagem de senhas fáceis (*password guessing*);
- SNMP;
- Possibilidade de DoS.

4.2 – Política de Segurança Ruim

Ter uma política de segurança ruim pode ser uma margem para um incidente. Se pensarmos um pouco mais, podemos conseguir uma cópia da política de segurança da companhia alvo e estudá-la. Possivelmente se alguma condição não estiver prevista, podemos utilizá-la para um ataque ou para conhecermos quais ataques são inúteis, ou ainda sabermos como é configurado um firewall.

4.2.1 – Senhas fáceis

A política de segurança deve estabelecer uma complexidade mínima para as senhas. Se estas não forem respeitadas, podemos utilizá-las para ataque. Ter algum tipo de informação sobre o dono da senha pode diminuir o escopo de busca.

Recentemente, um teste real feito por este autor em uma base de dados com 2000 contas (previamente despejadas ou colocadas em um arquivo com o *login* e senha em MD5) pôde demonstrar o quão fraca são as senhas utilizadas por usuários simples que não possuem instrução de como proceder.

O software utilizado para o *password guessing* foi o "John The Ripper" (http://www.openwall.com/john/), que utiliza basicamente um dicionário de cerca de 1000 senhas mais comuns (tais como 123, 123456, abcdef, asdfg, etc.) e um algoritmo simples de busca exaustiva. Pude perceber que em menos de 5 minutos, o simples PC usado neste experimento havia descoberto cerca de 60% das senhas. Após 48 horas de execução mais de 98% das senhas haviam sido descobertas.

4.2.1.1 – Como evitar?

Facilmente podemos criar uma senha fácil de ser lembrada e difícil de ser criada.

- Elaborar sempre uma senha que contenha pelo menos 8 caracteres, compostos de letras números e símbolos; como por exemplo 12k97$xD;
- Nunca use senhas como seu nome, sobrenome, telefone, números de documentos, placas de carros, datas que possam ser relacionadas com você (pai, mãe, irmão, casamento, etc.) ou palavras constantes em dicionários;
- Utilizar uma senha diferente para cada serviço;
- Nunca usar senhas clichês, tais como 1234, 123456, 654321, admin, !@#$%^, asdfgh, zxcvb, etc.;
- Nunca anote sua senha em um papel e cole no seu monitor;
- Alterar a senha com freqüência.

Entretanto entendemos que para um usuário final, muitas dessas coisas são difíceis de serem lembradas.

Uma dica para elaborar senhas seria combinar itens. Por exemplo, se seu nome é João e a placa do seu carro é ASX8821, você pode combinar e criar uma senha "jo8821@o" ou "oaASXoj1288", etc. Ou ainda decorar uma frase "Pude fazer bons negócios em setembro de 2005" que faria uma senha "Pfbnesd2". Podemos ainda substituir o *s* por $ para criar uma senha ainda mais robusta "Pfbne$d2".

4.2.2 - Senhas públicas

Muitas vezes pessoas não se atentam aos detalhes e utilizam as senhas padrões dos programas. Não é incomum, estes usuários usarem a senha do arquivo de configuração padrão para formar uma VPN.

Esses arquivos de configuração padrões de fábrica devem ser evitados pois não só você tem acesso, como todo mundo pode descobrir essa senha

4.2.3 - Vazamento de Informações

Como vimos rapidamente, uma má política de segurança pode causar a revelação de informações ao atacante. Como já vimos, algumas maneiras de conseguir informação seria:

- Simplesmente pedir a informação;
- Usando simpatia, culpa e a intimidação;
- Entrando fisicamente nas instalações;
- Investigando lixo, sucata e papeis para reciclagem;
- Ataque a empregados iniciantes;
- Alteração de cargo;
- Ataque a empregados ou ex-empregados.

4.3 – Ataques de Rede

4.3.1 – Sniffing ou Eavesdropping

Ferramentas são fornecidas pelos fabricantes para auxiliar na solução de problemas de redes, para ver o tráfego e analisar suas características. Estes softwares podem facilmente capturar informações valiosas na rede, tais como as senhas em protocolos não criptografados como POP, FTP, Telnet, SMTP, HTTP, etc.

O SNMP é altamente perigoso nestes casos. O SNMP é um protocolo utilizado para gerenciar equipamentos (desde coletar estatísticas sobre o seu funcionamento, até alterar alguma configuração do mesmo) e a senha é enviada em branco na maioria das vezes.

O uso de switches ou roteadores que segmentam a rede já dificultam o *insider* a efetuar o ataque, entretanto, se um *sniffer* for instalado em um gateway da rede, tudo o que não tiver cifrado está comprometido.

Além disso, se o atacante conseguir a senha do switch, ele pode usar o espelhamento de portas para coletar todo o tráfego de rede sem sair do seu PC.

Exemplos destes softwares são o Ethereal (Windows e Linux), snoop (Solaris), tcpdump (linux), Network Analyzer (Windows), entre muitos outros.

Normalmente utilizada por um *Insider* ou associada a um ataque Físico.

4.3.2 – 802.1q e Trunking

Em switches é possível criar "redes virtuais" (*VLAN* ou *Virtual Local Area Networks*). Isto serve para segmentar a rede, e eventualmente separar o tráfego da rede "externa" (internet) da rede corporativa interna.

Entretanto o IEEE criou o protocolo 802.1q para criar *Trunking*, e estender uma VLAN à vários switches. Desta maneira uma VLAN lógica de um switch pode existir em outros switches, e para conseguir isso é adicionado um cabeçalho ao quadro da Ethernet a fim de identificar de qual VLAN o pacote pertence.

Existe então a possibilidade de injetar um pacote na rede com esta identificação da VLAN adulterada, de forma que alguns quadros possam ser inseridos em uma VLAN com destino à uma VLAN diferente. [LIC 03]

4.3.3 – *IP Spoofing*

IP Spoofing é uma técnica de subversão de sistemas de informática que consiste em mascarar (*spoof*) pacotes IP com endereços remetentes falsificados. [WIK 05]

Devido às características do protocolo IP, o reencaminhamento de pacotes é feito com base numa premissa muito simples: o pacote deverá ir para o destinatário, sem que haja verificação do remetente — o roteador anterior pode ser outro, e ao nível do IP, o pacote não tem qualquer ligação com outro pacote do mesmo remetente. Assim, torna-se trivial falsificar o endereço de origem.

É uma técnica na qual o endereço real do atacante é mascarado de forma a evitar que ele seja encontrado. Muito utilizada em acesso a sistemas nos quais a autenticação tem como base endereços IP. Porém, obriga a utilização de outras técnicas para obter as respostas, uma vez que o IP do atacante não é real.

Eficiente também para ataques do tipo DoS para evitar a identificação do atacante e para não sofrer os efeitos colaterais dos ataques *SYN Flood*, *Flangle, Land*.

4.3.3.1 – Como evitar o IP Spoofing?

Basicamente como vimos, o IP *Spoofing* é difícil de detectar, já que não existe confirmação da origem pelos roteadores. Entretanto, o mais complicado do *Spoofing* fica por conta de um ataque de exploração de confiança.

Imagine que o computador A e B possam trocar trafego entre si sem pedir senha. Se um atacante T pudesse fazer-se passar por B, ele teria acesso à A sem necessidade de senhas.

Uma das maneiras de evitar o IP *Spoofing* é criando um firewall com mais de uma interface de rede, e evitasse que um pacote com IP origem da rede internachegasse pela placa externa, ou ainda, filtrar um pacote com o IP de origem da rede externa chegando pela placa conectada à rede interna.

4.3.4 – War Dialing

O *war dialing* é um ataque importante a ser combatido. O atacante utiliza o modem para entrar em uma rede corporativa, pois muitos usuários instalam modem em suas máquinas para – por exemplo – receber fax, entretanto, o modem pode receber dados e ataques. [LIC 03]

Existem empresas que usam modem para administração remota (conectado a um RAS – *Remote Access Server*) dentro da rede e esses acessos normalmente não passam pelo firewall da empresa.

Estreitamente ligada à engenharia social para descobrir os números de acesso (hotlines), o *war dialing* complementa a técnica ao tentar descobrir quais números telefônicos são utilizados para os modem atenderem as chamadas. O *war dialer* é algo que deve ser verificado com cuidado, pois os atacantes possuem uma entrada dos fundos que não passa pelo firewall. “Não adianta construir um muro de 5 m de altura se você deixa uma pequena porta aberta nos fundos da empresa”

4.3.5 – Source Routing

Primeiramente entenderemos o que é o roteamento. Roteamento é o ato de mover informações da origem para o destino para pacotes da camada 3 do modelo OSI (Rede). Ao longo do caminho pelo menos um host é encontrado (o gateway). [CIS 05].

Como já vimos, o roteador não olha o endereço de origem e sim o endereço de destino, e move os pacotes para o próximo roteador, de acordo com uma tabela de roteamento.

Source Routing é uma opção do protocolo IP, que permite que o tráfego vá do cliente A até o servidor B, passando pelo servidor X. O Source Routing pode ser de duas maneiras:

- **SSRR** – *Strict Source and Record Route* (IP código 0x89): Neste caso, no cabeçalho IP (camada 3), devemos colocar todos os roteadores pelo qual o pacote deve passar. Devemos atribuir manualmente a rota (do inicio ao fim) que o pacote deve tomar.
- **LSRR** – *Loose Source and Record Ro*ute (IP código 0x83): Neste caso, no cabeçalho IP (camada 3), devemos colocar por quais roteadores (um ou mais) o pacote deve passar. Se atribuirmos apenas o endereço IP X, o pacote iniciado em A vai à direção à X, que por sua vez encaminhará à B.

O *Source Routing* pode ser usado para os seguintes fins como cita [ISS 05]:

- Mapear a rede: usado em conjunto com o *traceroute*[12] a fim de achar todas as rotas possíveis entre dois pontos da rede;
- *Troubleshooting*: usado para entender porque o cliente A consegue enviar pacotes para B e X não consegue;
- Performance: um gerente de rede decide por forçar uma rota específica ao invés da rota padrão, como por exemplo um link de satélite ao invés da Internet.

Devemos então pensar que tal recurso pode ser utilizado de maneira ilícita para resolver o problema da "volta" dos pacotes de um *Spoofing*, pois ele instruiria a máquina remota a passar os pacotes por ele (*source routing LSRR*), mas com o destino da máquina que sofreu o *Spoofing*.

Também pode ser usado num ataque de prognóstico do número de seqüência do TCP em conjunto com o IP *Spoofin*g, de forma que o servidor atacado devolve o número de seqüência para o hacker e não para o IP falsificado. [CAOR 05]

Este recurso deve ser evitado de ficar habilitado nas operações do dia-a-dia, entretanto, este recurso por constar em [RFC 791], é normalmente habilitado por padrão em quase todos os sistemas operacionais.

4.4 – Negação de Serviço

Ataques de negação de serviço, também conhecida como DoS (*Denial Of Service*) faz com que servidores ou hosts fiquem indisponíveis. Pode-se usar uma técnica do tipo SYN *Flood, Smurf, Flangle, Buffer Overflow*, etc. Normalmente boa parte desses ataques só são possíveis devido a ineficácia dos programadores destes aplicativos alvos.

O ataque de negação de serviço também pode ser distribuído e é caracterizado por ter mais de uma origem (*DDoS – Distributed Denial of Service*)

[12] Aplicativo do sistema operacional usado para verificar por qual rota um pacote está passando. Normalmente usa a técnica ICMP TTL Exceeded ou UDP TTL Exceeded

4.4.1 – Bugs em Serviços ou Sistemas

Falhas na implementação do serviço de rede de um sistema pode dar margens à DoS ou até a abrir "brechas" de segurança como o *buffer overflow* (será visto no item 4.5).

Por exemplo, um *port scan* não é considerado um ataque de DoS, pois tem como objetivo coletar informações, entretanto porque um port scan do tipo Xmas em Windows 98 podia causar uma tela azul (*Blue Screen of Death*), mais tarde, este bug foi corrigido.

Em um servidor de FTP, o qual eu usava há alguns anos, ele simplesmente caia (DoS) com a mensagem de "Este programa executou uma operação ilegal e será fechado" se o nome de usuário fosse maior que 2047 caracteres. Isto seria facilmente contornável se o sistema abrisse um novo espaço de memória quando fosse necessário ou não permitisse que o usuário digitasse algo tão estanho [COL 01].

4.4.2 – SYN Flooding

Este ataque explora o algoritmo de criação de novas conexões do TCP chamado "*Three-way-handshaking*". Quando uma conexão deve ser criada, o remetente envia um pacote SYN ao destino, que aloca os espaço em memória necessário para atender à essa conexão e responde com um pacote SYN+ACK. Desta vez, o remetente reserva o espaço em memória dele e envia outro ACK para iniciar a transferência de informações.

O *SYN Flooding* é ocasionado pelo envio massivo de pacotes SYN, fazendo que o servidor reserve um pouquinho de memória para cada novo pacote que chega, até estourar a pilha de memória (acabar a RAM disponível para o aplicativo). O resultado pode ser apenas uma indisponibilidade momentânea até o travamento completo do sistema operacional.

Devido a este ataque gerar um efeito colateral (todas as respostas de volta ao atacante), normalmente este é usado com a técnica de *IP Spoofing*, fazendo que o servidor fique enviando respostas para locais inexistentes até cair.

4.4.3 – ICMP Flooding.

Usa a mesma técnica do *SYN Flooding*, enviando pacotes de ping (ICMP) até o host cair. Entretanto, esta técnica é difícil de ter efeitos positivos pois o host de destino não reserva memória prematuramente como é o caso do TCP.

Precisaríamos de uma ação coordenada (DDoS) para que isto surta algum efeito prático.

4.4.4 – Smurf/Flanggle

O *Smurf* é um ataque no nível de rede no qual um grande tráfego de pacotes ping (ICMP echo) é enviado para o endereço IP de broadcast da rede tendo como origem o IP da vítima. Com o broadcast cada host passa a responder para a vítima que tem uma avalanche de pacotes na entrada. O tráfego na rede é pode ser multiplicado por um fator entre 10 e 200 vezes.

Como a vitima deve processar estes pacotes, possivelmente haja um excesso de pacotes e ele acabe caindo. Este ataque usa o ICMP Flooding

O Franggle utiliza pacotes UDP echo em lugar de ICMP echo. [CAOR 05]

4.4.5 – Fragmentação IP

Relacionada com o MTU (*Maximum Transfer Unit*) dos pacotes IP, que é a quantidade máxima de dados permitida num pacote. Uma rede Ethernet limita a transferência a 1500 bytes por pacote, enquanto o FDDI permite 4470 e a X25 permite 536 bytes. Quando um pacote FDDI passa por uma rede Ethernet ele é fragmentado em pacotes de 1500 bytes. Esta situação ocorre normalmente quando um pacote IP passa por diversas redes que possuem diferentes MTU.
Quando o pacote chega ao destino, ele espera para que todos os pedaços do pacote cheguem para remontá-los antes de enviar a camada superior.

Isso pode criar a possibilidade de *overflow* na pilha do TCP quando há o reagrupamento de pacotes maiores que o permitido.

Este é a técnica usada por um ataque comum nos anos 90 chamado *Ping Of Death*, que nada mais era do que um ping (ICMP Echo Request) com 65535 bytes de dados, fazendo que os sistemas operacionais travassem ou ficassem momentaneamente indisponíveis, pois era uma situação inesperada do *kernel* dos sistemas operacionais da época. A correção veio através de atualizações de software para os sistemas de rede (*patches*).

Para se proteger, teremos que violar as regras da RFC para ganharmos mais segurança, entretanto e deveremos pesar os favores e contras de perder padronização vs. ganhar segurança. Entretanto, com a padronização da internet, a maioria das redes permite uma MTU de 1500 bytes. Os roteadores/firewalls podem eventualmente descartar pacotes fragmentados para evitar um problema no servidor.

Hoje, a maioria dos firewalls são capazes de impedir os pacotes IP fragmentados.

4.4.5.1 – Ferramentas de Fragmentação IP – Teardrop e Land

O Teardrop é uma ferramenta utilizada para explorar os problemas de fragmentação de IP nas implementações TCP/IP vistas na seção acima. O *land* é uma ferramenta empregada para explorar vulnerabilidades no TCP, onde um pacote é construído especificamente de modo que o SYN tenha o endereço de origem e o destino iguais (IP e porta), ou seja, um *IP Spoofing*. A solução é criar regras para evitar o *IP Spoofing* na rede. [LIC 03].

4.4.6 – Man-in-the-Middle

O ataque do tipo *man-in-the-middle* é um ataque de repetição. O atacante pode inserir, modificar ou criar mensagens entre duas entidades sem que nenhuma delas saiba que a comunicação foi comprometida. Este tipo de ataque é possível inclusive em comunicações criptografadas, como por exemplo o protocolo original *Diffie-Hellman*.

Imagine que o computador A deseja trocar informações com B de forma criptografada e T é o atacante. No momento que A inicia a conexão com B, o host T intercepta essa conexão e engana A de modo que T faça se passar por B. O host A pensa que estabeleceu uma conexão segura com B, quando na verdade ele está com uma conexão segura com T, conforme ilustra a Figura 2.

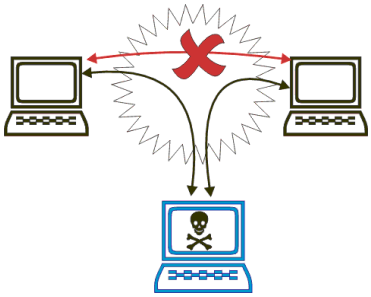


Figura 2: Um ataque man-in-the-middle

Neste caso, o atacante T, abre uma conexão para B criptografada e fica no meio da comunicação. Quando A deseja falar com B, o host T recebe a mensagem de A, descriptografá-a, lê ou altera dependendo do seu desejo e reenvia para B.

Man-In-The-Middle é também conhecido por MITM.

4.4.7 – DNS Tampering, ARP Tampering

DNS *Tampering* e ARP *Tampering* são técnicas usadas para desviar o tráfego para um destino especificado pelo atacante. O protocolo DNS é um serviço que transforma endereços FQDN em endereços IPs que posteriormente são utilizados para formar o endereço de destino dos pacotes IP. O ARP é um protocolo que resolve endereços IP para endereços físicos (*MAC Address*).

Quando um usuário digita www.conexão.com.br, o computador consulta o arquivo de hosts e se não for possível resolver, pergunta ao servidor de DNS. O DNS possui uma estrutura hierárquica. Entretanto, é contraproducente que a cada requisição que os clientes façam, o servidor de DNS procure por toda a Internet para resolver o endereço IP.

Por este motivo, o servidor DNS faz cachê, guardando as respostas das requisições feitas recentemente, assim, se outro cliente perguntar sobre o mesmo FQDN, o servidor de DNS responderá prontamente à solicitação.

O ARP trabalha de forma semelhante, fazendo uma pesquisa na rede local para saber quem é o dono de um determinado endereço IP. Quando a resposta é recebida, o computador local guarda em uma tabela na memória qual é o endereço físico (*MAC Address*) associado ao endereço lógico (IP).

Como ambas as técnicas utilizam cachê, esse é o motivo pelo qual essas aplicações podem ser atacadas.

Se o atacante puder injetar um endereço inválido no cachê, quando o usuário deseja acessar seu banco por exemplo, ele será redirecionado para um endereço que não pertence ao servidor do banco, e sim a uma página falsa com intenção de roubar os dados da conta do cliente do banco.

Esta técnica de também é conhecida como envenenamento de cachê (*Cache poisoning*).

Uma outra exploração possível em versões antigas de sistemas operacionais, era que o atacante enviasse uma resposta ARP para uma vítima fazendo com que ele se tornasse o gateway, por exemplo, para um ataque do tipo *man-in-the-middle*. Os sistemas mais antigos permitiam que uma entrada na tabela ARP fosse incluída mesmo sem que fosse feito uma pergunta, sendo uma fácil técnica de envenenamento de cache.

4.4.8 – Seqüestro de conexão

O seqüestro de conexão é um ataque ativo que explora a injeção de dados em conexões para uma determinada máquina. Normalmente associado à protocolos de senha de uma única vez (*one time password*), este permite que possamos redirecionar o tráfego entre dois hosts para ataques de exploração de confiança ou *man-in-the-middle*.

Este ataque também pode ser conhecido por seqüestro de sessão ou *session hijacking*.

O ataque tem como base a exploração do estado de desincronização entre dois lados de uma conexão [LIC 03]. O protocolo de transporte UDP é altamente susceptível a ataques do tipo seqüestro de conexão, já que não conta com nenhum tipo de dispositivo de sincronia. O protocolo de transporte TCP usa técnica chamada janela deslizante, utilizando os campos de Numero de Seqüência e Número de Reconhecimento (*Sequence Number* e *Acknowledgment Number*).

Em uma conexão TCP um host atacante deve utilizar-se de IP *Spoofing* simultaneamente, pois uma conexão IP possui 5 parâmetros (protocolo, IP de origem e destino, porta de origem e destino) que devem continuar os mesmos durante o Seqüestro.

Em linhas gerais, o atacante usa isto para explorar uma pré-autenticação de um cliente. Imagine que o cliente A pode se conectar à um servidor sem digitar senha, pois o administrador confia que os pacotes que chegam de determinado endereço IP são realmente do cliente, e o atacante quer se usufruir desta confiança. Para o atacante, ele deve se fazer passar pelo cliente A para que o servidor possa aceitar a conexão sem questionar senha.

Outra utilidade é usar uma autenticação já feita por um usuário legítimo com um servidor para injetar comandos (dados) que o atacante deseja. Infelizmente não é uma tarefa simples, pois o atacante deve conseguir acompanhar a sessão de autenticação do cliente e interromper a conexão logo após o cliente ser autenticado.

4.4.9 – Predição de número de Seqüência

O protocolo TCP usa dois números para organizar o sequenciamento do fluxo em uma conexão. Basicamente, cada um deles serve para uma direção do tráfego *full-duplex* do TCP.

A predição de número de seqüência (*Sequence Number Prediction*) é basicamente saber qual será o próximo número de seqüência válido, para que possamos usar em um ataque do tipo seqüestro de conexão. O ataque mais famoso de predição de número de seqüência foi o de Kevin Mitnick *versus* Shimomura, ao qual será apresentado a frente neste capítulo.

Em alguns sistemas o número de seqüência é de fácil predição, pois sobe um valor fixo a cada evento esperado. No caso do ataque de Shimomura, o servidor subia o número de seqüência em 128000 a cada 100ms ou a cada nova conexão. Hoje em dia, as boas implementações de TCP fazem essa tarefa bem mais difícil. As implementações modernas, como a TCP New Reno, usam janelas pequenas e que crescem conforme a conexão se procede (*Slow Start Window*). O número de reconhecimento não é mais baseado no relógio + número da conexão, e sim em um número aleatório. A chance de se descobrir é pequena, pois a proporção é de uma janela de 4196 bytes para 4 bilhões possíveis (4GB).

4.4.10 – Redes sem Fio

Na vida moderna, todos os dias redes sem fios são instaladas. Estas nos dão uma maior mobilidade, boa velocidade e os equipamentos modernos fazem a tarefa de instalação ser fácil. Os novos computadores pessoais já estão vindo de fábrica com uma placa do tipo cliente sem fio (*wireless*).

Para a segurança, as Redes sem fios devem ser encaradas como um *switch* público onde qualquer indivíduo pode se conectar. Adicionalmente devemos ter em mente que as ondas não saem do computador do usuário diretamente para o *Access Point*[13], estes sinais espalham-se e podem ser capturados a uma boa distância, facilitando o *eavesdropping*.

4.4.10.1 – Wireless 802.11a/b/g

Para que diversos dispositivos possam trocar informações, foi necessário criar um padrão. O padrão mais comum para redes sem fio (*wireless*) é o criado pelo IEEE (*Institute of Electrical and Electronics Engineers, Inc.*). O projeto 802.11 define as redes sem fio.

Este padrão vários protocolos entre eles:

- 802.11b: Redes DSSS com taxa máxima de 11 Mbps na freqüência de 2,4GHz;
- 802.11g: Redes OFDM/DSSS com taxa máxima de 54 Mbps na freqüência de 2,4GHz e totalmente compatível com 802.11b;
- 802.11a: Redes OFDM/DSSS com taxa máxima de 54 Mbps na freqüência de 5GHz.

Os padrões permitem que todos os equipamentos que sigam estas normas, possam trocar dados entre si, porém permite também que qualquer pessoa possa acessar uma rede 802.11 ou capturar os quadros com informações da sua rede.

Se as redes sem fio criam uma maior facilidade no dia-a-dia, estas requerem um índice adicional de segurança, entre eles:

- Desligar compartilhamento de disco, impressora, etc. quando não estiver em uso;
- Desabilitar o modo *ad-hoc*. Utilize esse modo apenas se for absolutamente necessário e desligue-o assim que não precisar mais;
- Usar criptografia WEP (*Wired Equivalent Privacy*) sempre que possível;
- Considerar o uso de criptografia nas aplicações, como por exemplo o uso de PGP para o envio de e-mails, SSH para conexões remotas ou ainda o uso de VPNs;
- Habilitar a rede wireless somente quando for usá-la e desabilitá-la após o uso.
- Mudar configurações padrões que acompanham o seu AP;
- Habilitar, se houver, a opção para não divulgar o nome da rede (SSID);
- Trocar as chaves WEP que acompanham a configuração padrão do equipamento. Procure usar o maior tamanho de chave possível (128 bits);
- Desligar seu AP quando não estiver usando sua rede;
- Usar controle de *MAC Address* ou autenticação dos clientes.

[13] *Access Point* é o equipamento concentrador mais comum de conexões em uma rede sem fio. Ele recebe e distribui os sinais sem fio para os computadores conectados, trabalhando como uma *bridge*.

Usuários de redes sem fio devem ter em mente que a criptografia utilizada em 802.11 (WEP) já foi quebrada no passado e que não deve ser a única forma de proteção.

Em aplicações onde a segurança da informação é crítica devemos utilizar alguma solução proprietária, como é o caso do protocolo TurboCell da Karlnet [BRG 05] que oferece um índice adicional de segurança, pois o protocolo não é público e faz com que o atacante não consiga enxergar os pacotes no ar.

4.4.10.2 – Celulares (TDMA, CDMA, GSM)

No antigo sistema analógico AMPS (*Advanced Mobile Phone System*) onde ouvir conversas era uma tarefa relativamente simples, e podia ser feito por um scanner de rádio. Nesta época a segurança do aparelho era baseada no número de série do aparelho (ESN – *Electronic Serial Number*) que era enviado em aberto. Qualquer um então poderia "ouvir" e clonar um aparelho.

Os sistemas CDMA e TDMA, conhecidos pelo público como sistemas digitais, dificultaram a captura dos dados das conversações já que as mesmas eram enviadas moduladas digitalmente e multiplexadas por tempo (TDMA – *Time Division Multiplex Access*) ou por divisão de código (CDMA - *Code Division Multiplex Access*), porém mesmo assim um *hacker* poderia capturar os dados, decodificá-lo e interceptar as informações.

No GSM (*Global System for Mobile)*, as partes importantes da segurança em celulares ficam por conta de autenticar o assinante, confidencialidade da informação (dados, voz e sinalização). Especificamente em GPRS (*General Packet Radio Service* - Serviço de Radio Geral por Pacotes) é preciso ainda proteger informações que dizem respeito à cobrança. Como o pagamento em GPRS é feito sobre volume de dados, informação desnecessária precisa ser filtrada e os tickets de cobrança gerados pelo sistema, precisam ser protegidos.

As redes GSM-GPRS autenticam a identidade do assinante através de um mecanismo de desafio e resposta (*challenge-response mechanism*). O sistema de segurança do GSM depende do SIM (*Subscriber Identity Module*). O SIM contém uma identidade única (IMSI), uma chave pessoal (Ki), um algoritmo de geração de chaves (A8), um algoritmo de autenticação (A3) e um número de Identificação Pessoal (PIN – *Personal Identificator Number*) e o telefone contém um algoritmo de criptografia (A5). Os algoritmos (A3, A5 e A8) estão presentes em uma rede GSM. O Centro de Autenticação (AUC), parte da rede GSM, consiste no banco de dados de autenticação dos assinantes. Todos os itens são necessários. Isto resolve os problemas de segurança e impede fraudes como clonagem e escuta das conversas telefônicas.

O A5 é o algoritmo de criptografia que tem três versões:

- A5/2 – utilizada geralmente pelo GSM (exceto quem usa A5/1).
- A5/1 – utilizada nos EUA e Europa e é mais robusta do que o A5/2.
- A5/0 – utilizada por países sob sanções da ONU e não utiliza criptografia.
- A5/3 – uma nova versão do A5, baseada no algoritmo *Kasumi* (que também será usada em 3G) foi regulamentada e padronizada, porém ainda não está efetivamente em uso.

O A5/1 (versão com exportação restrita) tem chave de 64 bits, mas seus 10 últimos bits não são usados, gerando uma chave real de 54 bits. Uma implementação do algoritmo A5/1 pode ser encontrada na Internet, criptoanalisada em 1999, baseada na característica do protocolo GSM que durante os primeiros 0,1s os codificadores de voz

codificam um silêncio gerando aproximadamente 1300 bits de dados que podem ser criptoanalisados. Mais tarde descobriu-se algo semelhante no A5/2, e oficialmente um ataque ao A5/2 tem complexidade de $O(2^{16})$. [BRG 03]

Resumidamente, em laboratório, o GSM já foi clonado e apesar da grande dificuldade deste procedimento *over-the-air*, sabe se que isto é possível. Além disso, é possível que em poucos dias um SIM Card seja criptoanalisado de modo a vazar a chave do cartão.

Se tudo não bastasse, um funcionário que violar as políticas de segurança pode acabar por causar uma falha de segurança sem que o atacante precise usar de avançadas técnicas de criptoanálise.

4.5 – Ataques às Aplicações

Um ataque de aplicação são aqueles que dependem de uma falha de software para ocorrer. Com o passar dos anos, o número de falhas nas camadas de rede vem diminuindo, pois a maioria delas já foi corrigida.

Por outro lado, cada dia são criados novos aplicativos, e novos programadores (iniciantes) começam. Com a pressão para entrega em prazos curtos dos aplicativos, muitas vezes os testes não são bem feitos e abrem brechas para que um atacante use isto a seu favor de maneira à efetuar um ataque à aplicação.

4.5.1 – Buffer Overflow

Buffer Overflow é uma técnica de ataque à aplicações válidas. Imagine uma máquina servidora com todas as portas fechadas, técnicas implementadas de IP Spoofing, IDS, etc. e apenas o servidor WWW rodando. Como podemos atacar?

A resposta é *Buffer Overflow*. Esta técnica é causada por uma falha na programação do servidor de WWW. Ultimamente esse ataque vem sendo comum, e a maioria dos patches que são lançados corrigem algum tipo de *Buffer Overflow* ou *Format String Attack*.

Um caso simples (e clássico) era um servidor de FTP que possuía um espaço de 2 Kbytes de memória para o nome do usuário. Basta alguém de fora digitar um nome de usuário com 2048 bytes + alguns bytes de código binário, para inserir um código seu na memória do servidor remoto, rescrevendo a pilha o endereço de retorno (obviamente para o código que está colocado logo após o nome de usuário).

4.5.1.1 – Exemplos práticos

O problema de *Buffer Overflow* está sempre associado com uma falha de segurança. Muitos dos problemas de segurança estão associados à ataques de nível de aplicação. [LIN 03]

O conhecimento básico de C e de como funciona o Linux em uma plataforma x86 é desejável para entender o porquê isso ocorre em detalhes.

O buffer é um espaço de memória alocado para uma finalidade de uso em por um aplicativo, tal como um ponteiro ou uma matriz de valores. Em C não existe verificação automática de onde começa um buffer e onde ele termina, isso significa que um usuário pode escrever em um buffer indefinidamente. Por exemplo:

```
int main () {
    int buffer[10];
    buffer[20] = 10;
}
```

O programa acima é válido e não vai gerar erros durante a execução, mas vai sobrescrever endereços de memória além do alocado para o buffer. Como esse programa é na verdade um processo do sistema, alguns dados estão no espaço de memória, logo após os dados usados para leitura do teclado. Na Figura 3 podemos ver a estrutura básica de um processo.

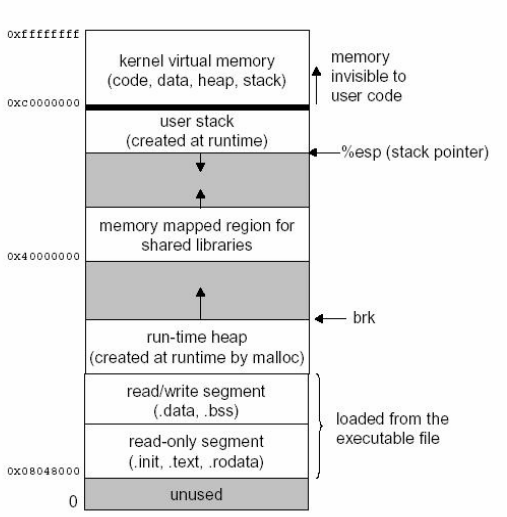


Figura 3: Estrutura básica de um processo na plataforma x86, adaptada de [LIN 03]

A região de pilha é um bloco de memória contínua que contém os dados. Um Ponteiro (SP) aponta para o topo da pilha. Quando uma função é chamada, os parâmetros da função são puxados para do fim para o começo (da direita para esquerda). Quando um endereço de retorno é criado (o endereço que será repassado à quem chamou esta função), seguido de um ponteiro de frame (FP) é enviado à pilha. O FP é usado para referenciar as variáveis locais e as variáveis da função (local ou global). Na maior parte das implementações dos compiladores, a pilha cresce de cima para baixo (do fim para o começo).

Vejamos mais um exemplo de uma função, com três parâmetros que cria dois buffers. Lembrando que na memória, os buffers são múltiplos de quatro bytes.

```
void funcao (int a, int b, int c) {
        char buffer1[5];
        char buffer2[10];
}

int main() {
        funcao(1,2,3);
}
```

Figura 4: Fração de código exemplo em C mostrando a passagem de parâmetros

(deixado em branco intencionalmente)

a pilha será algo como (veja Fig. 5):

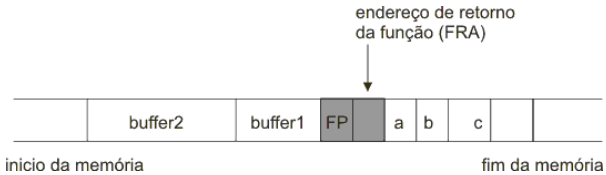


Figura 5: Estrutura de memória FRA (Function Return Address)

Veja que a, b e c estão mais no "fim" e o buffer está mais no começo. O buffer1 ocupou 8 bytes e o buffer2 ocupou 12 bytes (devido ao próximo múltiplo de 4). Temos também o endereço de retorno que será executado logo ao fim do processamento da função.

Agora que entendemos como o C funciona, vejamos um exemplo mais prático de overflow do buffer (ainda sem querer):

```
void funcao (char *str) {
   char buffer[16];
   strcpy (buffer, str);
}
int main () {
  char *str = "Aqui tem um texto com mais de 16 bytes";
  funcao (str);
}
```

Aqui claramente os 28 bytes sobrescreverão o buffer e entrarão na parte do FP e do endereço de retorno da função.

A idéia é essa para o ataque. Se *str* fosse lido pelo teclado ou por algum outro parâmetro de uma comunicação de rede, poderíamos escrever um *str* para alterar o endereço de retorno de função, e executarmos, ao invés do código de retorno do valor para a função referenciada, executaríamos um código arbitrário que faça algum tipo de "escalagem de privilégios" (*Privilege escalation*) ou instalação de algum código malicioso no computador que sofreu o ataque.

Com o código apropriado, podemos rapidamente ter acesso ao *shell* com direitos de super usuário.

4.5.1.2 - Como evitar?

Como este tipo de ataque se utiliza de uma falha de programação, a maneira mais simples de evitar, é usando boas técnicas de programação. Se você utiliza um sistema de terceiros, o importante é manter seu programa atualizado sempre. Em [LIN 03] vemos que existem várias funções vulneráveis em C que devem ser evitadas, ou se usadas, devemos fazer uma invalidação da pilha.

Os fabricantes costumam possuir sites, listas, e-mails para avisar de vulnerabilidades desse nível.

Remover programas não necessários do servidor também pode ajudar, pois diminui a possibilidade de *buffer overflow* se o programa não estiver executando.

4.5.2 – Ataque de Formato de String

O Ataque de Formato de String (*Format String Attack*) é um ataque que possui alguma semelhança com o *buffer overflow*. Em julho de 2000, um grupo de hackers publicou uma técnica de exploração desconhecida pela comunidade de segurança. Eles

afirmavam que dominavam a técnica desde 1994 possuindo acesso a diversos sistemas remotos através desta técnica.

A função de biblioteca printf da linguagem C é utilizada na grande maioria dos programas para formatar dados para apresentação, seja esta para tela ou arquivo. Explorando o comportamento normal desta função enquanto processa os dados de formatação e os parâmetros passados, permite que sejam inseridos dados arbitrários (por exemplo código executável) e que seja modificada a pilha de execução do programa.

Isso porque quando uma função printf() (como por exemplo printf(char *fmt,...)) é chamada, o fmt é fornecido pelo usuário. O usuário pode colocar no formatstring %s %p, %x e fmt. O printf na mesma hora irá convertê-los com os argumentos fornecidos.

As várias funções printf recebem strings formatadas (%s, %n, %p, etc.) em seus argumentos, mas não convertem, ou seja, pensam que as strings representam entradas de novos argumentos. O problema é que o printf() não sabe onde está o delimitador de fim do argumento. Quanto a nova string for formatada, ela somente lê os próximos dados na pilha (stack). Desta forma, é possível enviar um pequeno programa pelos dados de entrada (com direção à pilha) e que seja desviada a execução para ele, permitindo assim a entrada de um invasor. Esta técnica também é conhecida como *FormatBugs*. [TRE 03].

Detalhes de implementações podem ser encontradas em [NAS 00].

4.5.3 – SQL Injection

O *SQL Injection* é um tipo bem mais simples de ataque, que lembra muito o *buffer overflow*. A intenção do SQL Injection é que possamos criar um código que possa ser executado em um servidor de banco de dados, para que "vaze" informações ou que tenhamos acesso à partes do banco de dados que não deveríamos ter acesso.

Isto acontece por uma falha na programação de um website, normalmente escrito por programadores inexperientes em JSP, PHP, CGI, Perl, ASP, Asp.net, etc.

4.5.3.1 – Como funciona o SQL Injection?

Vejamos um código ASP e HTML legítimos que validam um usuário e uma senha. Se este usuário for validado, mostraremos uma mensagem de sucesso. Claro que na vida real poderíamos estar ganhando acesso à intranet contendo os dados de fornecedores ou clientes desta empresa.

formulario.htm

```
<FORM METHOD="POST" ACTION="resp.asp">
        <INPUT TYPE="TEXT" NAME="login" VALUE="">
        <INPUT TYPE="PASSWORD" NAME="senha" VALUE="">
</FORM>
```

resp.asp

```
<%
l = Request.Form("login")
s = Request.Form("senha")
Set cn = Server.CreateObject("ADODB.Connection")
cn.Open Session("conexaoBanco")
Set rs = cn.execute ("Select count(*) as contador
                from Usuarios where Login = '" & l & "' AND senha = '" & s & "';")
If rs("contador") > 0 then
        Response.write "SUCESSO"
Else
        Response.write "SUCESSO"
End If
Set rs = Nothing
Set cn = Nothing
%>
```

Aqui vemos que o programador deixou uma brecha para entrarmos sem sermos chamado. Se um usuário honesto digita o username "pedro" e a senha "123456" o código ASP irá enviar um *statement SQL* semelhante à:

```
Select count(*) as contador from Usuarios where Login = 'pedro' AND senha = '123456'
```

Porém, se nós enviássemos na senha ao invés de 123456, colocássemos algo "errado", como por exemplo "` OR 1=1"

Então a *query* seria:

```
Select count(*) as contador from Usuarios where Login = 'pedro' AND senha = '' OR 1=1
```

Vemos que como 1=1, o usuário seria autenticado, pois o OR 1=1 faria existir mais de uma linha no *resultset*.

Ainda é possível executarmos comandos e despejarmos uma tabela inteira em um arquivo para ser lido através do servidor web, executarmos comando de shell, listar as tabelas, acessar registros indevidos, entre outro [SQL 02]

4.5.3.1 – Como evitar o SQL Injection?

Novamente, o importante para evitar o SQL Injection está basicamente em como o programador escreve os seus scripts.

Para evitar o SQL Injection basta:

- Filtrar todos os caracteres de aspas simples e duplas, barras e barras contrárias, caracteres extendidos (como NULL, CR, CRLF) de:
  - Formulários dos usuários (POST);
  - Parâmetros da URL (GET);
  - Valores de Cookies;
  - Valores de arquivos de configuração (INI, CFG).
- Converter os valores de inteiro para string antes de enviar para o servidor de banco de dados ou usar funções de checagem (como o ISNUMERIC do ASP ou ASP.net)
- Rodar a aplicação web conectando no banco de dados com um usuário com o mínimo de privilégios

4.5.4 – Ataque na WEB

Bugs em servidores de WEB são vulnerabilidades altamente exploradas hoje em dia devido ao grande numero de servidores web no mundo. São ataques a servidores WEB, navegadores, scripts CGI (*Common Gateway Interface*) e scripts ASP (*Active Server Pages*), etc. [CAOR 05]

É por meio destes que os *hackers* conseguem fazer os *web defeaceaments*.

4.5.4.1 – Poison Null

Esta técnica permite que o conteúdo de um diretório seja visto, pois em alguns casos é até possível ler e alterar arquivos dos servidores WEB. O mecanismo utilizado

mascara comandos de checagem de segurança do CGI, ocultando-os por trás de um byte NULL ("*null byte*") – um pacote de dados que o script CGI não detecta – a menos que o programador tenha escrito um código específico para tal.

4.5.4.2 – Upload Bombing

Upload Bombing afeta sites que oferecem upload, tais como desenhos, fotos, blogs, webmails, etc. Normalmente usado para preencher o disco do servidor com dados inúteis. Acontece quando os scripts não verificam o tamanho dos arquivos antes de serem enviados para o servidor.

4.5.4.3 – WebSpoofing ou Hyperlink Spoofing

Por meio das técnicas de *Web Spoofing* ou *Hyperlink Spoofing* o usuário por algum outro método de ataque cai em uma página que é falsa. Por exemplo, é comum nos dias de hoje recebermos um e-mail com um link que aparentemente é direcionado para o nosso banco e que na verdade redireciona à um servidor com uma página espelho falsificada que tem como função coletar dados (tais como senha e conta). A única maneira de confirmar a identidade do site é através do SSL.

4.5.5 – Exploits

Um *exploit* é um pequeno programa ou *script* que permite tirar proveito de vulnerabilidades de outros programas - como o próprio sistema operacional. Geralmente elaborados por especialistas (*hackers*) como programas de demonstração das vulnerabilidades, pode cair em mão erradas e causar bastante estrago [WIK 05-2].

Os *script kiddies* são os maiores usuários dos *exploits*.

Os *exploits* usam diversas técnicas (spoof, buffer overflow, upload bombing, SQL Injection, etc.) para conseguir entrar em um sistema. Normalmente o *exploit* é lançado quando o fabricante já está ciente do problema, ou já lançou uma correção para o mesmo.

Nestes casos, os administradores que não fizerem as correções necessárias podem estar vulneráveis aos exploits usados pelos *script kiddies*.

Na Internet é fácil encontrar uma dezena desses programas para vários sistemas operacionais ou aplicativos comuns no mercado [SEC 05].

4.5.6 – Vírus, Worms, Cavalos de Tróia

Vírus, *Worms* e Cavalos de tróia podem ser definidos como programas que são instalados em computadores, sem a autorização ou consciência do usuário, a fim de explorar recursos, roubar dados e prejudicar o computador.

Entretanto cada um deles tem uma característica própria:

- **Vírus:** São códigos incluídos em programas legítimos que atuam em tempo de execução. Conceitualmente, espalham-se com a ajuda humana, transportando arquivos entre locais diferentes. Alguns deles se proliferam sozinhos dentro do próprio computador. Existem Vírus de Setor de Boot, Vírus de Arquivos Executáveis, Vírus de Macro e Vírus de Script.
- **Worms: (Vermes):** Têm a capacidade de espalharem-se automaticamente entre sistemas utilizando vulnerabilidades conhecidas em produtos, tais

como o Windows. Eles podem, por exemplo, se proliferar por e-mail ou conectando-se em portas que possuem programas susceptíveis à buffer overflows.

- **Cavalo de Tróia:** É um programa aparentemente benigno mas que se comportam de maneira inesperada. Podem estar neste grupo os programas espiões (que coletam informações digitadas, tais como senhas, números de cartões de crédito, etc.)

4.5.6.1 – Como evitar?

Uma boa política de segurança pode ajudar a reduzir o uso ou acesso indevido à locais com maior possibilidade de vírus, entretanto uma das mais eficazes e atualmente utilizadas soluções são os programas antivírus. Esses programas procuram por características (assinaturas) no computador para detectar, bloquear e remover o programa malicioso. Eles também podem remover Worms.

Evitar instalar programas de origem desconhecida, e limitar o uso da Internet a sites que não façam parte do cotidiano necessário da organização.

Os cavalos de tróia, com a proliferação da Internet, estão também aparecendo com outros nomes, tais como Spywares, Adwares, etc. Existem programas como o Microsoft Antispyware, Spybot e LavaADWare que podem remover estes programas maliciosos do computador.

4.6 – Ataques de Engenharia Social

4.6.1 - Definição

Então, já definimos ataque de engenharia social como a denominação dada às atividades que visam o acesso não autorizado à informações por meio de ações fraudulentas, subterfúgios, influência, uso indevido de informações confidenciais, etc.; sem necessariamente um grande aporte tecnológico. [CAOR 05]

A engenharia social usa como maior falha de segurança o alvo do ser humano. Para ter sucesso na engenharia social, é importante entendermos o ser humano e como podemos persuadi-los a fazer o que o *hacker* quer. A falta de informação, o medo de perder o emprego e a vontade de ascender na corporação faz com que ele aja desta maneira. Nesse sentido, observa-se que a engenharia social possui uma seqüência de passos na qual um ataque pode ocorrer: [ESP 04]

1. Coleta de informações: O hacker busca qualquer informação que lhe possa ser (ou não) útil no futuro, tais como CPF, data de nascimento, nome dos pais ou amigos, manuais da empresa, políticas de segurança, etc. Essas informações ajudarão no estabelecimento de uma relação com alguém da empresa visada.
2. Desenvolvimento de relacionamento: O *hacker* então explora a natureza humana intrínseca de confiar nas pessoas até que se prove o contrário.
3. Exploração de um relacionamento: O hacker então vai tentar obter informações da vítima, tais como senha, agendas, dados de contas bancárias, cartões de crédito, contatos, etc.

4. Execução do ataque: O hacker ou engenheiro social realiza o ataque a empresa ou vítima, fazendo uso de todas as informações e recursos obtidos.

4.6.2 – Evitando Ataques de Engenharia Social

O ataque de engenharia social tem sido mais e mais importante no dia a dia de grandes organizações, e o principal foco é a falha do ser humano em não saber quais informações devem e não devem ser divulgada. Entretanto, há mecanismos através dos quais uma organização pode implementar a fim de detectar e prevenir ataques de engenharia social. Tais medidas visam, principalmente, atenuar a participação do componente humano. Essas medidas compreendem [ESP 04]:

- Educação e treinamento – Importante conscientizar as pessoas sobre o valor da informação que elas dispõem e manipulam, seja ela de uso pessoal ou institucional. Informar os usuários sobre como age um engenheiro social.
- Segurança física – Permitir o acesso a dependências de uma organização apenas às pessoas devidamente autorizadas.
- Política de segurança – Estabelecer procedimentos que eliminem quaisquer trocas de senhas. Por exemplo, uma política contendo recomendações de como utilizar o sistema, a Internet, E-mail, Programa de mensagem instantânea, Antivírus, etc.
- Controle de acesso – Os mecanismos de controle de acesso tem o objetivo de implementar privilégios mínimos a usuários a fim de que estes possam realizar suas atividades.

4.6.3 - Dumpster Diving

Dumpster Diving é o ato de procurar o lixo em busca de artefatos úteis para um ataque. Pode ser considerada uma maneira de se conseguir informações importantes vindo de papéis de rascunho de funcionários, por exemplo. É um tipo de Engenharia Social, porém o *dumpster diving* apenas buscam em lugares que foram eliminados do local alvo.

E deve-se ter em mente, que os *dumpsters* tem uma enorme paciência para buscar informações, e até mesmo juntar pedaços de papel rasgados a fim de conseguir a informação completa.

Entre as informações que podemos eventualmente encontrar estão rascunhos de projetos, senhas, nomes e dados de clientes, números de cartão de crédito (aqueles comprovantes que jogamos fora todos os dias), planos de trabalho, lista de funcionários ou até mesmo *hard disks* com problemas (que podem ser recuperado em laboratório externo).

4.6.3.1 – Como evitar os "Dumpsters"?

Empresas ou Organizações com alto índice de segurança se preocupam com essa possibilidade. Evitar um problema com isso é relativamente simples:

- Picotadores de Papel em todos os documentos;
- Equipamentos que podem destruir coisas duras, tais como cartões magnéticos, CDs e DVDs, Circuitos Impressos, etc.;

- Destruir informações de HDs defeituosos, geralmente com fornos de alta temperatura, *Thermite* (um processo para derreter metal), ou até mesmo rolos compressores.

4.6.4 - Ataque Físico

Podemos entender isso da maneira simples.

- Roubos ou furtos: podemos levar os dados de uma empresa de diversas maneiras. Uma delas é roubar um servidor (relativamente difícil), outra mais fácil é roubar mídias de backup (fitas DAT, DLT, CDs, etc.)
- Destruição de informação: A única recompensa ao atacante seria que o alvo ficaria incapacitado de operar por um tempo ou para sempre. Exemplos seriam incêndios propositais, atentados a bomba, jogar água nas instalações (disparar os sistema de incêndio).

Podemos evitar facilmente com segurança física, tais como seguranças patrimoniais treinados e armados, portas com controle de acesso, câmeras (CFTV), etc.

4.6.5 – Exemplo 1 – Engenharia Social

Citamos neste trabalho um exemplo clássico: o caso Rifkin. Este indivíduo roubou US$10,2 milhões em 26 de outubro de 1978 do banco que hoje em dia não existe mais *Security Pacific National Bank* em Los Angeles, EUA.

Esse é uma das maiores fraudes bancárias da história, mas provavelmente nunca ouviu-se falar em Stanley Rifkin. Isto porque ele não usou nenhuma arma, não explodiu o cofre ou nem mesmo *hackeou* o sistema do banco. Agindo sozinho, ele fez um ataque de engenharia social usando um telefone público do banco em horário comercial para transferir o dinheiro para uma conta em um paraíso fiscal.

Rifkin era contratado de uma empresa que desenvolvia um sistema de backup para os dados da sala de transferência. Ele tinha acesso físico à sala de onde os funcionários autorizados todas as manhãs recebiam as senhas que mais tarde seriam usadas para as transações daquele dia. Os funcionários do banco primeiramente achavam que nunca iriam ser roubados dentro de um banco, e devido a uma falta de empenho ou displicência eles simplesmente anotavam a senha em um papel e afixavam-o próximo ao telefone, de maneira que não precisassem memorizar uma senha todos os dias.

Rifkin, não precisou de muito esforço. Durante algum tempo ele ouviu as ligações para se familiarizar com as gírias e termos utilizados durante as transferências, nomes das pessoas do lugar. No dia da fraude, ele entrou na sala com a desculpa de corrigir alguma falha no sistema de backup, memorizou a senha do dia e saiu. Foi até o saguão do prédio e fez a ligação para a sala se fazendo passar por Mike Hansen, um membro do dep. Internacional do Banco. A atendente pediu o número do escritório, o que Rifkin já tinha, e a senha. O resultado da ação foi a transferência de US$10,2 milhões do Irving Trust Company de Nova York para o Wozchod Handel Bank de Zurique na Suíça.

Rifkin poderia até realmente ter fugido e desaparecido com o dinheiro, permanecendo anônimo para sempre se não tivesse se vangloriado em cima do seu advogado, que o delatou para a polícia. Neste dia, imediatamente a polícia notificara o

banco. O incrível é que até então, a segurança do banco nem mesmo estava ciente que faltava tanto dinheiro em sua agência. [LAB 04, CAOR 05, ALE 02, MIT 02]

Este é um exemplo clássico de engenharia social, de forma a burlar as políticas de segurança que só priorizam a tecnologia.

4.6.6 – Exemplo 2 – Dumpster Diving

Em [ANS 05], é citado um exemplo bastante interessante:

Um incidente da história, conta que um estudante (Jerry Schneider) foi procurar por algumas coisas em uma lata de lixo e encontrou um manual rejeitado de um sistema automatizado (precariamente desenvolvido) de pedidos/entregas jogado fora pela Pacific Telephone. Com essa informação ele foi capaz de fazer algo semelhante. Ele pediu o equipamento e teve um sucesso e acabou com uma loja cheia, que acabou vendendo de volta para a empresa. Ele foi julgado e condenado a US$500 e 40 dias de prisão, mas acabou sendo contratado como gerente de segurança da mesma.

4.6.7 – Exemplo 3 – Cavalo de Tróia e Engenharia Social

Este é outro exemplo ocorrido de Engenharia Social ligado ao uso de Cavalo de Tróia. A história começa em Agosto/2000, quando Edgar fez o primeiro contato com Daniele, 12, em uma sala de bate papo. O par trocou e-mail antes de Edgar enviar uma foto dele para a jovem Daniele.

De fato, a foto continha um cavalo de tróia que comprometeu o computador da família de Daniele. Um mês depois, o pai de Daniele, Ravi, foi surpreendido por uma fraude em sua conta bancária através de seus cartões de crédito. A policia ficou decepcionada com a investigação tentando adivinhar os passos dados pelo hacker e como ele encobriu seus rastros.

Entretanto, em 2001, Edgar voltou a flertar com Daniele, porém esperta e sabendo dos truques, desta vez ela enviou umas questões tais como sua cor favorita, seus programas de TV e qual é o seu nome. Então o garoto de 15 anos respondeu a todas as perguntas, até incluindo seu telefone celular, pensando que iria obter êxito novamente neste tipo de ação. Com algumas semanas, Daniele conseguiu os dados e passou para a polícia que prendeu Edgar em Junho de 2001, e seu computador com as provas dos seus crimes, incluindo fraudes de US$710 do pai de Edgar e US$2000 em outras fraudes.

Edgar foi pego porque esqueceu-se em quem tinha dado o golpe há 1,5 anos antes. O Xerife Kenneth Barr sentenciou Edgar, agora com 18, a 100 horas de serviços comunitários. [REG 03]

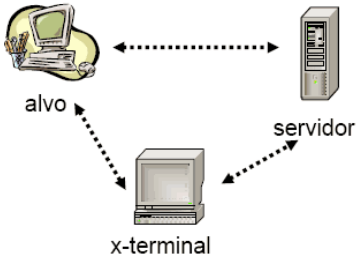
4.6.8 – Exemplo 4 – Ataque de Shimomura – Vários Ataques

Este é o ataque clássico do TCP/IP. O Ataque foi muito bem pensado, e usa as técnicas de DoS, IP Spoofing, Seqüestro de Conexão TCP, Predição de Número de Seqüência e Exploração da Confiabilidade.

O Hacker era Kevin Mitnick, condenado e preso em Fevereiro 95, libertado em janeiro de 2000. [NAC 02]

O ataque clássico contra Shimomura aconteceu em etapas bem definidas. Primeiramente o ataque iniciou 14h09min em 25 de dezembro de 94. Os comandos foram:

```
14:09:32 toad.com# finger -l @target
14:10:21 toad.com# finger -l @server
14:10:50 toad.com# finger -l root@server
14:11:07 toad.com# finger -l @x-terminal
14:11:38 toad.com# showmount -e x-terminal
14:11:49 toad.com# rpcinfo -p x-terminal
14:12:05 toad.com# finger -l root@x-terminal
```



[NAC 02]

Figura 6: Comandos executados por Mitnick

O objetivo de Mitnick era entender se havia alguma confiança entre o servidor (computador de Shimomura) e o servidor de terminal.

Seis minutos depois, vimos um TCP SYN Flooding "half-open" vindos de 130.92.6.97 (endereço falso gerado através de "IP Spoofing") em destino à TCP/513. Isto causou um DoS para evitar que o x-terminal (equipamento do Shimomura) pudesse responder a SYN-ACKs inesperados com um TCP RST.

```
14:18:22.516699 130.92.6.97.600 > server.login: S 1382726960:1382726960(0) win 4096
14:18:22.566069 130.92.6.97.601 > server.login: S 1382726961:1382726961(0) win 4096
14:18:22.744477 130.92.6.97.602 > server.login: S 1382726962:1382726962(0) win 4096
14:18:22.830111 130.92.6.97.603 > server.login: S 1382726963:1382726963(0) win 4096
14:18:22.886128 130.92.6.97.604 > server.login: S 1382726964:1382726964(0) win 4096
```

Figura 7: TCPDump dos pacotes SYN do ataque de Mitnick

Como a porta 513 (*login*) é privilegiada, (< IPPORT_RESERVED), server.login pode agora pode fazer o ataque de spoofing e conectar na máquina x-terminal através de rsh e rlogin.

A segunda parte, é fazer a predição do número de seqüência através de 20 tentativas de conexões de apollo.it.luc.edu para x-terminal.shell.

```
apollo.it.luc.edu> x-terminal: S 1382726990

      x-terminal > apollo.it.luc.edu: S 2021824000 ack1382726991
      apollo.it.luc.edu> x-terminal: R 1382726991
      apollo.it.luc.edu> x-terminal: S 1382726991
      x-terminal > apollo.it.luc.edu: S 2021952000 ack1382726992
      apollo.it.luc.edu> x-terminal: S 1382726992
      x-terminal > apollo.it.luc.edu: S 2022080000 ack1382726993
```

Figura 8: TCPDump simplificado dos números de seqüência usados por Mitnick

Podemos observar na Figura 7 (números em vermelho negrito) que o valor cresce de 128000 em 128000. Então podemos fazer o ataque de predição.

O ataque em si aconteceu usando um TCP SYN forjado (inicio de conexão), vindo teoricamente de server.login em direção à x-terminal.shell. O x-terminal entende que ele confia em server, então server deixa entrar sem pedir senha.

X-terminal responde como SYN-ACK para server (Que neste momento está fora do ar devido ao DoS). Mitnick só precisa enviar um "ACK" para completar o ataque. Se o ACK da conexão anterior já é conhecido, apenas somamos 128000, e este é o resultado do ACK desejado.

```
14:18:36.245045 server.login > x-terminal.shell: S 1382727010:1382727010(0) win 4096
14:18:36.755522 server.login > x-terminal.shell: . ack 2024384001 win 4096
```

Em uma conexão normal, os dois começariam a trafegar dados, entretanto Mitnick tem um endereço falso e não pode ouvir os retornos de x-terminal. Para tanto, ele vai tentar enviar os dados necessários para o ataque no mesmo pacote do ACK (3º. passo do *Three-way-handshaking*) como aparece a seguir:

```
14:18:37.265404 server.login > x-terminal.shell: P 0:2(2) ack 1 win 4096
14:18:37.775872 server.login > x-terminal.shell: P 2:7(5) ack 1 win 4096
14:18:38.287404 server.login > x-terminal.shell: P 7:32(25) ack 1 win 4096
```

O que corresponde ao comando:

```
14:18:37 server# rsh x-terminal "echo + + >>/.rhosts"
```

Este comando diz que Qualquer host pode se logar no x-terminal sem digitar senha (como se x-terminal confiasse em todo mundo) [TOT 95]

Agora Mitnick acessa o x-terminal normalmente, através de ferramentas normais como o rsh. Ele entrou, instalou um cavalo-de-tróia, apagou seus rastros e saiu.

Alguns minutos depois, o computador de Shimomura voltou ao normal, após o ataque de DoS efetuado e assim aconteceu o ataque.

O importante desta história, é saber que um ataque legítimo de um *hacker* é normalmente complicado, demanda tempo e estudos para fazer uma coisa simples (neste caso digitar o comando “echo ++ >> ./rhosts”). Na verdade o ataque serve para “criar” a porta do fundo (*backdoor*), por onde o *hacker* realmente fará seu ataque.

Capítulo 5 – Estratégias de Defesas para Segurança

5.1 – Porque é difícil se proteger?

Proteção é algo difícil por diversos motivos que já citamos neste documento. Seja ele por falta de capacidade técnica dos profissionais envolvidos, por falta de recursos financeiros devido dificuldade de explicar aos diretores que se deve investir nesta área, ou ainda pela falta de entendimento do valor da informação.

Adicionado a estas difíceis barreiras, ainda temos nossas redes aumentando em complexidade, conectadas 24 horas por dia à Internet e às redes dos fornecedores, clientes e parceiros.

5.1.1 – Ambientes Cooperativos

A necessidade de estarmos constantemente conectados e com informações *on-line*, precisamos pensar na complexidade das redes de computadores em ambientes colaborativos.

É importante entender que em ambientes colaborativos, onde mais de uma organização está conectada na mesma estrutura colaborando com dados, ou usufruindo de dados de outra organização. O aumento do número de conexões (físicas através de LPs o Frame Relays ou lógicas através de VPNs e RAS) faz com que a complexidade da rede cresça.

Um exemplo seria de um e-commerce que acessamos via Internet e que tenha a matriz e mais 5 ou 10 filiais, conexões com seus fornecedores para consulta de estoques e pedidos, e as companhias de entrega.

Este cenário mental está longe de um ambiente altamente complexo ainda, porque temos que levar em conta que além destas conexões da matriz com as filiais, teremos conexões chegando às filiais, como por exemplo distribuidores ou parceiros. Limitar não só o acesso das filiais, mas também das conexões "terceirizadas" das filiais pode ser uma tarefa bastante complicada. Como abordar a segurança da informação com tantas conexões entrando e saindo da rede?

Entender as implicações de segurança para um ambiente cooperativo, onde recursos são disponibilizados *on-line* para todos, deve-se ter um estudo prévio. Normalmente as redes não começam neste nível de complexidade. Em geral, tudo começa com a matriz e algumas filiais.

Devemos então, analisar as implicações à cada nova conexão estabelecida ou cada novo serviço disponibilizado, entretanto é possível que além de tudo haja implicações políticas.

Além dessa conexão com a filial permitir a autenticação, um atacante que invadir a filial, poderá atacar a matriz. Muitos criam VPN (*Virtual Private Network*) pensando que está é a solução para os problemas, e em diversas vezes o servidor de VPN fica mais perto da rede interna do que o firewall. Se o ataque chegar por dentro da conexão de VPN, o firewall não poderá evitar e o atacante poderá ter mais chances de sucesso.

Outro ponto a ser observado é que a defesa é muito mais trabalhosa que o ataque. O *hacker* só precisa de um ponto de falha para que o ataque seja bem sucedido, e se a técnica de ataque não funcionar, ele pode usar outra ou procurar um outro ponto de falha, até que um seja encontrado. O profissional de segurança deve pensar em todos

os pontos de falha e se somente uma das defesas não funcionar, esta já é suficiente para que a rede inteira possa ser colocada em risco. [LIC 03]

5.1.2 – Internet

Hoje, todas as empresas de tecnologia possuem acesso ininterrupto à Internet.

Em comparação com a Internet na década de 90, onde utilizávamos modem, tínhamos uma conexão com um período restrito e a cada nova conexão obtínhamos um novo endereço de IP. Naquela época, os ataques eram menos diversificados e em menor número. Segundo [CER 05] foram reportados pouco mais de 3.000 ataques em 1999, enquanto em 2004 foram 75.700 ataques reportados.

As ferramentas de ataque estão ficando mais poderosas, conforme podemos ver no gráfico abaixo. [NAK 02]

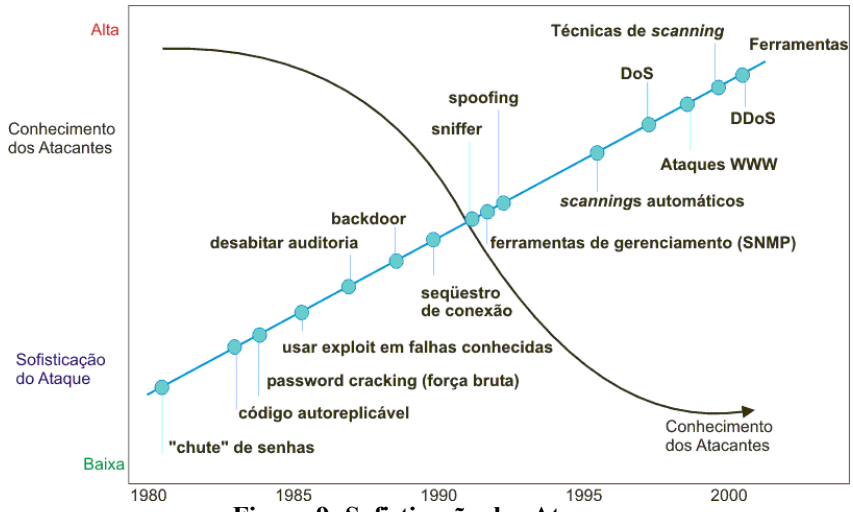


Figura 9: Sofisticação dos Ataques

Podemos ver também que conforme o tempo passa, os *hackers* desenvolvem ferramentas mais complexas, e a segurança se torna mais completa também, além da complexidade das redes estar se tornando maior.

Devemos entender que novas tecnologias trazem novas vulnerabilidades e que novas formas de ataques são criadas para burlar as técnicas de defesas. Conforme ataques são criados, os profissionais de segurança sempre conseguem achar dispositivos para o bloqueio do mesmo, que por sua vez fazem os *hackers* conseguirem novas formas de ataque.

5.1.3 – Redes Públicas, Privadas e DMZ

Redes públicas são aquelas diretamente conectadas à Internet, onde os servidores possuem endereços IP válidos. Entendemos por IP válido aqueles que não estão nas classes de IP atribuídos por IANA (*Internet Assigned Number Authority*) para o uso em redes privadas: 10.0.0.0 à 10.255.255.255, 172.16.0.0 à 172.31.255.255 e 192.168.0.0 à 192.168.255.255. [RFC 1597].

Mas redes privadas, estão associados a endereços IPs que não podem ser alcançados pela Internet. A principal função é economizar endereços IPs, fazendo com que vários clientes tenham endereços válidos acessando a Internet através de um único IP válido. Este procedimento é comumente chamado de NAT (*Network Address Translation*).

Entretanto, o NAT é uma tarefa de tradução de endereços de IPs inválidos para válidos e vice-versa. É possível que um administrador de rede coloque seus servidores em endereços inválidos e coloque um firewall traduzindo o endereço público para acessar o servidor que usa um endereço privado.

Enquanto muitos administradores acreditam que isto traz um índice adicional de segurança, este procedimento não traz uma proteção adicional, apenas aumenta a complexidade do firewall.

Durante este procedimento de NAT reverso, a qual o administrador de rede faz uma tradução entre um endereço IP válido e uma porta redirecionado para um IP inválido e um outro número de porta, é feito de modo estático, ou seja, definido manualmente. Se um ataque chegar para uma porta ao qual o NAT reverso foi feito, o mesmo será integralmente passado para o host na porção de endereços IPs inválidos da rede, pois o NAT não destrói o pacote e cria novamente (como um proxy), apenas troca as informações de endereço IP e porta de destino.

Para entendermos um pouco melhor o uso de NAT reverso, temos que estar basicamente familiarizado com o conceito de um firewall básico que utiliza-se destas técnicas. Um firewall que faz tradução de um endereço público para um privado normalmente precisa estar entre três redes: Internet, Interna e DMZ.

- A rede interna é onde estão os recursos que não desejamos compartilhar. Tipicamente incluem-se nesta rede as estações de trabalho, impressoras, servidores de banco de dados (os que não serão usados por um servidor web), servidores de arquivos, etc.
- A rede internet, que possui os recursos que desejamos acessar (websites, e-mails, etc.) e também estão os clientes ao quais queremos que tenham acesso à nossa rede pública, como por exemplo, nosso site de comércio eletrônico.
- A rede desmilitarizada ou DMZ (*DeMilitarized Zone*) que é um pedaço da rede a qual colocamos os recursos que desejamos compartilhar tais como Web Servers, Servidores de E-mail, DNS, etc.

No gráfico abaixo podemos ver claramente a divisão das sub redes em uma organização. [BEN 05]

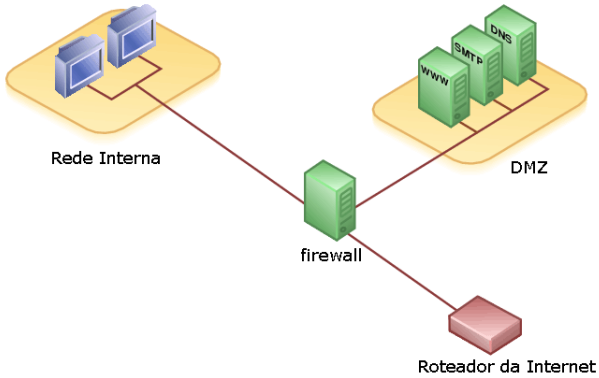


Figura 10: Firewall em uma rede DMZ

A situação de serviços públicos (tais como e-mail, servidor web, etc.) em redes com endereços privados que não estão em uma DMZ (*DeMilitarized Zone*) – e que por erro no planejamento estão na rede interna – acabam por gerar uma falha adicional de segurança de rede, pois se um *hacker* conseguir efetuar um ataque que eleve seus privilégios (*exploits, buffer overflow, etc.*) ele já está em nossa rede interna.

Por isso uma DMZ bem construída é um índice ou fator adicional de segurança na rede, pois se a rede DMZ for comprometida, o *hacker* não pode acessar a rede interna a partir da Internet ou da DMZ, pois um firewall não deve permitir conexões entrantes na rede interna. Porém, se a DMZ possuir IPs públicos ou privados, isto não deve ter implicações adicionais de segurança na rede. Vale lembrar que o endereço IP da DMZ deverá ser de classe diferente da rede interna, dificultando algum tipo de ataque adicional.

Concluindo que o uso de DMZ é altamente recomendada e o uso de redes privadas também em organizações que desejam estar publicando informações na Internet. Na rede privada podemos ter um tipo de segurança personalizada que permite maior funcionalidade e produtividade, já que o tráfego vindo da Internet deve ser filtrado e o tráfego intra-rede pode ser liberado. O uso de endereços IPs públicos ou privados na DMZ não traz índice adicional de segurança à este ambiente, mas traz um índice adicional de complexidade.

5.1.4 – Custos Decorrentes da Proteção

Proteger-se contra invasores invariavelmente possui um custo associado. Seja ele de um equipamento fornecido por um fabricante de firewalls em um *appliance*[14] com fácil configuração ou o custo de mão de obra de um profissional qualificado e um servidor com sistema operacional *open source*[15].

O custo está intrinsecamente associado ao nível de proteção. Uma boa proteção provavelmente demanda um custo mais alto, pois temos que levar em conta diversos aspectos incluindo equipamentos de rede como *firewall*, segurança física, auditoria, antivírus, treinamento, mão-de-obra especializada, pesquisa, etc.

Temos em mente que a segurança não deve se manter apenas na esfera tecnológica e que existe muito mais falhas do que as que podem estar dentro do computador.

A segurança implantada também deve ser constantemente revista. Novas vulnerabilidades estão todos os dias sendo descobertas, e uma só falha não atualizada e explorada por um atacante já é o suficiente para colocarmos tudo a perder.

5.1.5 – Segurança versus Funcionalidade

Antigamente as organizações implementavam suas redes apenas com o objetivo de prover funcionalidades e aumentar a produtividade da organização, a preocupação naquela época com a segurança era menor em virtude do menor número de incidentes de segurança e ao acesso limitado da rede ao mundo exterior.

Hoje em dia, a importância da segurança cresce mais rapidamente em ambientes complexos como os vistos hoje em dia. Um ponto fundamental é que a segurança é inversamente proporcional à funcionalidade, ou seja, quanto maior a segurança, menor será o índice de funcionalidade de recursos, serviços, aplicativos e demais facilidades.

Quanto mais funcionalidades, maior será o número de aplicativos ou maior a complexidade dos códigos dos aplicativos envolvidos, que por conseqüência cria um

[14] Appliance é um nome informal dado a equipamentos de rede prontos com uma aplicação especifica, tal como firewalls ou roteadores.
[15] Open Source é uma licença de uso de software livre, em essência, define critérios de utilização e distribuição que incluem a livre redistribuição e uso sem custo.

maior número de bugs de segurança que culmina com o aumento na responsabilidade da equipe de segurança envolvida no processo. [LIC 03]

Portanto precisamos refletir que quanto mais restrito for o ambiente e menos recursos não necessários forem disponibilizados para o usuário, menor o índice de ataque, porém a cada nova necessidade, o usuário terá que procurar o administrador e pedir uma autorização para instalação ou acesso à um novo recurso. Difícil é a tarefa do administrador que precisa de bom senso para dar os privilégios e recursos necessários para que tenhamos a funcionalidade requerida pelos usuários, e por outro lado restringir os potenciais problemas que possam ser causados com isso, além de combater o lado político normalmente como ator o diretor, que particularmente não gosta de usar senhas difíceis, quer acesso completo ao sistema (mesmo sem precisar ou sem saber utilizar), não está a par dos riscos de segurança que seu perfil de usuário provê, usa normalmente computadores do tipo *notebook* que pode ser roubado fazendo dele um risco iminente ao aumento de incidentes de segurança na organização.

5.2 – Métodos de Defesa

Portanto estivemos discorrendo neste trabalho os tipos de ataques, vulnerabilidades e conceitos que precisamos para compreender o quão complicado estão os ambientes cooperativos de rede em questão a segurança da informação.

Pudemos ver que muitas vezes é a falha humana que é explorada e muitas vezes nem mesmo o computador é necessário para tal. Veremos agora as técnicas que poderemos utilizar para diminuir, e talvez eliminar, os riscos de segurança em uma empresa.

5.2.1 – Firewall

Firewall é o componente mais conhecido de segurança em uma rede ou computador. Muitos até acreditam que o *firewall* é algo complicado e que só pode ser implantado por grandes organizações que possuem milhares de dólares para investir nesta tecnologia avançada, porém enquanto parte disso é verdade, temos que entender que o *firewall* é nada mais do que um filtro, que bloqueia os pacotes de redes definidos pelo administrador e que pode e deve ser usado por todos que estejam conectados a ambientes hostis como a Internet.

5.2.1.1 – Definição

Existem várias definições relacionadas a o que é um *firewall*, vejamos algumas:

- Um sistema desenhado para prevenir acesso não autorizado de ou para a rede privada. *Firewalls* podem ser implementados em hardware ou software ou uma combinação de ambos. Estes sistemas são frequentemente usados para evitar acesso de uma rede pública, como a Internet, para uma rede privada, especialmente as Intranets. [ANS 05-2];
- Um ponto entre duas ou mais redes, no qual circula todo o tráfego. A partir deste ponto é possível controlar e autenticar o tráfego, além de registrar por meio de logs todo o tráfego da rede facilitando sua auditoria [LIC 03];
- Firewall é o principal método para manter um computador seguro de intrusos. Um *firewall* permite ou bloqueia tráfego para dentro ou fora da rede privada ou do computador do usuário. São globalmente usados para dar acesso seguro aos usuários para a Internet e também separar a rede pública

da companhia da rede privada. Eles também podem ser usados para manter segura uma sub rede; como por exemplo, a rede de contabilidade deve estar protegida de *eavesdropping*.

A arquitetura do *Firewall* normalmente utiliza componentes de roteadores escrutinadores (*screening router*), filtros de pacotes com estado ou sem estado (*statefull* e *stateless*), NAT, DMZ, proxy.

Entende-se por roteador escrutinador como um roteador de rede capaz de distinguir tráfego da rede baseado em tipos protocolos, valores dos cabeçalhos dos pacotes ou segmentos que por ele passam (camadas rede, transporte e eventualmente aplicação e física) [ATI 00] ou análise do estado da comunicação (*statefull analysis*).

5.2.1.2 – Componentes

Um *Firewall* pode desempenhar diversas funções, sendo que cada componente desempenha uma tarefa específica. Normalmente algumas dessas funcionalidades são combinadas de forma a fazê-lo mais seguro. Lembrando que devemos analisar o impacto de cada novo componente a ser adicionado ao *Firewall*, pois nem sempre devemos ter todos em um único local.

Algumas das tarefas de um *Firewall*:

- Filtro de Pacotes: Os filtros são regras criadas por um administrador. Estes filtros elegem quais pacotes serão aceitos para entrar ou sair da rede por meio de análise dos cabeçalhos dos protocolos de rede (camadas 2 à 4 tipicamente);
- Análise de Estado: Quando uma conexão é estabelecida, esta pode ser acompanhada pelo *Firewall* de modo que possa analisar o estado da conexão (abrindo, conectado, fechada, ouvindo, etc.). Este estado é usado para complementar a ação dos filtros de pacotes que utilizam a informação da análise de estado para construir regras mais eficientes de filtragem;
- NAT: Os pacotes que possuem endereços públicos podem ser traduzidos e encaminhados para a rede DMZ ou pedidos da rede interna com endereço privado pode ser traduzida para uma requisição da rede pública (Internet);
- Zona Desmilitarizada (DMZ ou *DeMilitarized Zone*) é o pedaço da rede a qual colocamos os recursos (servidores *bastionizados*[16]) que desejamos compartilhar com a rede pública;
- IDS (*Intrusion Detection System*) é um recurso de análise de conexões (camadas 2 à 7) onde utiliza-se de assinaturas de ataques para posterior análise no processo de auditoria. Quando um ataque é detectado por um IDS, este faz um *log* e guarda as informações da conexão para verificação pela equipe de segurança;
- Redes Privadas Virtuais (VPN ou *Virtual Private Network*) são conexões entre duas redes de forma privada dentro de um túnel virtual cifrado. A VPN deve permitir que duas redes remotas possam estabelecer uma conexão entre si, de forma que os usuários acreditem que estão conectados diretamente a ela através de um link dedicado;
- Autenticação é o ato de garantir a veracidade da identidade dos clientes remotos e muito usado para prover um acesso especial à um determinado local da rede. A autenticação é vital em casos de VPNs onde as conexões se

[16] Equipamentos que passaram pelo processo de Hardening (seção 5.2.4)

formem dinamicamente, como por exemplo um diretor remoto acessando a rede interna da empresa

- Balanceamento de carga é o ato de que quando haja uma sobrecarga no *Firewall* possamos ativar alguma forma de contingência de forma que o tráfego excedente possa ser desviado para outro *Firewall*. Se houver dois ou mais *Firewalls* balanceados, todos devem utilizar a mesma política de filtragem para que não haja falhas nas regras.

Sendo o filtro de pacotes como a tarefa mais importante do *Firewall*, veremos a seguir em detalhes o processo utilizado na definição das regras.

5.1.2.3 – Filtro de Pacotes em Firewalls

O Firewall tem como a principal tarefa filtrar pacotes. Em uma situação típica, o firewall ou sistema deve ter duas ou mais interfaces de rede. Sendo ele o único caminho de interligação entre elas, podemos aceitar ou rejeitar pacotes baseadas em regras.

No início, os filtros só eram simples e apenas observavam conexões baseadas em protocolos de rede ou transporte. Basicamente as opções possíveis em um pacote:

- Protocolo (ICMP, UDP, TCP, etc.);
- Endereço IP de origem;
- Endereço IP de destino;
- Porta de origem;
- Porta de destino.
- Ação (passar ou descartar)

A primeira vista este cenário pode ser completo, mas não evita a maioria dos problemas. Vejamos uma simples conexão entre o host A (origem) e o host qualquer na porta 80 usando o protocolo TCP. Sabemos que o TCP utiliza a porta de origem não-privilegiada (acima de 1024). Portanto escreveríamos uma regra para o envio e outro para o recebimento (Tabela I):

	Origem		Destino		
Protocolo	IP	Porta	IP	Porta	Ação
TCP	Host A	> 1024	qualquer	80	permitir
TCP	qualquer	80	Host A	>1024	permitir

Tabela 1: Exemplo de uma regra de firewall simples

Se houvesse apenas a primeira regra esta permitiria o pacote sair, mas não entrar. Por este motivo, precisamos da regra da linha dois. Porém, esta simples regra aparentemente inofensiva daria uma margem para um ataque pois o atacante poderia encaminhar um pacote com a porta de origem 80 com o destino da porta acima de 1024 no host A. Vale lembrar que diversas aplicações usam portas acima de 1024 para receber conexões, tais como Banco de dados SQL Server (TCP/1433), Emulador de terminal VNC (TCP/5800 e 5900), Servidores Proxy (TCP/8080 ou TCP/3128), etc. Todas as aplicações citadas acima estariam vulneráveis a ataques se o atacante usasse a porta de origem 80. Como já dissemos anteriormente, os *hackers* burlam as regras básicas (como iniciar uma conexão com porta origem acima de 1024) para se beneficiar de uma falha.

Vemos que o atacante então, usaria a regra 2 para entrar e a regra 1 para sair da rede.

Em *firewalls* modernos, a filtragem é feita baseada no estado e na direção do início da conexão em protocolo TCP. A filtragem das conexões UDP e ICMP feita pelo *firewall* é um pouco diferente do TCP, pois no UDP não existe o termo conexão nem os flags de conexão que existem no TCP. No ICMP, a filtragem é feita com base nos tipos e códigos das mensagens.

Resumidamente, o filtro de estado é uma tabela armazenada pelo *firewall* do estado de cada conexão que está passando por ele em um espaço de tempo. Imagine que o Host A quer falar com o Host B. Como o Host A inicia a conexão enviando um "TCP SYN" para B, o *firewall* cria uma regra temporária dizendo que foi estabelecida uma conexão entre A e B usando as portas e protocolos especificados no pedido de conexão (tabela 2).

	Origem		Destino		
Protocolo	IP	Porta	IP	Porta	Ação
TCP	Host A	1026	Host B	80	permitir
TCP	Host B	80	Host A	1026	permitir

Tabela 2: Exemplo de regras de firewall

As regras acima foram "criadas" dinamicamente e permitem que o filtro esteja estabelecido com segurança contra a falha citada acima.

Este é um cenário simples, onde as conexões são TCP e criadas de maneira previsível. Filtros de estado devem também ser capazes de entender certos protocolos e abrir ou fechar portas dinamicamente conforme sua necessidade.

Um típico e fácil exemplo clássico é o FTP. O FTP (*File Transfer Protocol*) utiliza duas portas. Uma é a porta TCP/21 e a outra é criada dinamicamente. O FTP pode ser ativo ou passivo. Em ambos, a porta TCP/21 é usada para trafegar o controle (ou comandos) e a porta adicional é utilizada para transferência dos dados. Em modo ativo, a transferência é feita pela porta TCP/20. No modo passivo, o cliente abre uma porta (LISTEN) e envia um comando (PASV) pela porta de controle (21), informando que o servidor deve abrir uma conexão para o cliente na porta especificada no comando. O *firewall* deve abrir uma porta ao ouvir o comando (PASV) passando pela porta de controle, de modo que o servidor tenha direito a acessar o cliente pelo tempo determinado da transferência.

Os *firewalls* podem também filtrar baseado em:

- Protocolo da Conexão;
- Endereço IP Origem;
- Porta de Origem (UDP, TCP);
- Endereço IP Destino;
- Porta de Destino (UDP, TCP);
- Tipo de Mensagem e Código da Mensagem (ICMP);
- Estado da Conexão (SYN_SENT, SYN_RECV, ESTABLISHED, FIN_WAIT, CLOSE_WAIT, etc).;
- Direção da conexão (in/out);
- Interface de origem da conexão;
- Endereços físicos (*MAC Address*);
- Flags do TCP (SYN, ACK, FIN, RST, etc.).

Estes filtros baseados em estado são conhecidos também por *statefull packet filter*. Nestes casos, o *firewall* trabalha somente verificando o primeiro pacote de cada

conexão de acordo com as regras definidas pela equipe de segurança. Se o primeiro pacote de uma conexão TCP (SYN) é bloqueado e os demais eventualmente forem configurados para serem aceitos, o servidor se defende sozinho, pois não adianta nada enviar dados para ele sem uma conexão estar previamente estabelecida.

Se um filtro de estado detectar um pacote que estiver aceito em uma sessão já estabelecida, o mesmo será repassado, porém se os pacotes não fizerem parte de nenhuma das conexões atuais estabelecidas e não estiverem contidas na tabela de estados o pacote é descartado.

Entretanto, dependendo da forma como a conexão é bloqueada (descarte simples, TCP RST, *ICMP Port Unreachable*) ou aceitada, um atacante pode fazer um *version determination* baseada nas características (assinatura) destes pacotes.

O filtro mais conhecido para *firewall* é o IPTables, desenvolvimento em open-source e que está incluído na maioria das distribuições do linux. Este *firewall* usa a técnica padrão, onde o primeira regra em que condizer com as características do pacote que estiver passando, executa-se o ato desejado (aceitar ou rejeitar).

Um filtro simples implementado com o IPTables pode ser visto no exemplo abaixo: [NET 05]

```
## Carregando módulos de acompanhamento de conexões
insmod ip_conntrack
insmod ip_conntrack_ftp

# cria regra de firewall para forward (Pacotes que estão passando
# pelo firewall) em direção à outro local
# linhas 1,2 e 3
iptables –A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables –A FORWARD -m state --state NEW -i ! eth0 -j ACCEPT
iptables –A FORWARD –j DROP

# cria regra de firewall para INPUT (pacotes que serão processados
# pelo firewall como um host final)
iptables –A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables –A INPUT -m state --state NEW -i ! eth0 -j ACCEPT
iptables –A INPUT –j DROP
```

Neste caso, a primeira linha define que se o estado for “ESTABLISHED” (já estabelecido anteriormente) ou “RELATED” (Relacionado com o tráfego já estabelecido – como por exemplo FTP ativo) tem como permissão ACEITAR.

A segunda linha faz o controle de novas conexões (NEW). Neste caso, só é aceito conexões novas se não vier pela interface de rede eth0, que em nosso exemplo está conectada à internet. Se o pacote chegar pela interface eth1 vindo da rede interna, ele será aceito e o retorno da Internet em direção ao cliente será garantido pela primeira regra, pois a conexão já estará estabelecida.

Na ultima linha, todos os pacotes que não estiverem de acordo com outras regras serão barrados.

Por fim, para concluirmos, precisamos ter em mente que um *firewall* deve ser inexpugnável. No entendimento deste trabalho esta expressão refere-se a aquele que não pode ser comprometido. Um *firewall*, por exemplo, que é um filtro de pacotes com estado (*statefull*), contemplando um IDS e NAT, que não rode nenhum serviço, não pode ser atacado, porque não possui aplicativos para ser explorado. O único ataque teórico possível seria um DoS através de *flooding*, entretanto, devido a natureza de quão

leve são os filtros de um *firewall* é improvável que a organização possua banda suficiente para um ataque deste porte.

Se o *firewall* for comprometido, não há nenhuma saída técnica para proteção da rede, pois normalmente ele é que controla o tráfego do que entra e que sai da rede. O *hacker* de posse de um *firewall* pode desabilitar as regras, aumento de privilégios, *sniffing* de conexão, ataques a VPNs, DoS, etc.

5.1.2.4 – Arquiteturas de Firewalls

A arquitetura do *firewall* deve estar de acordo com a topologia da rede da organização. São quatro as arquiteturas existentes: [LIC 03]

- Dual Homed Host
- Screened Host
- Screened Subnet
- Firewall Cooperativo

Um *firewall dual-homed* é um servidor que possui duas interfaces de rede, cada uma conectada a uma rede separada. Para que clientes de uma rede se utilizem de recursos de outras, as comunicações devem ser estabelecidas em duas etapas: uma conexão até o *firewall* e outra até o destino final.

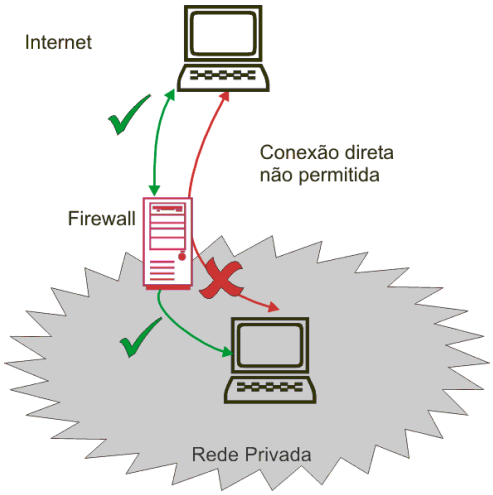


Figura 11: Um Firewall Dual-Homed

Um exemplo seria em uma aplicação de terminal remoto, tais como SSH ou telnet. Se eu desejasse da Internet me conectar ao servidor da minha empresa, eu deveria, neste ambiente, conectar-me no proxy através de uma conexão telnet, por exemplo, e aí após autenticado no servidor, fazer um novo telnet até o servidor destino. Hoje em dia isto é um índice adicional de segurança, e é possível que façamos uma conexão sem a necessidade de NAT reverso, além de requerer duas autenticações separadamente.

Um *firewall* da arquitetura *Screenet Host* é formado por um roteador com filtro de pacote e um *bastion host*. O filtro deve ter regras que permita apenas acesso através do *bastion host* e não diretamente de uma rede para outra. Um típico caso deste item são os servidores proxy em uma rede com acesso a Internet. Neste caso os clientes devem habilitar a opção de uso de servidor proxy (*bastion host*) que controlará o acesso a Internet. Se o usuário desejar acessar Internet sem passar pelo *proxy*, o acesso será bloqueado.

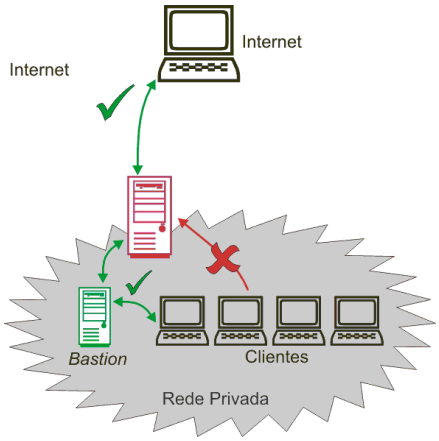


Figura 12: Um Firewall da arquitetura Screened Host

A maior falha desta arquitetura é que se o *hacker* entrar no *bastion host*, este já estará dentro da rede privada. Uma saída é o uso de uma DMZ através da arquitetura *Screened Subnet*.

Em uma arquitetura *Screened subnet*, resolvemos o problema citado isolando o servidor ou bastion em uma rede especifica. Neste caso, a sub rede que contém os serviços que devem possuir acesso à rede pública. Se um *hacker* entrar no *bastion host*, este ainda terá mais uma barreira até chegar à rede interna.

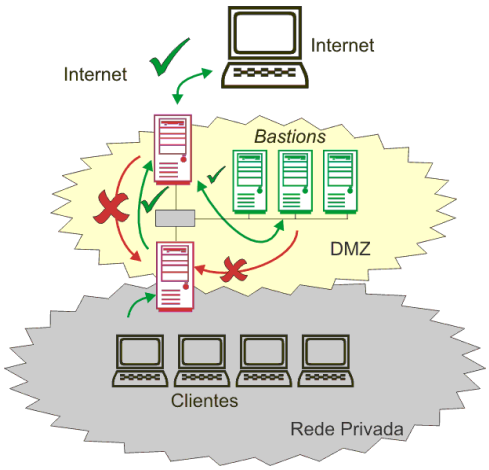


Figura 13: Um Firewall da arquitetura Screened Subnet

Nesta ilustração podemos verificar que nenhuma conexão da rede pública ou da DMZ pode entrar na rede interna. A rede pública pode apenas acessar a DMZ e a rede interna e DMZ podem acessar a rede pública.

Existe uma outra arquitetura que usa um único firewall com três interfaces de rede para esta divisão e que pode ser usado sem problemas. Deve-se ter em mente que o uso de um *firewall* com três interfaces (pública, privada e DMZ) traz o nível de segurança igual ao esquema com dois *firewalls*, porém aumenta a complexidade das regras a serem criadas.

O *firewall* colaborativo é uma arquitetura onde são inseridos novos componentes, como a VPN (*Virtual Private Network*), o IDS (*Intrusion Detection System*) e a PKI (*Public Key Infrastructure*) [LIC 03]. O firewall colaborativo deve separar e filtrar as comunicações entre as conexões entre o mundo externo (Internet), parceiros, conexões remotas (VPNs e RAS) a rede desmilitarizada e a rede interna.

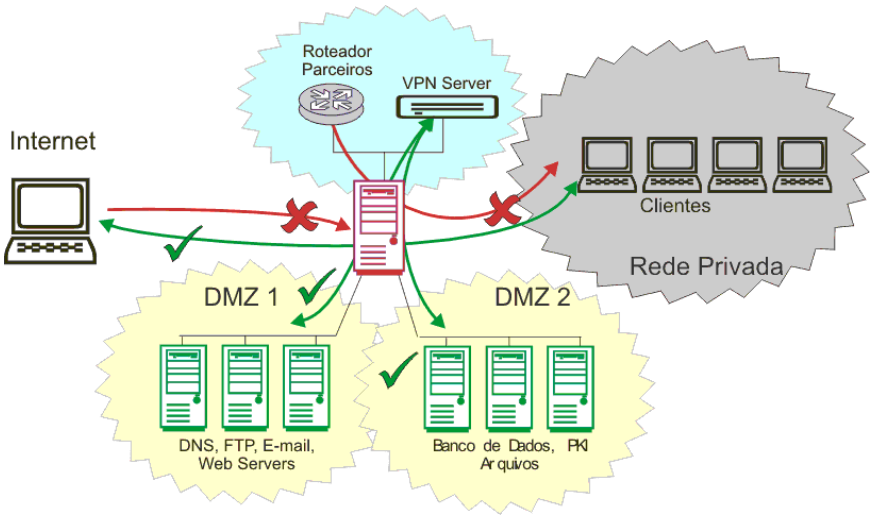


Figura 14: Firewall em uma arquitetura colaborativa

5.2.2 – Segurança Física

Segurança física é bastante fácil de ser entendido e já foi rapidamente explanado neste documento. Como o próprio nome pode descrever, a segurança física tem a ver com a proteção do hardware da rede contra supostos atacantes ou desastres naturais. Segurança não esta relacionado apenas à proteção contra ataques, mas nos dias de hoje, este assunto está ligado a continuidade das operações em que se diz respeito a proteção.

Pode-se rapidamente perceber que de nada adianta um servidor com uma ótima regra de *firewall*, softwares atualizados, pessoal treinado e demais itens, se o atacante puder entrar na organização e roubar o servidor que contém os dados que deveriam estar protegidos ou ainda acesso indevido através do *console*. Neste exemplo roubar tem a ver com a idéia de furto simples, furto qualificado ou roubo levando o *hardware* que contém os dados.

Porém, este tipo de incidente é relativamente raro, mas deve ser previsto em uma organização.

Os servidores devem estar fisicamente protegidos, em sala exclusiva com portas corta-fogo, sistema de incêndio, detectores de movimento (alarme), equipe de segurança (guardas) e monitoração por CFTV. O acesso à sala dos servidores deve ser restrito apenas ao extremamente necessário e apenas para pessoal devidamente autorizado. Deve-se também ter cuidado na entrada de mais de uma pessoa que não faz parte do quadro de funcionários (como um suposto técnico da operadora de telefonia que pode ser o atacante), pois estes podem estar coagindo o funcionário ou levando algum dispositivo ilícito, como por exemplo, um explosivo dentro da maleta de ferramentas ou dentro de um dispositivo de rede (switch, hub, roteadores, etc). [CAOR 05]

Portanto, a regra neste caso é que nunca devemos confiar nas pessoas, e devemos estar um passo a diante na proteção, pois desconhecemos o que um meliante pode fazer se estiver disposto ao ataque.

5.2.3 – Antivírus, AntiSpywares

Vírus, worms, spywares ou cavalos de tróia podem ser definidos como programas que são instalados em computadores, sem a autorização ou consciência do usuário, a fim de explorar recursos, roubar dados e prejudicar o computador.

Entretanto cada um deles tem uma característica própria:

- Vírus: São códigos incluídos em programas legítimos que atuam em tempo de execução. Estes códigos são pequenos programas que são criados deliberadamente para interferir com a operação de um computador corrompendo, alterando, deletando dados ou deixando outros processos lentos. Conceitualmente, os vírus espalham-se com a ajuda humana, transportando arquivos entre locais diferentes. Alguns deles se proliferam sozinhos dentro do próprio computador. Existem Vírus de Setor de Boot, Vírus de Arquivos Executáveis, Vírus de Macro e Vírus de Script.
- Worms: (Vermes): São como vírus, mas tem a capacidade de espalharem-se automaticamente entre sistemas utilizando vulnerabilidades conhecidas em produtos, tais como o Windows. Eles podem, por exemplo, se proliferar por e-mail ou conectando-se em portas que possuem programas susceptíveis à proliferação através de uma falha de segurança do aplicativo servidor.
- Cavalo de Tróia: É um programa aparentemente benigno mas que se comportam de maneira inesperada. Podem estar neste grupo os programas espiões (que coletam informações digitadas, tais como senhas, números de cartões de crédito, etc.).

5.2.3.1 – Detectando vírus

Muitos usuários e administradores podem ter dúvidas se um vírus está sendo alvo de uma onda de vírus em seu computador. Felizmente hoje em dia, a informação sobre os vírus está se proliferando entre os usuários que aparentemente estão tomando um pouco mais de cuidado.

Alguns sinais podem indicar que seu sistema pode estar infectado: [MIC 05-2]

- Roda constantemente mais lento que o normal;
- Pára de responder ou trava constantemente;
- Reinicializa regularmente sem sua requisição;
- Aplicações não funcionam de maneira apropriada;
- Arquivos ou discos ficam inacessíveis ou corrompidos regularmente;
- Sua conexão da Internet está com alto tráfego, mesmo sem que aplicativos estejam executando;
- Vê mensagens de erros estranhas;
- Recebe retornos de e-mails com erro, dos quais não enviou.

Claro que alguns desses sinais podem estar indicando problemas de software, hardware ou *network*, porém estes podem ter algum fundamento. Para tanto, aplicativos antivírus são freqüentemente instalados para detectar um vírus do computador antes que este possa causar algum prejuízo maior.

A principal forma de proliferação dos vírus atualmente é através de e-mail ou através de servidores de arquivos compartilhados.

5.2.3.2 – Spywares

Hoje em dia, o problema dos programas espiões ou *spywares* estão em alta. Os programas espiões podem também ser conhecidos como *adware*. Na prática, os usuários nem sequer sabem que uma nova onda de programas do tipo cavalo-de-tróia está na rede, ou então acreditam que os antivírus tradicionais ou firewall podem protegê-lo (apenas melhoram um pouco o índice de segurança). Atualmente existem programas que evitam *spywares*, exatamente como os *antivírus*, mas com uma finalidade específica.

De modo geral, *spyware* é um termo usado por um software para proceder com certos comportamentos como publicidade não desejada (*adware*), coletar informações pessoais, mudar parâmetros do computador o consentimento do usuário.

Alguns sinais podem indicar que seu sistema pode estar infectado por um spyware:

- Você vê janelas com publicidade (*popup*) quando não está navegando;
- A página inicial do seu browser foi alterada sem seu consentimento;
- Uma nova barra de ferramentas que você não instalou, ou que está com dificuldade para removê-la;
- Aplicativos de Internet mais lentos;
- Aumento no índice de travamentos no computador.

Esses softwares indesejados causarão um comportamento desagradável no seu computador que se não for tratado pode se agravar. Existem várias maneiras de este software espião chegar ao seu sistema, porém a mais usada é instalar este aplicativo através de programas de compartilhamento de música ou vídeo na Internet (*P2P Sharing*). Verifique antes de instalar qualquer aplicativo os contratos (*license agreement*), onde normalmente consta essa informação. O que ocorre com frequência é que o usuário aceita sem saber destes termos.

5.2.3.3 – Como evitar?

Para evitar vírus, uma boa política de segurança pode ajudar a reduzir o risco, pois proíbe o uso de programas nocivos ou acesso indevido à locais com maior possibilidade de vírus. Os Antivírus também são uma das mais eficazes soluções para combater este problema. Esses programas procuram por características (assinaturas) no computador para detectar, bloquear e remover o programa malicioso. Eles também podem remover Worms.

Os cavalos de tróia, com a proliferação da Internet, estão também aparecendo com outros nomes, tais como Spywares, Adwares, etc. Existem programas como o Microsoft Antispyware, Spybot e LavaADWare que podem remover estes programas maliciosos do computador.

5.2.4 – Hardening (Bastionização)

Hardening é um processo inicialmente usado na indústria metalúrgica para deixar os metais mais resistentes. [ANS 05-3]. Em um ambiente de segurança, temos a idéia de endurecimento da segurança, ou seja, a idéia é deixar um host mais seguro. Os servidores *bastion* normalmente passam por esse processo, pois desejamos a maior segurança possível. Em uma situação ideal, um host que sofreu um *Hardening* ("bastionização") seria capaz de se defender sozinho, sem o uso de um *firewall* entre ele e a Internet.

Tipicamente, o processo de *Hardening* depende de cada sistema em uma rede. Cada aplicativo deve ser meticulosamente revisado de modo que recursos desnecessários sejam desabilitados, alterações no sistema operacional são feitos, fechando portas inúteis, removendo contas de usuários desativadas, revendo os parâmetros de autenticação e confiança, aplicando ACL apropriada, etc.

Hoje em dia existem ferramentas de auditoria que nós dá um relatório detalhado do que pode ser alterado de modo a deixar o sistema mais seguro, entre eles o MBSA (*Microsoft Baseline Security Analyzer*), Bastille Linux, YASC Harden It, etc. Por exemplo o MBSA pode verificar vulnerabilidades como:

- detecção se o firewall está habilitado;
- programa de atualização de software automático está habilitada;
- não existem senhas "fracas";
- todas as contas "convidado" estão desabilitadas;
- versões de software e *patches* aplicados;
- detecção de serviços desnecessários rodando;
- possibilidade de exploração de aumento de privilégio;
- detecção de falhas na configuração dos aplicativos instalados

Vemos na Fig. 15 podemos ver a tela com parte do relatório oferecido pelo MBSA:

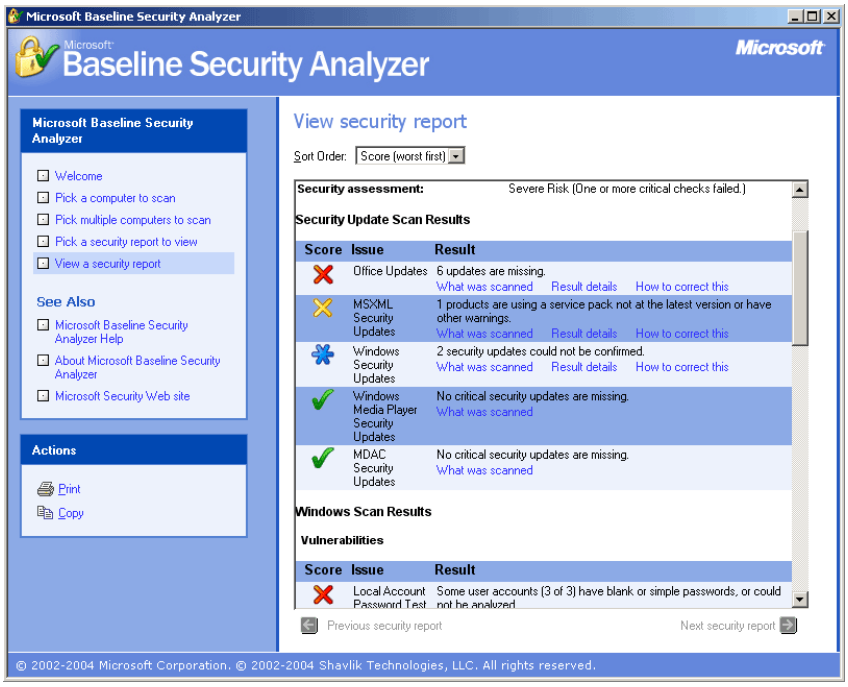


Figura 15: Tela do MBSA

E outro *screenshot* do software de *Hardening* do Bastille-Linux (Figura 12):

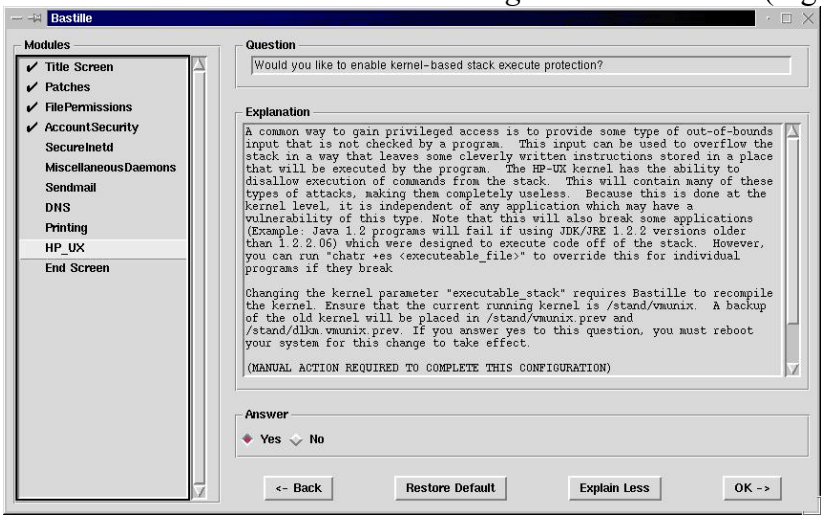


Figura 16: Tela do Bastille-linux

Portanto, a tarefa de Hardening não é fácil, e demanda altíssimo grau de conhecimento do administrador de rede ou da equipe de segurança. Deve-se estudar o ambiente para que seja feita a melhor proteção do sistema.

Em Windows, por exemplo, podemos prover alguns procedimentos básicos a serem feitos em um processo de Hardening em um servidor *bastion*: [MIC 05-4]

- Configurá-lo como servidor standalone;
- Configurar os direitos e políticas de cada usuário;
  - Acesso a arquivos e recursos;
  - Logon local;
  - Acesso ao computador pela rede;
- Configurar auditoria (Event Log);
- Desabilitar serviços desnecessários, incluindo:
  - Desabilitar o remote administration;
  - Remote registry service;
  - Server;
  - TCP/IP NetBIOS helper service;
  - Terminal services;
- Habilitar o *firewall*;
- Traçar um perfil de uso do servidor através do performance log;
- Remover *bindings* desnecessários entre protocolos e redes;
- Proteger as contas de usuários conhecidas (guest e administrator);
- Atualizar as versões de software;
- Usar o MSBA;
- Instalar antivírus e antiSpyware (se possível);

No ambiente Linux, podemos citar:

- Configurar os direitos dos usuários;
- Enviar o log (syslog) para outro servidor;
- Habilitar auditoria;
- Remover serviços desnecessários;

- Rever todos os parâmetros dos arquivos de configuração dos serviços que estão executando;
- Traçar um perfil de uso através do top ou snmpd;
- Remover usuários inúteis do /etc/shadow, /etc/passwd;
- Atualizar versões de software;
- Habilitar *firewall*;
- Fazer uma verificação dos aplicativos instalados, e verificar diariamente com aplicativos de *hash* (tais como tripwire);
- *ChRootar* serviços que possam comprometer a segurança (como servidores de e-mail, hosting compartilhado, etc.);
- Auditar log diariamente.

Existe um número infinito de itens a ser considerado em um *bastion* além dos citados acima, que requer um estudo mais profundo baseado em um ambiente específico.

5.2.5 – Criptografia

A partir do momento em que duas pessoas desejaram conversar em segredo, temos que pensar em uma maneira de escrever os dados de forma que outra pessoa não possa entender. A criptografia é a solução para este dilema.

Criptografia vem da palavra grega kryptos (escondida) e graphia (escrever). Criptografar significa transformar uma mensagem em outra (escondendo a mensagem original), usando para isso funções matemáticas de modo que seja impossível (ou quase) que um criptoanalista sem o conhecimento da chave consiga desvendar a mensagem e descobrir o texto original.

Decriptografar é o ato de pegar a mensagem cifrada e com o uso de uma senha (chave) ela possa ser revertida para o texto original [BRG 03-2]. Outra definição formal para criptografia é o estudo das técnicas matemáticas relacionadas com os aspectos de segurança da informação, tais como confidencialidade, integridade dos dados, autenticação de entidades e verificação da origem. [HAN 96]

Os algoritmos de criptografia são baseados nas dificuldades em resolver problemas difíceis. Criptoanalistas são pessoas que estudam como comprometer (ou desvendar) os mecanismos de criptografia, e a criptologia é o resultado do estudo da criptografia pelos criptoanalistas.

5.2.5.1 – Criptografia Básica

Existem dois tipos de criptografia, cada uma com características, vantagens e desvantagens: criptografia simétrica e criptografia assimétrica.

- Criptografia simétrica é basicamente a criptografia onde é usada apenas uma chave para cifrar e decifrar o texto. A criptografia simétrica em alguns locais também é referida como criptografia de chave secreta (*secret-key criptography*).
- Criptografia assimétrica é a criptografia onde é usada uma chave para ciframento e uma outra (diferente) para deciframento.

Algumas idéias devem ser levadas em conta. Criptografia assimétrica não é mais ou menos seguro que a criptografia simétrica. Cada uma tem seu uso específico. O que

define a segurança a não permitir a criptoanálise é o tamanho da chave. Quanto maior a chave, mais difícil é criptoanalisar o dado.

O que todos devem estar se perguntando é se existe a criptografia de chaves públicas porque ainda usar a criptografia simétrica? A criptografia assimétrica exige muito mais processamento do que a criptografia simétrica. Na vida real a criptografia assimétrica é usada para combinar uma chave que será usada posteriormente por uma criptografia simétrica, ou no caso de assinaturas digitais é feito um hash da mensagem e a criptografia acontece no hash para diminuir o *overhead*.

Uma breve comparação entre criptosistema de chave simétrica e assimétrica está descrita na Tabela 3. Este compara, de forma rápida, as características principais, confrontando-as.

	Criptosistema de Chave Simétrica	Criptosistema de Chave Assimétrica
Velocidade	Alta	Baixa
Confiabilidade	Boa	Muito Boa
Nível de Segurança	Alto	Alto
Requer uma Terceira Parte Confiável	Algumas vezes	Sempre
Quantidade de Chaves Usadas	Uma	Duas

Tabela 3: Quadro comparativo de algoritmos com chave simétrica e assimétrica

Para a área de segurança de redes, a criptografia é o que nos permite uma comunicação segura entre dois computadores, sem isso, estaríamos altamente susceptíveis a ataques de *sniffing* e teríamos problemas sérios com a autenticação dos usuários.

A assinatura digital é o procedimento que garante a identidade de um usuário remoto e é feita com ajuda da criptografia assimétrica e de algoritmos de *hashing*.

Alguns serviços que utilizam criptografia: VPN (IPSec ou L2TP), autenticação, assinatura digital, PKI (Public Key Infrastructure), SSL (Secure Socket Layer), SSH (Secure Shell), TLS (Transport Layer Security), entre outras.

5.2.5.2 – Algoritmos de Criptografia

Temos inúmeros tipos de algoritmos de criptografia. Vejamos alguns dos mais famosos na Tabela 4:

Tipo	Algoritmo	Descrição	Uso	Ano
Simétrico	DES / 3DES	Transposição de blocos de 64 bits com chaves de 56 ou 168	Cifragem	1970's
Simétrico	IDEA	Transposição e usa Chaves de 128 bits	Cifragem	1990's
Simétrico	RC5	Criado por RSA, bastante flexível	Cifragem	1994
Simétrico	AES	Criado para o governo americano. É um padrão de-facto	Cifragem	2001

(continua)

Assimétr.	Diffie-Hellman	Problema de logaritmos discretos	Distribuição de chaves	1976
Assimétr.	RSA – Rivest -Shamir-Adleman	baseado em fatoração de inteiros	Cifragem + Assinatura	1978
Assimétr.	DSA	Esquema de assinaturas baseado no problema do logaritmo discreto.	Assinatura	1991
Assimétr	Curvas Elípticas	Intratabilidade do problema do logaritmo discreto em grupos aritméticos definidos sobre os pontos de uma curva elíptica.	Cifragem + Assinatura	2000's

Tabela 4: Algoritmos de criptografia mais populares

5.2.6 – Autenticação

Autenticação é o ato de garantir que um cliente possa provar sua identidade. Normalmente, uma tarefa relativamente simples como identificar um usuário pode ser complicado, devido à técnica utilizada em tal procedimento.

Um exemplo simples é o uso de nome de usuário e senha para se autenticar frente ao servidor. Imaginemos que o host A deseja se autenticar em B. Se simplesmente for enviado a senha para o host B, um atacante pode efetuar um *sniffing* e coletar a senha. Daí em diante o atacante pode se autenticar em B, fazendo se passar por A.

Então, a solução simples seria que a senha estivesse criptografada. Porém precisamos ter cuidado para não sofrermos um ataque de repetição, onde o atacante envia o mesmo pacote cifrado ao host B. Para resolver isto, normalmente se usa um sistema de autenticação baseada em chaves públicas ou através de desafio-resposta.

Os meios mais comuns de autenticação são o RADIUS e o KERBEROS.

- RADIUS (*Remote Authentication Dial In User Service*) é um protocolo de autenticação de acesso à rede relacionado com a mobilidade que provê AAA (*Authentication, Authorization and Accounting*). Por exemplo, quando um usuário disca para o RAS da empresa, a senha é repassada ao RADIUS Server que verifica, autoriza e bilheta o acesso. Inicialmente foi desenvolvido pela Livingston e agora é um padrão de facto para autenticação [RFC 2058, WIK 05-3].
- KERBEROS é um serviço de autenticação desenvolvido pelo MIT que usa criptografia simétrica e um centro de distribuição de chaves para autenticar o usuário na rede. O servidor de autenticação distribui as chaves e contém as informações de quais recursos o usuário tem acesso. [KUR02]

5.2.7 – Política de Segurança

Como vimos a política de segurança é um documento que contém a estratégia de segurança da empresa, incluindo aspectos tecnológicos, humanos e culturais.

No modo de vista da proteção, a política de segurança é a ferramenta que permite aos administradores configurar a rede para trabalhar de acordo com a política de segurança previamente estabelecida em conjunto com o corpo técnico e diretor.

Entretanto, tão importante quanto a criação da política e implementação através de recursos tecnológicos para cumpri-los, precisamos fazer uma auditoria na rede através da análise de log de forma que possamos acompanhar se a política de segurança está sendo cumprida ou se outras ações tecnológicas precisam ser implementadas para prevenir algum problema que ocorreu.

Uma política de segurança bem implementada pode resolver problemas em diversas frentes, tais como :

- Evitar *Password Guessing*: Se a política de segurança define um tamanho e complexidade mínimo, a chance de um atacante descobrir a senha no chute tem sua probabilidade reduzida
- Evitar falhas de segurança física: Uma boa política define quais pessoas possuem acesso físico à sala dos servidores
- Ajudar na configuração do *firewall*: Se os administradores já sabem quais serão os serviços públicos, quais recursos serão disponibilizados, o firewall pode ser corretamente verificado. A política também deve definir qual a política de auditoria dos logs.
- Ajudar na definição das ACLs: Uma boa política de segurança define o direito de acesso de cada classe de funcionários em uma organização, facilitando a configuração dos direitos nos servidores.
- Evitar vírus e spywares: Os usuários que conhecem a política sabem que devem restringir a instalação ou cópia de informações de fora para dentro da rede, além de instruí-los a que fazer em caso de infecção. Para os administradores, o servidor proxy deve restringir o acesso indevido à locais da Internet onde existe uma maior probabilidade de contaminação, pois a política define quais usuários terão acesso à cada grupo de websites.
- Evitar vazamento de informações: A política deve ser de conhecimento de todos, e se existe uma absorção daquele conteúdo no cotidiano das pessoas que possuem acesso à informações valiosas da empresa, podemos acreditar que estes indivíduos não fornecerão informações sigilosas. Incluindo ainda na política a punição para casos como este, o usuário fica intimidado a ceder informações a terceiros sem prévia autorização da organização.
- Evitar *dumpster diving*: O conhecimento adquirido pelos usuários da política devem fazer que eles destruam as informações (como papéis, discos rígidos, CDs, etc.) antes de jogá-los no lixo
- Evitar ataques à redes sem fio: As redes sem fio devem estar protegidas de acordo com a política de segurança mínima da corporação. Devemos ter cuidado com as chaves utilizadas no acesso aos *Access Points* e seguir as recomendações do grupo de segurança.
- Evitar ataques de engenharia social: Se a falha por engenharia social é causado pelo humano, ele também poderá evitá-lo, limitando o círculo de pessoas com quem conversa sobre assuntos de trabalho (mesmo os menos importantes), evitar deixar papéis importantes espalhados sobre a mesa e deve sempre desconfiar das pessoas aos quais não tem algum relacionamento profissional solidificado.

5.2.8 – Equipe de Funcionários Especializados

Toda empresa que possui informações importantes é um potencial alvo para um hacker. Muitas empresas, erroneamente, implantam um esquema de segurança apenas após o primeiro incidente e nesta oportunidade faz gastos enormes para corrigir as falhas já exploradas e rotineiramente se esquecem do assunto novamente.

Uma equipe de segurança deve estar trabalhando na continuidade e melhoria do plano de ação estabelecido, na época da implantação dos esquemas de segurança.

Esta equipe deve estar constantemente preparada para evitar, e se necessário responder a incidentes de segurança com alta velocidade de modo que a rede não fique comprometida. A equipe tem uma longa lista de tarefas para ser executada diariamente:

- aplicações de *patches*;
- configuração de ACLs;
- manutenção da base de usuários;
- monitoração de conexões;
- auditar e analisar os *logs* em busca de características de ataques;
- atualizações de antivírus nas máquinas dos clientes;
- constante atualização dos seus conhecimentos (estudos);
- responder a incidentes de segurança se houverem;
- aprender com comportamentos da rede de modo à melhorar o esquema de segurança;
- atualização de regras de *firewall*, IDSs, etc.

A equipe de segurança é composta por membros que tem acesso privilegiado à todo o parque tecnológico instalado, e este cargo é considerado de confiança. Infelizmente, a empresa tem a difícil tarefa de confiar algo (a segurança) à quem não conhecemos ainda. Algumas técnicas são efetuadas para testar este novo profissional:

- Auditoria cruzada: onde uma pessoa de segurança verifica os passos da nova contratada. Nestes casos a empresa deve ter uma pessoa a quem ambos se reportam, porém um não deve se reportar ao outro na intenção de evitar uma "fraude múltipla";
- Indicação: Ocorre quando os novos apenas são empregados se forem indicados por uma pessoa já de confiança da empresa. Cria-se um relacionamento um pouco melhor ou (quando há confiança recíproca) em se confiar em alguém que confia no novo profissional.
- Checagem de passado: Não é incomum que a empresa que contrata um novo profissional, entre em contato com a empresa anterior para verificar se houve algum incidente envolvendo o mesmo. Além disso, atualmente é de praxe as grandes empresas pedirem antecedentes criminais do novo funcionário;

Temos que ter em mente que em segurança, temos que nos proteger por todos os lados, de forma que, como comentamos, se apenas uma brecha existir então a organização está com uma falha de segurança que eventualmente será explorada por algum *hacker* determinado.

5.2.9 – IDS

IDS (*Intrusion Detection System*) são softwares que executam uma análise do tráfego que passa, tipicamente, por um *firewall* de modo a procurar por assinaturas de ataques conhecidos.

Basicamente, podemos definir IDS como uma ferramenta inteligente capaz de detectar tentativas de invasão e tempo real. Esses sistemas podem atuar de forma a somente alertar as tentativas de invasão, como também em forma reativa, aplicando ações necessárias contra o ataque. [CAOR 05].

5.2.9.1 - Sistemas Baseados em Regras (*Rule-based systems*)

Esse tipo é baseado em bibliotecas ou bases de dados que contenham assinaturas dos ataques. Quando algum tráfego coincide com um critério ou regra, ele é marcado como sendo uma tentativa de intrusão.

A maior desvantagem dessa técnica é a necessidade de se manter a base de dados constantemente atualizada, e além de que essa técnica somente identifica os ataques conhecidos. Além disso, às vezes pode existir uma relação inversa entre a especificação da regra e sua taxa de acerto. Isto é, se uma regra for muito específica, ataques que sejam similares, mas não idênticos, não serão reconhecidos.

5.2.9.2 - Sistemas Adaptáveis (Adaptive Systems):

Esse tipo emprega técnicas mais avançadas, incluindo inteligência artificial, para reconhecer novos ataques e não somente ataques conhecidos através de assinaturas.

As principais desvantagens dos sistemas adaptáveis é o seu custo muito elevado e a dificuldade no seu gerenciamento, que requer um grande conhecimento matemático e estatístico.

5.2.9.3 – NIDS e HIDS

Existem duas arquiteturas de IDS. O NIDS (*Network Intrusion Detection System*) que é um IDS para a rede e o HIDS (*Host Intrusion Detection System*) que é um IDS para um computador específico.

A grande parte dos sistemas comerciais de detecção de intrusão é baseada em rede. Nesse tipo de IDS os ataques são capturados e analisados através de pacotes de rede.

Ouvindo um segmento de rede, o NIDS pode monitorar o tráfego afetando múltiplas estações que estão conectadas ao segmento de rede, assim protegendo essas estações. Os NIDSs também podem consistir em um conjunto de sensores ou estações espalhados por vários pontos da rede.

Essas unidades monitoram o tráfego da rede, realizando análises locais do tráfego e reportando os ataques a um console central. As estações que rodam esses sensores devem estar limitadas a executar somente o sistema de IDS, para se manterem mais seguras contra ataques. Muitos desses sensores rodam num modo chamado *stealth*, de maneira que torne mais difícil para o atacante determinar as suas presenças e localizações.

Os HIDSs operam sobre informações coletadas em computadores individuais. Através disso os HIDS podem analisar as atividades das estações com confiança e precisão, determinando exatamente quais processos e usuários estão envolvidos em um tipo particular de ataque no sistema operacional. Além disso, ao contrário dos sistemas

baseados em rede, os baseados em host (estação) podem ver as conseqüências de uma tentativa de ataque, como eles podem acessar diretamente e monitorar os arquivos e processos do sistema usualmente alvos de ataques.

Alguns HIDSs suportam um gerenciamento centralizado e relatórios que podem permitir que um apenas um console possa gerenciar várias estações. Outros geram mensagens em formatos que são compatíveis com os sistemas de gerenciamento de redes.

A seguir, é mostrada pela tabela 5 uma comparação entre NIDS e HIDS [CAOR 05]:

	NIDS	HIDS
Vantagens	• A implementação de um NIDS tem pouco impacto sobre o desempenho da rede. Eles geralmente ficam em modo passivo, apenas escutando o tráfego da rede sem interferir no seu funcionamento. • NIDS bem posicionados podem monitorar uma grande rede. • Os IDSs baseados em rede podem ser muito seguros contra a maioria dos ataques, além de ficarem invisíveis aos atacantes.	• Esse tipo de IDS tem a capacidade de monitorar eventos locais de um host, podendo detectar ataques que não poderiam ser detectados por um IDS de rede. • Eles podem operar em um ambiente onde o tráfego de rede é criptografado. • Quando o IDS de host opera em nível de sistema operacional, ele pode ajudar a detectar Cavalos de Tróia.
Desvantagens	• Os NIDS podem ter dificuldade em processar todos os pacotes em uma rede de grande tráfego; • Eles não podem analisar o tráfego de informações criptografadas; • Muitas vantagens dos NIDSs não se aplicam mais as modernas redes baseadas em switches a não ser que o IDS seja instalado no gateway; • A maioria dos NIDSs não podem reconhecer se um ataque foi bem sucedido. Eles apenas apontam que um ataque foi iniciado. • Alguns IDSs baseados em rede têm problemas em lidar com pacotes de dados fragmentados.	• Esse tipo de IDS é difícil de gerenciar porque cada host monitorado precisa ser configurado. • Como as informações utilizadas para análise do HIDS estão armazenadas no host, um atacante pode invadir o sistema e desabilitar essas funcionalidades. • Os HIDSs não podem reconhecer ataques que sejam destinados a rede inteira porque apenas conseguem monitorar os pacotes de redes recebidos pelo próprio host. • Um IDS baseado em host consome recursos de processamento do host monitorado, influenciando na sua performance.

Tabela 5: Comparativo entre as arquiteturas NIDS e HIDS

5.2.9.4 – Exemplo

O IDS mais famoso da atualidade é o Snort. Este é um NIDS que faz log em tempo real, criado por Martin Roesch para plataforma Linux *open-source* e é distribuído na licença GNU GPL. A sua principal característica é a facilidade de implantação e um banco de dados de ataques conhecidos (*well know attacks*) que provê uma boa performance.

O *Snort* detecta vários tipos de tráfego em redes IP através da análise de pacotes (como o tcpdump), inclusive pode detectar pacotes que contenham assinatura semelhante aos cavalos de tróia, *scanning*, ataques de CGI, *scanning de OS Version Determination*, e mais. O Snort também pode estar integrado a sistemas de avisos em tempo real, tais como Syslog, arquivos de log, Unix socket ou via WinPOPup (requer smbclient).

Plugins adicionais permitem que ele seja estendido incluindo um bloqueio de ataques (inline snort), log em banco de dados MySQL ou ainda permite que os administradores escrevam suas próprias regras dentro dos arquivos de configuração.

Quando um pacote passa pelo NIDS do *Snort* com as características que estão na base de dados do mesmo, os dados deste pacote são colocados em log para posterior análise do administrador de segurança. Vejamos um exemplo dado por [BEI 02]:

```
09/22 - 15:18:06.911226  [**] [1:469:1] ICMP PING NMAP [**] [Classification: Attempted
Information Leak] [Priority: 2] {ICMP} 192.168.1.20 -> 192.168.1.237
```

Este é um típico log de uma tentativa de *scanning* feito pelo NMAP. No inicio da linha temos a data (09/22 – 22 de Setembro), seguido da hora (15h 18m 06s), o código da assinatura do ataque (SID 469), a mensagem e a classificação do erro. Por fim, temos o protocolo juntamente com a origem e destino do pacote ({ICMP} 192.168.1.20 -> 192.168.1.237)

5.2.10 – HoneyPots – Aprendendo com Erros Alheios

Na vida, normalmente aprendemos efetivamente com os erros que cometemos ou erros que pessoas ligadas à nós cometem. Estas lições de vida podem ser úteis na área de segurança de rede, porque cada ataque não é singular, cada ataque tem sim peculiaridades, mais todos são baseados nos mesmos princípios da falta de segurança citados neste trabalho.

Honeypots é um termo em inglês que quer dizer pote de mel. Este termo já existe há algum tempo e na verdade *Honeypots* são armadilhas para os atacantes que consiste em uma máquina conectada à rede com o objetivo de ser atacada. O objetivo deste mecanismo é deixarmos um sistema vulnerável, normalmente com um *firewall* com IDS transparente em frente a ele, para que possamos aprender novas técnicas de ataque.

Se o atacante fizer um ataque não conhecido e conseguir com sucesso comprometer o *Honeypots*, então podemos ter aprendido uma lição com os erros propositais. Tal ataque pode ser utilizado como uma "vacina" na rede corporativa real, onde implementaremos técnicas (ou novas técnicas) para evitar o ataque previamente detectado no Honeypot.

Uma variedade de produtos permite a todos criar a sua própria Honeypot. Tais opções incluem:

- *Deception ToolKit*. Criado por Fred Cohen (http://all.net/dtk/dtk.html), normalmente chamada de DTK, é uma coleção de scripts que emula

diversas vulnerabilidades conhecidas. Em seguida, esses scripts serão executados em um sistema hospedeiro. O propósito dele é enganar atacantes, de modo a aprendermos com vulnerabilidades conhecidas. Entretanto, tal abordagem pode ser considerada inútil, pois o objetivo maior da *Honeypot* é aprender sobre ataques não conhecidos.

- *Cybercop Strings*: é um *Honeypot* executado em Windows NT que emula toda a pilha de rede dos diversos sistemas operacionais, e com um único host, podemos emular até 15 sistemas disponíveis.
- *Recouse Mantrap*: é um produto comercial que tem a funcionalidade próxima a uma *honeynet*, pois não replica um sistema operacional, mas sim executa uma imagem do sistema dentro de outro em uma jaula. Desta maneira o atacante realmente pode interagir com o sistema atacado.

5.2.10.1 – HoneyNet

A Honeynet.BR entrou em operação em março de 2002, com o objetivo de ter uma rede no Brasil para que seja possível aprender com os novos ataques da Internet.

A topologia da Honeynet.BR está dividida em duas partes distintas: a Rede Administrativa e a Honeynet propriamente dita.

A Rede Administrativa tem as funções principais de conter a saída de tráfego malicioso da *Honeynet* e monitorar todo o tráfego, seja ele interno ou não. Ela é transparente tanto para a Internet quanto para a própria *Honeynet*. Esta rede é composta por:

- Um *Firewall*, que permite a entrada de todo o tráfego para a Honeynet, mas possui regras para impedir a saída de tráfego malicioso. Este controle é feito na camada de enlace, com o Firewall operando como uma bridge;
- Uma máquina (*Hogwash*), configurada de modo a bloquear a saída de tráfego com conteúdo sabidamente malicioso. Esta máquina também opera como bridge;
- Um IDS (*Intrusion Detection System*), que captura e analisa o tráfego da Honeynet e emite alertas no caso de seu comprometimento. Também é responsável pela emissão de sumários diários sobre a atividade observada;
- Uma máquina destinada a armazenar artefatos e imagens dos discos dos hosts da Honeynet.BR quando de seu comprometimento (*Forensics*).

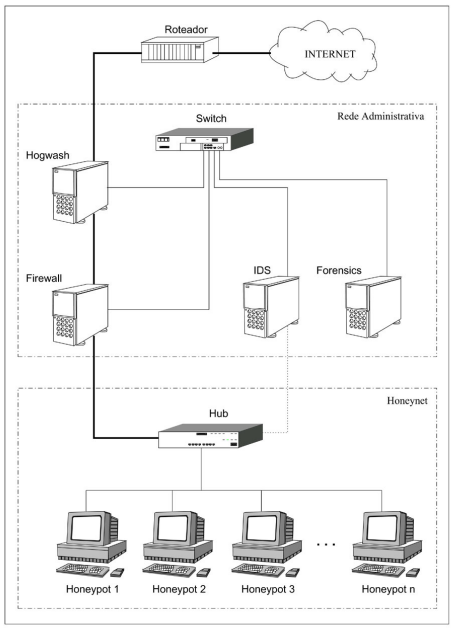


Figura 17: Topologia da Honeynet.BR em detalhes

Uma vez instalada, a *Honeynet* está pronta para receber os ataques. Veja que existe uma parte da rede que o atacante enxergará (*Firewall*, Roteador e *Honeypots*). O *Hogwash* impede que o atacante use a *Honeynet* para atacar outros lugares (saída), fazendo com que o ataque possa entrar, mas não possa sair. Como o *Hogwash* roda em nível de enlace (nível 2), o atacante não tem sabe da existência da mesma e se tentar usar um *Honeypot* para atacar outro lugar não conseguirá e terá a sensação que o ataque não funcionou.

O IDS registra em detalhes todos os pacotes que a *Honeynet* trafega para posterior análise, parte mais importante da *HoneyNet*. O forensics tem como tarefa guardar as imagens de disco dos *Honeypots* e ferramentas utilizadas pelos invasores.

5.2.11 – PenTests

Como já visto, existem inúmeras formas de proteger a rede e cada uma tem uma função específica dentro do contexto de segurança. Desde a política de segurança, até modernos firewalls e IDSs, podemos ter uma complexa gama de ferramentas para a proteção da nossa empresa.

Mas como qualquer outra forma de projeto na informática, esta segurança precisa ser testada e então entra o *PenTest*.

O *PenTest* (*Penetration Test*) é um teste das ferramentas de segurança, e através desta técnica deveremos testar todos os componentes da rede. Normalmente são usadas *scripts* prontos (usados pelos *script kiddies*), juntamente com modernas técnicas de ataques para assegurarmos que o mesmo se encontra seguro

O *PenTest* não apenas deve dizer quais são os recursos que podem ser explorados por atacantes, mas também definir o risco que este causa à organização.

De modo geral, devemos fazer *PenTest* à intervalos regulares de modo que possamos garantir que tudo está bem. Sempre que possível, contratar uma empresa externa que execute este teste é uma boa idéia, já que ela pode atacar de algumas formas

que o administrador não conhecia, pois as pessoas tendem a atacar em pentests contra vulnerabilidades que elas se protegeram previamente.

Porém, como sabemos, mesmo as empresas que trabalham com este procedimento são um risco para a organização. Enquanto a grande parte das empresas terceirizadas de pentest são confiáveis, devemos ter em mente os riscos usuais de um ataque pentest.

Porque manter a segurança da informação é importante em várias empresas, incluindo instituições financeiras, essas devem entender a habilidade das pessoas externas a organização de penetrar em seus sistemas para fraudes. Tais testes, como dito, trazem seus próprios riscos, tanto para a organização quanto para os usuários que podem ter uma indisponibilidade do serviço ou até mesmo a queda de um sistema. Toda organização que está contemplando um teste de penetração contra a rede de produção deve ter uma ampla idéia dos riscos provenientes desta decisão, porém também deve-se levar em conta os riscos de se os mesmos ataques fossem feito por hackers que desejassem encontrar uma brecha de segurança na organização.

Mas porque devemos terceirizar este *PenTest*:

- para determinar a existência e extensão das vulnerabilidades não detectadas pela equipe interna;
- para mostrar aos clientes o quão seguro é fazer *e-commerce* através da WWW. O marketing deve frequentemente utilizar estes resultados positivos a seu favor;
- Como uma prevenção à segurança de rede de uma organização de modo a aperfeiçoar as técnicas utilizadas atualmente;
- Para entendermos o que deve ser alterado em uma reestruturação interna, incluindo rede e equipe;
- Ao fazer um *outsourcing*, deveremos ter certeza da segurança na transferência do know how.

Também devemos levar em conta o objetivo e nível de sofisticação do ataque feito pela equipa de teste. Existem normalmente três tipos de sofisticação [LOW 02]:

- Hacker por esporte: Normalmente divididos em subcategorias como o novato (atacante da própria máquina), *script kiddies* ou crackers.
- Inteligência competitiva: Aqueles hackers que tem como objetivo ganhar tecnologia do seu concorrente através do ataque. Normalmente se utiliza de um *eavesdropping/sniffer*;
- Inteligência estrangeira: Atacantes que ganham informações usadas por um país estrangeira ou como um terrorista. Por exemplo, se um terrorista pudesse criar um *backdoor* para descobrir fórmulas de bombas, ou terroristas fazendo *web defeacement*"em órgãos públicos.

A maioria dos ataques deve estar no primeiro ou segundo item citado anteriormente para uma rede com uma segurança razoável, entretanto temos que ter em mente que dependendo da motivação do hacker deveremos fazer uma abordagem mais séria em relação a segurança da informação.

De modo geral, para a maioria das redes de pequenas e médias empresas, o uso de ferramentas prontas de análise de vulnerabilidades tais como o Microsoft MBSA, o Nessus [NES 05], NMAP [INS 05], entre outras, juntamente com uma equipe experiente de análise deve ser o suficiente.

5.2.12 – Análise Forense

Uma vez que um sistema pode ter sido alvo de comprometimento por questões de falha de segurança, devemos fazer uma análise forense para detectarmos quais itens foram alterados, quais brechas foram utilizadas, o que podemos fazer para recuperar o controle do equipamento, etc.

Ela é especialmente importante em casos de fraude financeira, suspeitas de pedofilia ou crimes virtuais, roubo de informação confidenciais de modo a conseguir informações digitais relevantes para um processo judicial futuro.

Tipicamente, se não existe uma evidencia comprovada que o equipamento foi comprometido, é interessante fazermos uma análise antes de voltar o mesmo em produção. As coisas que devem ser vistas são o perfil dos processos e do uso de rede, os arquivos binários, os arquivos de contas de usuários, grupos e senhas, se permissões foram alteradas, e assim por diante.

Durante qualquer operação de análise forense, precisa-se ter em mente que o sucesso deste processo é deixar a máquina o menos deturpada possível. Devemos então saber que existem riscos em destruir as evidencias durante a análise. O ideal em ambientes de segurança crítica é:

- Retire o computador da rede local imediatamente;
- Retirar o serviço do ar assim que possível;
- Notificar aos usuários do ocorrido; pois se houver comprometimento das contas deve ser obrigatória a troca da senha;
- Usar uma outra máquina para ler as informações do disco rígido do servidor comprometido;
    - Após o acesso, primeiramente faça backup de todas as informações, incluindo o estado atual do equipamento, binários, logs de acesso, etc.
- Análise do *Firewall* e do IDS;
- Verificação imediata da abrangência do ataque (um host ou a rede toda?);
    - Verificação do uso de espaço em disco;
    - Verificação do uso de recursos da rede (banda);
    - Verificação das comunicações;
    - Verificação das aplicações relevantes;
- Pesquisa sobre os rastros do problema.

Quase todas as ações realizadas no ataque em algum momento se refletem na modificação do ambiente de trabalho (documentos, programas, arquivos, históricos de acesso, bancos de dados, etc.). Os exames de análise forense requerem altíssimo conhecimento do analista da plataforma, pois é necessário varrer minuciosamente todos os elementos capazes de armazenar informação, incluindo arquivos deletados.

5.2.13 - O que fazer após o ataque?

Diversas técnicas são utilizadas para a proteção dos ataques visando a maior segurança das informações, entretanto é possível que em algum momento haja um incidente de segurança.

Assim que houver uma suspeita de um incidente de segurança é necessário:

- Antes de tudo, retire o serviço do ar;
- Se for seu computador pessoal, desconecte-se da Internet;
- Guarde as informações de logs da rede, incluindo *firewall*, IDC, últimos acessos, etc.;
- Se preferir, contate um profissional de segurança da informação especializado;
- Verificar se não é um falso positivo, antes de notificar um incidente.

Se realmente houver um incidente com provas, devemos notificar as autoridades competentes da Internet com o seguinte procedimento:

- Incluir logs completos (com data, horário, timezone, endereço IP de origem, portas envolvidas, protocolo utilizado, etc.) e qualquer outra informação que tenha feito parte da identificação do incidente;
- Enviar a notificação para os contatos da rede e para os grupos de segurança das redes envolvidas;
- Notificar seu provedor de serviços de acesso à Internet ou administradores de redes;
- Enviar cópia dessas informações para a NIC BR - *Brazilian Computer Emergency Response Team* através de nbso@nic.br;

Identificar a origem do ataque é uma tarefa simples (ver o IP do atacante), porém na prática é muito mais difícil, pois o IP não leva ao atacante, e ainda corremos o risco da origem ser "*spoofada*". Devemos primeiramente nos prevenir contra um possível *spoofing* de forma a encontrarmos a origem certa de um ataque. Algumas dicas são:

- Na Internet são mantidas diversas bases de dados com as informações a respeito dos responsáveis por cada bloco de números IPs existente. Estas bases de dados estão nos chamados Servidores de *Whois*.
- O servidor de *Whois* para os IPs alocados ao Brasil pode ser consultado em http://registro.br/. Para os demais países e continentes existem diversos outros servidores.
- O site http://whois.geektools.com/cgi-bin/proxy.cgi aceita consultas referentes a qualquer número IP e redireciona estas consultas para os servidores de Whois apropriados.

Mesmo de posse do número IP e do nome do provedor de serviço ao qual o atacante está conectado, temos pela frente a identificação de qual usuário (hacker) estava utilizando aquele IP no momento do ataque. Para isso os ISPs tem ferramentas que permitem fazer esta detecção, porém no Brasil, a informação como Nome, Endereço e Telefone de Origem da Ligação é apenas cedida após pedido judicial, semelhante ao processo de quebra do sigilo bancário.

Capítulo 6 – Conclusão

Diante do exposto neste trabalho, podemos ver que a segurança da informação é algo que deve estar presente no dia-a-dia em todos os locais onde o ambiente computacional for necessário e que neste seja armazenada algum tipo de informação restrita.

Temos que estar constantemente reciclando os conhecimentos do profissional de segurança da informação, aprendendo com seus acertos, erros, sites da internet ou comunidade *blackhat*. Os profissionais desta área normalmente têm dificuldades em mostrar quais são os objetivos da segurança, e justificar os cursos relacionados aos procedimentos necessários para tal, conforme descreve o Capítulo 1.

Os profissionais ainda devem estar cientes do enorme número de ataques (Capítulo 2), vindo do mundo exterior e muitas vezes da própria organização (*insiders* – Seção 2.1.2), ou ainda os ataques não-tecnológicos como a Engenharia Social (Seção 4.6). Os ataques são tantos que podem ser divididos em categorias tais como Scanning (Seção 4.1) , Política de Segurança Ruim (Seção 4.2), Ataques de Rede (Seção 4.3) ou de Aplicações (Seção 4.5), Ataques de Negação de Serviço (Seção 4.4) e Ataques às pessoas (Engenharia Social – Seção 4.6).

Obviamente que para cada tipo de ataque conhecido, existem contramedidas que podem ser utilizadas para proteger-se dos ataques. A cada dia que houver um novo ataque, as equipes de segurança estarão encontrando novas formas de se protegerem destas vulnerabilidades. Entre os métodos de defesa mais comuns estão o Firewall (Seção 5.2.1), Segurança Física (Seção 5.2.2), Antivírus (Seção 5.2.3), Criptografia (Seção 5.2.5) e Autenticação (Seção 5.2.6), Boas políticas de Segurança (Seção 5.2.7), IDS (Seção 5.2.9). Da mesma maneira que nenhuma ferramenta é totalmente útil sem uma Equipe de Segurança (Seção 5.2.8) que possa proteger a rede contra intrusos, testar a segurança (Pentest - Seção 5.2.11), reciclar seus conhecimentos aprendendo mais sobre os invasores (HoneyPots - Seção 5.2.10) ou por fim, quando todos as proteções ainda não forem suficientes, responder à incidentes de segurança (Seção 5.2.13) e efetuar a análise forense (Seção 5.2.12).

Tantas são as preocupações da equipe de segurança frente aos ataques e proteções necessárias que poderíamos resumir em uma tabela as proteções necessárias contra os ataques aqui citados. O presente trabalho consolida por meio da tabela 6 estas proteções:

	Firewall	IDS	Antivírus	Anti Spyware	Criptografia	Hardening	Profissional Segurança	Boa Programação	Upgrade Software
Network Scanning	✓	✓					~		
Port Scanning	✓	✓					~		
Firewalking	✓	✓					~		
DNS Querying							~		
Version Determination		✓				~	~	~	
OS Version Determination	✓	✓				~	~	~	

(continua)

	Firewall	IDS	Antivírus	Anti Spyware	Criptografia	Hardening	Profissional Segurança	Boa Programação	Upgrade Software
Password Guessing		~			✓	✓	✓		
Eavesdropping	✓				✓		~		
IP Spoofing	✓	✓			~		~		~
Source Routing	✓	✓				✓	✓		
SYN Flooding	~	✓					~		
Smurf Flaggle	~	✓				✓	✓		
Fragmentação IP	✓	✓				✓	~		
Man-in-the-Middle					✓		~		
Seqüestro de Conexão	~				✓	~	~		✓
Redes sem fio	~				✓	~	~		
Buffer Overflow		✓				~	~	✓	✓
Format String Attack		✓				~	✓	✓	✓
SQL Injection						~	✓	✓	✓
Poison Null		✓				✓	✓	~	~
Upload Bombing		✓				✓	✓	~	~
WebSpoofing						✓	✓	~	~
Exploits		✓	~	~		~	✓	✓	✓
Vírus, Worms, Cavalos Tróia	~	~	✓	✓		~	~	~	✓

Tabela 6: Proteções necessárias aos ataques específicos.

Legenda:

✓ = Existe proteção, detecção ou melhora no nível de segurança através da utilização da técnica requerida, e eventualmente mais do que uma técnica pode se fazer necessária para alcançar a segurança máxima.
~ = Existe uma pequena ajuda na detecção ou resolução do problema, neste caso sendo importante observação de várias técnicas simultaneamente.
vazio = Não existe nenhum nível de proteção neste caso. Outras técnicas devem ser feitas para resolver o problema citado

Concluindo, o profissional de segurança exigido para proteger uma empresa de segurança não pode ser apenas no domínio tecnológico, mas deve agregar uma visão técnica ampla e uma visão gerencial apurada para que possa analisar um ambiente complexo, como um todo e cada um dos componentes, pois apenas uma brecha é o suficiente para o atacante ter sucesso. Este profissional de segurança deve ter no mínimo 24 meses de experiência. No dia-a-dia deve prover um suporte decisivo para a organização de modo geral no planejamento, implantação, auditoria, reciclagem e aperfeiçoamento, testes e resposta a incidentes na questão da segurança da informação.

Adendo 1 - Referencias Bibliográficas

[AUR 89] FERREIRA, Aurélio Buarque de Holanda. Dicionário Aurélio Básico da Língua Portuguesa. Editora Nova Fronteira. 1ª. edição, 1989.

[ALE 02] ALEXEY, Kulikov - University of Warwick. Social Engineering. Capturado em 07/07/05. http://www.inses.ru/lj/chapter4_rev5_2.pdf

[ANS 05] Site da Answers.com "Dumpster Diving". Capturado em 09/09/2005. http://www.answers.com/topic/dumpster-diving

[ANS 05-2] Site da Answers.com. "Firewall Definition". Capturado em 13/09/2005. http://www.answers.com/topic/firewall

[ANS 05-3] Site da Answers.com – "Hardening" – Capturado em 13/09/2005. http://www.answers.com/topic/hardening

[ATI 00] Site ATIS Alliance for Telecommunications Industry Solutions. "*Atis Telecom Glossary 2000*". Capturado em 13/09/2005. http://www.atis.org/tg2k/_screening_router.html

[CAOR 05] RAMIREZ, Prof. Carlos Alberto de Oliveira. Slides da Aula de Segurança da Informação da Faculdades de Valinhos, Pós Graduação em Gestão de Tecnologia da Informação/2005.

[BAS 05] Site do Bastille Linux – Screenshots. Capturado em 14/09/2005. http://www.bastille-linux.org/screenshot.htm

[BEI 05] Richard Bejtlich. TaoSecurity – "An Introduction to Snort". Houston ISSA Meeting 11 Apr 02. Capturado em 17/09/2005. http://southtexas.issa.org/Program%20Files/bejtlich_houston_issa1.pdf

[BRA 05] Site do Banco Bradesco. Capturado em 29/08/2005. http://www.bradesco.com.br/br/seguranca_informacao/oquee.html

[BEN 05] Site "*La taverne de benj*". Capturado em 13/09/2005. http://www.grassouille.org/docs/graphics/dmz.png

[BRG 03] BRAGHETTO, Luis F. SILVA, Sirlei da, et al. "Redes GSM e GPRS". Capturado em 01/09/2005. http://www.braghetto.eti.br/files/Trabalho%20Final%20GSM.pdf

[BRG 03-2] BRAGHETTO, Luis F. SILVA, Sirlei da. BRISQUI, Marcelo, et al. "RSA Criptografia Assimétrica e Assinatura Digital." Capturado em 10/09/2005. http://www.braghetto.eti.br/files/Trabalho%20Oficial%20Final%20RSA.pdf

[BRG 05] BRAGHETTO, Luis Fernando/American Satelite. "Whitepaper Terabeam Security". Capturado em 12/09/2005. ftp://ftp.americansatelite.com.br/pub/Wireless/Terabeam/Documentacao/howto_portugues/TeraBeam_Security.pdf

[CER 04] Site de Estatística do CERT.br. Capturado em 28/08/2005. http://www.cert.br/stats/incidentes/2004-oct-dec/top-atacantescc.html

[CER 05] Site de Estatística do CERT.br. Capturado em 28/08/2005. http://www.cert.br/stats/incidentes/2005-jan-mar/top-atacantescc.html

[CIS 05] Site da Cisco.com. "*Routing Basics*". Capturado em 09/09/2005. http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/routing.htm

[COL 01] Site da Colasoft – "*Multiple Vulnerabilities In FaSTream FTP++"* – Capturado em 07/09/2005 - http://www.colasoft.com/resources/vulnerability.php?id=CAN-2001-0256

[CSI 02] Site da Computer Science Institute. "*Cyber crime bleeds U.S. corporations, survey shows; financial losses from attacks climb for third year in a row*" Capturado em 29/08/2005. http://www.gocsi.com/press/20020407.jhtml

[DEL 05] Site da Deloitte - Estudo com instituições financeiras confirma a importância da gestão de risco das informações. Capturado 29/08/2005. http://www.deloitte.com/dtt/press_release/0,1014,sid%253D6129%2526cid%253D56363,00.html

[ESP 04] FILHO, Antonio M da Silva. Revista Espaço Acadêmico – Nº. 43 – Dezembro 2004, Ano IV. "Entendendo e evitando a engenharia Social: Protegendo Sistemas e Informações". Capturado em 04/02/2005. http://www.espacoacademico.com.br/043/43amsf.htm

[IDS 01] WARREN, Trevor. Site Freeos.com. "Intrusion Detection System" Capturado em 19/09/2005. http://www.freeos.com/articles/3496/

[INF 04] Site InfoWester - Colunas - Ataques de engenharia social na Internet. Emerson Alecrim. http://www.infowester.com/col120904.php. Capturado em 07/09/2005.

[INS 05] NMAP Version Determination. Capturado em 12/09/2005. http://www.insecure.org/nmap/versionscan.html

[FIR 98] GOLDSMITH, David / Schiffman, Michael. "*A Traceroute-Like Analysis of IP Packet Responses to Determine Gateway Access Control Lists*" - Capturado em 19/03/2005. http://www.packetfactory.net/projects/firewalk/firewalk-final.pdf

[FOL 05] Site da Folha de São Paulo – "*Polícia prende jovem de 19 anos por desviar R$ 6,7 mi pela internet*" – Capturado em 05/09/2005 http://www1.folha.uol.com.br/folha/cotidiano/ult95u105116.shtml

[JUL 04] Site da cefetsc. Capturado em 19/08/2005. http://www3.cefetsc.edu.br/julio/paginas/alunos/Estru_PS.pdf

[HAN 96] A Menezes, P van Oorschot and S. Vanstone. "*Handbook of Applied Cryptography*". Editora CRC Press, 1996 – 1ª. Edição

[HNN 01] Site do HoneyNet Project. Capturado em 29/08/2005. http://project.honeynet.org/papers/stats/

[HUR 99] Hurley, Jim. Site "*Survival of the Fittest*". Capturado em 07/09/2005. http://infosecuritymag.techtarget.com/articles/1999/jancover2.shtml

[HON 02] The HoneyNet Project, Conheça seu Inimigo. Tradução Kátia A Roque. 2002, Editora Makron Books e Pearson Education.

[HON 05] Honeynet.BR Team, Instituto Nacional de Pesquisas Espaciais (INPE) e NIC BR (NBSO). "HONEYNET.BR: Desenvolvimento e Implantação de um sistema para avaliação das atividades hostis na Internet brasileira". Capturado em 19/09/2005. http://www.honeynet.org.br/papers/hnbr-ssi2002.pdf

[ISS 05] Internet Security Systems. "Source Routing". Capturado em 09/09/2005. http://www.iss.net/security_center/advice/Underground/Hacking/Methods/Technical/Source_Routing/default.htm

[KAT 77] KATZAN, Harry. Segurança de dados em computação. Editora LTC, 1977

[KUR 02] KUROSE, James, ROSS, Keith. Computer Networking A top down approach Featuring the Internet. 2ª. ed. Ed. Addison Wesley

[LAB 04] Site "The Art of Deception: Controlling the Human Element of Security". Capturado em 04/07/2005. http://labmice.techtarget.com/BookReviews/articles/artofdeception.htm

[LIC 03] GEUS, Paulo Licio e Nakamura, Emilio T. Segurança de Redes em Ambientes Colaborativos. Editora Futura, 2003.

[LIN 03] GROVER, Sandeep. "Buffer Overflow Attacks and Their Countermeasures". Linux Journal, 10/03/2003. Capturado em 02/04/2005. http://www.linuxjournal.com/article/6701

[LOW 02] LOWERY, Jessica. "Penetration Testing - The Third Party Hacker". SANS Institute 2002. Capturado em 19/09/2005. http://www.sans.org/rr/whitepapers/testing/264.php

[NAC 02] Nakamura, Emilio Tissato. Vulnerabilidades e Ataques. Capturado em 01/09/2005. http://www.las.ic.unicamp.br/srac/SdS%20-%20Vulnerabilidades%20e%20Ataques.pdf

[NAS 00] LEON, Nash. "Clube dos Mercenários". Capturado em 13/09/2005. http://www.frontthescene.com.br/CdM/artigos/format.txt

[NET 05] Site da NetFilter.org. "*Linux 2.4 Packet Filtering HOW-To: Guia Ultra rápido do Rusty para filtragem de pacotes*". Capturado em 13/09/2005. http://www.netfilter.org/documentation/HOWTO/pt/packet-filtering-HOWTO-5.html

[NES 05] Nessus Open Source Vulnerability Scanner Project. Capturado em 19/09/2005. http://www.nessus.org/

[NIC 05] Cartilha de Segurança da NIC.br. Capturado em 07/09/2005. http://cartilha.cert.br/download/cartilha-07-incidentes.pdf

[MIC 05] Site da Microsoft Brasil. Capturado em 27/08/2005. http://www.microsoft.com/brasil/pequenasempresas/issues/sgc/articles/why_security_matters.mspx

[MIC 05-2] Site da Microsoft. "*What you can do about spy wares and other unwanted software*". Capturado em 13/09/2005. http://www.microsoft.com/athome/security/spyware/spywarewhat.mspx

[MIC 05-3] Site da Microsoft. *"What is a computer virus?"* Capturado em 13/09/2005. http://www.microsoft.com/athome/security/viruses/intro_viruses_what.mspx

[MIC 05-4] Site da Microsoft. "Windows 2003 Server Security". Capturado em 14/09/2005. http://www.microsoft.com/technet/security/prodtech/windowsserver2003/W2003HG/SGCH00.mspx

[MIT 02] MITNICK, Kevin: The Art of Deception. Editora Wiley Publishing Inc. 2002

[MOD 05] As Normas BS7799 e ISO17799. Capturado em 06/09/2005. http://www.modulo.com.br/empresa/site/pop_bs7799.jsp

[OVE 04] Site da Overclockers.ru – "Microsoft BaseLine Security Analyzer" – Capturado em 10/09/2005. http://www.overclockers.ru/softnews/17409.shtml

[PAT 04] Site da Isto É Online. "Patrick Gray - O James Bond da internet". Capturado em 29/08/2005. http://www.terra.com.br/istoe/1812 /1812_vermelhas_01.htm

[PEN 05] Site Forinsect. "*Forensics, Intrusion Detection, Security Technology*". Capturado em 19/09/2005. http://www.forinsect.de/security/

[REG 03] The Register. "*Schoolgirl turns tables on email credit card fraudster*" Capturado em 09/09/2005. http://www.theregister.co.uk/2003/02/20/schoolgirl_turns_tables_on_email/

[RFC 791] Internet Protocol. September 1981. – Capturado em 08/08/2005. http://www.ietf.org/rfc/rfc0791.txt

[RFC 1597] Rekhter, Y. Moskowitz, B. et al. "*Address Allocation for Private Internets*". Capturado em 13/09/2005. http://www.faqs.org/rfcs/rfc1597.html

[RFC 2058] Rigney, C, Livingston, et al. "*Remote Authentication Dial In User Service*". Capturado em 14/09/2005. http://www.faqs.org/rfcs/rfc2058.html

[RNP 97] RNP.br – "Segurança – Você se preocupa com isso?" Capturado em 29/08/2005. http://www.rnp.br/newsgen/9705/n1-3.html

[SEC 05] SecuriTeam "*Exploits Directory*". Capturado em 09/09/2005. http://www.securiteam.com/exploits/archive.html

[SOA 02] SOARES, Luiz Fernando Gomes, LEMOS, Guido, COLCHER, Sérgio. Redes de computadores: das LANS, MANS e WANS as redes ATM

[SQL 02] SecuriTeam. "*SQL Injection Walkthrough*". Capturado em 09/09/2005. http://www.securiteam.com/securityreviews/5DP0N1P76E.html

[TER 05] Vírus & Cia., Terra. "FBI prende supostos criadores do vírus Zotob". Capturado em 03/09/2005. http://tecnologia.terra.com.br/interna/0,,OI643514-EI4805,00.html

[TRE 03] TREVENZOLI, Ana Cristina. MUSUTA, Silvana Mieko. "Trabalho sobre Buffer Overflow e ataque format string – INF542/Especialização de Redes de computadores da Unicamp". Redigido em Novembro/2003. Capturado em 14/09/2005 através das autoras.

[TOT 95] SHIMOMURA, Tsutomu. "*How Mitnick hacked Tsutomu Shimomura with an IP Sequence Attack*". Capturado em 09/09/2005. http://www.totse.com/en/hack/hack_attack/hacker03.html

[VUN 05] VUNet "Zotob suspect linked to other viruses". Capturado em 07/09/2005. http://www.vnunet.com/vnunet/news/2141631/zotob-suspect-linked-viruses

[WEB 05] Site da webopedia. "What is a firewall?" Capturado em 13/09/2005. http://www.webopedia.com/TERM/f/firewall.html

[WIK 05] Site da Wikipedia – IP Spoofing – Capturado em 07/09/2005. http://pt.wikipedia.org/wiki/IP_spoofing

[WIK 05-2] Site da Wikipedia – Exploits – Capturado em 09/09/2005. http://pt.wikipedia.org/wiki/Exploit

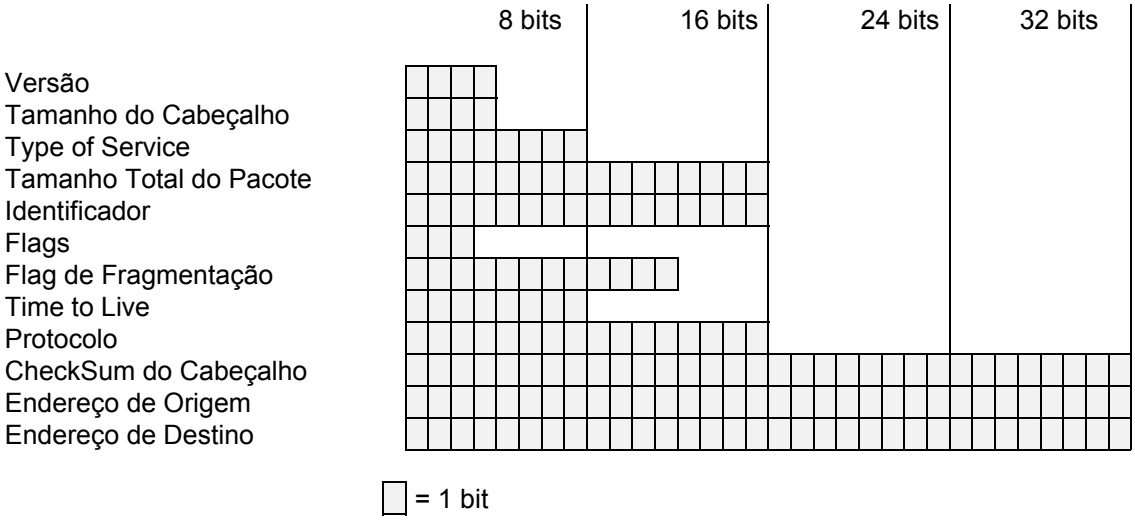
[WIK 05-3] Site da Wikipedia – RADIUS – Capturado em 14/09/2005. http://en.wikipedia.org/wiki/RADIUS

[WRI 05] Wright, Timothy E, SecurityFocus. "*A Method for Forensic Previews*" http://www.securityfocus.com/print/infocus/1825

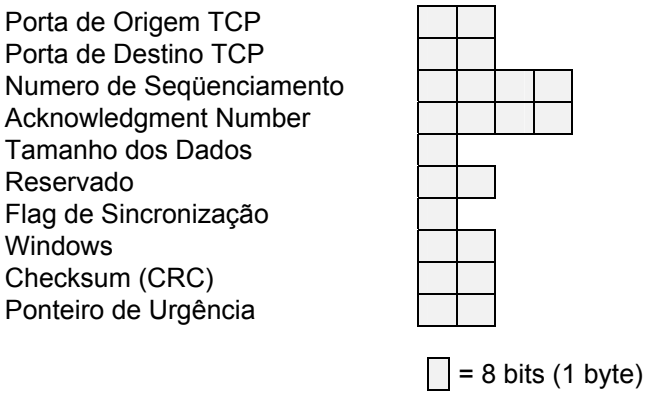
[WHA 05] Site Whatis.com. "*searchSecurity.com Definitions*" – Capturado em 07/09/2005. http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci802800,00.html

Adendo 2 – Referências Adicionais

A3.1 - Cabeçalho IP



A3.2 - Cabeçalho do TCP



A3.3 - Cabeçalho UDP