Informações sobre a disciplina

Ementa

Introdução ao Linux como ferramenta de apoio às práticas;.Noções de segurança. Os serviços de rede e suas vulnerabilidades. Métodos conhecidos de ataque. O conceito “Firewall” – filtragem de pacotes/NAT/PAT. Monitoração e auditoria.

Carga horária

60 horas

Objetivos

Reconhecer conceitos teóricos e, principalmente, práticos sobre segurança em redes de computadores.

Distinguir os métodos básicos de configuração e procedimentos conscientes com a segurança da informação.

Metodologia

O conteúdo programático será apresentado sob a forma de textos e exemplos, com atividades a serem realizadas.

Avaliação

Prova escrita ao fim da disciplina e atividades.

Plano de unidades

Unidade 1 – Linux

Conteúdo	***Onde encontrar***
Histórico – evolução da preocupação com a segurança da informação, métodos históricos de proteção, aparecimento de vírus, hackers x crackers Linux e distribuições Kernel, shell e aplicações Usuário comum X Administrador do Sistema Nomenclatura de dispositivos Processos Download e preparação do CD Instalando o Linux Utilizando o sistema – comandos	Textos 1 a 15

Unidade 2 – Noções de Segurança

Conteúdo	***Onde encontrar***
Segurança física – backup Usuário: - autenticação do usuário - usuários do sistema Permissão de acesso: - dono, grupo e outros - tipo de permissão - validação do acesso - permissões avançadas - permissões em números - trocando dono, grupo e permissões Aumentando a segurança	Textos 16 a 28

Unidade 3 – Serviços de rede

Conteúdo	*Onde encontrar*
Serviços & portas Protocolo TCP Conectividade excessiva Software – ciclo do software Configuração se sistemas Linux Controlando portas e serviços	Textos 29 a 37

Unidade 4 - Ataques conhecidos

Conteúdo	*Onde encontrar*
Perfil e tipos de hacker Anatomia de um ataque de hacker Honeypots Principais tipos de ataques Principais vulnerabilidades	Textos 38 a 64

Unidade 5 – Melhorando a configuração no Windows XP Professional

Conteúdo	*Onde encontrar*
Aspectos da configurações do Windows XP Professional Serviços do Windows 2000/XP Direito de acesso de usuários-	Textos 65 a 67

Sumário

Informações sobre a disciplina.....3
Plano de unidades.....4
Unidade 1 – Linux.....8
Texto 1 – Apresentando o Linux.....9
Texto 2 – As distribuições.....11
Texto 3 - Kernel, shell e aplicações.....12
Texto 4 – Usuário comum X Administrador do sistema.....13
Texto 5 – Nomenclatura de dispositivos.....14
Texto 6 – Processos 16
Texto 7 – Download e preparação do CD.....17
Texto 8 – Instalando o Linux.....20
Texto 9 – Utilizando o sistema.....21
Texto 10 – Comandos de ajuda.....23
Texto 11 – Comandos de manipulação de arquivo.....25
Texto 12 – Comandos de manipulação de diretório.....26
Texto 13 – Comandos de manipulação de discos e partições.....27
Texto 14 – Comandos de manipulação de permissão.....28
Texto 15 – Comandos de gerenciamento de processos.....29
Unidade 2 – Noções de Segurança.....31
Texto 16 – Segurança física.....32
Texto 17 – Backup 33
Texto 18 – Usuário 34
Texto 19 – Autenticação do usuário.....38
Texto 20 – Sign-On (SO) 39
Texto 21 – Single Sign-On (SSO).....43
Texto 22 – Autenticação forte.....45
Texto 23 – Usuários do sistema.....52
Texto 24 – Permissão de acesso.....53
Texto 25 – Tipo de permissão.....54
Texto 26 – Trocando dono, grupo e permissões.....58
Texto 27 – Prática dos comandos de permissão.....60
Texto 28 – Aumentando a segurança.....64
Unidade 3 – Serviços de rede.....68
Texto 29 – Serviços & portas.....69
Texto 30 – Protocolo TCP.....70
Texto 31 – Conectividade excessiva.....72
Texto 32 – Software 73
Texto 33 – Ciclo do Software.....76
Texto 34 – Configuração.....77

Texto 35 – Controlando o que deve ficar ligado.....83
Texto 36 – Desligando o serviço.....87
Texto 37 – Iniciando no boot.....89
Unidade 4 – Ataques conhecidos.....91
Texto 38 – Ataques conhecidos.....92
Texto 39 – Perfil e tipos de hacker 93
Texto 40 – Engenharia social.....95
Texto 41 – Anatomia de um ataque de cracker.....97
Texto 42 – Honeypots.....97
Texto 43 – Sistema de logs e auditorias incompleto ou inexistente.....99
Texto 44 – Tipos de ataques.....100
Texto 45 – Buffer overflow.....106
Texto 46 – Falta de proteção nos compartilhamentos em redes Windows. . 107
Texto 47 – Vazamento de informações em sessão anônima.....109
Texto 48 – Codificação fraca de senhas no SAM (LAN manager hash) .111
Texto 49 – Buffer overflow no RPC.....114
Texto 50 – Vulnerabilidades no sendmail.....115
Texto 51 – Liberdade de acesso a servidores por comandos remotos. .119
Texto 52 – Serviços de administração remota e sistema de arquivos....120
Texto 53 – Serviço de impressora.....121
Texto 54 – Mensagens-padrão do SNMP.....122
Texto 55 – Negação de Serviço (DoS) e SYN Flooding.....122
Texto 56 – Distributed Denial of Service (DDoS).....128
Texto 57 – Ataques de força bruta.....128
Texto 58 – Man-in-the-middle attack.....129
Texto 59 – Birthday attack em funções de hash.....131
Texto 60 – Spoofing e sequence number attack.....131
Texto 61 – Smurf.....136
Texto 62 – Session hijacking.....138
Texto 63 – Exploits.....139
Texto 64 – Trojans horses (cavalos de Tróia).....139
Unidade 5 – Melhorando a configuração no Windows XP Professional.....142
Texto 65 – Aspectos da configurações do Windows XP Professional.....143
Texto 66 – Serviços do Windows 2000/XP.....146
Texto 67 – Direito de acesso de usuários.....149
Referências.....152
Índice Alfabético.....153

Unidade 1 – Linux

Pretendemos explicar a facilidade de se obter gratuitamente e dentro da lei um exemplar de Linux, com exemplos práticos em segurança da informação.

Apresentamos o básico de uma distribuição de Linux, o Slax, atualmente considerado um dos mais fáceis de usar, sem necessidade inicial de instalar no HD, pois pode rodar diretamente a partir de um CD que dá boot. Ele será utilizado para compreender os conceitos e para que você possa exercitá-los em sua própria máquina, junto aos exemplos do curso.

Não se trata de um curso de Linux, apenas uma introdução à utilização do Slax, com a finalidade de obter respaldo de um sistema real para estudo. O aprofundamento no assunto, seguindo os links indicados, é altamente recomendado, a partir de sites com boa documentação em português.

Texto 1 – Apresentando o Linux

O LINUX apareceu em 1991, como um projeto pessoal do então estudante finlandês de 24 anos, Linus Torvalds. Ele simplesmente queria ter acesso ao sistema operacional UNIX da universidade, remotamente, usando um emulador de terminal. Depois, o objetivo passou a ser: ter um sistema operacional que fosse parecido com um UNIX para seu próprio uso, sem gastar muito dinheiro na compra de uma estação de trabalho e licença de sistemas operacionais. Na época, só havia UNIX para estações de trabalho (RISC) e cada estação custava, no mínimo, cinco mil dólares. E ainda estaria faltando a licença de uso do sistema operacional em si.

Em 1990, um 386 custava uma fortuna: 18 mil marcos finlandeses, cerca de US$ 3.500. Em janeiro de 91, finalmente, Linus conseguiu comprar a maravilha tecnológica e foi nesse PC – potentíssimo para 1991, com 4Mb de RAM e processador de 33MHz – que o Linux nasceu.

Naquele inverno, Torvalds não saiu da toca. Começou a trabalhar no Minix, variante do UNIX que planejava usar para se comunicar de casa com o UNIX da faculdade, emulando seu terminal. Foi aí que notou o quanto o emulador do Minix era ruim. Então, começou a escrever seu próprio emulador. Este criou pernas, pois Linus logo sentiu necessidade de escrever também um driver de disco e um de sistema de arquivos. Percebeu que ia ter um trabalhão, mas o frio continuava, nada havia para fazer... e entrou num esquema "programar-dormir-programar-comer-programar-tomar banho...". Em julho de 1991, parte do sistema estava ficando pronta, quando Linus concluiu que seria importante seguir padrões e passou a procurá-los em listas (*news*).

O inverno passou, a primavera e o verão também, e, em 17 de setembro de 1991, Torvalds fez o upload da versão 0.01 do sistema para um site ftp da universidade. Ele não queria o nome Linux, pois achava egocêntrico demais (pensava chamar o sistema de Freax), mas Ari Lemke, o professor que montou o site ftp, acabou batizando-o oficialmente.

A partir do momento em que o código fonte foi disponibilizado publicamente na Internet, sem custos, o desenvolvimento teve um grande impulso. Qualquer pessoa interessada tinha acesso para usar, testar e, se pudesse, auxiliar com correções e/ou novos códigos. Isto é chamado de código aberto de uso livre.

A licença de uso foi a mesma do projeto GNU, que garante que qualquer código inserido continuará sendo livre. Desde 1984, o projeto GNU, mantido pela *Free Software Foundation (FSF)*, desenvolve um sistema operacional livre similar ao UNIX. O próprio autor do projeto, Richard Stallman, escreveu: “GNU, que significa gnu, não é UNIX, é o nome para um sistema de software completo e compatível com o UNIX, que estou escrevendo para fornecê-lo gratuitamente a todos que queiram utilizá-lo”. A escolha de palavras foi descuidada. A intenção era que ninguém pagasse pela “permissão” para usar o sistema GNU.

Este é o primeiro “mito” em relação ao projeto GNU e ao Linux, pois as palavras não deixam isso claro e as pessoas freqüentemente as interpretam como: as cópias do GNU serão sempre distribuídas gratuitamente ou por um valor simbólico. Entretanto, a intenção nunca foi essa. Lendo atentamente o “manifesto GNU”, notamos que menciona a possibilidade de as empresas fornecerem o serviço de distribuição objetivando o lucro.

Devemos distinguir cuidadosamente entre "free" no sentido de liberdade e "free" no sentido de preço. O Software Livre (*Free Software*) é aquele que os usuários têm liberdade de distribuir e modificar. Alguns usuários obtêm cópias sem custo, enquanto outros pagam para receber cópias – e se a receita ajuda a aperfeiçoar o software, melhor ainda. O mais importante é: quem tem uma cópia pode cooperar com outras pessoas utilizando o software. Esta é a base da Licença Pública Geral (GPL) aplicada ao uso do software livre.

Quando Linus escreveu seu código do núcleo do sistema operacional, o projeto GNU já estava bem avançado. Por isso, usou uma série de programas do GNU para que seu ambiente funcionasse. Por exemplo, o interpretador de linha de comando (*shell*), o compilador de linguagem “C” (GCC), entre dezenas de outros.

Aí reside um segundo mito: o núcleo do sistema operacional escrito por Linus, e que foi batizado de LINUX, nome que ficou famoso, corresponde apenas a aproximadamente 3% dos programas que rodam quando se usa um computador com Linux. Todo o resto, em sua maioria, é constituído por programas oriundos do projeto GNU. O núcleo pode ser considerado a parte mais importante, porém, sem os demais programas, não haveria o sistema operacional, tal como o conhecemos. Então, para ser ético e coerente, o mínimo a fazer é chamar este sistema operacional de **GNU/Linux.**

Texto 2 – As distribuições

O que é uma distribuição (ou “distro”, para os mais íntimos)?

Já que tudo é baseado em software livre, qualquer pessoa pode:

- pegar uma versão do kernel do Linux, mais uma centena de programas de diversos autores, e reuni-las de modo que tudo funcione como um sistema operacional;
- criar programas que facilitem/otimizem e tornem bonita a instalação;
- colocar tudo organizado disponível em um site de FTP/http;
- gerar uma imagem de CD que dê boot e apresente a tela de instalação... (ou mesmo mandar gravar cópias desses CDs); e...
- distribuir!

Este é, resumidamente, o caminho para criar uma nova distribuição, com gosto e objetivos pessoais. Embora seja um processo trabalhoso, requer apenas conhecimento e horas de trabalho; nada terá que se pagar pela licença de uso por ter reunido os diversos programas, desde que todos sejam “software livre”. Você também poderá criar a sua distribuição!

Desde a criação do Linux até 2004, já surgiram 235 distribuições.

Mais um mito: qual foi a distribuição pioneira de Linux, a primeira a abrir caminho para os muitos sabores de pingüim existentes hoje (Red Hat, SuSE, Conectiva, Mandrake...)? Nenhuma delas. Foi o Slackware, não por acaso, o sistema preferido por quem lida com o Linux desde o começo da década de 90.

A história do Slackware se confunde com a do Linux. Numa festa, o *geek* Patrick Volkerding ficou sabendo da existência do Linux e resolveu baixar da Internet uma versão do pingüim, o SLS (*SoftLanding Linux System*). Na época, o kernel do SO ainda era o 0.98. Patrick fez vários patches no SLS e ofereceu uma nova versão através de um ftp. O sistema estava tão diferente do original que sugeriram a Patrick mudar seu nome para Slackware. Em 17 de julho de 1993, então, foi postada a versão 1.0.0.

Texto 3 - Kernel, shell e aplicações

É importante que estes conceitos, vistos na disciplina Computadores e Redes de Computadores, estejam na memória (na sua, não no computador!). Aproveite para relembrá-los!

Os programas que rodam em uma máquina com o sistema operacional GNU/Linux podem ser classificados como sendo do **kernel**, um **shell** ou uma **aplicação de usuário**.

Kernel

Toma conta e gerencia todos os recursos da máquina. Em processadores da família i386[1], o kernel roda em modo protegido, de forma que o controle sobre o hardware é completo: com o auxílio da CPU nada passa despercebido do kernel. Os acessos aos recursos de hardware (memória, CPU e dispositivos de entrada/saída) são controlados pelo kernel.

Shell

É o responsável por interpretar os comandos do usuário. Existe uma série de "sabores" de shell disponíveis para o usuário escolher. Por exemplo: sh, ash, ksh, zsh, bash, csh, tcsh. Cada um tem suas particularidades que melhor se adaptam às necessidades de uso e, principalmente, ao gosto do usuário. Por isso existem tantos; cada autor personalizou o seu de acordo com gosto e objetivo próprios.

Hoje, o mais utilizado, e que é o padrão da instalação do GNU/Linux, é o Bash. O Bash é o shell, ou interpretador de comando que apareceu no SO GNU e apresenta as seguintes características:

- é compatível com o "sh" (o primeiro interpretador de comandos do UNIX) e incorpora funções interessantes do *Korn Shell* (ksh) e do *C shell* (csh);
- presta-se a ser compatível com o padrão para shell e ferramentas "IEEE POSIX P1003.2/ISO 9945.2";
- oferece melhorias funcionais se comparado ao sh, tanto para programação quanto para uso interativo. Assim, muitos scripts[2] escritos para o sh podem

1 i386 indica processador compatível com Intel 80386.
2 Script: conjunto de comandos de uma linguagem interpretada (no caso, do shell), escritos em um arquivo texto. Ao ser executado cada linha do arquivo texto é interpretada e executada em seqüência, na ordem em que está escrita.

ser rodados pelo bash sem modificação.

Aplicativo de usuário

Qualquer programa que não seja do kernel ou um shell, faça parte ou não da distribuição.

Como o kernel roda em modo protegido, traz como conseqüência o fato de que um aplicativo de usuário não consegue realizar ações maléficas ao sistema ou a outro programa, sem ser detectado e parado. Se um programa causar uma exceção de falha geral (13) no processador, o kernel simplesmente derruba o aplicativo que causou a falha e toma de volta todos os recursos que estavam alocados para este aplicativo.

Na verdade, o controle é ainda mais rígido: a ação que dispara uma exceção de falha geral não precisa ser necessariamente maléfica a outros programas. Basta ser uma ação que não esteja previamente disponibilizada para o programa em questão. Por exemplo, tentar acessar um endereço de memória que não faz parte de um bloco previamente pedido e alocado para o programa pelo kernel. Mesmo que o endereço esteja sem uso, livre, o programa que tentou acessá-lo será derrubado, pois não tinha autorização explícita para tal acesso. O mesmo vale para acesso a dispositivos de entrada e saída (I/O). Esta forma rígida de operar torna o sistema muito estável e explica por que existem máquinas ligadas e sem dar boot há anos .

Texto 4 – Usuário comum X Administrador do sistema

A conta de administrador do sistema possui, por padrão, o login **root** nos sistemas GNU/Linux (como em outros sistemas UNIX). Essa conta também é chamada de **super usuário**. Trata-se de um login sem restrições de segurança, ou seja, com ele pode-se fazer tudo, mesmo que no sistema de permissões de acesso esteja configurada uma proibição. Por isso, a **conta root** somente deve ser usada para fazer a **administração do sistema**, e durante o menor tempo possível, principalmente quando o sistema estiver instalado no HD (com possibilidade de escrever, apagar e cometer danos irreparáveis).

Em geral, por possuir “poderes ilimitados”, a conta root é a única que se diferencia das demais contas de usuários, que seguem o estipulado pelo sistema de permissão de acesso.

Veremos estes tópicos em detalhes adiante. No momento é importante ter em mente que a conta root só é usada para fazer alterações no sistema, tais como:

- instalação de novo software ou upgrade;
- manutenção e alteração de configuração;
- inclusão ou eliminação de usuários.

> Nunca se usa a conta root para navegar na Internet, passar email ou usar qualquer tipo de chat.

Um usuário comum somente altera (cria, modifica ou apaga) arquivos e dados de sua própria conta. Em um sistema GNU/Linux bem configurado e atualizado, ele não consegue causar danos ao sistema ou a contas de outros usuários. Seja por acidente ou intencionalmente, um usuário comum não prejudica (por exemplo, alterando o que não devia) outro usuário, nem o próprio sistema. Já com a conta root pode-se, por exemplo, apagar todo o disco, com um simples comando!

Texto 5 – Nomenclatura de dispositivos

Tudo no GNU/Linux/UNIX é acessado como a um arquivo.

Às vezes, tal afirmativa aparece como “tudo no Linux é arquivo”. Ela é baseada no fato de que diretórios, arquivos e dispositivos de entrada e saída são tratados como nomes de arquivos em sistemas UNIX[3]. O GNU/Linux também usa este conceito: impressoras, modems, portas seriais, discos, disquete, CD/DVD... Tudo possui um nome que está na árvore de diretórios do UNIX, debaixo do **diretório raiz “/”** (o barra). Isto é similar ao que vem sendo feito recentemente, ao colocar tudo abaixo de “meu computador”. No UNIX, sempre foi assim.

Todos os dispositivos de E/S estão dentro do diretório **/dev/**, que pode ser considerado especial. Por exemplo, /dev/ttyS0 é o nome da porta serial S0. No DOS, é chamada de CON1:. A primeira porta paralela de impressora se chama /dev/lp0 (LPT1: no DOS).

3 Neste contexto, tudo que estiver relatado para UNIX se aplica ao Linux.

Os discos rígidos possuem uma nomenclatura bastante similar. No UNIX não existem letras (C, D etc.) para denominar os discos.

No entanto, no quesito de nomenclatura de dispositivos, cada sabor de UNIX tem sua própria regra para dar nomes a eles. Cada fabricante garante que seu sistema de nomenclatura é o melhor. Não entraremos nesta discussão e vamos nos limitar a apresentar uma introdução à nomenclatura no GNU/Linux. Qualquer outro UNIX terá nomes diferentes, esteja avisado. Felizmente, todas as distribuições de GNU/Linux usam a mesma nomenclatura, por ser uma função do kernel do sistema operacional.

No DOS, não havia distinção entre disco, partição e unidade lógica (sistema de arquivo formatado). Por exemplo: instala-se o disco C:, cria-se a partição C:, formatada a partição, acessa-se o C: etc. São coisas diferentes que têm o mesmo nome. No GNU/Linux, o nome que o disco recebe já depende de sua tecnologia: se for um disco IDE (paralelo – PATA), seu nome começa com /dev/hd. Se for um SCSI ou um Serial ATA, o nome começa com /dev/sd. A seguir vem uma letra, que depende da controladora e configuração *master/slave*, ou da posição configurada (número do disco na controladora SCSI). A Tabela 1 mostra os nomes dos discos IDE, dependendo da controladora e da configuração mestre/escravo (*master/slave*) do IDE.

Controladora	Master/slave	Nome do disco
Primária	Master	/dev/hda
Primária	Slave	/dev/hdb
Secundária	Master	/dev/hdc
Secundária	Slave	/dev/hdd

Tabela 1: Nomenclatura de discos IDE no Linux.

Assim, para um disco IDE de interface paralela (PATA), a posição "física" determinada pela controladora a que está ligado e o *strap master/slave* determinam o nome do dispositivo, independente de haver ou não outros discos instalados na máquina.

Para discos SCSI, a ordem numérica dos discos dará as letras em seqüência, começando por a (/dev/sda, /dev/sdb, /dev/sdc etc.). Tal seqüência é conseqüência da configuração do número do disco (configurado por *strap* no próprio disco). Estes números podem ser reconfigurados a qualquer momento, facilitando a inserção de discos. Não é necessário configurar os números de modo consecutivo. Por exemplo, pode-se ter quatro discos com os números 0, 1, 4 e 5.

Nos discos SATA, a nomenclatura depende da ordem de reconhecimento do disco pela controladora. O primeiro disco será o **/dev/sda** e assim por diante.

Uma máquina com um disco IDE ligado como *master* na controladora secundária e um disco SATA, terá os discos **/dev/hdc** e **/dev/sda**. Entretanto, se o IDE for colocado na controladora primária, teremos **/dev/hda** e **/dev/sda**.

Qual a conseqüência? Por exemplo, no DOS, ter dois discos instalados e com uma partição cada, eles serão chamados de C: e D:, não importa se o segundo disco está como *slave* na controladora primária ou como *master* na controladora secundária. No segundo caso (segundo disco como *master* na controladora secundária), se for instalado um terceiro disco como *slave* na controladora primária, este novo será chamado de D: e o antigo D: será automaticamente renomeado para E:. Com isso, todos os links internos ficarão confusos.

No GNU/Linux, o HD instalado como *master* na controladora secundária sempre será o **/dev/hdc**, tendo ou não outros discos.

Uma vez nomeados os discos, falta nomear as partições. A nomenclatura é feita com números que contam a partição, sendo 1 (um) a primeira. Assim, a partição chamada de C: no DOS, no GNU/Linux será a **/dev/hda1**.

Texto 6 – Processos

Todo programa rodando em um UNIX passa a ser chamado de processo. No UNIX, onde tudo é controlado pelo kernel, este dá ao processo uma identificação numérica única: *Process ID* (PID).

Um processo executa de duas formas:

- **primeiro plano** – também chamado de *foreground*. Se iniciado na linha de comando, deve-se esperar o término da execução, para executar um novo comando. O aviso de comando (*prompt*) só é mostrado após o término de execução do comando/programa;
- **segundo plano** - também chamado de *background*. Neste caso, o aviso de comando é mostrado imediatamente. Após iniciar um programa em *background*, é mostrado um número PID (identificação do processo) e o aviso de comando é novamente mostrado, permitindo o uso normal do sistema.

O programa executado em background continua sendo executado internamente. Após ser concluído, o sistema retorna uma mensagem de pronto acompanhado do número PID do processo que terminou.

Para iniciar um programa em `primeiro plano`, basta digitar seu nome normalmente. Para iniciar um programa em `segundo plano`, acrescenta-se o caractere "&" ao fim do comando.

Obs.: Mesmo que um usuário execute um programa em segundo plano e saia do sistema, o programa continuará sendo executado até que seja concluído ou finalizado pelo usuário que iniciou a execução (ou pelo usuário root).

As mensagens que o comando venha a escrever aparecerão na tela normalmente, até quando executando em segundo plano. Isto pode trazer algum desconforto e confusão na tela, ao se misturar mensagens de programas executando em segundo plano com o que estiver sendo realizado no console.

Exemplo:

```
find / -name boot.b &
```

O comando será executado em segundo plano e deixará o sistema livre para outras tarefas. Após o comando `find` terminar, será mostrada uma mensagem indicando o seu término.

Os comandos “ps” e “top” são muito úteis para administração de processos. Recomendamos que você estude nos manuais (man) e na bibliografia sugerida.

Texto 7 – Download e preparação do CD

Para obtermos uma distribuição GNU/Linux, é necessário fazer o download da imagem ISO do CD de instalação. A maior parte das distribuições disponibiliza em suas páginas oficiais na Internet uma lista de espelhos (*mirrors*), em que essas imagens podem ser baixadas.

Daqui em diante, para exemplificação, adotaremos a distribuição Slax – pode ser obtida no site http://slax.linux-live.org/download.php. Trata-se de um live-CD baseado na distribuição Slackware que utilizaremos nos demais exemplos do curso.

Alguns sistemas podem associar a extensão .ISO como um de seus arquivos de software de compactação (por exemplo, winrar). Apesar disso, não descompacte o arquivo, pois ele não funcionará. Grave exatamente como veio a imagem, conforme exemplo na Figura 1.

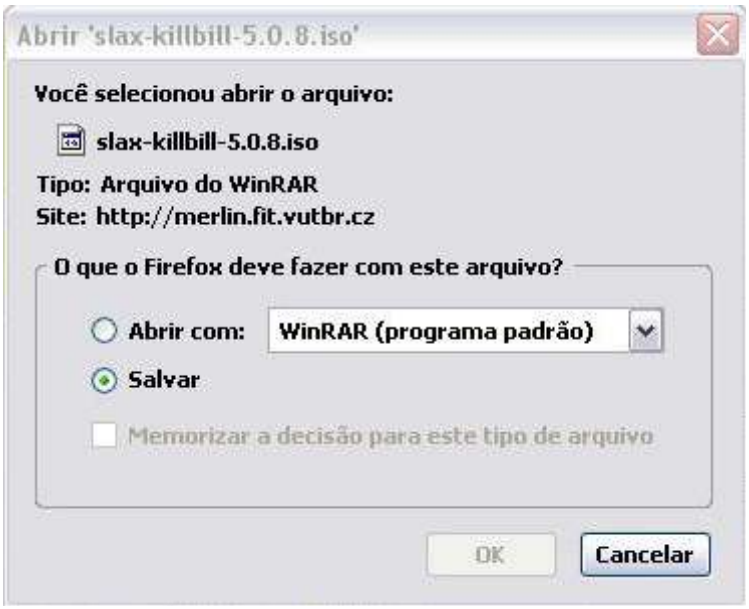


Figura 1 - Baixando a imagem do SLAX

Durante o *download* de arquivos da Internet, existe a probabilidade de ocorrerem erros de transmissão, o que causaria a possibilidade do CD gravado não funcionar. Por isso, junto com a imagem do CD, é disponibilizado um código md5 do arquivo. É um número gerado com o comando md5sum do GNU/Linux. Se o arquivo estiver corrompido, o código md5 gerado localmente será diferente do original.

Para conferir o código, é necessário um programa que trabalhe como o comando md5sum do GNU/Linux. Em http://www.md5summer.org/download.html você encontra um programa destes, para Windows. A Figura 2 mostra o programa após a verificação concluída.

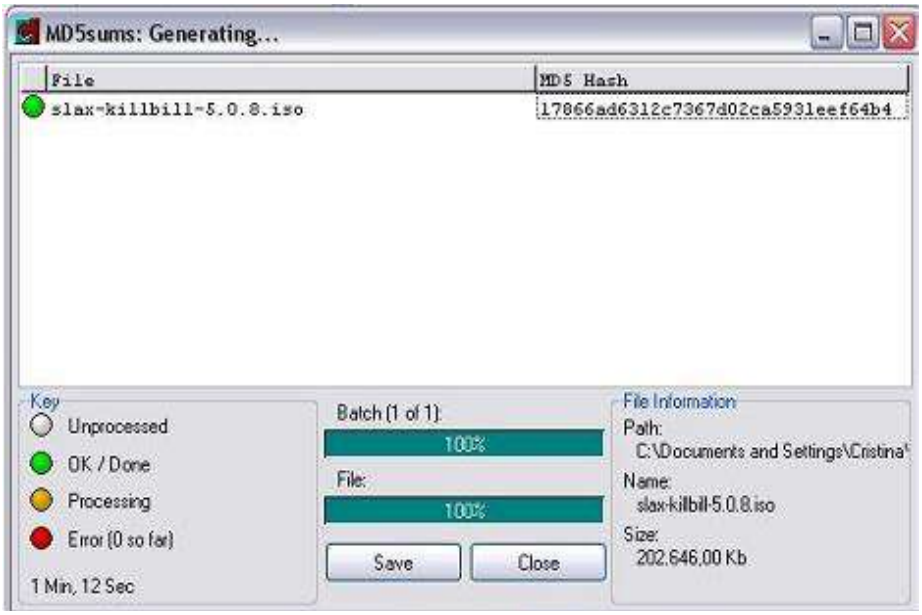


Figura 2 - Verificação de integridade com MD5

Depois de feito o download e garantida sua integridade, é preciso gravar um CD com o arquivo ISO. A imagem ISO é uma imagem binária do CD de instalação e não funcionará se for apenas gravada como um arquivo comum. Para gravar uma imagem ISO, escolha a opção “burn image” ou “gravar imagem iso” no software de gravação de CD preferido. Veja o exemplo da Figura 3.

Figura 3 - Gravação da imagem (ISO) do CD

Texto 8 – Instalando o Linux

Para rodar o Linux, é preciso dar boot pelo CD gravado. A maioria dos computadores suporta dar boot pelo CD, porém parte deles vem configurada para procurar por boot primeiramente no hard disk (HD).

Para garantir que o boot ocorrerá pelo CD, entre no *setup* da *bios* do computador (usualmente apertando a tecla DEL durante a inicialização do computador). Procure em "*advanced setting*" ou "*Bios Features setup*" a opção "*boot sequence*" ou "*first device*". Configure-a para que o boot pelo CD-ROM ocorra antes do boot pelo hard disk.

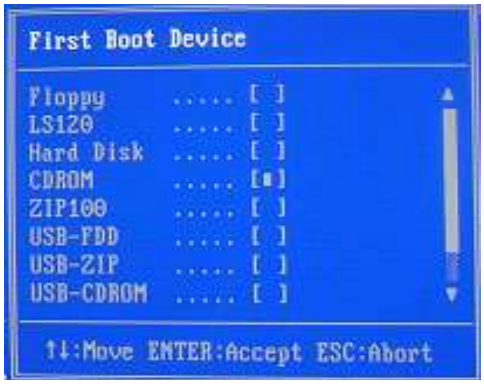


Figura 4 - Boot pelo CD

A Figura 4 mostra um exemplo. Saia do *setup* da *bios* salvando (opção F10, normalmente). O computador deve reiniciar e rodar o CD logo após o teste da memória RAM.

Os processos que serão visualizados na tela envolvem o seguinte: durante a inicialização, o Slax irá carregar a primeira imagem do kernel Linux. Depois, um disco virtual de 4 MB é criado na memória RAM do computador. A imagem do Rootdisk é descompactada dentro do disco virtual e, então, é montada como o sistema de arquivos raiz. Descreveremos estes termos oportunamente. Diretórios maiores são montados diretamente do CDROM (apenas em modo leitura).

A maior parte da configuração do computador será verificada e o live-CD a auto instalará.

Durante o processo de boot, aparece o logotipo do Slax seguido por diversas inicializações. Ao fim do processo, o console mostra uma imagem de login. Entre

com o login e senha indicados na tela (login: *root* e senha: *toor*).

A partir desse momento, você já estará rodando o Linux pelo CD, podendo explorar a configuração tanto no modo texto como no modo gráfico.

Para iniciar o modo gráfico, digite “startx”. Se o modo gráfico não funcionar, tente digitar “xconf” para que as placas de vídeo, monitor, teclado e mouse sejam configuradas.

Texto 9 – Utilizando o sistema

A interface mais usual do Linux é o shell; basicamente um interpretador de comandos em modo texto. É o shell que executa comandos lidos do dispositivo de entrada padrão (teclado) ou de um arquivo executável. Trata-se da principal ligação entre usuário, programas e kernel.

Os principais exemplos que daremos estão direcionados para administradores de sistema utilizando o shell em linha de comando. Normalmente, os administradores preferem as linhas de comando por serem mais ágeis do que seqüências intermináveis de clique-clique. Além disso, no mundo de quem administra, a similaridade entre comandos de linha dos diversos sistemas UNIX e Linux é grande, sendo a maioria idêntica. Já as interfaces gráficas de gerenciamento mudam muito de um UNIX ou Linux para outro. Por isso, um bom administrador não se apega à interface gráfica, pois terá seu conhecimento limitado a somente este ou aquele sistema. O bom administrador conhece “UNIX”.

Existem dois tipos de execução de um shell:

- **interativo**

  Os comandos são digitados no aviso de comando e passados ao interpretador de comandos um a um. Neste modo, o computador depende do usuário para executar uma tarefa ou o próximo comando.

- **não interativo**

  O usuário cria arquivos de comandos (scripts). O computador executa os comandos na ordem encontrada no arquivo. Neste modo, o computador executa os comandos do arquivo um por um e, dependendo do término do

comando, o script pode checar qual será o próximo comando a ser executado e dar continuidade ao processamento.

Há várias versões de shell disponíveis para Linux. Porém, a mais utilizada atualmente é o bash.

O bash é mostrado em modo texto no console ou terminal. Em modo texto, você pode acessar outros terminais virtuais. No Slackware temos seis terminais preconfigurados. Assim, segurando a tecla ALT e pressionando de F1 a F6, acessam-se os terminais. Cada tecla de função corresponde a um terminal, do 1 ao 6, o sétimo é usado por padrão pelo ambiente gráfico X. O GNU/Linux permite mais de 63 terminais virtuais, mas apenas seis estão disponíveis inicialmente devido à economia de memória RAM.

O prompt do console é configurável. No padrão do Slackware, ele indica que usuário está usando o sistema e a máquina e, ainda, o diretório em que o usuário se encontra, desta forma:

```
[<usuário>@<nome da máquina> <diretório atual>]$
```

No caso do usuário normal, o sinal final é um cifrão ($). Se o usuário for o root, é uma tralha (#).

Um comando do shell é um arquivo que executa uma função no Linux. Como boa parte dos comandos básicos do shell são programas provenientes do projeto GNU, eles compartilham algumas características comuns, como a sintaxe, tendo o seguinte formato:

```
comando [opção] <parâmetro>
```

A seguir, relacionamos os comandos mais usuais do shell, agrupados por suas funções. Estude-os com a ajuda do manual. Tenha em mente que este é só um início, com poucos comandos. Um GNU/Linux/UNIX típico possui mais de 2.000 (sim, dois mil) comandos. Mas, saber os básicos e como procurar os demais, é o suficiente e necessário para ser um ótimo administrador.

É importante que você teste no GNU/Linux os comandos apresentados nos textos a seguir.

Texto 10 – Comandos de ajuda

Dúvidas são comuns durante o uso do GNU/Linux e existem maneiras de obter ajuda e encontrar a resposta para algum problema. O GNU/Linux é um sistema bem documentado, provavelmente tudo o que você imaginar fazer ou quiser aprender já está disponível para leitura e aprendizado.

man

As *páginas de manual* acompanham quase todos os programas GNU/Linux. Elas trazem uma descrição básica do comando/programa e detalhes sobre o funcionamento da opção. Uma página de manual é visualizada na forma de texto único com rolagem vertical. Também documenta parâmetros usados em alguns arquivos de configuração.

```
man [seção] comando/arquivo
```

Onde:

- *seção* é um número que indica a seção do manual onde o comando será procurado. É opcional. Sem este número, a procura é seqüencial a partir da seção 1.Cada seção engloba os manuais referentes a um assunto, conforme a lista:

1- Comandos do usuário;

2- Chamadas ao sistema;

3- Bibliotecas de funções;

4- Dispositivos;

5- Formatos de arquivos;

6- Jogos;

7- Informações gerais;

8- Administração do sistema;

9- Programação em geral.

Se omitido o número da seção, é apresentada a *primeira* seção encontrada sobre o comando, sendo que o man procura em ordem crescente.

– *comando/arquivo* , como o nome diz, é o comando/arquivo que se deseja pesquisar.

info

Idêntico às páginas de manual, mas usa navegação entre as páginas. Se pressionarmos <Enter> em cima de uma palavra destacada, a info pages nos levará à seção correspondente. A *info pages* é útil quando sabemos o nome do comando e queremos saber para que ele serve. Também traz explicações detalhadas sobre uso, opções e comandos.

```
info comando/programa
```

help on line

Trata-se de uma ajuda rápida, quando não precisamos dos detalhes das páginas de manual. Serve para identificarmos quais opções podem ser usadas com o comando/programa. Quase todos os comandos/programas GNU/Linux oferecem este recurso, útil, por exemplo, quando sabemos o nome do programa e desejamos saber quais são as opções disponíveis e para que cada uma serve.

```
Comando --help
```

help

Ajuda rápida, útil para saber que opções podem ser usadas com os *comandos internos* do interpretador de comandos. Note que mostra ajuda apenas para comandos internos do sistema operacional.

```
help comando
```

apropos/whatis

O comando apropos (do francês, “sobre”) procura por *programas/comandos* nas páginas do manual que trazem alguma menção na descrição do comando com a palavra fornecida na descrição do usuário. É útil quando você tem idéia do que necessita fazer, mas não sabe qual comando usar.

```
apropos/whatis descrição
```

Texto 11 – Comandos de manipulação de arquivo

cat

Originalmente criado para fazer a con**cat**enação entre dois ou mais arquivos texto, é muito usado para visualizar o conteúdo de um arquivo binário ou texto.

```
cat [opções] [diretório/arquivo] [diretório/arquivo]
[diretório/arquivo] ...
```

rm

Apaga arquivos. Também pode ser usado para apagar diretórios e sub-diretórios vazios ou que contenham arquivos.

```
rm [opções][caminho]arquivo/diretório
```

cp

Copia arquivos.

```
cp [opções] origem destino
```

mv

Move ou renomeia arquivos e diretórios. O processo é semelhante ao do comando cp, mas o arquivo de origem é apagado após o término da cópia.

```
mv [opções] origem destino
```

less

Permite fazer a paginação de arquivos ou da entrada padrão. Pode ser usado como comando para leitura de arquivos que ocupem mais de uma tela. Quando toda a tela é ocupada, o less efetua uma pausa e permite que você pressione Seta para Cima e Seta para Baixo ou PgUP/PgDown para fazer o rolamento da página.

```
less arquivo
```

tail

Mostra as linhas finais de um arquivo texto. Com um número (-N) em opções, lista as N últimas linhas. Com -f, mostra o fim do arquivo e o mantém aberto, apresentando as modificações que ocorreram. É útil para acompanhar logs em tempo real.

```
tail [opções] arquivo
```

Texto 12 – Comandos de manipulação de diretório

ls

Lista os arquivos de um diretório.

```
ls [opções] [caminho/arquivo]
```

cd

Entra em um diretório, mas requer que você tenha permissão de execução.

```
cd [diretório]
```

pwd

Mostra o nome e caminho do diretório atual.

```
pwd
```

mkdir

Cria um diretório no sistema. Um diretório é usado para armazenar arquivos de um determinado tipo. O diretório pode ser entendido como uma *pasta* onde se guardam papéis (arquivos). Para manter a organização, deve-se utilizar uma pasta para guardar cada tipo de documento, dando nomes coerentes aos diretórios. Por exemplo, cria-se um diretório vendas para naquele local guardar arquivos relacionados a vendas.

```
mkdir [opções] caminho/diretório
```

rmdir

Remove um diretório do sistema, ou seja, faz exatamente o contrário do mkdir. O diretório deve estar vazio e, para removê-lo, é necessário ter permissão de gravação.

```
rmdir caminho/diretório
```

Para remover diretórios que contenham arquivos, use o comando **rm** com a opção **-r**

Texto 13 – Comandos de manipulação de discos e partições

fdformat

Formata fisicamente disquetes.

```
fdformat caminho dispositivo
```

O caminho do dispositivo do disquete é /dev/fdx .

mkfs

Formata logicamente o sistema de arquivos.

```
mkfs [formado do tipo de partição] caminho do dispositivo
```

mount

Acessa uma partição do disco ou dispositivo de armazenamento.

```
mount [dispositivo] [ponto de montagem] [opções]
```

- *dispositivo* é a identificação da unidade de disco/partição que se deseja

acessar (como /dev/hda1 (disco rígido) ou /dev/fd0 (primeira unidade de disquetes).

- *ponto de montagem* é o diretório de onde a *unidade de disco/partição* será acessada. O diretório deve estar vazio para montagem de um sistema de arquivo. Normalmente usa-se o diretório /mnt para armazenamento de pontos de montagem temporários.

umount

Para desmontar um sistema de arquivos montado com o comando mount. Requer permissões de root para desmontar uma partição.

```
umount dispositivo/ponto de montagem
```

df

Lista as partições e quanto destas partições está ocupado por dados. É originário de *Disk Free*.

```
df
```

Texto 14 – Comandos de manipulação de permissão

Estes comandos serão vistos com mais detalhes no assunto permissão de acesso. Ficam aqui como referência para consulta.

chmod

Muda a permissão de acesso a um arquivo ou diretório.

```
chmod [opções] permissões diretório/arquivo
```

onde, em permissões, usa-se:

ugoa+-=rwxXst

- *ugoa* – controla que nível de acesso será mudado. Especifica, em ordem, usuário (u), grupo (g), outros (o), todos (a).
- +-= – coloca a permissão (+), retira a permissão do arquivo (-), define a permissão exatamente como especificado (=).
- rwx – permissão de leitura do arquivo (r), permissão de gravação (w), permissão de execução (ou acesso a diretórios) (x).

Em vez de utilizar os modos de permissão +r, -r etc., pode ser usado o modo octal para se alterar a permissão de acesso a um arquivo. O modo octal é um conjunto de oito números em que cada número define um tipo de acesso diferente.

- 0 - Nenhuma permissão de acesso. Equivalente a -rwx.
- 1 - Permissão de execução (x).
- 2 - Permissão de gravação (w).
- 3 - Permissão de gravação e execução (wx).
- 4 - Permissão de leitura (r).
- 5 - Permissão de leitura e execução (rx).
- 6 - Permissão de leitura e gravação (rw).
- 7 - Permissão de leitura, gravação e execução. Equivalente a +rwx.

chgrp

Muda o grupo de um arquivo/diretório.

```
chgrp [opções] grupo arquivo/diretório
```

chown

Muda o dono de um arquivo/diretório. Opcionalmente também pode ser usado para mudar o grupo.

```
chown [opções] dono[grupo] diretório/arquivo
```

Texto 15 – Comandos de gerenciamento de processos

ps

Lista os processos que estão sendo usados no computador.

```
ps [opções]
```

pstree

Mostra a árvore de processos, identificando visualmente as relações entre processos pais e filhos.

```
pstree [opções] [pid/usuário]
```

top

Programa interativo de monitoração e gerenciamento de processos. Apresenta os programas em execução ativos, parados, tempo usado na CPU, detalhes sobre o uso da memória RAM, Swap, disponibilidade para execução de programas no sistema etc.

```
top [opções]
```

fuser

Lista o pid dos processos que estão acessando determinado arquivo ou socket. Útil para detectar problemas de concorrência de acesso ou bloqueio de dispositivos.

```
fuser [opções] [arquivo]
```

kill

Permite enviar um sinal a um comando/programa.

```
kill [opções] [sinal] pid
```

A lista de todos os sinais pode ser obtida com a opção -l.

Se nenhum sinal for informado, o kill envia o SIGTERM (-15) ao programa. Trata-se de um pedido de finalização, dando-lhe a chance de salvar os dados que estão em memória.

O extremo dos sinais é o SIGKILL (-9) . Derruba imediatamente o programa, perdendo-se todos os dados em memória.

Unidade 2 – Noções de Segurança

Nesta Unidade você verá diversos aspectos conceituais da segurança em sistemas sistemas operacionais para servidores de redes de computadores e, ainda, detalhes de configuração que, uma vez examinados e corrigidos, podem aumentar a segurança e minimizar futura possibilidade de comprometimento do sistema.

Texto 16 – Segurança física

Os primeiros computadores ocupavam prédios ou andares inteiros e todos os usuários tinham que submeter "jobs", por meio de cartões de papelão perfurados, em que cada cartão continha uma linha de comando. Seu programa tem três mil linhas? Então, prepare o carrinho para carregar três mil e poucos cartões. O usuário entregava os cartões para um funcionário, que os levava para a leitora de cartões (ninguém sabia exatamente onde). Depois, o computador imprimia os resultados e as folhas eram colocadas em escaninhos na sala das perfuradoras de cartões. Assim, o usuário final nem chegava perto do computador (*mainframe*). Naqueles tempos, a segurança de acesso físico era facilitada pela total falta de interatividade entre usuário e computador: bastava um guarda na porta.

Hoje, com a proliferação dos microcomputadores, pode-se vê-los em qualquer local. Em instalações profissionais, os principais computadores, que contêm ou concentram os softwares e informações importantes, são, normalmente, fechados em um local especialmente designado e projetado: a sala dos servidores. Nelas tem-se o cuidado de manter alguns quesitos ambientais, como umidade e temperatura, dentro de determinados limites e o mais estável possível. Logicamente, a entrada é protegida por fechaduras (eletrônicas, em muitos casos). Como antigamente, poucas pessoas têm acesso físico a um servidor, ou à sala deles.

O acesso físico indevido pode causar ou permitir a paralisação dos serviços e/ou roubo ou adulteração de informações.

Mas, e nos locais onde ainda não foi criada a sala dos servidores? O acesso físico indevido pode causar ou permitir a paralisação dos serviços e/ou roubo ou adulteração de informações, ocasionado por:

- desligamento da rede elétrica (sim: desligar o micro da tomada! É fácil parar uma empresa!);
- desligamento da rede de dados;
- reset do equipamento;
- boot no modo de segurança (Windows), ou no modo de administração (*single user*) em UNIX/Linux. Algumas distribuições de GNU/Linux permitem acesso como root, SEM senha, ao se entrar em *single user mode* (modo de administração);

- boot com mídia externa (CD ou disquete) que permita acesso ilimitado a todo o conteúdo armazenado naquela máquina – é fácil fazer com um CD de GNU/Linux em qualquer microcomputador, e olhar/copiar/alterar dados em qualquer sistema (UNIX/Windows);
- roubo de componentes (disco rígido, por exemplo);
- roubo do servidor inteiro – por mais incrível que pareça, existem relatos assim: cheguei pela manhã e não consegui entrar no sistema, a partir de meu micro. Tentei no do vizinho e em outro. Ninguém conseguia. Então olhamos para a mesa onde ficava o servidor e... ele simplesmente não estava lá.

As informações guardadas em servidores são, normalmente, sigilosas, o que, por si só, justifica que a permissão de acesso físico a estas máquinas seja dada somente para pessoas autorizadas.

Texto 17 – Backup

Um procedimento importante em empresas preocupadas com a disponibilidade dos dados é o backup. Os dados são copiados para outras mídias (outros HDs, fitas magnéticas, CD/DVD). A questão é a seguinte: onde estas mídias são guardadas? É necessário uma constante preocupação com a guarda e qualidade das mídias do backup, pois todos os dados estão disponíveis nelas.

No caso de ocorrer um incidente, e visto que nenhuma empresa está totalmente segura e talvez nunca esteja, recuperar-se desse imprevisto irá requerer backups atualizados e métodos de recuperação dos dados previamente testados.

Algumas empresas fazem backups diários, mas raramente verificam se estão realmente funcionando. Outras criam políticas e procedimentos de backup, mas não de restauração. Freqüente e infelizmente, tais erros são descobertos somente depois de um incidente, seja uma invasão ou perda de dados por falha no equipamento.

Um segundo problema que envolve backups é a falta de proteção física das mídias. Os backups contêm a mesma informação sensível que reside no servidor, portanto devem ser protegidos de igual maneira.

É recomendável fazer um inventário, identificando todos os sistemas críticos e, para cada um, executar uma análise avaliando o risco e a ameaça correspondentes. As políticas e os procedimentos de backup devem ser claramente associados aos sistemas. No caso de sistemas críticos, o processo de validação pode ser efetuado com as seguintes perguntas:

- Existem procedimentos de backup para esse sistema?
- O intervalo dos backups é adequado?
- Os backups estão sendo realizados de acordo com os procedimentos definidos pelo administrador de segurança?
- A mídia usada nos backups é verificada para haver certeza de que os dados estão sendo armazenados corretamente?
- A mídia de backup está corretamente protegida, dentro ou fora da empresa?
- Há cópias do sistema operacional e de aplicativos de restauração armazenadas fora da empresa?
- Os procedimentos de restauração foram validados e testados?

No mínimo, os backups devem ser feitos diariamente. Na maioria das organizações, o requisito mínimo é que sejam executados backups completos semanalmente e backups incrementais[4] diariamente. Ao menos uma vez por mês, a mídia é verificada, restaurando-se os dados em um servidor de teste para comprovar se estão sendo corretamente restaurados.

Essas são as exigências indispensáveis, mas outras podem ser acrescidas. Algumas empresas executam backups completos uma ou várias vezes por dia. Em sistemas de tempo real críticos, por exemplo sistemas financeiros e de comércio eletrônico, a solução consiste em ter redes inteiramente redundantes que implementam tolerância a falhas.

Texto 18 – Usuário

Um usuário em um sistema GNU/Linux é aquele que possui um nome e senha associada, de modo que possa se logar (digitando o login e a senha) no sistema. Do ponto de vista do usuário, trata-se da visão mais simples do que ele é no

4 O programa de backup copia apenas os dados (arquivos) que foram modificados desde o último backup completo.

sistema. Na verdade, é um usuário que possui permissão de login – às vezes, de acesso remoto – e, até, usa seu login como conta de email. É um usuário comum.

A todo usuário está associado um grupo primário (como se fosse a família em que nasceu).

Assim, o mínimo que um administrador cadastra para criar um novo usuário é seu login, a qual grupo primário ele pertence e a senha de acesso.

Internamente, cada login de usuário é associado a um número: o Identificador de Usuário (UID – *User ID*). É este número que é realmente usado pelo sistema operacional para todas as verificações de acesso e que é associado aos arquivos. O login mesmo é apenas uma palavra que só tem valor para nós (principalmente na hora de fornecê-lo, mais a senha, para se logar). No entanto, esta palavra pode até ser trocada pelo administrador, mantendo todas as características da conta[5].

Todo o sistema de permissão de acessos é fortemente baseado em usuários e em grupo de usuários. Por este motivo, todo e qualquer programa que rode na máquina possui um usuário e um grupo associado. O usuário e grupo serão usados em cada tentativa de acesso que este programa fizer, para verificar se têm permissão para realizar o acesso requisitado.

> Todo e qualquer programa que rode na máquina possui um usuário e um grupo associado.

Quando um usuário qualquer roda um programa, seja chamando-o em uma linha de comando ou clicando em uma interface gráfica, o programa roda herdando o usuário que o chamou. Assim, o programa irá tentar fazer acessos como sendo o usuário que o chamou e com o grupo primário deste usuário. Logo, o comportamento de um programa (o que ele pode fazer ou não) depende de qual usuário o está chamando. Veremos mais tarde que isto pode ser mudado, com configuração de permissão de acesso.

A maioria dos sistemas é configurada para usar senhas como a primeira e única linha de defesa. Normalmente, a identidade do usuário é razoavelmente fácil de obter, e as empresas oferecem acesso discado para seus funcionários, logo o firewall que controla o acesso discado permite que esses usuários entrem por fazerem parte do quadro.

Conseqüentemente, se um atacante puder determinar um nome e uma senha de cliente, poderá também ter acesso à rede. Senhas-padrão e fáceis de adivinhar

5 Se isto for feito, o arquivo de grupos terá que ser alterado manualmente, pois nele o login (palavra) é referenciado.

constituem um problema grave e, piores ainda, são as contas sem senha.

Na prática, todas as contas com senhas fracas, senhas-padrão ou sem senhas devem ser removidas do sistema.

Adicionalmente, muitos sistemas ou softwares têm contas-padrão (fazem parte da instalação-padrão), as quais geralmente mantêm a mesma senha em todas as instalações do software. Por exemplo, a conta “system” é criada na instalação do banco de dados Oracle, com uma senha-padrão que qualquer pessoa com um pouco de conhecimento ou consultando manuais pode descobrir. Isso não quer dizer que o Oracle seja um banco de dados que não prime pela segurança, pelo contrário, mas significa que os cuidados devem ser redobrados em qualquer instalação.

Os atacantes procuram, geralmente, essas contas, amplamente conhecidas pela comunidade e eventualmente esquecidas durante a instalação. É fundamental que sejam identificadas e removidas do sistema ou, se for necessário mantê-las, que se troque a senha por outra mais eficiente e desconhecida publicamente.

Para avaliar se existem contas de usuários com senhas em tais condições, é recomendável examiná-las nos sistemas e criar uma lista mestre. Deve-se verificar, também, senhas de sistemas como roteadores e impressoras, copiadoras e controladores de impressora conectados à Internet.

É importante:

- desenvolver procedimentos para adicionar contas pré-autorizadas à lista e para removê-las quando não estão mais em uso;
- verificar a lista regularmente para certificar-se de que nenhuma conta nova foi adicionada sem permissão e as não utilizadas foram removidas;
- utilizar ferramenta de quebra de senhas para identificar contas com senhas fracas ou sem senha – embora essas ferramentas estejam na Internet, de uso irrestrito, é importante verificar a procedência desta e se há permissão oficial por escrito;
- manter procedimentos rígidos para remoção de contas de empregados ou contratados quando saem da empresa ou quando as contas não são mais necessárias.

O ideal é que todas as contas sem senhas recebam uma senha, no caso de não poderem ser definitivamente removidas. Senhas fracas devem ser fortalecidas, mas infelizmente quando tal procedimento é solicitado aos usuários,

freqüentemente eles escolhem outra igualmente fácil de se adivinhar. Assim, torna-se necessário tomar outras medidas: as novas senhas, quando alteradas, também precisam ser verificadas. Há programas específicos que só permitem alterações de senha quando estão de acordo com a política de segurança definida pela empresa.

Uma vez configurados pelo administrador de segurança, com base na política de segurança, esses programas garantem que as senhas tenham comprimento e composição tais que se tornem difíceis de serem quebradas ou descobertas. Diversos fabricantes de sistemas UNIX incluem algum mecanismo de suporte interno para a construção de senhas confiáveis, mas também existem outros pacotes disponíveis no mercado.

Há organizações que adicionam programas de controle de senha. São mecanismos que garantem que as senhas sejam mudadas regularmente e as antigas não sejam reutilizadas.

Se a expiração de senhas for implementada, certifique-se de que os usuários sejam alertados antes que o prazo expire. Diante de uma mensagem do tipo “Sua senha expirou, ela deve ser mudada”, a maioria dos usuários tende a escolher uma senha fraca, por serem pegos de surpresa e o momento não ser oportuno. Uma mensagem do tipo “Sua senha irá expirar em 10 dias” permite ao usuário se organizar, pensar e respeitar as normas da empresa quanto a uma escolha adequada.

O sistema Windows 2000, da Microsoft, inclui opções de restrição de senha no Group Policy. Um administrador pode configurar a rede de forma tal que as senhas de usuários requisitem comprimento mínimo, idades mínima e máxima e outros parâmetros. A idade mínima é interessante, porque sem ela os usuários tendem a mudar sua senha quando requeridos, mas tornam a usá-la logo em seguida. Com uma idade mínima em senhas, o usuário não poderá escolher nem trocar sua senha por outra que tenha sido utilizada durante um intervalo de tempo, o que diminui as chances de voltarem a usá-las no futuro. Isto será visto com mais detalhes adiante.

Quando falamos de senhas, um ponto importante é a conscientização dos usuários para entender “como” e “por que” escolher senhas fortes. É fundamental para o sucesso de uma política forte de senhas e deve atingir todos os níveis, como administradores, usuários, supervisores, diretores etc.

A conscientização será abordada no item Política de Segurança e as senhas, propriamente ditas, no item *Sign-On*. Esse tópico aprofunda a delicada questão de senhas e contém dicas interessantes para uma boa escolha.

Texto 19 – Autenticação do usuário

Autenticação é a capacidade de garantir que alguém, ou alguma coisa, é de fato quem diz ser, dentro de um contexto definido.

Definir o contexto da autenticação é importante porque um determinado usuário ou equipamento pode ser autêntico em uma circunstância, mas pode não ser em outra. Por exemplo, o passaporte autentica uma pessoa dentro de um país, um cartão de acesso autentica o empregado à empresa, um cartão do banco 24 horas autentica somente o dono da conta corrente etc. Da mesma forma, no mundo virtual, um determinado usuário é autêntico dentro de uma Infra-estrutura de Chave Pública (ICP), porém pode não o ser dentro de outra ICP[6].

A autenticação é sempre feita entre duas entidades, o **objeto** (o usuário ou o cliente) que quer acesso à determinada informação e terá ou não sua identidade validada pelo **autenticador** (host ou servidor). Para esse ciclo ser concluído, é necessária que o objeto da autenticação forneça uma informação de autenticação para o autenticador.

Essa informação de autenticação pode ser dividida em quatro categorias:

- **algo que sabemos**, como conjunto de nome do usuário e senha, assim como números PIN[7] usados para acesso a banco 24 horas e combinações de cofres – embora seja o mecanismo mais conhecido, é um dos mais inseguros;
- **algo físico que temos**, como chaves de carro, cartões de banco 24 horas, entre outros que exigem a posse física de um dispositivo;
- **algo que somos**, como impressões digitais, análise de retina, reconhecimento de voz, rosto e padrões de assinatura – esse tipo de tecnologia utiliza a análise de características humanas. A vantagem sobre as outras tecnologias de autenticação é o fato do usuário ser identificado por características únicas, pessoais e intransferíveis, dispensando o uso de

6 O assunto ICP será visto em outra disciplina.
7 Número de Identificação Pessoal (Personal Identification Number).

senhas, cartões ou crachás. É utilizado tanto para controle de acesso físico como para o de acesso lógico.

- **algum lugar onde estamos**, como endereços de adaptador de rede e sistemas baseados em posicionamento global via satélite (*Global Position Satellite* – GPS). Esses mecanismos provêem informação de autenticação com base na localização do usuário.

Chama-se autenticação forte quando combinamos mais de um método de autenticação. Em certos casos, ter somente uma conta de usuário e senha não é suficiente para liberar determinada informação.

Se a informação de autenticação estiver sob controle total das duas entidades (autenticado e autenticador), o esquema de autenticação é chamado de *Two-Party Authentication*. Se utilizarmos uma terceira entidade para validar a autenticidade do interlocutor, esse esquema será chamado de *Trusted Third-Party Authentication*.

Outro fator importante a ser destacado é manter a integridade e confidencialidade da informação de autenticação. É necessário que a informação usada na autenticação seja segura e não possa ser obtida por pessoas não autorizadas.

Veremos nos próximos textos os tipos de autenticação mais relevantes para a Infra-estrutura de Chave Pública: sign-on, single sign-on e autenticação forte.

Texto 20 – Sign- On (SO)

Mais conhecido como *login,* é o processo pelo qual o usuário fornece uma identificação, normalmente um nome de usuário e uma informação de autenticação, como uma senha ou outro valor secreto.

Partindo da premissa de que ninguém, além do próprio usuário, sabe a conta e a senha, esse processo pode ser usado seguramente para autenticar um usuário dentro de um contexto ou de uma aplicação.

O esquema de autenticação por senha é o mais utilizado naquele entre duas entidades, porém o mais fraco em matéria de segurança. Neste esquema, usuário e servidor precisam saber previamente a senha ou a frase, e cabe ao

servidor armazenar a informação para ser comparada no momento da autenticação.

Há aspectos relevantes a serem observados. Em se tratando de senhas, tanto podem ser:

- difíceis de lembrar – conseqüentemente estimulando o usuário a escrever sua senha em algum lugar, o que a torna menos segura;
- fáceis de descobrir, ou seja, como é o usuário que a escolhe, coloca informações pessoais que o ajudem a lembrá-la, como datas de aniversário, placa de carro, matrícula da empresa, nome de filhos, esposa etc., facilitando a memorização, mas pecando em complexidade da escolha.

Um invasor pode estudar o usuário e fazer combinações que o levem a descobrir a senha, ou pode, simplesmente, utilizar o ataque de força bruta, testando várias combinações, com base no dicionário da língua nativa.

Para minimizar o problema, é possível usar uma informação aleatória durante o armazenamento da senha, geralmente chamada de semente (*seed*). Esse paliativo diminui as chances de falha na segurança, mas está longe de resolver outros problemas.

É importante ressaltar que não podemos somente culpar o usuário pela escolha de uma senha fraca. Na maioria dos casos, ele não recebeu orientação alguma, nem há uma política forte de troca de senha na empresa, o que exige cuidados. Mesmo não tendo o propósito de ser uma cartilha, seguem algumas dicas para melhorar a segurança das senhas.

• As contas de usuários demitidos, prestadores de serviço ou as de usuários criadas para testes devem ser imediatamente bloqueadas quando não mais utilizadas.

• Os usuários devem poder alterar a própria senha a qualquer momento e devem fazê-lo caso suspeitem que foi “descoberta”.

• O tamanho mínimo da senha deve ser de oito caracteres.

• A periodicidade da troca deve ser mensal, sem permitir repetição das últimas cinco senhas. Caso não seja possível, a troca deve ser feita, no máximo, trimestralmente.

• A senha deve ser composta de uma combinação de caracteres maiúsculos e minúsculos, sinais e números, fácil de lembrar, porém difícil de ser descoberta. Exemplo: S3nHa!$4t3 (fácil de lembrar, porque traduz Senhaforte).

• Não ser baseada em informações pessoais, como o próprio nome, nome de familiares, bichos de estimação, nome de time de futebol, placa do automóvel, nome da empresa ou departamento, nome do chefe ou superior com ou sem adjetivos etc.

- Não ser constituída de combinações óbvias de teclado, tais como “abc123”, “q1w2e3r4”, “12345asdfg” etc.

• O melhor método é criar uma frase e usar as primeiras letras de suas palavras, aliando ao processo de substituição de letras.

Veja um exemplo com a frase “Este é um curso acerca de segurança em redes e criptografia”. (OK, este português é de “Portugal” – “acerca” ninguém usa aqui!). A senha poderia ser:

“3e1(@d$Rc”. Consegue identificar as trocas? Note que a seqüência – a senha – nunca precisará ser memorizada. Memoriza-se a frase. Por associação, o cérebro lembrará de cada caractere enquanto o usuário pensa na frase e digita a senha.

•Não devem existir senhas genéricas. A senha é pessoal e não deve ser compartilhada.

•Nunca escrever a senha e guardá-la, seja na gaveta, abaixo do teclado, em agenda eletrônica, Palm etc.

•Nunca enviar a senha pela Internet sem um mecanismo de criptografia, ou seja, no modo original em que foi digitada no momento de sua solicitação.

•Evitar a digitação da senha perto de pessoas, mesmo conhecidas, e usar, sempre que possível, computadores confiáveis. Computadores de outras pessoas, compartilhados, como em cyber cafés, por exemplo, podem estar com vírus ou programas instalados que capturam a digitação do teclado (*Key Logger)*. Já existem programas que também capturam cliques do mouse. Assim, o “teclado virtual” oferecido por alguns bancos também é ineficiente para preservar a senha.

•Nunca permita ao navegador, no caso da Internet, “lembrar” ou guardar as senhas. Essa opção disponível no browser[8] permite que um estranho utilize sua máquina sem prévia autorização. Além disso, por ficar armazenada em disco, um atacante que tenha invadido uma máquina pessoal poderá descobrir todas as senhas de outros sistemas armazenadas nesta.

•Nunca use a mesma senha em vários contextos de autenticação, como senhas

8 Navegador WEB (Internet Explorer, Netscape, Mozila, FireFox, Ópera etc).

idênticas para autenticar no domínio do trabalho, para acesso ao banco 24 horas, para acesso a uma rede VPN etc.

Outro problema no esquema de autenticação por senha é a necessidade de armazená-la em algum lugar, por exemplo em um servidor, para compará-la com a fornecida pelo usuário. Dessa forma, o usuário precisa confiar na segurança de acesso ao servidor e o arquivo que armazena as senhas precisa de uma política de acesso seguro, sem permissão para todos, e estar protegido contra gravação.

Uma maneira de não guardar as senhas em sua forma original, no servidor, é utilizar uma função de hash, que as embaralhará. Como se trata de uma função conhecida, produz um resultado padronizado. Assim, um invasor pode utilizar um ataque de força bruta para ir testando várias combinações de senhas até conseguir o mesmo resultado armazenado em um arquivo de senha. Mas não existe um algoritmo reverso simples que permita obter a senha diretamente a partir do hash (lembra?).

Como a senha é transmitida pela Internet do usuário até o servidor, dependendo do meio de comunicação, ela pode ser capturada no meio da transmissão. Desta forma, não importa a complexidade da escolha dessa senha, o processo ficou comprometido, e pior, foi por meio de um ataque passivo, em que nenhuma das entidades percebeu que a comunicação tornou-se vulnerável.

E, finalmente, sendo a senha fornecida pelo usuário, ele pode errar no momento da digitação, causando uma falha na autenticação. Este processo repetido várias vezes possibilita qualificar uma tentativa de acesso por uma pessoa não autorizada, ocasionando o bloqueio de um usuário autêntico, seja pelo fato do próprio usuário ou outra pessoa ter errado a digitação algumas vezes. Embora traga o inconveniente do bloqueio, o procedimento é necessário para minimizar ou eliminar o sucesso dos casos de ataque por força bruta (tentativa de senhas constantes em dicionários).

Apesar de todos os problemas a respeito da autenticação por senha, o *sign-on* ainda é a mais familiar e utilizada e constitui uma das maiores preocupações dos especialistas em segurança.

Na nossa “lista de dicas”, a última foi para evitar ao máximo o uso da mesma senha em vários contextos de autenticação, porém, cada vez mais se utilizam vários servidores e aplicações, o que leva o usuário à necessidade de muitas contas e senhas para cada contexto.

A Infra-estrutura de Chave Pública (ICP) é capaz de ajudar nesse problema, mas o

assunto será tratado em outra disciplina. Num ambiente local, o usuário faz a autenticação local para utilizar a própria estação de trabalho e, depois, acessa alguma aplicação remota, em qualquer servidor da empresa, para utilizar algum software específico.

A Figura 5 ilustra um processo de autenticação em uma aplicação remota.

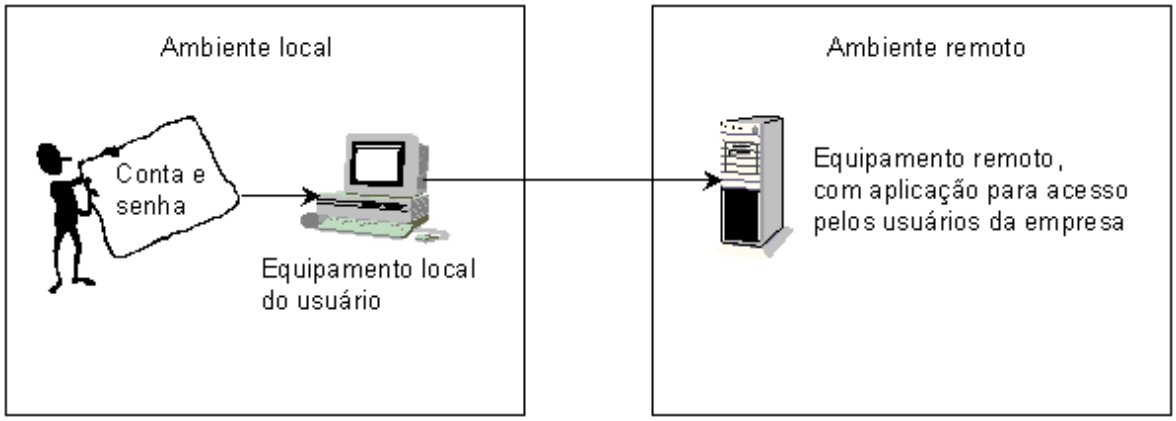


Figura 5 - Autenticação em uma aplicação remota

Mesmo que um mecanismo de autenticação forte seja utilizado, como, por exemplo, smartcard ou token, para reforçar a segurança por meio de algum atributo que o usuário saiba, como conta e senha, e algo que ele tenha, como cartão, esse processo nunca deve permitir que informações de autenticação, como a senha, sejam enviadas para o servidor remoto – ou seja, para "fora" do dispositivo local que o usuário está interagindo – da mesma forma que ele digitou. É necessário um meio para proteger tais informações no intuito de ninguém entender o que foi digitado.

Texto 21 – Single Sign-On (SSO)

O problema de autenticação por senha aumenta consideravelmente quando o usuário precisa acessar várias aplicações que solicitam senha. Usando a mesma para todas as aplicações, ele reduz a segurança, porque, se for descoberta, todas

as aplicações ou servidores precisarão ser bloqueados. Isto pode ser muito complicado, principalmente no caso da administração destes servidores não ser centralizada.

No entanto, senhas diferentes para cada contexto dificultam o acesso e tornam o sistema menos “amigável”. É muito provável que o usuário peça ajuda ao *browser* para “lembrá-las”, ou seja, irá encontrar uma maneira, bem insegura, de se livrar ou de minimizar o problema de múltiplas solicitações.

A infra-estrutura de segurança é capaz de prover a comunicação entre entidades e garantir que a informação seja entregue de forma confiável para os integrantes. Essa segurança pode ser estendida para que um evento de autenticação bem-sucedido seja sinalizado para vários dispositivos remotos, eliminando a necessidade de múltiplas autenticações.

A esse processo se dá o nome de *Single Sign-One* ou SSO, conforme está ilustrado na Figura 6.

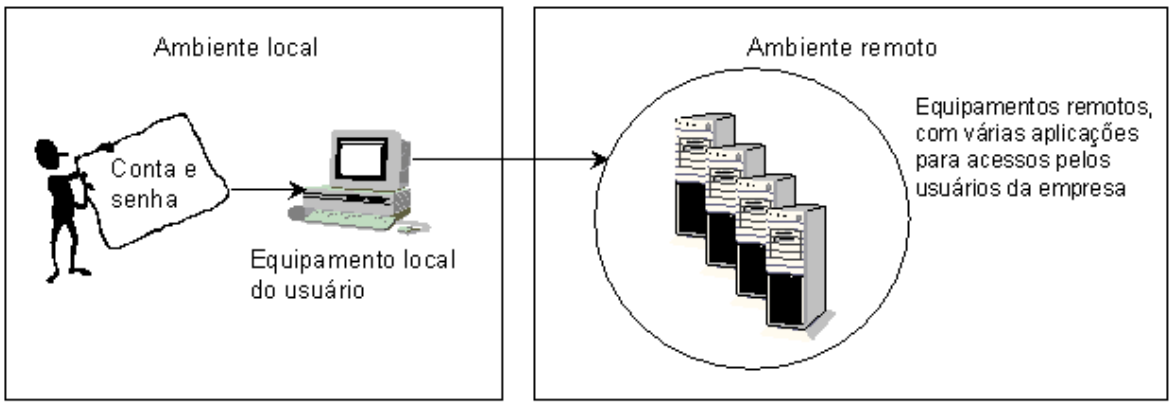


Figura 6 - Autenticação SSO entre servidores.

Como qualquer sistema ou aplicação, uma vez que o usuário fornece a identificação dentro da infra-estrutura, a informação de autenticação pode ser comparada e replicada entre os diferentes sistemas, ou seja, a autenticação fica ativa para todos os dispositivos ou servidores conectados à infra-estrutura de segurança, não sendo necessário fornecer uma conta e uma senha para cada aplicação ou servidor remoto.

Embora facilite a gerência e a administração centralizada de identidades, há uma desvantagem a considerar: a identidade do usuário não é replicável a ponto de permitir, numa mesma conexão remota, o acesso a diferentes contas em vários equipamentos.

Por outro lado, a autenticação pode ser combinada com outro tipo de controle de acesso, autorização, feito logo em seguida, ou seja, autenticação múltipla. Essa combinação é bastante interessante do ponto de vista da facilidade e da segurança, pois permite que:

- o usuário forneça uma única informação e consiga usar vários sistemas ou aplicações sem haver necessidade de telas intermediárias;
- a senha trafegue com menos regularidade do que no modelo tradicional.

O SSO constitui um serviço que pode ser utilizado por todos os dispositivos dentro da infra-estrutura, quando e onde for necessário.

Texto 22 – Autenticação forte

Iremos analisar os principais métodos de autenticação que combinam mais de uma característica, como algo que sabemos, tipicamente login e senha, com algo físico que temos, tokens e smartcard, ou algo que somos, sistemas biométricos.

a) Tokens

Autenticações baseadas em senha envolvem um sério problema: o fator humano, pois é o usuário, ou responsável pela comunicação, que irá estabelecer as senhas, perguntas etc. Como vimos, qualquer senha possui dois princípios: fáceis de descobrir ou complexas o bastante para nos esquecermos delas.

São consideradas fáceis porque o usuário a escolheu baseado em alguma informação pessoal, como data de aniversário com o nome do filho, ou porque usou uma combinação de caracteres que pode ser descoberta por ataque de força bruta.

As senhas complexas tendem a levar o usuário a algumas alternativas para se lembrar delas. Normalmente ele a escreve em uma agenda, coloca-a em cima de sua mesa, embaixo do teclado etc. A dica de formação de senha baseadas em

frases ajuda a minimizar este problema.

Em todos os casos, qualquer esforço é em vão, a menos que tenhamos um mecanismo de geração e validação de senha por hardware, por meio de um dispositivo que forneça uma senha dinâmica para o usuário e o acompanhe onde ele for. Este é o princípio dos tokens, um dispositivo do tamanho de um chaveiro que gera uma seqüência de números, a cada fração de tempo, baseado numa semente (***seed***).

O token de mercado mais conhecido para os serviços de Internet Banking, mais de 90% do mercado, é a Vasco. A solução de segurança para Internet Banking envolve tecnologia mediante o uso de tokens com senhas que mudam a cada minuto, fazendo com que somente os portadores, com suas senhas particulares, tenham acesso às informações de suas contas. Assim, de forma simples, prática e intuitiva, evitam o roubo de identidades.

Trata-se de uma solução já utilizada por mais de 200 bancos em todo o mundo, de fácil implementação e rápida integração com outras aplicações. Oferece também soluções envolvendo certificados digitais ICP-Brasil com validade jurídica, presentes em tokens que se conectam à porta USB das estações.

Tokens USB são eficientes para armazenagem de certificados digitais simples ou ICP-Brasil. Uso de certificados digitais para assinatura digital de transações web permite a verificação de algumas possíveis vulnerabilidades do site, com recomendações de correção.

Outro token também muito conhecido é o SecurID (Security Dynamics), da RSA Security. Cada token tem um número de série que o diferencia dos outros e gera uma seqüência de oito números a cada 60 segundos. Somente o servidor a ele associado pode validar o número gerado, por meio uma senha fixa estabelecida para seu usuário (PIN, número de identificação pessoal) que pode ser uma combinação de números e caracteres ou só caracteres, e não necessariamente apenas números.

Esse esquema de autenticação está dentro do conjunto que chamamos de autenticação forte, porque agrega vários conceitos de segurança extremamente eficientes. Requer algo que sabemos, no caso uma senha do usuário, que pode conter os princípios que citamos**,** tornando-a difícil de descobrir, e algo físico que temos, no caso o token.

Mesmo que o atacante descubra o PIN, o que é uma tarefa complicadíssima, uma vez que a senha irá partir já criptografada do cliente (a seqüência de números

gerada pelo token), o invasor jamais descobrirá a seqüência de números gerada naquele minuto, a menos que esteja ao lado do usuário no momento da invasão.

Embora eficiente, a utilização de tokens peca pelo custo, uma vez que é um dispositivo individual. Logo, quanto mais usuários, maior o custo da solução como um todo.

b) Smartcards

Trata-se de um dispositivo, no formato de um cartão de crédito, com um chip de computador, que tem funções de armazenamento e processamento interno. Os smartcards são utilizados da mesma forma que usamos um cartão instantâneo para retirar dinheiro do banco, visto que usamos o PIN como senha.

O cartão é introduzido num dispositivo que lê a senha que está dentro dele, bem como outras informações, já que possui uma unidade de armazenamento própria.

Os smartcards possibilitam agregar funcionalidades. Devido ao poder de armazenamento, são capazes de conter a chave de criptografia, o que aumenta a velocidade da transmissão, diferentemente do token, que tem somente a função de autenticação. Mesmo com essas funcionalidades, os algoritmos são gravados sem permissão de escrita e também é necessário uma senha fornecida pelo usuário.

Um smartcard tem a capacidade de armazenar e executar programas internamente, gerar chaves em seu circuito e de protegê-las com criptografia. A chave privativa é gerada dentro dele e nunca sai daí, sendo todas as operações de assinatura e autenticação realizadas dentro do cartão. Sua estrutura de segurança o torna um dispositivo muito difícil de ser clonado, ao contrário dos cartões comuns.

O esquema de autenticação também está dentro do conjunto que chamamos de autenticação forte, porque requer algo que sabemos, no caso uma senha do usuário, que pode conter os princípios que citamos, tornando a senha difícil de ser descoberta, e algo físico que temos, no caso o cartão.

Se alguém roubar nosso cartão do banco 24 horas, não poderá sacar nosso dinheiro porque desconhece a senha de acesso, mas, se souber, não há muito a fazer para evitar o furto em nossa conta corrente. Em informática não é diferente. Se uma pessoa souber nosso PIN e tiver posse do token ou do smartcard, é bom recorrer rapidamente ao administrador de rede e solicitar o bloqueio imediato da

conta.

Embora eficiente, esse esquema peca ainda mais em relação ao custo. Como o token, o cartão é um dispositivo único para cada usuário e, quanto mais usuários conectados, maior o custo da solução como um todo. Porém, para que o cartão possa ser lido, requer um dispositivo de leitura, que também deve acompanhar o usuário onde quer que ele esteja. Existem combinações de modems com leitora de smartcard e placas PCMCIA para serem conectadas aos laptops, mas, mesmo assim, seu uso não é muito popular, principalmente pelo elevado custo do dispositivo de leitura de cartão.

Cada solução para usuários da ICP deve ser estudada com cuidado. O ideal é escolher a opção de mercado apropriada a cada necessidade.

Um esquema de autenticação forte para todos os usuários constitui uma decisão dispendiosa. Entretanto, a restrição de serviços ou de informações extremamente importantes pode tornar a decisão mais econômica e irá prover o mesmo nível de segurança necessário à empresa. Voltamos novamente à questão do bom senso quando falamos de segurança. O equilíbrio é a chave do sucesso.

c) Biometria

O emprego de sistemas biométricos, ou seja, baseados em algo que somos, constitui um outro grupo de autenticação, aliás bastante difundido em filmes de ficção . Apesar de sua quase exclusividade, em função do custo da tecnologia, tais sistemas vêm ganhando espaço no mundo virtual.

Utilizar a impressão digital para ligar um computador ou abrir a porta do carro é algo cada vez mais difundido, apesar dos transtornos que pode causar por ser pessoal. Adotar esse processo para um carro talvez não seja uma boa idéia, porque compramos o carro e podemos querer compartilhá-lo com outros membros da família, mas para um computador de uso exclusivo de um alto executivo, que só deve ser utilizado pelo próprio executivo, será mais razoável.

A vantagem sobre os outros esquemas de autenticação é que o usuário é identificado por características únicas, pessoais e intransferíveis, dispensando senhas, cartões, tokens ou crachás. É utilizado tanto para controle de acesso físico como para o de acesso lógico.

Os sistemas baseados em biometria possuem componentes bem definidos, como o sensor e o medidor.

Sensor

Constitui um dispositivo de medida e forma a interface do usuário. Como deve reduzir a possibilidade de erros, a facilidade no uso por grande quantidade de pessoas é um fator importante, especialmente para aquelas não familiarizadas com tal método de autenticação. Normalmente, este hardware serve tanto para capturar as informações de autenticação como para armazenar as individuais de cada usuário.

Medidor

Trata-se de um software de operação, incluindo o algoritmo matemático que checará a medida contra um modelo (template). Os algoritmos mais recentes dependem menos da modelagem estatística e mais da programação dinâmica, das redes neurais e da lógica fuzzy (fuzzy logic), o que aumenta sua flexibilidade. Eles se tornam menos suscetíveis a rejeitar alguém por causa de uma sujeira, por exemplo, se o resto do modelo estiver de acordo.

O sistema de autenticação por biometria é subdividido em dois eventos: registro e verificação. O administrador do sistema registra cada usuário e seus níveis de acesso. O sistema verifica se o indivíduo é um usuário autorizado e quais níveis de acesso tem. Por exemplo, é possível haver pessoas autorizadas a determinados horários, como vigilantes e operadores noturnos. O processo de registro consiste no armazenamento de uma característica biológica do indivíduo (física ou comportamental) para ser usada, posteriormente, na verificação da identidade dele.

A característica biológica é tipicamente adquirida por um dispositivo de hardware, conhecido como sensor, que é um mecanismo de autenticação por biometria. Quando uma característica física é apresentada a ele, produz um sinal, que é modulado em resposta às variações da quantidade física sendo medida. Se, por exemplo, o sensor for um microfone usado para capturar um padrão de voz, irá produzir um sinal cuja amplitude muda com o tempo em resposta à variação da freqüência em uma frase falada.

Pelo fato de, por natureza, os sinais produzidos pela maioria dos sensores serem analógicos, é necessário convertê-los para digitais, a fim de serem processados por um computador. Em vez de usar todos os dados do sensor, os sistemas biométricos freqüentemente realizam o processamento extraindo apenas as informações relevantes ao processo de autenticação. Quando a representação

digital está processada no ponto desejado, ela é armazenada. A característica biológica armazenada na forma digital é chamada de modelo (template). Muitos dispositivos biométricos capturam amostras múltiplas durante o processo de registro para contabilizar graus de variação na medida das características.

No momento do usuário ser autenticado, sua característica física é capturada pelo sensor. A informação analógica do sensor é, então, convertida para a representação digital e, a seguir, é comparada com o modelo biométrico armazenado.

A representação digital usada na verificação é chamada de amostra (*live scan*). Tipicamente, não confere exatamente com o modelo armazenado. Em função da variação na medida, os sistemas não podem exigir uma comparação exata entre o modelo original armazenado (template) e a amostra corrente. Porém, a amostra corrente é considerada válida se estiver dentro de um intervalo estatístico de valores. Emprega-se um algoritmo de comparação para determinar se um usuário, quando verificado, é o mesmo que foi registrado.

O algoritmo de comparação produz um resultado semelhante à representação digital do modelo armazenado. Se o resultado for um valor aceitável, uma resposta afirmativa é dada. A aceitação difere para cada dispositivo biométrico.

O administrador do sistema pode configurar o nível do valor de aceitação. Se o nível configurado for muito baixo, ou seja, qualquer coisa é considerada válida, o dispositivo biométrico falhará como mecanismo de autenticação, por estar sendo muito promíscuo. Já se for muito alto, os usuários talvez tenham problemas na autenticação, pois a medida ter'a que ser muito perfeita (por exemplo, a voz idêntica, a pressão do dedo exata etc). Esse padrão de comparação é fundamental para a operação de qualquer sistema biométrico e, assim, deve ser criteriosamente considerado quando se avalia um produto biométrico específico.

Outro aspecto que afeta a autenticação por biometria é a recuperação do modelo pelo algoritmo de comparação. Pode ser usado na identificação ou na verificação de usuários. Muitos dispositivos usam a verificação, mas alguns, a identificação.

A identificação biométrica pode ser um processo um-para-muitos ou um-para-um. No primeiro processo, uma amostra é submetida ao sistema que a compara com todos os modelos da base de dados, a fim de verificar se esta coincide com qualquer um deles e, em caso positivo, determina a identidade do usuário a quem aquele modelo pertence.

Já no processo um-para-um, o sistema verifica a identidade de um usuário comparando a amostra com um modelo específico. Por meio de uma informação fornecida, o sistema localiza o modelo desejado e compara-o com a amostra apresentada. Se houver coincidência entre eles, o sistema confirmará que o usuário realmente possui a identidade afirmada. Por exemplo, um usuário digita seu nome e, assim, se adquire uma amostra para a verificação. O algoritmo de comparação usará apenas o modelo armazenado àquele nome.

Os sistemas biométricos baseiam-se em características físicas e comportamentais de pessoas. Uma característica física deve ser relativamente estável, tal como impressão digital, estrutura da mão, padrão de retina, padrão de íris ou uma outra característica facial. Assim são basicamente imutáveis ou variam pouco no decorrer do tempo. Em contrapartida, uma característica comportamental reflete o estado psicológico de uma pessoa (isto é, pode ser afetada por problemas como estresse, fadiga, gripe etc.).

Entretanto, existem elementos psicológicos que podem ser usados na identificação de uma característica. Por exemplo, o método de identificação baseado em comportamento mais comum é a assinatura de uma pessoa, adotado pela sociedade há décadas. Outros comportamentos usados incluem o ritmo de digitação e o padrão de voz.

Há sistemas que modificam o modelo de referência original cada vez que este é usado. Isso ocorre porque muitas características comportamentais mudam no decorrer do tempo e, assim, depois de muitos acessos com sucesso, o modelo pode ser diferente (às vezes significativamente) do modelo inicial, produzindo melhor desempenho na identificação de um usuário inválido. Os esquemas que utilizam esse método trabalham melhor apenas quando usados regularmente. Um exemplo é o caso do método de identificação por meio da assinatura, no qual podemos mudar o tamanho de um acento ou o comprimento de linhas de determinado caractere. Outro exemplo é o método de identificação por voz, em que pode haver pequenas variações no som.

Em geral, como o grau de variação entre as pessoas é maior em relação a uma característica comportamental do que em uma física, é mais difícil para desenvolvedores de sistemas baseados em comportamento o ajuste da variação individual. Entretanto, sistemas que medem atributos físicos tendem a ser maiores e mais dispendiosos e, em algumas aplicações, seu uso pode ser considerado ameaçador aos usuários.

Os dispositivos biométricos baseados em comportamento são normalmente

menores em tamanho, sua implementação é menos custosa e o uso é mais amigável. Ambas as técnicas fornecem um meio de autenticação de usuários muito mais confiável que os mecanismos de segurança baseados em senhas ou cartões.

Em virtude dessas diferenças, nenhum sistema biométrico irá servir para todas as necessidades e, para serem efetivos, é necessário escolher as técnicas adequadas a diferentes situações. Por exemplo, um sistema de verificação de voz pode ser usado em um escritório, enquanto um sistema de reconhecimento de retina pode ser usado no controle de acesso a áreas de segurança máxima.

Texto 23 – Usuários do sistema

Como visto, todo programa que roda faz seus acessos utilizando o usuário que o chamou para verificar as permissões de acesso. Durante o processo de boot de um servidor, diversos programas são chamados a rodar, para prestarem os serviços a que se destinam. Durante esta fase, o usuário que opera estas chamadas automáticas é o “root”. Assim, a princípio, todos os programas chamados durante a fase de boot rodam como root e poderiam fazer tudo!

Sem dúvida trata-se de um possível furo de segurança: um programa com defeito, que rode como root e seja manipulado por um atacante, é sinônimo de comprometimento da máquina.

Então, os programas servidores não devem rodar como root. Para tal, existem sistemas que não são de pessoa alguma (não existe um ser humano dono da conta), mas que precisam rodar na máquina e acessar recursos nesta, sem ser root. Para tais casos, foram criados usuários do sistema que não estão relacionados com uma pessoa do mundo real: não há dono ou senha para aquela conta. Na verdade, são contas sem senha real: ninguém consegue logar nelas (exceto o todo poderoso root, que pode fazer-se passar por uma delas).

Como exemplos, citamos os sistemas que prestam serviços, como email, FTP e WWW, cada um prestado por um programa servidor, que roda como sendo de um usuário.

O administrador do sistema (ou o autor da distribuição do GNU/Linux) deve escolher um nome de usuário que tenha baixa possibilidade de ser usado por

outro programa. É recomendável haver um usuário para cada programa aplicativo que faça acessos a dados importantes, principalmente se os acessos forem de gravação (escrita).

Alguns softwares, conforme citamos anteriormente o caso do Oracle, incluem em seu script de instalação a criação de um usuário que será utilizado exclusivamente pelo banco de dados para permitir o acesso aos arquivos do banco em disco. Também são exemplos os servidores de bancos de dados "livres", tais como o MySQL e o Postgresql. Se uma distribuição de GNU/Linux vier com o MySQL instalado, provavelmente será encontrado no sistema um usuário e um grupo destinados ao uso deste banco. O nome geralmente é "mysql" (para o usuário e para o grupo).

Um cuidado nas distribuições preocupadas com segurança é que as contas destes usuários de sistema (como o mysql) estão, desde a instalação, com a senha bloqueada, ou seja, não existe uma senha com a qual possamoss nos logar no sistema com esta conta de usuário, seja a partir do console ou acesso remoto. Veremos, logo a seguir, em permissões de acesso, como isto funciona.

Verifique e liste os usuários de sistema, usando o seguinte comando no Linux:

```
cat /etc/passwd
```

Texto 24 – Permissão de acesso

O sistema de permissão de acesso é a ferramenta com a qual todos os acessos a dispositivos de entrada e saída e a arquivos e diretórios (pastas) são controlados.

O usuário root não respeita as permissões de acesso: possui acesso livre a tudo!

Todos os arquivos possuem um usuário (dono) e um grupo associado. Isto pode ser verificado em qualquer diretório, ao se executar o comando:

```
ls -l
```

Mostra a lista de arquivos do diretório onde você estiver em colunas, conforme exemplo:

```
-rw-r--r--   1 marcos users      7763 Jun 16  2003
tabela.subrede.sxc
```

A listagem traz as seguintes colunas, na ordem: tipo de arquivo/permissão de acesso; Número de links; Usuário; Grupo; Tamanho; Data; Nome.

Então, a terceira coluna indica o usuário dono (marcos, no exemplo) e a quarta coluna indica o grupo (users, no exemplo).

Para o sistema de permissão de acesso, são consideradas três classes:

- **dono**

  Todo arquivo[9] possui um dono. De modo geral, o dono de um arquivo é o usuário que o criou. Somente o dono pode alterar as permissões de um arquivo.

- **grupo**

  Todo arquivo possui um grupo associado. Normalmente, o grupo do arquivo é o grupo primário do usuário que criou o arquivo. A idéia do uso de grupos é dar permissão de acesso a muitas pessoas, todas pertencentes ao grupo. No UNIX, apenas usuários podem pertencer a grupos, ou seja, um grupo não pode ter outro grupo dentro dele. A lista de grupos existentes fica no arquivo /etc/group.

- **outros**

  São os demais usuários, retirando-se o dono e todos os que pertencem ao grupo do arquivo.

Texto 25 – Tipo de permissão

Conforme vimos ao listar arquivos, a primeira coluna mostra uma série de letras e traços. A informação é lida assim: a primeira posição possui a especificação de tipo de arquivo (está com um "-"). As especificações possíveis estão na Tabela 2.

9 Está sendo usada a palavra arquivo, mas lembre-se de que "tudo em UNIX é arquivo". Então isto vale também para diretórios e dispositivos.

Caractere	Tipo de arquivo
-	Arquivo comum
d	Diretório
l	Link simbólico (atalho)
c	Dispositivo orientado a caractere
b	Dispositivo orientado a bloco
p	Pipe

Tabela 2 - Tipos de arquivos

No caso do exemplo, verificando o primeiro caractere em “-rw-r--r--”, percebe-se que se trata de um arquivo comum.

Retirando o primeiro caractere, sobram nove. São os modos de permissão.

Para ler seu significado, deve-se dividi-lo em três grupos de três caracteres (rw- r-- r--).

O primeiro grupo de letras diz respeito às permissões de acesso do dono do arquivo; o segundo grupo diz respeito a quem pertence ao grupo do arquivo; o último grupo diz respeito aos outros usuários.

As letras que podem aparecer aqui são r, w, x, t, s, S, T.

Concentrando-se nas três primeiras letras, e dependendo ainda se o arquivo em questão é um arquivo comum ou um diretório, temos o significado conforme a Tabela 3. Se a letra estiver aparecendo, significa que existe esta permissão. Se no lugar da letra estiver um hífen (-), significa que não existe a permissão (permissão desligada).

Letra	Arquivo (–)	Diretório (d)
r	Permite leitura	Permite listar (ls)
w	Permite escrita	Permite criar, apagar ou renomear itens
x	Permite execução	Permite acessar

Tabela 3 - Permissões de acesso a arquivos e diretórios

Pontos importantes a serem observados:

- poder ser executado ou não, é um atributo do arquivo (uma permissão), e não depende do nome do arquivo;
- permissão de escrita em um diretório permite **apagar** arquivos. Mesmo que um usuário não tenha permissão para ler ou escrever em um arquivo,

se ele tiver permissão para escrever no diretório onde está o arquivo, poderá apagar o arquivo!

- acessar um diretório significa poder ir para o diretório (com cd) – para chamar um arquivo para executar, é necessário ter permissão de acesso no diretório onde este arquivo se encontra.

> Qualquer arquivo com qualquer nome pode, ou não, ser um executável. Depende de uma permissão e não do nome ou extensão do arquivo.

Validação do acesso

Somente um dos conjuntos de permissão é utilizado em uma tentativa de acesso. A validação de um acesso tem um procedimento seqüencial:

a) Se for o usuário declarado como dono do arquivo

Verificam-se somente as permissões de acesso para o dono, e mais nenhuma. Se as permissões do dono não permitirem o acesso, este será negado e a verificação terminada, mesmo que haja uma outra permissão que possa permitir o acesso.

b) Se não for o dono

Verifica-se se pertence ao grupo do arquivo. Se pertencer, então o segundo grupo de permissões irá validar ou não o acesso;

c) Por último, se não for nem o dono, nem pertencer ao grupo do arquivo, então será verificado o terceiro grupo de permissões (outros).

Permissões avançadas

Além das permissões básicas (leitura, escrita e execução), existem outros “bits” de permissões que alteram o comportamento de programas e diretórios. Estes bits são chamados de Set User ID (SUID); Set Group ID (SGID) e stick bit.

O **SUID** troca o usuário efetivo na execução de um programa (arquivo). Assim, se um arquivo tem o SUID ligado, não importa quem chame (coloque para executar) o programa, ele sempre rodará e tentará fazer acessos como se o dono dele o tivesse chamado.

O **SGID** troca o grupo efetivo na execução de um programa (arquivo). Assim, se

um arquivo tem o SGID ligado, não importa quem chame (coloque para executar) o programa, ele sempre rodará e tentará fazer acessos como se quem chamou pertencesse ao grupo do arquivo.

O **stick** tem seu maior uso em diretórios onde, se ligado, impede que outros usuários, que não o dono, possam apagar um arquivo. Lembre-se que quem tem permissão de escrita em um diretório pode apagar qualquer arquivo, mesmo que não seja seu.

Antigamente, quando as máquinas tinham pouca memória física e usavam muita memória swap (qual o problema nisso?), o stick ligado em um executável prevenia que ele fosse retirado da memória principal para o swap. Isto melhorava o desempenho deste programa, principalmente em sistemas interativos.

Como "falta espaço" ao listar arquivos com o "ls -l", a posição onde fica o "x" (execução) teve que compartilhar a informação sobre estes bits especiais. Os bits SUID e SGID lembram a letra "s" e, para não confundir, o stick ficou com o "t". Se esses bits especiais não estiverem ligados, tudo continua como já vimos. Mas, se estiverem ligados, teremos:

"x" e (SUID ou SGID) = s;

"-" e (SUID ou SGID) = S (maiúsculo);

"x" e Stick = t;

"-" e Stick = T (maiúsculo).

Aplicados a arquivos ou diretórios, estes modificadores alteram as permissões de acesso, como mostra a Tabela 4.

Bit	Arquivo	Diretório
SUID	Se for executado, o arquivo roda como o usuário dono do arquivo (e não como quem o chamou)	Nada
SGID	Se for executado, o arquivo roda como o grupo do arquivo (e não como o grupo de quem chamou)	Todo arquivo criado neste diretório será do mesmo grupo do diretório, e não do grupo primário de quem criou o arquivo
Stick	Impede que o sistema operacional coloque o programa (executável) na área de swap, quando estiver rodando	Revoga a permissão de apagar arquivos para usuários que tenham permissão de escrita no diretório. Apenas o dono do arquivo (normalmente quem o criou) poderá apagar o arquivo

Tabela 4 - Bits especiais de permissão de acesso

Permissões em números

Se a existência ou não de cada posição de letra for substituída por um bit, podemos dizer que, até agora, temos nove bits de permissão básica. Com os três bits especiais (SUID, SGID e stick, nessa ordem, e colocados à esquerda dos demais), temos 12 bits. Se a cada três bits substituirmos o valor pelo correspondente em decimal, teremos um número, em notação octal, com quatro casas. Substitui-se a letra por 1, o hífen por 0. Por exemplo:

-rwxr--r-- , fica:

111 100 100

Substituindo o valor binário pelo seu correspondente decimal:

111 = 7

100 = 2

Então, dizer que um arquivo possui permissão de leitura, escrita e execução para o dono, de leitura para o grupo e de leitura para os demais, e nenhum bit especial ligado, é idêntico a dizer que o arquivo possui permissão 0722 (o zero à esquerda normalmente é omitido).

Texto 26 – Trocando dono, grupo e permissões

chown (de ***CH****ange* ***OWN****er*)

Comando usado para trocar o dono. Pode-se trocar grupo ao mesmo tempo, separando o usuário do grupo por um ponto (.) ou por dois pontos (:).

Para usar, o comando é seguido pelo nome do usuário e, depois, pelo nome do arquivo alvo. Exemplo:

chown joao texto.txt (o arquivo texto.txt passa a ter joao como dono).

Na prática, apenas o root pode trocar donos de arquivos, pois os usuários comuns não têm este poder.

chgrp (de ***CHa****nge* ***GR****ou****P*****)**

Comando usado para trocar o grupo.

Seu emprego é similar ao chown, sendo o nome do grupo colocado na segunda posição. Exemplo:

chgrp users /tmp/carta.doc (coloca o arquivo carta.doc, que está no diretório / tmp, como sendo do grupo users.

chmod (de ***CH****ange* ***MOD****e*)

Comando usado para trocar permissões tanto no modo texto (com caracteres alfabéticos), como no numérico. O modo caracter normalmente é utilizado para acrescentar (+) ou retirar (-) uma ou outra permissão e o modo numérico, para colocar todas as permissões da forma final que se quer.

O modo texto utiliza “u” para usuário dono; “g” para grupo; e “o” para outros. Também se emprega “a” para todos (*All*).

Exemplos:

```
chmod u+x arquitetura (

acrescenta permissão de executar para o dono do arquivo
“arquitetura”).

chmod og-r listagem (retira permissão de leitura para grupo e
para outros do arquivo “listagem”).

chmod og+rx listar (acrescenta permissão de leitura e
execução para grupo e outros, ao arquivo “listar”).
```

Os comandos chmod numéricos usam exatamente a permissão numérica para ajustar de uma só vez as permissões. Veja nos exemplos a seguir.

Texto 27 – Prática dos comandos de permissão

Onde for possível, execute os comandos, acompanhando os exemplos.

a) Novamente usando o exemplo:

```
-rw-r--r--   1 marcos users      7763 Jun 16  2003
tabela.subrede.sxc
```

Permissão numérica: 422

Comando chmod para deixar a permissão assim:

```
chmod 422 tabela.subrede.sxc
```

Se este arquivo não existir em seu sistema, irá retornar um erro.

Temos permissão de leitura e escrita para o usuário “marcos”, e somente leitura para quem pertencer ao grupo “users” e para os demais usuários.

b) Usando:

```
ls –l /bin
```

Veremos, dentre outros, o seguinte arquivo:

```
-rwxr-xr-x  1 root bin   72608 Mar 15  2004 ls
```

Permissão numérica: 755

chmod 755 /bin/ls

Este arquivo possui permissão de leitura, escrita e execução para o usuário “root”, e leitura e execução para os demais usuários. Perceba que resumimos em “demais usuários” os usuários pertencentes ao grupo “bin” e os outros usuários (pois têm permissões iguais).

c) ls -l /etc/shadow

```
-rw-r-----  1 root shadow 888 Feb 24 21:05 /etc/shadow
```

Permissão numérica: 640

Possui permissão de leitura e escrita para o “root” e leitura apenas para quem pertence ao grupo “shadow”. Ninguém mais acessa este arquivo.

Trata-se do o arquivo onde são guardadas as senhas dos usuários. Assim, somente programas rodando como root podem alterar senhas. E programas que fazem verificação de senha podem rodar como algum usuário que pertença ao grupo shadow.

d) ls -l /etc/shadow-

```
-rw-------    1 root root       855 Feb 24 21:04 shadow-
```

Permissão numérica: 600

O arquivo /etc/shadow- (com um hífen no final) é um backup do arquivo shadow. Este backup é feito pelo sistema e o “root” é o único que pode ler e escrever nele.

e) -rw-r-x--x 1 marcos users   7763 Jun 16 2003 teste.bin

Permissão numérica: 651

Este exemplo possui um conjunto de permissões diferente para cada grupo de permissões.

Se o usuário marcos tentar acessar, terá permissão para **ler e escrever** no arquivo (executar, não);

Se um usuário pertencente ao grupo users tentar acessar, terá permissão para **ler e executar** o arquivo (mas não escrever);

Se qualquer outro usuário tentar acessar, terá permissão **somente para executar** o arquivo (nem ler, nem escrever).

O exemplo mostra uma importante proteção nos níveis de acesso do UNIX: para arquivos binários que não sejam scripts interpretados, não é necessário permissão de leitura, para que ele possa ser executado. Isto protege contra cópias indevidas, pois para copiar é necessário permissão de leitura. Então, para executar um binário, o usuário chamador não precisa ter permissão de leitura, somente de execução. Imagine que quem lê o arquivo para carregá-lo na memória e executar é o sistema operacional, e não o usuário. Por isso, ele não precisa da permissão de leitura.

Já para scripts interpretados, o usuário chamador tem que ler o arquivo para entregar ao interpretador. Por isso, requer permissão de leitura. A permissão de execução é necessária para permitir chamar o script pelo nome do arquivo, diretamente.

f) drwxr-xr-x 6 root root 408 Oct 13 01:17 ssl/

Permissão numérica: 755 (A letra d não interfere na conversão numérica).

Este diretório pode ser **lido, escrito e acessado** somente pelo "root", e pode ser **lido e acessado** pelos demais usuários. Novamente concatenamos as informações das permissões de acesso de grupo e de outros.

g) ls -l /usr/bin/passwd

```
-rws--x--x  1 root bin 37880 Jun 21  2004 /usr/bin/passwd
```

Permissão numérica: 4711. O bit SUID ligado é o representado pelo 4 (100 em binário).

Permissão de execução para todos os usuários (dono + pertencentes a um grupo + outros = todos); mas somente o root pode ler e escrever.

Note que:

- não importa quem chame o programa, ele passará a rodar como se o dono o tivesse chamado, no caso, o root;
- o programa tem acesso a tudo e não existem permissões negativas para ele (root).

Este programa binário é o responsável pela troca de senhas no sistema. Assim, tem que ler e escrever no arquivo onde ficam guardadas as senhas, o /etc/shadow. Relembre, em exemplo anterior, quais as permissões de acesso.

Então, o único que pode alterar este arquivo é o root. Por isso o passwd roda com SUID root! Embora isso pareça muito perigoso, como o comando passwd é muito confiável e comportado, ele nunca faz o que não deve, e o restante do sistema está a salvo.

h) ls -l /

Procure a linha similar a esta (os números e datas não estarão iguais):

```
drwxrwxrwt   62 root   root   17232 Mar 20 16:24 tmp/
```

Permissão numérica: 1777

Perceba a promiscuidade das permissões de acesso neste diretório: todos podem fazer tudo! Qualquer um pode criar qualquer coisa dentro do diretório. Mas, como o Stick está ligado (“t” na última posição), indica que somente o dono de um arquivo (ou diretório) pode apagá-lo.

Isto é a alternativa no UNIX para criar um diretório público e razoavelmente seguro, normalmente usado para arquivos temporários.

i) Acompanhe está seqüência de comandos, estando logado como root:

```
id marcos

uid=1000(marcos) gid=100(users) groups=100(users),5(tty),6
(disk),14(uucp),145(video),17(audio),19(cdrom)

cat > po

teste (pressiona-se ENTER seguido de Control D)

chgrp cdrom po

chmod 660 po

ls -l po

-rw-rw----  1 root cdrom 6 Mar 20 21:56 po
```

Em outro terminal (console) e logado como o usuário marcos, digite:

```
cat /tmp/po

teste
```

Você notou que o fato do usuário pertencer ao grupo cdrom permitiu o acesso de leitura.

De volta ao terminal logado como root:

```
chown marcos po
```

```
chmod 060 po

ls -l po

----rw----  1 marcos cdrom 6 Mar 20 21:56 po
```

De volta ao terminal logado como usuário comum:

```
cat /tmp/po

cat: /tmp/po: Permission denied
```

Então, o fato do dono coincidir com o usuário que está tentando o acesso, e este não ter permissão, bloqueia o acesso e nada mais é verificado, mesmo o grupo tendo permissão para acesso.

Texto 28 – Aumentando a segurança

Com o uso correto das permissões de acesso, é possível elevar o nível de segurança do sistema e impedir acessos indevidos. Infelizmente, em muitos locais onde a segurança não é um fator crítico, é comum observarmos erros de configuração de permissões de acesso, seja por displicência ou por falta de domínio da ferramenta.

Neste item veremos um caso fictício, mas que pode ser adaptado para muitos sistemas reais.

Vamos imaginar que temos um programa de folha de pagamento, que acessa seu arquivo de dados. O nome do programa é FDP (Engenheiros de Telecomunicações às vezes confundem com Função Densidade Probabilidade!), e seu arquivo de dados se chama “folha.dat”.

Quais seriam as permissões mínimas para que este sistema funcionasse?

O FDP precisa ser executável e o arquivo de dados deve ter permissão de leitura e escrita.

Após colocar os dois arquivos no diretório apropriado, o administrador do sistema

pode executar os seguintes comandos:

```
chmod 755 FDP

chmod 666 folha.dat

ls -l

-rwxr-xr-x   1 root root     41208 Oct 13 10:17 FDP

-rw-rw-rw-   1 root root      4708 Oct 13 10:17 folha.dat
```

Assim, qualquer um que digitar FDP colocará o sistema de folha para funcionar. O programa herdará o usuário que o chamar, mas, como todos têm acesso de leitura e escrita no arquivo folha.dat, o programa FDP poderá acessá-lo para leitura e escrita.

Percebe-se, no entanto, uma falha de segurança: se qualquer usuário pode ler e escrever no arquivo de dados da Folha de Pagamento, ele pode ser facilmente adulterado ou corrompido.

Pensando nisto, o administrador cria o grupo “folha” e nele coloca somente os funcionários que têm permissão para acessar o sistema de Folha de Pagamento. Veja como fazer:

```
groupadd folha

usermod -G users,floppy,folha usuário1

usermod -G users,floppy,folha usuário2

usermod -G users,floppy,folha usuário3

chgrp folha FDP folha.dat

chmod 750 FDP

chmod 660 folha.dat

ls -l
```

```
-rwxr-x---    1 root folha     41208 Oct 13 10:17 FDP

-rw-rw----    1 root folha      4708 Oct 13 10:17 folha.dat
```

Agora, os usuários 1 2 e 3 passam a pertencer ao grupo folha (além dos grupos users e floppy). Assim, somente eles poderão colocar o programa FDP para rodar e acessar ao arquivo de dados.

Mesmo tendo melhorado a segurança, ainda é considerado falta de segurança o fato destes três usuários conseguirem acessar diretamente o arquivo de dados "folha.dat".

Com as permissões básicas não é possível fazer mais nada. No entanto, com o uso de SUID, impedimos que o arquivo de dados seja acessado diretamente, mas permitimos que seja acessado pelo programa FDP. Para tal, criamos um usuário de sistema para ser usado somente pelo Folha de Pagamento – vamos chamá-lo de "folha". Fazemos com que o arquivo de dados somente possa ser acessado por "folha". O movimento final (o pulo do gato) é colocar o executável como sendo de "folha" e com o SUID ligado. Observe a seqüência de comandos:

```
useradd folha -g folha

chown folha FDP folha.dat

chmod u+s,g-r FDP

chmod g-rw folha.dat

ls -l

-rws--x---    1 folha folha     41208 Oct 13 10:17 FDP

-rw-------    1 folha folha      4708 Oct 13 10:17 folha.dat
```

Assim:

- foi adicionado o usuário folha, que tem como grupo primário o grupo folha;
- o dono dos arquivos foi trocado para folha;

- foi ligado o SUID (u+s) e o administrador aproveitou para aumentar a segurança e impedir que o arquivo do executável (FDP) pudesse ser copiado, tirando a permissão de leitura para o grupo (g-r);
- o arquivo folha.dat teve retiradas as permissões de acesso para grupo.

Agora, apenas os usuários 1, 2 e 3 cadastrados anteriormente no grupo folha podem executar o arquivo FDP. Mas, ao ser executado, o usuário efetivo (usado para testes de permissão de acesso) será o usuário “folha”, devido ao SUID (“s” na posição de usuário). O usuário folha tem permissão para acessar (leitura e escrita) ao arquivo folha.dat. Assim, o sistema continua funcionando e com total segurança, no que diz respeito ao controle de permissão de acesso do UNIX.

Logicamente, a segurança do sistema de Folha de Pagamento ainda depende, em parte, da segurança interna do programa, que também deve ter níveis de acesso para permitir somente a usuários habilitados e credenciados, níveis distintos de acessos aos diferentes tipos de dados. Mas isto foge ao escopo e objetivo deste curso.

Exercícios

Usando os exemplos desta unidade, reescreva todos os comandos de chmod, invertendo o método utilizado (texto X numérico), de modo que o resultado seja o mesmo, ao fim do comando.

Unidade 3 – Serviços de rede

Veremos agora pontos intrinsecamente falhos nos sistemas de serviços de rede; de que modo um UNIX disponibiliza serviços de rede; aspectos que tornam falha a prestação de serviços de rede; como determinar se um serviço está ativo ou não; e alterar a configuração padrão para ligar ou não um serviço ao ligar a máquina.

Texto 29 – Serviços & portas

Serviços de rede é o que está disponível para ser acessado pelo usuário. No TCP/IP, cada serviço é associado a um número chamado *porta. A porta é o endereço de transporte (TCP ou UDP)* onde o servidor espera pelas conexões de transporte dos computadores clientes. Uma porta de rede é referenciada tanto pelo número como pelo nome do serviço. Um servidor, com um único endereço IP pode prestar diversos serviços de rede, através da utilização de uma porta para cada serviço.

A princípio, qualquer serviço funciona em qualquer porta. No entanto, diversas portas se tornaram bem conhecidas e, assim, padronizadas, por se tornarem uma unanimidade para determinado serviço. Alguns exemplos de portas padrões usadas em serviços TCP/IP são:

- 21 – FTP (transferência de arquivos);
- 23 – Telnet (terminal virtual remoto;
- 25 – Smtp (envio de emails);
- 53 – DNS (resolvedor de nomes);
- 79 – Finger (detalhes sobre usuários do sistema);
- 80 – http (protocolo www - transferência de páginas Internet);
- 110 – POP-3 (recebimento de mensagens);
- 119 – NNTP (usado por programas de noticias);
- 143 – IMAP (recebimento de mensagens mais eficiente que POP);
- 443 – HTTPS (HTTP seguro, com SSL);
- 993 – IMAPS (IMAP seguro, com SSL);
- 995 – SPOP3 (POP3 seguro, com SSL).

No arquivo /etc/services do UNIX, obtém-se a relação completa das portas bem conhecidas. O arquivo consta só de uma tabela que relaciona um número de porta com o nome do serviço. Não contém alguma indicação se um serviço está ou não disponível ou em uso. Trata-se de um arquivo que não deve ser editado, a não ser para incluir novos serviços (portas).

Para verificar o que está disponível e em uso em termos de serviços TCP/IP na sua máquina, use o comando:

```
netstat –an ––ip
```

Texto 30 – Protocolo TCP

Aqui é importante relembrar como funciona o TCP, assunto que você estudou na disciplina de Computadores e Redes de Computadores.

A maioria dos sistemas que necessitam de transporte de dados utiliza o TCP como protocolo de transporte. Por ter sido desenvolvido em uma época mais "inocente", os desenvolvedores do TCP [INT1][INT2] não se preocuparam demasiadamente com a segurança da informação. A preocupação era que o transporte de dados fosse confiável fim a fim, que os dados não desaparecessem no caminho.

O princípio do estabelecimento da conexão e como o TCP garante a integridade são os pontos chave que permitem o seqüestro de conexões, ou mesmo forjar ser um cliente válido, para obter informações sigilosas. O seqüestro só tem validade para o atacante se os dados não estiverem protegidos por alguma instância de criptografia.

Toda a confiabilidade do TCP é baseada no número de seqüência do pacote

(*Packet Sequence Number* – PSN). Se estes números estiverem na seqüência correta nos dois sentidos, a comunicação continuará, sem problemas.

O início do estabelecimento da conexão é realizado conforme mostra a Figura 7. O primeiro número de seqüência deve ser gerado aleatoriamente, para dificultar exatamente que se possa acompanhar ou seqüestrar uma conexão.

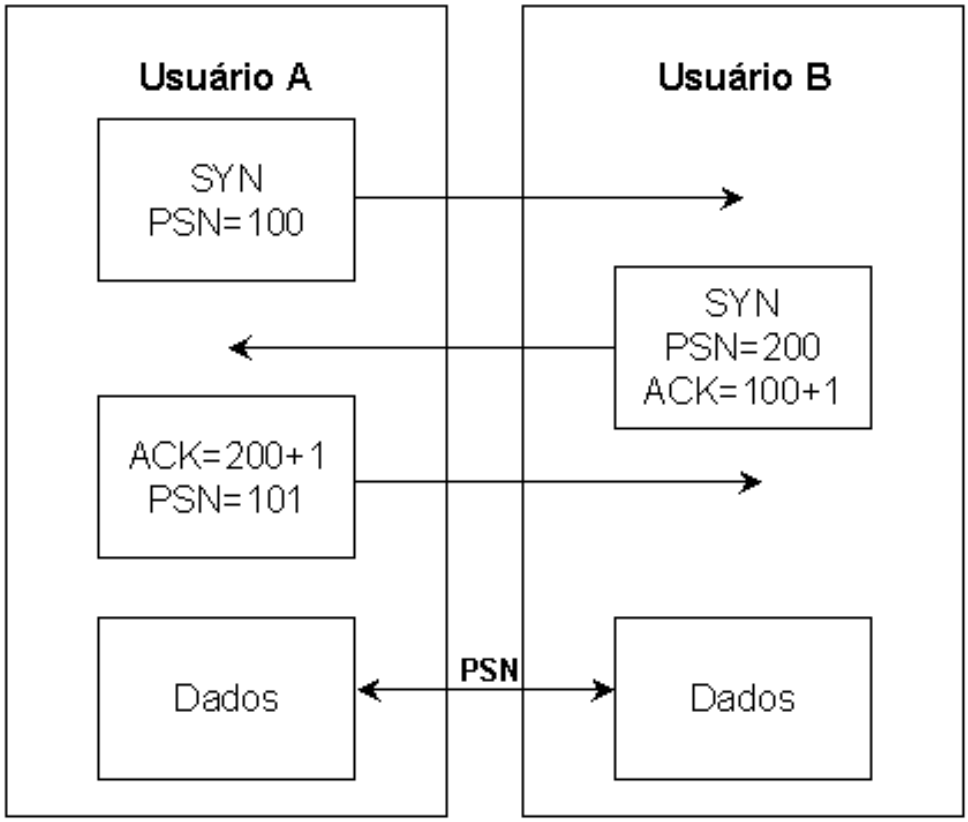


Figura 7 - Estabelecimento de conexão entre usuários com TCP.

Como toda a “segurança” do TCP se baseia nestes números, qualquer sistema que consiga imitar a seqüência nos dois sentidos, é capaz de se passar por outro. Copiar e replicar o seqüenciamento pode ser razoavelmente fácil. O problema para o sucesso do ataque está na duplicidade de pacotes, ou seja, cada pacote com um número de seqüência apareceria vindo da estação verdadeira e vindo do atacante.

Adiante, abordaremos este assunto. No momento, basta saber que a confiabilidade do TCP é apenas quanto à garantia de entrega, e não quanto ao sigilo ou segurança das informações.

Texto 31 – Conectividade excessiva

Tanto os usuários legítimos como os atacantes conectam-se aos sistemas por meio de portas abertas. Quanto maior o número de portas abertas, maiores serão as possibilidades de alguém se conectar ao sistema. Conseqüentemente, é importante manter o menor número de portas abertas necessárias para o correto funcionamento do sistema. Todas as outras devem ser fechadas.

O comando “netstat” pode ser executado localmente, a fim de identificar as portas que estão abertas. No entanto, a forma mais confiável de fazer isso é utilizando uma ferramenta de scan de portas no próprio sistema. O resultado é uma lista de todas as portas que estão realmente ativas. Caso os resultados obtidos com o comando “netstat” sejam diferentes dos obtidos com a ferramenta de scan, é preciso que se investigue o porquê disso.

Uma vez que ambas as listas coincidam, procure saber por que cada uma das portas indicadas está aberta e o que está sendo executado em cada uma. Feche todas as portas que não puderem ser justificadas. A lista final de portas deve ser gravada em um local seguro e utilizada para fazer auditorias regularmente, garantindo assim que não apareça nenhuma porta adicional sem uma justificativa plausível.

Dentre as muitas ferramentas de scan de portas, a mais popular é o Nmap. Existem outras ferramentas de scan de portas que também funcionam bem. Qualquer que seja a ferramenta escolhida, é aconselhável que seja configurada para varrer todas as portas UDP e TCP no intervalo de 1-65535.

É importante ter uma autorização por escrito antes de executar algum tipo de varredura de portas nos sistemas dentro de uma empresa, porque alguns sistemas operacionais, em particular dispositivos que implementem TCP/IP nativo, quando examinados por essas ferramentas, podem apresentar um comportamento imprevisível. Isso pode também acionar o firewall ou sistemas de detecção de intrusos na rede interna, dando margem à interpretação de se tratar de um ataque. Recomenda-se que esse scan seja sinalizado de maneira apropriada, como, por exemplo, identificando o usuário pelo nome.

Uma vez identificados as portas abertas e o conjunto mínimo que deve permanecer aberto para o bom funcionamento do sistema, fecham-se todas as restantes, como já dissemos . Para fechar uma porta, desabilite ou remova o serviço a ela associado.

Em sistemas UNIX, muitos dos serviços são controlados pelo super daemon inetd e seu correspondente arquivo de configuração, inetd.conf. O arquivo inetd.conf lista os serviços associados a uma determinada porta e, freqüentemente, é usado para fechar as portas.

Ao remover um serviço do arquivo inetd.conf e reinicializado o *daemon*, a porta associada a tal serviço é automaticamente fechada. Outros serviços são inicializados por meio de scripts, os quais são executados durante o processo de inicialização do sistema, tais como: /etc/rc, /etc/rc.local, ou scripts encontrados nos diretórios /etc/rc. Isto será visto em mais detalhes, em outro tópico de nossa disciplina.

Para saber como desabilitar os scripts, consulte a documentação do sistema uma vez que o procedimento varia para cada versão de UNIX.

Além disso, existe um programa chamado lsof que pode ser usado para auditar portas abertas. O lsof mostra todos os recursos em uso na máquina, sejam eles arquivos, dispositivos, rede etc.

Em sistemas Windows NT e 2000, para determinar que serviço/programa está escutando em uma determinada porta, há a ferramenta fport; no Windows XP, há o comando netstat com a opção "-o". O retorno desse comando permite identificar e desabilitar o serviço, fechando as portas a ele associadas.

O problema relatado neste item é multiplicado pelo número de máquinas visíveis, principalmente na Internet. Assim, é importante que se tenha um sistema de firewall controlando o trânsito de pacotes entre redes não confiáveis entre si. O principal ponto de proteção é a conexão com a Internet, mas podem existir ligações com outras redes consideradas internas que possam merecer atenção e filtragem.

Texto 32 – Software

A maioria dos softwares, incluindo sistemas operacionais, aplicativos, banco de dados etc., vem com scripts ou programas que têm por objetivo instalar os sistemas o mais rapidamente possível, com a máxima funcionalidade e o mínimo de esforço por parte do administrador. Grande parte dos softwares apresenta essa característica, ou seja, são fáceis de instalar e de usar, visando a um retorno mais rápido do investimento feito na compra do software. Entretanto, essa premissa é sempre prejudicial para os administradores de segurança, porque, de modo geral, expõe o servidor que está hospedando o software.

Para atingir o objetivo da simplicidade, os programas normalmente instalam mais componentes do que a maioria dos usuários necessita. A filosofia do fabricante é habilitar a maior quantidade possível de funções, que não são necessárias no primeiro momento, em vez de o usuário ter que instalar funções adicionais à medida que for precisando delas.

Essa visão, embora conveniente para o usuário e para o marketing do produto, abre um grande conjunto de vulnerabilidades críticas. Não pode haver descuido de quem está instalando, mas muitos usuários desconhecem o que realmente é instalado, deixando programas perigosos no sistema, simplesmente porque não sabem que estão presentes.

Os serviços vulneráveis fornecem meios para os atacantes invadirem seus sistemas. No que diz respeito a sistemas operacionais, as instalações-padrão normalmente incluem serviços adicionais, abrindo assim as portas a eles associadas. É justamente por essas portas que os atacantes costumam invadir. Quanto menos portas abertas, menor a probabilidade do sistema ser invadido.

O mesmo problema ocorre com os aplicativos cujas instalações-padrão incluem scripts ou programas de exemplos, desnecessários na maioria dos casos.

Uma das vulnerabilidades mais sérias para servidores web está relacionada aos scripts de exemplo. Os atacantes os usam para invadir o sistema ou obter informações sobre ele e é bastante comum os administradores ignorarem as instalações desses scripts.

Os scripts de exemplo constituem um grave problema, pois normalmente não passam por um processo de controle de qualidade tão apurado quanto outros softwares. Aliás, comumente, eles são escritos sem preocupação alguma com a segurança. A verificação de erros é freqüentemente esquecida, criando, com isso, um terreno fértil para vários tipos de ataques, como, por exemplo, o *buffer overflow*.

Para verificar se existem tais falhas, recomenda-se executar alguma ferramenta de scan de portas e de vulnerabilidades contra qualquer sistema que irá ser conectado à Internet. O ideal é executar o scan antes e depois da instalação, comparando e analisando os resultados obtidos com bastante cuidado. Como vimos, do ponto de vista de segurança, os sistemas devem funcionem com o menor número de serviços e de pacotes de software necessários para executar as tarefas requeridas pelo sistema.

Qualquer programa ou serviço adicional constitui uma ameaça, possibilitando sua utilização por um atacante, especialmente porque grande parte dos administradores de sistemas não corrige os programas e serviços que efetivamente não estão sendo usados.

Para se proteger dessa vulnerabilidade, o ideal é remover todos os softwares desnecessários, desabilitar os serviços inoperantes e fechar portas que não serão utilizadas.

Como essa “limpeza” pode ser uma tarefa longa e árdua, muitas das grandes empresas desenvolveram, para cada sistema operacional e conjunto de aplicativos usados, diretrizes padronizadas de instalação, que incluem a instalação das funcionalidades mínimas necessárias para que o sistema atue de maneira eficaz.

Pensando nisso, o *Center for Internet Security* desenvolveu um *benchmark* para avaliar a configuração mínima de segurança em sistemas Solaris e Windows 2000, resultado da experiência e do conhecimento de mais de 170 organizações de diversos países. Vale a pena uma consulta ao site do CIS para acompanhar essa avaliação e utilizá-la sempre que necessário.

As ferramentas de teste e de *benchmarking* para outros sistemas estão em fase de desenvolvimento, assim é recomendável uma visita periódica ao site do CIS.

As ferramentas do CIS servem para testar o nível de segurança e comparar a segurança de sistemas entre as várias divisões de uma empresa, além de poderem ser utilizadas para melhorar a dos sistemas operacionais.

O grande problema com os softwares instalados sem que o administrador tome conhecimento é que eles se tornam velhos e obsoletos, sem que o administrador saiba de sua existência. Normalmente, um software não apresenta vulnerabilidade ou defeito, quando é disponibilizado. Mas, se tiver um defeito, descoberto mais tarde, que gere uma vulnerabilidade que seja, esta pode propiciar um ataque que culmine em uma invasão. Neste caso, muito provavelmente o administrador estará contando com a sorte de não ser atacado.

Texto 33 – Ciclo do Software

De modo geral, um software novo (ou nova versão de um software, como é mais comum falarmos) é, distribuído sem nenhum defeito ou vulnerabilidade conhecida. Esta versão é dita **confiável**.

Com o passar do tempo, é possível descobrir defeitos (bugs). Se o fato for amplamente divulgado, o software pode ser classificado como defeituoso, com **bug conhecido**.

Caso o defeito no software gere uma vulnerabilidade que permita ataque, então teremos um software perigoso com **vulnerabilidade conhecida**.

Dependendo de quem descubra o defeito, às vezes divulga-se um outro programa ou script para explorar a vulnerabilidade encontrada: o chamado *exploit*. Sua principal função é automatizar um ataque, de modo que a pessoa que o utiliza não precise entender nada de programação ou sistemas operacionais. Este é o pior dos casos e, pior ainda, se o software vulnerável for um prestador de serviço na rede e se a vulnerabilidade puder ser explorada remotamente – o sonho dos crackers. Comumente o ataque visa conseguir acesso remoto ao sistema atacado e, no melhor dos casos para o atacante, um acesso de root.

Como você vê, todo cuidado é pouco. Os atacantes se esmeram em buscar as novas vulnerabilidades publicadas e atacam estes softwares, através de *exploits* publicados na Internet.

O software que contém exploração conhecida de uma vulnerabilidade que ainda não possui correção é um **software vulnerável**. O administrador tem duas opções: retirar o sistema servido por este software de operação ou contar com a sorte de não ser atacado, até que uma versão corrigida seja publicada. Isso, é claro, quando o administrador participa ativamente de listas de divulgação de vulnerabilidades.

Por outro lado, junto com a descoberta do defeito, pode ser divulgado um *patch* para corrigi-lo. Isto é bastante comum em softwares de código aberto (*open source*), em que o pesquisador (*hacker*) tem os fontes e pode descobrir como consertar o defeito. Neste caso, um administrador consciente, e que domine a tecnologia, pode buscar os fontes originais, aplicar o *patch* ao código fonte, recompilar e gerar novo binário corrigido. Algo que consome uma ou duas horas de trabalho. Normalmente, o pesquisador envia o *patch* também para o mantenedor (pessoa responsável) do software e este pode aplicá-lo e liberar, em

poucas horas, uma versão oficial, corrigida e testada.

Se o código não é aberto ou se não se domina a tecnologia, resta esperar que a empresa responsável pelo software libere uma correção ou uma nova versão.

Com o tempo, o software vulnerável é esquecido, face a novas vulnerabilidades descobertas em outros softwares. Pode-se, então, classificá-lo como **software obsoleto**. Hoje em dia, com a automação dos sistemas de scan de rede, um software obsoleto é tão perigoso quanto um com vulnerabilidade recém-publicada.

Este é o principal efeito (problema) das instalações padrões de sistemas, em que o administrador não tem exata consciência do que foi instalado: como ele não sabe que aquele software está instalado, nunca irá verificar se existe um *update* para ele. E, algum tempo após a instalação, teremos uma máquina com software obsoleto, pronta para ser invadida. Este é o caso da maioria das invasões.

Texto 34 – Configuração

Uma vez instalado, um software que presta um serviço de rede, por exemplo, deve ser colocado para rodar a cada boot da máquina, de maneira automática. Este lançamento inicial é feito por uma série de scripts que rodam durante o boot.

O init – pai de todos.

Tudo começa com o kernel: ele identifica todos os componentes (hardware) da máquina e os configura adequadamente. Depois, monta o diretório raiz (/), com a partição informada no próprio kernel ou passada para ele pelo programa que carregou o linux (Linux Loader – LILO). Após a fase inicial, já com o / montado, o kernel chama o primeiro processo: o “init”. Este processo lê seu arquivo de configuração: /etc/inittab, que é o coração do boot de um UNIX. Tudo o que vai ser chamado parte desta configuração. Até aqui, os vários sistemas UNIX são bastante semelhantes. As configurações do arquivo e tudo o que ele faz o init chamar daqui para frente estabelecem as diferenças entre os diversos sistemas de boot.

O arquivo de configuração é escrito na ordem em que os scripts de boot são

chamados. Cada linha deste arquivo tem vários campos, que são separados por dois pontos (:). Mas a utilização ou não de uma linha depende do segundo campo (com números): a linha será usada somente quando o número for equivalente ao nível que a máquina estiver rodando.

Já na configuração, é possível perceber se o sistema segue a linha de boot BSD ou System V, que veremos adiante.

Nível de operação (*runlevel*)

Historicamente, os UNIX possuem sete níveis de operação, numerados de 0 a 6. Eles permitem ao administrador programar perfis de boot, que podem ser escolhidos via teclado no momento de iniciar o boot. Alguns níveis já têm perfis fixos, que não devem ser alterados. São eles:

0 – Parada – usado quando se quer desligar a máquina;

1 – Administração – (o original *single user)* apenas um usuário deve usar. Nenhum serviço é carregado, sendo ligado o mínimo para possibilitar o uso do console da máquina. Então, este nível é feito para o root (o único usuário) fazer manutenção, com a certeza de que ninguém estará usando nada nesta máquina. Até os serviços de rede ficam desabilitados;

6 – Reiniciar – equivale a reiniciar (reboot) a máquina.

Os níveis de 2 a 5 são para uso geral e cada fabricante de UNIX escolhe como usá-los.

O inittab tem uma linha que indica qual o nível padrão (automático) no qual o sistema entrará sozinho. É a linha que contém a palavra *initdefault*. Veja e comprove que o nível automático é o 3.

BSD

O sistema de boot BSD, mais novo que o System V, é mais simples e rápido. Parte do princípio que tudo deve funcionar, e o que não funcionar será logado para o administrador verificar depois.

Todos os scripts de boot estão no diretório /etc/rc.d/

Por exemplo, no caso do Slackware Linux, que segue o padrão de boot BSD, os scripts estão localizados no diretório /etc/rc.d, e são chamados na seguinte

ordem (verifique no arquivo de configuração /etc/inittab):

```
/etc/rc.d/rc.S
```

```
/etc/rc.d/rc.M
```

Nos dois casos, o término da execução dos scripts é aguardado para se seguir adiante.

Se formos curiosos, e sem pretensão de decorar nada disto, podemos olhar o interior destes scripts (uma aula de simples programação em shell script).

Analisando o interior do rc.S, constatamos que, dentre outras coisas, este script:

- realiza a montagem dos sistemas de arquivos especiais (proc, sys etc);
- verifica a integridade do sistema de arquivos da raiz (/), rodando o programa de verificação de disco (aquele que verifica o disco quando a máquina é desligada sem fazer *shutdown*);
- se houver configuração para tal, carrega módulos do kernel;
- verifica e monta os demais sistemas de arquivo, se existirem;
- acerta data e hora do sistema operacional a partir do relógio de tempo real da placa (CMOS);
- liga o swap (memória virtual);
- prepara, se estiver em uso, o sistema de gerenciamento de volume lógico;
- verifica o antigo sistema ISAPNP (*Plug and Play* para barramento ISA – se você não sabe o que é, não se preocupe;
- verifica se existe um script (e roda caso exista) que provê compatibilidade com o sistema System V, o /etc/rc.d/rc.sysvinit.

Note que, mesmo sendo orientado ao sistema de boot BSD, o Slackware possui compatibilidade com o System V. Com isso, se algum software instalado possuir script de boot para System V, ele irá funcionar normalmente, sendo chamado durante o boot.

O segundo script chamado, o rc.M (de Multiuser), procura por diversos outros scripts no diretório /etc/rc.d. Na versão 10.2 do Slackware, há 32 scripts chamados pelo rc.M. São exatamente o método para ativar os programas durante o boot e desativar durante o shutdown. A maior parte dos softwares instalados na distribuição, e que devem ser iniciados durante o boot, está nesses scripts. A

título de curiosidade, listamos os nomes dos scripts que são chamados, na ordem:

```
/etc/rc.d/rc.syslog

/etc/rc.d/rc.pcmcia

/etc/rc.d/rc.inet1

/etc/rc.d/rc.hotplug

/etc/rc.d/rc.inet2

/sbin/ldconfig

/usr/X11R6/bin/fc-cache

/etc/rc.d/rc.dnsmasq

/etc/rc.d/rc.cups

/etc/rc.d/rc.lprng

/etc/rc.d/rc.atalk

/usr/sbin/smartd

/sbin/genpowerd

/sbin/accton

/usr/sbin/crond

/usr/sbin/atd

/sbin/quotacheck

/sbin/quotaon

/etc/rc.d/rc.saslauthd

/etc/rc.d/rc.sendmail

/usr/sbin/apmd

/etc/rc.d/rc.acpid

/etc/rc.d/rc.alsa

/etc/rc.d/rc.font

/etc/rc.d/rc.keymap
```

```
/etc/rc.d/rc.hpoj

/etc/rc.d/rc.mysqld

/etc/rc.d/rc.httpd

/etc/rc.d/rc.samba

/etc/rc.d/rc.gpm

/etc/rc.d/rc.sysvinit

/etc/rc.d/rc.local
```

Acabou a curiosidade? Então, vamos nos concentrar no assunto que vem a seguir.

O último script, **rc.local**, embora venha vazio, é muito importante. É de uso geral, para incluir chamadas ou configurações que devem ser feitas durante o boot e que não são feitas pelos demais scripts. Também é usado para chamar no boot um software que tenha sido instalado manualmente, através da compilação dos fontes, e que precise ser chamado durante o boot. Neste tipo de instalação, normalmente, nenhum script é instalado automaticamente e, por isso, o administrador recorre ao rc.local, inserindo as linhas de comandos necessários para iniciar um software durante o boot.

O rc.local é o único script que pode ser livremente modificado pelo administrador, pois esta é a sua função. Embora não haja nenhum grande dano em se modificar os demais scripts, pois tudo é texto e é fácil editá-los, o administrador precisará tomar cuidado quando for realizado um update do Linux, pois as novas versões dos scripts certamente não terão as modificações feitas por ele.

O script rc.inet1 é o responsável por configurar as interfaces de rede local. Lê seu arquivo de configuração, o rc.inet1.conf, que configura os endereços IP de cada interface de rede, a netmask e, é claro, o gateway default. Para a utilização com modem, no entanto, não é necessário modificar este arquivo, pois ele diz respeito somente às placas de rede local.

Cabe ao script rc.inet2 configurar diversos serviços de rede, como o inetd, chamado de “super processo”. Sua função é ficar esperando por conexões em portas de serviços e, ao receber o pedido da conexão, o inetd chama o software que é o servidor do serviço. Adiante voltaremos ao assunto inetd.

System V

O boot do System V verifica se cada *daemon* (programa lançado e colocado para rodar em *background*) está ativo após seu lançamento e, geralmente, coloca um OK (verde) na tela se estiver OK. Este processo se repete para cada serviço e acaba por consumir tempo.

Os scripts principais de boot ficam no diretório /etc/init.d, mas não são chamados diretamente.

Para cada nível do UNIX (0 a 6) existe um diretório separado, chamados de

/etc/rc0.d até /etc/rc6.d.

Dentro dos diretórios são colocados links simbólicos (o equivalente a atalho no Windows) para os arquivos que estão no diretório principal (init.d). A ordem de chamada dos arquivos depende da ordem alfabética/numérica dos nomes dos links. Assim, por exemplo, um script de nome S023samba é chamado antes do S043http.

Felizmente para o administrador, existem interfaces gráficas que permitem arrastar com o mouse e escolher a ordem do que ele quer que rode, e a interface escolhe os nomes apropriadamente.

inetd

Presente nos UNIX em geral, o inetd tem como principal motivação para o seu uso o seguinte: os serviços pouco usados não necessitam que o programa servidor fique o tempo todo rodando, consumindo recursos (principalmente memória) do sistema. Então, coloca-se um único programa rodando, o inetd, que fica monitorando pedidos de conexão em várias portas. Ao chegar um pedido, chama o software responsável e encaminha o pedido.

O arquivo de configuração do inetd é o:

/etc/inetd.conf (vem pronto para uso)

O simples fato de comentar ou descomentar uma linha neste arquivo habilita ou desabilita o serviço, após o inetd ser reiniciado. Para reiniciar o inetd, usa-se o seguinte comando, estando logado como root:

```
killall -HUP inetd
```

Atenção: este comando só é adequado em máquinas com GNU/Linux. Em alguns UNIX, causa o reboot da máquina.

Texto 35 – Controlando o que deve ficar ligado

Este é o principal item desta unidade. Os anteriores são apenas uma introdução para que tenhamos idéia do que realmente é feito para controlar que serviços estarão rodando no sistema, e como iniciá-los ou não no boot.

Portas abertas

O mais importante é determinar quem e o que está prestando serviços de rede via IP. Para isto use o comando:

```
netstat -an –ip
```

Após o netstat, temos “hífem an, espaço, hífem, hífem ip”. Um exemplo de resposta:

```
Active Internet connections (servers and established)

Proto Recv-Q Send-Q Local Address           Foreign Address         State

tcp        0      0 0.0.0.0:6000            0.0.0.0:*               LISTEN

tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN

tcp        0      0 200.20.10.11:45703      200.999.999.121:22
ESTABLISHED

tcp        0      0 200.20.10.11:36039      200.999.00.162:22
ESTABLISHED

udp        0      0 0.0.0.0:514             0.0.0.0:*
```

A saída deste comando informa o protocolo de transporte (tcp ou udp), o número de pacotes na fila de entrada e saída, o endereço local, o endereço remoto, e o estado do serviço. Endereço IP com 0.0.0.0 indica “qualquer IP”. Alguns IPs neste

exemplo são fictícios e inválidos.

Cada serviço tcp possui uma linha em LISTEN (ouvindo[10]). Cada serviço UDP (aparecem sempre mais para o fim da lista) mostra somente uma linha com o IP local em que está atendendo, a porta e qual IP remoto pode atender.

É exatamente esta linha que indica se há uma porta aberta. Porta aberta é o termo simples para dizer que existe um servidor atendendo a pedidos de serviços naquela porta tcp ou udp.

Ter portas abertas é normal (se não nada funciona!). O que é importante é determinar se a máquina em questão deveria ou não estar com estas portas abertas.

Trata-se de uma máquina cliente: não tem servidor convencional algum rodando nela, exceto o servidor de X (porta 6000) e o ssh (porta 22) em tcp, e o serviço de log remoto (porta 514/udp). Confira isto no arquivo /etc/services.

Além disso, alguém, local na máquina, estava com duas conexões TCP estabelecidas (ESTABLISHED) para a porta 22 de outras máquinas. Ou seja, havia duas conexões SSH em uso.

A seguir, apresentamos a saída do comando netstat para um servidor. Mostramos, aqui, apenas um quarto (1/4) da saída. Usaremos esta saída mais tarde, para um exercício. Por enquanto, veja os diferentes estados de uma conexão TCP e determine quais são os serviços tcp e udp ativos.

Será muito útil se aprofundar nos estudos acompanhando os estados das conexões TCP.

As conexões que envolvem o IP 127.0.0.1 são conexões internas, onde um software realiza operações dentro da própria máquina.

```
Active Internet connections (servers and established)

Proto Recv-Q Send-Q Local Address           Foreign Address         State

tcp        0      0 127.0.0.1:11553         0.0.0.0:*               LISTEN

tcp        0      0 0.0.0.0:3306            0.0.0.0:*               LISTEN

tcp        0      0 0.0.0.0:587             0.0.0.0:*               LISTEN
```

10 A tradução feita nas versões brasileiras é “ouça”. Mas o estado em que se encontra o serviço é “ouvindo” a porta à espera de pedidos de conexões. Deve-se prestar atenção ao contexto.

```
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN

tcp        0      0 200.999.00.999:53       0.0.0.0:*               LISTEN

tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN

tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN

tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN

tcp        0      0 0.0.0.0:443             0.0.0.0:*               LISTEN

tcp        1      0 127.0.0.1:11553         127.0.0.1:36141
CLOSE_WAIT

tcp        0      0 200.999.00.190:25       200.999.00.29:4911
ESTABLISHED

tcp        0      0 200.999.00.190:25       60.999.999.52:4292
ESTABLISHED

tcp        0      0 200.999.00.190:25       201.99.99.999:1997
ESTABLISHED

tcp        0      0 200.999.00.190:25       168.999.99.8:29009
ESTABLISHED

tcp        0      0 200.999.00.190:25       200.999.00.196:43837
TIME_WAIT

tcp        0      0 200.999.00.190:25       200.999.00.39:1379
ESTABLISHED

tcp        0     81 200.999.00.190:25       200.999.00.34:10324
ESTABLISHED

tcp        0      0 200.999.00.190:25       200.999.999.217:1184
TIME_WAIT

tcp        0      1 200.999.00.162:36143    66.999.000.22:2703
SYN_SENT

tcp        0      0 127.0.0.1:36109         127.0.0.1:11553
TIME_WAIT
```

```
tcp        0      0 200.999.00.190:25       157.999.000.62:1458
TIME_WAIT

tcp        0      0 200.999.00.162:36122    200.999.00.146:25
TIME_WAIT

tcp        0      0 200.999.00.162:36128    200.999.00.146:25
TIME_WAIT

tcp        0      0 200.999.00.190:25       201.999.00.164:3331
TIME_WAIT

tcp        0      0 200.999.00.190:25       24.999.00.11:2482
TIME_WAIT

tcp        0      0 200.999.00.190:25       201.999.00.16:1499
ESTABLISHED

tcp        1      0 200.999.00.190:25       24.999.000.26:3027
CLOSE_WAIT

tcp        0      0 200.999.00.162:443      201.999.00.2:50514
TIME_WAIT

tcp        0      0 200.999.00.162:443      201.999.00.2:50513
TIME_WAIT

tcp        0   4192 200.999.00.162:22       200.00.00.11:36039
ESTABLISHED

tcp        0      0 200.999.00.190:25       69.000.00.92:4694
ESTABLISHED

tcp        0      0 200.999.00.190:25       200.000.00.9:1192
ESTABLISHED

tcp        0      0 200.999.00.190:25       88.222.96.37:1137
TIME_WAIT

tcp        0      0 200.999.00.190:25       200.000.0.203:11774
TIME_WAIT

tcp        0      0 200.999.00.190:25       69.999.000.102:4128
TIME_WAIT

tcp        0      0 200.999.00.190:25       161.999.000.108:32891
```

```
TIME_WAIT

udp        0      0 127.0.0.1:59392         127.0.0.1:53
ESTABLISHED

udp        0      0 0.0.0.0:45825           0.0.0.0:*

udp        0      0 0.0.0.0:514             0.0.0.0:*

udp        0      0 127.0.0.1:54442         127.0.0.1:53
ESTABLISHED

udp        0      0 127.0.0.1:31277         127.0.0.1:53
ESTABLISHED

udp        0      0 127.0.0.1:36657         127.0.0.1:53
ESTABLISHED

udp        0      0 200.999.00.999:53       0.0.0.0:*

udp        0      0 127.0.0.1:53            0.0.0.0:*

udp      740      0 127.0.0.1:45750         127.0.0.1:53
ESTABLISHED

udp        0      0 10.1.0.4:36664          10.1.0.1:53
ESTABLISHED

udp        0      0 127.0.0.1:62012         127.0.0.1:53
ESTABLISHED

udp        0      0 127.0.0.1:10688         127.0.0.1:53
ESTABLISHED

udp        0      0 0.0.0.0:123             0.0.0.0:*
```

Texto 36 – Desligando o serviço

Para desligar um serviço, basta parar o processo (programa) que o está prestando.

A maneira mais rápida, mas não necessariamente a melhor, é enviar um sinal de terminação ao processo com o comando kill. Para descobrir quem é o programa (processo) que está mantendo uma porta aberta, usa-se o comando netstat, mostrado abaixo já com sua saída:

```
netstat -anp –ip

Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name

tcp        0      0 0.0.0.0:6000            0.0.0.0:*               LISTEN
4045/X

tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
3614/sshd

tcp        0      0 200.20.10.11:45703      200.239.245.121:22
ESTABLISHED 5982/ssh

tcp        0      0 200.20.10.11:36039      200.142.88.162:22
ESTABLISHED 4591/ssh
```

A saída normalmente possui muitos caracteres por linha. Como o terminal (console) possui um tamanho limitado de caracteres por linha, cada linha da saída não cabe em uma linha só do terminal.

A listagem nos diz que o programa que está ouvindo na porta tcp/6000 é o “X” e ele está rodando com o PID 4045. E, ainda, o serviço da porta tcp/22 (SSH) está sendo prestado pelo programa com PID 3614 e se chama “sshd”.

Então, para parar o serviço SSH do exemplo podemos usar o seguinte comando:

```
kill 3641
```

Embora este método não seja o mais adequado, ele realmente irá parar o serviço e é o mais genérico, pois funciona em qualquer UNIX.

No entanto, o administrador também deve procurar aprender os métodos baseados nos scripts e interfaces tipo texto ou gráficas de administração. De modo geral são típicos e, às vezes, proprietários de uma distribuição GNU/Linux ou um UNIX.

No caso do Slackware Linux, estes serviços são administrados por meio dos scripts presentes no diretório /etc/rc.d. Assim, veja os scripts nele disponíveis.

Para o exemplo do servidor de SSH, o método de parada é:

```
/etc/rc.d/rc.sshd stop
```

Derruba todos os processos sshd que estejam rodando, e não somente o processo que espera por novos pedidos de conexão. Com isto, sessões de SSH em uso serão derrubadas. Tendo consciência disto, o administrador pode usar um método ou outro, de acordo com suas reais intenções.

No entanto, o comando:

```
/etc/rc.d/rc.sshd restart
```

Reinicia apenas o processo pai de todos, que está esperando por novos pedidos de conexão. Ele é usado quando trocamos uma configuração do sshd ou instalamos uma nova versão. As conexões em uso não são perdidas.

Texto 37 – Iniciando no boot

Uma vez instalado um software, normalmente um servidor, deve-se controlar se ele deve ser iniciado ou não, no boot, para prestar o serviço. Para realizar o controle, o primeiro passo é descobrir se o sistema que se deseja controlar é servido pelo inetd, ou se o daemon do servidor trabalha sozinho. Como as distribuições já vêm com o arquivo de configuração do inetd preparado, basta verificar se há uma linha referenciando o serviço em questão.

Inetd

Se o serviço é controlado pelo inetd, então o programa que presta o serviço não é iniciado durante o boot. Para que seja prestado, basta que a linha referente a ele no arquivo

/etc/inetd.conf esteja sem comentário, ou seja, sem a “#” no início da linha.

Certifique-se de que o próprio inetd está sendo iniciado durante o boot.

Standalone

Os serviços que não dependem do inetd e que rodam com seus próprios processos dedicados à espera de um pedido de conexão são chamados, durante o boot, pelos scripts.

No caso do Slackware, que segue a linha BSD, estes scripts devem ser colocados como executáveis para que, no próximo boot, rodem automaticamente. Da mesma forma, tirar a permissão de execução faz com que um serviço não mais esteja rodando depois do próximo boot.

Exemplo: Para ligar o servidor que presta o serviço de SSH no próximo boot, usa-se o comando:

```
chmod +x /etc/rc.d/rc.sshd
```

Normalmente as distribuições que seguem a linha do System V possuem interfaces gráficas para esta configuração. Como vimos anteriormente, o procedimento básico realizado pela interface é a criação de um link simbólico no diretório correspondente ao nível que está rodando o sistema, para o script do serviço, no diretório /etc/init.d. Para que o serviço não seja mais iniciado no próximo boot, o link simbólico deve ser apagado.

Exercício

Localize e analise o script rc.sshd. Qual o comando usado para derrubar todos os processos? Atenção: este comando só funciona desta forma no GNU/Linux.

Unidade 4 – Ataques conhecidos

A maioria dos ataques bem-sucedidos é alcançada pela exploração de vulnerabilidades encontradas nos sistemas operacionais. Utilizam as falhas mais comuns, por meio de ferramentas evidentes e difundidas na própria Internet.

É fundamental a atenção de administradores de rede e de segurança para a correção das falhas de segurança e de vulnerabilidades do sistema.

Neste Unidade, abordaremos a questão dos ataques mais conhecidos e a proteção contra eles.

Texto 38 – Ataques conhecidos

Boa parte dos ataques bem-sucedidos, principalmente os que culminam em invasão, é alcançada por meio de vulnerabilidades encontradas nos sistemas operacionais e softwares. A maioria dos atacantes são oportunistas e escolhem caminhos fáceis e convenientes. Exploram as falhas mais comuns, utilizando ferramentas evidentes e difundidas na própria Internet. Partem do princípio de que as empresas não aplicam as últimas correções disponíveis no mercado e, constantemente, procuram sistemas expostos na Internet.

Há grande quantidade de listas de discussões dos próprios fornecedores de software, com o propósito de divulgar, para os usuários, correções e falhas encontradas. Mas, como as listas são públicas, os próprios atacantes também utilizam suas informações.

A maioria dos administradores de rede e de segurança tem dificuldade em priorizar as falhas mais perigosas pelos seguintes motivos: desconhecem as vulnerabilidades, estão ocupados corrigindo as falhas existentes dentro de uma rede que foi submetida a algum tipo de verificação ou, mesmo, ocupados em manter a segurança atual.

Além dos ataques conhecidos e registrados ao longo dos anos, o *System Administration, Networking and Security* (SANS) e o *National Infrastructure Protection Center*, FBI (NIPC/FBI) divulgaram um documento que descreve as mais críticas vulnerabilidades de segurança na Internet: *The Top 20 Most Critical Internet Security Vulnerabilities* ou, simplesmente, “Top 20”. Seu conteúdo é dividido em três categorias: vulnerabilidades gerais, vulnerabilidades no Windows e vulnerabilidades no UNIX.

A lista do SANS/FBI é muito valiosa, porque a maioria dos ataques bem-sucedidos, direcionados a sistemas de computadores através da Internet, pode ser atribuída à exploração das falhas de segurança aí incluídas. Milhares de empresas no mundo a usam para priorizar esforços e investimentos a fim de se prevenir, o que não significa que devemos somente nos preocupar com a lista. Ela serve como ponto de partida para priorizar esforços, manter-se sempre atualizado e incluir novas vulnerabilidades pertinentes a cada particularidade da empresa.

Um dos grandes aliados dos administradores de segurança é a auditoria na rede, também chamada de scan de vulnerabilidades. Nessa varredura, é possível

detectar falhas e corrigi-las a tempo, porém algumas ferramentas de scan de vulnerabilidades procuram por 300, 500 ou até mesmo 1.000 falhas, diminuindo assim o foco principal dos administradores de sistemas, que precisam garantir que todos os sistemas encontram-se protegidos dos ataques mais comuns.

A lista Top 20 foi criada justamente para ajudar os casos mais delicados. Combina o conhecimento de dezenas de especialistas em segurança de reconhecidas agências federais, das principais firmas de consultoria, dos fabricantes de software de segurança, do CERT/CC e do instituto SANS e inclui os melhores programas na área de segurança mantidos nas universidades. Trata-se de um documento dinâmico, periodicamente revisto, que permite que você se mantenha atualizado.

Bob Todd, autor da ferramenta de scan SARA, criou uma versão especial da ferramenta para identificar a presença das vulnerabilidades da lista Top 20 do SANS/FBI. Pode ser obtida no site do *Center for Internet Security* [CIS]. Há também, nas referências, vários fornecedores dessas ferramentas de scan, cabendo um estudo mais aprofundado para quem for usá-las.

As ameaças que podemos citar são hackers, antigos funcionários ou funcionários insatisfeitos, parceiros extranet e usuários curiosos que querem ter posse de determinada informação para uso pessoal ou para benefício próprio.

A vulnerabilidade nas empresas se dá pela falta de uma política de segurança (regras e métodos de proteção a serem usados dentro da empresa), sistemas desatualizados (principalmente versões antigas com furos de proteção conhecidos e ainda não corrigidos), gestão inadequada dos softwares e dispositivos existentes praticados por pessoas sem o conhecimento necessário para tal.

Texto 39 – Perfil e tipos de hacker

A maioria dos hackers é jovem, trabalha em projetos de computadores e é um técnico altamente especializado. O que os motiva a desempenhar suas atividades? *Status* por terem conseguido quebrar algum sistema de segurança, ganhando mais respeito dos seus “colegas de profissão”. Vingança de alguém que o prejudicou, obtenção de vantagens lucrativas, qualquer que seja o motivo, o verdadeiro hacker é sempre difícil de ser localizado, pois, para manter-se no

anonimato, ele consegue apagar todas as pistas deixadas após uma invasão.

Hackers x crackers

Convencionou-se chamar qualquer pessoa especializada em sistemas de computação de hacker. De modo geral, são pessoas autodidatas que adoram "fuçar" um sistema ou programa, para entender como ele funciona e, assim, vão aprendendo.

Mas existem diferenças de caráter. Os que usam suas habilidades para solucionar problemas de segurança em sistemas, contribuindo para o desenvolvimento de correções (*patches*), são chamados de hackers. O seu oposto em termos de comportamento (mas com capacitação técnica similar), ou seja, os que invadem sistemas com finalidades ilícitas, visando ao lucro, roubo de informações, reconhecimento perante a comunidade etc., são chamados de crackers.

Apesar disto, mesmo neste texto, o termo hacker está sendo usado indevidamente, pois o hacker não ataca, somente o cracker o faz. Infelizmente, a mídia inseriu o termo hacker indiscriminadamente, de maneira que se formou um consenso sem que se preste atenção no conceito aqui exposto.

Script kiddies

São crackers que não fazem nada original, apenas usam informações, ferramentas e programas desenvolvidos por terceiros para realizar suas invasões. A maioria das tentativas de invasão é feita por *script kiddies*.

Funcionários insatisfeitos e ex-funcionários

É importante notar que ataques não partem apenas de fora da rede corporativa. Grande parte das invasões bem-sucedidas vem de dentro da própria empresa. A disponibilidade de informações sensíveis na rede para diversos funcionários (mesmo os não autorizados) pode comprometer bastante a política de segurança implementada.

Texto 40 – Engenharia social

Dentro da área de segurança, podemos definir engenharia social como a aquisição de informações preciosas ou obtenção de privilégios de acesso por um indivíduo fora da organização, baseado em uma relação de confiança estabelecida, inadequadamente, com um indivíduo de dentro da organização.

Basicamente, é a arte de fazer com que outras pessoas concordem com você e atendam a seus pedidos ou desejos, mesmo que você não tenha autoridade para tal. Em outras palavras, pode-se dizer que engenharia social é simplesmente a arte de se contar uma mentira bastante convincente.

Por mais extraordinário que possa parecer, o método mais simples, usado e, talvez, eficiente de se recolher informações é simplesmente chegar e perguntar. E assim, adquirem-se informações confidenciais, como organogramas de organizações, números de cartões de crédito e telefone, senhas de acesso, diagrama da rede etc., com o objetivo de avaliar as vulnerabilidades de uma organização para futuros ataques.

De fato, o método é bastante utilizado e existem crackers que sabem usá-lo com grande destreza. Um bom exemplo é o do famoso Kevin Mitnick, especialista em contar histórias para conseguir as informações que precisava.

Geralmente, esse tipo de aproximação envolve mais do que simples raciocínio rápido e uma variedade de frases na ponta da língua. Engenharia social pode envolver muito trabalho de aquisição de informação antes de uma real ação de qualquer tipo. Para iniciar um ataque, a maior parte do trabalho está na preparação, e não no próprio ataque.

Dizem que o único computador totalmente seguro é aquele desligado da tomada. A arte da engenharia social concentra-se no elo mais fraco de corrente da segurança dos computadores: os seres humanos. O simples fato de que se pode facilmente convencer uma pessoa a ligar o computador, torna vulneráveis, até mesmo, os computadores desligados.

Na medida em que a parte humana de um sistema de segurança é a mais essencial, porém não há computador algum na face da Terra que não necessite de seres humanos.

Isso significa que é uma fraqueza universal, independentemente de plataforma, software, tipo de conexão de rede ou idade do equipamento. Qualquer pessoa

com acesso a partes do sistema, física ou remota, é uma potencial falha de segurança . Alguma informação adquirida pode ser utilizada para um outro tipo de ataque de engenharia social, o que significa que uma pessoa aleatória, mesmo que não seja considerada integrante da política de segurança, pode servir como uma porta de entrada.

O primeiro método é também o mais óbvio. Um pedido simples e direto, em que se solicita ao indivíduo-alvo que execute uma determinada tarefa. Embora esse método seja o menos provável para produzir um resultado positivo, é, com certeza, bastante simples, o indivíduo sabe exatamente o que você quer que ele faça.

O segundo método requer muito trabalho por parte de quem faz o ataque e, com certeza, envolve recolhimento de informação e conhecimento prévio do alvo. Consiste em criar uma situação em que o indivíduo é apenas uma parte dela.

Se a situação proposta, real ou imaginária, possuir certas características, o indivíduo-alvo estará mais propenso a concordar com os pedidos. Essas características incluem:

- difusão da responsabilidade – o alvo acredita que não é o único responsável por suas ações e pelas informações que venha a divulgar, que o atacante irá manter a responsabilidade longe do alvo;
- troca de favores – o alvo acredita que está prestando um favor ao atacante que se mostrará extremamente grato. As pessoas geralmente ficam mais dispostas a cooperar quando acreditam que poderão obter alguma vantagem no futuro;
- dever moral – o alvo coopera, pois acha que é a atitude certa a tomar. É seu dever moral. Parte disso é culpa. As pessoas procuram evitar o sentimento de culpa e farão o possível para evitá-lo.

Os atacantes escolhem seus alvos levando em consideração envolvimento com a empresa, experiência e tempo de trabalho. Alunos, estagiários, secretárias e profissionais iniciantes mostram-se sempre mais dispostos a cooperar. Isto porque, em geral, esses indivíduos têm, ainda, pouco conhecimento e experiência a respeito do sistema-alvo e desejam cooperar.

Um bom início para ganhar a confiança do alvo, por exemplo, é ser gentil, utilizando um tom de voz calmo (se ao telefone) e sendo amável.

A solução está na educação e difusão da informação, explicando aos empregados

e pessoas ligadas direta ou indiretamente ao sistema a importância da política de segurança, evitando, assim, ataques de pessoas que poderão tentar manipulá-los para ganhar acesso a informações privadas. Trata-se de um excelente começo para tornar seguros as redes ou sistema.

Texto 41 – Anatomia de um ataque de cracker

Antes de qualquer atividade ilícita, o cracker gasta de 80% a 90% do seu tempo em busca de informações sobre sistemas internos da empresa. Quanto mais informações coletadas, maiores serão as chances de um ataque bem-sucedido. Como exemplo de informações básicas que são coletadas, citamos:

- endereço IP e/ou endereço de rede;
- sistemas operacionais utilizados e softwares instalados nos servidores;
- topologia da rede;
- perfil e atividade principal da empresa.

Por meio de ferramentas de rede, um cracker consegue buscar informações importantes que o ajudarão num ataque futuro. Basicamente existem dois métodos de coleta de informações:

- técnico – usar ferramentas de software para levantar informações, como *sniffers*, pings, telnet e traceroute;
- engenharia social – o cracker engana pessoas para conseguir informações úteis para seu ataque. Por exemplo: passa por uma pessoa importante da própria empresa e, valendo-se da boa-fé de funcionários, consegue informações vitais que possam direcionar futuros ataques.

Texto 42 – Honeypots

A tradução literal seria “pote de mel”. Trata-se de uma técnica ou tecnologia utilizada para atrair o cracker para um determinado lugar e poder, assim, identificar o invasor, por onde ele veio, que comandos ou ferramentas estava

utilizando, qual era o alvo primário e o secundário, que motivações tem etc.

Com tais informações, o administrador de segurança tem um benefício enorme. Conhece detalhes intrínsecos da rede que administra, identifica pontos fracos, alvos valiosos e como melhor protegê-los.

Existem diversos tipos de honeypots, sendo os mais comuns:

- Lambs – computadores que servem de alvo, vulneráveis a ataques;
- Facades **–** computadores que simulam serviços e aplicações de rede;
- Instrumented systems **–** sistema híbrido, combina Lambs com Facades;
- HoneyNet **–** rede de honeypots, isto é, uma rede fictícia que tem diversos alvos de ataque.

Tanto os honeypots como o honeynet são entidades totalmente separadas da rede real e, pelo menos teoricamente, não trazem perigo algum de serem utilizados como ponte pelo invasor para chegar à rede real.

Existem diversas metodologias para a utilização de honeypots e, talvez, a mais popular seja a colocação de um honeypot no lugar de um servidor que vem sofrendo ataques freqüentes. Para todos os efeitos, os invasores terão a sensação de estarem atacando um servidor real, ou uma rede, se for um honeynet, e o administrador será alertado quanto ao acesso não autorizado.

A partir daí, o invasor passa a ser totalmente monitorado pelo sistema, ou seja, identifica-se de onde estão vindo o ataque, o alvo, as motivações, por onde ele entrou, que comandos está utilizando etc. O administrador de segurança passa a monitorar cada passo do invasor e, com isso, aprende com ele.

As tecnologias de honeypot avançaram muito nos últimos anos. Atualmente há soluções que, com um único computador, simulam redes completas, com usuários entrando e saindo, arquivos sendo escritos, criados, apagados, portas de acesso, serviços, aplicações, e cada servidor dessa falsa rede, com um endereço IP diferente.

Apesar da evolução, crackers bastante experientes têm suas próprias técnicas para detectar se estão em uma rede ou em um honeypot, porém a maioria dos ataques parte de pessoas inexperientes, facilmente enganadas.

Do ponto de vista corporativo, parece difícil implementar redes e servidores exclusivos para serem invadidos. Os custos e, mesmo, a validade da idéia acabam fazendo com que outras medidas ganhem prioridade, deixando os honeypots de lado. Além disso, não deixa de ser crítica a existência de máquinas

vulneráveis, mesmo que de propósito, no ambiente corporativo. E se perdermos o controle? É um risco pequeno, mas talvez seja difícil aceitá-lo.

Também há o problema do risco à imagem. Até mesmo a invasão de um honeypot pode gerar publicidade negativa. Como convencer imprensa, clientes e auditores externos que aquela máquina, na rede, registrada em nome de sua empresa ou, até, com um nome no seu domínio, realmente tem como objetivo ser invadida?

Recentemente uma máquina com nome dentro do domínio iss.net foi invadida e o ataque, divulgado. A empresa afirma que era um honeypot, mas a dúvida quanto à veracidade da informação leva a um impacto significativo na imagem da empresa em questão, justamente por ser uma empresa cuja missão é evitar que esse tipo de situação ocorra.

Os estudos a respeito dos honeypots não param de crescer e já existem empresas se especializando no assunto, com o intuito de conhecer os invasores e tornar os sistemas de segurança cada vez mais pró-ativos.

Texto 43 – Sistema de logs e auditorias incompleto ou inexistente

Uma das premissas de segurança é: “A prevenção é ideal, mas a detecção é imprescindível”.

Enquanto houver a necessidade de tráfego entre a rede interna da empresa e a Internet, existirá a oportunidade de ataque. Novas vulnerabilidades surgem a cada semana, e poucas são as maneiras de defender-se de um atacante que as use.

Em uma rede, um sistema, um servidor etc. que tenha sido atacado e sem registros, sem logs da invasão, a possibilidade de descobrir o que o invasor fez é mínima.

Na falta de tais informações, a única alternativa razoável é uma restauração completa do sistema operacional com base na mídia original, torcendo para que os dados armazenados estejam corretos, caso contrário, corre-se o risco de possuir um sistema ainda controlado pelo atacante.

Não há possibilidade de detectar um ataque sem saber o que está acontecendo no sistema e cabe aos logs essa função, indicando os sistemas que estão sendo

atacados e os que foram efetivamente invadidos.

O registro de eventos deve ser feito de maneira regular em todos os sistemas críticos e os logs, armazenados e arquivados, pois nunca se sabe quando serão necessários.

A maioria dos especialistas recomenda o envio de todos os logs a um servidor central que grave os dados em uma mídia que não possa ser apagada, de forma que o atacante não consiga adulterá-los, evitando assim sua detecção.

É importante realizar regularmente auditoria nos sistemas mais críticos. Qualquer sistema estará vulnerável se os logs não existirem ou se eles não estiverem sendo armazenados em um servidor central e copiados em mídia segura.

Configure todos os sistemas de log para registrar as informações localmente e para enviar os logs a um sistema remoto. Isso provê redundância e adiciona uma camada extra de segurança. Além disso, é possível comparar ambos os sistemas de registro e qualquer discrepância pode indicar atividade suspeita.

Adicionalmente, esse esquema permite o cruzamento de informações: uma entrada isolada no arquivo de logs de um único servidor pode não ser suspeito, mas a mesma entrada em 50 servidores de uma organização, diferindo em apenas um minuto entre uma e outra, pode ser sinal de um problema maior.

Texto 44 – Tipos de ataques

A seguir, analisaremos tipos de ataques mais comuns, sem considerar os específicos realizados por “profissionais”. Foram observados por meio de ferramentas e de vulnerabilidade disponíveis na Internet, onde infelizmente há cada vez mais informações a serem utilizadas para prejudicar os usuários da rede.

Ataques a DNS e NFS – servidores de correio eletrônico

Como DNS é um banco de dados distribuído de utilização pública, existe pouco controle sobre a veracidade das informações divulgadas pelos inúmeros domínios

existentes na Internet.

É possível, por exemplo, que um atacante, que controle o mapeamento inverso de algum domínio, tenha algum endereço IP local seu sendo mapeado no nome de uma máquina confiável de um outro sistema qualquer, o sistema-alvo.

Dessa forma, o atacante pode se passar pela máquina confiável se a autenticação no sistema-alvo for baseada no nome de quem estabeleceu a conexão.

Tal tipo de ataque pode, e de fato é, evitado na maioria dos sistemas, comparando-se o mapeamento inverso com o mapeamento direto (cross-check) em chamadas de sistema como gethostbyname. Mesmo assim, autenticação baseada em nomes é ainda mais fraca que a baseada no endereçamento IP. Por esse motivo, sempre que possível, arquivos de configuração de sistema devem conter os endereços IP das máquinas, e não seus nomes.

Normalmente, o DNS oferece informações preciosas para um atacante a respeito de uma rede privada. Uma boa política é não autorizar transferências entre servidores secundários (zone transfer), mas talvez seja insuficiente para um atacante determinado e paciente. Ele pode pesquisar todo o espaço de endereçamento via pedidos de revolução inversas.

Os ataques a DNS são em geral de três tipos: listagem de mapas, contaminação de cache e acesso remoto não autorizado.

O primeiro caso permite ao atacante ter uma visão das máquinas e da rede da vítima, tornando possível, então, definir a melhor estratégia, as máquinas mais vulneráveis e/ou com informações privilegiadas.

Já a contaminação remota de cache se dá em servidores DNS anteriores às versões 4.9.6 e 8.2.2P5, e BIND nativo do NT, sem o SP6a instalado.

Configurações que permitem qualquer máquina remota montar sistemas de arquivos são utilizadas facilmente para ter acesso a áreas do disco da vítima e, por vezes, constituem a única ação necessária para a invasão definitiva. A prevenção para ataques via NFS e contaminação de cache passa pela correta configuração desses serviços.

Servidores de correio eletrônico

O objetivo do atacante é descobrir nomes de usuários válidos em um

determinado servidor, usando um dicionário de nomes comuns e combinando-o, quando possível, com consultas ao servidor de e-mail. De posse de um nome, o atacante tenta descobrir a senha do usuário. As variações mais comuns são tentar o próprio nome como senha e variações de senhas mais comuns.

O ponto de autenticação para os usuários descobertos normalmente é o serviço de pop3, mas pode ter variações como telnet, rlogin e ftp. É importante verificar freqüentemente os logs dos servidores para descobrir ataques desse tipo.

Fragilidades no *Domain Name Service* – DNS

O pacote BIND, *Berkeley Internet Name Domain*, é a implementação mais usada do *Domain Name Service* (DNS).

O DNS nos permite localizar sistemas na Internet por meio do nome (por exemplo, www.sans.org) sem ter que saber endereços IP específicos, o que faz com que ele seja um alvo favorito para ataques.

De acordo com um estudo realizado em meados de 1999, cerca de 50% do total de servidores de DNS conectados à Internet está executando versões vulneráveis do BIND.

O típico exemplo de ataque ao BIND é o caso em que os atacantes apagam os logs do sistema e instalam ferramentas para obter acesso com privilégios de administrador.

Feito isto, eles compilam e instalam programas de IRC e ferramentas para vasculhar a Internet em busca de redes vulneráveis (*network scanners*). Já aconteceu de varrerem mais de uma dúzia de redes classe-B, à procura de outros sistemas com versões vulneráveis do BIND.

Em questão de minutos, estavam usando o sistema comprometido para atacar centenas de sistemas remotos, resultando em outras invasões bem-sucedidas.

O exemplo ilustra o caos gerado por uma única vulnerabilidade de um software como o DNS. Versões velhas do BIND incluem também vulnerabilidades do tipo *buffer overflow*, veremos a seguir, em que os atacantes exploram para obter acesso não autorizado.

Execute uma ferramenta automática de scan de vulnerabilidades (*vulnerability scanner*) para identificar a versão do BIND ou verifique manualmente os arquivos para ver se estão vulneráveis. Em caso de dúvida, seja precavido e faça uma atualização do serviço.

Desligue o daemon do BIND (chamado “named”) em todos os sistemas não autorizados a atuar como servidores de DNS. Alguns especialistas recomendam remover também o software de DNS.

Nas máquinas que são servidores autorizados de DNS, atualize a versão e aplique os patches mais recentes.

Execute o BIND como um usuário sem privilégios, protegendo-se assim de eventuais ataques remotos. No entanto, somente os processos executados como root podem ser configurados para usar portas abaixo de 1.024 - um requisito do DNS. Conseqüentemente, você deve configurar o BIND para mudar o user-id após fazer a associação à porta. Para se proteger de eventuais ataques remotos, execute o BIND numa estrutura de diretórios chroot(). Bloqueie a transferência de zonas, exceto de máquinas autorizadas. Desabilite as opções de “recursion e glue fetching”, para se defender de ataques de contaminação do cache do DNS. Configure seu servidor de modo que a esconder a versão do BIND utilizada.

Ataques a páginas web

São conseqüências de invasão ou de vulnerabilidade encontrada em um programa que atualiza página remotamente. As mudanças de páginas (chamadas pichação) vêm crescendo ultimamente não só para marcar um ataque a um determinado site, mas existem variações de caráter político. Também existem ataques realizados por crackers inconformados com a segurança frágil de um determinado site, que teoricamente teria que possuir uma segurança melhor. Por exemplo, sites militares e provedores de backbone.

Programas CGI vulneráveis

A maioria dos servidores Web, incluindo IIS, da Microsoft e Apache, suporta programas CGI (*Common Gateway Interface*) para proporcionar interatividade em páginas web, permitindo algumas funções como o levantamento e a verificação de dados.

De fato, grande parte dos servidores web é distribuída com programas CGI de exemplo, que permitem a qualquer usuário, de qualquer lugar na Internet, usar e ter uma ligação direta com o sistema operacional da máquina que abriga o servidor web. Infelizmente, muitos administradores dos servidores esquecem esses CGIs.

Os programas CGIs vulneráveis representam um alvo particularmente atraente porque são relativamente fáceis de serem localizados e operam com os privilégios do próprio servidor web. Assim, os atacantes costumam utilizá-los para desfigurar websites, roubar números de cartão de crédito ou instalar backdoors para permitir futuras invasões.

Como regra geral, os programas de exemplos que acompanham as distribuições dos servidores web sempre devem ser retirados dos sistemas de produção. Se for o caso de utilizar programas CGIs legítimos, certifique-se de se tratar da versão mais recente e execute contra o seu site uma ferramenta de scan de vulnerabilidades.

Simulando o comportamento de um atacante, você estará preparado para proteger seus sistemas. Para encontrar scripts CGI vulneráveis, você pode usar a ferramenta Whisker.

A seguir, listamos as diretrizes básicas a serem observadas para proteger seu site das vulnerabilidades nos programas CGIs:

- remova do seu servidor web de produção todos os programas CGIs de exemplo;
- examine os programas CGIs restantes e remova os que são considerados inseguros;
- assegure-se de que todos os programadores de CGI sigam uma estrita política de verificação de tamanho nos buffers de entrada;
- aplique patches para as vulnerabilidades que não podem ser removidas;
- certifique-se de que seu diretório/cgi não inclua nenhum compilador ou interpretador;
- remova o script “view-source” do diretório cgi-bin;
- não rode seu servidor web com privilégios de administrador – grande parte dos servidores web pode ser configurada para rodar como processos de algum usuário menos privilegiado, tal como o usuário “nobody”;
- não habilite suporte a CGI em servidores web que não precisem dele.

Falha no Unicode

Esta falha é também conhecida como “*Web server folder traversal*”

O Unicode é um padrão para representar caracteres, em que cada símbolo utiliza

dois bytes, o que permite um total de 65.536 combinações, com as quais é possível representar o alfabeto da maioria dos idiomas do mundo.

O padrão Unicode foi adotado pela maioria dos fabricantes de software, incluindo a Microsoft. O envio de uma URL que contém uma seqüência inválida de Unicode UTF-8 a um servidor IIS por um atacante pode forçar o servidor a executar comandos arbitrários.

Esse tipo de ataque é conhecido também como o ataque "*directory transversal*". Os caracteres equivalentes em Unicode de / e \ são %2f e %5c, respectivamente.

Entretanto, também é possível representá-los usando seqüências denominadas *overlong*: são representações inválidas de Unicode, mais longas do que o realmente requerido para representar o caractere.

Tanto / como \ podem ser representados com um único byte. Uma representação *overlong*, tal como %c0%af, representa o caractere / usando dois bytes.

O IIS não foi escrito para verificar a segurança em seqüências do tipo *overlong*. Assim, ao enviar uma seqüência de *Unicode overlong* em uma URL, as verificações de segurança da Microsoft serão contornadas.

Se o pedido for feito a um diretório marcado como "executável", o atacante poderá fazer com que os arquivos sejam executados no servidor.

Correções dessa vulnerabilidade estão disponíveis no site da Microsoft. A melhor maneira de determinar se está presente é utilizar a ferramenta hfnetchk, indicada para checar o estado de instalação de patches em um ou em vários sistemas, podendo ser utilizada na rede.

Para uma verificação mais específica, executa-se o ataque no próprio sistema, observando se é bem-sucedido. Um bom exemplo é tentar executar o seguinte comando no browser contra o servidor IIS:

http://vitima/winnt/system32/cmd.exe?/c%2Bdir%2Bc:%5C%20

Entretanto, talvez seja necessário fazer modificações na URL para testar um determinado sistema. Se o diretório scripts foi removido, o que é recomendado, o comando poderá falhar.

Por outro lado, mesmo que o comando falhe, é capaz de haver outros diretórios no servidor com permissão de escrita. Os atacantes normalmente procuram por diretórios com essa característica. Assim, é recomendado que se tenha as últimas correções instaladas e o servidor reconfigurado para permitir somente leitura em seus diretórios, se for o caso.

As ferramentas “IIS Lockdown” e “URL Scan” também protegerão contra esta vulnerabilidade. A ferramenta URL Scan funciona como um filtro para as requisições HTTP. Por exemplo, filtra pedidos que contenham caracteres codificados com UTF8.

Texto 45 – Buffer overflow

Este tipo de vulnerabilidade pode estar presente em qualquer software escrito em linguagens de programação mais eficientes, tal como o “C”, pois a linguagem não oferece proteção automática contra abusos no uso de suas variáveis e, em especial, arrays. Tal vulnerabilidade permite, em geral, que sejam executados comandos arbitrários no sistema atacado. Como exemplo típico, citamos o IIS da Microsoft.

Buffer overflow no IIS

Quando o IIS é instalado, diversas extensões de ISAPI também o são automaticamente. O ISAPI (*Internet Services Application Programming Interface*) permite aos programadores estender as potencialidades de um servidor IIS utilizando bibliotecas DLLs.

Várias DLLs, como idq.dll, contêm erros de programação que resultam na realização imprópria da checagem de erros. Em particular, não bloqueiam strings de entrada longos (*long input strings*).

Os atacantes podem enviar dados a essas DLLs, conhecidos como *buffer overflow*, resultando no controle completo do servidor IIS por parte do atacante.

O *buffer overflow* do idq.dll afeta o Microsoft Indexing Server 2.0 e o Indexing Service no Windows 2000. O *buffer overflow* do printer, o servidor Windows 2000 Server, Advanced Server e Server Data Center Edition com IIS 5.0 instalado.

A DLL vulnerável também acompanha a versão profissional do Windows 2000, mas não é mapeada na instalação-padrão. Por precaução, deve-se usar o Group Policy, onde for possível, para desabilitar a impressão via Web nas estações de trabalho.

Esta configuração de segurança será tratada posteriormente.

Para se prevenir, é necessário instalar os patches mais recentes da Microsoft. Retira-se o mapeamento de todas as extensões de ISAPI que não forem necessárias e, regularmente, verifica-se quais foram novamente mapeadas.

Recomenda-se utilizar a ferramenta IIS Lockdown para proteger servidores IIS e a ferramenta URLScan para filtrar requisições HTTP.

Brecha nos serviços de dados remotos (RDS) do IIS

Para executar comandos remotos com privilégios de administrador, atacantes exploram falhas de programação nos serviços RDS (Remote Data Services),

não sendo possível corrigi-las com um patch. Para se proteger, deve-se seguir os passos encontrados nos boletins de segurança do site da Microsoft.

Recomenda-se também a atualização para uma das versões de MDAC mais recentes, disponíveis no site da Microsoft.

Texto 46 – Falta de proteção nos compartilhamentos em redes Windows

O protocolo *Server Message Block* (SMB), conhecido também como *Common Internet File System* (CIFS), permite compartilhar arquivos em redes. A configuração incorreta do SMB pode expor arquivos críticos do sistema ou permitir seu acesso completo a qualquer usuário hostil conectado à Internet.

Muitos usuários, de forma ingênua, abrem seus sistemas aos crackers quando tentam facilitar a conveniência para colegas de trabalho e, também, a alguns usuários externos, quando abrem o acesso de leitura e escrita nos compartilhamentos realizados pela rede.

Um exemplo ocorreu em um site do governo utilizado para o desenvolvimento de um software voltado para o planejamento de missão. Administradores de rede permitiram acesso irrestrito aos arquivos, de modo que pessoas de outro departamento do governo pudessem ter fácil acesso aos arquivos. No período de dois dias, os atacantes descobriram os compartilhamentos de rede abertos e

roubaram o software de planejamento de missão.

Abrir o acesso para o compartilhamento de arquivos pode ser útil, mas é preciso tomar cuidado, principalmente em relação à escrita, para que não ocorra roubo de informação nem proliferação de determinados tipos de vírus.

Os mecanismos de SMB que permitem compartilhar arquivos também podem ser utilizados por atacantes para obter informações sensíveis dos sistemas Windows. Informações do usuário e do grupo (usernames, últimas datas de login, política de senha, informação de RAS), do sistema e determinadas chaves do registro podem ser obtidas por meio de uma conexão “null session” ao serviço de sessão do NetBIOS. São úteis aos hackers porque os ajudam a adivinhar uma senha ou descobrir uma senha via ataque de força bruta.

Um teste rápido, seguro e grátis, para identificar a presença de compartilhamento de arquivos SMB e as vulnerabilidades associadas, e que funciona em qualquer sistema Windows, está disponível no site da empresa Gibson Research Corporation.

Clicando no ícone “ShieldsUP”, obtém-se uma avaliação em tempo real se o sistema tem algum arquivo exposto por meio do SMB. As instruções detalhadas estão disponíveis para ajudar usuários do Microsoft Windows a lidar com as vulnerabilidades no SMB.

É importante observar se existe algum dispositivo que bloqueie SMB, como firewall, pois pode mostrar que não há vulnerabilidades quando de fato existem.

É o caso, também, dos usuários de cable modem, com o provedor bloqueando o SMB. A ferramenta “ShieldsUP” irá reportar que não está vulnerável, no entanto os 4.000 ou mais usuários do mesmo sistema de cabo poderão explorar essa vulnerabilidade.

O conselheiro pessoal de segurança da Microsoft (*Microsoft Personal Security Advisor*) relatará se a estação é vulnerável a ataques de SMB e poderá corrigir o problema. Como seu funcionamento é local, os resultados são confiáveis.

É aconselhável seguir os seguintes passos para defesa contra compartilhamentos desprotegidos:

- ao compartilhar arquivos, assegure-se de que somente diretórios necessários estão compartilhados;
- para segurança adicional, permita o compartilhamento somente a

endereços IP específicos, porque os nomes DNS podem ser forjados (spoofed);

- em sistemas Windows (NT e 2000), utilize o sistema de permissão de acesso a arquivos para permitir o compartilhamento somente com as pessoas que requerem acesso;
- para sistemas Windows, impeça a enumeração anônima dos usuários, grupos, configuração do sistema e das chaves de registro, por meio da conexão "null session" – veja a próxima vulnerabilidade para mais informações;
- bloqueie conexões entrantes na sua rede (inbound) ao serviço de sessão de NetBIOS (porta 139 tcp) e ao Microsoft CIFS (porta 445 TCP/UDP) no roteador ou no próprio sistema operacional;
- considere a implementação da chave de registro "RestrictAnonymous" para sistemas conectados à Internet de forma isolada ou em domínios não confiáveis – para mais informações, procure ajuda no site da Microsoft.

Texto 47 – Vazamento de informações em sessão anônima

Uma conexão de sessão nula, também conhecida como o início de uma sessão anônima, é um mecanismo que permite a um usuário anônimo obter informações (tal como nomes de usuários e arquivos compartilhados) sobre a rede ou conectar-se sem autenticação.

É usada por aplicativos como explorer.exe para listar arquivos compartilhados em servidores remotos. Em sistemas Windows NT e Windows 2000, muitos serviços funcionam sob a conta SYSTEM, conhecida como LocalSystem no Windows 2000.

A conta SYSTEM tem privilégios virtualmente ilimitados e não possui senha, o que impede que seja realizado o login com a conta SYSTEM. É usada para várias operações críticas do sistema. Quando uma máquina precisa recuperar dados de outro sistema, a conta SYSTEM abre uma sessão nula com a outra máquina.

Às vezes, precisa acessar informações em outras máquinas, como compartilhamentos, nomes de usuários, funcionalidades do tipo *Network*

Neighborhood.

Como não é possível logar nos outros sistemas com um identificador de usuário (UserID) e senha, a sessão nula é utilizada para se conseguir acesso.

Infelizmente, crackers também utilizam o mesmo mecanismo para realizar o login.

Tente conectar-se o seu sistema com uma sessão nula, usando o seguinte comando:

net use \\a.b.c.d\ipc$ “” /user:”” (onde a.b.c.d é o endereço IP do sistema remoto.)

Se você receber uma resposta “*connection failed*”, então seu sistema não é vulnerável. Se não receber resposta, o comando foi bem-sucedido e o sistema é vulnerável.

Há, ainda, o recurso de utilizar o programa “Hunt for NT”. Ele é um componente do NT *Forensic Toolkit* disponibilizado no site da Fundstone.

Os controladores de domínio requerem sessões nulas para se comunicar. Conseqüentemente, se você estiver trabalhando em um ambiente de domínio, irá minimizar as informações que os atacantes podem obter, mas não conseguirá evitar completamente o vazamento.

Para limitar a informação disponível aos atacantes, em uma máquina de Windows NT 4.0, pode-se modificar a seguinte chave do registro:

HKLM/System/CurrentControlSet/Control/LSA/RestrictAnonymous=1

Mesmo ajustando a chave de registro RestrictAnonymous para 1, ainda deixará alguma informação disponível aos usuários anônimos. No Windows 2000, você pode ajustar o valor para 2. Isto bloqueia o acesso de usuários anônimos a toda informação em que o acesso explícito não foi concedido a eles ou ao grupo Everyone (Todos), que inclui usuários anônimos.

Sempre que você modificar o registro, existe a chance de seu sistema parar de funcionar corretamente. Conseqüentemente, as mudanças precisam ser testadas. Lembre-se de fazer um backup para garantir uma futura restauração.

Se você não precisa compartilhar arquivos e impressoras, desabilite o serviço de NetBIOS do TCP/IP.

Nota: Configurar o RestricAnonymous em controladores de domínio e determinados outros servidores é capaz de comprometer várias operações

normais da rede. Por essa razão, recomenda-se que somente aquelas máquinas que são visíveis à Internet tenham esse valor configurado.

Todas as outras máquinas devem ser protegidas por um firewall configurado para bloquear NetBIOS e CIFS.

Os usuários de Internet nunca devem ter permissão de acesso a qualquer controlador interno de domínio ou a outro computador não configurado especificamente para o acesso externo.

Para evitar tal acesso, bloqueiam-se as seguintes portas no roteador ou no firewall externo: TCP e UDP 135 até 139 e 445.

Texto 48 – Codificação fraca de senhas no SAM (LAN manager hash)

Embora a maioria dos usuários de Windows não necessite do suporte do gerente de LAN (Lan Manager), a Microsoft armazena hashes de senhas do Lan Manager na configuração-padrão de sistemas Windows NT e em Windows 2000.

O Lan Manager usa um esquema (scheme) muito fraco de criptografia para as senhas, mais antigo do que o adotado em aplicativos mais recentes da Microsoft. Com isso, é possível quebrar as senhas do Lan Manager em curto período de tempo.

Mesmo os hashes de senha fortes chegam ser quebrados em menos de um mês. As principais fragilidades dos hashes de senha Lan Manager está no fato de que as senhas:

- têm tamanho fixo de, no máximo, 14 caracteres;
- quando curtas, são preenchidas com “espaços” para conter 14 caracteres;
- são convertidas para letras maiúsculas;
- são divididas em dois blocos de sete letras.

Dessa forma, é necessário apenas quebrar duas senhas de sete letras, sem mesmo ter que testar as letras minúsculas.

Além disso, o LAN Manager é vulnerável à interceptação dos hashes da senha,

fornecendo aos atacantes as senhas do usuário.

Se você possuir uma instalação-padrão do Windows NT ou 2000, estará vulnerável, já que os hashes do LAN Manager são criados por default. Teste a facilidade de quebrar sua senha em seu próprio sistema, usando uma ferramenta de quebra de senhas, como o LC3 (versão 3 do l0phtcrack).

A proteção contra as senhas fracas geradas pelo LMHash pode ser feita de duas maneiras:

- desabilitar a autenticação, usando LAN Manager através da rede e o método de autenticação NTLMv2 (NT LanManager versão 2);
- utilizar os métodos de desafio/resposta do NT LanManager que superam a maioria das fragilidades no Lan Manager(LM), têm criptografia mais forte e melhoram os mecanismos de segurança e autenticação de sessão.

Com o Windows NT 4.0 SP4 e sistemas mais novos, incluindo o Windows 2000, a Microsoft possibilitou usar somente o NTLMv2 em sua rede.

A chave do registro que controla essa opção no Windows NT e 2000 é:

HKLM\System\CurrentControlSet\Control\LSA\LMCompatibilityLevel

Se você ajustar o valor para 3, a estação de trabalho ou o servidor apresentarão somente as credenciais NTLMv2 para autenticação.

Se você ajustar o valor para 5, todos os controladores de domínio recusarão a autenticação do LM e do NTLM e aceitarão somente o NTLMv2.

É necessário planejar as mudanças com cuidado se você ainda tiver sistemas mais antigos, tais como Windows 95, em sua rede. Eles não usam NTLMv2 com o Microsoft Network Client.

Em Windows 9x, o parâmetro é:

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\LSA\LMCompatibility, e os valores permitidos são 0 ou 3 (com o *Directory Services Client*).

A opção mais segura é livrar-se daqueles sistemas mais velhos, já que não permitem que você forneça o nível mínimo da segurança que uma organização requer.

Consulte artigos úteis como:

- *How to disable LM authentication on Windows NT*, da Microsoft Technet – detalha as mudanças necessárias no registro de sistemas Windows 9x e

Windows NT/2000;

-
- *LMCompatibilityLevel and its effects* – explica os problemas com esse parâmetro;
- *How to enable NTLMv2 authentication for Windows 95/98/2000/NT*, da Technet – explica o uso do *Directory Services Client* do Windows 2000 para conseguir que os sistemas Windows 95/98 superem a limitação de incompatibilidade com o NTLMv2.

Simplesmente remover os hashes de LanMan na rede não resolve o problema pois eles ainda são criados e armazenados no SAM ou Active Directory.

A Microsoft recentemente desenvolveu um novo mecanismo para desligar a criação dos hashes de LanMan. Em sistemas Windows 2000, vá à seguinte chave do registro:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa

No menu de edição do RegEdt32 ou do RegEdit, clique Add Key e adicione uma chave chamada NoLMHash.

Depois, feche o editor de registro e reinicialize o computador. A próxima vez que um usuário mudar a senha, o computador não criará um hash LanMan.

Se essa chave for criada em um controlador de domínio Windows 2000, os hashes de LanMan não serão criados, nem armazenados no Active Directory.

Nos Windows XP, a mesma funcionalidade pode ser implementada ajustando-se o valor do registro para:

```
Hive: HKEY_LOCAL_MACHINE

Key: System\CurrentControlSet\Control\Lsa

Value: NoLMHash

Type: REG_DWORD

Data: 1
```

Isso terá o idêntico efeito ao da criação da chave NoLMHash nos sistemas

Windows 2000.

Texto 49 – Buffer overflow no RPC

As chamadas RPC (*Remote Procedure Call*) permitem que programas de um computador executem outros num diferente equipamento. Esse recurso é extensamente usado no acesso a alguns serviços de rede, tais como arquivos compartilhados via NFS ou ainda, no NIS. NIS é o Serviço de Informações de Rede do UNIX, por onde são compartilhadas informações. Bastante utilizado para implementar autenticação central de senhas e centralização de informações usadas para acessos e permissões, tais como grupos.

Muitas vulnerabilidades têm sido causadas por falhas no RPC e estão sendo ativamente exploradas. Há evidências de que a maioria dos sistemas que participaram dos ataques DDoS ocorridos durante 1999 e 2000 foi comprometida por meio de vulnerabilidades no serviço de RPC.

O ataque bem-sucedido contra sistemas militares dos Estados Unidos, incidente conhecido como *Solar Sunrise*, também explorou uma falha do RPC encontrada em centenas de sistemas no Departamento de Defesa dos Estados Unidos.

Os três serviços de RPC mais freqüentemente atacados são:

- rpc.ttdbserverd
- rpc.cmsd
- rpc.statd

São explorados pelos ataques de *buffer overflow*, pois os programas que implementam as RPCs não fazem a devida verificação de erro.

Uma vulnerabilidade do tipo *buffer overflow* permite que um atacante envie dados que o programa não está esperando e, como este não faz a devida verificação de erro, termina por passá-los para o processamento.

Os passos apresentados a seguir servem para proteger sistemas de ataques de RPC:

- onde for possível, desligar e/ou eliminar esses serviços das máquinas diretamente acessíveis via Internet;

- onde for, de fato, necessário utilizar RPC, instalar os patches mais recentes;
- consultar regularmente a base de dados de patches do fabricante, buscando novas versões e instalando-as imediatamente;
- bloquear a porta de RPC (porta 111) no roteador de borda ou no firewall;
- bloquear as portas de "loopback" do RPC: 32770-32789 (TCP e UDP).

Um documento contendo detalhes específicos sobre cada uma das três principais vulnerabilidades no RPC pode ser encontrado no site da CERT.

Texto 50 – Vulnerabilidades no sendmail

O sendmail é o programa que envia, recebe e encaminha a maioria do correio eletrônico processado em computadores UNIX e Linux. Por ser amplamente usado na Internet, constitui um alvo em potencial para os atacantes.

Ao longo dos anos, foram encontradas diversas falhas. O primeiro alerta foi emitido pelo CERT/CC, já em 1988.

Um dos ataques mais comuns acontece quando o atacante cria e envia uma mensagem de correio eletrônico, especialmente formatada, para a máquina que está usando sendmail e este a interpreta como instrução para enviar o arquivo de senhas para a máquina do atacante (ou, ainda, para outra vítima), onde, então, as senhas podem ser quebradas.

O sendmail tem um grande número de vulnerabilidades e deve ser freqüentemente atualizado, de acordo com as últimas versões e correções.

Verifique qual a última versão disponível e quais os *patches* mais recentes. Se sua versão não seguir esses requisitos, então você provavelmente está vulnerável.

Para proteger o sendmail, siga os passos:

- faça o upgrade para a versão mais recente e/ou aplique os patches recomendados;
- não o execute em modo daemon (desligue o switch -bd) em máquinas que não atuam como servidoras ou relays de correio eletrônico.

Veja a seguir um exemplo de última hora, no mundo real.

Recentemente (22/03/2006) foi divulgada uma vulnerabilidade do sendmail, até então sem nenhum *exploit* publicado. Leia o original:

> *Sendmail, Inc. has recently become aware of a security vulnerability in certain versions of sendmail Mail Transfer Agent (MTA) and UNIX and Linux products that contain it. Sendmail was notified by security researchers at ISS that, under some specific timing conditions, this vulnerability may permit a specifically crafted attack to take over the sendmail MTA process, allowing remote attackers to execute commands and run arbitrary programs on the system running the MTA, affecting email delivery, or tampering with other programs and data on this system. This vulnerability is being tracked as CVE-2006-0058 and can be found at http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-0058.*
>
> *Sendmail is not aware of any public exploit code for this vulnerability. This connection-oriented vulnerability does not occur in the normal course of sending and receiving email. It is only triggered when specific conditions are created through SMTP connection layer commands.*
>
> *Sendmail has confirmed the technical issue exposing this vulnerability and is providing patches that resolve it in our open source and commercial products. Sendmail has also alerted CERT® Coordination Center (CERT/CC), who has notified US-CERT.*
>
> *In close coordination with CERT/CC and Internet Security Systems (ISS), Sendmail has taken the following actions:*
>
> *Implemented and certified software patches for open source sendmail MTA versions 8.12 and 8.13*
>
> *Implemented and certified software patches/upgrades for impacted commercial Sendmail products*
>
> *Worked with ISS to validate the developed patches and assure their effectiveness*
>
> *Collaborated with CERT/CC to notify and provide other vendors who use the sendmail MTA with the required source code patches*

Após a divulgação, ainda no dia 22, às 13h, o autor recebeu um email da empresa responsável pelo sendmail, divulgando que já estava disponível uma nova versão com a falha corrigida:

> *Sendmail, Inc., and the Sendmail Consortium announce the availability of sendmail 8.13.6. It contains a fix for a security problem discovered by Mark Dowd of ISS X-Force. Sendmail thanks ISS for bringing this problem to our attention and reviewing the patch for it. sendmail 8.13.6 also includes fixes for other potential problems, see the release notes below for more details. Sendmail urges all users to upgrade to sendmail 8.13.6. If this is not possible,patches for 8.13 and 8.12 are available at our FTP site. However, note that those patches may not (cleanly) apply to versions other than 8.13.5 and 8.12.11, respectively. There are no patches for versions before 8.12 because those outdated versions use a different I/O layer and hence it would require a major effort to rewrite that layer. For those not running the open source version, check with your vendor for a patch.*

Apenas três horas depois, a distribuição Slackware Linux enviou ao autor um email informando que lançara novos pacotes corrigidos, às 17h16m:

> *[slackware-security] sendmail (SSA:2006-081-01)*
>
> *New sendmail packages are available for Slackware 8.1, 9.0, 9.1, 10.0, 10.1, 10.2, and -current to fix a security issue.*
>
> *Sendmail's advisory concerning this issue may be found here:*
>
> *http://www.sendmail.com/company/advisory/index.shtml*
>
> *This issue will appear in the Common Vulnerabilities and Exposures (CVE) database at the following location:*
>
> *http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-0058*

Um administrador criterioso e atento estaria com seus sistemas corrigidos, antes de terminar o expediente!

Mas, continuando o caso. Para fazer o upgrade, o sistema de correio eletrônico ficou fora do ar durante menos de 10 segundos. Digitou-se a seqüência de comandos a seguir, em uma única linha, separados por ponto e vírgula. A intenção de colocar os comandos em uma única linha é fazer com que tudo seja executado o mais rápido possível, em seqüência, na velocidade da máquina. Se fosse digitado linha a linha, certamente demoraria mais. A linha com o mv (move), que renomeia o atual binário executável do sendmail, serve para fazer um backup como forma de prevenção de desastre – todo administrador responsável sabe que "Murphy" existe.

Então, embora confiando em todos os profissionais envolvidos no desenvolvimento e preparação do pacote, não custa manter a versão antiga em backup por uns dias. Se a versão nova não funcionar, basta colocar o arquivo antigo no lugar.

Depois que o serviço voltou ao ar, apareceram impressas as datas de início do upgrade (quando o sistema foi desligado) e do término (é verdade, só atualizei no dia seguinte, que vergonha! Mas ainda não existe um *exploit* publicado.)

```
date; /etc/rc.d/rc.sendmail stop ;

    mv /usr/sbin/sendmail /usr/sbin/sendmail-8.13.3;

    upgradepkg sendmail-8.13.6-i486-1.tgz;

    /etc/rc.d/rc.sendmail start ; date

Thu Mar 23 11:21:40 BRT 2006



+==============================================================
====

| Upgrading sendmail-8.13.3-i486-2 package using ./sendmail-
8.13.6-i486-1.tgz

+==============================================================
====

Package sendmail-8.13.3-i486-2 upgraded with new package ./
```

```
sendmail-8.13.6-i486-1.tgz.



Starting MTA daemon

Starting MSP queue runner

Thu Mar 23 11:21:47 BRT 2006
```

Texto 51 – Liberdade de acesso a servidores por comandos remotos

As relações de confiança são comuns em ambientes UNIX, principalmente na administração de sistemas. Muitas companhias nomeiam um único administrador como responsável por dezenas ou até centenas de sistemas, sobrecarregando-o. Por isso, os administradores freqüentemente usam as relações de confiança e os comandos UNIX remotos para trabalhar, acessando vários sistemas mais confortavelmente.

Comandos remotos permitem ao usuário acessar um sistema sem a exigência da senha. Em vez de solicitar uma combinação de nome de usuário/senha, a máquina remota autentica qualquer usuário que tente acessá-la por meio de endereços IP confiáveis.

Assim, se um atacante conseguir controlar qualquer máquina de uma rede confiável, poderá ter acesso às demais máquinas que confiam na máquina comprometida.

Os comandos remotos usados com freqüência são:

- rlogin – remote login;
- rsh – remote shell;
- rcp – remote copy.

As relações de confiança são estabelecidas com a configuração de dois arquivos:

- /etc/hosts.equiv

- ~/.rhosts.

É importante verificar esses arquivos em sistemas UNIX para determinar se existem relações de confiança configuradas.

Recomenda-se não permitir relações de confiança baseadas em endereços IP e, preferencialmente, não usar os comandos remotos.

A autenticação baseada em endereços IP é bastante fácil de ser burlada. Deve ser feita por mecanismos mais seguros, tais como o token ou, pelo menos, por senhas. Se os comandos remotos forem necessários, convém limitar o acesso e controlar o perímetro da rede cuidadosamente.

Texto 52 – Serviços de administração remota e sistema de arquivos

O sadmind é um *daemon* que permite à administração remota de sistemas Solaris, por meio de uma interface gráfica, que disponibiliza funções de administração do sistema. O mountd é outro *daemon* que controla o acesso aos arquivos mapeados pelo *Network File System* (NFS) em hosts UNIX.

As falhas de *buffer overflow* existentes nesses aplicativos originam-se por erros de programação nos daemons. Dessa forma, possibilitam que um atacante obtenha o controle do sistema com privilégios de superusuário (root).

Para descobrir se a vulnerabilidade está presente na rede, aconselha-se utilizar alguma ferramenta de scan.

Contudo, você se protege desse tipo de fragilidade, adotando as seguintes precauções:

- desabilite ou remova os *daemons* sadmind e mountd, se for possível, das máquinas com conexão direta à Internet;
- instale os patches mais recentes do sistema operacional UNIX;
- configure listas de exportação baseadas no host e endereço IP;
- configure os sistemas de arquivo exportados como read-only e, se possível, sem acesso à suid;
- utilize o nfsbug para fazer a varredura em busca de vulnerabilidades de NFS.

Texto 53 – Serviço de impressora

Em ambientes UNIX, o in.lpd (ou nome similar) permite aos usuários interagirem com a impressora local. O Line Print *daemon* (LPD) normalmente aguarda requisições de impressão através da porta 515 TCP, porém os programadores que desenvolveram o código responsável por transferir trabalhos para impressão de uma máquina para outra cometeram um erro que originou uma vulnerabilidade de *buffer overflow*.

Se o daemon receber muitos trabalhos para impressão dentro de um curto intervalo de tempo, deixará de funcionar ou processará código arbitrário com privilégios elevados.

É recomendável utilizar alguma ferramenta de scan para verificar se essa vulnerabilidade está presente.

Outra possibilidade é a verificação manual. A maneira mais fácil de fazê-la é verificar se o sistema está executando o LPD e qual a versão adotada.

Em 30 de agosto de 2001, a Sun Microsystems divulgou um boletim de segurança, o *Sun Security Bulletin #00206*, que trata dessa vulnerabilidade e contém informações sobre os patches necessários.

Outras medidas para se proteger são:

- desabilitar o serviço de impressão remota no arquivo /etc/inetd.conf, caso seja desnecessário o seu;
- habilitar o noexec_user_stack, que é ajustado mediante a inclusão das seguintes linhas no arquivo /etc/system e reinicializando-se a máquina (reboot):

```
_ set noexec_user_stack = 1

_ set noexec_user_stack_log = 1
```

- bloqueio do acesso à porta 515/tcp.

Texto 54 – Mensagens- padrão do SNMP

O protocolo SNMP (*Simple Network Management Protocol*) é muito usado pelos administradores de rede para monitorar e administrar todos os tipos de equipamentos conectados à rede, desde roteadores e impressoras, até servidores e estações de trabalho.

Como único mecanismo de autenticação, o SNMP tem uma community string **sem criptografia**. A falta de criptografia por si só é um fato ruim. Além disso, a community string, definida como padrão e utilizada por grande parte dos equipamentos SNMP, é public. Somente alguns dos fabricantes de equipamentos de rede se preocupam com isso e alteram a community para private, quando se trata de informações mais sensíveis.

Os atacantes se valem dessa vulnerabilidade no SNMP para reconfigurar ou desligar remotamente os equipamentos. O tráfego SNMP, quando interceptado, pode revelar muitas informações a respeito da estrutura de sua rede, bem como sobre os sistemas e equipamentos a ela conectados. Tais informações servem para os invasores escolherem alvos e planejarem ataques.

Se não precisar do SNMP, o ideal é desabilitar o serviço. Pode-se bloquear o tráfego SNMP (porta 161/UDP) no roteador de borda ou no firewall, a menos que seja absolutamente necessário gerenciar equipamentos de fora da rede local.

Se houver alguma gerência pró-ativa que utilize SNMP, verifique os arquivos de configuração em busca das vulnerabilidades mais comuns, como Community name SNMP padrão ou não definida.

É importante conferir se as communities são difíceis de descobrir e estabelecer uma política para sua troca, em intervalos regulares. Sempre que possível, defina as MIBs como read only.

A ferramenta snmpwalk identificar as community names.

Texto 55 – Negação de Serviço (DoS) e SYN Flooding

O ataque que gera a negação de serviço (Denial of Service – DoS) conhecido como SYN Flood é um dos exemplos mais interessantes de ataques na arquitetura TCP/IP.

Relembrando o que vimos anteriormente, no processo de estabelecimento de uma conexão TCP, uma máquina, ao receber um pacote com a flag SYN ligada, responde com um pacote SYN/ACK à máquina que iniciou a conexão. Somente após receber um pacote ACK, a conexão TCP é finalmente estabelecida, conforme ilustra a Figura 8.

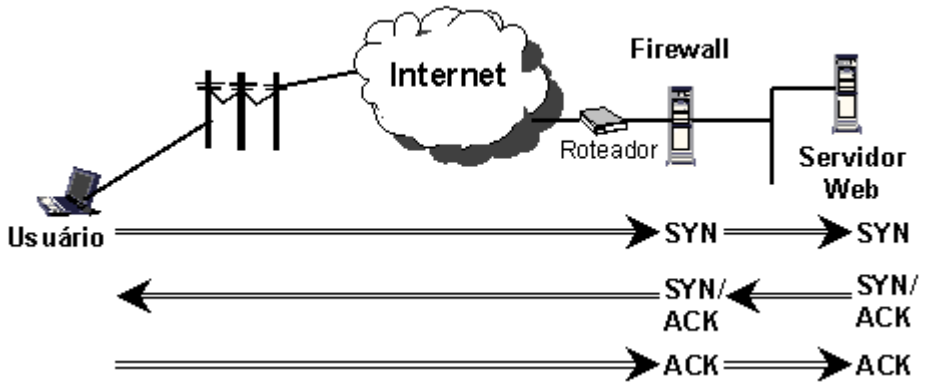


Figura 8 – Estabelecimento de conexão TCP/IP.

Na Figura foi colocado um firewall protegendo a rede da empresa, o que melhor descreve casos reais e servirá, no decorrer do texto, para mostrar como o firewall ajuda a minimizar os efeitos desse tipo de ataque.

Caso a máquina do usuário que originou o pacote SYN não envie o pacote ACK, a conexão fica em estado pendente na máquina receptora, ou seja, o pedido fica registrado numa tabela de estados para ser atendido mais tarde, quando o pacote vier . Por meio desse artifício, é possível estourar a quantidade de conexões com as quais uma máquina pode lidar, evitando-se que outras conexões sejam estabelecidas.

É importante observar que o endereço de origem da máquina atacante pode ser estabelecido de maneira aleatória, o que torna bastante difícil monitorar e identificar a fonte do ataque. A Figura 9 mostra um ataque de SYN Flood.

Como você observa na Figura 9, o objetivo do atacante é tornar o servidor indisponível à medida que irá receber uma quantidade absurda de pedidos de conexão, sem que esta seja finalizada, ou seja, o atacante não irá enviar um ACK

para o servidor.

Uma solução de contorno é identificar as conexões pendentes, durante um intervalo de tempo, e fechá-las com pacotes RST (reset da conexão) enviados à máquina de destino. Não cabe ao servidor, que irá atender aos pedidos, estipular esse tempo. Assim, a maioria dos firewalls de mercado oferece essa possibilidade de percepção, como é mostrado na Figura 10.

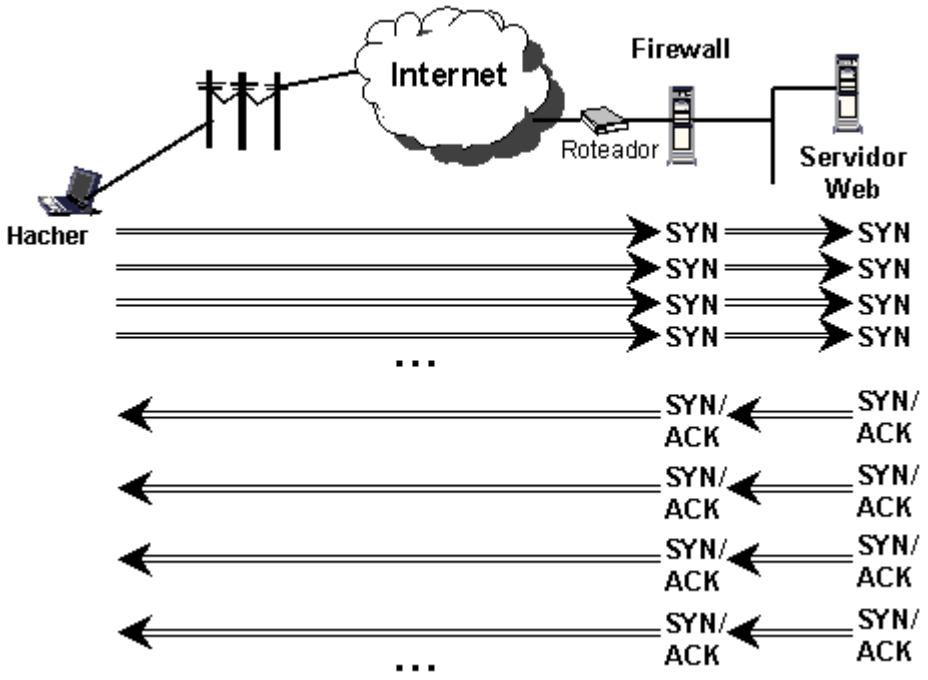


Figura 9 – Ataque de SYN Flood.

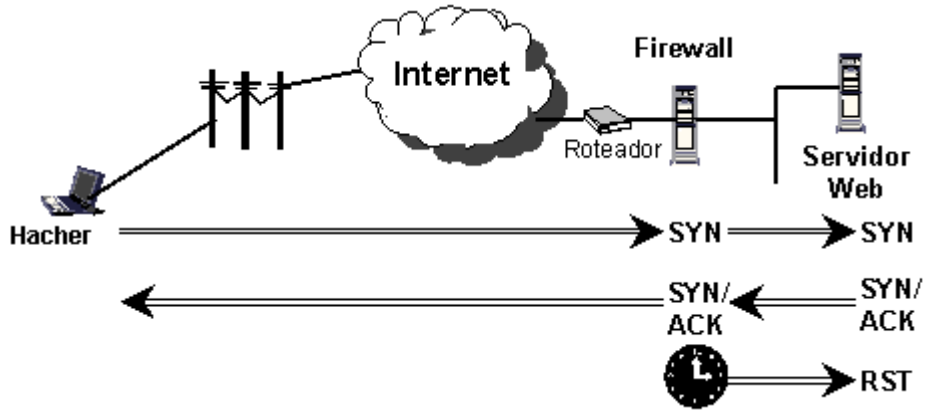


Figura 10 – Firewall interceptando um ataque de SYN Flood.

Em sistemas mais fortes de segurança, outra solução de contorno está em utilizar o gateway, no caso o firewall, como escudo, ou seja, o firewall será responsável por manter sozinho a tabela de pedidos de conexão e, se receber uma resposta positiva do usuário, estabelecerá a conexão com o servidor e seguirá normalmente. Caso contrário, descartará os pedidos automaticamente. Veja a Figura 11.

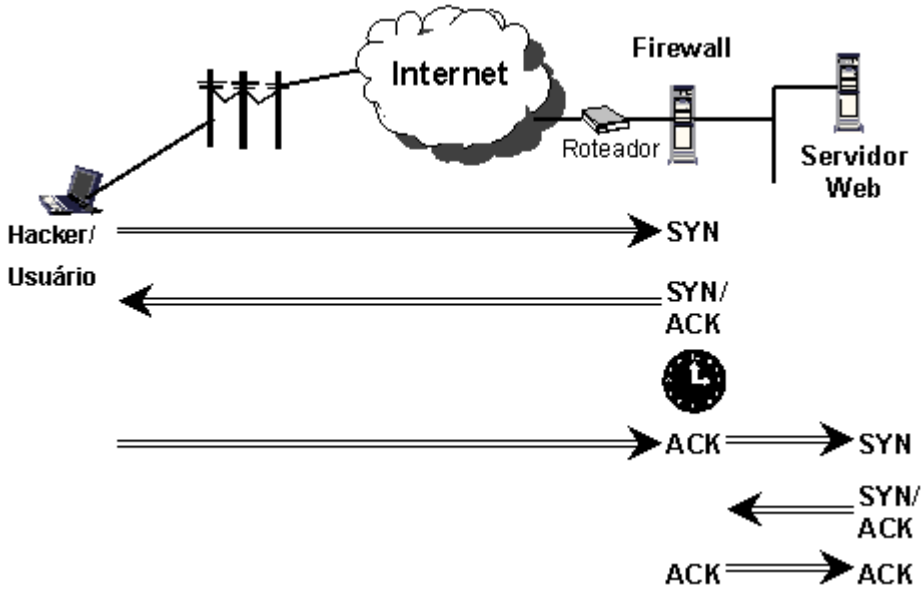


Figura 11 – Firewall como escudo da rede.

Na Figura 11, o firewall recebe o pedido de conexão, mas não o repassa ao servidor, ou seja, o mantém em uma tabela à parte e responde ao usuário como se fosse o servidor. Se for um pedido feito por um usuário autêntico, este enviará um ACK e, nesse momento, o firewall enviará o SYN para o servidor. Depois de receber o SYN/ACK dele, enviará o ACK recebido pelo usuário e fechará a conexão. Se passado um tempo configurado o firewall não receber o ACK do usuário, descartará o pedido.

Outra saída para minimizar o problema, mas não muito utilizada ultimamente, é fechar a conexão com o servidor como se estivesse tudo bem, mas controlar o tempo de resposta do usuário. Se o usuário não responder após um intervalo de tempo, deve-se fechar a conexão com o servidor. Tal técnica pode ser observada na Figura 12.

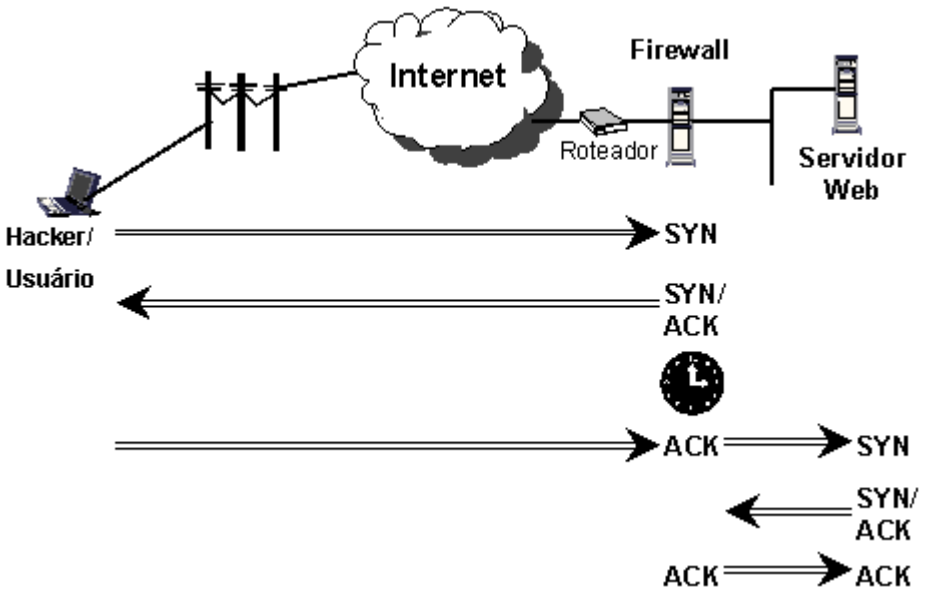


Figura 12 – Firewall como escudo da rede.

Na situação que estamos tratando, o firewall permite que o ataque chegue ao servidor, mas controla o tempo máximo de espera por um pacote ACK. Se o pacote não chegar do usuário durante o tempo máximo configurado, o firewall se encarrega de fechar a conexão enviando um pacote RST para o servidor. Se, por outro lado, o usuário enviar o ACK, o firewall não precisará fazer mais nada, porque a conexão já foi estabelecida no passo anterior. Essa técnica não é muito

empregada atualmente, porque permite que o ataque chegue ao servidor, o que pode ser um risco à rede da empresa.

Outra opção a considerar: cotejar um limite de tempo *versus* quantidade de pedidos de conexão e tentar, com isso, “perceber” que estamos diante de um ataque SYN Flood.

Por exemplo, se durante um segundo forem recebidos mais de 1.000 pedidos de conexão, talvez estejamos diante de um ataque. Mas, cuidado, há sites em que o volume de pedidos é tão elevado que esse parâmetro representa um número pequeno.

Se temos essa sensibilidade e percebemos o ataque, anulamos todos os pedidos do endereço de origem, mas precisamos ser extremamente rápidos. Como estamos enfrentando um ataque maciço, temos que ser pró-ativos o suficiente para cancelá-lo, ou talvez todos os pedidos, e prover algum tipo de sinalização para o administrador de segurança agir e restaurar a normalidade.

A decisão de negar um ou todos os pedidos é difícil: se negarmos todos, estamos diante de realizar internamente o próprio ataque de negação de serviço, ou seja, estamos sendo “forçados” a colocar o site inteiro negando pedidos e, conseqüentemente, “saindo do ar”. Podemos dizer que é conceitualmente impossível determinar uma solução definitiva para esse problema. O Kernel do Linux já possui uma proteção contra este tipo de ataque, o TCP syncookie support. Veja em http://cr.yp.to/syncookies.html.

Essencialmente, um ataque DoS interrompe ou nega completamente serviço a usuários legítimos, congestionando redes, sistemas ou outros recursos. O pior ataque DoS é, sem dúvida, o que consome largura de banda de rede.

É muito comum uma guerra de flood entre desafetos em canais de chat, fazendo com que um determinado usuário seja desconectado por não responder ao servidor de IRC. A razão é que este último está ocupado tentando tratar os milhares de pacotes que chegam até ele. Existindo uma disputa, levará vantagem quem conseguir enviar o maior número de pacotes para seu oponente.

Pode fazer parte do ataque uma invasão a um site interligado a um link veloz que permita enviar grande quantidade de pacotes ao inimigo.

Em ataques a servidores, exploram-se vulnerabilidade como *Ping of Death*, variações de SYN flood e, ultimamente, um ataque bem difícil de ser rastreado, o Smurf Attack. Porém, têm-se visto ataques mais simples com as fontes facilmente localizáveis na Internet, como Land e Teardrop e suas variações.

O atacante é capaz de tornar inoperante a conexão de rede da vítima simplesmente porque possui mais largura de banda disponível. Um cenário possível é o uso de uma conexão E1 (1,544 Mbps) para tornar inoperante um acesso de rede de cliente de 56 Kbps ou 128 Kbps.

Texto 56 – Distributed Denial of Service (DDoS)

Uma evolução do ataque DoS é o ataque **DDoS** (ataque DoS distribuído), em que o atacante amplifica seu ataque DoS, engajando máquinas de múltiplas redes para tornar inoperante a rede da vítima.

O princípio é o mesmo do DoS, mas o ataque parte de centenas ou milhares de servidores, previamente invadidos e sob seu controle, simultaneamente para um único servidor. Trata-se de uma variante do DoS ainda mais danosa porque o atacante não é único, ou seja, a opção de filtro nos roteadores para uma origem específica desaparece, só restando a opção de filtro para um conjunto de endereços de origem. Mas, e se o conjunto for muito grande? Infelizmente não há muito a ser feito, o ataque de fato é bem-sucedido, o servidor pára e precisa ser reinicializado.

Esses ataques de negação de serviço custam às empresas milhões de dólares a cada ano, sendo uma ameaça séria a qualquer sistema ou rede. Os custos estão relacionados a tempo de sistema fora do ar, perda de receita e trabalho físico envolvido em identificar e reagir a tais ataques. Como exemplos de ataques DoS, citamos: Smurf, fraggle, SYN floods etc.

Existem diversas maneiras de obter acesso não autorizado a uma determinada máquina. Isso fica facilitado se o atacante já possuir um bom conhecimento da estrutura interna da empresa (footprinting).

Texto 57 – Ataques de força bruta

É a forma de ataque mais básica. Consiste em adivinhar uma combinação de ID de usuário e senha pelo método de tentativa e erro (força bruta). Se o atacante dispuser de tempo suficiente, descobre qualquer par ID de usuário/senha por

força bruta.

Os sistemas de segurança procuram tornar esse tempo suficientemente longo, centenas ou milhares de anos, de modo a tornar esse tipo de ataque ineficaz. Entretanto, ao escolherem senhas fáceis, usuários descuidados estarão contribuindo para a redução desse tempo de descoberta a apenas alguns dias, tornando o ataque viável.

Inúmeras ferramentas de ataque por força bruta utilizam dicionários de senhas comuns, nomes de pessoas, objetos e produtos em diferentes línguas etc. Tais ferramentas também são capazes de realizar tentativas com base em informações pessoais do usuário. Por exemplo, se o usuário possui o nome João Paulo, senhas do tipo Jpaulo, OaojoluaP, JP1965 são testadas, além de outros artifícios. Como exemplo de serviços sujeitos a esses ataques, citamos: Telnet, FTP, SSH, POP.

Vale ressaltar, considerando-se o que foi dissemos anteriormente, a importância de utilizar sistemas de segurança que obriguem os usuários a utilizar senhas complexas (combinação aleatória de letras e números) e a trocarem as senhas periodicamente. E, além disso, conscientizar os usuários de sua responsabilidade pela segurança da empresa, pois nenhum sistema de segurança é capaz de prever todos os comportamentos possíveis de seus usuários.

Texto 58 – Man-in-the-middle attack

Também conhecido como ataque ao nó intermediário: o atacante entra no meio da comunicação com o objetivo de descobrir as informações que estão trafegando. Funciona como se tivéssemos uma extensão telefônica e alguém a pegasse no meio de uma conversa privada e entendesse o diálogo.

Com a utilização da criptografia, tentamos garantir a privacidade na comunicação, porém, com ela, surgiu a criptoanálise: a ciência de descobrir um texto original sem saber qual é a chave de criptografia.

O objetivo da criptoanálise é obter, de um texto criptografado, o texto original ou até mesmo a chave usada para criptografar. Essas tentativas exploram as fraquezas matemáticas dos algoritmos e o modo de operação em uso ou do protocolo de comunicação.

O controle, por um atacante, de algum nó intermediário no percurso entre o emissor e o receptor é estrategicamente muito interessante. Se o atacante for capaz apenas de grampear o meio de comunicação, isto é, um **ataque passivo**, em que nenhuma das entidades percebe o "grampo", todos os dados criptografados por um determinado algoritmo estarão disponíveis para o processo de criptoanálise.

Isso aumenta significativamente a probabilidade da criptoanálise ser bem-sucedida. No caso, o diferencial é a chave utilizada no algoritmo empregado para criptografar os dados, ou seja, quanto mais resistente for o tempo para descobrir a chave, maiores serão as chances do ataque ser anulado.

Outro fator a considerar é ocultar o algoritmo empregado, isto é, se o atacante perceber que possui dados criptografados pelo algoritmo DES, saberá que a chave é simétrica, logo, única nos dois sentidos da comunicação. É recomendável utilizar protocolos que não forneçam informações sobre o algoritmo empregado, como o *Encapsulation Security Payload*, em que se utiliza um número inteiro N que representa a associação de segurança entre duas entidades. A informação sobre o algoritmo empregado está dentro dessa associação, que fica oculta durante a transmissão dos dados entre as entidades.

Se o atacante tiver a habilidade de interceptar e transmitir mensagens modificadas no meio de comunicação, um **ataque ativo**, poderá assumir a identidade de uma entidade, fazendo-se passar pelo emissor ou receptor, personificando uma das extremidades (entidade) para a outra.

Outro ataque possível: combinar duas mensagens criptografadas com a mesma chave, para produzir uma terceira mensagem de significado distinto. Trata-se de técnica conhecida como ataque de recorte e colagem.

Finalmente, ataques baseados na repetição de mensagens são também mais facilmente implementados por atacantes que disponham do posicionamento estratégico como descrito anteriormente. Esse tipo de ataque se confunde com ataques de negação de serviço, cujo objetivo é anular as atividades de uma entidade pela quantidade excessiva de dados que lhe são enviados.

Texto 59 – Birthday attack em funções de hash

Este ataque é baseado em um problema probabilístico clássico, conhecido como paradoxo da data de aniversário. A questão é simples: quantas pessoas são necessárias em uma sala, para que a probabilidade de, pelo menos uma, dentre 50% das pessoas tenha a mesma data de nascimento? A resposta é 183 pessoas.

Por outro lado, se a pergunta for quantas pessoas em média são necessárias em uma sala, para que pelo menos duas possuam a mesma data de nascimento, a resposta será um valor bem menor: somente 23.

Analogamente, dada a saída de uma função de hashing de n-bits de comprimento, são necessárias em média 2^{n-1} mensagens aleatórias para produzir o mesmo resultado. Dessa forma, encontrar duas mensagens que produzam a mesma saída de uma função de hash requer somente $2^{n/2}$ mensagens aleatórias em média. Para evitar a viabilização desse tipo de ataque, em função do poder computacional atual, as funções de hashing devem ser projetadas para produzir valores de saída com tamanho superior a 128 bits.

Mais uma vez estamos diante do paradigma atual, que fundamenta a maioria das técnicas de segurança. Partimos do princípio de que é quase, se não totalmente impossível, num curto espaço de tempo escolher 2^{64} ($2^{128/2}$), ou 18.446.744.073.709.551.616, mensagens aleatórias para produzir o mesmo resultado e, conseqüentemente, descobrir a combinação correta.

Texto 60 – Spoofing e sequence number attack

Vamos recordar como é feito o estabelecimento de uma conexão entre dois usuários ou hosts. Para tal, a Figura 13 irá auxiliar a análise.

Quando o usuário A envia um pacote SYN ao usuário B, o pacote carrega também o número de seqüência do pacote (PSN) que será usado por ele nesta conexão.

O pacote SYN/ACK correspondente, transmitido pelo usuário B, transporta o PSN usado por ele nessa nova conexão e reconhece o número de seqüência usado pelo usuário A.

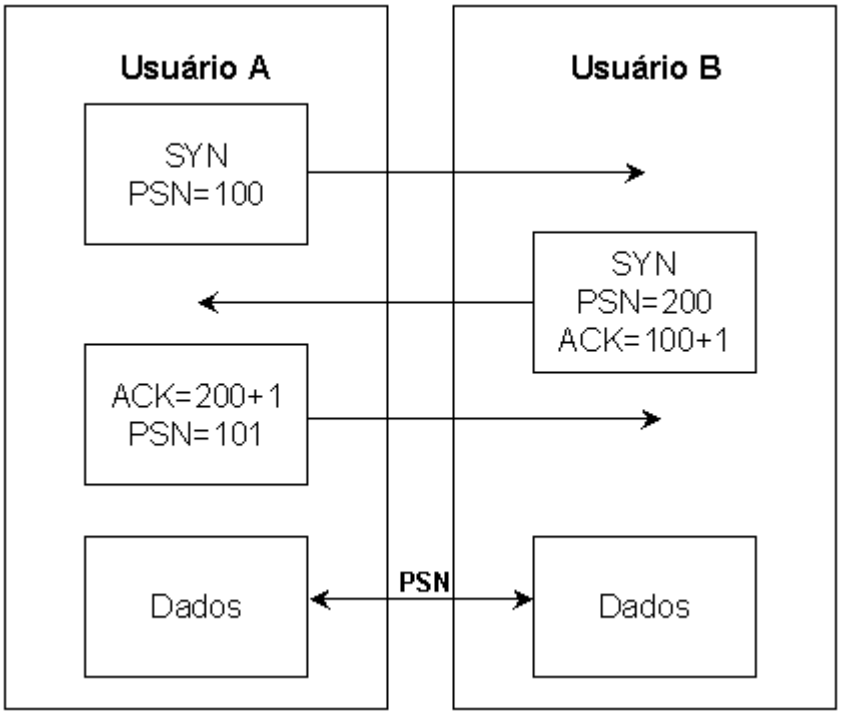


Figura 13 – Estabelecimento de conexão entre usuários.

Finalmente, a conexão é estabelecida quando o usuário A retorna ao B um outro pacote ACK, reconhecendo o número de seqüência do pacote deste último. Com a conexão estabelecida, os números de seqüência são usados para garantir a ordem de transferência dos dados para aplicação.

O spoofing é um tipo de ataque de rede que pode ser direcionado contra qualquer servidor ou equipamento que esteja conectado. Tem por objetivo estabelecer uma conexão entre um usuário desconhecido, fazendo-se passar por um outro legítimo, com um servidor e, uma vez conectado à rede, explorar essa relação de confiança para ganhar acessos não autorizados à rede.

Como as implementações de TCP de alguns sistemas operacionais usam algoritmos simples para geração desses números, permitem que atacantes facilmente descubram qual será o próximo número da seqüência utilizado por uma máquina.

A Figura 14 vai nos ajudar a entender o objetivo do atacante.

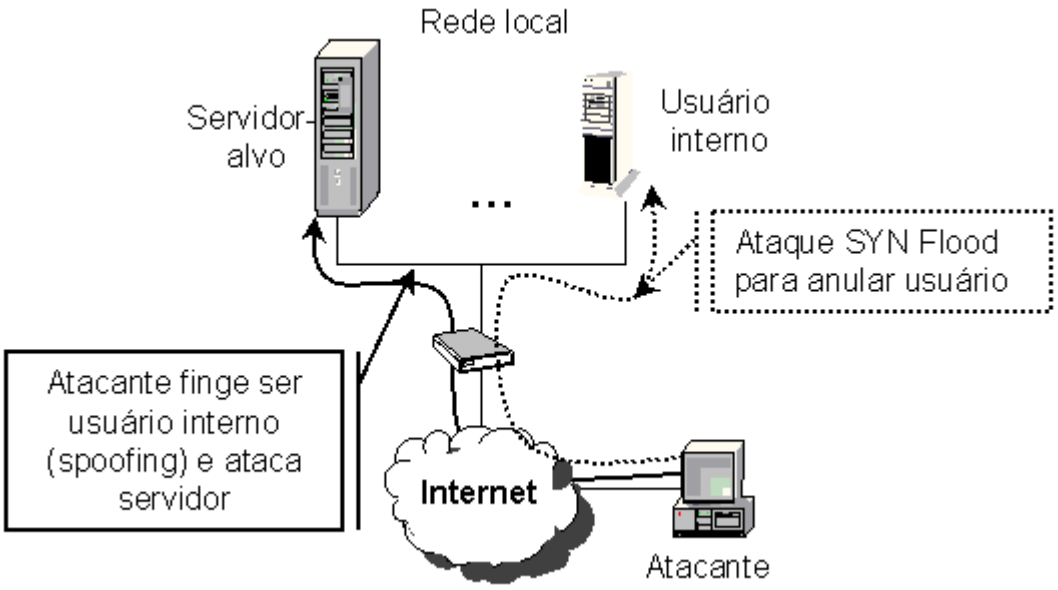


Figura 14 – Ataque Spoofing com Sequence Number Attack.

O primeiro passo do atacante é anular o cliente legítimo, ou seja, gerar algum tipo de SYN flood no usuário real para que ele não seja capaz de responder aos pedidos do servidor. Isso é necessário pois quem irá responder aos pedidos será o atacante e não o usuário real. O usuário, recebendo um ataque SYN flood, não será capaz de participar da negociação por um período de tempo, mas o atacante irá se passar por ele para obter acesso à rede.

Para se passar por um usuário legítimo, o atacante irá utilizar um *Sequence Number Attack* combinado com o *spoofing*. Vamos analisar esse caso visualizando a Figura 15.

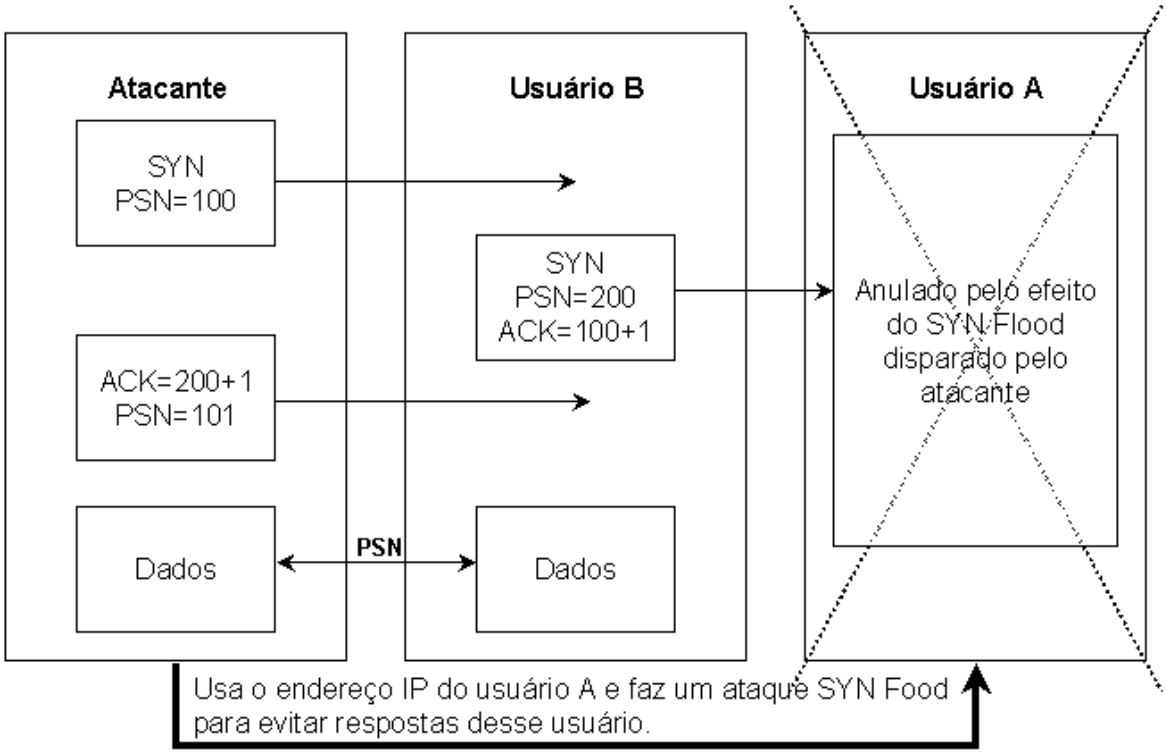


Figura 15 – Ataque de número de seqüência.

Depois do primeiro passo, ou seja, com o usuário legítimo “congelado” por um período de tempo, em conseqüência do SYN Flood, é possível começar o ataque. Ele solicita um pedido de conexão, tipicamente enviando um SYN para iniciar a comunicação, porém com o endereço IP do usuário real para o usuário B. Como o endereço IP do atacante indica um usuário autêntico, ele pode prosseguir com o pedido. O servidor envia um SYN, AKC e um número de seqüência gerado aleatoriamente para o usuário A e para o usuário B, chamado de PSN (*Packet Sequence Number*).

A resposta do usuário B é enviada para o usuário A legítimo, porém o servidor ainda não sabe que, na verdade, o usuário A não está respondendo porque está sob ataque de SYN Flood.

Agora, há o fator surpresa e o atacante não se preocupa com o retorno do servidor, porque sabe que o pacote não irá chegar a ele. Por outro lado, o que importa daqui em diante é o PSN contido dentro do pacote. O atacante envia um SYN, PSN dele e o PSN do usuário B. Em teoria, o endereço IP não precisa mais

ser validado, uma vez que foi verificado no passo anterior, dessa forma, que o usuário B espera que os PSNs estejam corretos.

Descobrir o PSN do usuário legítimo é uma tarefa simples, porque se trata de um número aleatório e seqüencial. Logo, antes de iniciar um ataque spoofing, o atacante executa alguma ferramenta de scan para capturar os pacotes enviados pelo servidor a fim de descobrir qual o PSN que ele está usando e, com esse valor, somar um ou dois deles pode ser uma boa opção para um ataque.

A maioria dos atacantes é paciente e escolhe os melhores momentos para uma investida. Se escolher o valor correto de ambos os PSNs, poderá estabelecer uma conexão entre eles. Feito isso, pode explorar essa relação para obter acesso ao servidor. Os sistemas UNIX e Linux, por exemplo, permitem login remoto (rlogin), no qual, com um acesso já estabelecido, faz-se login em outro servidor, sem que seja necessário fornecer uma senha de acesso, porque a fase inicial de conexão já validou o usuário.

Como as implementações de TCP de alguns sistemas operacionais usam algoritmos simples para geração desses números, permitem que atacantes facilmente descubram qual será o próximo número da seqüência utilizado por uma máquina.

Apesar de previsto por Steve Bellovin em 1989, esse tipo de problema começou a ser mais conhecido com o suposto ataque de Kevin Mitnick aos computadores de Tsutomu Shimomura, em que Mitnick descobriu os números de seqüência e conseguiu entrar na rede. Entretanto, algumas das afirmações contidas na Internet nunca foram comprovadas totalmente. Dessa forma, recomendamos um senso crítico ao ler o material disponibilizado [TKD], que possui, inclusive, a mensagem enviada por Tsutomu Shimomura para o grupo Usenet News em 25/Jan/1995.

Durante vários anos, o ataque spoofing foi um dos mais freqüentes feitos pela Internet. Embora não seja tão utilizado hoje em dia, continua sendo um dos mais perigosos e deve ser constantemente prevenido, principalmente em virtude da grande quantidade de ferramentas disponíveis na Internet que têm por objetivo auxiliar nas tentativas de descobrir o PSN e fazer ataques SYN flood.

Os sistemas mais vulneráveis a esse ataque eram o UNIX, baseado na distribuição de Berkeley (BSD), e Windows NT. A solução no sistema operacional foi modificar os programas para que gerassem números de PSN baseados na combinação do endereço IP do usuário, mais um número aleatório que devesse

ser validado pelo usuário real e o servidor logo na segunda fase de negociação da conexão.

Para administradores de segurança, a solução é garantir, sempre que possível, que as premissas sejam seguidas.

- Firewalls e roteadores devem ser configurados para rejeitar qualquer tipo de IP spoofing, ou seja, endereços IPs da rede interna circulando pela Internet, endereços inválidos ou endereços fora de um padrão bem conhecido. Por outro lado, as tentativas devem ser sinalizadas por algum tipo de alarme para análise e possível bloqueio do atacante.
- Dentro da rede interna, somente os servidores necessários devem estabelecer uma relação de confiança, como é o caso típico do arquivo /etc/hosts.equiv, e deve-se modificar corretamente os atributos dos arquivos dentro do servidor, permitindo ou não o acesso para escrita a certos diretórios.
- Somente servidores necessários devem ser configurados dentro do arquivo /.rhost para liberar o acesso a login remoto ou, se possível, remover totalmente essa possibilidade.

Texto 61 – Smurf

O ataque smurf é o mais conhecido por se beneficiar do sistema de endereços broadcast de uma rede. Baseia-se em negação de serviço, ou seja, também constitui um Denial of Service. Seu funcionamento é simples, porém o resultado pode ser bem impressionante, pois não tem intenção de parar um computador, mas, sim, uma rede inteira.

É realizado ao se enviar contínuas streams (cadeias) de pacote ICMP modificadas para a rede-alvo. A Figura 16 ilustra um ataque smurf.

Seu funcionamento apresenta as seguintes características:

- o atacante se conecta na Internet e procura uma vítima. Estuda suas fragilidades e descobre seu endereço IP;
- o atacante procura outros usuários ou servidores na mesma rede da vítima. Ele pode criar uma lista de endereços ou endereços de broadcast (vários IPs) de rede;

- ele envia pacotes do tipo ICMP Echo Request “spoofados”, ou seja, o endereço de origem não é o do atacante, e, sim, o da vítima, para cada um desses endereços;
- as máquinas das redes que receberam os pacotes respondem com pacotes do tipo ICMP Echo Replay, mandando todos os pacotes para a vítima e, não, para o computador do atacante.

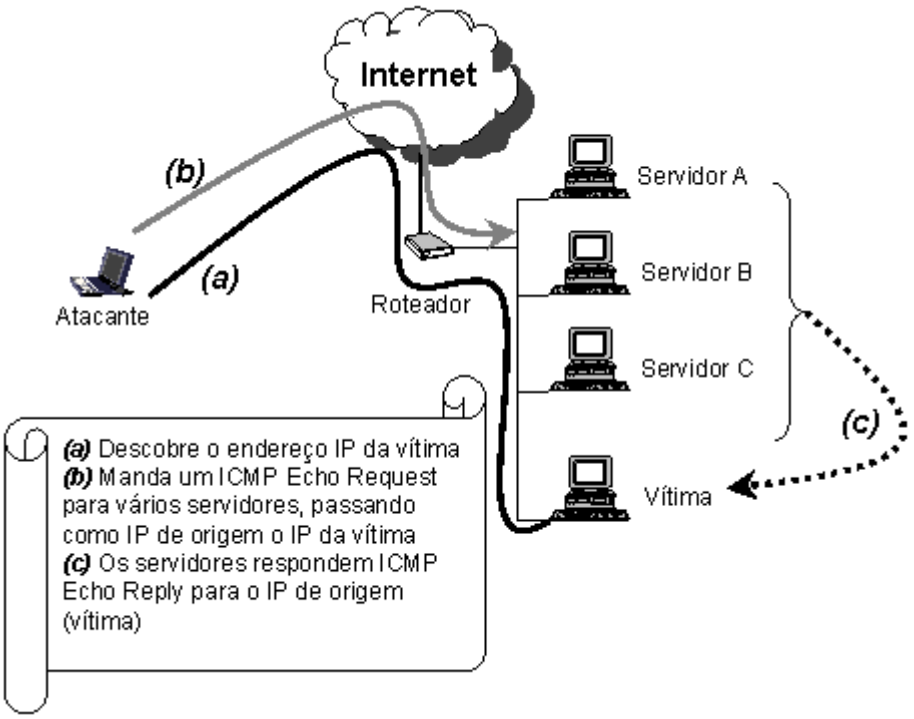


Figura 16 – Ataque smurf.

Dependendo das quantidades de endereços utilizados e das de máquinas dentro das redes, o ataque pode ser realmente devastador.

Logo após a descoberta do smurf, começaram a surgir variações, como o smurf baseado em UDP, que, em vez de enviar pacotes de ICMP ECHO_RESPONSE, envia pacotes UDP “spoofados” para os endereços de broadcast. As máquinas da rede, ao receberem o pacote UDP, respondem à vítima.

A vítima, em certas circunstâncias, envia mais uma resposta, gerando grande quantidade de tráfego na rede. Tal tipo de ataque tem sido usado para parar vários provedores de acesso à Internet e todos os seus usuários. Efetivamente, é muito difícil proteger-se de um ataque de broadcast bem planejado. Se a pessoa que está atacando conseguiu elaborar uma lista de endereços de broadcast vulneráveis e se a vítima não possui a proteção de um firewall, sua conexão certamente ficará congestionada.

Em muitos casos, firewalls também são ineficientes. Os administradores de rede podem evitar que suas redes sejam utilizadas como amplificadores. Instalar firewalls que impeçam a entrada de pacotes ICMP/UDP, cujo destino seja o endereço de broadcast, é uma saída. É possível, ainda, tomar precauções simples, como desativar serviços que não estão sendo utilizados pelos computadores da rede.

Texto 62 – Session hijacking

Num ataque de seqüestro de sessão, o atacante procura uma conexão TCP já existente entre dois equipamentos e tenta controlar a sessão. É fundamental, também, que o ataque do tipo *Man-in-the-Middle* esteja estrategicamente localizado, inspecionando todo o fluxo de dados da comunicação.

Nessa posição estratégica, ou seja, no meio da comunicação, o atacante acompanha e controla toda a sessão, com possibilidade de dessincronizar ou negar serviços a um dos participantes. O atacante pode, por exemplo, enviar um reset para uma parte e assumir por completo a sessão.

A proteção é extremamente difícil. Até mesmo mecanismos de autenticação mais rígidos nem sempre têm êxito na tentativa de evitar tais ataques, porque os participantes já foram autenticados no início do estabelecimento da sessão.

O uso de um bom esquema de autenticação, sensível o suficiente quanto a mudanças de localização de origem, e canais seguros combinados, pode ser uma boa solução para evitar tais ataques.

Texto 63 – Exploits

Técnica utilizada para explorar vulnerabilidade de sistemas operacionais e software. Esse tipo de ataque aumentou bastante depois de outubro de 1996, quando foi publicado pelo e-Zine PHRACK um artigo intitulado *Smashing the stack for fun and profit*, detalhando como funciona e como corromper uma pilha de um programa. Infelizmente, existem ferramentas que podem apontar falhas de segurança em sistemas operacionais e software.

Ultimamente, os exploits remotos mais populares são statd (rcp.statd), imap, inn, routd, qpop3 e bind. Logo, é extremamente aconselhável instalar todas as correções (patches) fornecidas pelo fabricante. Porém, mesmo quando se adota essa política, a realidade mostra que a velocidade dos fabricantes em prover correções é menor que a dos hackers de descobrir novos furos de segurança (bugs).

Texto 64 – Trojans horses (cavalos de Tróia)

O termo vem da *Ilíada*, de Homero, em que se relata que os gregos dão um enorme cavalo de madeira a seus inimigos, os troianos, como sinal de que estavam desistindo da guerra. O povo de Tróia leva o "presente" para sua cidade. Soldados gregos, que esperaram a noite dentro do cavalo, saem e abrem os portões para seus compatriotas, o que permite capturar e dominar a cidade. A história também gerou o conceito do "presente de grego", definição que se encaixa nos trojans da era digital.

Assim, um trojan é um programa que oculta seu objetivo sob a camuflagem de um outro programa útil ou inofensivo. É um programa aparentemente apto a executar uma tarefa, que realiza ou não, mas, de fato, executa outra ação, e esta pode danificar seriamente o computador, como apagar informações ou capturar senhas, contas de e-mail etc.

Sua propagação acontece normalmente pela Internet, onde são colocados e oferecidos como programas úteis ou anexados a algum tipo de piada ou brincadeira. São, assim, voluntariamente copiados por usuários incautos, enganados quanto a seu reais efeitos.

Um exemplo ocorreu em 1995, quando alguém começou a distribuir uma atualização do programa Pkzip, uma ferramenta de compactação de arquivos. Como a versão original do Pkzip era instalada a partir do próprio executável, por meio de um duplo clique, essa atualização (um trojan) também possuía essa característica, porém, quando isso era feito, o trojan executava comandos DOS *deltree* e *format*. Esses comandos eram utilizados para deletar arquivos e pastas a partir do diretório principal e formatar o disco rígido do usuário.

A única forma de prevenção era a substituição dos comandos por outros, levando a atualização maldosa a sinalizar erro e não seguir o caminho destrutivo. Mas, até que essa informação fosse divulgada, vários usuários perderam informações preciosas e muito tempo foi gasto, gerando prejuízos para muitas pessoas e empresas.

Outro caso que alarmou a Internet ocorreu em janeiro de 1999, com o nome de happy99, com capacidade de autodistribuição. Esse trojan vinha como um anexo infectado e, quando executado, o programa mostrava uma bonita janela com fogos de artifício. Nesse momento, ele instalava no sistema, sem permissão do usuário, dois arquivos (SKA.EXE e SKA.DLL) que alteravam o programa WSOCK32.DLL. A modificação continha rotinas para detectar a lista de endereços de e-mail do usuário e passava a enviar uma cópia do arquivo SKA.EXE, renomeada como happy99.exe, para cada destinatário. Esse trojan não tinha ação destrutiva conhecida.

Como os trojans não se limitam às características dos vírus, são potencialmente mais perigosos, arquivos com extensão exe, bat, com etc., desconhecidos ou de origem duvidosa, mesmo que passem pelo antivírus, só devem ser executados com cautela. O ideal é evitar ao máximo abrir qualquer arquivo recebido pela Internet, mesmo que seja de alguém conhecido ou da própria família, porque nunca se tem certeza absoluta da origem.

Se a política de segurança da empresa permite que sejam instalados programas freeware, prefira sempre buscar o utilitário desejado direto de quem desenvolveu a ferramenta e nunca o aceite por e-mail ou outra fonte insegura. É importante seguir a política de segurança da empresa, mas também observar as seguintes medidas de prevenção:

- jamais executar um programa ou abrir um arquivo sem antes executar o antivírus sobre a pasta que o contenha;
- atualizar o antivírus constantemente;

Diariamente as empresas distribuem cópias gratuitas dos arquivos que atualizam a lista dos vírus e vacinas. e os programas podem ser configurados para verificar a existência de novas assinaturas mais de uma vez ao dia. Então, para usuários conectados durante tempo integral, o melhor é configurar o antivírus para fazer atualizações automáticas, sendo alertado pelo fornecedor sobre novas atualizações. É necessário acompanhar o log dessas atualizações para ter certeza de que foram instaladas com sucesso e fazer uma verificação completa após a atualização.

- desativar a opção de executar documentos diretamente do programa de correio eletrônico;
- jamais executar programas que não tenham sido obtidos de fontes absolutamente confiáveis.

Unidade 5 – Melhorando a configuração no Windows XP Professional

Nesta unidade faremos, de forma bastante objetiva, algumas recomendações para aumentar a segurança do Windows XP Professional.

Proteção de tela

Habilitar a proteção de tela com senha, configurando para entrar em ação após cinco minutos de inatividade, usando o *Logon Screen Save.*

Aplicação não responde

"Em *Startup and Recovery*, configurar o tempo de espera para escolha do sistema operacional em três segundos e selecionar a opção de falha do ativo (*system failure*) para geração de *dump* de memória."

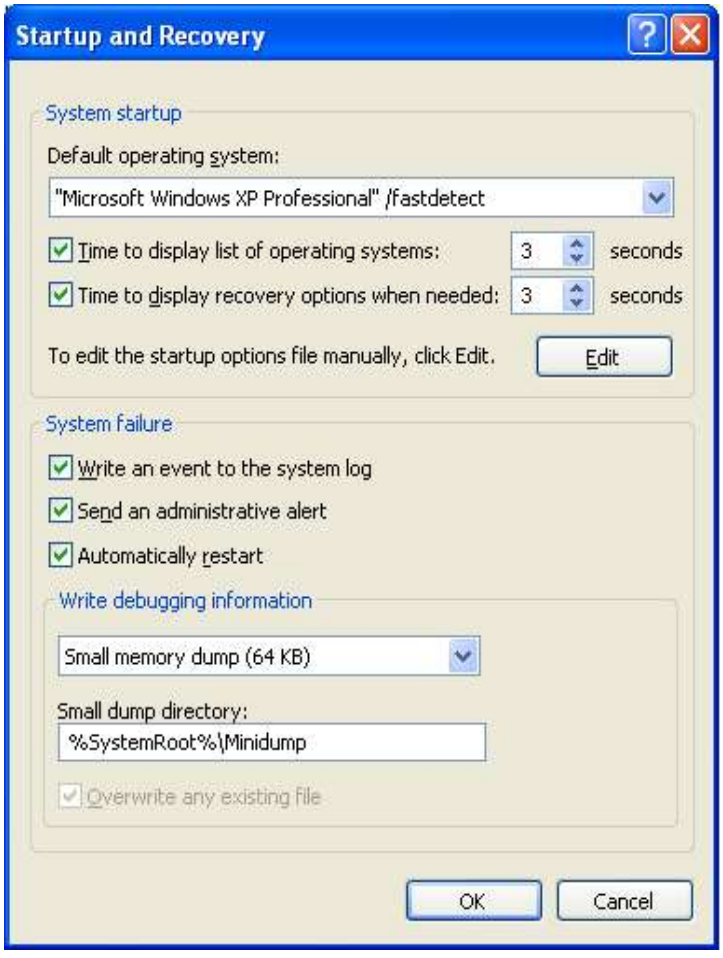


Figura 17 - Configurando o Startup and Recovery

Senhas

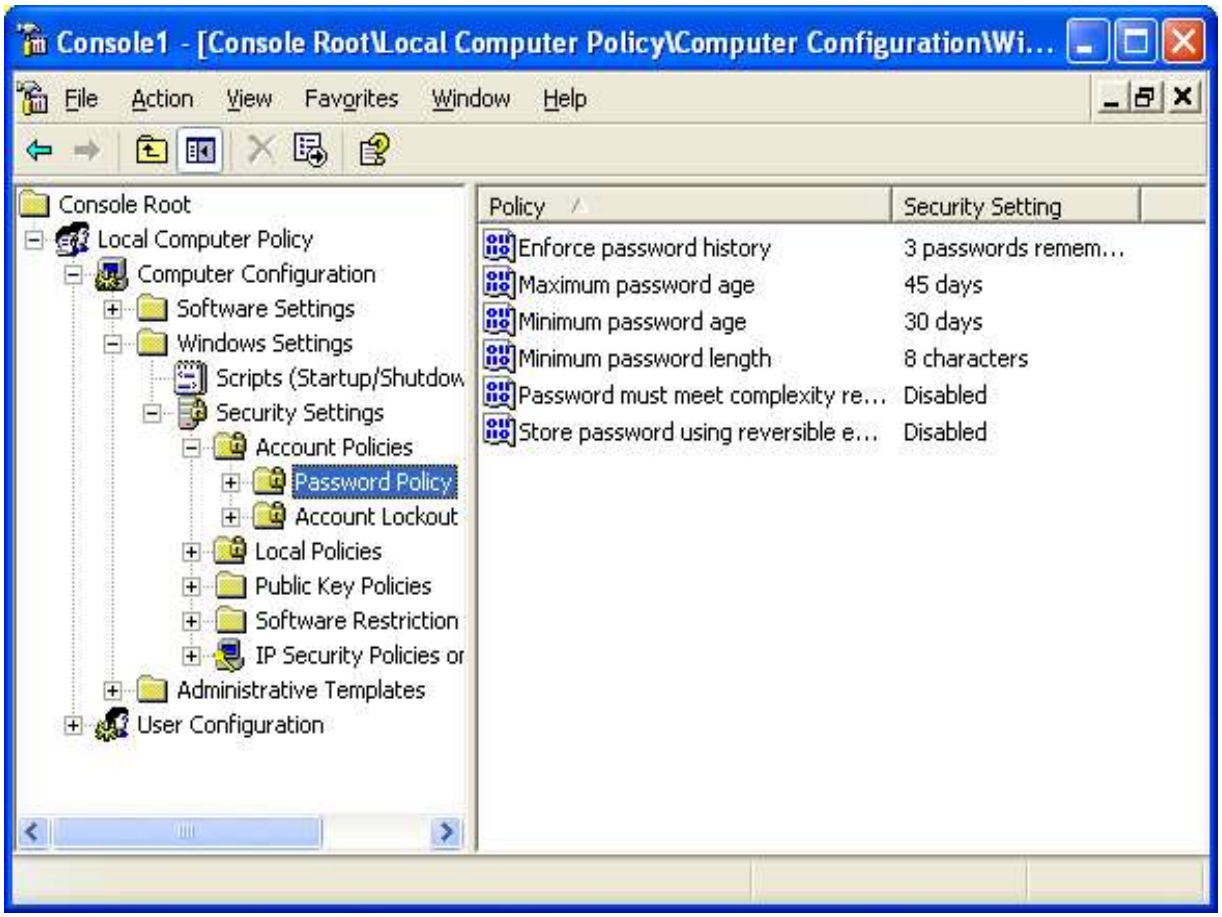


Figura 18 - Configurações da senha: validade, tamanho, repetição etc.

A Figura 18 mostra onde configurar o sistema de senha:

- quantas senhas anteriores serão lembradas para não permitir repetição delas na troca de senha;
- idade máxima da senha – quando será pedida a troca;
- idade mínima da senha –não será possível ao usuário trocar a senha por sua vontade antes do prazo;
- tamanho mínima da senha;

- obrigatoriedade da senha ser complexa – o sistema verifica se há caracteres minúsculos e maiúsculos, caracteres de pontuação etc.

Bloqueio de contas

Para evitar que ocorram testes sucessivos, na tentativa de descobrir a senha de uma conta, é possível bloquear uma conta que tenha mais do que um determinado número de tentativas inválidas de login. Uma vez bloqueado, também é configurável por quanto tempo este bloqueio estará ativo.

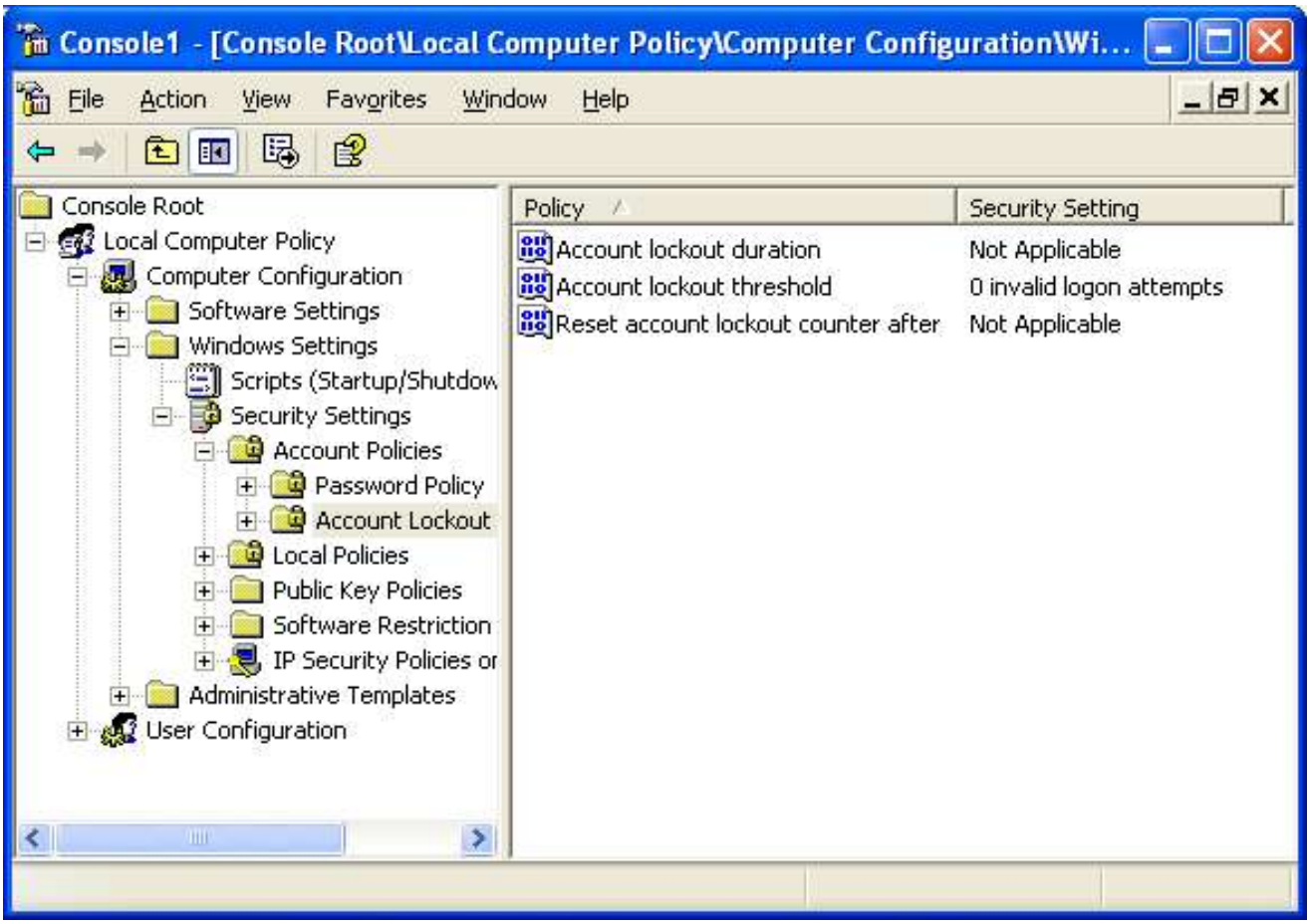


Figura 19 - Bloqueio de contas após sucessivos erros de login.

Auditoria

Poder fazer verificações de eventos que ocorreram é essencial para um administrador de sistema. Com a ativação do sistema de logs, é possível ter gravado em disco diversos eventos relacionados com o funcionamento do sistema operacional.

No entanto, o administrador deve ser cuidadoso, pois este sistema de log consome recursos da máquina, ou seja, CPU, memória e acesso a disco. Ligar os sistema de logs para tudo, certamente, irá comprometer a eficiência e desempenho do servidor.

A tela de configuração deste sistema de auditoria está na Figura 20.

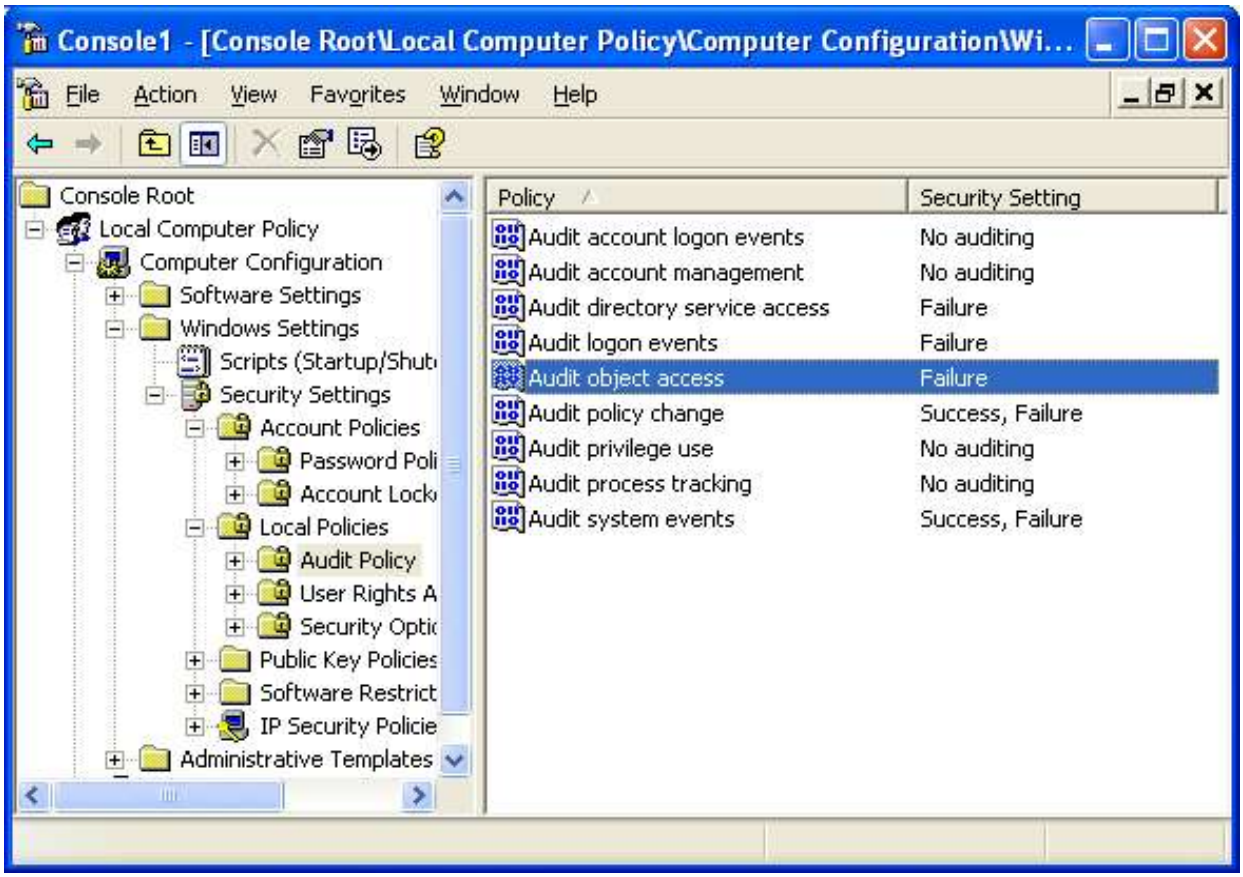


Figura 20 - Auditoria: configuração do que ficará gravado para inspeção futura.

Texto 66 – Serviços do Windows 2000/XP

Existem inúmeras vulnerabilidades associadas a serviços do Windows 2000 / XP. O administrador deve ficar atento a listas de avisos e às atualizações do sistema.

Há softwares disponíveis na Internet que exploram tais vulnerabilidades

relacionadas a serviços do SO, como:

- Distributed File System;
- Print Spooler;
- Remote Registry Service;
- Telnet.

Outros serviços devem ser desabilitados buscando um melhor desempenho do sistema, como é o caso de **Alerter; Messenger etc.**

A Tabela 5 mostra a relação de serviço e sua situação. Recomenda-se aprofundamento no assunto, estudando, nos manuais de ajuda, a funcionalidade prestada por cada serviço para, a partir deste conhecimento, tomar decisões sobre quais serviços precisam ou não ficar habilitados, em cada máquina. Tais decisões devem levar em consideração qual é o objetivo do uso da máquina.

Serviço	Status
Alerter	*Disabled*
Automatic Update	*Disable*
ClipBook	*Not defined*
COM+ Event Systems	*Not defined*
Computer Browser	*Not defined*
DHCP Client	*Not defined*
Distributed File System	*Disabled*
Distributed Link Tracking Client	*Not defined*
Distributed Link Tracking Server	*Not defined*
Distributed Transaction Coordinator	*Not defined*
DNS Client	*Not defined*
Event Log	*Not defined*
Fax Service	*Not defined*
File Replication	*Not defined*
IIS Admin Service	*Not defined*
Indexing Service	*Not defined*
Internet Connection Sharing	*Not defined*
Intersite Messaging	*Not defined*
IPSEC Policy Agent	*Disabled*
Kerberos Key Distribution Center	*Not defined*
License Logging Service	*Not defined*
Logical Disk Manager	*Not defined*
Logical Disk Manager Administrative Service	*Not defined*
Messenger	*Disable*
Net Logon	*Not defined*
Network Connections	*Not defined*

Serviço	Status
Network DDE	*Not defined*
Network DDE DSDM	*Not defined*
NT LM Security Support Provider	*Not defined*
Performance Logs and Alerts	*Not defined*
Plug and Play	*Not defined*
Print Spooler	*Disable*
Protected Storage	*Not defined*
QoS Admission Control (RSVP)	*Not defined*
Remote Access Auto Connection Manager	*Disable*
Remote Access Connection Manager	*Disabled*
Remote Procedure Call (RPC)	*Disable*
Remote Procedure Call (RPC) Locator	*Disable*
Remote Registry Service	*Disabled*
Removable Storage	*Not defined*
Routing and Remote Access	*Disable*
Run As Service	*Not defined*
Security Accounts Manager	*Not defined*
Server Provides RPC	*Not defined*
Smart Card	*Not defined*
Smart Card Helper	*Not defined*
Simple Mail Transport Protocol (SMTP)	*Disable*
SNMP Service	*Disable*
SNMP Trap Service	*Not defined*
System Event Notification	*Not defined*
Task Scheduler	*Disabled*
TCP/IP NetBIOS Helper Service	*Not defined*
Telephony	*Disabled*
Telnet	*Disable*
Terminal Services	*Not defined*
Uninterruptible Power Supply	*Not defined*
Utility Manager	*Not defined*
Windows Installer	*Disable*
Windows Management Instrumentation	*Not defined*
Windows Management Instrumentation Driver Extensions	*Not defined*
Windows Time	*Not defined*
Workstation	*Not defined*
World Wide Web Publishing Service	*Not defined*

Tabela 5 - Serviços no XP

Texto 67 – Direito de acesso de usuários

O sistema de direito de acesso permite ao administrador configurar acessos com base em usuários e/ou em grupos para algumas funções preestabelecidas. A Tabela 6 mostra os itens configuráveis.

Direito de usuários	Grupos/Usuários
Access this computer from network	SYSTEM, Administrators
Act as part of the operating system	Not defined
Add workstations to domain	Not defined
Back up files and directories	Administrators
Bypass traverse checking	Users
Change the system time	Administrators
Create a token object	Not defined
Create permanent shared objects	Not defined
Debug programs	Not defined
Deny access to this computer from the network	Not defined
Deny logon as a batch job	Not defined
Deny logon localy	Not defined
Deny logon as service	Not defined
Deny logon Locally	Not defined
Enable computer and user accounts to be trusted for delegation	Not defined
Generate security audits	Not defined
Increase quotas	Administrators
Increase scheduling priority	Administrators
Load and unload device drivers	Administrators
Lock pages in memory	Not defined
Log on as a batch job	Not defined
Log on as a service	Not defined
Log on locally	Administrators
Manage auditing and security log	Administrators
Modify firmware environment values	Administrators
Profile single process	Administrators
Profile system performance	Administrators
Remove computer from docking station	Not defined
Replace a process level token	Not defined
Restore files and directories	Administrators
Shut down the system	Users
Synchronize directory service data	Not defined
Take ownership of files or other objects	Administrators

Tabela 6 - Configuração dos direitos de acesso ao sistema

A Tabela 7 apresenta um exemplo das permissões básicas e restritivas aplicáveis a acesso a disco. Deve-se ter em mente que uma permissão é automaticamente herdada para todos os diretórios que estejam abaixo. Assim, uma permissão especificada para o raiz, vale para todo o disco. Especificações de permissões feitas para diretórios abaixo terão preferência. Assim, na Tabela 7, a segunda linha retira a permissão dada a System.

Pasta	Grupos / Usuários	Permissões
C:\	Administrators e System Authenticated Users	Full Control Read and Execute
C:\winnt\repair	Administrators (local)	Full Control

Tabela 7 - Permissões de acesso a disco

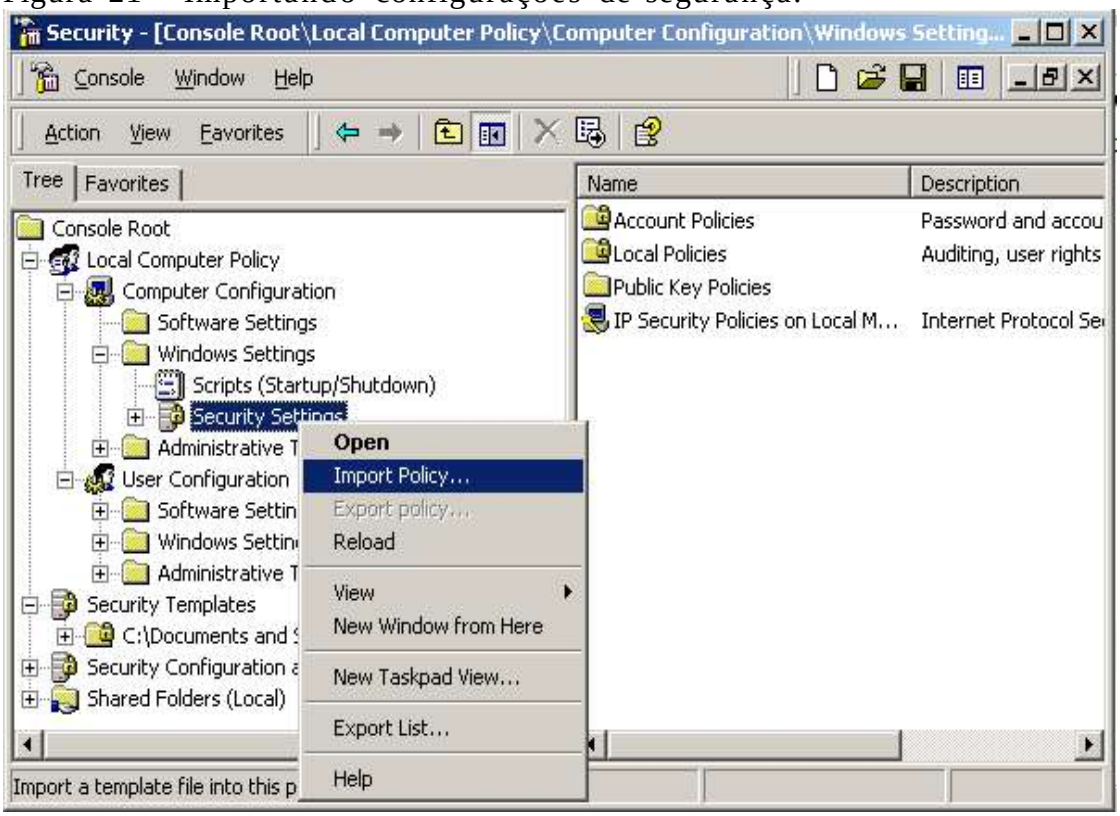
Para instalar as definições de segurança, observe os procedimentos a seguir.

Através de Iniciar \ Executar \ MMC, na barra de tarefas, clicar em console \ Add\Remove Snap-in.

Na aba Standalone, dar um Add e selecionar a opção Group Policy. Navegar - computer configuration \ Windows Settings e clicar com o botão direito do mouse em Security Settings. Escolher a opção Importy Policy e apontar para o Template configurando anteriormente.

Fechar o MMC e, quando perguntado, confirmar para salvar as configurações no antigo template configurado.

Figura 21 - Importando configurações de segurança.



Referências

ESCAMILLA, Terry.

Intrusion detection: network security beyond the firewall.

John Wiley & Sons, Inc.,1998.

GARFINKEL, Simson; SPAFFORD, Gene.

Practical UNIX & Internet security. 2. ed.

O'Reilly Media, Inc., 1996.

TIPTON, Hal; KRAUSE Micki.

Handbook of information security management.

CRC Press LLC, 1998.

STERLING, Bruce.

The hacker crackdown: law and disorder on the electronic frontier.

Macmillan Computer Publishing, 1991.

Links:
http://www.netfilter.org/documentation/
http://www.securiteam.com/unixfocus/Detecting_sniffers_on_your_network.html
http://www.securityfocus.com/infocus/1723
http://www.nessus.org
http://www.snort.org
http://www.numaboa.com.br/criptologia/
http://focalinux.cipsga.org.br/ - Gleydson Mazioli da Silva, Guia Foca Linux, ,
http://www.vle.org/presentations/20040706/001/img0.html

http://www.betterproductdesign.net/npi/products/linux.htm
http://en.uptime-project.net/page.php?page=toplist

Índice Alfabético

Aplicativo de usuário 13
Ari Lemke 9
Ataques
  Ping of Deat 129
  Script Kiddies 96
  Syn Flood 129
Autenticação 38
  criar senha 41
  Fuzzy Logic 49
  Live Scan 50
  One-Time Password por Tokens 45
  SecurID 46
  Seed 40, 46
  Single sign-on 43
  Smartcards 47
  Trusted Third-Party Authentication 39
  Two-party Authentication 39
Autenticação Forte 46
autenticador 38
BIND 104
boot BSD 79
browser 41, 44
buffer overflow 75, 122
código aberto 77
Denial of Service 124
DoS 124
exploit 77, 118
Facades 100
Footprinting 130
Free Software 10
Free Software Foundation 10
GNU 10
Gnu-Linux 10
Gnu/Linux 90
GPS - Global Position Satellite 39
grupo de usuários 35
Hackers 95
HoneyNet 100
Honeypots 99
ICMP 138
ICP 38, 43
Infra-estrutura de Chave Pública 43
Instrumented systems 100
interfaces de rede local 82
ISAPI 108
Kernel 12

Kevin Mitnick 137
Key Logger 41
Lambs 100
LILO 78
Linus Torvalds 9
Minix 9
negação de serviço 124
NetBIOS 111
Network File System, NFS 122
Nmap 73
número de seqüência do pacote 71
open source 77
Packet Sequence Number 71
Packet Sequence Number, PSN 136
permissão de acessos 35
PID 16
PIN 38, 46, 47
Process ID 16
PSN 71
rc.inet1.conf 82
rc.local 82
Richard Stallman 10
SARA 95
Server Message Block, SMB 109
shell 12
Sigle Sign-One - SSO 44
Sign-on 39
Sign-On 38
Slackware 11
smartcard 45
SNMP 124
SoftLanding Linux System 11
Software Livre 10
software obsoleto 78
software vulnerável 77
Steve Bellovin 137
SYN Flood 124
System Administration, Networking and Security 94
teclado virtual 41
tokens 45
Tokens USB 46
Tsutomu Shimomura 137
UID 35
Unicode 106
User ID 35
usuário 35
Volkerding 11
Web server folder traversal 106
“administrador do sistema 13
“jobs 32
“root 13
“sala dos servidores 32

“single user mode 32
“super usuário 13