Securing Debian Manual

Javier Fernández-Sanguino Peña <jfs@debian.org>
'Autores' on this page

v3.1, Sat, 21 Mar 2009 23:30:47 +0000

Resumo

Este documento descreve a segurança no sistema Debian. Iniciando com o processo de tornar mais seguro e fortalecer a instalação padrão da distribuição Debian GNU/Linux. Ele também cobre algumas das tarefas mais comuns para configurar um ambiente de rede seguro usando a Debian GNU/Linux, oferece informações adicionais sobre as ferramentas de segurança disponíveis e fala sobre como a segurança é fornecida na Debian pelo time de segurança
.

Nota de Copyright

Sumário

Capítulo 1

Introdução

Uma das coisas mais difíceis sobre escrever documentos relacionado a segurança é que cada caso é único. Duas coisas que deve prestar atenção são o ambiente e as necessidades de segurança de um site, máquina ou rede. Por exemplo, a segurança necessária para um usuário doméstico é completamente diferente de uma rede em um banco. Enquanto a principal preocupação que um usuário doméstico tem é confrontar o tipo de cracker script kiddie, uma rede de banco tem preocupação com ataques diretos. Adicionalmente, o banco tem que proteger os dados de seus consumidores com precisão aritmética. Em resumo, cada usuário deve considerar o trajeto entre a usabilidade e paranóia/segurança.

Note que este manual somente cobre assuntos relacionados a software. O melhor software do mundo não pode te proteger se alguém puder ter acesso físico a máquina. Você pode coloca-la sob sua mesa, ou você pode coloca-la em um cofre fechado com uma arma de frente para ela. Não obstante a computação desktop pode ser muito mais segura (do ponto de vista do software) que uma fisicamente protegida caso o desktop seja configurado adequadamente e o programa na máquina protegida esteja cheio de buracos de segurança. Obviamente, você deverá considerar ambos os casos.

Este documento apenas lhe dará uma visão do que pode aumentar em segurança no sistema Debian GNU/Linux. Se ler outros documentos relacionados a segurança em Linux, você verá que existem assuntos comuns que se cruzarão com os citados neste documento. No entanto, este documento não tentará ser a última fonte de informações que deverá estar usando, ele tentará adaptar esta mesma informação de forma que seja útil no sistema Debian GNU/Linux. Distribuições diferentes fazem coisas de forma diferente (inicialização de daemons é um exemplo); aqui, você encontrará materiais que são apropriados para os procedimentos e ferramentas da Debian.

1.1 Autores

O mantenedor atual deste documento é Javier Fernández-Sanguino Peña (mailto:jfs@debian.org). Por favor encaminhe a ele quaisquer comentários, adições e sugestões, e ele considerará a inclusão em lançamentos futuros deste manual.

Este manual foi iniciado como um *HOWTO* por Alexander Reelsen (mailto:ar@rhwd.de). Após ter sido publicado na Internet, Javier Fernández-Sanguino Peña (mailto:jfs@debian.org) o incorporou no Projeto de Documentação da Debian (http://www.debian.org/doc). Um número de pessoas tem contribuído com este manual (todos os contribuidores estão listados no changelog) mas os seguintes merecem especial menção pois fizeram contribuições significantes, seções completas, capítulos ou apêndices):

- Stefano Canepa
- Era Eriksson
- Carlo Perassi
- Alexandre Ratti
- Jaime Robles
- Yotam Rubin
- Frederic Schutz
- Pedro Zorzenon Neto
- Oohara Yuuma
- Davor Ocelic

1.2 Como obter o manual

Você poderá baixar ou ver a versão mais nova do Manual Como Tornar a Debian mais Segura no Projeto de Documentação da Debian (http://www.debian.org/doc/manuals/securing-debian-howto/). Sinta-se l livre para checar o sistema de controle de versões através do endereço servidor CVS (http://cvs.debian.org/ddp/manuals.sgml/securing-howto/?cvsroot=debian-doc).

Você poderá também baixar uma versão texto (http://www.debian.org/doc/manuals/securing-debian-howto/securing-debian-howto.ptbr.txt) do site do projeto de Documentação da Debian. Outros formatos, com o PDF, (ainda) não estão disponíveis. No entanto, você poderá baixar ou instalar o pacote harden-doc (http://packages.debian.org/harden-doc) que contém o mesmo documento em formatos HTML, txt e PDF. Note no entanto, que o pacote pode não estar completamente atualizado com o documento fornecido pela Internet (mas você sempre poderá usar o pacote fonte para construir você mesmo uma versão atualizada).

1.3 Notas de organização/Retorno

Agora a parte oficial. No momento, eu (Alexandre Reelsen) escrevi a maioria dos parágrafos deste manual, mas em minha opinião este não deve ser o caso. Eu cresci e vivi com software livre, ele é parte do meu dia a dia e eu acho que do seu também. Eu encorajo a qualquer um para me enviar retorno, dicas, adições ou qualquer outra sugestão que possa ter.

Se achar que pode manter melhor uma certa seção ou parágrafo, então escreva um documento ao maintainer (mantenedor) e você será bem vindo a faze-lo. Especialmente se você encontrar uma seção marcada como FIXME, que significa que os autores não tem tempo ainda ou precisam de conhecimento sobre o tópico, envie um e-mail para eles imediatamente.

O tópico deste manual torna isto bastante claro que é importante mantê-lo atualizado, e você pode fazer sua parte. Por favor contribua.

1.4 Conhecimento necessário

A instalação do sistema Debian GNU/Linux não é muito difícil e você deverá ser capaz de instala-lo. Se você já tem algum conhecimento sobre o Linux ou outros tipo de Unix e você está um pouco familiar com a segurança básica, será fácil entender este manual, como este documento não explicará cada detalhe pequeno de características (caso contrário você terá um livro ao invés de um manual). Se não estiver familiar, no entanto, você poderá dar uma olhada em 'Esteja ciente dos problemas gerais de segurança' on page 25 para ver onde achar informações atualizadas.

1.5 Coisas que precisam ser escritas (FIXME/TODO)

Esta seção descreve todas as coisas que precisam ser corrigidas neste manual. Alguns parágrafos incluem as tags *FIXME* ou *TODO* descrevendo qual conteúdo esta faltando (ou que tipo de trabalho precisa ser feito). O propósito desta seção é descrever todas as coisas que precisam ser incluídas em um lançamento futuro do Manual, ou melhorias que precisam ser feitas (ou que são interessantes de serem adicionadas).

Se sente que pode fornecer ajuda contribuindo com a correção de conteúdo em qualquer elemento desta lista (ou anotações inline), contacte o autor principal ('Autores' on page 1

- Expanda informações de resposta a incidentes, talvez adicione algumas idéias vindas do guia de segurança da Red Hat capítulo sobre resposta a incidentes (`http://www.redhat.com/docs/manuals/linux/RHL-9-Manual/security-guide/ch-response.html`).
- Escreva sobre ferramentas de monitoramento remoto (para verificar a disponibilidade do sistema) tal como `monit`, `daemontools` e `mon`. Veja `http://linux.oreillynet.com/pub/a/linux/2002/05/09/sysadminguide.html`.

- Considere escrever uma seção sobre como fazer operações em rede com redes baseadas em sistemas Debian (com informações tal como o sistema básico, `equivs` e FAI).
- Verifique se http://www.giac.org/practical/gsec/Chris_Koutras_GSEC.pdf tem informações relevantes ainda não cobertas aqui.
- Adicione informações sobre como configurar um notebook com a Debian http://www.giac.org/practical/gcux/Stephanie_Thomas_GCUX.pdf
- Adicione informações sobre como fazer um firewall usando o sistema Debian GNU/Linux. A seção relacionada a firewall é orientada atualmente sobre um sistema simples (não protegendo outros...) também fale sobre como testar a configuração.
- Adicionar informações sobre como configurar um firewall proxy com a Debian GNU/Linux iniciando especificamente com pacotes fornecendo serviços proxy (como `xfwp`, `xproxy`, `ftp-proxy`, `redir`, `smtpd`, `nntp-cache`, `dnrd`, `jftpgw`, `oops`, `pdnsd`, `perdition`, `transproxy`, `tsocks`). Deverá ser apontado para um manual com mais informações. Note que o `zorp` esta agora disponível como um pacote da Debian e *é* um firewall proxy (ele também fornece pacotes upstream da Debian).
- Informações sobre a configuração de serviços com o file-rc
- Verifique todas as URLs de referência e remova/corrija as que não estão mais disponíveis.
- Adicione informações sobre as substituições disponíveis (na Debian) para serviços padrões que são úteis para funcionalidades limitadas. Exemplos:
  - lpr local com o cups (pacote)?
  - lpr remota com o lpr
  - bind com dnrd/maradns
  - apache com dhttpd/thttpd/wn (tux?)
  - exim/sendmail com ssmtpd/smtpd/postfix
  - squid com tinyproxy
  - ftpd com oftpd/vsftp
  - ...
- Mais informações sobre patches do kernel relacionadas a segurança, incluindo os acima e informações específicas de como ativar estes patches em um sistema Debian.
  - Detecção de Intrusão do Linux (`lids-2.2.19`)
  - Linux Trustees (no pacote `trustees`)
  - NSA Enhanced Linux (http://www.coker.com.au/selinux/)
  - kernel-patch-2.2.18-openwall (http://packages.debian.org/kernel-patch-2.2.18-openwall)
  - `kernel-patch-2.2.19-harden`

  - kernel-patch-freeswan, kernel-patch-int

- Detalhes sobre como desligar serviços desnecessários (como o inetd), é parte do procedimento de fortalecimento mas pode ser um pouco mais abrangente.
- Informações relacionadas a rotacionamento de senhas que é diretamente relacionada a política.
- Policy, e educação de usuários sobre a política.
- Mais sobre tcpwrappers, e wrappers em geral?
- O arquivo hosts.equiv e outros maiores buracos de segurança.
- Assuntos relacionados a serviços de compartilhamento de arquivos tais como Samba e NFS?
- suidmanager/dpkg-statoverrides.
- lpr e lprng.
- Desligar os ítens do gnome relacionados a IP
- Falar sobre o pam_chroot (ver http://lists.debian.org/debian-security/2002/debian-security-200205/msg00011.html) e como ele é útil para limitação de usuários. Introduzir informações relacionadas ao http://online.securityfocus.com/infocus/1575. Pdmenu, por exemplo está disponível na Debian (enquanto o flash não).
- Falar sobre como executar serviços em ambiente chroot, mais informações em http://www.linuxfocus.org/English/January2002/article225.shtml, http://www.nuclearelephant.com/papers/chroot.html e http://www.linuxsecurity.com/feature_stories/feature_story-99.html
- Fale sobre programas para fazer jaulas chroot. Compartment e chrootuid estão aguardando na incoming. Alguns outros como o (makejail, jailer) podem também serem introduzidos.
- Adicionar informações fornecidas por Pedro Zorzenon sobre como fazer chroot do Bind 8 somente para a :(, veja http://people.debian.org/~pzn/howto/chroot-bind.sh.txt (incluir todo o roteiro?).
- Mais informações relacionadas a programas de análise de logs (i.e. logcheck e logcolorise).
- roteamento 'avançado' (policiamento de tráfego é relacionado a segurança)
- limitando o acesso do ssh a executar somente certos comandos.
- usando o dpkg-statoverride.
- métodos seguros de compartilhar um gravador de CD entre usuários.

- métodos seguros de fornecer som em rede em adição a características display (assim o som de clientes X são enviados para o hardware de som do servidor X).
- tornando navegadores mais seguros.
- configurando ftp sobre `ssh`.
- usando sistemas de arquivos loopback criptográficos.
- encriptando todo o sistema de arquivos.
- ferramentas de steganografia.
- ajustando um PKA para uma empresa.
- usando o LDAP para gerenciar usuários. Existe um howto do ldap+kerberos para o Debian em www.bayour.com escrito por Turbo Fredrikson.
- Como remover informações de utilidade reduzida em sistemas de produção tal como `/usr/share/doc`, `/usr/share/man` (sim, segurança pela obscuridade).
- Mais informações baseadas em ldap dos pacotes contendo os arquivos README (bem, não ainda, mas veja Bug #169465 (`http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=169465`)) e a partir do artigo LWN: desenvolvimento do Kernel (`http://lwn.net/1999/1202/kernel.php3`).
- Adicionar o artigo do Colin sober como configurar um ambiente chroot para um sistema sid completo (`http://people.debian.org/~walters/chroot.html`)
- Adicionar informações sobre como executar múltiplos sensores do snort em um determinado sistema (checar pelos relatórios de falhas enviados para o snort)
- Adicionar informações sobre como configurar um honeypot (`honeyd`)
- Descrever situações sobre o (orphaned) e OpenSwan. A seção sobre VPN precisa ser reescrita.

1.6 Alterações/Histórico

1.6.1 Versão 3.1 (Janeiro de 2005)

Alterações feitas por Javier Fernández-Sanguino Peña

- Adicionado esclarecimentos a /usr somente leitura com o patch de Joost van Baal
- Aplicação do do patch de Jens Seidel corrigindo alguns erros.
- FreeSWAN está morto, longa vida ao OpenSWAN.

- Adicionadas informações sobre a restrição de acessos a serviços RPC (quando não podem ser desativados) incluído também o patch fornecido por Aarre Laakso.
- Atualização do script apt-check-sigs do aj.
- Aplicação do patch de Carlo Perassi corrigindo URLs.
- Aplicação do patch de Davor Ocelic corrigindo muitos erros, enganos, urls, gramática e FIXMEs. Também adicionadas mais informações adicionais a respeito de algumas seções.
- Reescrita a seção sobre auditoria do usuário, destacando o uso do script que não tem as mesmas restrições associadas ao histórico do shell.

1.6.2 Versão 3.0 (Dezembro de 2004)

Alterações feitas por Javier Fernández-Sanguino Peña

- Reescrita as informações sobre auditoria de usuário incluindo exemplos de como usar o script.

1.6.3 Versão 2.99 (Março de 2004)

Alterações feitas por Javier Fernández-Sanguino Peña

- Adicionadas informações sobre referências nos DSAs e compatibilidade com o CVE.
- Adicionadas informações a respeito do apt 0.6 (apt-secure colocado na experimental)
- Corrigida a localização do HOWTO sobre como executar daemons em ambiente chroot como sugerido por Shuying Wang.
- Alterada a linha do APACHECTL no exemplo de chroot do Apache (até se não for usado) como sugerido por Leonard Norrgard.
- Adicionada uma nota de rodapé a respeito de ataques usando hardlinks caso as partições não fossem configuradas adequadamente.
- Adicionada passos faltantes para executar o bind como named como descrito por Jeffrey Prosa.
- Adicionada notas sobre o Nessus e Snort desatualizados na woody e disponibilidade de pacotes portados para esta versão.
- Adicionado um capítulo a respeito da checagem de integridade periódica.
- Esclarecido o estado de testes a respeito de atualizações de segurança. (Debian bug 233955)

- Adicionadas mais informações a respeito de conteúdo esperado em securetty (pois é específicas de kernel).
- Adicionadas referências ao snoopylogger (bug da Debian 179409)
- Adicionadas referências ao guarddog (bug da Debian 170710)
- Apt-ftparchive está no pacote apt-utils, não no apt (obrigado por Emmanuel Chantreau apontar isto)
- Removido o jvirus da lista AV.

1.6.4 Versão 2.98 (Dezembro de 2003)

Alterações por Javier Fernández-Sanguino Peña

- Corrigidas URLs como sugerido por Frank Lichtenheld.
- Corrigido o erro PermitRootLogin como sugerido por Stefan Lindenau.

1.6.5 Versão 2.97 (Setembro de 2003)

Alterações feitas por Javier Fernández-Sanguino Peña

- Adicionadas as pessoas que fizeram as contribuições mais significantes a este manual (por favor, envie um e-mail se achar que deveria estar nesta lista e não está).
- Adicionadas algumas notas a respeito de FIXME/TODOs
- Movidas informações a respeito de atualizações de segurança para o inicio da seção como sugerido por Elliott Mitchell.
- Adicionado o grsecurity a lista de patches do kernel para segurança, mas adicionada uma nota de rodapé sobre situações atuais como sugerido por Elliott Mitchell.
- Removidos os loops (echo para 'todos') no script de segurança de rede do kernel, como sugerido por Elliott Mitchell.
- Adicionadas informações (atualizadas) na seção sobre antivirus.
- Reescrita da seção de proteção contra buffer overflows e adicionadas mais informações sobre patches no compilador para ativar este tipo de proteção.

1.6.6 Versão 2.96 (Agosto de 2003)

Alterações feitas por Javier Fernández-Sanguino Peña

- Removido (e então novamente adicionado) apêndice sobre como rodar o Apache em ambiente chroot. O apêndice agora tem dupla licença.

1.6.7 Versão 2.95 (Junho de 2003)

Alterações feitas por Javier Fernández-Sanguino Peña

- Corrigido erros enviados por Leonard Norrgard.
- Adicionada uma seção sobre como contactar o CERT para manipulação de incidentes (`#after-compromise`)
- Mais informações sobre como tornar um proxy mais seguro.
- Adicionada uma referência e removido um FIXME. Agradecimentos a Helge H. F.
- Corrigido um erro (save_inactive) observado por Philippe Faes.
- Corrigido diversos erros descobertos por Jaime Robles.

1.6.8 Versão 2.94 (Abril de 2003)

Alterações feitas por Javier Fernández-Sanguino Peña

- Segundo as sugestões de Maciej Stachura's, expandi a seção sobre limitação de usuários.
- Corrigidos erros relatados por Wolfgang Nolte.
- Corrigidos os links com o patch contribuído por Ruben Leote Mendes.
- Adicionado um link para o excelente documento de David Wheeler na footnote sobre a contagem de vulnerabilidade de segurança.

1.6.9 Versão 2.93 (Março de 2003)

Alterações feitas por Frédéric Schütz.

- reescrita toda a seção sobre atributos ext2 (lsattr/chattr)

1.6.10 Versão 2.92 (Fevereiro de 2003)

Alterações feitas por Javier Fernández-Sanguino Peña e Frédéric Schütz.

- União da seção 9.3 ("patches úteis do kernel") na seção 4.13 ("Adicionando patches no kernel"), e adicionado algum conteúdo.
- Adicionados alguns TODOs adicionais

- Adicionadas informações sobre como checar manualmente por atualizações e também sobre o cron-apt. Desta forma, o Tiger não é declarado como o único método de verificação de atualizações.

- Regravação da seção sobre a execução de atualizações de segurança devido aos comentários de Jean-Marc Ranger.

- Adicionada uma nota sobre a instalação da Debian (que sugere que o usuário execute uma atualização de segurança após a instalação).

1.6.11 Versão 2.91 (Janeiro/Fevereiro de 2003)

Alterações feitas por Javier Fernández-Sanguino Peña.

- Adicionado um patch contribuído por Frédéric Schütz.

- Adicionadas algumas referências sobre capacidades. Agradecimentos a Frédéric.

- Pequenas alterações na seção sobre o bind adicionando uma referência sobre a documentação on-line do BIND9 e referências apropriadas na primeira área (oi Pedro!)

- Corrigida a data do changelog - ano novo :-)

- Adicionada uma referência sobre o artigo do Colins para os TODOs.

- Removida a referência para o antigo patch ssh+chroot

- Mais patches de Carlo Perassi.

- Correção de enganos (recursivo no Bind é recursão), apontados por Maik Holtkamp.

1.6.12 Versão 2.9 (Dezembro de 2002)

Alterações feitas por Javier Fernández-Sanguino Peña (me).

- Reorganizadas informações sobre o chroot (unidas duas seções, não tem muito sentido tê-las em separado)

- Adicionada as notas sober como executar o Apache em ambiente chroot por Alexandre Ratti.

- Aplicação de patches contribuídos por Guillermo Jover.

1.6.13 Versão 2.8 (Novembro de 2002)

Alterações feitas por Javier Fernández-Sanguino Peña (me).

- Aplicados patches de Carlo Perassi, as correções incluem: nova quebra de linhas, correções de URL, e corrigidos alguns FIXMEs.
- Atualizado o conteúdo da FAQ do time de segurança da Debian.
- Adicionado um link para a FAQ do time de segurança da Debian e da referência do desenvolvedor da Debian, as seções duplicadas podem (apenas podem) serem removidas no futuro.
- Corrigida a seção sobre auditoria manual com comentários de Michal Zielinski.
- Adicionado links para lista de palavras (contribuídos por Carlo Perassi)
- Corrigidos alguns enganos (outros mais estão por vir).
- Corrigidos links TCP como sugerido por John Summerfield.

1.6.14 Versão 2.7 (Outubro de 2002)

Alterações feitas por Javier Fernández-Sanguino Peña (me). Nota: Eu ainda tenho várias correções pendentes em minha caixa postal (que tem atualmente o tamanho de 5MB).

- Algumas correções de erros contribuídas por Tuyen Dinh, Bartek Golenko e Daniel K. Gebhart.
- Nota sobre rootkits relacionados com /dev/kmem contribuído por Laurent Bonnaud
- Corrigidos enganos e FIXMEs contribuídos por Carlo Perassi.

1.6.15 Versão 2.6 (Setembro de 2002)

Alterações feitas por Chris Tillman, tillman@voicetrak.com.

- Alterações para melhorar a gramática/ortografia.
- s/host.deny/hosts.deny/ (1 local)
- Aplicado o patch de Larry Holish's (um pouco grande, corrige diversos FIXMEs)

1.6.16 Versão 2.5 (Setembro de 2002)

Alterações feitas por Javier Fernández-Sanguino Peña (me).

- Corrigidos alguns pequenos erros enviados por Thiemo Nagel.
- Adicionada uma footnote sugerida por Thiemo Nagel.
- Corrigido um link de URL.

1.6.17 Versão 2.5 (Agosto de 2002)

Alterações feitas por Javier Fernández-Sanguino Peña (me). Existem muitas coisas aguardando em minha caixa postal para serem incluídos. assim estarei fazendo isso no lançamento *de volta da lua de mel* :)

- Aplicado um patch enviado por Philipe Gaspar com relação ao Squid que também fecha um FIXME.
- Sim, outro item da FAQ com relação a banners de serviços pego da lista de segurança debian-security (thread "Telnet information" iniciada em 26 de Julho de 2002).
- Adicionada uma note com relação a referências cruzadas do CVE no item da FAQ *Quanto tempo o time de segurança da Debian....*
- Adicionada uma nova seção relacionada a ataques ARP contribuída por Arnaud "Arhuman" Assad.
- Novo item da FAQ com relação ao dmesg e logind e console pelo kernel.
- Pequenos detalhes de informações relacionadas a checagem de assinaturas em pacotes (ele parecia não ter uma versão beta passada).
- Novo item da FAQ com relação a falso positivo de ferramentas de checagem de vulnerabilidades.
- Adicionadas nova seções ao capítulo que contém informações sobre assinatura de pacotes e reorganizando-as como um novo capítulo sobre *Infraestrutura de Segurança na Debian*.
- Novo ítem da FAQ com uma comparação da Debian com outras distribuições Linux.
- Nova seção sobre agentes de mensagem de usuários com funcionalidade GPG/PGP no capítulo ferramentas de segurança.
- Esclarecimentos de como ativa senhas MD5 na woody, adicionadas referências à PAM assim também como uma nota relacionada a definição de max na PAM.

- Adicionado um novo apêndice sobre como criar um ambiente chroot (após brigar um pouco com makejail e corrigindo, também, alguns de seus bugs), integradas informações duplicadas em todo o apêndice.
- Adicionadas mais informações relacionadas ao chroot de `SSH` e seu impacto na transferência segura de arquivos. Algumas informações que foram pegas da lista de discussão debian-security (da thread de Junho de 2002: *transferências de arquivos seguras*).
- Novas seções sobre como fazer atualizações automáticas em sistemas Debian assim também como dicas de uso de testing ou unstable relacionadas com atualizações de segurança.
- Nova seção relacionada sobre como manter-se atualizado com patches de segurança na seção *Antes do comprometimento* assim como uma nova seção sobre a lista de discussão debian-security-announce mailing.
- Adicionadas informações sobre como gerar automaticamente senhas fortes.
- Nova seção com relação a usuários inativos.
- Reorganização da seção sobre como tornar um servidor de mensagens mais seguro com base na discussão sobre instalação *Segura/fortalecida/mínima da Debian (Ou "Porque o sistema básico é do jeito que é?")* que ocorreu na lista debian-security (em Maio de 2002).
- Reorganização da seção sobre parâmetros de rede do kernel, com dados fornecidos pela lista de discussão debian-security (em Maio de 2002, *syn flood attacked?*) e também adicionado um novo ítem da FAQ.
- Nova seção sobre com verificar senhas de usuarios e que pacotes instalar para fazer isto.
- Nova seção sobre a criptografia PPTP com clientes Microsoft discutido na lista debian-security (em Abril de 2002).
- Adicionada uma nova seção descrevendo que problemas existem quando direciona um serviço a um endereço IP específico, esta informação foi escrita baseada em uma lista de discussão da bugtraq com a thread: *Linux kernel 2.4 "weak end host" (anteriormente discutida na debian-security como "problema no")* (iniciada em 9 de Maio de 2002 por Felix von Leitner).
- Adicionadas informações sobre o protocolo versão 2 do `ssh`.
- Adicionadas duas sub-seções relacionadas a configurações seguras do Apache (coisas específicas a Debian, é claro).
- Adicionada uma nova FAQ relacionada a soquetes simples, um relacionado a /root, um ítem relacionado ao grupo users e outra relacionada a log e permissões de arquivos de configuração.
- Adicionada uma referência ao problem na libpam-cracklib que ainda pode estar aberto... (precisa ser verificado)

- Adicionadas mais informações com relação a análise forense (pendente mais informações sobre ferramentas de inspeção de pacotes como `tcpflow`).
- Alterado o ítem "o que posso fazer com relação a comprometimento" na listagem e adicionado mais conteúdo.
- Adicionadas mais informações sobre como configurar o Xscreensaver para bloquear a tela automaticamente após um tempo limite estabelecido.
- Adicionada uma nota relacionada a utilitários que não deve instalar no sistema. Inclui uma nota relacionada ao Perl e porque ele não pode ser facilmente removido da Debian. A idéia veio após ler documentos do Intersects relacionado com o fortalecimento do Linux.
- Adicionadas informações sobre o lvm e sistemas de arquivos com journaling, o ext3 é recomendado. No entanto, as informações lá devem ser muito genérica.
- Adicionado um link para a versão texto on-line (verificar).
- Adicionados mais alguns materiais com relação a informações sobre como fazer um firewall em um sistema local, levado por um comentário feito por Huber Chan na lista de discussão.
- Adicionadas mais informações sobre a limitação do PAM e ponteiros aos documentos de Kurt Seifried's (relacionada a uma postagem por ele na bugtrack em 4 de Abril de 2002 respondendo a uma pessoa que "descobriu" uma vulnerabilidade na Debian GNU/Linux relacionada a esgotamento de recursos).
- Como sugerido por Julián Muñoz, fornecidas mais informações sobre a umask padrão da Debian e o que um usuário pode acessar se tiver um shell em um sistema (provided more information on the default Debian umask and what a user can access if he has been given a shell in the system
- Incluir uma nota na seção sobre senha de BIOS devido a um comentário de Andreas Wohlfeld.
- Incluir patches fornecido por Alfred E. Heggestad corrigindo muitos dos erros ainda presentes no documento.
- Adicionada uma referência ao changelog na seção créditos, pois muitas pessoas que contribuem estão listadas aqui (e não lá).
- Adicionadas algumas notas a mais sobre a seção chattr e uma nova seção após a instalação falando sobre snapshots do sistema. Ambas idéias foram contribuídas por Kurt Pomeroy.
- Adicionada uma nova seção após a instalação apenas para lembrar os usuários de alterar a seqüência de partida.
- Adicionados alguns itens a mais no TODO, fornecidos por Korn Andras.

- Adicionado uma referência as regras do NIST sobre como tornar o DNS mais seguro, fornecidas por Daniel Quinlan.
- Adicionado um pequeno parágrafo relacionado com a infraestrutura de certificados SSL da Debian.
- Adicionadas sugestões de Daniel Quinlan's com relação a autenticação `ssh` e configuração de relay do exim.
- Adicionadas mais informações sobre como tornar o bind mais seguro incluindo alterações propostas por Daniel Quinlan e um apêndice com um script para fazer algumas das alterações comentadas naquela seção.
- Adicionado um ponteiro a outro ítem relacionado a fazer chroot do Bind (precisam ser unidas).
- Adicionada uma linha contribuída por Cristian Ionescu-Idbohrn para pegar pacotes com o suporte a tcpwrappers.
- Adicionada um pouco mais de informações sobre a configuração padrão de PAM da Debian.
- Incluída uma questão da FAQ sobre o uso de PAM para fornecer serviços sem contas shell.
- movidos dois itens da FAQ para outra seção e adicionada uma nova FAQ relacionada com detecção de ataques (e sistemas comprometidos).
- Incluídas informações sobre como configurar uma firewall ponte (incluindo um Apêndice modelo). Obrigado a Francois Bayart quem enviou isto para mim em Março.
- Adicionada uma FAQ relacionada com o syslogd *MARK heartbeat* de uma questão respondida por Noah Meyerhans e Alain Tesio em Dezembro de 2001.
- Incluídas informações sobre proteção contra buffer overflow assim como mais informações sobre patches de kernel.
- Adicionadas mais informações e reorganização da seção sobre firewall. Atualização da informação com relação ao pacote iptables e geradores de firewall disponíveis.
- Reorganização das informações disponíveis sobre checagem de logs, movidas as informações sobre checagem de logs de detecção de intrusão de máquinas para aquela seção.
- Adicionadas mais informações sobre como preparar um pacote estático para o bind em chroot (não testado).
- Adicionado um ítem da FAQ relacionado com servidores/serviços mais específicos (podem ser expandidos com algumas das recomendações da lista debian-security).
- Adicionadas mais informações sobre os serviços RPC (e quando são necessários).

- Adicionadas mais informações sobre capacidades (e o que o lcap faz). Existe alguma boa documentação sobre isto? e não encontrei qualquer documentação em meu kernel 2.4
- Corrigidos alguns enganos.

1.6.18 Versão 2.4

Alterações feitas por Javier Fernández-Sanguino Peña.

- Parte da seção sobre BIOS foi reescrita

1.6.19 Versão 2.3

Alterações feitas por Javier Fernández-Sanguino Peña.

- Trocadas algumas localizações de arquivos com a tage de arquivo.
- Corrigido problema notificado por Edi Stojicevi.
- Leve alteração na seção sobre ferramentas de auditoria remota.
- Adicionados alguns itens para fazer.
- Adicionadas mais informações com relação a impressoras e o arquivo de configuração do cups (pego de uma thread na debian-security).
- Adicionado um patch enviado por Jesus Climent com relação ao acesso de usuários válidos ao sistema no proftpd quando se está configurando um servidor anônimo.
- Pequena alteração nos esquemas de partição para o caso especial de servidores de mensagens.
- Adicionado uma referência do livro "Hacking Linux Exposed" na seção livros.
- Corrigido um erro de diretório notificado por Eduardo Pérez Ureta.
- Corrigido um erro na checklist do /etc/ssh observado por Edi Stojicevi.

1.6.20 Versão 2.3

Alterações feitas por Javier Fernández-Sanguino Peña.

- Corrigida a localização do arquivo de configuração do dpkg.
- Remoção do Alexander das informações de contato.
- Adicionado um endereço alternativo de e-mails.

- Corrigido o endereço de e-mail do Alexander
- Corrigida a localização das chaves de lançamento (agradecimentos a Pedro Zorzenon por nos apontar isto).

1.6.21 Versão 2.2

Alterações feitas por Javier Fernández-Sanguino Peña.

- Corrigidos problemas, agradecimentos a Jamin W. Collins.
- Adicionada uma referência a página de manual do apt-extracttemplate (documenta a configuração do APT::ExtractTemplate).
- Adicionada uma seção sobre o SSH restrito. Informações baseadas naquilo postadas por Mark Janssen, Christian G. Warden e Emmanuel Lacour na lista de segurança debian-security.
- Adicionadas informações sobre programas de anti-vírus.
- Adicionada uma FAQ: logs do su devido a execução do cron como usuário root.

1.6.22 Versão 2.1

Alterações feitas por Javier Fernández-Sanguino Peña.

- Alterado o FIXME a partir do lshell agradecimentos a Oohara Yuuma.
- Adicionados pacote a sXid e removido comentário pois ele *está* disponível.
- Corrigido um número de erros descobertos por Oohara Yuuma.
- ACID está agora disponível na Debian (a partir do pacote acidlab) obrigado a Oohara Yuuma por notificar isto.
- Correção dos links da LinuxSecurity (agradecimentos a Dave Wreski pelo aviso).

1.6.23 Versão 2.0

Alterações feitas por Javier Fernández-Sanguino Peña. Eu queria altera-la para 2.0 quanto todos os FIXMEs estivessem corrigidos, mas esgotei os números 1.9x :(

- Conversão do HOWTO em um Manual (agora eu poderei propriamente dizer RTFM)
- Adicionadas mais informações com relação ao tcp wrappers e a Debian (agora mutos serviços são compilados com suporte a eles assim não será mais um assunto relacionado ao `inetd`).

- Esclarecidas informações sobre a desativação de serviços para torna-lo mais consistente (informações sobre o rpc ainda referenciadas ao update-rc.d)
- Adicionada uma pequena nota sobre o lprng.
- Adicionadas ainda mais informações sobre servidores comprometidos (ainda de uma forma grossa),
- Corrigidos problemas reportados por Mark Bucciarelli.
- Adicionados mais alguns passos sobre a recuperação de senhas para cobrir os casos onde o administrador tem a opção "paranoid-mode=on" ativada.
- Adicionadas mais informações sobre como definir "paranoid-mode=on" quando executa o logon em um console.
- Novo parágrafo para introduzir configuração de serviços.
- Reorganização da seção *Após a instalação* assim ela será quebradas em diversos assuntos e será facilmente lida.
- Escrever informações sobre como configurar firewalls com a configuração padrão da Debian 3.0 (pacote iptables).
- Pequeno parágrafo explicando porque a instalação conectada a Internet não é uma boa idéia e como evitar isto usando ferramentas da Debian.
- Pequeno parágrafo sobre patching referenciando um paper do IEEE.
- Apêndice de como configurar uma máquina Debian com o snort, baseada no que Vladimir enviou para a lista de segurança debian-security (em 3 de Setembro de 2001)
- Informações sobre como o logcheck é configurado na Debian e como ele pode ser configurado para realizar HIDS.
- Informações sobre contabilização de usuários e análise de perfis.
- Incluída a configuração do apt.conf para /usr somente leitura copiada da postagem de Olaf Meeuwissen's para a lista de discussão debian-security.
- Nova seção sobre VPN com alguns ponteiros e pacotes disponíveis na Debian (precisa conteúdo sobre como configurar VPNs e assuntos específicos da Debian), baseada na postagem de Jaroslaw Tabor's e Samuli Suonpaa's post na lista debian-security.
- Pequena nota com relação a alguns programas que podem construir automaticamente jaulas chroot.
- Novo ítem da FAQ relacionado a ident, baseado em uma discussão na lista de discussão debian-security (em Fevereiro de 2002, iniciada por Johannes Weiss).
- Novo ítem da FAQ com relação ao `inetd` baseada em uma discussão da lista de discussão debian-security (em Fevereiro de 2002).

- Introduzida uma nota ao rcconf na seção “desabilitando serviços”.
- Variação da abordagem com relação a LKM, agradecimentos a Philipe Gaspar
- Adicionado ponteiros a documentos do CERT e recursos

1.6.24 Versão 1.99

Alterações feitas por Javier Fernández-Sanguino Peña.

- Adicionado um novo ítem da FAQ com relação ao tempo para se corrigir vulnerabilidades de segurança.
- Reorganizada as seções da FAQ.
- Iniciada a escrita de uma seção com relação a firewall na Debian GNU/Linux (pode ser um pouco alterada).
- Corrigidos problemas enviados por Matt Kraai
- Corrigidas informações relacionadas a DNS
- Adicionadas informações sobre o whisker e nbtscan na seção auditoria.
- Corrigidas algumas URLs incorretas

1.6.25 Versão 1.98

Alterações feitas por Javier Fernández-Sanguino Peña.

- Adicionada uma nova seção sobre auditoria usando Debian GNU/Linux.
- Adicionadas informações relacionadas ao daemon do finger a partir da lista de segurança.

1.6.26 Versão 1.97

Alterações feitas por Javier Fernández-Sanguino Peña.

- Corrigido o link para o Linux trustees
- Corrigidos problemas (patches de Oohara Yuuma e Pedro Zorzenon)

1.6.27 Versão 1.96

Alterações feitas por Javier Fernández-Sanguino Peña.

- Reorganizada a instalação e remoção de serviços e adicionadas algumas novas notas.
- Adicionadas algumas notas com relação ao uso de verificadores de integridade como ferramentas de detecção de intrusão.
- Adicionado um capítulo com relação a assinatura de pacotes.

1.6.28 Versão 1.95

Alterações feitas por Javier Fernández-Sanguino Peña.

- Adicionadas notas com relação a segurança no Squid enviada por Philipe Gaspar.
- Corrigido os links sobre rootkit agradecimentos a Philipe Gaspar.

1.6.29 Versão 1.94

Alterações feitas por Javier Fernández-Sanguino Peña.

- Adicionadas algumas notas com relação ao Apache e o Lpr/lprng.
- Adicionadas mais informações com relação as partições noexec e read-only.
- Reescrita a parte sobre como os usuários podem ajudar a Debian em assuntos relacionados a segurança (item da FAQ).

1.6.30 Versão 1.93

Alterações feitas por Javier Fernández-Sanguino Peña.

- Corrigida a localização do programa mail.
- Adicionados alguns novos itens na FAQ.

1.6.31 Versão 1.92

Alterações feitas por Javier Fernández-Sanguino Peña.

- Adicionada uma pequena seção sobre como a Debian trabalha com a segurança
- Esclarecimentos sobre as senhas MD5 (agradecimentos a "rocky")
- Adicionadas mais informações com relação ao harden-X de Stephen van Egmond
- Adicionados alguns novos itens a FAQ

1.6.32 Versão 1.91

Alterações feitas por Javier Fernández-Sanguino Peña.

- Adicionadas mais informações de forense enviadas por Yotam Rubin.
- Adicionadas informações sobre como construir um honeypot usando a Debian GNU/Linux.
- Adicionados alguns TODOS a mais.
- Corrigidos mais problemas (agradecimentos a Yotam!)

1.6.33 Versão 1.9

Alterações feitas por Javier Fernández-Sanguino Peña.

- Adicionado um patch para corrigir problemas de escrita e algumas informações novas (contribuídas por Yotam Rubin)
- Adicionadas referências a outras documentações online (e offline) ambas na seção (veja 'Esteja ciente dos problemas gerais de segurança' on page 25) e junto com o texto em outra seções.
- Adicionadas algumas informações sobre a configuração de opções do Bind para restringir o acesso ao servidor DNS.
- Adicionadas informações sobre como fortalecer automaticamente um sistema Debian (com relação ao pacote harden e o bastille).
- Removido alguns TODOs fechados e adicionados alguns novos.

1.6.34 Versão 1.8

Alterações feitas por Javier Fernández-Sanguino Peña.

- Adicionada a lista padrão de usuários/grupos fornecidas por Joey Hess a lista de discussão debian-security.
- Adicionadas informações a respeito de root-kits LKM ('Loadable Kernel Modules (LKM)' on page 149) contribuído por Philipe Gaspar.
- Adicionada informação a respeito do Proftp contribuído por Emmanuel Lacour.
- Apêndice de checklist recuperado de Era Eriksson.
- Adicionados alguns novos itens na lista TODO e removidos outros.
- Incluir manualmente os patches e Era pois nem todos foram incluídos na seção anterior.

1.6.35 Versão 1.7

Alterações feitas por Era Eriksson.

- Correção de erros e alterações de palavras

Alterações feitas por Javier Fernández-Sanguino Peña.

- Pequenas mudanças em tags para manter a remoção de tags tt e sua substituição por tags prgn/package.

1.6.36 Versão 1.6

Alterações feitas por Javier Fernández-Sanguino Peña.

- Adicionado ponteiro para documentos como publicado na DDP (deverá substituir o original em um futuro próximo).
- Iniciada uma mini-FAQ (deverá ser expandida) com algumas questões recuperadas de minha caixa de mensagens.
- Adicionadas informações gerais que devem ser consideradas durante a segurança.
- Adicionado um parágrafo relacionado a entrega de mensagens locais (entrada).
- Adicionadas algumas referências a mais informações.
- Adicionadas informações com relação ao serviço de impressão.

- Adicionada uma lista de checagem de fortalecimento de segurança.
- Reorganizadas as informações a respeito de NIS e RPC.
- Adicionadas mais novas notas durante a leitura deste documento em meu novo visor :-)
- Corrigidas algumas linhas mal formatadas.
- Corrigidos alguns problemas.
- Adicionada a idéia do Genus/Paranóia contribuída por Gaby Schilders.

1.6.37 Versão 1.5

Alterações feitas por Josip Rodin e Javier Fernández-Sanguino Peña.

- Adicionados parágrafos relacionados ao BIND e alguns FIXMEs.

1.6.38 Versão 1.4

- Pequeno parágrafo sobre checagem de setuid
- Várias pequenas limpezas
- Encontrado como usar o `sgml2txt -f` para fazer a versão texto

1.6.39 Versão 1.3

- Adicionada uma atualização de segurança após o parágrafo de instalação
- Adicionado um parágrafo relacionado ao proftpd
- Agora realmente escrevia algo sobre o XDM, desculpe pelo atraso

1.6.40 Versão 1.2

- Várias correções gramaticais feitas por James Treacy, novo parágrafo sobre o XDM

1.6.41 Versão 1.1

- Correção de erros, adições diversas

1.6.42 Versão 1.0

- Lançamento inicial

1.7 Créditos e Agradecimentos!

- Alexander Reelsen escreveu o documento original.
- Javier Fernández-Sanguino adicionou mais informações ao documento original.
- Robert van der Meulen forneceu parágrafos relacionados a quota e muitas boas idéias.
- Ethan Benson corrigiu o parágrafo sobre PAM e adicionou boas idéias.
- Dariusz Puchalak contribuiu com algumas informações para vários capítulos.
- Gaby Schilders contribuiu com uma bela idéia geniosa/paranóica.
- Era Eriksson suavizou idiomas em vários lugares e contribuiu com o apêndice com a lista de checagens.
- Philipe Gaspar escreveu detalhes sobre LKM.
- Yotam Rubin contribuiu com correções para muitos erros assim como com informações relacionadas a versões do bind e senhas md5.
- Todas as pessoas que fizeram sugestões para melhorias que (eventualmente) serão incluídas aqui (veja 'Alterações/Histórico' on page 6)
- (Alexander) todas as pessoas que me encorajaram a escrever este HOWTO (que mais adiante se tornou em um Manual).
- A todo o projeto Debian.

Capítulo 2

Antes de você iniciar

2.1 Para que finalidade você quer este sistema?

A segurança no Debian não é tão diferente da segurança em qualquer outro sistema; para implementar a segurança de maneira adequada, você deve primeiro decidir o que você pretende fazer com seu sistema. Após isto, você terá que considerar que as seguintes tarefas precisam ser executadas com cuidado se você realmente quer ter um sistema seguro.

Durante a leitura deste manual você verá tarefas para fazer antes, durante e após você instalar seu sistema Debian. As tarefas são ações como:

- Decidir quais serviços você necessita e limitar o sistema a eles. Isto inclui desativar/desinstalar serviços desnecessários e adicionar filtros como firewall ou tcpwrappers.
- Limitar usuários e permissões em seu sistema.
- Proteger os serviços oferecidos de modo que, em caso de problemas com um serviço, o impacto em seu sistema seja minimizado.
- Utilizar ferramentas apropriadas para garantir que o uso desautorizado seja detectado, de modo que você possa tomar as medidas apropriadas.

2.2 Esteja ciente dos problemas gerais de segurança

Este manual normalmente não entra em detalhes do "porque" algumas coisas são consideradas risco de segurança. Porém, você deve procurar algum conhecimento a mais sobre segurança em sistemas UNIX e em sistemas Linux especificamente. Reserve algum tempo para ler alguns documentos sobre segurança, de modo que você decida conscientemente quando se deparar com diferentes escolhas. O Debian é baseado no kernel do Linux, então você deve procurar muita informação sobre kernel Linux, Debian, outras distribuições e sobre segurança UNIX (mesmo que as ferramentas usadas ou os programas disponíveis sejam diferentes).

Alguns documentos úteis incluem:

- O Linux Security HOWTO (http://www.tldp.org/HOWTO/Security-HOWTO/) (também disponível em LinuxSecurity (http://www.linuxsecurity.com/docs/LDP/Security-HOWTO.html)) é uma das melhores referências sobre Segurança Linux.
- O Security Quick-Start HOWTO for Linux (http://www.tldp.org/HOWTO/Security-Quickstart-HOWTO/) também é um excelente ponto de partida para usuários novos em Linux e em segurança.
- O Linux Security Administrator's Guide (http://seifried.org/lasg/) (fornecido no Debian através do pacote `lasg`) é um guia completo que aborda tudo relacionado a segurança Linux, da segurança do kernel até VPNs. É importante observar que este guia não é atualizado desde 2001, mas algumas informações ainda são relevantes. [1]
- O Securing Linux Step by Step (http://seifried.org/security/os/linux/20020324-securing-linux-step-by-step.html) de Kurt Seifried.
- Em Securing and Optimizing Linux: RedHat Edition (http://www.tldp.org/links/p_books.html#securing_linux) você pode encontrar uma documentação similar a este manual mas relacionada ao Red Hat, alguns assuntos não são específicos de distribuição e podem ser aplicados também ao Debian.
- IntersectAlliance publicou alguns documentos que podem ser usados como referência para reforçar a segurança em servidores linux (e seus serviços), os documentos estão disponíveis em their site (http://www.intersectalliance.com/projects/index.html).
- Para administradores de rede, uma boa referência para construir uma rede segura é o Securing your Domain HOWTO (http://www.linuxsecurity.com/docs/LDP/Securing-Domain-HOWTO/).
- Se você quer avaliar os programas que pretende usar (ou quer construir seus próprios programas) você deve ler o Secure Programs HOWTO (http://www.tldp.org/HOWTO/Secure-Programs-HOWTO/) (uma cópia está disponível em http://www.dwheeler.com/secure-programs/, ela inclui slides e comentários do autor, David Wheeler)
- Se você está considerando instalar um firewall, você deve ler o Firewall HOWTO (http://www.tldp.org/HOWTO/Firewall-HOWTO.html) e o IPCHAINS HOWTO (http://www.tldp.org/HOWTO/IPCHAINS-HOWTO.html) (para kernels anteriores ao 2.4).
- Finalmente, uma boa fonte de consulta para manter em mãos é o Linux Security ReferenceCard (http://www.linuxsecurity.com/docs/QuickRefCard.pdf)

Em qualquer caso, existe mais informação sobre os serviços explanados aqui (NFS, NIS, SMB...) em muitos HOWTOs de The Linux Documentation Project (http://www.tldp.org/). Alguns destes documentos falam em segurança relacionada a um determinado serviço, então certifique-se de procurar com cuidado.

[1] Por um tempo ele foi substituído pelo "Linux Security Knowledge Base". Esta documentação era fornecida no Debian através do pacote `lskb`. Agora ela voltou ao pacote *Lasg* novamente.

Os documentos HOWTO do Projeto de Documentação do Linux (Linux Documentation Project) estão disponíveis no Debian GNU/Linux através dos pacotes `doc-linux-text` (versão texto) ou `doc-linux-html` (versão html). Após a instalação estes documetnos estarão disponíveis em `/usr/share/doc/HOWTO/en-txt` e `/usr/share/doc/HOWTO/en-html`, respectivamente.

Outros livros sobre Linux recomendados:

- Maximum Linux Security : A Hacker's Guide to Protecting Your Linux Server and Network. Anônimo. Paperback - 829 páginas. Sams Publishing. ISBN: 0672313413. Julho 1999.
- Linux Security By John S. Flowers. New Riders; ISBN: 0735700354. Março 1999
- Hacking Linux Exposed (http://www.linux.org/books/ISBN_0072127732.html) de Brian Hatch. McGraw-Hill Higher Education. ISBN 0072127732. Abril, 2001.

Outros livros (que podem ser relacionados a assuntos sobre UNIX e segurança e não especificamente sobre Linux):

- Practical Unix and Internet Security (2nd Edition) (http://www.ora.com/catalog/puis/noframes.html) Garfinkel, Simpson, and Spafford, Gene; O'Reilly Associates; ISBN 0-56592-148-8; 1004pp; 1996.
- Firewalls and Internet Security Cheswick, William R. and Bellovin, Steven M.; Addison-Wesley; 1994; ISBN 0-201-63357-4; 320pp.

Alguns Web sites úteis sobre segurança para manter-se atualizado:

- NIST Security Guidelines (http://csrc.nist.gov/fasp/index.html).
- Security Focus (http://www.securityfocus.com) o servidor que hospeda o banco de dados e a lista do Bugtraq e fornece informações gerais, notícias e relatórios sobre segurança.
- Linux Security (http://www.linuxsecurity.com/). Informações gerais sobre segurança(ferramentas, notícias,...). A página mais útil é main documentation (http://www.linuxsecurity.com/resources/documentation-1.html) .
- Linux firewall and security site (http://www.linux-firewall-tools.com/linux/). Informações gerais sobre firewalls Linux e ferramentas para controlá-los e administrá-los.

2.3 Como o Debian controla a segurança do sistema?

Agora que você tem uma visão geral da segurança no Debian GNU/Linux observe mais algumas considerações para construir um sistema seguro:

- Problemas do Debian são sempre discutidos abertamente, mesmo os relacionados a segurança. Problemas relacionados a segurança são discutidos abertamente na lista debian-security e são publicados no servidor como prevê o Debian Social Contract (`http://www.debian.org/social_contract`):

  Nós não esconderemos problemas

  Nós manteremos nosso banco de dados de bugs e relatórios abertos ao público durante todo o tempo. Relatórios que os usuários enviarem estarão imediatamente on-line para que os outros tenham acesso.

- O Debian sempre procura corrigir os problemas de segurança. A equipe de segurança verifica muitas fontes relacionadas a segurança, a mais importante sendo Bugtraq (`http://www.securityfocus.com/cgi-bin/vulns.pl`), sempre a procura de pacotes que aumentem a segurança e que possam ser incluídos.

- Atualizações de seguranaça estão em primeira prioridade. Quando um problema aparece em um pacote Debian, a atualização de segurança é preparada o mais rápido possível e incluída nas distribuições estável e instável para todas as arquiteturas.

- Informações sobre segurança estão centralizadas em `http://security.debian.org/`.

- O Debian está sempre tentando aumentar sua segurança através de novos projetos como o mecanismo automático de verificação de assinatura dos pacotes.

- O Debian fornece um grande número de ferramentas de segurança para administração de monitoramento do sistema. Desenvolvedores tentam integrar estas ferramentas com a distribuição para fazer um sistema opeacional cada vez mais seguro. Estas ferramentas incluem: verificação da integridade do sistema, firewall, detecção de intrusos, etc.

- Mantenedores de pacote estão cientes dos problemas de segurança. Isto leva a pensar que algumas restrições poderiam ser impostas para alguns serviços em seu uso normal. O Debian, porém, tenta balancear segurança e facilidade de administração - os programas não são desativados quando você os instala (como é o caso nas distribuições da família BSD). Em qualquer caso, implementações de segurança tipo programas `setuid` são parte da política Debian Policy (`http://www.debian.org/doc/debian-policy/`).

Publicando informações de segurança específica para o Debian e complementando outros documentos relacionados a segurança (veja 'Esteja ciente dos problemas gerais de segurança' on page 25), este documento ajuda a produzir sistemas mais seguros.

Capítulo 3

Antes e durante a instalação

3.1 Escolha uma senha para a BIOS

Antes de instalar qualquer sistema operacional em seu computador, configure uma senha para a BIOS. Após a instalação (uma vez que você tenha habilitado o boot a partir do HD) você deve voltar a BIOS e alterar a sequência de boot desabilitando o boot a partir do disquete (floppy), cdrom e outros dispositivos. Se você não fizer assim, um cracker só precisará de acesso físico e um disco de boot para acessar o sistema inteiro.

Desabilitar o boot a menos que uma senha seja fornecida é bem melhor. Isto pode ser muito eficaz num servidor, porque ele não é reiniciado constantemente. A desvantagem desta tática é que o reinício exige intervenção humana, o que pode causar problemas se a máquina não for facilmente acessível.

Observação: muitas BIOS vem de fábrica com senhas padrão bem conhecidas e existem programas que recuperam estas senhas, ou seja, alteram a senha atual para a senha original, para o caso de uma perda da senha pelo administrador. Assim, não dependa desta medida para proteger o acesso ao sistema.

3.2 Particionando o sistema

3.2.1 Escolha um esquema de partição inteligente

Um esquema de partição inteligente depende de como a máquina será usada. Uma boa regra é ser razoavelmente generoso com suas partições e prestar atenção aos seguintes fatores:

- Qualquer diretório que um usuário tenha permissões de escrita, como o `/home`, `/tmp` e o `/var/tmp/`, devem estar separados em uma partição. Isto reduz o risco de um usuário malicioso utilizar o DoS (Denial of Service) para encher seu diretório raiz (/) e tornar o sistema inutilizável (Observação: isto não é totalmente verdade uma vez que

sempre existe algum espaço reservado para o usuário root que o usuário normal não pode preencher), e também previne ataques tipo hardlink. [1]

- Qualquer partição com dados variáveis, isto é, `/var` (especialmente `/var/log`) também deve estar numa partição separada. Em um sistema Debian você deve criar `/var` um pouco maior que em outros sistemas porque o download de pacotes (cache do apt) é armazenado em `/var/cache/apt/archives`.
- Qualquer partição onde você queira instalar software que não é padrão da distribuição deve estar separada. De acordo com a Hierarquia Padrão do Sistema de Arquivos, estas são `/opt` ou `/usr/local`. Se estas partições estão separadas, elas não serão apagadas se você (tiver que) reinstalar o Debian.
- Do ponto de vista da segurança, é sensato tentar mover os dados estáticos para sua própria partição e então montar esta partição somente para leitura. Melhor ainda será colocar os dados numa mídia somenta para leitura. Veja abaixo para mais detalhes.

No caso de um servidor de email é importante ter uma partição separada para o spool de email. Usuários remotos (conhecidos ou não) podem encher o spool de email (`/var/mail` e/ou `/var/spool/mail`). Se o spool está em uma partição separada, esta situação não tornará o sistema inutilizável. Porém (se o diretório de spool está na mesma partição que `/var`) o sistema pode ter sérios problemas: log não serão criados, pacotes podem não ser instalados e alguns programas podem ter problemas ao iniciar (se eles usam `/var/run`).

Para partições que você não tem certeza do espaço necesário, você pode instalar o Logical Volume Manager (`lvm-common` e os binários necessário para o kernel, estes podem ser `lvm10`, `lvm6`, ou `lvm5`). Usando `lvm`, você pode criar grupos de volume que expandem múltiplos volumes físicos.

Escolhendo o sistema de arquivos apropriado

Durante o particionamento do sistema você também tem que decidir qual sistema de arquivos usar. O sistema de arquivos padrão em uma instalação Debian para partições Linux é o `ext2`. Porém é recomendado alterar para um sistema de arquivos journalling como `ext3`, `reiserfs`, `jfs` ou `xfs`, para minimizar os problemas derivados de uma quebra do sistema nos seguintes casos:

- Para laptops em todos os sistemas de arquivos instalados. Assim se acabar a bateria inesperadamente ou o sistema congelar você correrá menos risco de perda de dados durante a reinicialização do sistema.

[1] Um bom exemplo deste tipo de ataque usando /tmp é detalhado em The mysteriously persistently exploitable program (contest) (http://www.hackinglinuxexposed.com/articles/20031111.html) e The mysteriously persistently exploitable program explained (http://www.hackinglinuxexposed.com/articles/20031214.html) (Observe que o incidente é um relato Debian) Ele é basicamente um ataque no qual um usuário local usa uma aplicação setuid vulnerável através de um hard link para ela analisando qualque atualização (ou remoção) do próprio binário feita pelo administrador do sistema. Dpkg foi recentemente corrigido para prevenir isto (veja 225692 (http://bugs.debian.org/225692)) mas outros binários setuid (não controlados pelo gerenciador de pacotes) correm o risco se as partições não estiverem configuradas corretamente.

- para sistemas que armazenam grande quantidade de dados (como servidores de email, servidores ftp, sistemas de arquivos de rede ....). Assim, em caso de queda, menos tempo será gasto para o servidor checar o sistema de arquivos e a probabilidade da perda de dados será menor.

Deixando de lado a performance dos sistemas journalling (uma vez que isto pode iniciar uma verdadeira guerra), normalmente é melhor usar o `ext3`. A razão para isto é que ele é compatível com o antigo `ext2`, assim se existe alguma parte do seu sistema com journalling você pode desabilitar este recurso e ainda ter um sistema em condições de trabalhar. Também, se você precisar recuperar o sistema com um disco de boot (ou CDROM) você não precisa personalizar o kernel. Se o kernel é 2.4, o suporte a `ext3` já está disponível, se é um kernel 2.2 você será capaz de iniciar o sistema de arquivos mesmo se perder as capacidades journalling. Se você estiver usando outro sisteama journalling diferente do `ext3`, você pode não ser capaz de recuperar o sistema a menos que você tenha um kernal 2.4 com os módulos necessários instalados. Se seu disco de resgate tem o kernel 2.2 pode ser mais difícil acessar sistemas `reiserfs` ou `xfs`.

Em qualquer caso, a integridade dos dados pode ser melhor usando `ext3` uma vez que ele usa file-data journalling enquanto outros usam apenas meta-data journalling, veja `http://lwn.net/2001/0802/a/ext3-modes.php3`.

3.3 Não conecte-se a internet até estar pronto

O sistema não deve ser imediatamente conectado a internet durante a instalação. Isto pode parecer estúpido mas intalação via internet é um método comum. Uma vez que o sistema instalará e ativará serviços imediatamente, se o sistema estiver conectado a internet e os serviços não estiverem adequadamente configurados, você estará abrindo brechas para ataques.

Observe também que alguns serviços podem ter vulnerabilidades de segurança não corrigidas nos pacotes que você estiver usando para a instalação. Isto normalmente será verdade se você estiver instalando a partir de mídia antiga (como CD-ROMs). Neste caso, o sistema poderia estar comprometido antes de terminar a instalação!

Uma vez que a instalação e atualizações do Debian podem ser feitas pela internet você pode pensar que é uma boa idéia usar este recurso na instalação. Se o sistema está diretamente conectado (e não está protegido por um firewall ou NAT), é melhor instalar sem conexão com a grande rede usando um mirror local com os pacotes do Debian e as atualizações de segurança. Você pode configurar mirrors de pacotes usando outro sistema conectado com ferramentas específicas do Debian (se ele é um sistema tipo Debian) como `apt-move` ou `apt-proxy`, ou outras, para fornecer os arquivos para o sistema instalado. Se não puder fazer isto, você pode configurar regras de firewall para limitar o acesso ao sistema enquanto estiver atualizando (veja 'Atualização de segurança protegida por um firewall' on page 205).

3.4 Configure a senha do root

Configurar uma boa senha para o root é o requerimento mais básico para ter um sistema seguro. Veja `passwd(1)` para mais dicas de como criar boas senhas. Você também pode usar um programa gerador de senhas para fazer isto para você (veja 'Gerando senhas de usuários' on page 57).

Muita informação sobre a escolha de boas senhas pode ser encontrada na internet; dois locais que fornecem um sumário decente e racional são How to: Pick a Safe Password (`http://wolfram.org/writing/howto/password.html`) do Eric Wolfram e Unix Password Security (`http://www.ja.net/CERT/Belgers/UNIX-password-security.html`) do Walter Belgers.

3.5 Ative os recursos senhas shadow e senhas MD5

No final da instalação, você será perguntado se senhas shadow deve ser habilitada. Responda sim (yes), então as senhas serão mantidas no arquivo `/etc/shadow`. Apenas o root e o grupo shadow terá acesso de leitura a estes arquivo, assim nenhum usuário será capaz de pegar uma cópia deste arquivo para rodar um cracker de senhas nele. Você pode alternar entre senhas shadows e senhas normais a qualquer hora usando `shadowconfig`.

Leia mais sobre senhas Shadow em Shadow Password (`http://www.tldp.org/HOWTO/Shadow-Password-HOWTO.html`) (`/usr/share/doc/HOWTO/en-txt/Shadow-Password.txt.gz`).

Além disso, você é perguntado durante a insatalação se quer usar senhas MD5 .. Isto geralmente é uma boa idéia uma vez que permite senhas longas e melhor encriptação. MD5 permite o uso de senhas com mais de 8 caracteres. Isto, se usado sabiamente, pode tornar mais difícil ataques as senhas do sistema. MD5 é a opção padrão quando instalando a última versão do pacote `password`. Você pode alterar isto a qualquer hora após a instalação executando `dpkg-reconfigure -priority=low passwd`. Você pode reconhecer senhas md5 no arquivo `/etc/shadow` pelo prefixo 1.

Isto modifica todos arquivos sob `/etc/pam.d` pela substituição da linha de senha e inclusão do md5 nela:

```
password requer pam_unix.so md5 nullok obscure min=6 max=16
```

Se `max` não está configurado para mais de 8 a alteração não será útil. Para mais informações leia 'Autenticação do Usuário: PAM' on page 47.

Observação: o padrão de configuração do Debian, mesmo quando ativada a senha MD5, não modifica o valor `max` previamente configurado.

3.6 Rode o mínimo de serviços necessários

Serviços são programas como servidores ftp e servidores web. Uma vez que eles tem que estar *escutando* por conexões que requisitem o serviço, computadores externos podem conectar-se a eles. Serviços algumas vezes são vulneráveis (i.e. podem estar comprometidos sobre um certo ataque) e oferecem risco a segurança.

Você não deve instalar serviços que não são necessários em sua máquina. Todo serviço instalado pode introduzir novos, talvez não óbvios ou conhecidos, buracos de segurança em seu computador.

Como você já deve saber, quando você instala um serviço o padrão é ele ser ativado. Em uma instalação Debian padrão, sem nenhum serviço a mais instalado, o footprint de serviços rodando é baixo mesmo quando falamos de serviços oferecidos para a rede. o footprint no Debian 2.1 não é tão firme quanto no Debian 2.2 (alguns serviços do `inetd` foram habilitados por padrão) e no Debian 2.2 o rpc portmapper é habilitado logo após a installation. Rpc é instalado por padrão porque ele é necessário para muitos serviços, NFS por exemplo. Ele pode ser facilmente removido, porém, veja 'Desabilitando daemons de serviço' on this page como desabilitá-lo.

Quando você instala um novo serviço de rede (daemon) em seu sistema Debian GNU/Linux ele pode ser habilitado de duas maneiras: através do superdaemon `inetd` (uma linha será adicionada ao `/etc/inetd.conf`) ou através de um programa que serve de interface. Estes programas são controlados pelos arquivos `/etc/init.d`, que são chamados no momento da inicialização através do mecanismo SysV (ou outro alternativo) pelo uso de symlinks em `/etc/rc?.d/*` (para mais informações de como isto é feito leia `/usr/share/doc/sysvinit/README.runlevels.gz`).

Se você quer manter algum serviço, mas que será usado raramente, use os comandos update, isto é, `update-inetd` e `update-rc.d` para removê-los do processo de inicialização.

3.6.1 Desabilitando daemons de serviço

Desabilitar um daemon de serviço é simples. Existem vários métodos:

- remover ou renomear os links de `/etc/rc${runlevel}.d/` de modo que eles não iniciem com a letra 'S'
- mover ou renomear o script `/etc/init.d/_service_name_` pra outro nome, por exemplo `/etc/init.d/OFF._service_name_`
- remover a permissão de execução do arquivo `/etc/init.d/_service_name_`.
- editar o script `/etc/init.d/_service_name_` para parar o serviço imediatamente.

Você pode remover os links de `/etc/rc${runlevel}.d/` manualmente ou usando `update-rc.d` (veja `update-rc.d(8)`). Por exemplo, você pode desabilitar um serviço do runlevel multiusuário executando:

```
update-rc.d stop XX 2 3 4 5 .
```

Observe que, se você *não* está usando `file-rc`, `update-rc.d -f _service_ remove` não trabalhará apropriadamente, pois embora *todos* links sejam removidos, após reinstalação ou upgrade do pacote estes links serão regenerados (provavelmente não é o que você quer). Se pensa que isto não é intuitivo você provavelmente está certo (veja Bug 67095 (http://bugs.debian.org/67095)). Texto da manpage:

```
Se qualquer arquivo /etc/rcrunlevel.d/[SK]??name já existe então
update-rc.d não faz nada. É desta maneira que o administrador do sistema po
reorgananizar os links, contanto que eles deixem pelo menos um link remanes
sem ter sua configuração reescrita.
```

Se você está usando `file-rc`, toda informação sobre serviços é manipulada por um arquivo de configuração comum e é mantida mesmo se os pacotes forem removidos do sistema.

Você pode usar a TUI (Text User Interface) fornecida por `rcconf` para fazer todas estas alterações facilmente (`rcconf` trabalha com runlevels `file-rc` e System V).

Outro método (não recomendado) de desabilitar serviços é: `chmod 644 /etc/init.d/daemon` (mas exibe uma mensagem de erro quando iniciando o sistema), ou modificando o script `/etc/init.d/daemon` (adicionando `exit 0` no início ou comentando a instrução `start-stop-daemon`). Como os arquivos do `init.d` são arquivos de configuração, eles não serão reescritos por ocasião da upgrade.

Infelizmente, diferente de outros sistemas operacionanais tipo UNIX, os serviços no Debian não podem ser desabilitados pela modificação dos arquivos em `/etc/default/_servicename_`.

FIXME: Adicione mais informação sobre manipulação de daemons usando file-rc

3.6.2 Desabilitando o `inetd` ou seus serviços

Você deve checar se realmente precisa do daemon `inetd`. Inetd sempre foi uma maneira de compensar deficiências do kernel, mas estas deficiências foram corrigidas. Existe possibilidade de ataques DoS (Denial of Service) contra o `inetd`, então é preferível usar daemons individuais do que rodar um serviço do `inetd`. Se você ainda quer rodar algum serviço do `inetd`, então no mínimo alterne para um daemon mais configurável como `xinetd`, `rlinetd` ou `openbsd-inetd`.

Você deve parar todos os serviços Inetd desnecessários, como `echo`, `chargen`, `discard`, `daytime`, `time`, `talk`, `ntalk` e r-services (`rsh`, `rlogin` e `rcp`) os quais são considerados ALTAMENTE inseguros (use `ssh` no lugar destes).

Você pode desabilitar os serviços editando o arquivo `/etc/inetd.conf` diretamente, mas o Debian fornece uma alternativa melhor: `update-inetd` (o qual comenta os serviços de modo que eles possam facilmente ser reativados). Você pode remover o daemon `telnet` para alterar o arquivo de configuração e reiniciar o daemon (neste caso o serviço `telnet` é desabilitado):

```
/usr/sbin/update-inetd --disable telnet
```

Se você quer um serviço, mas não o quer disponível para todos os IP do seu host, você deve usar um recurso não documentado no `inetd` (substitua o nome do serviço por serviço@ip) ou use um daemon alternativo como `xinetd`.

3.7 Instale o mínimo de software necessário

O Debian vem com *uma grande quantidade* de software, por exemplo o Debian 3.0 *woody* inclui quase 6 CD-ROMs de software e milhares de pacotes. Apesar da grande quantidade de software, a instalação do sistema base utiliza poucos pacotes. [2] você pode estar mal informado e instalar mais que o realmente necessário para seu sistema.

Sabendo o que seu sistema realmente precisa, você deve instalar apenas o que for realmente necessário para seu trabalho. Qualquer ferramenta desnecessária pode ser usada por um usuário malicioso para comprometer o sistema ou por um invasor externo que tenha acesso ao shell (ou código remoto através de serviços exploráveis).

A presença, por exemplo, de utilitários de desenvolvimento (um compilador C) ou linguagens interpretadas (como `perl`, `python`, `tcl`...) pode ajudar um atacante a comprometer o sistema da seguinte maneira:

- permitir a ele fazer escalação de privilégios. Isto facilita, por exemplo, rodar exploits locais no sistema se existe um depurador e compilador prontos para compilar e testar.
- fornecer ferramentas que poderiam ajudar um atacante a usar o sistema comprometido como *base de ataque* contra outros sistemas [3]

É claro que um invasor com acesso ao shell local pode baixar suas próprias ferramentas e executá-las, além disso o próprio shell pode ser usado para fazer complexos programas. Remover software desnecessário não impedirá o problema mas dificultará a ação de um possível atacante. Então, se você deixar disponíveis ferramentas em um sistema de produção que poderiam ser usadas remotamente para um ataque (veja 'Ferramentas de verificação remota de vulnerabilidades' on page 131), pode acontecer de um invasor usá-las.

[2] Por exemplo, no Debian Woody ela gira em torno de 40Mbs, tente isto para ver quanto os pacotes necessários ocupam no sistema:

```
$ size=0 $ for i in `grep -A 1 -B 1 "^Section: base"/var/lib/dpkg/available
| grep -A 2 "^Priority: requiredgrep "^Installed-Sizecut -d : -f 2 `; do
size=$(($size+$i)); done $ echo $size 34234
```

[3] Muitas invasões são feitas mais para acessar os recursos e executar atividades ilícitas (ataques denial of service, spam, rogue ftp servers, poluição dns...) do que para obter dados confidenciais dos sistemas comprometidos.

3.7.1 Removendo Perl

Remover o `perl` pode não ser fácil em um sistema Debian pois ele é muito usado. O pacote `perl-base` tem prioridade classificada como requerida (*Priority: required*), o que já diz tudo. Você pode removê-lo mas não será capaz de rodar qualquer aplicação `perl` no sistema; você ainda terá que enganar o sistema de gerenciamento de pacotes para ele pensar que o `perl-base` ainda está instalado. [4]

Quais utilitários usam `perl`? Você mesmo pode verificar:

```
$ for i in /bin/* /sbin/* /usr/bin/* /usr/sbin/*; do [ -f $i ] && {
type=`file $i | grep -il perl`; [ -n "$type" ] && echo $i; }; done
```

Estes incluem os seguintes utilitários em pacotes com prioridade *required* ou *important*:

- `/usr/bin/chkdupexe` do pacote `util-linux`.
- `/usr/bin/replay` do pacote `bsdutils`.
- `/usr/sbin/cleanup-info` do pacote `dpkg`.
- `/usr/sbin/dpkg-divert` do pacote `dpkg`.
- `/usr/sbin/dpkg-statoverride` do pacote `dpkg`.
- `/usr/sbin/install-info` do pacote `dpkg`.
- `/usr/sbin/update-alternatives` do pacote `dpkg`.
- `/usr/sbin/update-rc.d` do pacote `sysvinit`.
- `/usr/bin/grog` do pacote `groff-base`.
- `/usr/sbin/adduser` do pacote `adduser`.
- `/usr/sbin/debconf-show` do pacote `debconf`.
- `/usr/sbin/deluser` do pacote `adduser`.
- `/usr/sbin/dpkg-preconfigure` do pacote `debconf`.
- `/usr/sbin/dpkg-reconfigure` do pacote `debconf`.
- `/usr/sbin/exigrep` do pacote `exim`.
- `/usr/sbin/eximconfig` do pacote `exim`.
- `/usr/sbin/eximstats` do pacote `exim`.
- `/usr/sbin/exim-upgrade-to-r3` do pacote `exim`.

[4] Você pode fazer (em outro sistema) um pacote dummy com o `equivs`

- `/usr/sbin/exiqsumm` do pacote `exim`.
- `/usr/sbin/keytab-lilo` do pacote `lilo`.
- `/usr/sbin/liloconfig` do pacote `lilo`.
- `/usr/sbin/lilo_find_mbr` do pacote `lilo`.
- `/usr/sbin/syslogd-listfiles` do pacote `sysklogd`.
- `/usr/sbin/syslog-facility` do pacote `sysklogd`.
- `/usr/sbin/update-inetd` do pacote `netbase`.

Assim, sem Perl e, a menos que você recompile estes utilitários em um script shell, você provavelmente não será capaz de gerenciar nenhum pacote (assim você também não será capaza de atualizar o sistema, o que *não é uma coisa boa*).

Se você está determinado a remover o Perl do Debian e tem tempo de sobra, envie os relatórios de bugs referentes aos pacotes acima referidos incluindo possíveis substituições para os utilitários escritas em shell.

3.8 Leia as listas de segurança do Debian (security mailing lists)

Nunca é demais dar uma olhada na lista debian-security-announce, onde avisos e correções dos pacotes são anunciadas pela equipe de segurança do Debian, ou na `mailto:debian-security@lists.debian.org`, onde você pode participar de discussões sobre assuntos relacionados a segurança Debian.

Para receber importantes atualizações de segurança e alertas envie email para debian-security-announce-request@lists.debian.org (`mailto:debian-security-announce-request@lists.debian.org`) com a palavra "subscribe" como assunto. Você também pode inscrever-se nesta lista no endereço `http://www.debian.org/MailingLists/subscribe`

Esta lista tem pouco volume de mensagens e assinando ela você será imediatamente alertado sobre atualizações de segurança para a distribuição Debian. Isto lhe permitirá rapidamente baixar os novos pacotes com atualizações de segurança, as quais são muito importantes na manutenção de um sistema seguro. (Veja 'Executar uma atualização de segurança' on page 40 para detalhes de como fazer isto.)

Capítulo 4

Após a instalação

Assim que o sistema for instalado, você ainda poderá fazer mais para deixá-lo mais seguro; alguns dos passos descritos neste capítulo podem ser seguidos. É claro que isto depende de sua configuração, mas para prevenção de acesso físico você deverá ler 'Altere a BIOS (de novo)' on page 41,'Configurar a senha do LILO ou GRUB' on page 41,'Remover o aviso de root do kernel' on page 42, 'Desativando a inicialização através de disquetes' on page 43, 'Restringindo o acesso de login no console' on page 44 e 'Restringindo reinicializações do sistema através da console' on page 44.

Antes de se conectar a qualquer rede, especificamente se for uma rede pública, no mínimo execute uma atualização de segurança (veja 'Executar uma atualização de segurança' on the next page). Opcionalmente, você deverá fazer um snapshot do seu sistema (veja 'Fazendo um snapshot do sistema' on page 77).

4.1 Inscreva-se na lista de discussão "Anúncios de Segurança do Debian"

Para receber informações sobre atualizações e alertas de segurança (DSAs) disponíveis e DSAs você deverá se inscrever na lista de discussão debian-security-announce. Veja 'O time Debian Security' on page 113 para mais informações sobre como o time de segurança do Debian funciona. Para mais informações sobre como se inscrever nas listas de discussões do Debian, leia `http://lists.debian.org`.

Os DSAs são assinados pelo time de segurança do Debian e as assinaturas podem ser pegas através do endereço `http://security.debian.org`.

Você deverá considerar, também, em se inscrever na lista de discussão debian-security (`http://lists.debian.org/debian-security`) para discussões gerais de problemas de segurança no sistema operacional Debian. Na lista você poderá entrar em contato com outros administradores de sistemas experientes, assim como também desenvolvedores do Debian e autores de ferramentas de segurança que podem responder suas questões e oferecer recomendações.

FIXME: também adicionar a chave aqui?

4.2 Executar uma atualização de segurança

Assim que novos bugs são descobertos nos pacotes, os mantenedores do Debian e autores de software geralmente aplicam patches dentro de dias ou até mesmo horas. Após uma falha ser corrigida, um novo pacote é disponibilizado em `http://security.debian.org`.

Se estiver instalando um lançamento do Debian, você deverá ter em mente que desde que o lançamento foi feito devem existir atualizações de segurança que podem determinar um pacote como vulnerável. Também existem lançamentos menores (foram sete no lançamento da 2.2 *potato*) que incluem estas atualizações de pacotes.

Você precisa anotar a data em que a mídia removível foi feita (se estiver usando uma) e verificar o site de segurança para ter certeza que existem atualizações de segurança. Se existem atualizações e você não puder baixar os pacotes de um site security.debian.org em outro sistema (você não está conectado na Internet ainda? está?) antes de se conectar a rede você deverá considerar (se não estiver protegido por um firewall, por exemplo) adicionar regras de firewall assim seu sistema somente poderá se conectar a security.debian.org e então executar a atualização. Um modelo de configuração é mostrado em 'Atualização de segurança protegida por um firewall' on page 205.

*Nota:*Desde o Debian woody 3.0, após a instalação você terá a oportunidade de adicionar atualizações de segurança ao sistema. Se disser "sim" a isto, o sistema de instalação tomará os passos apropriados para adicionar a fonte de origem para as atualizações de segurança para sua origem de pacotes e seu sistema. Se já tiver uma conexão de Internet, o sistema baixará e instalará qualquer atualização de segurança que produziu após a mídia ser criada. Se estiver atualizando a partir de uma versão anterior do Debian, o perguntou ao sistema de instalação para não fazer isto, você deverá realizar os passos descritos aqui.

Para atualizar manualmente o sistema, insira a seguinte linha em seu `sources.list` e você obterá as atualizações de segurança automaticamente, sempre que atualizar seu sistema.

```
deb http://security.debian.org/ stable/updates main contrib non-free
```

Assim que instalar isto, você poderá usar ou o `apt` ou `dselect` para atualizar:

- Se quiser usar o `apt` simplesmente execute (como root):

  ```
  # apt-get update
  # apt-get upgrade
  ```

- Se quiser usar o `dselect` então primeiro execute o [U]pdate, então [I]nstall e depois, finalmente, [C]onfigure pata instalar/atualizar os pacotes.

Se quiser, você também poderá adicionar linhas deb-src ao seu arquivo `/etc/apt/sources.list`. Veja `apt(8)` para mais detalhes.

Nota: Você *não* precisa adicionar a seguinte linha:

```
deb http://security.debian.org/debian-non-US stable/non-US main contrib non
```

isto é porque security.debian.org é hospedado em uma localização fora dos Estados Unidos e não possui um arquivo separado non-US.

4.3 Altere a BIOS (de novo)

Se lembra 'Escolha uma senha para a BIOS' on page 29? Bem, então você deve agora, uma vez que não precisa inicializar através de uma mídia removível, alterar a configuração padrão da BIOS, desta forma ela poderá somente *inicializar* a partir do disco rígido. Tenha certeza de que não perderá a senha da BIOS, caso contrário, se ocorrer uma falha no disco rígido você não será capaz de retornar a BIOS e alterar a configuração e recuperá-la usando, por exemplo, um CD-ROM.

Outro método mais conveniente, mas menos seguro, é alterar a configuração para ter o sistema inicializando a partir do disco rígido e, caso falhe, tentar a mídia removível. Por agora, isto é feito freqüentemente porque a maioria das pessoas não usam a senha de BIOS com freqüência; pois se esquecem dela facilmente.

4.4 Configurar a senha do LILO ou GRUB

Qualquer um pode facilmente obter uma linha de comando de root e alterar sua senha entrando com o parâmetro `<name-of-your-bootimage> init=/bin/sh` no aviso de boot. Após alterar a senha e reiniciar o sistema, a pessoa terá acesso ilimitado como usuário root e poderá fazer qualquer coisa que quiser no sistema. Após este processo, você não terá acesso root ao seu sistema, já que não saberá mais sua senha.

Para se assegurar que isto não ocorra, você deverá definir uma senha para o gerenciador de partida. Escolha entre uma senha global ou uma senha para determinada imagem.

Para o LILO, você precisará editar o arquivo de configuração `/etc/lilo.conf` e adicionar uma linha `password` e `restricted` como no exemplo abaixo.

```
image=/boot/2.2.14-vmlinuz
   label=Linux
   read-only
   password=mude-me
   restricted
```

Quando terminar, re-execute o lilo. Caso omita `restricted` o lilo sempre perguntará por uma senha, não importando se foram passados parâmetros de inicialização. As permissões padrões do `/etc/lilo.conf` garantem permissões de leitura e gravação para o root e permite o acesso somente leitura para o grupo do `lilo.conf`, geralmente root.

Caso utilize o GRUB ao invés do LILO, edite o `/boot/grub/menu.lst` e adicione as seguintes duas linhas no topo do arquivo (substituindo, é claro `mude-me` pela senha designada). Isto evita que usuários editem os itens de inicialização. A opção `timeout 3` especifica uma espera de 3 segundos antes do `grub` inicializar usando o item padrão.

```
timeout 3
password mude-me
```

Para fortalecer futuramente a integridade da senha, você poderá armazenar a senha em um formato criptografado. O utilitário `grub-md5-crypt` gera um hash de senha que é compatível com o algoritmo de senha encriptada pelo grub (md5). Para especificar no `grub` que uma senha no formato md5 será usada, use a seguinte diretiva:

```
timeout 3
password --md5 $1$bw0ez$tljnxxKLfMzmnDVaQWgjP0
```

O parâmetro –md5 foi adicionado para instruir o `grub` a fazer o processo de autenticação md5. A senha fornecida é uma versão encriptada md5 do mude-me. O uso do método de senhas md5 é preferido em contrapartida da seleção de sua versão texto plano. Mais informações sobre senhas do `grub` podem ser encontradas no pacote `grub-doc`.

4.5 Remover o aviso de root do kernel

Os kernels 2.4 do Linux oferecem um método de acessar um shell de root durante a inicialização que será logo mostrado após de carregar o sistema de arquivos cramfs. Uma mensagem aparecerá para permitir ao administrador entrar com um interpretador de comandos executável com permissões de root, este shell poderá ser usado para carregar manualmente módulos quando a auto-detecção falhar. Este comportamento é padrão para o `linuxrc` do `initrd`. A seguinte mensagem será mostrada:

```
Press ENTER to obtain a shell (waits 5 seconds)
```

Para remover este comportamento, você precisará alterar o `/etc/mkinitrd /mkinitrd.conf` e definir:

```
# DELAY  O número de segundos que o script linuxrc deverá aguardar para
# permitir ao usuário interrompe-lo antes do sistema ser iniciado
DELAY=0
```

Então gere novamente sua imagem do disco ram. Um exemplo de como fazer isto:

```
# cd /boot
# mkinitrd -o initrd.img-2.4.18-k7 /lib/modules/2.4.18-k7
```

ou (preferido):

```
# dpkg-reconfigure -plow kernel-image-2.4.x-yz
```

Note que o Debian 3.0 woody permite aos usuários instalarem o kernel 2.4 (selecionando *tipos de kernels*), *no entanto* o kernel padrão é o 2.2 (salvo para algumas arquitetura no qual o kernel 2.2 ainda não foi portado). Se você acha que isto é um bug, veja Bug 145244 (http://bugs.debian.org/145244) antes de reporta-lo.

4.6 Desativando a inicialização através de disquetes

O MBR padrão no Debian antes da versão 2.2 não atua como setor mestre de partida como recomendado e deixa aberto um método de se fazer a quebra do sistema:

- Pressione shift durante a inicialização, e um aviso MBR aparecerá
- Então aperte F e o sistema inicializará pelo disquete. Isto pode ser usado para se obter acesso root ao sistema.

Este comportamento pode ser alterado com:

```
lilo -b /dev/hda
```

Agora o LILO foi colocado na MBR. Isto também pode ser feito adicionando-se `boot=/dev/hda` ao arquivo de configuração `lilo.conf`. Existe também outra solução que desativa o prompt MBR completamente:

```
install-mbr -i n /dev/hda
```

Por outro lado, esta "porta dos fundos", no qual muitas pessoas simplesmente não se preocupam, podem salvar pessoas que tiverem problemas com sua instalação por quaisquer razões.

FIXME verifique se isto é realmente verdade no kernel 2.2, ou foi no 2.1? INFO: Os disquetes de inicialização no Debian 2.2 não instalam o mbr, mas somente o LILO.

4.7 Restringindo o acesso de login no console

Algumas políticas de segurança podem forçar os administradores a entrar no sistema através do console com seus usuários/senhas e então se tornar o superusuário (com o `su` ou `sudo`). Esta política é implementada no Debian editando-se o arquivo `/etc/login.defs` ou `/etc/securetty` quando utilizar PAM. Em:

- `login.defs`, editando a variável CONSOLE que define um arquivo ou lista de terminais nos quais o login do root é permitido
- `securetty` [1] adicionando/removendo os terminais nos quais o root tem permissão de acesso. Se você deseja permitir somente acesso a console local então você precisa por *console*, *ttyX* [2] e *vc/X* (se estiver usando dispositivos *devfs*), você pode querer adicionar também *ttySX* [3] se estiver usando um console serial para acesso local (onde X é um inteiro, você pode querer ter múltiplas instâncias [4] dependendo do nível de consoles virtuais que tem ativado no `/etc/inittab` [5]). Para mais informações sobre dispositivos de terminais, leia o Text-Terminal-HOWTO (http://tldp.org/HOWTO/Text-Terminal-HOWTO-6.html)

Quando utilizar PAM, outras alterações no processo de login, que podem incluir restrições a usuários e grupos em determinadas horas, podem ser configurados no `/etc/pam.d/login`. Uma característica interessante que pode ser desativada é a possibilidade de fazer login sem senhas. Esta característica pode ser limitada removendo-se *nullok* da seguinte linha:

```
auth       required   pam_unix.so nullok
```

4.8 Restringindo reinicializações do sistema através da console

Caso seu sistema tenha um teclado conectado, qualquer um (sim, *qualquer um*) poderá reinicializar o sistema sem efetuar login. Isto pode se encaixar ou não em sua política de segurança. Se deseja restringir isto, você deverá alterar o arquivo `/etc/inittab` assim a linha que inclui a chamada para `ctrlaltdel` executará `shutdown` com a opção `-a` (lembre-se de executar o `init q` após realizar qualquer modificação neste arquivo). O padrão no Debian inclui esta opção:

```
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now
```

[1] O arquivo `/etc/securetty` é um arquivo de configuração que pertence ao pacote `login`.

[2] Ou *ttyvX* no GNU/FreeBSD, e *ttyE0* no GNU/NetBSD.

[3] Ou *comX* no GNU/Hurd, *cuaaX* no GNU/FreeBSD, e *ttyXX* no GNU/KNetBSD.

[4] A configuração padrão na *woody* incluem 12 tty e consoles vc locais. Na *sarge* a configuração padrão oferece 64 consoles para consoles tty e vc. Você pode remover seguramente isto se não estiver usando mais do que estas consoles.

[5] Procure pelas chamadas *getty*.

Agora para permitir somente que *alguns* usuários possam desligar o sistema, como descreve a página de manual `shutdown(8)`, você deverá criar o arquivo `/etc/shutdown.allow` e incluir lá os nomes de usuários que podem reiniciar o sistema. Quando a combinação de três teclas (a.k.a. *Ctrl+Alt+del*) for feita, o programa verificará se qualquer um dos usuários listados estão conectados ao sistema. Se nenhum deles estiver, o `shutdown` *não* reiniciará o sistema.

4.9 Montando partições do jeito certo

Quando montar uma partição ext2, existem diversas opções adicionais que pode utilizar para a chamada de montagem ou para o `/etc/fstab`. Por exemplo, esta é minha configuração do fstab para a partição `/tmp`:

```
/dev/hda7    /tmp    ext2    defaults,nosuid,noexec,nodev    0    2
```

Observe as diferenças na seção opções. A opção `nosuid` ignore os bits setuid e setgid completamente, enquanto a `noexec` proíbe a execução de qualquer programa naquele ponto de montagem, e a `nodev` ignora dispositivos. Isto soa muito bem, mas elas:

- somente se aplicam a sistemas de arquivos ext2
- podem ser burlados facilmente

A opção `noexec` evita que os binários sejam executados diretamente, mas isto é facilmente contornado:

```
alex@joker:/tmp# mount | grep tmp
/dev/hda7 on /tmp type ext2 (rw,noexec,nosuid,nodev)
alex@joker:/tmp# ./date
bash: ./date: Permission denied
alex@joker:/tmp# /lib/ld-linux.so.2 ./date
Sun Dec  3 17:49:23 CET 2000
```

No entanto, muitos script kiddies tem exploits que tentam criar e executar arquivos em `/tmp`. se eles não tem conhecimento disto, caem nesta restrição. Em outras palavras, um usuário não pode ser convencido a executar um binário alterado em `/tmp` e.g. quando acidentalmente adicionar `/tmp` em sua variável PATH.

Esteja já avisado que muitos scripts dependem de `/tmp` sendo executável. Mais notavelmente, o Debconf tem (ainda?) alguns problemas relacionados a isto, para mais informações veja o bug 116448 (`http://bugs.debian.org/116448`).

A parte a seguir é mais um tipo de exemplo. Uma nota, no entanto: `/var` pode ser ajustado para noexec, mas alguns programas [6] mantém seus programas sob `/var`. O mesmo se aplica a opção nosuid.

[6] Alguns destes incluem o gerenciador de pacotes `dpkg` pois os scripts de instalação (post,pre) e remoção (post,pre) estão em `/var/lib/dpkg/` e também o Smartlist

```
/dev/sda6   /usr          ext2    defaults,ro,nodev       0       2
/dev/sda12  /usr/share    ext2    defaults,ro,nodev,nosuid        0       2
/dev/sda7   /var          ext2    defaults,nodev,usrquota,grpquota0       2
/dev/sda8   /tmp          ext2    defaults,nodev,nosuid,noexec,usrquota,grpqu
/dev/sda9   /var/tmp      ext2    defaults,nodev,nosuid,noexec,usrquota,grpqu
/dev/sda10  /var/log      ext2    defaults,nodev,nosuid,noexec    0       2
/dev/sda11  /var/account  ext2    defaults,nodev,nosuid,noexec    0       2
/dev/sda13  /home         ext2    rw,nosuid,nodev,exec,auto,nouser,async,usrq
/dev/fd0    /mnt/fd0      ext2    defaults,users,nodev,nosuid,noexec      0
/dev/fd0    /mnt/floppy   vfat    defaults,users,nodev,nosuid,noexec      0
/dev/hda    /mnt/cdrom    iso9660 ro,users,nodev,nosuid,noexec            0
```

4.9.1 Ajustando a opção noexec em `/tmp`

Tenha cuidado em ajustar a opção noexec em `/tmp` quando desejar instalar novos programas, pois alguns programas o utilizam para a instalação. O `Apt` é um dos tais programas (veja http://bugs.debian.org/116448), isto pode ser resolvido alterando-se a variável `APT::ExtractTemplates::TempDir` (veja `apt-extracttemplates(1)`). Você poderá definir esta variável no arquivo `/etc/apt/apt.conf` apontando para outro diretório com privilégio de execução ao invés de `/tmp`.

Com relação a noexec, esteja alertado que ela pode não oferecer tanta segurança assim. Considere este exemplo:

```
$ cp /bin/date /tmp
$ /tmp/date
(does not execute due to noexec)
$/lib/ld-linux.so.2 /tmp/date
(funciona, pois o comando date não é executado diretamente)
```

4.9.2 Definindo o /usr como somente-leitura

Se configurar o `/usr` como somente leitura, você não será capaz de instalar novos pacotes em seu sistema Debian GNU/Linux. Você terá primeiro que remontá-lo como leitura-gravação, instalar os pacotes e então remontá-lo como somente-leitura. A última versão do `apt` (no Debian woody 3.0) pode ser configurada para executar comandos antes e após instalar pacotes, assim você pode querer configurá-lo corretamente.

Para fazer isto, modifique o `/etc/apt/apt.conf` e adicione:

```
DPkg
{
    Pre-Invoke  { "mount /usr -o remount,rw" };
    Post-Invoke { "mount /usr -o remount,ro" };
};
```

Note que o Post-Invoke pode falhar com a mensagem de erro "/usr busy". Isto acontece basicamente porque está usando arquivos durante a atualização que foram atualizados. Você encontrará estes programas executando

```
# lsof +L1
```

Interrompa ou reinicie estes programas e execute manualmente o Post-Invoke. *Cuidado!* Isto significa que você provavelmente precisará reiniciar sua seção do X (se estiver executando uma) cada vez que fizer uma grande atualização em seu sistema. Você deverá levar em conta se um sistema de arquivos `/usr` somente-leitura é adequado ao seu sistema. Veja também isto discussion on debian-devel about read-only /usr (http://lists.debian.org/debian-devel/2001/11/threads.html#00212).

4.10 Fornecendo acesso seguro ao usuário

4.10.1 Autenticação do Usuário: PAM

O PAM (módulos de autenticação alteráveis) permite ao administrador do sistema escolher como os aplicativos autenticarão os usuários. Note que o PAM não pode fazer nada caso o aplicativo não esteja compilado com suporte a PAM. A maioria dos aplicativos que vem com o Debian 2.2 tem este suporte ativado. Além do mais, o Debian não tem suporte a PAM em versões anteriores a 2.2. A configuração atual para qualquer serviço que tenha PAM ativado é para emular a autenticação do UNIX (leia `/usr/share/doc/libpam0g/Debian-PAM-MiniPolicy.gz` para mais informações sobre como os serviços PAM *devem* funcionar no Debian).

Cada aplicação com suporte a PAM fornece um arquivo de configuração em `/etc/pam.d/` que pode ser usado para modificar seu comportamento:

- que método é usada para autenticação.
- que método é usada para sessões.
- como a checagem de senha se comportará.

A seguinte descrição está longe de ser completa, para mais informações você deve ler o Guia do Administrador de Sistemas Linux-PAM (http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam.html) (presente no site primário de distribuição do PAM (http://www.kernel.org/pub/linux/libs/pam/)). Este documento também pode ser encontrado no pacote do Debian `libpam-doc`.

O PAM lhe oferece a possibilidade de utilizar vários passos de autenticação de uma só vez, sem o conhecimento do usuário. Você pode autenticar em um banco de dados Berkeley e no banco de dados no arquivo `passwd` padrão, e o usuário somente entrará no sistema caso ele se autentique corretamente em ambos. Você pode restringir muita coisa com o PAM, como

também abrir bastante as portas do seu sistema. Assim, seja cauteloso. Uma linha de configuração típica que tem o tempo de controle como seu segundo elemento: Geralmente ele deve ser ajustado para `requisite`, que retorna uma falha de login caso um dos módulos falhe.

A primeira coisa que eu gosto de fazer é adicionar o suporte a MD5 nas aplicações com suporte a PAM, pois isto nos ajuda a se proteger contra ataques de dicionário (as senhas podem ser maiores se estiver usando MD5). As seguintes duas linhas podem ser adicionadas em todos os arquivos em `/etc/pam.d/` que garantem acesso a máquina, como `login` e `ssh`.

```
# Tenha certeza de instalar primeiro o libpam-cracklib ou então não será ca
# de se logar no sistema
password   required     pam_cracklib.so retry=3 minlen=12 difok=3
password   required     pam_unix.so use_authtok nullok md5
```

Assim, o que este encanto faz? A primeira linha carrega o módulo cracklib do PAM, que fornece checagem de senhas fracas, pergunta por uma nova senha com no mínimo de 12 caracteres, uma diferença de pelo menos 3 letras da antiga senha e permite 3 novas tentativas. O Cracklib depende do pacote wordlist (tal como `wenglish`, `wspanish`, `wbritish`, ...), assim tenha certeza de instalar um que seja apropriado a seu idioma ou o cracklib pode não ser totalmente útil. [7] A segunda linha introduz o módulo de autenticação padrão com senhas MD5 e permite uma senha de tamanho zero. A diretiva `use_authtok` é necessária para pegar a senha do módulo anterior.

Para se assegurar que o usuário root pode somente se logar no sistema de terminais locais, a seguinte linha deverá ser ativada no `/etc/pam.d/login`:

```
auth     requisite  pam_securetty.so
```

Então você deverá modificar a lista de terminais no qual o usuário root pode se logar no sistema no arquivo `/etc/securetty`. Alternativamente, você poderá ativar o módulo `pam_access` e modificar o arquivo `/etc/security/access.conf` que possui um controle de acesso mais fino e geral, mas (infelizmente) não possui mensagens de log decentes (o log dentro do PAM não é padronizado e é particularmente um problema a ser tratado). Nós voltaremos no arquivo `access.conf` um pouco mais a frente.

Em seguida, a seguinte linha deverá ser ativada no `/etc/pam.d/login` para ativar a restrição dos recursos do usuário.

```
session  required   pam_limits.so
```

Isto restringe os recursos do sistema que os usuários têm permissão (veja abaixo em 'Limitando o uso de recursos: o arquivo `limits.conf`' on page 50). Por exemplo, você pode restringir o número de logins concorrentes (de um determinado grupo de usuários, ou de todo o sistema), número de processos, tamanho de memória, etc.

[7] Esta dependência não foi corrigida, no pacote do Debian 3.0. Por favor veja Bug #112965 (http://bugs.debian.org/112965).

Agora, edite o arquivo `/etc/pam.d/passwd` e altere a primeira linha. Você deverá adicionar a opção "md5" para usar senhas MD5, altere o tamanho mínimo da senha de 4 para 6 (ou mais) e ajuste o tamanho máximo, se quiser. A linha resultante deverá se parecer com isto:

```
password   required   pam_unix.so nullok obscure min=6 max=11 md5
```

Se planeja proteger o su, faça isto de forma que somente algumas pessoas possam usá-lo para se tornar o usuário root em seu sistema, você precisará adicionar um novo grupo chamado "wheel" em seu sistema (que é o método mais limpo, pois nenhum arquivo tem tal permissão deste grupo ainda). adicione neste grupo o root e outros usuários que devem ter permissão de `su` para se tornar root. Então adicione a seguinte linha no `/etc/pam.d/su`:

```
auth        requisite   pam_wheel.so group=wheel debug
```

Isto assegura que somente algumas pessoas do grupo "wheel" poderão usar `su` para se tornar o usuário root. Outros usuários não poderão ser capazes de se tornar root. De fato eles obterão uma mensagem de acesso negado ao tentarem se tornar root.

Se deseja que somente alguns usuários se autentiquem em um serviço do PAM, é muito fácil fazer isto usando arquivos onde os usuários que tem permissão de fazer login (ou não) são armazenados. Imagine que você somente deseja permitir o usuário "ref" a fazer o login usando `ssh`. Assim, coloque o usuário no arquivo `/etc/sshusers-allowed` e escreva o seguinte no arquivo `/etc/pam.d/ssh`:

```
auth        required    pam_listfile.so item=user sense=allow file=/etc/ssh
```

Depois crie o arquivo `/etc/pam.d/other` e entre com as seguintes linhas:

```
auth     required       pam_securetty.so
auth     required       pam_unix_auth.so
auth     required       pam_warn.so
auth     required       pam_deny.so
account  required       pam_unix_acct.so
account  required       pam_warn.so
account  required       pam_deny.so
password required       pam_unix_passwd.so
password required       pam_warn.so
password required       pam_deny.so
session  required       pam_unix_session.so
session  required       pam_warn.so
session  required       pam_deny.so
```

Esta linhas lhe oferecerão uma boa configuração padrão para todas as aplicações que suportam PAM (o acesso é negado por padrão).

4.10.2 Limitando o uso de recursos: o arquivo `limits.conf`

Você realmente deverá dar uma olhada séria neste arquivo. Aqui você poderá limitar os recursos usados pelos usuários. Se utilizar PAM, o arquivo `/etc/limits.conf` será ignorado e deverá usar o `/etc/security/limits.conf` ao invés deste.

Se você não restringir o uso de recursos, *qualquer* usuário com um interpretador de comandos válido em seu sistema (ou até mesmo um intruso que comprometeu o sistema através de um serviço) pode usar a quantidade de CPU, memória, pilhas, etc. que o sistema puder fornecer. Este problema de *exaustão de recursos* pode somente ser corrigido com o uso de PAM. Note que lá existe um método para adicionar limitação de recursos para alguns interpretadores de comandos (por exemplo, o `bash` possui `ulimit`, veja `bash(1)`), mas nem todos os interpretadores oferecem as mesmas limitações e também o usuário pode mudar seu shell (veja `chsh(1)`). Então é melhor colocar as limitações nos módulos do PAM.

Mais detalhes podem ser lidos em:

- artigo de configuração do PAM (http://www.samag.com/documents/s=1161/sam0009a/0009a.htm).
- Tornando o Linux mais seguro passo a passo (http://seifried.org/security/os/linux/20020324-securing-linux-step-by-step.html) na seção *Limitando a visão dos usuários*.
- LASG (http://seifried.org/lasg/users/) na seção *Limitando e monitorando usuários*.

FIXME: Colocar um belo `limits.conf` acima deste local

4.10.3 Ações de login do usuário: edite o `/etc/login.defs`

O próximo passo é editar a configuração e ação básica que será feita após o login do usuário. Note que este arquivo não é parte da configuração do PAM, é um arquivo de configuração lido pelos programas `login` e `su`, assim não faz sentido configurá-lo para casos onde nem um dos programas são indiretamente chamados (o programa `getty` que é executado através do console e requer login e senha, *executa* o `login`).

```
FAIL_DELAY          10
```

Esta variável deve se ajustada para um valor alto para tornar difícil ataques brute force através de tentativas de logon no terminal. Caso uma senha incorreta seja digitada, o possível atacante (ou o usuário normal!) terá que aguardar 10 segundos para obter um novo aviso de login, que é bastante tempo quando se utiliza programas automatizados para esta tarefa.

```
FAILLOG_ENAB        yes
```

Se ativar esta variável, as falhas nas tentativas de login serão registradas. É importante mantê-las para pegar alguém que tente fazer um ataque brute force.

```
LOG_UNKFAIL_ENAB    yes
```

Se ajustar a variável *FAILLOG_ENAB* para yes, então você deverá também ajustar esta variável para yes. Isto gravará nomes de usuários desconhecidos caso o login falhar. Se fizer isto, tenha certeza que os logs tenham permissões corretas (640 por exemplo, com a configuração apropriada de grupo tal como adm), pois os usuários podem acidentalmente entrar com suas senhas como se fossem nomes de usuários e você não desejará que outros as vejam.

```
SYSLOG_SU_ENAB      yes
```

Isto somente permite que sejam registradas tentativas do uso de `su` no `syslog`. Muito importante em máquinas em produção, mas note que isto pode criar também problemas de privacidade.

```
SYSLOG_SG_ENAB      yes
```

O mesmo que *SYSLOG_SU_ENAB* mas se aplica ao programa `sg`.

```
MD5_CRYPT_ENAB      yes
```

Como mencionado acima, as senhas em MD5 reduzem fortemente o problema de ataques de dicionário, pois você poderá usar senhas grandes. Se estiver usando a versão slink, leia os documentos sobre MD5 antes de ativar esta opção. Caso contrário, isto é definido no PAM.

```
PASS_MAX_LEN        50
```

Caso senhas MD5 sejam ativadas em sua configuração do PAM, então esta variável deverá ter o mesmo valor da que é usada lá.

4.10.4 Restringindo o ftp: editando o `/etc/ftpusers`

O arquivo `/etc/ftpusers` contém uma lista de usuários que não podem logar no sistema usando ftp. Somente use este arquivo se você realmente deseja permitir ftp (que não é recomendado em geral, pois utiliza autenticação de senhas em texto plano). Caso seu daemon suporta PAM, você poderá também usá-lo para permitir e bloquear usuários para certos serviços.

FIXME (BUG): É m bug que o arquivo `ftpusers` padrão no Debian *não* inclua todos os usuários administrativos (do `base-passwd`).

4.10.5 Usando su

Se você realmente precisa que os usuários se tornem superusuário em seu sistema, e.g. para instalar pacotes ou adicionar usuários, você pode usar o comando `su` para alterar sua identidade. Você deverá tentar evitar se logar como usuário root, usando o `su` ao invés disto. Atualmente a melhor solução é remover o `su` e utilizar os mecanismos do `sudo` que tem uma lógica mais geral e mais características que o `su`. No entanto, o `su` é mais comum e é usado em muitos outros sistemas Unix.

4.10.6 Usando o sudo

O programa `sudo` permite ao usuário executar comandos definidos com outra identidade de usuário, até mesmo como usuário root. Se o usuário for adicionado ao arquivo `/etc/sudoers` e se autenticar corretamente, ele será capaz de executar comandos que foram definidos no `/etc/sudoers`. Violações, tais com senhas incorretas ou tentativa de executar um programa que não tem permissões, são registradas e enviadas para o usuário root.

4.10.7 Desativação de acesso administrativo remoto

Você deverá modificar o `/etc/security/access.conf` para bloquear logins remotos para contas administrativas. Desta forma, usuários precisarão executar o `su` (ou `sudo`) para usar qualquer poder administrativo e traços para auditoria apropriada sempre serão gerados.

Você precisará adicionar a seguinte linha no arquivo `/etc/security/access.conf`, o arquivo padrão de configuração do Debian tem a linha de exemplo comentada:

```
-:wheel:ALL EXCEPT LOCAL
```

Lembre-se de ativar o módulo `pam_access` para cada serviço (ou configuração padrão) em `/etc/pam.d/` se quiser que suas alterações em `/etc/security/access.conf` sejam mantidas.

4.10.8 Restringindo acessos de usuários

Algumas vezes você deve pensar que precisa ter seus usuários criados em seu sistema local para oferecer acesso a um determinado serviço (serviço de mensagens pop3 ou ftp). Antes de fazer isto, primeiro lembre-se que a implementação do PAM no Debian GNU/Linux lhe permite validar usuários em uma grande variedade de serviços de diretório externos (radius, ldap, etc.) através dos pacote libpam.

Caso os usuários precisem ser criados e o sistema poderá ser acessado remotamente, tenha em mente que os usuários poderão ser capazes de se logar no sistema. Você poderá corrigir isto dando aos usuários um shell nulo (`/dev/null`) (ele não precisará estar listado no arquivo

/etc/shells). Se deseja permitir aos usuários acessar o sistema mas com seus movimentos limitados, você poderá usar o /bin/rbash, que é equivalente a adicionar a opção -r ao bash (*RESTRICTED SHELL* veja bash(1)). Por favor observe que até mesmo em um shell restrito, um usuário pode acessar um programa interativo (que pode permitir a execução de um subshell) e poderá pular as limitações do shell.

O Debian atualmente fornece em sua versão instável (e poderá ser incluída em uma futura versão estável) do módulo pam_chroot (no pacote libpam-chroot). Uma alternativa a ele é o chroot o serviço que fornece log remoto (ssh, telnet).[8]

Se você deseja restringir *quando* seus usuários podem acessar o sistema, você deverá personalizar o /etc/security/access.conf a suas necessidades.

Informações sobre como fazer chroot dos usuários acessando o sistema através do ssh é descrito em 'Ambiente chroot para SSH' on page 207.

4.10.9 Auditoria do usuário

Se você é realmente paranóico, pode querer adicionar uma configuração em todo o sistema para auditar o que os usuários estão fazendo no sistema. Esta seção mostra algumas dicas de utilitários diversos que poderão ser usados.

Auditoria de entrada e saída com o script

Você poderá usar o comando script para auditar ambos o que os usuários executam e quais são os resultados destes comandos.Não é possível definir o script como um interpretador de comandos (até mesmo se ele for adicionado ao arquivo /etc/shells). Mas você poderá ter o arquivo de inicialização do shell executando o seguinte:

```
umask 077
exec script -q -a "/var/log/sessions/$USER"
```

É claro, se você fizer isto de forma que afete todo o sistema, significa que o shell não continuará lendo arquivos de configurações pessoais (pois ele será substituído pelo script). Uma alternativa é fazer isto nos arquivos de inicialização do usuário (mas então o usuário poderá removê-la, veja os comentários sobre isto abaixo)

Você também precisa ajustar os arquivos no diretório de auditoria (no exemplo /var/log/sessions/) assim os usuários poderão gravar para ele, mas não poderão remover o arquivo. Isto pode ser feito, por exemplo, criando os arquivos de seção de usuário antecipadamente e definindo a opção *append-only* usando o chattr.

Uma alternativa útil para administradores de sistemas, que inclui informações sobre data, pode ser:

[8] Libpam-chroot ainda não foi ainda testado, ele funciona com o login mas ele não é fácil de ser configurado para funcionar com outros programas

```
umask 077
exec script -q -a "/var/log/sessions/$USER-`date +%Y%m%d`"
```

Usando o arquivo de histórico do interpretador de comandos

Se deseja rever o que o usuário está digitando no seu shell (mas não se sabe qual seu resultado) você poderá configurar um `/etc/profile` para todo o sistema que configura o ambiente de forma que todos os comandos são salvos em um arquivo de histórico. A configuração de todo o sistema precisa ser feita de forma que os usuários não possam remover as capacidades de auditoria em seu shell. Isto muitas vezes é específica de cada shell assim tenha certeza que todos os usuários estão utilizando um shell que suporte isto.

Por exemplo, para o bash, o `/etc/profile` deverá ser ajustado da seguinte forma [9] :

```
HISTFILE=~/.bash_history
HISTSIZE=10000
HISTFILESIZE=999999
# Don't let the users enter commands that are ignored
# in the history file
HISTIGNORE=""
HISTCONTROL=""
readonly HISTFILE
readonly HISTSIZE
readonly HISTFILESIZE
readonly HISTIGNORE
readonly HISTCONTROL
export HISTFILE HISTSIZE HISTFILESIZE HISTIGNORE HISTCONTROL
```

Para isto funcionar, o usuário poderá somente adicionar dados ao `.bash_history`. Você *também* precisará ajustar a opção *append-only* usando o programa `chattr` para o `.bash_history` de todos os usuários. [10].

Note que você poderá introduzir as configurações acima no arquivo `.profile` do usuário. Mas então você precisará ajustar permissões adequadamente de tal forma que isto prevenirá que o usuário modifique este arquivo. Isto inclui: ter os diretórios home dos usuários *não* pertencendo ao usuário (pois ele seria capaz de remover o arquivo) mas da mesma forma permitir ler o arquivo de configuração `.profile` e gravar no `.bash_history`. Seria bom ajustar a opção *imutável* (usando também o `chattr`) também para o `.profile` se fizer desta forma.

[9] A definição de HISTSIZE para um número elevado poderá causar problemas em algumas circunstâncias, pois o histórico é mantido em memória para cada sessão do usuário. Você estará mais seguro caso defina esta variável para um valor suficientemente grande e realizar o backup de arquivos de histórico de usuários (se precisar do histórico de todos os usuários por alguma razão).

[10] Sem a opção append-only, os usuários poderão ser capazes de apagar o conteúdo do arquivo de histórico executando `> .bash_history`

Auditoria completa do usuário com ferramentas de contabilização

O exemplo anterior é um método simples de se configurar a auditoria do usuário, mas pode não ser útil para sistemas complexos ou para este em que os usuários não precisam executar um shell (de forma exclusiva). Neste caso, você precisará dar uma olhada no pacote `acct`, que contém ferramentas de contabilização. Estes utilitários registrarão todos os comandos executados pelos usuários ou por processos no sistema, ao custo de espaço em disco.

Quando ativar a contabilização, todas as informações sobre processos e usuários são mantidas sob `/var/account/`, mais especificamente em `pacct`. O pacote de contabilização inclui algumas ferramentas como (`sa`, `ac` e `lastcomm`) para realizar a análise destes dados.

Outros métodos de auditoria do usuário

Se você for completamente paranóico e deseja auditar cada comando do usuário, você deverá pegar o código fonte do `bash`, editá-lo e assim ter ele enviando tudo o que o usuário digitar para outro arquivo. Ou ter o pacote `ttysnoop` monitorando constantemente qualquer novo ttys [11] e gravar sua saída para um arquivo. Outro programa útil é o `snoopy` (veja também the project page (`http://sourceforge.net/projects/snoopylogger/`)) que é um programa transparente ao usuário que trabalha em cima de uma biblioteca fornecendo um gancho nas chamadas *execve()*, qualquer comando executado é registrado no `syslogd` usando a facilidade `authpriv` (normalmente armazenada em `/var/log/auth.log`).

4.10.10 Revisando perfis de usuários

Se deseja *ver* o que os usuários estão atualmente fazendo quando entram no sistema você poderá usar o banco de dados `wtmp` que inclui todas as informações de login. Este arquivo pode ser processado por vários utilitários, entre eles o `sac` que pode enviar como saída um perfil sobre cada usuário mostrando o intervalo de tempo que eles geralmente entram no sistema.

No caso de ter a contabilização ativada, você também poderá usar as ferramentas fornecidas por ele para ser capaz de determinar quando os usuários acessam o sistema e o que eles executam.

4.10.11 Ajustando a umask dos usuários

Dependendo de sua política de usuários você pode querer alterar como as informações são compartilhadas entre os usuários, o que significa, o que cada permissão padrão permite. Esta alteração é feita definindo uma configuração apropriada de `umask` para todos os usuários. Você poderá alterar a configuração de *UMASK* no arquivos `/etc/limits.conf`, `/etc/profile`, `/etc/csh.cshrc`, `/etc/csh.login`, `/etc/zshrc` e provavelmente em alguns outros (dependendo do tipo de shell que tem instalado em seu sistema). De todos

[11] Ttys são criados para logins locais e logins remotos durante seções do ssh e telnet

estes, o último executado tem preferência. A ordem é: `limits.conf` do PAM, o padrão de configuração do sistema para o shell do usuário, o shell do usuário(seu `~/.profile`, `~/.bash_profile`...)

A configuração padrão de `umask` no Debian é *022* isto significa que o arquivo (e diretórios) podem ser lidos e acessados pelo grupo de usuário e por outros usuários no sistema. Se isto é muito permissivo para o sistema você terá que ajustar a configuração de umask para todos os shells (e para o PAM). Não se esqueça de modificar os arquivos sob `/etc/skel/` pois estes se tornarão os novos padrões do sistema quando criados pelo comando `adduser`.

Note, no entanto que os usuários podem modificar sua própria configuração de `umask` se desejarem, torná-la mais permissiva ou mais restritiva.

4.10.12 Limitando o que os usuários podem ver/acessar

FIXME: É necessário mais conteúdo. Falar das consequências de alterar as permissões de pacotes quando atualiza o sistema (e administração desta paranóia deverá ser através de `chroot` em seus usuários).

Se você precisa garantir acesso dos seus usuários ao sistema usando um interpretador de comandos, pense sobre isto muito cuidadosamente. Um usuário pode por padrão, a não ser que esteja em um ambiente severamente restrito (como uma jaula `chroot`), obter muitas informações sobre o seu sistema, incluindo:

- alguns arquivos de configuração em `/etc`. No entanto, as permissões padrões do Debian para alguns arquivos sensíveis (que podem conter senhas, por exemplo), não terão acesso devido a informações críticas. Para ver que arquivos são somente acessíveis pelo usuário root, por exemplo, execute como superusuário o comando `find /etc -type f -a -perm 600 -a -uid 0`.
- você instalou pacote, ou vendo no banco de dados de pacotes, ou no diretório `/usr/share/doc` ou adivinhando olhando nos binários e bibliotecas instalados em seu sistema.
- alguns arquivos de registro em `/var/log`. Também note que alguns arquivos de registro somente são acessíveis aos usuários root e grupo `adm` (tente executar `find /var/log -type f -a -perm 640`) e alguns são somente disponíveis ao usuário root (tente executar `find /var/log -type f -a -perm 600 -a -uid 0`).

O que um usuário pode ver em seu sistema? Provavelmente muitas coisas, tente isto (faça uma breve parada):

```
find / -type f -a -perm +006 2>/dev/null
find / -type d -a -perm +007 2>/dev/null
```

A saída mostra a lista de arquivos que um usuário pode *ver* e os diretórios que ele tem acesso.

Limitando acesso a outras informações de usuários

Se você ainda permite acesso a shell para os usuários você deverá querer limitar que informações eles podem ver de outros usuários. Os usuários com acesso a shell têm a tendência de criar um número de arquivos dentro do seu diretório pessoal: caixas de correio, documentos pessoais, configurações de aplicativos do X/GNOME/KDE...

No Debian, cada usuário é criado com um grupo associado e nunca dois usuários pertencerão ao mesmo grupo. Este é o comportamento padrão: quando uma conta de usuário é criada, um grupo com o mesmo nome também é criado, e o usuário é adicionado a ele. Isto evita o conceito do grupo *users* compartilhado, que torna mais difícil aos usuários ocultarem informações de outros.

No entanto, os diretórios de usuários em *$HOME* são criados com permissões 0755 (lido pelo grupo e por todos). As permissões de grupo não são críticas pois somente o usuário pertence ao grupo, no entanto as permissões de todos os outros pode (ou não) ser um problema dependendo de sua política local.

Você poderá alterar este comportamento, assim a criação de usuários oferecerá uma permissão diferente em *$HOME*. Para alterar o comportamento para *novos* usuários quando forem criados, altere *DIR_MODE* no arquivo de configuração `/etc/adduser.conf` para 0750 (sem acesso de leitura para todos).

Os usuários ainda poderão compartilhar informações mas não diretamente em seus diretórios *$HOME*, a não ser que eles mudem suas permissões.

Note que a desativação de leitura para todos em diretórios de usuários evitará que os usuários criem suas páginas pessoais no diretório `~/public_html`, pois o servidor web não será capaz de ler um componente no path - seu diretório *$HOME*. Se deseja permitir aos usuários publicar páginas HTML em seus diretórios `~/public_html`,então altere *DIR_MODE* para 0751. Isto permitirá o servidor web acessar o diretório final `public_html` (que terá por si próprio a permissão) e oferecerá o conteúdo publicado pelos usuários. É claro, nós estamos somente falando aqui sobre uma configuração padrão; os usuários podem geralmente ajustar os modos de seus próprios arquivos completamente a seu gosto, ou você poderá manter o conteúdo que tem a intenção de publicação na web em um diretório separado que não seja um subdiretório do diretório de usuário *$HOME*.

4.10.13 Gerando senhas de usuários

Existem muitos casos quando um administrador precisa criar muitas contas de acesso de usuários e fornece senhas a a todas elas. É claro, o administrador poderia somente ajustar a senha para ser a mesma da conta de usuário, mas isto não seria uma atitude muito segura. Uma alternativa melhor é gerar um programa gerador de senhas. O Debian oferece os pacotes `makepasswd`, `apg` e `pwgen` que contém programas (o nome do programa é o mesmo do pacote) que podem ser usados para este propósito. O `makepasswd` gerará senhas aleatórias reais com uma ênfase em segurança até mesmo na pronunciabilidade, enquanto o `pwgen` tentará criar senhas pronunciáveis (é claro que isto dependerá de sua língua mãe). O `apg` tem algo-

rítmos para oferecer ambos (existe uma versão cliente/servidor deste programa mas não está incluída no pacote do Debian).

O `passwd` não permite que uma senha seja definida de forma interativa (pois ele utiliza acesso direto a tty). Se deseja alterar senhas quando cria um grande número de usuários, você poderá criá-las usando o `adduser` com a opção `--disabled-login` e então usar o `usermod` ou `chpasswd` [12] (ambos vêm no pacote `passwd` assim você já os terá instalados). Se desejar usar um arquivo com todas as informações dos usuários como um processo não interativo, será melhor usar o `newusers`.

4.10.14 Verificando senhas de usuários

Senhas de usuários podem algumas vezes ser o *ponto vulnerável* na segurança de um determinado sistema. Isto é devido ao fato de que alguns usuários escolherem senhas fracas para suas contas (e quanto mais deles têm acesso ao sistema, maiores as chances disto acontecer). Até mesmo se você estabelecer checagens com o módulo cracklib do PAM e limitações de senhas como descrito em 'Autenticação do Usuário: PAM' on page 47 os usuários ainda serão capazes de usar senhas simples. Pois o acesso a usuários remotos pode incluir acesso a um shell remoto (felizmente sobre `ssh`) tornando possível deduzir a senha mais difícil para invasores remotos. Especialmente se eles são capazes de coletar informações importantes, tais como nomes de usuários e até dos próprios arquivos `passwd` e `shadow`.

Um administrador de sistema deverá, dado um grande número de usuários, verificar se a senha que eles têm são consistentes com a política local de segurança. Como verificar? Tente quebrá-las assim como um invasor faria se ele tivesse acesso ao hash de senhas (o arquivo `/etc/shadow`).

Um administrador poderia usar o `john` ou `crack` (ambos crackers de senhas força bruta) juntos com um dicionário apropriado para procurar senhas de usuários e ter um plano de ação quando uma senha fraca for detectada. Você pode procurar por pacote Debian que contém lista de palavras de dicionário usando `apt-cache search wordlist` ou visitando os sites clássicos de pesquisas de dicionário tais como `ftp://ftp.ox.ac.uk/pub/wordlists` ou `ftp://ftp.cerias.purdue.edu/pub/dict`.

4.10.15 Logout de usuários ociosos

Usuários inativos geralmente são um risco de segurança, um usuário pode estar inativo porque saiu para comer ou porque ocorreu um problema com sua conexão remota, que não foi restabelecida. Por alguma razão, os usuários inativos podem levar a um comprometimento do sistema:

- porque o console do usuário pode ser destravado e pode ser acessado por um intruso.

[12] O `chpasswd` não trabalha com a geração de senhas em md5, assim ele precisa ser informado que forma de criptografia das senhas será utilizado, com a opção `-e`.

- porque um intruso pode ser capaz de reconectar a si mesmo a uma conexão de rede fechada e enviar comandos ao shell remoto (isto é muito fácil de ser feito caso o shell remoto não seja criptografado como no caso do `telnet`).

Alguns sistemas remotos podem ter sido comprometidos através de uma `screen` inativa (ou desconectada).

A desconexão automática de usuários idle é geralmente parte da política local de segurança que deve ser forçada. Existem várias formas de se fazer isto:

- Caso o interpretador de comandos do usuário seja o `bash`, o administrador do sistema poderá definir um valor para a variável `TMOUT` (veja `bash(1)`) que fará o shell deslogar os usuários inativos automaticamente. Note que ela deverá ser definida com a opção `-o` ou os usuários serão capazes de alterá-la (ou desativá-la).
- Instale o `timeoutd` e configure `/etc/timeouts` de acordo com sua política de segurança local. O daemon observará usuários inativos e respectivamente fará o logout de suas seções.
- Instale o `autolog` e o configure para remover usuários inativos.

Os daemons `timeoutd` ou `autolog` são os métodos preferidos, pois, após tudo, os usuários podem alterar seu shell padrão ou podem alterar para um outro shell que não possua tais controles.

4.11 Usando os tcpwrappers

Os TCP wrappers foram desenvolvidos quando não existiam filtros de pacotes disponíveis e eram necessários controle de acesso. Mesmo assim, eles ainda são muito interessantes e úteis. Com os TCP wrappers é possível permitir ou negar um serviço para uma máquina ou domínio e definir uma regra padrão também para permitir ou negar (tudo feito a nível de aplicação). Se desejar mais informações, dê uma olhada em `hosts_access(5)`.

Muitos dos serviços instalados no Debian são executados de duas formas:

- carregados através do serviço tcpwrappers (`tcpd`)
- compilados com o suporte a libwrapper embutido

De um lado, para serviços configurados no `/etc/inetd.conf` (isto inclui o `telnet`, `ftp`, `netbios`, `swat` e `finger`) você verá que o arquivo de configuração executa primeiro o `/usr/sbin/tcpd`. De outro lado, até mesmo se um serviço não for carregado pelo super-daemon `inetd`, o suporte a regras do tcp wrappers pode ser compilado nele. Os serviços compilados com o tcp wrappers no Debian incluem o `ssh`, `portmap`, `in.talk`, `rpc.statd`, `rpc.mountd`, `gdm`, `oaf` (o daemon ativador do GNOME), `nessus` e muitos outros.

Para ver que pacotes usam o tcpwrappers, execute:

```
$ apt-cache showpkg libwrap0 | egrep '^[[:space:]]' | sort -u | \
      sed 's/,libwrap0$//;s/^[[:space:]]\+//'
```

Leve isto em conta quando executar o `tcpdchk` (um verificar de sintaxe e regras de arquivos muito útil que vem com o TCP wrappers). Quando adicionar serviços stand-alone (que são ligados diretamente com a biblioteca wrapper) nos arquivos `hosts.deny` e `hosts.allow`, o `tcpdchk` deverá te alertar que não é capaz de encontrar o serviço mencionado pois ele somente procura por eles no arquivo `/etc/inetd.conf` (a página de manual não é totalmente precisa com relação a este ponto).

Agora, vem uma pequena dica, e provavelmente o menor sistema de detecção de intrusão disponível. Em geral, você deverá ter uma política de firewall decente como primeira linha e o tcp wrappers como segunda linha de defesa. Um pequeno truque é configurar um comando *SPAWN* [13], no arquivo `/etc/hosts.deny` que envia uma mensagem para o root assim que for tentado acesso a um serviço negado:

```
ALL: ALL: SPAWN ( \
  echo -e "\n\
  TCP Wrappers\: Connection refused\n\
  By\: $(uname -n)\n\
  Process\: %d (pid %p)\n\
  User\: %u\n\
  Host\: %c\n\
  Date\: $(date)\n\
" | /usr/bin/mail -s "Conexão bloqueada para %d" root) &
```

Cuidado: O exemplo impresso acima é aberto a um ataque DoS fazendo muitas conexões em um curto período de tempo. Muitos e-mails significam muito I/O de arquivos pelo envio de poucos pacotes.

4.12 A importância dos logs e alertas

É fácil ver que o tratamento de mensagens de logs e alertas é um assunto importante em um sistema seguro. Suponha que um sistema está perfeitamente configurado e é 99% seguro. Se a probabilidade de 1% do ataque ocorrer e não existir medidas de segurança no lugar, para primeiro detectar e segundo disparar alarmes, o sistema não estará bem seguro.

O Debian GNU/Linux fornece algumas ferramentas que fazem análise de logs, mais notavelmente `swatch`, [14] `logcheck` ou `log-analysis` (todos precisarão de algumas personalizações para que coisas desnecessárias sejam removidas do relatório). Também pode ser útil, se o sistema estiver visivelmente próximo, ter as mensagens do sistema mostradas em um console virtual. Isto é útil, pois você pode (através de certa distância) ver se o sistema

[13] tenha certeza de usar maiúsculas, pois *spawn* não executará fork

[14] existe um artigo muito bom, escrito por Lance Spitzner (http://www.spitzner.net/swatch.html)

está se comportando adequadamente. O `/etc/syslog.conf` do Debian vem com uma configuração padrão comentada; para ativá-la, descomente as linhas e reinicie o `syslogd` (`/etc/init.d/syslogd restart`):

```
daemon,mail.*;\
      news.=crit;news.=err;news.=notice;\
      *.=debug;*.=info;\
      *.=notice;*.=warn       /dev/tty8
```

Para tornar os logs coloridos, você deverá dar uma olhada nos pacotes `colorize`, `ccze` ou `glark`. Existe muita coisa sobre análise de logs que não poderá ser coberta aqui, assim uma boa fonte de informações pode ser o site Log Analysis (http://www.loganalysis.org/). Em qualquer caso, até mesmo ferramentas automatizadas não batem a melhor ferramenta de análise: seu cérebro.

4.12.1 Usando e personalizando o `logcheck`

O pacote `logcheck` no Debian é dividido em três pacotes: `logcheck` (o programa principal), `logcheck-database` (um banco de dados de expressões regulares de um programa) e `logtail` (mostra linhas de logs que ainda não foram lidas). O padrão do Debian (em `/etc/cron.d/logcheck`) é executar o `logcheck` a cada hora e após reinicializações.

Esta ferramenta pode ser muito útil se personalizada adequadamente para alertar ao administrador de eventos estranhos. O `Logcheck` pode ser totalmente personalizado assim enviará mensagens baseadas em eventos encontrados nos logs e passíveis de atenção. A instalação padrão inclui perfis para eventos ignorados e violações de políticas para três diferentes configurações (workstation, server e paranoid). O pacote do Debian inclui um arquivo de configuração `/etc/logcheck/logcheck.conf`, instalado pelo programa, que define que usuário receberá as verificações. Ele também oferece um método para os pacotes que fornecem serviços para implementar novas políticas nos diretórios: `/etc/logcheck/cracking.d/_packagename_`, `/etc/logcheck/violations.d/_packagename_`, `/etc/logcheck/violations.ignore.d/_packagename_`, `/etc/logcheck/ignore.d.paranoid/_packagename_`, `/etc/logcheck/ignore.d.server/_packagename_` e `/etc/logcheck/ignore.d.workstation/_packagename_`. No entanto, são poucos os pacotes que fazem isto. Se tiver uma política que pode ser útil para outros, por favor envie-a como relatório de falha para o pacote apropriado (como um bug *wishlist*). Para mais informações, leia `/usr/share/doc/logcheck/README.Debian`.

O melhor método de configurar o `logcheck` é editar seu arquivo principal de configuração `/etc/logcheck/logcheck.conf` após a instalação. Altere o usuário padrão (root) para quem o relatório deverá ser enviado. Você deverá ajustar o nível de relatório lá também. O pacote `logcheck-database` possui três níveis de relatório para aumentar o detalhamento: workstation, server e paranoid. "server" (servidor) é o nível padrão, paranoid (paranóico) é somente recomendado para máquinas de alta segurança executando poucos

serviços quanto forem possíveis e workstation (estação de trabalho) para máquinas relativamente não críticas. Se desejar adicionar novos arquivos de logs, adicione-os em `/etc/logcheck/logcheck.logfiles`. Ele é ajustado para a instalação padrão do syslog.

Assim que isto for feito, você deverá olhar se os e-mails são enviados, durante os primeiros dias/semanas/meses. Se você achar que estão sendo enviadas mensagens que não deseja receber, apenas adicione as expressões regulares (veja `regex(7)` e `egrep(1)`) que correspondem a estas mensagens em `/etc/logcheck/ignore.d.`*`reportlevel`*`/local`. tente conferir com toda a linha de log. Detalhes sobre como escrever regras estão explicados em `/usr/share/doc/logcheck-database/README.logcheck-database.gz`. É um processo de ajuste fino constante; assim que as mensagens que são enviadas são sempre importantes, você deverá considerar este tunning finalizado. Note que se o `logcheck` não encontrar nada importante em seu sistema, ele não enviará um e-mail para você mesmo se ele for executado (assim se você obtiver somente um e-mail por semana, considere-se uma pessoa de sorte).

4.12.2 Configurando para onde os alertas são enviados

O Debian vem com uma configuração padrão do `syslog` (em `/etc/syslog.conf`) que registra mensagens em arquivos apropriados dependendo da facilidade do sistema. Você deverá estar familiarizado com isto; dê uma olhada no arquivo `syslog.conf` e na documentação caso não estiver registrando. Se você tem a intenção de manter um sistema seguro você deverá se atentar aonde as mensagens de log são enviadas, assim elas não passarão desapercebidas.

Por exemplo, o envio de mensagens para o console também é uma configuração interessante para muitos sistemas a nível de produção. Mas para muitos do sistemas também é importante adicionar uma nova máquina que servirá de servidor de logs (i.e. ela receberá os logs de todos os outros sistemas).

Os e-mails enviados para o root também deverão ser considerados, muitos controles de segurança (como o `snort`) enviam alertas para a caixa de correios do root. Esta caixa de correios normalmente aponta para o primeiro usuário criado no sistema (verifique no `/etc/aliases`). Tenha atenção de enviar as mensagens do root para algum lugar onde sejam lidas (ou localmente ou remotamente).

Existem outras contas e aliases em seu sistema. Em um sistema pequeno, é provavelmente o método mais simples de ter certeza que todos estes aliases apontam para a senha de root, e aquele e-mail do root é redirecionado para a caixa de mensagens pessoal do administrador do sistema.

FIXME: seria interessante em falar como um sistema Debian pode enviar/receber traps SNMP relacionado a problemas de segurança (jfs). Checar: `snmptraglogd`, `snmp` e o `snmpd`.

4.12.3 Usando um servidor de logs

Um servidor de logs é uma máquina que coleta dados do syslog remotamente através da rede. Se uma de suas máquinas for comprometida, o intruso não será capaz de cobrir seus rastros,

a não ser que ataque também o servidor de logs. Assim, esta máquina deverá estar especialmente segura. Fazer uma máquina de loghost é simples. Apenas inicie o `syslogd` com a opção `syslogd -r` e um novo servidor de logs nasce. Para tornar isto permanentemente na Debian, edite o arquivo `/etc/init.d/sysklogd` e adicione a linha

```
SYSLOGD=""
```

to

```
SYSLOGD="-r"
```

Em seguida, configure as outras máquinas para enviar dados para o servidor de logs. Adicione uma entrada como a seguinte no arquivo `/etc/syslog.conf`:

```
facility.level              @your_loghost
```

Veja a documentação sobre o que pode ser usado no lugar de *facility* e *level* (eles não devem ser usados na configuração que foi mostrada). Se quiser registrar tudo remotamente, apenas escreva:

```
*.*                         @your_loghost
```

em seu arquivo `syslog.conf`. O log remoto, assim como o local, é a melhor solução (o intruso pode presumir que cobriu seus rastros após apagar os arquivos de log locais). Veja as páginas de manual `syslog(3)`, `syslogd(8)` e `syslog.conf(5)` para informações adicionais.

4.12.4 Permissões dos arquivos de log

Não só é importante decidir como os alertas são usados, mas também quem tem acesso a leitura/modificação dos arquivos de histórico (caso não estiver usando um servidor de logs remoto). Não é difícil alterar ou desativar os alertas de segurança em um evento de intrusão. Você também deverá levar em conta que os arquivos de histórico podem revelar muitas informações sobre o sistema para um intruso caso ele tenha acesso a eles.

Algumas permissões de arquivos de log não são ideais após a instalação (mas é claro, isto depende da política de segurança local do sistema). Primeiro, os arquivos `/var/log/lastlog` e `/var/log/faillog` não precisam ser lidos por usuários normais. No arquivo `lastlog` você pode ver quem entrou recentemente no sistema e no arquivo `faillog` terá um resumo de logins que falharam. O autor recomenda fazer um `chmod 660` para ambos. De uma breve olhada em seus arquivos de log e decida cuidadosamente que arquivos de logs deverão se tornar legíveis para um usuário com um UID diferente de 0 e um grupo que não sejam 'adm' ou 'root'. Você deverá facilmente verificar isto em seu sistema com:

```
#  find /var/log -type f -exec ls -l {} \; | cut -c 17-35 |sort -u
(procura que usuários os arquivos em /var/log pertencem)
#  find /var/log -type f -exec ls -l {} \; | cut -c 26-34 |sort -u
(procura que grupos os arquivos em /var/log pertencem)
# find /var/log -perm +004
(procura que arquivos são lidos por qualquer usuário)
#  find /var/log \! -group root \! -group adm -exec ls -ld {} \;
(procura por arquivos que não pertencem ao grupo root ou adm)
```

Para personalizar a forma que os arquivos de log são criados, você provavelmente terá que personalizar o programa que os gera. Se os arquivos de log forem rotacionados, no entanto, você poderá personalizar o comportamento do rotacionamento e da criação.

4.13 Adicionando patches no kernel

O Debian GNU/Linux oferece alguns dos patches para o kernel do Linux que aumentam sua segurança. Estes incluem:

- Detecção de Intrusão no Linux (no pacote `lids-2.2.19`), por Huagang Xie e Philippe Biondi. Este patch do kernel torna o processo de fortalecimento do seu sistema Linux uma tarefa fácil permitindo que você restrinja, oculte e proteja processos, até mesmo do usuário root. Ele também permite que proteja ou oculte certos arquivos para que até mesmo o root não possa modificá-los. Adicionalmente, você poderá também definir capacidades para certos processos. Um "máximo" para o administrador de sistema paranóico. Página web http://www.lids.org
- *Listas de Controle de Acessos POSIX (ACLs) para Linux* (no pacote `kernel-patch-acl`). Este patch de kernel adiciona listas de controle de acesso, um método avançado de restringir acesso a arquivos. Ele permite a você um fino controle de acesso a arquivos e diretórios. Este patch foi adicionado ao kernel 2.6. Página do projeto: http://acl.bestbits.at/
- Linux Trustees (no pacote `trustees`). Este patch adiciona um gerenciamento avançado decente de permissões do sistema para o kernel do Linux. Objetos especiais (chamados trustees) são ligados a cada arquivo ou diretório e são armazenados na memória do kernel, permitindo pesquisa rápida de todas as permissões. Homepage: http://trustees.sourceforge.net/
- NSA Enhanced Linux (no pacote `selinux` também disponível de the developer's website (http://www.coker.com.au/selinux/))
- `kernel-patch-2.2.18-openwall`, por Solar Designer. Este contém um conjunto útil de restrições do kernel, como links restritos, FIFOs em `/tmp`, um sistema de arquivos `/proc` restrito, manipulação especial de descritores de arquivos, área não executável de pilha do usuário e outras. Página: http://www.openwall.com/linux/

- `kernel-patch-2.4-grsecurity`: O patch do Grsecurity [15] implementa Controle de Acesso Mandatário, oferece proteção contra estouro de buffer, ACLs, network randomness (para tornar OS fingerprint mais difícil) e muito mais características (http://www.grsecurity.net/features.php).

- `kernel-patch-2.2.19-harden`. *FIXME* Adicionar conteúdo.

- suporte a kernel IPSEC (no pacote `kernel-patch-freeswan`). Se deseja usar o protocolo IPSec com o Linux, você precisará deste patch. Você poderá criar VPNs com este muito facilmente, até em máquinas Windows, pois o IPsec é um padrão comum. As capacidades do IPsec foram adicionadas ao kernel de desenvolvimento 2.5, assim esta característica estará presente por padrão em um kernel 2.6 futuro. Página: http://www.freeswan.org. *FIXME*: Os últimos kernels 2.4 contidos no Debian incluem o backport do código ipsec do kernel 2.5. Comente sobre isto

- `cryptoapi-core-source`. Este patch adiciona capacidades de criptografia do kernel do Linux, como embaralhadores e funções digest. Usos tradicionais para estas funções são a criptografia de sistemas de arquivos ou swap. Note que no kernel 2.5.45, funcionalidades parecidas foram adicionadas ao fonte oficial do kernel do Linux, assim é possível que não precise mais deste patch em um kernel 2.6 futuro *Nota*: este pacote não existe em lançamentos do Debian antes da Sarge (http://www.debian.org/releases/sarge/). Homepage: http://www.kerneli.org/

- `cryptoloop-source`. Este patch lhe permite usar as funções do pacote `cryptoapi-core-source` para criar sistemas de arquivos criptografados usando o dispositivo de loopback.

- `kernel-patch-int`. Este patch também adiciona capacidades criptográficas ao kernel do Linux e foi útil com lançamentos do Debian até a Potato. Ele não funciona com a Woody e se você estiver usando a Sarge ou release mais novo, deverá usar um pacote mais recente do `cryptoapi-core-source`.

FIXME: adicionar mais conteúdo, explicar como estes patches específicos podem ser instalados no Debian usando os pacotes do kernel kernel-2.x.x-patch-XXX.

FIXME: Dividir patches que se aplicam somente nos kernels 2.2, patches que se aplicam nos kernels 2.4 e os que funcionam com ambos.

[15] Note que, dependendo do pacote de fonte do kernel 2.4 que você usar, você poderá encontrar problemas durante o patch de fontes de kernel. Se este for seu caso, você precisa usar o kernel vanilla. Você poderá fazer isto com os seguintes passos:

```
# apt-get install kernel-source-2.4.22 kernel-patch-debian-2.4.22 # tar
xjf /usr/src/kernel-source-2.4.22.tar.bz2 # cd kernel-source-2.4.22 #
/usr/src/kernel-patches/all/2.4.22/unpatch/debian
```

Para mais detalhes veja #194225 (http://bugs.debian.org/194225), #199519 (http://bugs.debian.org/199519), #206458 (http://bugs.debian.org/206458), #203759 (http://bugs.debian.org/203759), #204424 (http://bugs.debian.org/204424), #210762 (http://bugs.debian.org/210762), #211213 (http://bugs.debian.org/211213), e discussion at debian-devel (http://lists.debian.org/debian-devel/2003/debian-devel-200309/msg01133.html)

No entanto, alguns patches ainda não foram adicionados ainda no Debian. Se sentir que alguns destes devem ser incluídos, por favor pergunte por ele em Work Needing and Prospective Packages (http://www.debian.org/devel/wnpp/). Alguns destes pacotes são:

- patch do PaX (http://pageexec.virtualave.net/)
- patch HAP (http://www.theaimsgroup.com/~hlein/hap-linux/)
- Patch Stealth (http://www.energymech.net/madcamel/fm/)
- *SubDomain*. Uma extensão do kernel feita para oferecer confinamento com poucas permissões para programas possivelmente inseguros. Complemento de subdomínio e extensão para controle de acesso nativo. Enquanto é similar ao ambiente `chroot`, ele clama ser de fácil construção e mais flexível que um ambiente `chroot`.
- *Contexts (ctx) patch*. Uma extensão do kernel feita para implementar servidores privados virtuais. É parecido com o `jail` no BSD. Homepage: http://www.immunix.org/subdomain.html
- *UserIPAcct*. Não é um patch realmente relacionado a segurança, mas ele lhe permite criar quotas de tráfego por usuário em seu sistema. Você também pode obter estatísticas sobre o tráfego de usuário. Homepage: http://ramses.smeyers.be/useripacct.

4.14 Protegendo-se contra estouros de buffer

O *estouro de buffer (buffer overflow)* é o nome de um comum ataque a softwares [16] que faz o uso de checagem insuficiente de limites (um erro de programação, mais comum na linguagem C) para executar o código de máquina através de entrada de programas. Estes ataques, contra programas de servidores que escutam conexões remotamente ou contra softwares locais que garantem altos privilégios aos usuários (`setuid` ou `setgid`) podem resultar no comprometimento de qualquer sistema determinado.

Existem basicamente quatro métodos de se proteger contra estouro de buffer:

- aplicar um patch no kernel para prevenir a execução da pilha (você pode usar os patches OpenWall ou Grsecurity)
- usar uma biblioteca, tal como a libsafe (http://www.research.avayalabs.com/project/libsafe/), para substituir funções vulneráveis e introduzir a checagem apropriada (para informações sobre como instalar a `libsafe` leia isto (http://www.Linux-Sec.net/harden/libsafe.uhow2.txt)).
- corrigir o código fonte usando ferramentas para encontrar fragmentos de onde pode introduzir esta vulnerabilidade.

[16] Tão comum, de fato, que eles são a base de 20% de vulnerabilidades reportadas de segurança todo ano, como determinado pelas estatísticas do banco de dados dde vulnerabilidades ICAT's (http://icat.nist.gov/icat.cfm?function=statistics)

- recompilar o código fonte para adicionar checagens apropriadas que previnem buffer overflows, usando, por exemplo, StackGuard (http://www.immunix.org/stackguard.html) (que é usado pelo Immunix (http://www.immunix.org)) ou o patch Stack Smashing Protector (SSP) (http://www.research.ibm.com/trl/projects/security/ssp/) para o GCC (que é usado pelo Adamantix (http://www.adamantix.org))

O Debian GNU/Linux em seu lançamento 3.0, fornece software para introduzir todos estes métodos exceto a proteção de compilação do código fonte (mas isto foi requisitado no Bug #213994 (http://bugs.debian.org/213994))

Note que até mesmo se o Debian fornecer um compilador que possua proteção contra estouro de pilha/buffer, todos os pacotes precisariam ser recompilados para introduzir esta característica. Isto é, de fato, o que o Adamantix faz (entre outras características). O feito desta nova característica na estabilidade do software é algo que deverá ser determinado (alguns programas ou arquiteturas de processador podem ter problemas com seu uso).

Em qualquer caso, esteja alerta que até mesmo estas alternativas podem não prevenir buffer overflows, pois existem formas de burlá-los, como descrito na revista phrack's issue 58 (http://packetstorm.linuxsecurity.com/mag/phrack/phrack58.tar.gz) ou no aviso CORE's Múltiplas vulnerabilidades nas tecnologias de proteção de pilha (http://onlne.securityfocus.com/archive/1/269246).

4.14.1 Patches de kernel para proteção contra estouros de buffer

Os patches do kernel relacionados a estouro de buffer incluem o patch Openwall que oferece proteção contra buffer overflows nos kernels do Linux 2.2. Para kernels 2.4 ou superiores, utilize o patch Grsecurity (existente no pacote `kernel-patch-2.4-grsecurity`) que inclui o patch do Openwall e muito mais características (http://www.grsecurity.net/features.php) (incluindo ACLs e métodos de rede que dificultam a realização de OS fingerprinting), ou os módulos de Segurança do Linux (nos pacotes `kernel-patch-2.4-lsm` e `kernel-patch-2.5-lsm`). Para mais informações sobre como usar estes patch leia a seção 'Adicionando patches no kernel' on page 64.

4.14.2 Proteção da `Libsafe`

A proteção do sistema Debian GNU/Linux com a `libsafe` é bastante fácil, apenas instale o pacote e diga *Sim* para ter a biblioteca pré carregada globalmente. Tenha cuidado, no entanto, pois isto pode quebrar alguns programas (notavelmente, programas compilados usando a antiga `libc5`, assim tenha certeza de ler os relatórios de falhas reportadas (http://bugs.debian.org/libsafe) antes e testar os programas mais críticos em seu sistema com o programa `libsafe`.

Nota Importante: A proteção da `Libsafe` pode não ser efetiva atualmente como descrito em 173227 (http://bugs.debian.org/173227). Considere testá-la antes de usá-la em um ambiente de produção e não dependa exclusivamente dela para proteger seu sistema.

4.14.3 Testando problemas de estouro em programas

O uso de ferramentas para detecção de estouro de buffer requer, em qualquer caso, conhecimento de programação para corrigir (e recompilar) o código. O Debian contém, por exemplo: `bfbtester` (um verificar de estouro de buffer que faz ataques de força bruta em binários e estouro de ambiente) e o `njamd`. Outros pacotes de interesse podem também ser o `rats`, `pscan`, `flawfinder` e o `splint`.

4.15 Transferência segura de arquivos

Durante a administração normal do sistema, sempre são necessárias transferências de arquivos de um sistema para outro. A cópia de arquivos de maneira segura de um sistema para outro pode ser feita usando o pacote do servidor `sshd`. Outra possibilidade é usar o `ftpd-ssl`, um servidor ftp que faz uso da *Camada de Conexões Seguras* para encriptar as transmissões.

Qualquer um destes métodos precisam de clientes especiais. O Debian fornece programas clientes, como `scp` no pacote `ssh`, que trabalha como o `rcp` mas é completamente criptografada, assim os *maus meninos* não poderão nem saber O QUE você copia. Também existe o pacote `ftp-ssl` para o servidor equivalente. Você poderá encontrar clientes para estes softwares até mesmo para outros sistemas operacionais (não-UNIX), o `putty` e o `winscp` fornecem implementações de cópia segura para qualquer versão do sistema operacional da Microsoft.

Note que o uso de `scp` fornece acesso dos usuários a todos os arquivos do sistema a não ser que esteja dentro de um `chroot` como descrito em 'Executando o ssh em uma jaula chroot' on page 82. O acesso FTP pode ser feito usando `chroot`, possivelmente mais fácil dependendo do daemon escolhido, como descrito em 'Tornando o FTP mais seguro' on page 84. Se está preocupado sobre usuários navegando em seus arquivos locais e deseja ter comunicação encriptada, você poderá usar um daemon FTP com suporte a SSL ou combinar ftp texto plano com uma configuração de VPN (veja 'Redes Privadas Virtuais (VPN)' on page 134).

4.16 Limitações e controle do sistema de arquivos

4.16.1 Usando quotas

É importante se ter uma boa política de quotas, pois ela evita que os usuários ocupem todo o(s) disco(s) rígido(s).

Você poderá usar dois sistemas diferentes de quota: quota do usuário e quota do grupo. Você provavelmente notará que limites de quota de usuários definem o espaço que o usuário pode utilizar, a quota de grupo é equivalente para grupos. Mantenha isto em mente quando estiver trabalhando com tamanhos de quota.

Existem alguns pontos importantes que devem ser pensados sobre a configuração de um sistema de quotas:

- Mantenha as quotas suficientemente pequenas, assim os usuários não poderão acabar com todo seu espaço em disco.
- Mantenha as quotas grande o bastante, assim os usuarios não se importarão ou sua quota de e-mails os proibirá de receber mensagens por um longo período.
- Use quotas em todas as áreas graváveis por usuários, em `/home` como também em `/tmp`.

Cada partição ou diretório no qual os usuários tem acesso completo a gravação deverão ter a quota ativada. Calcule e defina um tamanho de quota funcional para estas partições e diretórios que combinam utilização e segurança.

Assim, você deseja usar quotas. A primeira coisa que precisa checar, é se ativou o suporte a quotas em seu kernel. Se não ativou, você terá que recompilá-lo. Após isto, verifique se o pacote `quota` está instalado. Caso negativo, você terá que instalá-lo.

Ativar as quotas para um respectivo sistema de arquivos é muito fácil, bastando modificar as configurações de `defaults` para `defaults,usrquota` em seu arquivo `/etc/fstab`. Se você precisar de quota de grupo, substitua `usrquota` por `grpquota`. Você também poderá usar ambos. Então crie os arquivos vazios `quota.user` e `quota.group` no raiz do sistema de arquivos que deseja ativar as quotas (e.g. `touch /home/quota.user /home/quota.group`, para um sistema de arquivos `/home`).

Reinicie o sistema de `quota` executando `/etc/init.d/quota stop;/etc/init.d/quota start`. Agora o sistema de quotas deverá estar funcionando e os tamanhos de quotas poderão ser definidos.

A edição de quotas de um usuário específico poderá ser feita através de `edquota -u <user>`. As quotas de grupos podem ser modificadas com `edquota -g <group>`. Então ajuste a quota soft e hard e/ou quotas de inodes se necessário.

Para mais detalhes sobre quotas, leia a página de manual do quota, e o mini-howto do quota (`/usr/share/doc/HOWTO/en-html/mini/Quota.html`).

Você pode ou não gostar do `lshell`, pois ele viola a FHS. Também tenha em mente que o `pam_limits.so` pode fornecer a mesma funcionalidade e `lshell` está atualmente orfanado (http://bugs.debian.org/93894)

4.16.2 Os atributos específicos do sistema de arquivos ext2 (chattr/lsattr)

Em adição as permissões atuais do Unix, os sistemas de arquivos ext2 e ext3 oferecem um conjunto de atributos específicos que lhe dão mais controle sobre os arquivos em seu sistema. De forma contrária a permissões básicas, estes atributos não são mostrados com o tradicional comando `ls -l` ou alterados usando-se o `chmod`, e você precisará de dois utilitários diferentes, o `lsattr` e o `chattr` (que estão no pacote `e2fsprogs`) para gerênciá-los. Note que isto significa que estes atributos normalmente não serão salvos quando fizer o backup do seu sistema, assim se alterar qualquer um deles, será um tormento salvar comandos `chattr` sucessivos em um script que será usado depois de ter restaurado o backup.

Entre todos os atributos disponíveis, os dois abaixo são os mais importantes para aumentar a segurança e são referenciados pelas letras 'i' e 'a' e podem ser somente definidos (ou removidos) pelo superusuário:

- O atributo 'i' ('imutável'): um arquivo com este atributo não pode ser modificado, excluído ou renomeado, e nenhum link poderá ser criado para ele, até mesmo pelo superusuário.
- O atributo 'a' ('incremental'): este atributo tem o mesmo efeito do atributo imutável, exceto que você ainda poderá abrir o arquivo em modo incremental. Isto significa que você poderá adicionar mais conteúdo a ele, mas será impossível modificar o conteúdo anterior. Este atributo é especialmente útil para arquivos de log armazenados em `/var/log/`, assim você deverá considerar que eles serão movidos sempre devido aos scripts de rotacionamento de logs.

Estes atributos também podem ser definidos para diretórios, neste caso ninguém terá o direito de modificar o conteúdo de um diretório (eg. renomear ou excluir um arquivo, ...). Quando aplicado a um diretório, o atributo incremental permite somente a criação de arquivos.

É fácil ver porque o atributo 'a' aumenta a segurança, dando a programas que não estão rodando sob o superusuário a capacidade de adicionar dados a um arquivo sem modificar seu conteúdo anterior. Por outro lado, o atributo 'i' parece ser menos interessante: depois de tudo, somente o superusuário poderá usar as permissões básicas do Unix para restringir o acesso a um arquivo, e um intruso que teria acesso a uma conta de superusuário poderia sempre usar o programa `chattr` para remover o atributo. Tal intruso ficara primeiramente confuso quando se ver não ser capaz de remover um arquivo, mas ele deverão não assumir que ele está blindado - acima de tudo, ele entrou no seu sistema! Alguns manuais (incluindo a versão anterior deste documento) sugerem remover os programas `chattr` e `lsattr` do sistema para aumentar a segurança, mas este tipo de estratégia, conhecida também por "segurança pela obscuridade", deve ser absolutamente evitada, pois ela fornece uma falsa sensação de segurança.

Um método de resolver isto é usar as capacidades do kernel do Linux, como descrito em 'Defesa pró-ativa' on page 149. A capacidade de interesse aqui é chamada `CAP_LINUX_IMMUTABLE`: se removê-la do conjunto de capacidades (usando por exemplo,o comando `lcap CAP_LINUX_IMMUTABLE`) não será possível alterar qualquer atributo 'a' ou 'i' em seu sistema, até mesmo pelo superusuário! Uma estratégia completa pode ser a seguinte:

1 Defina os atributos 'a' e 'i' nos arquivos que deseja;

2 Execute o comando `lcap CAP_LINUX_IMMUTABLE` (também como `lcap CAP_SYS_MODULE`, como sugerido em 'Defesa pró-ativa' on page 149) a um dos scripts de inicialização;

3 Defina o atributo 'i' neste script e em outros arquivos de inicialização, assim também como no próprio binário `lcap`;

4 Execute manualmente o comando acima (ou reinicie o seu sistema para ter certeza que tudo funciona como planejado).

Agora que a capacidade foi removida do seu sistema, um intruso não poderá alterar qualquer atributo em arquivos protegidos, e assim não poderá alterar ou excluir os arquivos. Se ele forçar a máquina a reiniciar (que é o único método de restaurar o conjunto de capacidades), ele será facilmente detectado, e a capacidade será removida novamente assim que o sistema for reiniciado. O único método de alterar um arquivo protegido seria inicializar o sistema em modo monousuário ou usar outro disco de inicialização. Duas operações que requerem acesso físico a máquina!

4.16.3 Verificando a integridade do sistema de arquivos

Você tem certeza que o `/bin/login` em seu disco rígido é ainda o binário que instalou alguns meses atrás? Se ele for uma versão hackeada, que armazena a senha que digitou em um arquivo oculto ou o envia por e-mails em texto plano através da Internet?

O único método que tem algum tipo de proteção é verificar seus arquivos a cada hora/dia/mês (eu prefiro diariamente) comparando o md5 do atual e do antigo. Dois arquivos nunca têm o mesmo md5sum (o digest do MD5 é de 128 bits, assim a chance de arquivos terem o mesmo md5sum é 3.4e3803), assim, você está do lado seguro aqui, a não ser que alguém tenha hackeado o algoritmo que cria md5sums em sua máquina. Isto é, bem, extremamente difícil e muito improvável. Você realmente deverá considerar esta auditoria de seus binários como muito importante, pois é um método fácil de reconhecer alterações. Ferramentas padrões usadas para isto são `sXid`, `AIDE` (Ambiente Avançado de Detecção de Intrusões), `TripWire`, `integrit` e `samhain`.

A instalação do `debsums` ajudará a verificar a integridade do sistema de arquivos, comparando o md5sum de cada arquivo com o md5sum usado no arquivo de pacotes do Debian. Tenha cuidado, estes arquivos podem ser facilmente alterados.

Você pode querer usar o `locate` para indexar todo o sistema de arquivos, se fizer isto, considere as implicações disto. O pacote `locate` no Debian é executado como usuário nobody, e assim ele somente indexa arquivos que são visíveis para todos. No entanto, se você alterar seu comportamento, você tornará todas as localizações de arquivos visíveis para todos os usuários. Se deseja indexar todo o sistema de arquivos (não os poucos que o usuário nobody pode ver) você poderá substituir o `locate` pelo `slocate`. O slocate tem a etiqueta de uma versão avançada e segura do locate da GNU, mas ele atualmente fornece funcionalidade adicional de localização de arquivos. Quando usar o `slocate`, o usuário somente verá os arquivos que ele tem acesso e você poderá ignorar qualquer arquivo ou diretório no sistema. O pacote `slocate` executa seus privilégios de atualização com altos privilégios se comparado ao locate e indexa cada arquivo. Os usuários são então capazes de localizar rapidamente cada arquivo que podem ver. O `slocate` não lhes permitem ver novos arquivos; ele faz a filtragem da saída baseado em sua UID.

FIXME: colocar referências ao snapshot feito após a instalação.

FIXME: Adicionar uma nota com relação a pacotes que não fornecem debsums de aplicativos instalados (não mandatário).

FIXME: Mencionar binários assinados usando digamos, bsign ou elfsign

4.16.4 Configurando verificação de setuid

O Debian oferece um trabalho do `cron` que é executado diariamente no arquivo `/etc/cron.daily/standard`. Esta tarefa do `cron` executará o script `/usr/sbin/checksecurity` que armazena informações destas alterações.

Para esta verificação ser feita, você deverá definir `CHECKSECURITY_DISABLE="FALSE"` no `/etc/checksecurity.conf`. Note que, este é o padrão, assim a não ser que tenha alterado algo, esta opção já estará definida para "FALSE".

O comportamento padrão é não enviar esta mensagem para o superusuário, mas ao invés disto manter cópias diárias das alterações em `/var/log/setuid.changes`. Você deverá alterar o CHECKSECURITY_EMAIL (no `/etc/checksecurity.conf`) para 'root' para ter estes dados enviados por e-mail para ele. Veja `checksecurity(8)` para mais detalhes.

4.17 Tornando o acesso a rede mais seguro

FIXME. Necessário mais conteúdo (específico o Debian)

4.17.1 Configurando características de rede do kernel

FIXME: Faltando conteúdo

Muitas características do kernel podem ser modificadas usando comandos echo no sistema de arquivos `/proc` ou usando o `sysctl`. Executando o `/sbin/sysctl -A` você poderá ver o que pode ser configurado e que opções existem, e elas podem ser modificadas executando `/sbin/sysctl -w variável=valor` (veja `sysctl(8)`). Somente em raros casos você precisará editar algo aqui, mas você poderá aumentar também a segurança desta forma. Por exemplo:

```
net/ipv4/icmp_echo_ignore_broadcasts = 1
```

Este é um *emulador de Windows* pois ele atua como o Windows em ping broadcast caso esta opção seja ajustada para 1. Que é, requisições ICMP_ECHO enviadas para o endereço de broadcast serão ignoradas. Caso contrário, ela não faz nada.

Se quer evitar que o seu sistema responda requisições ICMP, apenas ative esta opção de configuração:

```
net/ipv4/icmp_echo_ignore_all = 1
```

Para registrar pacotes com endereços impossíveis (devido a roteamento incorreto) em seu sistema, use:

```
/proc/sys/net/ipv4/conf/all/log_martians = 1
```

Para mais informações sobre que coisas podem ser feitas com `/proc/sys/net/ipv4/*` leia `/usr/src/linux/Documentation/filesystems/proc.txt`. Todas as opções estão descritas através do `/usr/src/linux/Documentation/networking/ip-sysctl.txt` [17].

4.17.2 Configurando Syncookies

Esta opção é uma faca de dois gumes. De um lado ela protege o seu sistema contra flood de pacotes syn; por outro lado ela viola os padrões definidos (RFCs).

```
net/ipv4/tcp_syncookies = 1
```

Se deseja alterar esta opção cada vez que o kernel estiver funcionando, você precisará alterá-la em `/etc/network/options` definindo `syncookies=yes`. Ela fará efeito sempre quando `/etc/init.d/networking` for executado (que é tipicamente feito durante a inicialização do sistema) enquanto o seguinte comando fará efeito imediatamente até a reinicialização:

```
echo 1 > /proc/sys/net/ipv4/tcp_syncookies
```

Esta opção somente estará disponível caso o kernel tenha sido compilado com a opção `CONFIG_SYNCOOKIES`. Todos os kernels do Debian são compilados com esta opção embutida, mas você poderá verificá-la executando:

```
$ sysctl -A |grep syncookies
net/ipv4/tcp_syncookies = 1
```

Para mais informações sobre os syncookies TCP, leia http://cr.yp.to/syncookies.html.

4.17.3 Tornando a rede segura em tempo de inicialização

Quando definir opções de configuração do kernel para a rede, você precisará configurá-la de forma que seja carregada sempre que o sistema for iniciado. O seguinte exemplo ativa muitas das opções anteriores assim como outras opções úteis.

FIXME Ao invés de fornecer este script, fornecer uma configuração modelo para o `sysctl.conf` (veja: `sysctl.conf(5)`). Também envie isto como um bug wishlist para o pacote.

Crie um script em `/etc/network/interface-secure` (o nome é dado como um exemplo) e o execute do arquivo `/etc/network/interfaces` desta forma:

[17] No Debian o pacote `kernel-image` instala o fonte sob `/usr/src/kernel-source-2.X.X`, apenas substitua *linux* com o tipo de kernel que está instalado

```
auto eth0
iface eth0 inet static
        address xxx.xxx.xxx.xxx
        netmask 255.255.255.xxx
        broadcast xxx.xxx.xxx.xxx
        gateway xxx.xxx.xxx.xxx
        pre-up /etc/network/interface-secure


#!/bin/sh
# Nome do Script: /etc/network/interface-secure
# Modifica o comportamento padrão para tornar o sistema seguro contra
# alguns tipos de ataques TCP/IP spoofing
# some TCP/IP spoofing & attacks
#
# Contribuído por Dariusz Puchalak
#
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
                                        # proteção contra broadcast de ECH
echo 0 > /proc/sys/net/ipv4/ip_forward     # desativação de forward de ip
echo 1 > /proc/sys/net/ipv4/tcp_syncookies # Proteção contra syn cookies ativ
echo 1 >/proc/sys/net/ipv4/conf/all/log_martians # Registra pacotes estranhos
# (isto inclui pacotes falsos, pacotes com a rota de origem alterada e pacote
redirecionados)
                          # mas tenha cuidado com isto em servidores web carre
echo 1 > /proc/sys/net/ipv4/ip_always_defrag
                                        #  opção de desfragmentação sempre
echo 1 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
                                        # proteção ativada contra mensagen

# proteção contra ip spoofing
echo 1 > /proc/sys/net/ipv4/conf/all/rp_filter

# e finalmente mais coisas:
# Não aceita redirecionamento de ICMP
echo 0 > /proc/sys/net/ipv4/conf/all/accept_redirects
echo 0 > /proc/sys/net/ipv4/conf/all/send_redirects

# Desativa pacotes com rota de origem
echo 0 > /proc/sys/net/ipv4/conf/all/accept_source_route

echo 1 > /proc/sys/net/ipv4/conf/all/log_martians
```

Você também poderá criar um script em `init.d` que é executado na inicialização (usando o `update-rc.d` para criar os links apropriados em `rc.d`).

4.17.4 Configurando características do firewall

Para ter capacidades de firewall, ou para proteger o sistema local ou outros *atrás* dele, o kernel precisa ser compilado com capacidades de firewall. O kernel padrão do Debian 2.2 (também 2.2) fornece o filtro de pacotes chamado `ipchains`, o Debian 3.0 usando o kernel padrão (kernel 2.4) oferece um filtro de pacotes de *estado* chamado `iptables` (netfilter). As distribuições antigas do Debian precisarão de um patch apropriado no kernel (o Debian 2.1 usa o kernel 2.0.34).

De qualquer forma, é muito fácil usar um kernel diferente do fornecido pelo Debian. Você poderá encontrar um kernel pré-compilado como pacotes que poderão facilmente instalar em seu sistema Debian. Você também poderá copiar os fontes do kernel usando os pacotes `kernel-source-X` e construir pacotes de kernel personalizados usando o `make-kpkg`.

A configuração de firewalls no Debian é discutida mais precisamente em 'Adicionando capacidades de firewall' on page 103.

4.17.5 Desativando assuntos relacionados a weak-end de máquinas

Sistemas com mais de uma interface de rede em diferentes redes podem ter os serviços configurados de forma que escutem somente a um determinado endereço IP. Isto normalmente evita o acesso a serviços quando são requisistados através de qualquer outro endereço. No entanto, isto não significa (que foi até mesma uma concepção correta que tive) que o serviço é oferecido ao endereço de *hardware* (interface de rede). [18]

Isto não é um assunto relacionado a ARP e não é uma violação da RFC (isto é chamado *máquina weak end* na RFC1122 (`ftp://ftp.isi.edu/in-notes/rfc1122.txt`), seção 3.3.4.2). Lembre-se, endereços IP não tem nada a ver com a interface física.

Nos kernels da série 2.2 (e anteriores) isto pode ser corrigido com:

```
# echo 1 > /proc/sys/net/ipv4/conf/all/hidden
# echo 1 > /proc/sys/net/ipv4/conf/eth0/hidden
# echo 1 > /proc/sys/net/ipv4/conf/eth1/hidden
.....
```

Em outros kernels, isto pode ser corrigido com uma das alternativas:

- regras do iptables.

[18] Para reproduzir isto (exemplo oferecido por Felix von Leitner na lista de discussão bugtraq):

```
máquina A (eth0 conectada a eth0 da máquina B): ifconfig eth0 10.0.0.1 ifconfig eth1
23.0.0.1 tcpserver -RHl localhost 23.0.0.1 8000 echo fnord máquina B: ifconfig eth0
10.0.0.2 route add 23.0.0.1 gw 10.0.0.1 telnet 23.0.0.1 8000
```

Parece, no entanto, não funcionar com serviços funcionando na interface 127.0.0.1, você precisará fazer os testes usando soquetes simples.

- roteamento corretamente configurado. [19]
- Patch do kernel [20]

Junto com este texto, existirão algumas ocasiões em que será mostrado como configurar alguns serviços (servidor sshd, apache, serviço de impressão...) para tê-los escutando em um determinado endereço, o leitor deverá ter em mente que, sem as correções fornecidas aqui, a correção não evitará acesso de dentro do mesmo segmento de rede (local). [21]

FIXME: os comentários na bugtraq indicam que lá existe um método específico do Linux para escutar em uma determinada interface.

FIXME: Enviar um bug contra o netbase, assim a correção de roteamento será o comportamento padrão no Debian?

4.17.6 Protegendo-se contra ataques ARP

Quando não confia em outras máquinas na sua rede (que deve sempre ser o caso, por ser uma atitude mais segura) você deverá proteger a si mesmo de vários ataques ARP existentes.

Como deve saber, o protocolo ARP é usado para ligar endereços IP a endereços MAC. (veja RFC826 (`ftp://ftp.isi.edu/in-notes/rfc826.txt`) para todos os detalhes). Cada vez que enviar um pacote para um endereço IP uma resolução arp é feita (primeiro procurando no cache ARP local, então se o endereço IP não estiver presente no cache faz o broadcast de uma requisição arp) para encontrar o endereço de hardware alvo. Todos os pacotes ARP tentam deixar sua máquina ingênua fazendo-a pensar que o endereço IP da máquina B é associado com o endereço MAC da máquina do invasor. Então cada pacote que deseja enviar para o endereço IP associado com a máquina B, será enviado para a máquina do invasor.

Estes ataques (envenenamento e cache, falsificação ARP...) permitem ao invasor capturar o tráfego até mesmo em redes com switches, para facilmente roubar conexões, para desconectar qualquer máquina da rede... ataques arp são poderosos e fáceis de serem implementados, e existem diversas ferramentas, tais como `arpspoof` através do pacote `dsniff`.

No entanto, sempre existe uma solução:

- Use um cache arp estático. Você pode configurar entradas "estáticas" em seu cache arp:

[19] O fato deste comportamento ser alterado através do roteamento foi descrito por Matthew G. Marsh na thread do bugtraq:

```
eth0 = 1.1.1.1/24 eth1 = 2.2.2.2/24 ip rule add from 1.1.1.1/32 dev lo table 1 prio
15000 ip rule add from 2.2.2.2/32 dev lo table 2 prio 16000 ip route add default dev
eth0 table 1 ip route add default dev eth1 table 2
```

[20] Existem alguns patches disponíveis para este comportamento como descrito na discussão do bugtraq em `http://www.linuxvirtualserver.org/~julian/#hidden` e `http://www.fefe.de/linux-eth-forwarding.diff`.

[21] Um invasor pode ter muitos problemas para contornar o acesso através da escuta de endereços IP se ele não está no mesmo domínio de broadcast (mesma rede) que a máquina que será atacada. Se a invasão for através do roteador, será bastante difícil as repostas retornarem a algum lugar.

```
arp -s host_name hdwr_addr
```

Configurando entradas estáticas para cada máquina importante em sua rede você se assegura de que ninguém poderá criar/modificar uma entrada (falsa) para estas máquinas (entradas estáticas não expiram e não podem ser modificadas) e respostas arp falsificadas serão ignoradas.

- Detectar tráfego ARP suspeito. Você poderá usar o pacote `arpwatch`, `karpski` ou ferramentas IDS mais gerais que também poderão detectar tráfego arp suspeitos como (`snort`, prelude (http://www.prelude-ids.org)...).
- Implementando filtragem na validação de tráfego IP no endereço MAC.

4.18 Fazendo um snapshot do sistema

Antes de por o sistema em produção você deverá tirar um snapshot de todo o sistema. Este snapshot deverá ser usado em um evento de compromisso (veja 'Depois do comprometimento do sistema (resposta a incidentes)' on page 155). Você deverá refazer este upgrade assim que o sistema for atualizado, especialmente se seu upgrade for para uma novo lançamento do Debian.

Para isto você deverá usar uma mídia removível gravável que poderá ser configurada como somente-leitura, isto poderá ser feito em um disquete (proteja como somente leitura após o uso) ou uma unidade de CD-ROM (você poderá usar um CD-ROM regravável assim poderá até mesmo manter backups de md5sums em diferentes datas).

O seguinte script criará o snapshot:

```
#!/bin/bash
/bin/mount /dev/fd0 /mnt/floppy
/bin/cp /usr/bin/md5sum /mnt/floppy
echo "Calculando banco de dados md5"
>/mnt/floppy/md5checksums.txt
for dir in /bin/ /sbin/ /usr/bin/ /usr/sbin/ /lib/ /usr/lib/
do
   find $dir -type f | xargs /usr/bin/md5sum >>/mnt/floppy/md5checksums-lib.t
done
/bin/umount /dev/fd0
echo "Pós instalação do banco de dados md5 calculada"
```

Note que o binário md5sum é colocado em uma unidade de disquetes assim ele poderá ser usado depois para verificar binários no sistema (como no caso de ser atacado por um trojan).

O snapshot não inclui os arquivos sob `/var/lib/dpkg/info` que incluem os hashs md5 de pacotes instalados (em arquivos que finalizam com `.md5sums`). Você poderá copiar esta informação também, no entanto você deverá saber:

- os md5sums fornecidos por pacotes do Debian incluem todos os arquivos fornecidos por eles, que torna o banco de dados grande (5 MB contra 600Kb em um sistema Debian GNU/Linux com um sistema gráfico e com aproximadamente 2.5 Gb de programas instalados)
- nem todos os pacotes do Debian contém md5sums de arquivos que foram instalados pois esta não é (atualmente) a política mandatória.

Assim que o snapshot for feito você deverá se assegurar de proteger a mídia como somente leitura. Você poderá então armazená-la para backup ou colocá-la na unidade e usá-la fazendo uma verificação com o `cron` toda a noite comparando os md5sums originais com estes no snapshot.

4.19 Outras recomendações

4.19.1 Não use programas que dependem da svgalib

A Svgalib é muito bonita para amantes de console, como eu, mas no passado ela provou diversas vezes que é muito insegura. Foram lançadas explorações de vulnerabilidades contra o `zgv` e era simples se tornar usuário root. Tente evitar o uso de programas usando Svgalib sempre que possível.

Capítulo 5

Tornando os serviços em execução do seu sistema mais seguros

Os serviços podem ser deixados mais seguros de duas formas:

- Tornando-os somente acessíveis em pontos de acessos (interfaces) que são utilizados.
- Configurando-os adequadamente, desta forma eles poderão somente ser usados por usuários legítimos de forma autorizada.

A restrição de serviços de forma que possam somente ser acessados de um determinado lugar pode ser feito restringindo o acesso a eles no nível de kernel (i.e. firewall), configure-os para operar somente em interfaces definidas (alguns serviços podem não ter esta característica) ou usando algum outro método, por exemplo o patch vserver do Linux (para 2.4.16) pode ser usado para forçar o kernel a utilizar somente uma interface de rede.

Com relação a serviços sendo executados a partir do `inetd` (`telnet`, `ftp`, `finger`, `pop3`...) é importante notar que o `inetd` pode ser configurado para que os serviços somente executem em uma interface definida (usando a sintaxe `serviço@ip`) mas esta é uma característica não documentada. Um de seus substitutos, o meta-daemon `xinetd` inclui uma opção chamada `bind` apenas para controlar este comportamento. Veja `xinetd.conf(5)`.

```
service nntp
{
        socket_type     = stream
        protocol        = tcp
        wait            = no
        user            = news
        group           = news
        server          = /usr/bin/env
        server_args     = POSTING_OK=1 PATH=/usr/sbin/:/usr/bin:/sbin/:/bin
+/usr/sbin/snntpd logger -p news.info
```

```
        bind            = 127.0.0.1
}
```

As seguintes seções detalham como alguns serviços individuais podem ser configurados adequadamente conforme sua utilização.

5.1 Tornando o ssh mais seguro

Caso ainda estiver usando o telnet ao invés do ssh, você deverá dar uma parada na leitura deste manual e alterar isto. O ssh deve ser usado para qualquer login remoto ao invés do telnet. Em uma era onde é fácil capturar o tráfego que circula na internet e obter senhas em texto plano, você deverá usar somente protocolos que utilizam criptografia. Assim, execute um `apt-get install ssh` agora em seu sistema.

Encoraje todos os usuários em seu sistema para utilizarem o ssh ao invés do telnet, ou até mesmo melhor, remova o telnet/telnetd. Em adição, você deverá evitar entrar no sistema usando o ssh como usuário root e ao invés disto, usar métodos alternativos para se tornar o root, como o `su` ou `sudo`. Finalmente, o arquivo `sshd_config` no diretório `/etc/ssh`, também deverá ser modificado para aumentar a segurança:

- `ListenAddress 192.168.0.1`

  Especifica que o ssh somente funcionará na interface especificada, caso tenha mais de uma interface (e não deseja que o ssh funcione através delas) ou em caso de adição de uma futura interface de rede (onde não deseja receber conexões ssh através dela).

- `PermitRootLogin no`

  Tenta não permitir o login do usuário Root sempre que possível. Se alguém quiser se tornar o usuário root usando ssh, agora dois logins são necessários e o ataque de força bruta não terá efeito no root via SSH.

- `Listen 666`

  Altera a porta do programa, assim o intruso não terá completa certeza de onde o daemon sshd é executado (esteja avisado, isto é segurança por obscuridade).

- `PermitEmptyPasswords no`

  Senhas em branco tornam a segurança do seu sistema um fiasco.

- `AllowUsers alex ref me@algumlugar`

  Permite somente certos usuários terão acesso via ssh a esta maquina. `usuario@maquina` pode também ser usado para restringir um determinado usuário de acessar somente através de uma maquina especificada.

- `AllowGroups wheel admin`

Permite somente membros de certos grupos de terem acesso ao ssh nesta maquina. AllowGroups e AllowUsers possuem diretivas equivalentes para bloquear o acesso a maquina. Não se surpreenda por eles serem chamados de "DenyUsers" e "DenyGroups".

- `PasswordAuthentication yes`

  Esta escolha fica completamente por sua conta. É mais seguro somente permitir o acesso a maquina de usuários com chaves ssh colocadas em `~/.ssh/authorized_keys`. Se deseja isto, ajuste esta opção para "no".

- Desative quaisquer outras formas de autenticação que realmente não precisa, se não usar, por exemplo `RhostsRSAAuthentication`, `HostbasedAuthentication`, `KerberosAuthentication` ou `RhostsAuthentication`, você deverá desativa-las, até mesmo se forem usadas por padrão (veja a página de manual `sshd_config(5)`).

- `Protocol 2`

  Desative o protocolo da versão 1, pois ele tem alguns problemas de design que torna fácil a descoberta de senhas. Para mais informações leia documento relacionando problemas do protocolo ssh (http://earthops.net/ssh-timing.pdf) ou o aviso Xforce (http://xforce.iss.net/static/6449.php).

- `Banner /etc/some_file`

  Adiciona um banner (ele será lido de um arquivo) para usuários se conectando ao servidor ssh, em alguns países o envio de avisos antes de acessar um determinado sistema alertando sobre acesso não autorizado ou monitoramento de usuários deverá ser emitido para ter proteção legal.

Você também poderá restringir o acesso ao servidor ssh usando o `pam_listfile` ou `pam_wheel` no arquivo de controle PAM para o ssh restringir os logins ssh. Por exemplo, se quiser manter qualquer pessoa não listada em `/etc/loginusers` adicionando esta linha no `/etc/pam.d/ssh`:

```
auth       required     pam_listfile.so sense=allow onerr=fail item=user file
```

Como nota final, tenha atenção que estas diretivas são válidas para um arquivo de configuração do OpenSSH. Atualmente, não freqüentemente usados três tipos de implementações conhecidas do daemon: ssh1, ssh2 e OpenSSH feito pelo time do OpenBSD. O ssh1 foi o primeiro daemon disponível e é ainda o mais usado (existem rumores que até existe um porte para Windows). O ssh2 possui mais vantagens sobre o ssh2, exceto que ele é lançado sob uma licença fonte fechado. O OpenSSH é um daemon ssh completamente livre, que suporta ambos os protocolos ssh1 e ssh2. O OpenSSH é a versão instalada junto o Debian quando o pacote `ssh` é escolhido.

Você pode ler mais informações sobre como configurar um SSH com suporte a PAM em arquivos da lista de segurança (http://lists.debian.org/debian-security/2001/debian-security-200111/msg00395.html).

5.1.1 Executando o ssh em uma jaula chroot

O OpenSSH atualmente não suporta um método de chroot automático durante a conexão do usuário (a versão comercial oferece esta funcionalidade). No entanto existe um projeto para fornecer esta funcionalidade também para o ssh, veja http://chrootssh.sourceforge.net, atualmente ele não esta empacotado para o Debian. Você poderá usar, no entanto, o módulo pam_chroot como descrito em 'Restringindo acessos de usuários' on page 52.

Em 'Ambiente chroot para SSH' on page 207 você terá diversas opções para criar um ambiente chroot para o SSH.

5.1.2 Clientes do ssh

Se estiver usando um cliente SSH com um servidor SSH, você deverá ter certeza que ele suporta os mesmos protocolos que são especificados no servidor. Por exemplo, se utilizar o pacote mindterm, ele somente utiliza a versão 1 . No entanto, o servidor ssh utiliza, por padrão, a configuração para aceitar somente conexões para o protocolo da versão 2 (por razões de segurança).

5.1.3 Desativando transferências de arquivos

Se *não* quiser que seus usuários transfiram arquivos do servidor ssh, você precisará restringir acesso ao sftp-server *e* ao scp. Você poderá restringir o sftp-server configurando o sub-sistema Subsystem no arquivo /etc/ssh/sshd_config. No entanto para restringir o acesso ao scp você deverá:

- bloquear o login de usuários ao servidor ssh (como descrito acima no arquivo de configuração ou configuração do PAM).
- não fornecer shells validas para usuários que não tem permissão de realizar transferências de arquivos seguras. O shell fornecido, no entanto, programas que podem tornar a conexão ao ssh útil, como o menu (estilo BBS). Caso contrário, a opção anterior é a preferida.

5.2 Tornando o Squid mais seguro

O Squid é um dos servidores proxy/cache mais populares e existem algumas considerações de segurança que devem ser levadas em conta. O arquivo de configuração padrão do squid nega todas as requisições de usuários. No entanto, o pacote do Debian permite o acesso através de "localhost", você apenas precisa configurar seu navegador adequadamente. Configure o Squid para permitir acesso aos usuários confiáveis, máquinas ou redes definindo uma lista de controle de acesso no arquivo /etc/squid.conf, veja o endereço Guia do Usuário Squid (http://squid-docs.sourceforge.net/latest/html/book1.html) para mais informações

sobre a definição de regras de ACLs. Note que o Debian oferece uma configuração mínima para o Squid que prevenirá tudo, exceto a conexão de *localhost* em seu servidor proxy (que é executado na porta padrão 3128) É necessária a personalização do arquivo de configuração `/etc/squid.conf` como necessário. A configuração mínima recomendada (fornecida com o pacote) é mostrada abaixo:

```
acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl SSL_ports port 443 563
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443 563     # https, snews
acl Safe_ports port 70          # gopher
acl Safe_ports port 210         # wais
acl Safe_ports port 1025-65535  # portas não registradas
acl Safe_ports port 280         # http-mgmt
acl Safe_ports port 488         # gss-http
acl Safe_ports port 591         # filemaker
acl Safe_ports port 777         # multiling http
acl Safe_ports port 901         # SWAT
acl purge method PURGE
acl CONNECT method CONNECT
(...)
# Somente permite acesso do cachemgr vindos de localhost
http_access allow manager localhost
http_access deny manager
# Somente permite requisições de purge vindas de localhost
http_access allow purge localhost
http_access deny purge
# Bloqueia requisições para portas desconhecidas
http_access deny !Safe_ports
# Bloqueia CONNECT a portas que não sejam SSL
http_access deny CONNECT !SSL_ports
#
# INSIRA SUAS PRÓPRIAS REGRAS AQUI PARA PERMITIR O ACESSO DE SEUS CLIENTE
#
http_access allow localhost
# E finalmente bloqueia qualquer outro acesso a este proxy
http_access deny all
#Padrão:
# icp_access deny all
#
#Permite requisições ICQ vindas de qualquer pessoa
icp_access allow all
```

Você também deverá configurar o Squid baseado nos recursos do seu sistema, incluindo a memória cache (opção `cache_mem`), localização dos arquivos de cache e quantidade de espaço que utilizarão no disco (opção `cache_dir`).

Note que, se não for corretamente configurado, alguém poderá enviar mensagens de e-mail através do squid, pois os protocolos HTTP e SMTP tem design similar. O arquivo de configuração padrão do Squid bloqueia o acesso a porta 25. Se desejar permitir conexões a porta 25, apenas adicione-a a lista Safe_ports. No entanto, isto *NÃO* é recomendado.

Ajustar e configurar um servidor proxy/cache é apenas parte da tarefa de manter um site seguro. Outra tarefa necessária é a análise dos logs do Squid para ter certeza que todas as coisas estão funcionando como deveriam estar. Existem alguns pacotes no Debian GNU/Linux que podem ajudar o administrador a fazer isto. Os seguintes pacotes estão disponíveis na woody (Debian 3.0):

- `calamaris` - Analisador de arquivos de log para o Squid ou log do proxy Oops
- `modlogan` - Um analisador de arquivos e log modular.
- `squidtaild` - Programa de monitoramento de logs do Squid.

Quando estiver usando o squid em modo acelerador, ele atuará como servidor web também. Ativando esta opção, a complexidade do código aumenta, tornando-a menos confiável. Por padrão, o squid não é configurado para atuar como um servidor web, assim não precisará se preocupar com isto. Note que se quiser usar esta característica, tenha certeza que é realmente necessária. Para encontrar mais informações sobre o modo acelerador do Squid, veja Squid User's Guide #Chapter9 (http://squid-docs.sourceforge.net/latest/html/c2416.html).

5.3 Tornando o FTP mais seguro

Se realmente precisar usar o FTP (sem transportá-lo com sslwrap ou dentro de um tunel SSL ou SSH), você deverá fazer um chroot dentro do diretório de usuários do ftp, assim o usuário será incapaz de ver qualquer coisa que não seja seu próprio diretório. Caso contrário, ele poderá atravessar seu sistema de arquivos raíz como se tivesse uma conta shell. Você poderá adicionar a seguinte linha no seu arquivo `proftpd.conf` na sua seção global para ativar esta característica chroot:

```
DefaultRoot ~
```

Reinicie o proftpd executando `/etc/init.d/proftpd restart` e verifique se agora pode escapar do seu diretório de usuário.

Para prevenir ataques DoS usando ../../.., adicione a seguinte linha no seu arquivo `/etc/proftpd.conf`: `DenyFilter \*.*/`

Lembre-se sempre que o FTP envia o login e senhas de autenticação em texto plano (isto não é um problema se estiver oferecendo acesso a serviços públicos. Entretanto existem alternativas melhores no Debian para isto, como o `sftp` (fornecido pelo pacote `ssh`). Também existem implementações livres do ssh para outros sistemas operacionais, por exemplo: putty (http://www.chiark.greenend.org.uk/~sgtatham/putty/) e o cygwin (http://www.cygwin.com).

No entanto, se você ainda mantém um servidor FTP enquanto disponibiliza o acesso através do SSH você deve encontrar um problema típico. Usuários acessando servidores FTP anônimos dentro de sistemas protegidos com o SSH devem tentar efetuar o login no *FTP server*. Enquanto o acesso será recusado, as senhas nunca serão enviadas na rede de forma desprotegida. Para evitar isto, o desenvolvedor TJ Sauders do ProFTPd , criou um patch que evita que os usuários utilizem um servidor FTP anônimo com uma conta válida do ssh. Mais informações e o patch estão disponíveis em: ProFTPD Patches (http://www.castaglia.org/proftpd/#Patches). Este patch também foi reportado para o Debian, veja Bug #145669 (http://bugs.debian.org/145669).

5.4 Tornando o acesso ao sistema X Window mais seguro

Hoje em dia, terminais do X são usados por mais e mais empresas onde é necessário para várias estações de trabalho. Isto pode ser perigoso, porque você precisa permitir o servidor de arquivos a se conectar aos clientes (a partir do ponto de vista do servidor X, o X altera a definição de cliente e servidor). Se você seguir a (péssima) sugestão de muitas documentações você digitará `xhost +` em sua máquina. Isto permitirá qualquer cliente do X a se conectar em seu sistema. Para ter um pouco mais de segurança, você deverá usar o comando `xhost +hostname` ao invés de somente permitir acessos através de máquinas específicas.

Uma solução muito mais segura, no entanto, é usar o ssh para fazer o túnel do X e criptografia para toda a seção. Isto é feito automaticamente quando você faz um ssh para a outra máquina. Para isto funcionar, você terá que configurar ambos o cliente ssh e o servidor ssh. No cliente ssh, a opção `ForwardX11` deverá estar ajustada para `yes` no arquivo `/etc/ssh/ssh_config`. No servidor ssh, a opção `X11Forwarding` deverá estar ajustada para `yes` no arquivo `/etc/ssh/sshd_config` e o pacote `xbase-clients` deverá estar instalado, pois o servidor ssh utiliza o `/usr/X11R6/bin/xauth` quando está configurando uma tela de pseudo terminal do X. Nos tempos do SSH, agora você deverá deixar de usar o controle de acesso baseado em xhost completamente.

Para melhor segurança, você não precisará permitir o acesso ao X a partir de outras máquinas, isto é feito desativando o servidor na porta 6000 simplesmente digitando:

```
$ startx -- -nolisten tcp
```

Este é o comportamento padrão do Xfree 4.1.0 (o Xserver fornecido no Debian 3.0). Se estiver executando o Xfree 3.3.6 (i.e. você tem o Debian 2.2 instalada) você poderá editar o arquivo `/etc/X11/xinit/xserverrcc` e fazer a alteração nestas seguintes linhas:

```
#!/bin/sh
exec /usr/bin/X11/X -dpi 100 -nolisten tcp
```

Se estiver usando o conjunto do XDM altere no arquivo `/etc/X11/xdm/Xservers` para: `:0 local /usr/bin/X11/X vt7 -dpi 100 -nolisten tcp`. Se estiver usando o Gdm tenha certeza que a opção `-nolisten tcp` está definida no arquivo `/etc/gdm/gdm.conf` (que é o padrão no Debian) tal como esta:

```
[server-Standard]
name=Standard Server
command=/usr/bin/X11/X -nolisten tcp
```

Você também poderá configurar o timeout padrão para o travamento do `xscreensaver`. Até mesmo se o usuário substituir este valor, você poderá editar o arquivo `/etc/X11/app-defaults/XScreenSaver` e alterar a linha:

```
*lock:                  False
```

(que é padrão no Debian) para:

```
*lock:                  True
```

FIXME: adicionar informações sobre como desativar as proteções de tela que mostra o desktop do usuário (que pode conter informações sensíveis).

Leia mais sobre a segurança em servidores X Window em XWindow-User-HOWTO (http://www.tldp.org/HOWTO/XWindow-User-HOWTO.html) (`/usr/share/doc/HOWTO/en-txt/XWindow-User-HOWTO.txt.gz`).

FIXME: Adicionar informações sobre a discussão na debian-security sobre como alterar os arquivos de configuração no servidor XFree 3.3.6 para fazer isto.

5.4.1 Verifique seu gerenciador de tela

Se somente quiser ter um gerenciador de tela instalado para uso local (tendo um lindo login gráfico) tenha certeza que tudo que estiver relacionado com o XDMCP (X Display Manager Control Protocol) está desativado. No XDM você poderá fazer isto através da linha em `/etc/X11/xdm/xdm-config`:

```
DisplayManager.requestPort:     0
```

Normalmente, todos os gerenciadores de tela estão configurados para não iniciar serviços do XDMCP por padrão no Debian.

5.5 Tornando o servidor de impressão mais seguro (sobre o lpd e lprng)

Imagine, você chegando ao trabalho e a impressora jogando fora uma quantidade impressionante de papel porque alguém esta fazendo um DoS em seu daemon de impressão. Desagradável, não é?

Em qualquer arquitetura de impressão do unix, deverá existir uma forma de enviar os dados do cliente para o servidor de impressão. No tradicional `lpr` e `lp`, os comandos do cliente copiam ou fazem um link simbólico de dados no diretório de spool (este é o motivo porque estes programas normalmente são SUID ou SGID).

Para evitar quaisquer anormalidade, você deverá manter o seu servidor de impressão especialmente seguro. Isto significa que precisa configurar seu serviço de impressão de forma que só permita conexões de um conjunto de máquinas confiáveis. Para fazer isto, adicione os servidores que deseja permitir a impressão em seu arquivo `/etc/hosts.lpd`.

No entanto, até mesmo se fizer isto, o `lpr` aceitará conexões de entrada na porta 515 de qualquer interface. Você deverá considerar fazer um firewall das conexões de redes/hosts que não tenham permissão de impressão (o daemon `lpr` não tem a possibilidade de aceitar conexões em somente um determinado endereço IP).

O `Lprng` deverá ser o preferido em cima do `lpr` pois ele pode ser configurado para fazer controle de acesso por IP. E você poderá especificar qual interface escutará por conexões (embora algumas vezes pareça um pouco estranho).

Se utilizar uma impressora em seu sistema, mas somente localmente, você não desejará compartilhar este serviço através de uma rede. Você poderá considerar o uso de outros sistemas de impressão, tal como o fornecido pelo pacote `cups` ou pelo PDQ (`http://pdq.sourceforge.net/`) que é baseado em permissões do usuário no dispositivo `/dev/lp0`.

No `cups`, os dados de impressão são transferidos ao servidores via protocolo http. Isto significa que o programa cliente não precisa de qualquer privilégio especial, mas requer que o servidor escute em uma porta, em algum lugar.

No entanto, se quiser usar o `cups`, mas somente localmente, você poderá configura-lo para escutar na interface loopback alterando o arquivo de configuração `/etc/cups/cupsd.conf`:

```
Listen 127.0.0.1:631
```

Existem muitas outras opções de segurança como permitir ou bloquear redes e máquinas neste arquivo de configuração. No entanto, se você não precisar delas, será melhor que limite simplesmente a porta onde o programa espera por conexões. O `Cups` também serve documentações através da porta HTTP. Se não quiser revelar informações úteis em potencial para invasores externos também adicione:

```
<Location />
  Order Deny,Allow
```

```
    Deny From All
     Allow From 127.0.0.1
</Locationi>
```

Este arquivo de configuração pode ser modificado para adicionar algumas outras características incluindo certificados SSL/TLS e criptografia. Os manuais estão disponíveis em http://localhost:631/ ou em cups.org.

FIXME: Adicionar mais conteúdo (o artigo em Amateur Fortress Building (http://www.rootprompt.org) fornecendo visões mais interessantes).

FIXME: Verificar se o PDG está disponível no Debian, e se estiver, sugerir como sistema de impressão preferido.

FIXME: Verificar se o Farmer/Wietse possui um substituto para daemon de impressão e se está disponível no Debian.

5.6 Tornando o serviço de e-mails seguro

Se seu servidor não for um servidor de mensagens, e realmente não precisa ter um programa esperando por conexões de entradas, mas deseja que as mensagens locais sejam entregues, por exemplo, para recebimento de mensagens do usuário root de qualquer alerta de segurança que tenha no local.

Se tiver o exim você não precisará do daemon funcionando para fazer isto, pois o pacote padrão do cron esvazia a fila de mensagens. Veja 'Desabilitando daemons de serviço' on page 33 para saber como fazer isto.

5.6.1 Configurando um programa de e-mails nulo

Você pode querer ter um daemon de mensagens locais assim ele poderá repassar os e-mails enviados localmente para outro sistema. Isto é comum quando você tem que administrar um número de máquinas e não quer conectar a cada uma delas para ler as mensagens enviadas localmente. Assim como todos os logs de cada sistema individual podem ser centralizados usando um servidor de logs central, as mensagens podem ser enviadas para um servidor de mensagens central.

Tal sistema *somente-repasse* deverá ser configurado adequadamente para fazer isto. O daemon poderá, também, ser configurado para somente esperar por conexões no endereço de loopback.

FIXME: Isto deverá ser atualizado para o exim4, que é o MTA padrão da sarge e distribuições mais atuais (e espera por conexões somente em localhost na configuração padrão mínima)

Para fazer isto em um sistema Debian 3.0 usando o pacote exim, você terá que remover o daemon smtp do inetd:

```
$ update-inetd --disable smtp
```

e configurar o daemon de mensagens para somente esperar por conexões na interface loopback. No `exim` (o MTA padrão) você poderá fazer isto editando o arquivo de configuração `/etc/exim.conf` e adicionando a seguinte linha:

```
local_interfaces = "127.0.0.1"
```

Reinicie ambos os daemons (inetd e exim) e você terá o exim esperando por conexões somente no soquete 127.0.0.1:25. Seja cauteloso e desative primeiro o inetd, caso contrário, o exim não iniciará pois o daemon do inetd já está esperando por conexões de entrada.

Para o `postfix`, edite o arquivo `/etc/postfix/main.conf`:

```
inet_interfaces = localhost
```

Se quiser somente mensagens locais, este método é melhor que utilizar o método tcp wrappers no daemon de mensagens ou adicionar regras de firewall para que ninguém acesse-o. No entanto, se precisar que ele escute em outras interfaces, você deverá considerar carrega-lo a partir do inetd e adicionar um tcp wrapper, assim as conexões de entradas são verificadas nos arquivos `/etc/hosts.allow` e `/etc/hosts.deny`. Também, você deverá estar atento sobre acessos não autorizados sendo tentados sobre o seu daemon de mensagens, se configurar adequadamente o log de mensagens do seu sistema para qualquer um dos métodos acima.

Em qualquer caso, para rejeitar tentativas de repasse de mensagens a nível SMTP, você deverá alterar o arquivo `/etc/exim/exim.conf` para incluir:

```
receiver_verify = true
```

Até mesmo se seu servidor de e-mails não repassar a mensagem, este tipo de configuração é necessário para o teste de relay em `http://www.abuse.net/relay.html` para determinar que seu servidor *não* é capaz de repassar mensagens.

No entanto, se desejar uma configuração somente de leitura, você poderá considerar a alteração do daemon de mensagens para programas que podem *somente* ser configurados para redirecionar as mensagens para servidores de mensagens remotas. O Debian atualmente oferece o pacote `ssmtp` e o `nullmailer` para este propósito. Em qualquer caso, você deverá avaliar por si mesmo quaisquer dos agentes de transporte de mensagens [1] fornecido com o Debian. Veja que programa atende melhor aos propósitos do sistema.

[1] para obter uma lista de todos os daemons de mensagens disponíveis no Debian, execute o comando:

```
$ apt-cache search mail-transport-agent
```

A lista não incluirá o `qmail`, que é distribuído somente como código fonte no pacote `qmail-src`.

5.6.2 Fornecendo acesso seguro às caixas de mensagens

Se quiser oferecer acesso remoto às caixas de mensagens, existe um número de daemons POP3 e IMAP disponíveis [2] . No entanto, se você oferecer acesso a IMAP, note que ele é um protocolo de acesso a arquivos, ele pode se tornar equivalente a um acesso shell porque os usuários podem ser capazes de obter qualquer arquivo através dele.

Tente, por exemplo, configurar como seu caminho para a inbox`{servidor.com}/etc/passwd`, se ele abrir o arquivo com sucesso seu daemon IMAP não está corretamente configurado para prevenir este tipo de acesso.

Dos servidores de IMAP existentes no Debian, o servidor `cyrus` (do pacote `cyrus-imapd`) contorna isto tendo todos os acessos sendo em um banco de dados mantido em uma parte restrita do sistema de arquivos. Também o `uw-imapd` (ou instale o `uw-imapd` ou melhor, se seus clientes IMAP o suportam, `uw-imapd-ssl`) poderá ser configurado para fazer o chroot do diretório dos usuários de mensagens mas isto não é ativado por padrão. A documentação fornecida oferece mais informações sober como configura-lo.

Também, você pode tentar executar um servidor IMAP que não precisa de usuários válidos sendo criados no sistema local (que também oferece acesso a shell). Ambos os pacotes `courier-imap` (para IMAP) e `courier-pop` `teapop` (para o POP3) e o `cyrus-imapd` (para ambos POP3 e IMAP) fornecem servidores com métodos de autenticação que não dependem de contas locais de usuários. O `cyrus` pode usar qualquer método de autenticação que possa ser configurado através do PAM tal como o `teapop` pode usar bancos de dados (tal como o `postgresql` e o `mysql`) para autenticação do usuário.

FIXME: Verifique: `uw-imapd` também precisa ser configurado com autenticação do usuário através de PAM...

5.6.3 Recebendo mensagens de forma segura

A leitura/recebimento de mensagens é o protocolo de texto puro mais comum. Se usar ou POP3 ou IMAP para obter suas mensagens, você enviará sua senha em texto plano através da rede, assim praticamente qualquer um poderá ler suas mensagens de agora em diante. Ao invés disto, utiliza-se SSL (Secure Sockets Layer) para receber seus e-mails. A outra alternativa é utilizar o ssh, se tiver uma conta shell na máquina que atua como seu servidor POP ou IMAP. Aqui está um arquivo de configuração `fetchmailrc` básico para demonstrar isto:

```
poll my-imap-mailserver.org via "localhost"
  with proto IMAP port 1236
      user "ref" there with password "hackme" is alex here warnings 3600
    folders
      .Mail/debian
```

[2] Uma lista de servidores/daemons que suportam estes protocolos podem ser obtidos com:

`$ apt-cache search pop3-server $ apt-cache search imap-server`

```
preconnect 'ssh -f -P -C -L 1236:my-imap-mailserver.org:143 -l ref
 my-imap-mailserver.org sleep 15 </dev/null > /dev/null'
```

A linha preconnect é importante. Ela executa uma seção ssh e cria o túnel necessário, que automaticamente redireciona conexões para localhost da porta 1236 para o servidor de mensagens IMAP, mas de forma criptografada. Outra possibilidade será usar o fetchmail com características ssl.

Se deseja fornecer serviços de mensagens criptografadas como POP e IMAP,`apt-get install stunnel` e inicie seus daemons da seguinte forma:

```
stunnel -p /etc/ssl/certs/stunnel.pem -d pop3s -l /usr/sbin/popd
```

Este comando direciona as conexões do daemon fornecido (-l) para a porta (-d) e utiliza o certificado ssl especificado (-p).

5.7 Tornando o BIND mais seguro

Existem diferentes métodos que podem ser usados para deixar o daemon de serviços de Domínio mais seguro, que são parecidos com os mostrados considerados quando tornamos qualquer determinado serviço mais seguro:

- configurando o próprio daemon adequadamente assim ele não poderá ser abusado de fora (veja 'Configuração do Bind para evitar má utilização' on this page) Isto inclui limitar requisições de clientes: transferências de zonas e pesquisas recursivas.
- limitar o acesso do daemon ao próprio servidor assim se ele for usado para um corrompimento, a falha no sistema será limitada. Isto inclui executar o daemon como um usuário não-privilegiado (veja 'Alterando o usuário do BIND' on page 94) e fazer ele rodar dentro um chroot (see 'Executando o servidor de nomes em uma jaula chroot' on page 96)

5.7.1 Configuração do Bind para evitar má utilização

Você deverá restringir algumas das informações que são servidas pelo BIND para clientes externos, assim não poderão ser usadas para obter informações sobre sua empresa que não deseja dar. Isto inclui adicionar as seguintes opções: *allow-transfer*, *allow-query*, *allow-recursion* e *version*. Você pode ou limitar esta seção global (assim aplicando a todas as zonas que são servidas) ou por zona. Esta informação está incluída no pacote `bind-doc`, leia mais sobre isto em `/usr/share/doc/bind/html/index.html` assim que o pacote for instalado.

Imagine que seu servidor (um servidor básico contendo múltiplos endereços) está conectado à Internet e à sua rede interna (seu endereço IP é 192.168.1.2), você não vai querer oferecer qualquer serviço para os computadores. Você poderá restringir o bind incluindo o seguinte no `/etc/bind/named.conf`:

```
options {
     allow-query { 192.168.1/24; } ;
     allow-transfer { none; } ;
     allow-recursion { 192.168.1/24; } ;
     listen-on { 192.168.1.2; } ;
     forward { only; } ;
     forwarders { A.B.C.D; } ;
};
```

A opção *listen-on* faz o BIND ser executado somente na interface que tem o endereço interno, mas, até mesmo se esta interface for a mesma que te conecta a internet (caso estiver usando NAT, por exemplo), as requisições serão aceitas somente se estiverem vindo de suas máquinas internas. Se o sistema tiver múltiplas interfaces e a opção *listen-on* não estiver presente, somente usuários internos poderão fazer requisições, mas, como a porta está acessível para possíveis invasores externos, eles podem tentar travar (ou tentar realizar ataques de estouro de buffer) no servidor DNS. Você poderia até fazê-lo escutar somente em 127.0.0.1, se não estiver oferecendo o serviço de DNS em qualquer outro sistema além do seu.

O registro version.bind na classe chaos contém a versão do processo do bind atualmente em execução. Esta informação é freqüentemente usada por scaneadores automáticos e individualmente por pessoas maliciosas que desejam determinar se o bind é vulnerável a um ataque específico. Oferecendo informações falsas ou não fornecendo informações ao registro version.bind, diminui a probabilidade que o servidor seja atacado baseado na versão publicada. Para fornecer sua própria versão, use a diretiva *version* da seguinte forma:

```
 options { ... várias opções aqui ...
version "Não disponível."; };
```

A alteração do registro version.bind não oferece proteção atualmente contra ataques, mas pode ser considerado útil para a segurança.

Um arquivo simples de configuração `named.conf` pode ser o seguinte:

```
acl internal {
        127.0.0.1/32;           // localhost
        10.0.0.0/8;             // interna
        aa.bb.cc.dd;            // IP da eth0
};

acl friendly {
        ee.ff.gg.hh;            // DNS escravo
        aa.bb.cc.dd;            // IP da eth0
        127.0.0.1/32;           // localhost
        10.0.0.0/8;             // interna
};
```

```
options {
        directory "/var/cache/bind";
        allow-query { internal; };
        allow-recursion { internal; };
        allow-transfer { none; };
};
// A partir daqui, a zona mysite.bogus é
// basicamente uma versão não modificada do padrão do Debian
logging {
        category lame-servers { null; };
        category cname { null; };
};

zone "." {
        type hint;
        file "/etc/bind/db.root";
};

zone "localhost" {
        type master;
        file "/etc/bind/db.local";
};

zone "127.in-addr.arpa" {
        type master;
        file "/etc/bind/db.127";
};

zone "0.in-addr.arpa" {
        type master;
        file "/etc/bind/db.0";
};

zone "255.in-addr.arpa" {
        type master;
        file "/etc/bind/db.255";
};

// zones I added myself
zone "mysite.bogus" {
        type master;
        file "/etc/bind/named.mysite";
        allow-query { any; };
        allow-transfer { friendly; };
};
```

Por favor (novamente) verifique o Sistema de Tratamento de Falhas a respeito do bind, especificamente Bug #94760 (relacionado com ACLs em transferência de zonas) (http://bugs.debian.org/94760). Sinta-se livre para contribuir para relatar falhas se achar que podem adicionar informações úteis.

5.7.2 Alterando o usuário do BIND

Com relação a limitação de privilégios do BIND, você deverá estar ciente que se um usuário não root executa o BIND, então o BIND não detectará novas interfaces automaticamente, por exemplo, se colocar uma placa PCMCIA no notebook. Verifique o arquivo README.Debian na documentação do named veja o diretório (`/usr/share/doc/bind/README.Debian`) para mais informações sobre este assunto. Ocorreram muitos problemas de segurança recentes relacionados com o BIND, assim a alteração do usuário é mais útil quando possível. Nós detalharemos os passos para fazer isto, no entanto, se quiser fazer isto de uma forma automática, tente o script fornecido em 'Exemplo de script para alterar a instalação padrão do Bind.' on page 199.

Para executar o BIND sob um usuário diferente, primeiro crie um usuário separado e um grupo (*não* é uma boa idéia usar o nobody ou nogroup para cada serviço que não estiver sendo executado como root). Neste exemplo, o usuário e grupo `named` serão usados. Você poderá fazer isto da seguinte forma:

```
addgroup named
adduser --system --home /home/named --no-create-home --ingroup named \
      --disabled-password --disabled-login named
```

Note que o usuário `named` será bastante restringido. Se você quiser, por alguma razão, ter uma configuração menos restrita, utilize:

```
adduser --system --ingroup named named
```

Agora, edite o arquivo `/etc/init.d/bind` com seu editor favorito e altere a linha que começa com

```
start-stop-daemon --start
```

para[3]

```
start-stop-daemon --start --quiet --exec /usr/sbin/named -- -g named -u named
```

Altere as permissões dos arquivo que são usados pelo Bind, incluindo `/etc/bind` `/rndc.key`:

[3]Note que dependendo de sua versão do BIND você pode não ter a opção `-g`, mais precisamente se estiver usando a woody e instalando o bind9 (9.2.1-2.woody).

```
-rw-r-----    1 root     named          77 Jan  4 01:02 rndc.key
```

e onde o bind cria seu arquivo de pid, usando, por exemplo, `/var/run/named` ao invés de `/var/run`:

```
$ mkdir /var/run/named
$ chown named.named /var/run/named
$ vi /etc/named.conf
[ ... atualize o arquivo de configuração para sua nova localização ...]
options { ...
        pid-file "/var/run/named/named.pid";
};
[ ... ]
```

Também, para evitar a execução de tudo como usuário root, altere a linha `reload` comentando-a:

```
reload)
       /usr/sbin/ndc reload
```

E altere para:

```
reload)
        $0 stop
        sleep 1
        $0 start
```

Nota: Dependendo de sua versão do Debian, você deverá também alterar a linha `restart`. Isto foi corrigido na versão do Bind do Debian `1:8.3.1-2`.

Tudo que precisa fazer agora é reiniciar o bind via '/etc/init.d/bind restart', e então procurar em seu syslog pelas seguintes duas linhas, como estas:

```
Sep  4 15:11:08 nexus named[13439]: group = named
Sep  4 15:11:08 nexus named[13439]: user = named
```

Voilá! Seu named agora *não é executado* como root. Se desejar ler mais informações sobre porque o BIND não pode ser executado por um usuário não-root em sistemas Debian, verifique o sistema de tratamento de falhas, especificamente Bug #50013: bind should not run as root (http://bugs.debian.org/50013) e Bug #132582: Default install is potentially insecure (http://bugs.debian.org/132582), Bug #53550 (http://bugs.debian.org/53550), Bug #128120 (http://bugs.debian.org/52745), e Bug #128120 (http://bugs.debian.org/128129). Sinta-se livre para contribuir para os relatórios de falhas se achar que pode adicionar informações úteis.

5.7.3 Executando o servidor de nomes em uma jaula chroot

Para obter o máximo de segurança no BIND, agora construa uma jaula chroot (veja 'Paranóia geral do chroot e suid' on page 100) em torno do seu daemon. Existe um método fácil de se fazer isto: a opção `-t` (veja a `named(8)` página de manual ou a página 100 do Documentação do Bind's 9 (PDF) (`http://www.nominum.com/content/documents/bind9arm.pdf`)). Isto instruirá o Bind a fazer uma jaula de si mesmo em um diretório especificado sem a necessidade de configurar uma jaula chroot e se preocupar com as bibliotecas dinâmicas. Os únicos arquivos que precisam estar na jaula são:

```
dev/null
etc/bind/        - deverá ter o named.conf e todas as zonas do servidor
sbin/named-xfer  - se fizer transferências de nomes
var/run/named/   - deverá ter a pid e o nome do servidor de cache (se tiver)
                   este diretório precisa ter permissões de gravação para o
                   usuário named.
var/log/named    - se configurar o log para um arquivo, este precisa ter permi
                   de gravação para o usuário named
dev/log          - o syslogd deverá estar escutando aqui caso o named estiver
                   configurado para realizar logs através dele.
```

Para seu daemon do Bind funcionar adequadamente, ele precisará de permissões nos arquivos do named. Esta é uma tarefa simples, pois os arquivos de configuração estão sempre localizados em `/etc/named/`. Tenha em mente que ele somente precisa de acesso de leitura aos arquivos de zonas, a não ser que seja um DNS secundário ou servidor de cache de nomes. Se este é seu caso, você terá que dar permissões completas para as zonas necessárias (assim as zonas transferidas do servidor principal funcionarão).

Adicionalmente, mais detalhes sobre o Bind e chroot pode ser encontrados no Chroot-BIND-HOWTO (`http://www.tldp.org/HOWTO/Chroot-BIND-HOWTO.html`) (relacionado com o Bind 9) e Chroot-BIND8-HOWTO (`http://www.tldp.org/HOWTO/Chroot-BIND8-HOWTO.html`) (relacionado com o Bind 8). Este mesmo documento deverá estar disponível através da instalação do `doc-linux-text` (versão texto) ou `doc-linux-html` (versão html). Outro documento útil é `http://web.archive.org/web/20011024064030/http://www.psionic.com/papers/dns/dns-linux`.

Se estiver configurando uma jaula completa do chroot (i.e. não somente `-t`) para o Bind 8.2.3 no Debian (potato), tenha certeza de possuir os seguintes arquivos nela:

```
dev/log - o syslogd deverá estar escutando aqui
dev/null
etc/bind/named.conf
etc/localtime
etc/group - com somente uma linha simples: "named:x:GID:"
etc/ld.so.cache - gerado com o ldconfig
lib/ld-2.1.3.so
```

```
lib/libc-2.1.3.so
lib/ld-linux.so.2 - link simbólico para ld-2.1.3.so
lib/libc.so.6 - link simbólico para libc-2.1.3.so
sbin/ldconfig - pode ser apagado após configurar a jaula chroot
sbin/named-xfer - se fizer transferências de nomes
var/run/
```

Também modifique o `syslogd` para escutar no $CHROOT/dev/log assim o servidor de nomes poderá gravar entradas do syslog no log local do sistema.

Se deseja evitar problemas com bibliotecas dinâmicas, você poderá compilar o binário estaticamente. Você poderá usar o `apt-get` para fazer isto, com a opção `source`. Ele pode até mesmo baixar os pacotes que precisa para compila-los adequadamente. Você deverá fazer algo similar a isto:

```
$ apt-get --download-only source bind build-dep bind
$ cd bind-8.2.5-2
(edite o Makefile.in assim CFLAGS incluirá a opção '-static'
antes da definição @CFLAGS@ substituída pelo autoconf)
$ dpkg-buildpackage -rfakeroot
$ cd ..
$ dpkg  -i bind-8.2.5-2*deb
```

Após a instalação, você precisará mover os arquivos para a jaula chroot [4] você poderá manter os scripts do `init.d` em `/etc/init.d` assim o sistema irá iniciar automaticamente o servidor de nomes, mas edite-os para adicionar `--chroot /location_of_chroot` nas chamadas para `start-stop-daemon` nestes scripts.

Para mais informações sobre como configurar jaulas chroot veja 'Paranóia geral do chroot e suid' on page 100.

FIXME, merge info from http://people.debian.org/~pzn/howto/chroot-bind.sh.txt, http://www.cryptio.net/~ferlatte/config/ (Debian-specific), http://web.archive.org/web/20021216104548/http://www.psionic.com/papers/whitep01.html and http://csrc.nist.gov/fasp/FASPDocs/NISTSecuringDNS.htm.

5.8 Tornando o Apache mais seguro

FIXME: Adicionar conteúdo: os módulos fornecidos com a instalação padrão do Apache (sob /usr/lib/apache/X.X/mod_*) e módulos que podem ser instalados separadamente pelos pacotes libapache-mod-XXX.

[4] a não ser que utilize a opção `instdir` quando executar o `dpkg` mas então a jaula chroot será um pouco mais complexa

Você poderá limitar o acesso ao servidor Apache se você somente deseja usar ele internamente (para propósitos de testes, para acessar os arquivos do `doc-central`, etc..) e não deseja que pessoas de fora o acessem. Para fazer isto, use as diretivas `Listen` ou `BindAddress` no `/etc/apache/http.conf`.

Using Listen:

```
Listen 127.0.0.1:80
```

Using BindAddress:

```
BindAddress 127.0.0.1
```

Então reinicie o apache com `/etc/init.d/apache restart` e você verá que ele somente esperará por requisições na interface loopback.

Em qualquer caso, se não estiver usando todas as funcionalidades fornecidas pelo Apache, você poderá querer dar uma olhada em outros servidores web fornecidos no Debian, como o `dhttpd`.

A Documentação do Apache (http://httpd.apache.org/docs/misc/security_tips.html) fornece informações relacionadas com medidas de segurança a serem tomadas no servidor web Apache (estes mesmos passos são oferecidos no Debian através do pacote `apache-doc`).

Mais informações sobre restrições do Apache configurando uma jaula chroot são mostradas em 'Ambiente `chroot` para `Apache`' on page 221.

5.8.1 Proibindo a publicação de conteúdo dos usuários

A instalação padrão do Apache no Debian permite que usuários publiquem conteúdo sob o diretório `$HOME/public_html`. Este conteúdo pode ser pego remotamente usando uma URL tal como: http://your_apache_server/~user.

Se não quiser permitir isto, você deverá alterar o arquivo de configuração `/etc/apache/http.conf` comentando a linha:

```
LoadModule userdir_module /usr/lib/apache/1.3/mod_userdir.so
```

Mas se um módulo foi incluído estaticamente (você poderá checar isto executando `apache -l`) você deverá utilizar a seguinte técnica:

```
Userdir disabled
```

Nota: A palavra chave `disabled` está somente disponível nas versões do Apache 1.3 e superior. Se estiver usando versões antigas do apache, você deverá alterar o arquivo de configuração e adicionar:

```
<Directory /home/*/public_html>
    AllowOverride None
    Order deny,allow
    Deny from all
</Directory>
```

Um invasor ainda pode usar enumeração de usuário, pois a resposta do servidor será um *403 Permissão negada* e não um *404 Não disponível*.

5.8.2 Permissões de arquivos de log

Os arquivos de log do Apache, desde a 1.3.22-1, tem como dono o usuário 'root' e grupo 'adm' com permissões 640, estas permissões são alteradas após o rotacionamento de logs. Um intruso que acessou o sistema através do servidor web não será capaz (sem escalação de privilégios) de remover entradas antigas do log.

5.8.3 Arquivos da Web Publicados

Os arquivos do Apache estão localizados sob `/var/www`. Apenas após a instalação o arquivo de configuração padrão fornecerá algumas informações sobre o sistema (principalmente que é um sistema Debian executando o Apache). As páginas web padrões tem como dono o usuário root e grupo root por padrão, enquanto o processo do Apache é executado como o usuário e grupo www-data. Isto torna difícil para invasores que comprometem o sistema através do servidor web, desfigurarem o site. Você deverá, é claro, substituir as páginas padrões por suas próprias (que fornecem informações que não deseja mostrar para pessoas de fora).

5.9 Tornando o finger mais seguro

Se desejar executar o serviço finger, primeiro pergunte a você mesmo porque o deseja. Se precisar dele, você verá que o Debian fornece vários daemons de finger (saída do comando `apt-cache search fingerd`):

- cfingerd - Daemon de finger configurável
- efingerd - Outro daemon de finger para unix, capaz de ajustes finos em sua saída.
- ffingerd - um daemon seguro do finger
- fingerd - Servidor remoto de informações do usuário.
- xfingerd - BSD-like daemon de finger com suporte a qmail.

O `ffingerd` é o daemon de finger recomendado se estiver usando-o em serviços públicos. Em qualquer caso, você é encorajado, quando estiver configurando através do inetd, xinetd ou tcpserver, a: limitar o número de processos que podem ser executados ao mesmo tempo, limitando o acesso ao daemon de finger de um número determinado de máquinas (usando o tcp wrappers) e escutando somente nas interfaces onde deve operar.

5.10 Paranóia geral do chroot e suid

O `chroot` é uma das mais poderosas possibilidades para restringir um daemon, ou um usuário ou outro serviço. Apenas imagine uma jaula em torno de seu alvo, onde o alvo não pode escapar dela (normalmente, mas existem várias condições que permitam que um escape de tal jaula). Se não confia em um usuário ou em um serviço, você poderá criar um ambiente root modificado para ele. Isto poderá usar algum espaço do disco para copiar todos os executáveis requeridos, assim como bibliotecas, na jaula. Mas então, até mesmo se o usuário fizer algo malicioso, o escopo do ano é limitado a jaula.

Muitos serviços executados como daemons poderão se beneficiar deste tipo de técnica. Os daemons que você instala no Debian não virão, no entanto, dentro de chroot [5] por padrão.

Isto inclui: servidores de nomes (tal como o `bind`), servidores web (tal como o `apache`), servidores de mensagens (tal como o `sendmail` e servidores ftp (tal como o `wu-ftpd`). Provavelmente basta dizer que a complexibilidade do BIND é a razão de que ele foi exposto a vários ataques nos últimos anos (see 'Tornando o BIND mais seguro' on page 91).

No entanto, o Debian não oferece muitos programas que podem ajuda-lo a configurar um ambiente `chroot`. Veja 'Criando automaticamente ambientes chroots' on the facing page.

De qualquer maneira, se executar qualquer serviço em seu sistema, considere torná-lo mais seguro o possível. Isto inclui: revogar os privilégios de root, executá-lo em um ambiente seguro (tal como uma jaula chroot) ou substituí-lo por um equivalente mais seguro.

No entanto, já esteja avisado que uma jaula `chroot` pode ser quebrada se o usuário dentro dela for o superusuário. Assim você deverá estar certo que o serviço está sendo executado por um usuário não privilegiado. Limitando seu ambiente, estará limitando os arquivos lidos/executáveis que o serviço poderá acessar, assim, limitando as possibilidade de uma escalação privilegiada usar as vulnerabilidade de segurança locais do sistema. Até mesmo nesta situação, você não poderá ter certeza completa de que lá não existe métodos para um invasor inteligente quebrar a jaula. Usando somente programas de servidor que tem a reputação de serem seguidos é uma boa medida adicional. Até mesmo minúsculos furos como arquivos abertos podem serem usados por um invasor com conhecimentos para quebrar o sistema. Após tudo isto, o `chroot` não foi designado como uma ferramenta de segurança, mas como uma ferramenta de testes.

[5] Eles não tentarão ser executados sob *mínimo privilégio* que inclui a execução de daemons com seus próprios usuários ao invés de tê-los executando como root

5.10.1 Criando automaticamente ambientes chroots

Existem diversos programas que fazem automaticamente o chroot de servidores e serviços. O Debian atualmente (aceita em maio de 2002) fornece o Wietse Venema's `chrootuid` no pacote `chrootuid`, assim como o pacote `compartment` e `makejail`. Estes programas podem criar um ambiente restritivo para a execução de qualquer programa (`chrootuid` lhe permite até executa-lo como um usuário restrito).

Algumas destas ferramentas podem ser usadas para criar facilmente um ambiente chroot. O programa `makejail` por exemplo, pode criar e atualizar uma jaula chroot com arquivos de configuração pequenos (ele fornece modelos de configuração para o `bind`, `apache`, `postgresql` e `mysql`). Ele tenta adivinhar e instalar na jaula todos os arquivos requeridos pelo daemon usando o `strace`, `stat` e dependências de pacotes do Debian. Mais informações podem ser obtidas em `http://www.floc.net/makejail/`. O `Jailer` é uma ferramenta similar que pode ser obtida de `http://www.balabit.hu/downloads/jailer/` e também está disponível como um pacote do Debian GNU.

5.11 Paranóia geral sobre senhas em texto puro

Você deverá tentar evitar qualquer serviço de rede que envia e receba senhas em texto puro através da rede, como o FTP/Telnet/NIS/RPC. O autor recomenda usar o ssh ao invés de telnet e ftp para qualquer um.

Tenha em mente que migrando do telnet para o ssh, mas continuando a usar outros protocolos de texto puro não aumenta sua segurança de qualquer modo! O melhor é remover o ftp, telnet, pop, imap, http e substituí-los por seus respectivos serviços criptografados. Você deverá considerar mover estes para suas versões SSL, ftp-ssl, telnet-ssl, pop-ssl, https ...

A maioria dos listados acima se aplicam para cada sistema Unix (você os encontrará se ler qualquer documento relacionado a tornar um sistema Linux (e outros tipos e Unix) mais seguro.

5.12 Desativando o NIS

Você não deverá usar o NIS, o Serviço de Informações de Rede, se possível, pois ele permite o compartilhamento de senha. Isto pode ser altamente inseguro se sua configuração for corrompida.

Se precisar de compartilhamento de senhas entre máquinas, você deverá considerar a adoção de outras alternativas. Por exemplo, a configuração de um servidor LDAP e o PAM para contactar o servidor LDAP para autenticação dos usuários. Você poderá encontrar uma configuração detalhada na LDAP-HOWTO (`http://www.tldp.org/HOWTO/LDAP-HOWTO.html`) (`/usr/share/doc/HOWTO/en-txt/LDAP-HOWTO.txt.gz`).

Mais detalhes sobre a segurança em NIS podem ser encontradas em NIS-HOWTO (`http://www.tldp.org/HOWTO/NIS-HOWTO.html`) (`/usr/share/doc/HOWTO/en-txt/NIS-HOWTO.txt.gz`).

FIXME (jfs): Adicionar detalhes de como configurar isto no Debian

5.13 Tornando serviços RPC mais seguros

Você deverá desativar RPC se não precisar dele.

Chamadas de Procedimentos Remotos (RPC) é um protocolo que os programas podem usar para solicitar serviços de outros programas localizados em diferentes computadores. O serviço `portmap` controla os serviços RPC mapeando números de programas RPC em números de portas DARPA; ele deverá estar sendo executado para executar chamadas RPC.

Serviços baseados em RPC tem tido um mal histórico de falhas de segurança, no entanto, o portmapper por si não (mas ainda fornece informações úteis ao atacante remoto). Note que alguns dos ataques DDoS (negação de serviço distribuídos) usam exploits rpc para entrar no sistema e atuar como o assim chamado agente/manipulador.

Você somente precisará do RPC se estiver usando um serviço baseado em RPC. Os serviços mais comuns baseados em RPC são o NFS (Network File System) e NIS (Network Information System). Veja a seção anterior para mais informações sobre o NIS. O Monitor de alterações de Arquivos (FAM) fornecido pelo pacote `fam` é também um serviço RPC, e assim depende do pacote `portmap`.

Os serviços NFS são muito importante em algumas redes. Se este for o caso para você, então terá que encontrar um balanceamento de segurança e usabilidade para sua rede. (Você poderá ler mais sobre a segurança em NFS no NFS-HOWTO (`http://www.tldp.org/HOWTO/NFS-HOWTO.html`) (`/usr/share/doc/HOWTO/en-txt/NFS-HOWTO.txt.gz`).)

5.13.1 Desativando completamente os serviços RPC

A desativação do portmap é bem simples. Existem diversos diferentes métodos. O mais simples no sistema Debian 3.0 e mais novos é desistalar o pacote `portmap`. Se estiver executando uma versão antiga do Debian, terá que desativar o serviço como visto em 'Desabilitando daemons de serviço' on page 33, porque o programa é parte do pacote `net-base` (que não pode ser removido sem quebrar o sistema).

Isto de fato remove cada link relacionado ao portmap em `/etc/rc${runlevel}.d/`, que é algo que pode fazer manualmente. Outra possibilidade é executar um `chmod 644 /etc/init.d/portmap`, mas isto mostrará uma mensagem de erro durante a inicialização. Você também poderá comentar a parte `start-stop-daemon` no script `/etc/init.d/portmap`.

5.13.2 Limitando o acesso a serviços RPC

Infelizmente em alguns casos a remoção dos serviços RPC não é uma opção. Alguns serviços de desktop locais (notavelmente o `fam` da SGI) são baseados em RPC e assim precisam de

um portmapper local. Isto significa que sob algumas situações, os usuários que estiverem instalando um ambiente de desktop (como o GNOME) instalarão também o portmapper.

Existem diversas formas de limitar o acesso ao portmapper e aos serviços de RPC:

- Bloqueando o acesso as portas usadas por estes serviços com um firewall local (veja 'Adicionando capacidades de firewall' on this page).
- Bloquear o acesso a estes serviços usando tcp wrappers, pois o portmapper (e alguns serviços RPC) são compilados com a `libwrap` (veja 'Usando os tcpwrappers' on page 59. Isto significa que você poderá bloquear o acesso a eles através do `hosts.allow` e `hosts.deny` na configuração do tcp wrappers.
- Desde a versão 5-5, o pacote `portmap` pode ser configurado para somente realizar conexões na interface loopback. Para fazer isto, modifique o arquivo `/etc/default/portmap`, e descomente a seguinte linha: `#OPTIONS="-i 127.0.0.1"` e reinicie o portmapper. Isto é suficiente para permitir que serviços RPC locais funcionem enquanto ao mesmo tempo evite que sistemas remotos os acessem (no entanto, veja 'Desativando assuntos relacionados a weak-end de máquinas' on page 75.

5.14 Adicionando capacidades de firewall

O sistema Debian GNU/Linux tem as capacidades embutidas fornecidas pelo kernel do GNU/Linux. Isto significa que se você instalar o sistema potato (Debian 2.2), que vem com o kernel padrão 2.2, você terá as capacidades do firewall `ipchains` no kernel, você precisará ter o pacote `ipchains`, que deverá, devido a sua prioridade, já estar instalado. Se estiver instalando o sistema woody (Debian 3.0), que vem com o kernel padrão 2.4, você terá o firewall `iptables` (netfilter) disponível. A principal diferença entre o `ipchains` e `iptables` é que o últimos é baseado em *inspeção de estado de pacotes* que lhe oferece configurações mais seguras (e fáceis de construir) de filtragem.

5.14.1 Fazendo um firewall no sistema local

Você poderá usar regras de firewall como uma forma de restringir o acesso a seu sistema local e, até mesmo, limitar comunicações feitas através dele. As regras de firewall também podem ser usadas para proteger processos que podem não estar corretamente configurados, *não* fornecendo serviços para algumas redes, endereços IP, etc...

No entanto, este passo é mostrado por último neste manual basicamente porque é *muito* melhor não depender solenemente das capacidades de firewall para proteger um dado sistema. A segurança em um sistema é feita através de camadas, o firewall deve ser a última a ser adicionada, uma vez que todos os serviços foram ajustados para serem mais seguros. Você pode facilmente imaginar uma configuração em que o administrador descuidadamente remove as regras de firewall por alguma razão (problemas com a configuração, descuido, erro humano

...), este sistema pode estar aberto para um ataque se não existir outro reforço no sistema para protege-lo.

Por outro lado, tendo regras de firewall no sistema local também evita que coisas ruins aconteçam. Até mesmo se os serviços fornecidos estão configurados de forma segura, um firewall pode proteger de má configurações ou de serviços instalados recentemente que ainda não foram configurados adequadamente. Também, uma configuração forte evitará que cavalos de tróia *chamem a origem* de funcionarem a não ser que o código do firewall seja removido. Note que um intruso *não* precisa de acesso de superusuário para instalar um cavalo de tróia localmente que pode ser controlado remotamente (pois a escuta a porta é permitido caso não sejam portas privilegiadas e as capacidades não foram removidas).

Assim, uma configuração apropriada de firewall é aquela com a política padrão deny, que é:

- conexões de entrada são permitidas somente para serviços locais por máquinas permitidas.
- conexões de saída somente são permitidas para serviços usados pelo seu sistema (DNS, web browsing, pop, email....) [6]
- a regra forward bloqueia tudo (a não ser que esteja protegendo outros sistemas, veja abaixo).
- todas as outras conexões de entrada ou saída são negadas.

5.14.2 Usando um firewall para proteger outros sistemas

Um firewall também pode ser instalado no Debian para proteger, com regras de filtragem, o acesso a sistemas *através* dela, limitando sua exposição na Internet. O firewall pode ser configurado para evitar que sistemas de fora da rede local acesse serviços (portas) que não são públicas. Por exemplo, em um servidor de mensagens, somente a porta 25 (onde o serviço de e-mail foi definido) precisa ser acessada de fora. Um firewall pode ser configurado para, até mesmo se existem outros serviços disponibilizados publicamente, descartar qualquer pacote (isto é conhecido como *filtragem*) direcionado a máquina.

Você pode até mesmo configurar a máquina Debian GNU/Linux como uma firewall bridge, i.e. um firewall de filtragem completamente transparente para a rede que deixa de lado um endereço IP e assim não pode ser atacada diretamente. Dependendo do kernel que tiver instalado, você poderá precisar fazer a instalação do patch de bridge no firewall e então ir para a seção *802.1d Ethernet Bridging* quando estiver configurando o kernel e uma nova opção *netfilter (firewalling) support*. Veja 'Configurando uma ponte firewall' on page 195 para mais detalhes sobre como fazer isto em um sistema Debian GNU/Linux).

[6] De forma diferente de firewalls pessoais em outros sistemas operacionais, o Debian GNU/Linux (ainda) não fornece uma interface de geração de firewall que possa fazer regras de limitação por processo ou usuário. No entanto, o código do iptables pode fazer isto (veja o módulo owner na página de manual `iptables(8)`)

5.14.3 Configurando o firewall

É claro que a configuração do firewall é sempre dependente de sistema e rede. Um administrador deverá conhecer de antemão qual é a estrutura da rede e os sistemas que deseja proteger, os serviços que precisam ser acessados e se ou não outras considerações de rede (como NAT ou roteamento) devem ser levadas em conta. Seja cuidadoso quando configurar seu firewall, como Laurence J. Lane diz no pacote `iptables`:

As ferramentas podem ser facilmente mal utilizadas, causando uma enorme quantidade de peso na consciência e cortando o acesso a um sistema. Não é terrivelmente incomum para um administrador de sistemas remotos travar si próprio fora de um sistema centenas de milhares de milhas de distância. É também possível que alguém deixe ele próprio fora de um computador em que o teclado está sob seus dedos. Por favor, use com a devida precaução.

Lembre-se disto: apenas a instalação do `iptables` (ou do antigo código de firewall) não oferece qualquer proteção, apenas fornece o programa. Para ter um firewall, você precisa *configurá-lo!*

Se não souber muito sobre firewall, leia o Firewalling-HOWTO que pode ser encontrado no pacote `doc-linux-text` (outros formatos de documentos também estão disponíveis). Veja 'Esteja ciente dos problemas gerais de segurança' on page 25 para mais referências (gerais).

Fazendo pelo método Debian

Se estiver usando o Debian 3.0, você notará que tem o pacote `iptables` instalado. Este é para suporte da implementação netfilter de kernels 2.4.4 e superiores. Pois apenas após a instalação o sistema pode não *saber* que regras de firewall (regras de firewall são bastante dependentes de sistema) você tem para ativar o iptables. No entanto, os scripts foram configurados de uma forma que o administrador possa configurar as regras de firewall e então ter os scripts de inicialização sempre *aprendendo-as* e usando sempre como configuração do firewall.

Para fazer isto você deverá:

- Configurar o pacote, assim ele será iniciado com o sistema. Nas versões novas (desde a 1.2.6a-1) isto é feito quando o pacote é instalado. Você poderá configurá-lo após isto com `dpkg-reconfigure -plow iptables`. *Nota*: em versões antigas, isto pode ser feito editando-se o arquivo `/etc/default/iptables` e verificando se a variável `enable_iptables_initd` foi definida para *true* (ativo).
- crie uma configuração de firewall usando o `iptables`, você poderá usar a linha de comando (veja `iptables(8)`) ou algumas outras ferramentas fornecidas pelos pacotes de Firewall do Debian (veja 'Usando pacotes de Firewall' on the next page). Você precisará criar um conjunto de regras de firewall para ser usado quando o firewall estiver em estado *ativo* e outro para ser usado no estado *inativo* do firewall (podem ser simplesmente regras vazias).
- salve as regras que criou usando o `/etc/init.d/iptables save_active` e `/etc/init.d/iptables save_inactive` executando estes scripts com as regras de firewall que deseja iniciar.

Assim que tiver terminado, sua configuração de firewall estará salva no diretório `/var/lib/iptables/` e será executado quando o sistema inicializar (ou quando executar o script initd com os argumentos *start* e *stop*). Por favor note que a configuração padrão do Debian inicia o código de firewall em níveis de execução multiusuário (2 a 5) e em breve (10). Também, ele é interrompido em modo monousuário (1), altere isto caso não confira com suas políticas locais.

Se não tiver uma dica de como configurar regras de firewall manualmente, consulte o documento *Packet Filtering HOWTO* e *NAT HOWTO* fornecidas pelo `iptables` para leitura offline em `/usr/share/doc/iptables/html/`. O arquivo de configuração `/etc/default/iptables` também fornece várias informações a respeito deste pacote.

Usando pacotes de Firewall

A configuração manual de um firewall pode ser complicada para o administrador novato (e muitas vezes para até mesmo o expert). No entanto, a comunidade de software livre tem criado um número de ferramentas que podem ser usadas para configurar facilmente um firewall local. Esteja avisado desde já que algumas destas ferramentas são orientadas somente para a proteção local (também chamadas de *firewall pessoal*) e algumas são mais versáteis e podem ser usadas para configurar regras complexas para proteger todas as redes.

Alguns softwares que podem ser usados para configurar regras de firewall em um sistema Debian são:

- `firestarter` orientado a usuários finais, inclui um assistente para definir regras de firewall rapidamente.
- `knetfilter`
- `fwbuilder` uma GUI orientada a objetos que inclui compiladores de políticas para várias plataformas de firewalls incluindo o iptables assim como listas de acesso do roteador. A funcionalidade completa do fwbuilder também está disponível através da linha de comando
- `shorewall` que oferece suporte a IPsec com um suporte bem limitado para controle de tráfego também como uma definição de regras de firewall.
- `mason`, que propõe regras de firewall baseados no tráfego de rede que seu sistema "enxerga".
- `bastille` (entre os passos de fortalecimento que podem fazer as novas versões do bastille, é a possibilidade de se adicionar regras de firewall ao sistema que serão executadas na inicialização)
- `guarddog`, um pacote de configuração de firewall baseada no KDE (alternativa/competidor ao pacote knetfilter)
- `ferm`
- `fwctl`

- `easyfw`
- `firewall-easy`
- `ipac-ng`
- `gfcc`
- `lokkit` ou `gnome-lokkit`

Os últimos pacotes: gfcc,firestarter e knetfilter são interfaces de administração GUI usando ou o GNOME (os dois primeiros) ou o KDE (o último) que são muito mais orientados a usuários (para usuários domésticos) que outros pacotes da lista que são mais orientadas a administradores.

Esteja já avisado que alguns pacotes destacados anteriormente irão provavelmente introduzir a scripts de firewall que serão executados quando o sistema for inicializado, isto sem dúvida alguma conflitará com a configuração padrão (se estiver configurada) e terá efeitos indesejados. Normalmente os scripts de firewall que são executados por último serão os que configurarão o firewall do sistema (que pode não ser o que você deseja). Consulte a documentação do pacote e use ou uma desta configurações. Geralmente, outros programas que te ajudam a configurar regras de firewall podem pesquisar outros arquivos de configuração.

FIXME: Adicionar mais informações a respeito destes pacotes

FIXME: Procure por informações sobre firewall no Debian e o que/como fazer sua alteração para outras distribuições.

FIXME: Onde o código de firewall personalizado poderá ser ativado (FAQ padrão na debian-firewall?)

FIXME: Adicionar informações sobre Zorp (`http://www.balabit.hu/downloads/zorp/stable/deb/`) no Debian (veja Bug #88347 (`http://bugs.debian.org/88347`). Os pacotes do Debian são fornecidos, mas eles dependem da libglib1.3 que não está disponível na distribuição Debian.

Capítulo 6

Fortalecimento automático de sistemas Debian

Após ler todas as informações dos capítulos anteriores você deve estar pensando "Eu tenho que fazer muitas coisas para ter meu sistema fortalecido, estas coisas não poderiam ser automatizadas?". A resposta é sim, mas tenha cuidado com ferramentas automatizadas. Algumas pessoas acreditam que uma ferramenta de fortalecimento não elimina a necessidade de uma boa administração. Assim não seja tolo em pensar que pode automatizar todo o processo e corrigir todos os problemas relacionados a ele. Segurança é um processo progressivo no qual o administrador deve estar participando e não somente ficar a espera deixando que as ferramentas façam todo o trabalho, já que nenhuma ferramenta poderia fazer: todas as implementações de políticas de segurança possíveis, cobrindo todos os ataques e todos os ambientes.

Desde a woody (Debian 3.0) existem dois pacotes específicos que são úteis para o fortalecimento do sistema. O pacote `harden` que tem sua estratégia baseada na dependência de pacotes para rapidamente instalar pacotes de segurança importantes e remover os que tem problemas de segurança, a configuração de pacotes deve ser feita pelo administrador. O pacote `bastille` que implementa uma dada política de segurança no sistema local baseada na configuração anterior do administrador (a construção da configuração pode ser feita usando um processo guiado com questões simples no estilo sim/não).

6.1 Harden

O pacote `harden` tenta tornar a instalação e administração fácil para máquinas que precisam de boa segurança. Este pacote deve ser usado por pessoa que desejam uma ajuda rápida para melhorar a segurança do sistema. Para fazer isto, ele conflita com pacotes com falhas conhecidas, incluindo (mas não limitado a): falhas conhecidas de segurança (como estouro de buffer), uso de senhas em texto plano, esquecimento de controle de acesso, etc. Ele automaticamente instala algumas ferramentas que aumentam a segurança de alguma forma: ferramentas de detecção de intrusão, ferramentas de análise de segurança, etc. O Harden instala os seguintes pacotes *virtuais* (por exemplo pacotes sem conteúdo, que apenas dependem de outros):

- `harden-tools`: ferramentas para aumentar a segurança do sistema (verificadores de integridade, detectores de intrusão, patches de kernel...)
- `harden-doc`: fornece este mesmo manual e outros pacotes de documentação relacionados a segurança.
- `harden-environment`: ajuda a configurar um ambiente fortalecido (atualmente vazio).
- `harden-servers`: servidores remotos considerados inseguros por alguma razão.
- `harden-clients`: exclui clientes considerados inseguros por alguma razão.
- `harden-remoteflaws`: exclui pacotes com furos de segurança conhecidos que podem ser usados por um invasor para comprometer o sistema (usa *Conflicts:* sobre versões).
- `harden-localflaws`: exclui pacotes com problemas de segurança que podem ser usados por um invasor local para comprometer o sistema (usa *Conflicts:* sobre versões).
- `harden-remoteaudit`: ferramentas para fazer a auditoria remota de um sistema.

Tenha cuidado se tiver um programa que precisa (e que não deseja desinstalar por alguma razão) e que ele conflite com alguns dos pacotes acima, assim não será capaz de fazer uso completo do `harden`. Os pacotes do harden não fazem (diretamente) coisa alguma. Eles realizam, no entanto, conflitos com pacotes conhecidamente inseguros. Desta forma, o sistema de empacotamento da Debian não aprovará a instalação destes pacotes. Por exemplo, quando tenta instalar um daemon telnet com o `harden-servers` o `apt` dirá:

```
# apt-get install telnetd
The following packages will be REMOVED:
 harden-servers
The following NEW packages will be installed:
telnetd
Do you want to continue (Y/n)
```

Isto deverá deixar o administrador mais tranqüilo, reconsiderando suas ações que serão tomadas.

6.2 Bastille Linux

O Bastille Linux (`http://www.bastille-unix.org`) é uma ferramenta de fortalecimento originalmente orientada sobre as distribuições RedHat e Mandrake. No entanto o pacote `bastille` fornecido com a Debian (desde a woody) é adaptado para fornecer a mesma funcionalidade para o sistema Debian GNU/Linux.

O Bastille pode ser usado com diferentes interfaces com o usuário (todas são documentadas em sua própria página de manual no pacote da Debian) que permite o administrador a:

- Responder questões passo a passo sobre a segurança requerida pelo seu sistema (usando `InteractiveBastille(8)`)
- Usar a configuração padrão de segurança (entre três: Fraca, Moderada, e Paranóica) em um determinado sistema (servidor e estação de trabalho) e deixar o Bastille decidir que política de segurança que será implementada (usando `BastilleChooser(8)`)
- Pegar um arquivo de configuração pré-definido (deve ser fornecido pelo Bastille ou feito pelo administrador) e implementar uma política de segurança determinada (usando `AutomatedBastille(8)`)

Capítulo 7

Infraestrutura do Debian Security

7.1 O time Debian Security

O Debian tem um Security Team (Time de Segurança), composto por cinco membros e duas secretárias que manipulam a segurança na distribuição *stable* (estável). Manipular a segurança significa que eles acompanham as vulnerabilidades que aparecem nos software (vendo foruns como bugtraq o vuln-dev) e determinam se a distribuição *stable* é afetada por eles.

O Debian Segurity Team também é o contato para problemas que são coordenados pelos desenvolvedores ou organizações como CERT (`http://www.cert.org`) que podem afetar muitos vendedores. Isto é, quando os problemas não são específicos do Debian. Existem dois contatos com o Security Team:

- team@security.debian.org (`mailto:team@security.debian.org`) o qual só é lido pelos membros do security team .
- security@debian.org (`mailto:security@debian.org`) o qual é lido por todos os desenvolvedores Debian (incluindo o security team). Emails enviados para esta lista não são publicados na internet (esta não é uma lista de email pública).

Informações sensíveis devem ser enviadas para o primeiro email e, em alguns casos, deve ser encriptada com a Debian Security Contact key (key ID 363CCD95).

Quando um provável problema for recebido pelo Security Team, ele investigará se a distribuição *stable* foi afetada e, caso positivo, uma correção será feita no código fonte base. Esta correção algumas vezes incluirá algum patch (que normalmente é mais recente que a versão distribuída pelo Debian). Após o teste da correção, novos pacotes são preparados e publicados em `security.debian.org` e podem ser baixados com o `apt` (veja 'Executar uma atualização de segurança' on page 40). Ao mesmo tempo um *Debian Security Advisory* (DSA) é publicado no web site e enviado para a listas de email incluindo debian-security-announce (`lists.debian.org/debian-security-announce`) e bugtraq.

Outras perguntas frequentes do Debian Security Team podems er encontradas em 'Questões relacionadas ao time de segurança da Debian' on page 176.

7.2 Debian Security Advisories

Debian Security Advisories são avisos emitidos quandos uma vulnerabilidade de segurança que afeta um pacote Debian é descoberta. Estes avisos, assinados por um membro do Security Team, inclui informação das versões afetadas assim como a localização das atualizações e seus MD5sums. Esta informação consiste de:

- número da versão para correção.
- tipo de problema.
- se ele é remoto ou localmente explorável.
- pequena descriçãod do pacote.
- descrição do problema.
- descrição da exploração.
- descrição da correção.

DSAs são publicados em Debian's mainserver frontpage (`http://www.debian.org/`) e em Debian security pages (`http://www.debian.org/security/`). Normalmente isto não é feito até a reconstrução diária do website, então eles podem não estar presentes imediatamente, o canal preferido é a debian-security-announce mailing list.

Usuários interessados podem, porém, usar o canal RDF para baixar automaticamente as DSAs para seu computador. Algumas aplicações, como o `Evolution` (um cliente de email e assistente de informações pessoais) e o `Multiticker` (um applet do GNOME), podem ser usados para baixar os avisos automaticamente. O canal RDF está disponível em `http://www.debian.org/security/dsa.rdf`.

Os DSAs publicados no website podem ser atualizados após enviados para as listas de email. Uma atualização comum é adicionada através de referências ao banco de dados de vulnerabilidades de segurança. Além disso, traduções [1] dos DSAs não são enviadas para as listas de email mas são diretamente incluídas no site.

7.2.1 Referências sobre vulnerabilidades

Debian fornece uma referência completa em crossreferenced table (`http://www.debian.org/security/crossreferences`) incluindo todas as remomendações publicadas desde 1998. Esta tabela é fornecida em complemento a reference map available at CVE (`http://cve.mitre.org/cve/refs/refmap/source-DEBIAN.html`).

Você notará que esta tabela fornece referências aos bancos de dados como Bugtraq (`http://www.securityfocus.com/bid`), CERT/CC Advisories (`http://www.cert.`

[1] Traduções estão disponíveis em dez idiomas

org/advisories/) and US-CERT Vulnerability Notes Database (http://www.kb.cert.org/vuls) assim como aos nomes CVE (veja abaixo). Estas referências são fornecidas para uso, porém apenas referências CVE são periodicamente revisadas e incluídas. Este recurso foi adicionado ao website em junho de 2002.

Uma das vantagens de adicionar referências ao banco de dados de vulnerabilidades é que:

- torna fácil aos usuários Debian ver e tratar com recomendações publicadas que já tenham sido resolvidas pelo Debian.
- administradores de sistema podem aprender mais sobre vulnerabilidades e seu impacto através das referências.
- esta informação pode ser usada para a checagem de vulnerabilidades referentes ao CVE e detectar avisos falsos. (veja 'O scanner de vulnerabilidade X diz que meu sistema Debian é vulnerável!' on page 172).

7.2.2 Compatibilidade CVE

As recomendações de segurança, Debian Security Advisories eram declared CVE-Compatible (http://www.debian.org/security/CVE-certificate.jpg)[2] em fevereiro de 2004.

Desenvolvedores Debian entendeream que precisavam fornecer precisas e atualizadas informações de segurança para a distribuição, permitindo aos usuários gerenciar o risco associado com novas vulnerabilidades. CVE fornece referências padronizadas que permitem aos usuários desenvolver um CVE-enabled security management process (http://www.cve.mitre.org/compatible/enterprise.html).

O projeto Common Vulnerabilities and Exposures (CVE) (http://cve.mitre.org) é mantido pela MITRE Corporation e fornece uma lista de nomes padronizados para vulnerabilidades e exposições de segurança.

Debian acredita que fornecer aos usuários informações relacionadas a segurança que afetem a distribuição é extremamente importante. A inclusão dos nomes CVE em avisos ajudam os usuários a associar vulnerabilidades genéricas com atualizações específicas, com redução do tempo gasto para manusear as vulnerabilidades. Além disso, é fácil o gerenciamento da segurança em um ambiente onde já existem ferramentas que utilizam o CVE, como redes ou sistemas de detecção de invasão, ou ferramentas de avalição de vulnerabilidades, mesmo que elas não sejam baseadas em uma distribuição Debian.

Debian iniciou adicionando nomes CVE aos DSAs em junho de 2002 e agora fornecer para todos os DSAs lançados desde setembro de 1998 após a revisão iniciada em agosto de 2002. Todos os avisos podem ser recuperados do website do Debian e notícias relacionadas a novas vulnerabilidades incluindo nomes CVE se disponíveis na época de seu lançamento. Avisos associados com um dado nome CVE pode ser procurado diretamente através do search engine (http://search.debian.org/).

[2] O completo capability questionnaire (http://cve.mitre.org/compatible/phase2/SPI_Debian.html) estava disponível para o CVE

Usuários que querem procurar por um nome CVE em particular podem usar o sistema de busca disponível em debian.org para recuperar avisos disponíveis (em inglês e traduzidos para outros idiomas). Uma busca pode ser feita para um nome específico (como aviso CAN-2002-0001 (http://search.debian.org/?q=advisory+%22CAN-2002-0001%22ps=50o=1m=all)) ou para nomes parciais (como todos os avisos de 2002 para CAN-2002 (http://search.debian.org/?q=advisory+%22CAN-2002%22ps=50o=1m=all)). Observe que você precisa entrar com a palavra advisory junto com o nome CVE para recuperar apenas avisos de segurança.

Em alguns casos você pode nãO encontrar um CVE em avisos publicados porque:

- No Debian os produtos não são afetados pela vulnerabilidades.
- Ainda não existe uma viso abordando a vulnerabilidade (ele pode ter sido informado para a security bug (http://bugs.debian.org/cgi-bin/pkgreport.cgi?tag=security) mas uma correção ainda não ter sido testada a atualizada)
- Um aviso foi publicado antes que um CVE fosse assinado para a vulnerabilidade em questão (procure por uma atualização no web site)

7.3 Infraestrutura da segurança Debian

Uma vez que o Debian é normalmente suportado em um grande número de arquiteturas, administradores algumas ficam admirados se uma dada arquitetura levar mais tempo para receber atualizações de segurança. De fato, exceto em raras circunstâncias, atualizações estão disponíveis para todas as arquiteturas ao mesmo tempo.

Enquanto antigamente a tarefa de construir atualizações de segurança era feita a mão, hoje não é mais (como Anthony Towns descreve em a mail (http://lists.debian.org/debian-devel-announce/2002/debian-devel-announce-200206/msg00002.html), enviado para a lista debian-devel-announce em 6 de junho de 2002.)

Pacotes atualizados pelo time de segurança (para security.debian.org:/org/security.debian.org/queue/unchecked ou ftp://security.debian.org/pub/SecurityUploadQueue) tem suas assinaturas checada com um patch adequado dentro de quinze minutos, uma vez isto feito eles são adicionados a lista de auto construtores. Então, os pacotes podem ser disponibilizados para *todas* as arquiteturas num tempo de trinta minutos a uma hora do momento em que foram atualizados. Porém, atualizações de segurança são um pouco diferentes da atualização normal envidada pelos mantenedores de pacotes, uma vez que, em alguns casos, antes de ser publicadas, elas precisam esperar até serem testadas, um aviso ser escrito ou, ainda, precisam esperar uma semana ou mais para evitar publicação da falhar até que todos os vendedores tenham chance de corrigí-la.

Assim, a atualização de segurança trabalha da seguinte maneira (chamada *"Accepted-Autobuilding"*):

- Alguém encontra um problema de segurança.

- Alguém corrige o problema e atualiza security.debian.org (este *alguém* normalmente é um membro do Time de Segurança mas pode ser também um mantenedor de pacote com uma correção apropriada que contactou o time de segurança previamente). O Changelog inclui uma indicação *testing-security* ou *stable-security*.

- Ocorre o upload checado e processado por um sistema Debian e movido para queue/accepted, e buildds são notificados. Arquivos aqui podem ser acessados pelo time de segurança e (indiretamente) pelos buildds.

- O Security-enable buildds pega o pacote fonte (que tem prioridade sobre os builds normais), o constrói, e envia logs para o time de segurança.

- O time de segurança reproduz os logs, e novos pacotes construídos são enviados para queue/unchecked, onde são processados pelo sistema Debian, e movidos para queue/accepted.

- Quando o time de segurança verifica que o pacote fonte está aceitável (isto é, ele foi corretamente construído para todas as arquiteturas, corrigiu os problemas de segurança e não introduziu novos problemas) eles rodam um script que:

  – instala o pacote em um arquivo de segurança.
  – atualiza os pacotes, fontes e release files de security.debian.org de uma maneira normal (`dpkg-scanpackages`, `dpkg-scansources...`)
  – configura um aviso modelo que o time de seguranaça pode encerrar os trabalhos.
  – (opcionalmente) envia os pacotes para as atualizações adequadas e eles podem ser incluídos assim que for possível.

Este procedimento, antes feito a mão, foi testado e usado completamente durante o estágio freeze do Debian 3.0 Woody (Julho de 2002). Graças a esta infraestrutura do Security Team foi possível ter pacotes atualizados prontos para o apache e OpenSSH para todas as arquiteturas suportadas (quase vinte) em menos de um dia.

7.3.1 Guia dos desenvolvedores de atualizações de seguranaça

Este email foi enviado por Wichert Akkerman para Debian-devel-announce mailing list (`http://lists.debian.org/debian-devel-announce/2002/debian-devel-announce-200206/msg00004.html`) a fim de descrever o comportamento do desenvolvedor Debian para manipulação de problemas de segurança em seus pacotes. Ele está publicado aqui tanto para os desenvolvedores quanto os usuários entenderem melhor como a segurança é manipulada no Debian.

Por favor observe que a última referência para esta informação é Debian Developer's Reference (`http://www.debian.org/doc/manuals/developers-reference/ch-pkgs#bug-security`), esta seção será removida em futuro próximo.

Coordenando com o time de segurança

Se um desenvolvedor tem conhecimento de um problema de segurança, seja em seu pacote seja em outro, ele deve sempre contactar o time de segurança (através de team@security.debian.org). Eles mantém controle dos problemas de segurança, podem ajudar mantenedores, corrigir os problemas, são responsáveis por enviar os avisos e manter o security.debian.org.

Observe que os avisos de segurança não são feitos apenas para releases, não apenas para testing, unstable (veja 'Como a segurança é tratada na `testing` e `unstable`?' on page 178) ou distribuições antigas (veja 'Eu uso uma versão antiga da Debian, ela é suportada pelo time de segurança?' on page 178).

Tomando conhecimento dos problemas de segurança

Como um desenvolvedor toma conhecimento de um problema de segurança:

- ele observa em um fórum público (mailing list, website, etc.):
- alguém arquiva um bugreport (um tag *Security* deve ser usada, ou adicionada)
- alguém o informa via email.

Nos dois primeiros casos a informação é pública e é importante ter uma correção o mais rápido possível. Em último caso porém ela pode não ser uma informação pública. Neste caso existem poucas opções para tratar o problema:

- Se é um problema trivial (como arquivos inseguros temporários) não há necessidade de manter o problema secreto e a correção deve ser feita e lançada.
- se o problema é grave (exploração remota, possibilitando adquirir privilégios de root) é preferível compartilhar a informação com outros vendedores e coordenar o lançamento. O time de segurança mantém contato com várias organizações e indivíduos e é cuidadoso com isto.

Em todos os casos, se a pessoa que reporta o problema pede para não divulgar a informação, deve ser respeitada, com execeção óbviade informar ao time de segurança (o desenvolvedor deve estar certo que ele disse ao time de segurança que a informação não deve ser divulgada).

Por favor observe que se o segredo é necessário o desenvolvedor pode também não atualizar uma correção para a unstable (ou qualquer outra), uma vez que o chagelog para a unstable é uma informação pública.

Existem duas razões para o lançamento da informação mesmo se o segredo é solicitado: o problema torna-se conhecido por muitos, ou a informação torna-se pública.

Construindo um pacote

A mais importante guideline quando fazendo um novo pacote que corrige um problema de segurança é mazer o mínimo de alterações necessário. As pessoas sabem exatamente o comportamento de um lançamento, assim qualquer alteração feita pode quebrar o sistema de alguém. Isto é especialmente verdade para bibliotecas: o desenvolvedor deve estar certo de nunca alterar a API ou a ABI, mesmo que seja uma pequena mudança.

Isto significa que mudar para uma nova versão não é uma boa solução, em vez disto só as alterações relevantes devem ser feitas. Geralmente os mantenedores estão dispostos a ajudar que precisa, se não, o time de segurança da Debian pode.

Em alguns casos não é possível fazer o backport de uma correção de segurança, por exemplo quando uma grande quantidade do código fonte precisa ser modificado ou reescrito. Se isto acontece pode ser necessário mover para uma nova versão, mas isto deve sempre ser coordenado com o time de segurança.

Relacionado a isto existe outro importante aspecto: desenvolvedores devem sempre testar suas alterações. Se existe uma falha que permita exploração, o desenvolvedor deve testar e verificar se ela aconteceu em um pacote não corrigido ou em um pacote corrigido. O desenvolvedor deve tentar o uso normal também, algumas vezes uma correção de segurança pode quebrar o uso normal sutilmente.

Finalmente algumas coisas que os desenvolvedores devem ter em mente:

- Esteja certo que você assinalou a distribuição correta em seu debian/changelog. Para a distrituição estável (stable) você deve assinalar como stable-security e para a distribuição em teste, testing-security. Não assinale <codename>-proposed-updates.
- Verifique o número da versão. Ele deve ser maior que o pacote atual, mas menor que versões do pacote em distribuições anteriores. Para a distribuição testing isto sigfinifica que deve haver uma versão maior na distribuição unstable. Se ainda não existe (testing e unstable tem a mesma versão por exemplo) atualize a nova versão para a unstable primeiro.
- Não faça atualizações source-only se seu pacote tem alguns pacotes binary-all. A infraestrutura de construção não construirá aqueles.
- Quando compilar um pacote faço isto em um sistema limpo, o qual tem só tem instalados pacotes da distribuição para a qual você está construindo. Se você não tem um sistema assim pode tentar a debian.org machine (veja http://db.debian.org/machines.cgi) ou configurar um chroot (os pacotes `pbuilder` e `debootstrap` podem ajudar neste caso).

Realizando o uplaod com as correções de segurança

Após o desenvolvedor ter criado e testado um novo pacote ele precisa realizar o upload pois assim a correçaõ será instalada nos archives. Por segurança os arquivos para upload devem ser colocados em ftp://security.debian.org/pub/SecurityUploadQueue/ .

Uma vez que o upload foi aceito o pacote será automaticamente reconstruído para todas as arquiteturas e armazenado para verifiração pelo time de segurança.

Uploads aguardando por aceitação e verificação só são acessíveis pelo time de segurança. Isto é necessário uma vez que podem ser correções para problemas de segurança que ainda não foram descobertos.

SE um mebro do time de segurança aceita um pacote ele será instalado em security.debian.org assim como o apropriado <codename>-proposed-updates em ftp-master ou non-US archive.

O aviso de segurança

Os avisos de segurança são escritos e postados pelo time de segurança. Porém, eles certamente não pensam se um mantenedor pode fornecer o texto para eles (pelo menos uma parte). As informações que devem fazer parte de um aviso são descritas em 'Debian Security Advisories' on page 114.

7.4 Assinatura de pacote no Debian

Esta seção também pode ser chamada "como atualizar seu sistema Debian GNU/Linux em segurança" e mereçe sua própria seção basicamente porque é uma parte importante da infraestrutura de segurança. Assinatura de pacote é uma coisa importante porque evita alterações de pacotes distribuídos em mirros. Atualização automatica de software é um recurso importante mas também é importante remover ameaças de segurança que poderiam ajudar a distribuir cavalos de tróia e comprometer os sistemas durante as atualizações. [3].

Atualmente (maio de 2005) o Debian não fornece assinatura de pacotes para as distribuições lançadas *woody* ou *sarge* (3.0 ou 3.1). Elas não possuem este recurso. Existe uma solução para isto que será fornecida na próxima distribuição (codename *etch*). Este novo recurso estará disponível no apt 0.6 (atualmente disponível numa distribuição *experimental*, veja 'Pacotes experimentais apt' on page 128).

Isto é melhor descrito em Strong Distribution HOWTO (`http://www.cryptnet.net/fdp/crypto/strong_distro.html`) por V. Alex Brennen.

7.4.1 O esquema proposto para checagem de assinatura dos pacotes

O esquema atual para checagem da assinatura dos pacotes usando `apt` é:

- o arquivo lançado incluirá o md5sum do Packages.gz (que contém os md5sum dos pacotes) e será assinado. A assinatura é de uma fonte certificada.

[3] Alguns sistemas operacionais já tiveram problemas com atualizações automáticas como Mac OS X Software Update vulnerabity (`http://www.cunap.com/~hardingr/projects/osx/exploit.html`). FIXME: probably the Internet Explorer vulnerability handling certificate chains has an impact on security updates on Microsoft Windows.

- A arquivo assinado é baixado pelo 'apt-get update' a armazenado com o Packages.gz.
- Quando o pacote está sendo instalado, ele primeiro é baixado, então o md5sum é gerado.
- A assinatura é checada (assinatura ok) e extraído o md5sum do arquivo Pakages.gz, este por sua vez é gerado e (se ok) o md5sum do pacote baixado é extraído.
- Se o md5sum do pacote baixado é o mesmo que o do Packages.gz, o pacote será instalado. Caso contrário o administrador será alertado e o pacote será colocado num cache (e o administrador pode decidir se instalará o pacote ou não). Se o pacote não estiver no Packages.gz e o administrador tiver configurado o sistema para só instalar pacotes checados, o pacote não será instalado.

A sequência seguinte de checagens MD5 do `apt` é capaz de verificar se o pacote origina de um release específico. Isto é menos flexível que a assinatura de cada pacote, mas pode ser combinada com este esquema também (veja abaixo).

Atualmente, este esquema é fully implemented (http://lists.debian.org/debian-devel/2003/debian-devel-200312/msg01986.html) no apt 0.6 par mais informações veja 'Pacotes experimentais apt' on page 128. Pacotes que fornecem um front-end para o apt precisam ser modificados para adaptar este novo recurso, isto é o caso do `aptitude` o qual tem feito modified (http://lists.debian.org/debian-devel/2005/03/msg02641.html) para adaptar-se a este esquema.

Assinatura de pacotes foi discutido no Debian por um bom tempo, para mais informações leia: http://www.debian.org/News/weekly/2001/8/ e http://www.debian.org/News/weekly/2000/11/.

7.4.2 Checando releases das distribuições

Caso você queira adicionar os novos recursos de checagem de segurança e não queira rodar a versão experimental do apt (embora nõs realmente apreciemos o teste dele) você pode usar o script abaixo, fornecido por Anthony Towns. Este script pode automaticamente fazer algumas novas checagens de segurança para permitir ao usuário certificar-se que o software que está baixando corresponde aquele distribuído pelo Debian. Isto é para desenvolvedores Debian usarem em sistemas sem a funcionalidade de uploading dos sistemas tradicionais, ou mirrors que tem quase tudo mas não como o Debian, ou mirrors que fornecem dados da versão unstable sem conhecimento dos problemas de segurança.

Este código, renomeado como `apt-check-sigs`, deve ser usado da seguinte maneira:

```
# apt-get update
# apt-check-sigs
(...resultados...)
# apt-get dist-upgrade
```

Primeiro você precisa:

- pagar as chaves para assinar os Release files, http://ftp-master.debian.org/ziyi_key_2003.asc e adicioná-las a ~/.gnupg/trustedkeys.gpg (que gpgv usa por padrão).

```
gpg --no-default-keyring --keyring trustedkeys.gpg --import ziyi_key_
```

- remover qualquer linha do /etc/apt/sources.list que não usa a estrutura normal de "dist", ou alterar o script para ele trabalhe com elas.
- estar preparado para ignorar o fato que o Debian security updates não assinou os Release files, e que os Sources files não tem os checksums apropriados no Release file (ainda).
- estar preparado para checar se as fontes estão assinadas com as chaves apropriadas.

Este é o código de exemplo do apt-check-sigs, a última versão pode ser conseguida de http://people.debian.org/~ajt/apt-check-sigs. Este código atualmente está em beta, para mais informações leia http://lists.debian.org/debian-devel/2002/debian-devel-200207/msg00421.html.

```
#!/bin/bash

# Copyright (c) 2001 Anthony Towns <ajt@debian.org>
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.

rm -rf /tmp/apt-release-check
mkdir /tmp/apt-release-check || exit 1
cd /tmp/apt-release-check

>OK
>MISSING
>NOCHECK
>BAD

arch=`dpkg --print-installation-architecture`

am_root () {
        [ `id -u` -eq 0 ]
```

```
}

get_md5sumsize () {
        cat "$1" | awk '/^MD5Sum:/,/^SHA1:/' |
          MYARG="$2" perl -ne '@f = split /\s+/; if ($f[3] eq $ENV{"MYARG"})
print "$f[1] $f[2]\n"; exit(0); }'
}

checkit () {
        local FILE="$1"
        local LOOKUP="$2"

        Y="`get_md5sumsize Release "$LOOKUP"`"
        Y="`echo "$Y" | sed 's/^ *//;s/  */ /g'`"

        if [ ! -e "/var/lib/apt/lists/$FILE" ]; then
                if [ "$Y" = "" ]; then
                        # No file, but not needed anyway
                        echo "OK"
                        return
                fi
                echo "$FILE" >>MISSING
                echo "MISSING $Y"
                return
        fi
        if [ "$Y" = "" ]; then
                echo "$FILE" >>NOCHECK
                echo "NOCHECK"
                return
        fi
        X="`md5sum < /var/lib/apt/lists/$FILE | cut -d\  -f1` `wc -c < /var/l
/apt/lists/$FILE`"
        X="`echo "$X" | sed 's/^ *//;s/  */ /g'`"
        if [ "$X" != "$Y" ]; then
                echo "$FILE" >>BAD
                echo "BAD"
                return
        fi
        echo "$FILE" >>OK
        echo "OK"
}

echo
echo "Checking sources in /etc/apt/sources.list:"
echo "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"
echo
```

```
(echo "You should take care to ensure that the distributions you're downloadi
"
echo "are the ones you think you are downloading, and that they are as up to"
echo "date as you would expect (testing and unstable should be no more than"
echo "two or three days out of date, stable-updates no more than a few weeks"
echo "or a month)."
) | fmt
echo

cat /etc/apt/sources.list |
  sed 's/^ *//' | grep '^[^#]' |
  while read ty url dist comps; do
        if [ "${url%%:*}" = "http" -o "${url%%:*}" = "ftp" ]; then
                baseurl="${url#*://}"
        else
                continue
        fi

        echo "Source: ${ty} ${url} ${dist} ${comps}"

        rm -f Release Release.gpg
        lynx -reload -dump "${url}/dists/${dist}/Release" >/dev/null 2>&1
        wget -q -O Release "${url}/dists/${dist}/Release"

        if ! grep -q '^' Release; then
                echo "  * NO TOP-LEVEL Release FILE"
                >Release
        else
                origline=`sed -n 's/^Origin: *//p' Release | head -1`
                lablline=`sed -n 's/^Label: *//p' Release | head -1`
                suitline=`sed -n 's/^Suite: *//p' Release | head -1`
                codeline=`sed -n 's/^Codename: *//p' Release | head -1`
                dateline=`grep "^Date:" Release | head -1`
                dscrline=`grep "^Description:" Release | head -1`
                echo "  o Origin: $origline/$lablline"
                echo "  o Suite: $suitline/$codeline"
                echo "  o $dateline"
                echo "  o $dscrline"

                if [ "${dist%%/*}" != "$suitline" -a "${dist%%/*}" != "$codel
                        echo "  * WARNING: asked for $dist, got $suitline/$co
                fi

                lynx -reload -dump "${url}/dists/${dist}/Release.gpg" >/dev/n
                wget -q -O Release.gpg "${url}/dists/${dist}/Release.gpg"
```

```
            gpgv --status-fd 3 Release.gpg Release 3>&1 >/dev/null 2>&1 |
                    if [ "$gpgcode" = "GOODSIG" ]; then
                        if [ "$err" != "" ]; then
                            echo "  * Signed by ${err# } key: ${rest#* }"
                        else
                            echo "  o Signed by: ${rest#* }"
                            okay=1
                        fi
                        err=""
                    elif [ "$gpgcode" = "BADSIG" ]; then
                        echo "  * BAD SIGNATURE BY: ${rest#* }"
                        err=""
                    elif [ "$gpgcode" = "ERRSIG" ]; then
                        echo "  * COULDN'T CHECK SIGNATURE BY KEYID: ${re
                        err=""
                    elif [ "$gpgcode" = "SIGREVOKED" ]; then
                        err="$err REVOKED"
                    elif [ "$gpgcode" = "SIGEXPIRED" ]; then
                        err="$err EXPIRED"
                    fi
                done
                if [ "$okay" != 1 ]; then
                    echo "  * NO VALID SIGNATURE"
                    >Release
                fi)
    fi
    okaycomps=""
    for comp in $comps; do
            if [ "$ty" = "deb" ]; then
                    X=$(checkit "`echo "${baseurl}/dists/${dist}/${comp}/
                    Y=$(checkit "`echo "${baseurl}/dists/${dist}/${comp}/
                    if [ "$X $Y" = "OK OK" ]; then
                            okaycomps="$okaycomps $comp"
                    else
                            echo "  * PROBLEMS WITH $comp ($X, $Y)"
                    fi
            elif [ "$ty" = "deb-src" ]; then
                    X=$(checkit "`echo "${baseurl}/dists/${dist}/${comp}/
                    Y=$(checkit "`echo "${baseurl}/dists/${dist}/${comp}/
                    if [ "$X $Y" = "OK OK" ]; then
                            okaycomps="$okaycomps $comp"
                    else
                            echo "  * PROBLEMS WITH component $comp ($X,
                    fi
            fi
    done
```

```
        [ "$okaycomps" = "" ] || echo "  o Okay:$okaycomps"
        echo
  done

echo "Results"
echo "~~~~~~~"
echo

allokay=true

cd /tmp/apt-release-check
diff <(cat BAD MISSING NOCHECK OK | sort) <(cd /var/lib/apt/lists && find . -

cd /tmp/apt-release-check
if grep -q ^ UNVALIDATED; then
    allokay=false
    (echo "The following files in /var/lib/apt/lists have not been validated.
    echo "This could turn out to be a harmless indication that this script"
    echo "is buggy or out of date, or it could let trojaned packages get onto
    echo "your system."
    ) | fmt
    echo
    sed 's/^/    /' < UNVALIDATED
    echo
fi

if grep -q ^ BAD; then
    allokay=false
    (echo "The contents of the following files in /var/lib/apt/lists does not
    echo "match what was expected. This may mean these sources are out of dat
    echo "that the archive is having problems, or that someone is actively"
    echo "using your mirror to distribute trojans."
    if am_root; then
        echo "The files have been renamed to have the extension .FAILED and"
        echo "will be ignored by apt."
        cat BAD | while read a; do
            mv /var/lib/apt/lists/$a /var/lib/apt/lists/${a}.FAILED
        done
    fi) | fmt
    echo
    sed 's/^/    /' < BAD
    echo
fi

if grep -q ^ MISSING; then
    allokay=false
```

```
    (echo "The following files from /var/lib/apt/lists were missing. This"
    echo "may cause you to miss out on updates to some vulnerable packages."
    ) | fmt
    echo
    sed 's/^/    /' < MISSING
    echo
fi

if grep -q ^ NOCHECK; then
    allokay=false
    (echo "The contents of the following files in /var/lib/apt/lists could no
    echo "be validated due to the lack of a signed Release file, or the lack"
    echo "of an appropriate entry in a signed Release file. This probably"
    echo "means that the maintainers of these sources are slack, but may mean
    echo "these sources are being actively used to distribute trojans."
    if am_root; then
        echo "The files have been renamed to have the extension .FAILED and"
        echo "will be ignored by apt."
        cat NOCHECK | while read a; do
            mv /var/lib/apt/lists/$a /var/lib/apt/lists/${a}.FAILED
        done
    fi) | fmt
    echo
    sed 's/^/    /' < NOCHECK
    echo
fi

if $allokay; then
    echo 'Everything seems okay!'
    echo
fi

rm -rf /tmp/apt-release-check
```

Você pode precisar aplicar o seguinte patch para *sid* uma vez que `md5sum` adiciona um '-' após o sum quando a entrada é stdin:

```
@@ -37,7 +37,7 @@
        local LOOKUP="$2"

        Y="`get_md5sumsize Release "$LOOKUP"`"
-       Y="`echo "$Y" | sed 's/^ *//;s/  */ /g'`"
+       Y="`echo "$Y" | sed 's/-//;s/^ *//;s/  */ /g'`"

        if [ ! -e "/var/lib/apt/lists/$FILE" ]; then
```

```
                if [ "$Y" = "" ]; then
@@ -55,7 +55,7 @@
                return
        fi
        X="`md5sum < /var/lib/apt/lists/$FILE` `wc -c < /var/lib/apt/lists/$F
-       X="`echo "$X" | sed 's/^ *//;s/  */ /g'`"
+       X="`echo "$X" | sed 's/-//;s/^ *//;s/  */ /g'`"
        if [ "$X" != "$Y" ]; then
                echo "$FILE" >>BAD
                echo "BAD"
```

7.4.3 Esquema alternativo de assinatura per-package

The additional scheme of signing each and every packages allows packages to be checked when they are no longer referenced by an existing Packages file, and also third-party packages where no Packages ever existed for them can be also used in Debian but will not be default scheme.

Este esquema de assinatura pode ser implementado usando `debsig-verify` e `debsigs`. Estes dois pacotes podem assinar e verificar assinaturas embutidas em pacotes .deb. Debian já tem a capacidade de fazer sito, mas a implementação de policiamento e ferramentas não será iniciada até as releases posteriores ao woody.

As últimas versões do dpkg (a partir de 1.9.21) incorporam um patch (http://lists.debian.org/debian-dpkg/2001/debian-dpkg-200103/msg00024.html) que fornece esta funcionalidade tão logo `debsig-verify` seja instalado.

NOTA: Atualmente `/etc/dpkg/dpkg.cfg` trabalha com "no-debsig" como padrão.

NOTA 2: Signatures from developers are currently stripped when they enter off the package archive since the currently preferred method is release checks as described previously.

7.4.4 Pacotes experimentais apt

O release do apt 0.6 inclui *apt-secure* que é uma ferramenta que permitirá a um administrador de sistema testar a integridade dos pacotes baixados através do esquema acima. Esta release inclui a ferramenta `apt-key` para adicionar novas chaves ao chaveiro do apt, o qual por padrão inclui apenas o arquivo de assinatura de chaves atual do Debian.

Se quer testar este recurso você precisa adicionar a distribuição experimental ao seu `sources.list` e rodar

```
# apt-get -t experimental install apt
```

Estas alterações são baseadas no patch para `apt` (disponível em Bug #203741 (http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=203741)) o qual fornece esta implementação.

Este recurso ainda está em desenvolvimento, se você acredita que pode encontrar bugs nele por favor tenha certeza que está usando a última versão e, se estiver rodando a última versão, envie o bug para o pacote `apt` package usando a tag *experimental*.

Observe que, usar esta versão experimental do apt não exige nada mais de sua parte a menos que você não use sources Debian, neste caso um passo extra de confirmação será requerido pelo apt-get. Isto é evitado fornecendo Release e Release.gpg em non-Debian sources. O arquivo Release pode ser gerado com `apt-ftparchive` (disponível em apt-utils 0.5.0 e posteriores), o Release.gpg é apenas uma assinatura destacada. Para gerar ambos siga este simples procedimento:

```
$ rm -f dists/unstable/Release
$ apt-ftparchive release dists/unstable > dists/unstable/Release
$ gpg --sign -ba -o dists/unstable/Release.gpg dists/unstable/Release
```

Capítulo 8

Ferramentas de segurança no Debian

FIXME: Necessário mais conteúdo.

Debian fornece também uma série de ferramentas de segurança que podem tornar uma máquina com o sistema Debian adaptada para os propósitos de segurança. Estes propósitos incluem proteção dos sistemas de informação através de firewalls (de pacotes ou de aplicação), detecção de intrusão (baseados em rede e host), verificação de vulnerabilidades, antivirus, redes privadas, etc.

Desde o Debian 3.0 (*woody*), a distribuição caracteriza-se pelo software de criptografia integrado com a distribuição principal. OpenSSH e GNU Privacy Guard estão incluídos na instalação padrão, e criptografia forte está agora presente em navegadores e servidores Web, bancos de dados, e assim por diante. Além disso, a integração de criptografia está planejada para futuros lançamentos. Este software, devido as restrições de exportação nos EUA não foi distribuído com a distribuição principal, sendo disponível apenas em sites non-US.

8.1 Ferramentas de verificação remota de vulnerabilidades

As ferramentas fornecidas pelo Debian para realizar verificação remota de vulnerabilidade são: [1]

- nessus
- raccess
- whisker
- nikto (substituto de whisker)
- bass (non-free)
- satan (non-free)

[1] Algumas delas são fornecidas na instalação do pacote `harden-remoteaudit`.

A ferramenta mais completa e atualizada é, de longe, o `nessus` que é composta por um cliente (`nessus`) usado com uma GUI e um servidor (`nessusd`) que inicia os ataques programados. Nessus inclui verificação de vulnerabilidades remotas para a grande maioria de sistemas incluindo dispositivos de rede, servidores ftp e www, etc. Os últimos plugins de segurança tem a capacidade de analisar um sítio Web e tentar descobrir as páginas interativas disponíveis que podem ser atacadas. Existem também clientes Java e Win32 (não incluidas no Debian) que podem ser usados para acessar o servidor de gerenciamento.

Note que se você está usando woody, os pacotes do Nessus estão realmente desatualizados (veja bug #183524 (`http://bugs.debian.org/183524`)). Não é difícil portar os pacotes disponíveis no unstable para woody, mas se você encontrar dificuldades, pode pensar em usar os pacotes portados fornecidos por um dos co-mantenedores e disponíveis em `http://people.debian.org/~jfs/nessus/` (essas versões podem não estar atualizadas como as versões disponíveis no *unstable*).

`Whisker` é um varredor de verificação de vulnerabilidades Web, que inclui táticas anti-IDS (a maioria não são mais anti-IDS). É um dos melhores varreadores baseados em CGI disponíveis, sendo capaz de detectar servidores WWW e iniciar um dado conjunto de ataques contra ele. O banco de dados usado para a varredura pode ser facilmente modificado para fornecer novas informações.

`Bass` (Bulk Auditing Security Scanner - BULK Varreador de auditoria de segurança) e `SATAN` (Security Auditing Tool for Analyzing Networks - Ferramenta de auditoria de segurança para análise de redes) devem ser na opinião da maioria das pessoas, mais como programas de "provas de conceitos" do que como ferramentas para serem usadas em auditorias. Ambas são bastantes antigas e não são mais atualizadas. Contudo, SATAN foi a primeira ferramenta para fornecer avaliação das vulnerabilidades de maneira simples (através de uma GUI) e Bass é ainda uma ferramenta de avaliação de alta performace.

8.2 Ferramentas de varredura de rede

Debian fornece algumas ferramentas usadas para a varredura remota de hosts (mas não para verificação de vulnerabilidades). Estas ferramentas são, em alguns casos, usadas pelos verificadores de vulnerabilidades como o primeiro tipo de "ataque" executado contra os hosts remotos na tentativa de determinar os serviços disponíveis. Atualmente Debian fornece os seguintes programas:

- nmap
- xprobe
- queso
- knocker
- isic
- icmpush

- nbtscan (para auditorias NetBIOS)
- fragrouter
- strobe (do pacote `netdiag`)
- hping2 (*Nota:* desatualizado)

Enquanto o `queso` e o `xprobe` fornecem apenas detecção remota de sistema operacional (usando TCP/IP fingerprinting), `nmap` e `knocker` fazem, ambos, detecção de sistema operacional e varreadura de portas nos hosts remotos. Por outro lado, `hping2` e `icmpush` podem ser usados nas técnicas de ataque ICMP remoto.

Desenvolvido especificamente para redes Netbios, `nbtscan` pode ser usado para varrer redes IP e recuperar informações de nome de servidores samba habilitados, incluindo nomes de usuários e de rede, endereços MAC... Por outro lado, `fragrouter` pode ser usado para testar sistemas de detecção de instrusão e ver se o NIDS pode ser iludido com ataques de fragmentação.

FIXME: Verificar Bug #153117 (`http://bugs.debian.org/153117`) (ITP fragrouter) para ver se está incluído.

FIXME adicionar informações baseadas em Debian Linux Laptop for Road Warriors (`http://www.giac.org/practical/gcux/Stephanie_Thomas_GCUX.pdf`) que descreve como usar Debian e um laptop para varrer redes wireless (Link não existe mais).

8.3 Auditoria Interna

Atualmente, somente a ferramenta `tiger` utilizada no Debian pode ser usada para executar auditorias internas de hosts (também chamadas de "caixa branca") de fato para determinar se o sistema de arquivos está corretamente configurado, que processos estão rodando no hosts, etc..

8.4 Auditoria de código fonte

Debian fornce três pacotes que podem ser utilizados para auditar códigos fontes em C/C++ e encontrar erros de programação que podem conduzir para potenciais falhas de segurança:

- flawfinder
- rats
- splint

8.5 Redes Privadas Virtuais (VPN)

Uma rede privada virtual (VPN - Virtual Private Network) é um grupo de dois ou mais sistemas computacionais, tipicamente conectados a uma rede privada com acesso público de rede limitado, que se comunicam seguramente através de uma rede pública. VPNs podem conectar um simples computador a uma rede privada (cliente-servidor), ou uma LAN remota a uma rede privada (servidor-servidor). VPNs, muitas vezes, incluem o uso de criptografia, autenticação forte de usuários ou hosts remotos, e métodos para esconder a topologia da rede privada.

Debian fornece a maioria dos pacotes para configurar uma rede privada virtual criptografada:

- `vtun`
- `tunnelv`
- `cipe`
- `vpnd`
- `tinc`
- `secvpn`
- `pptpd`
- `freeswan`, que está obsoleto e substituido por
- `openswan` (`http://www.openswan.org/`)

FIXME: Atualizar as informações aqui já que foram escritas com o FreeSWAN em mente. Verificar Bug #237764 e a mensagem: <200412101215.04040.rmayr@debian.org>.

O pacote OpenSWAN é provavelmente a melhor escolha, desde que ele promete interoperar com quase tudo que usa o protocolo IP seguro, IPSec (RFC 2411). Entretanto, os outros pacotes listados acima podem também ajudá-lo a ter um túnel seguro rapidamente. O protocolo de tunelamento ponto a ponto (PPTP) é um protocolo para VPN proprietário da Microsoft. É suportado no Linux, mas é conhecido por ter sérios problemas de segurança.

Para mais informações veja VPN-Masquerade HOWTO (`http://www.tldp.org/HOWTO/VPN-Masquerade-HOWTO.html`) (cobrindo IPSec e PPTP), VPN HOWTO (`http://www.tldp.org/HOWTO/VPN-HOWTO.html`) (cobrindo PPP sobre SSH), e Cipe mini-HOWTO (`http://www.tldp.org/HOWTO/mini/Cipe+Masq.html`), e PPP and SSH mini-HOWTO (`http://www.tldp.org/HOWTO/mini/ppp-ssh/index.html`).

Também vale a pena verificar o Yavipin (`http://yavipin.sourceforge.net/`), mas este programa ainda não possue um pacote Debian disponível.

8.5.1 Tunelamento ponto a ponto

Se você deseja fornecer um servidor de tunelamento para um ambiente misto (com clientes Microsoft e Linux) e IPSec não é uma opção (desde que só é fornecido no Windows 2000 e Windows XP), você pode usar *PoPToP* (Servidor de Tunelamento Ponto a Ponto) disponível no pacote `pptpd`.

Se você deseja usar autenticação e criptografia da Microsoft com o servidor fornecido pelo pacote `ppp`, veja o seguinte trecho do FAQ:

```
O uso do PPP 2.3.8 só faz-se necessário se você deseja ter autenticação e
criptografia MSCHAPv2/MPPE compatíveis com a Microsoft. A razão para isto é q
o patch MSCHAPv2/MPPE atualmente aplicado (19990813) está sobre o PPP 2.3.8.
você não precisa de autenticação/criptografia compatível com a Microsoft,
qualquer versão 2.3.X do fonte do PPP será suficiente.
```

Entretanto, você também terá que aplicar o patch para o kernel fornecido no pacote `kernel-patch-mppe`, que contém o módulo pp_mppe para o pppd.

Saiba que a criptografia no ppptd força o armazenamento de senhas de usuários em texto limpo, e o protocolo MS-CHAPv2 contém furos de segurança conhecidos (http://mopo.informatik.uni-freiburg.de/pptp_mschapv2/).

8.6 Infra-estrutura de Chave Pública (PKI)

Infra-estrutura de Chave Pública (PKI - Public Key Infrastructure) é uma arquitetura de segurança introduzida para fornecer um nível adicional de confiança para trocas de informação em redes inseguras. Utiliza os conceitos de chaves de criptografia pública e privada para verificar a identidade de um remetente (assinatura) e para assegurar a privacidade (criptografia).

Quando considerar uma PKI, você encontrará uma variedade de situações:

- uma Autoridade Certificadora (CA - Certificate Authority) que pode distribuir e verificar certificados, e que pode trabalhar sobre uma dada hierarquia.
- um Diretório para manter certificados públicos de usuário
- um Banco de Dados (?) para manter Listas de Revogação de Certificados (CRL - Certificate Revocation Lists)
- dispositivos que interagem com a CA a fim de imprimir em smart cards/ tokens USB ou qualquer outra forma para armazenar seguramente os certificados.
- aplicações aptas a utilizarem certificados que podem usar certificados fornecidos por uma CA para realizar uma comunicação criptografada e verificar certificados dados contra CRL (para soluções de autenticação e assinatura de uma única vez completa)

- uma autoridade de marcação de tempo para assinar documentos digitalmente
- um console de gerenciamento a partir do qual tudo isso pode ser corretamente usado (geração de certificados, controle de lista de revogações, etc...)

Debian GNU/Linux tem pacotes de software para ajudar você com alguns desses pontos da PKI. Eles incluem `OpenSSL` (para geração de certificados), `OpenLDAP` (como um diretório para manter os certificados), `gnupg` e `openswan` (com suporte para o padrão X.509). Entretanto, como na versão Woody (Debian 3.0), Debian não tem nenhuma das autoridades certificadoras disponíveis gratuitamente como pyCA, OpenCA (`http://www.openca.org`) ou os exemplos de CA do OpenSSL. Para mais informações, leia o livro Open PKI (`http://ospkibook.sourceforge.net/`).

8.7 Infra-estrutura SSL

Debian fornece alguns certificados SSL com a distribuição de modo que eles podem ser instalados localmente. Eles são encontrados no pacote `ca-certificates`, que fornece um repositório central dos certificados que foram submetidos para o Debian e aprovados (ou seja, verificados) pelo mantenedor do pacote e utéis para qualquer aplicação OpenSSL que verifica conexões SSL.

FIXME: leia o debian-devel para verificar se algo foi adicionado a ele.

8.8 Ferramentas Anti-vírus

Não existem muitas ferramentas anti-vírus incluídas no Debian GNU/Linux, provavelmente porque os usuários GNU/Linux não são aborrecidos com vírus. O modelo de segurança dos UN*X fazem uma distinção entre os processos privilegiados (root) e os processos de usuário, então quando um executável "hostil" é criado ou recebido por um usuário não-root e então executado, não pode "infectar" ou manipular o sistema em questão. Entretanto, worms e vírus no GNU/Linux existem, embora eles não tenham (ainda, esperançosamente) se espalhado em nenhuma distribuição Debian. Em qualquer caso, administradores podem querer construir gateways anti-vírus que os protejam contra vírus enviados para outros sistemas mais vulneráveis em suas redes.

Debian GNU/Linux atualmente fornece as seguintes ferramentas para a construção de ambientes anti-vírus:

- Clam Antivirus (`http://clamav.elektrapro.com/`), fornecido no Debian *sarge* (futura versão 3.1). Pacotes são fornecidos tanto para o varredor de vírus (`clamav`), quanto para o daemon varredor (`clamav-daemon`) e para os arquivos de dados necessários para o varredor. Como a atualização do anti-vírus é crítica para o seu funcionamento, há duas formas diferentes de fazê-la: `clamav-freshclam` fornece um modo para atualização

do banco de dados automaticamente através da Internet e `clamav-data` que fornece os arquivos de dados diretamente. [2]

- `mailscanner` um gateway de email com varredor de vírus e detector de spam. Usando o `sendmail` ou `Exim` como sua base, ele pode usar mais de 17 diferentes mecanismos de varredura de vírus (incluindo `clamav`)
- `libfile-scan-perl` que fornece File::Scan, uma extensão Perl para a varredura de arquivos em busca de vírus. Este módulo pode ser usado para fazer varredores de vírus independentes de plataforma.
- Amavis Nova Geração (`http://www.sourceforge.net/projects/amavis`), fornecido no pacote `amavis-ng` e disponível no *sarge*, é um varredor de vírus em emails que é integrado com diferentes MTAs (Exim, Sendmail, Postfix, ou Qmail) e suporta cerca de quinze mecanismos de varredura de vírus (incluindo clamav, File::Scan e openantivirus).
- sanitizer (`http://packages.debian.org/sanitizer`), uma ferramenta que usa o pacote `procmail` que pode varrer anexos de email em busca de vírus, bloquear anexos baseados em seus nomes de arquivos e outras opções.
- amavis-postfix (`http://packages.debian.org/amavis-postfix`), um script que fornece uma interface de um agente de transporte de email para um ou mais varredores de vírus comerciais (este pacote é construído para suportar apenas o MTA `postfix`).
- `exiscan`, um varreador de e-mails escrito em Perl que funciona com o Exim.
- `sanitizer`, um varreador de emails que pode remover anexos potencialmente perigosos.
- `blackhole-qmail` um filtro de spam para o Qmail que foi construído com suporte para o Clamav.

Alguns daemons de gateways já suportam extensões de ferramentas para construir ambientes anti-virus, incluindo `exim4-daemon-heavy` (a versão *pesada* do MTA Exim), `frox` (um servidor proxy e cache transparente para ftp), `messagewall` (um daemon proxy SMTP) e `pop3vscan` (um proxy transparente POP3).

Como você pode ver, Debian não fornece atualmente nenhum software anti-vírus em sua distribuição oficial principal (3.0 no momento da escrita desse documento), mas fornece múltiplas interfaces para a construção de gateways anti-vírus. O varredor `Clamav` estará disponível na próxima versão oficial.

[2] Se você usar este último pacote e estiver usando um Debian oficial, o banco de dados não será atualizado com as atualizações de segurança. Você poderá usar o `clamav-freshclam` e o `clamav-getfiles` para gerar novos pacotes `clamav-data` ou atualizar do repositório do mantenedor, através da localização:

```
deb http://people.debian.org/~zugschlus/clamav-data/ / deb-src http://people.debian.org/~zugschlus
/
```

Alguns outros projetos anti-vírus livres que podem ser incluídos numa futura versão Debian GNU/Linux:

- Open Antivirus (http://sourceforge.net/projects/openantivirus/) (veja Bug #150698 (ITP oav-scannerdaemon (http://bugs.debian.org/150698) e Bug #150695 (ITP oav-update (http://bugs.debian.org/150695)).

Existe também um pacote `virussignatures`, que fornece assinaturas para todos os pacotes. Este pacote contém um script para fazer o download das últimas assinaturas de virus de http://www.openantivirus.org/latest.php.

FIXME: Verificar se scannerdaemon é o mesmo que o daemon varredor open anti-virus (ver ITPs).

Por outro lado, Debian *nunca* irá fornecer softwares anti-vírus comerciais como: Panda Antivirus, NAI Netshield, Sophos Sweep (http://www.sophos.com/), TrendMicro Interscan (http://www.antivirus.com), ou RAV (http://www.ravantivirus.com). Para mais apontadores veja em Linux antivirus software mini-FAQ (http://www.computer-networking.de/~link/security/av-linux_e.txt). Isto não signifca que estes softwares possam ser instalados corretamente em um sistema Debian.

Para mais informações de como configurar um sistema de detecção de vírus, veja o artigo de Dave Jones Building an E-mail Virus Detection System for Your Network (http://www.linuxjournal.com/article.php?sid=4882).

8.9 Agentes GPG

É muito comum, atualmente, assinar digitalmente (e algumas vezes criptografar) e-mails. Você pode, por exemplo, verificar que muitas pessoas participando em listas de discussão assinam seus e-mails. Assinaturas de chave pública são atualmente o único mecanismo para verificar que um email foi enviado pelo remetente e não por qualquer outra pessoa.

Debian GNU/Linux fornece clientes de emails com funções embutidas para assinatura de emails que interagem com o `gnupg` ou `pgp`:

- `Evolution`.
- `mutt`.
- `kmail`.
- `sylpheed`. Dependendo de como a versão estável deste pacote evolua, você pode precisar usar a *versão bleeding edge*, `sylpheed-claws`.
- `gnus`, que quando instalado com o pacote `mailcrypt` , é uma interface `emacs` interface para o `gnupg`.

- `kuvert`, que fornece esta funcionalidade independentemente do agente de email do usuário (MUA - Mail User Agente) escolhido já que interage com o agente de transporte de email (MTA - Mail Transport Agente).

Servidores de chave permitem você fazer o download de chaves públicas publicadas que podem então verificar assinaturas. Um desses servidores de chaves é `http://wwwkeys.pgp.net`. `gnupg` pode automaticamente buscar chaves públicas que não estão em seu chaveiro público. Por exemplo, para configurar `gnupg` para usar o servidor de chaves acima, edite o arquivo `~/.gnupg/options` e adicione a seguinte linha: [3]

```
keyserver wwwkeys.pgp.net
```

A maioria dos servidores de chaves estão ligados, logo quando uma chave pública é adicionada em um servidor, esta é propagada para todos os outros servidores de chaves públicas. Existem também um pacote Debian `debian-keyring`, que fornece todas as chaves públicas dos desenvolvedores Debian. Os chaveiros do `gnupg` são instalados em `/usr/share/keyrings/`.

Para mais informações:

- GnuPG FAQ (`http://www.gnupg.org/faq.html`).
- GnuPG Handbook (`http://www.gnupg.org/gph/en/manual.html`).
- GnuPG Mini Howto (English) (`http://www.dewinter.com/gnupg_howto/english/GPGMiniHowto.html`).
- comp.security.pgp FAQ (`http://www.uk.pgp.net/pgpnet/pgp-faq/`).
- Keysigning Party HOWTO (`http://www.cryptnet.net/fdp/crypto/gpg-party.html`).

[3] Para mais exemplos de como configurar o `gnupg`, veja `/usr/share/doc/mutt/examples/gpg.rc`.

Capítulo 9

Antes do comprometimento do sistema

9.1 Atualizando continuamente o sistema

Você deve fazer as atualizações de segurança frequentemente. A grande maioria de exploits existentes é resultado de vulnerabilidades conhecidas que não foram corrigidas a tempo, como este paper by Bill Arbaugh (http://www.cs.umd.edu/~waa/vulnerability.html) (apresentando no IEEE Symposium on Security and Privacy em 2001) explica. Atualizações estão descritas em 'Executar uma atualização de segurança' on page 40.

9.1.1 Verificando manualmente quais atualizações de segurança estão disponíveis

O Debian oferece uma ferramenta específica para verificar se o sistema precisa de atualização (veja o programa Tiger abaixo), mas muitos usuários preferem verificar manualmente se as atualizações de segurança estão disponíveis.

Se você configurou o seu sistema como descrito em 'Executar uma atualização de segurança' on page 40 você só precisa fazer:

```
# apt-get update
# apt-get upgrade -s
```

O primeiro comando baixa a lista de pacotes disponíveis nos sources de pacotes configurados. A opção `-s` faz somente uma simulação, isto é, *não* baixa ou instala os pacotes e sim diz quais devem ser baixados/instalados. Você poderá saber que pacotes foram consertados pelo Debian e estão disponíveis para atualização. Por exemplo:

```
# apt-get upgrade -s
Reading Package Lists... Done
Building Dependency Tree... Done
2 packages upgraded, 0 newly installed, 0 to remove and 0  not upgraded.
```

```
Inst cvs (1.11.1p1debian-8.1 Debian-Security:3.0/stable)
Inst libcupsys2 (1.1.14-4.4 Debian-Security:3.0/stable)
Conf cvs (1.11.1p1debian-8.1 Debian-Security:3.0/stable)
Conf libcupsys2 (1.1.14-4.4 Debian-Security:3.0/stable)
```

Neste exemplo, você pode observar que precisa atualizar os pacotes cvs e cupsys, os quais estão sendo retornados do arquivo de atualização de segurança do *woody*. Se quiser entender porque estes pacotes são necessários, vá em `http://security.debian.org` e verifique quais Alertas de Segurança do Debian foram publicados e estão relacionados com esses pacotes. Neste caso, os alertas relacionados são DSA-233 (`http://www.debian.org/security/2003/dsa-233`) (para cvs) e DSA-232 (`http://www.debian.org/security/2003/dsa-232`) (para cupsys).

9.1.2 Verificando automaticamente por atualizações com o cron-apt

Um outro método para atualização de segurança automática é uso do `cron-apt`. Este pacote fornece uma ferramente para atualizar o sistema em intervalos regulares (usando um job do cron). Ele faz a atualização da lista de pacotes e baixa os pacotes novos por padrão. Ele também pode ser configurado para enviar mails para o administrador do sistema.

Note que você pode querer verificar a versão da distribuição, como descrito em 'Checando releases das distribuições' on page 121, se você pretende atualizar automaticamente o seu sistema (mesmo somente baixando pacotes). Caso contrário você não terá certeza que os pacotes baixados realmente são de origem confiável.

9.1.3 Usando o Tiger para verificar automaticamente atualizações de segurança

Se você está procurando por uma ferramenta que rapidamente verifique e relate vulnerabilidades de segurança do sistema, tente o pacote `tiger`. Este pacote fornece um conjunto de scripts shell, programas em C e arquivos de dados usados para realizar auditorias de segurança. O pacote do Debian GNU/Linux tem melhorias adicionais voltadas para a distribuição Debian, provendo mais funcionalidade do que os scripts Tiger fornecidos por TAMU (ou até TARA, uma versão do tiger distribuida por ARSC). Veja o arquivo `README.Debian` e a página de manual `tiger(8)` para mais informações.

Uma dessas melhorias é o script `deb_checkadvisories`. Este script recebe uma lista de DSA's (Alertas de Segurança do Debian) e verifica com a base de pacote instalada, informando quaisquer pacotes que estão vulneráveis conforme o Time de Segurança do Debian. Ele é um pouco mais genérico do que o script `check_signatures` implementado pelo Tiger, pois este é capaz de verificar MD5sums de programas vulneráveis conhecidos.

Já que o Debian atualmente não distribui uma lista de MD5sums de programas vulneráveis conhecidos (utilizado por algum outro sistema operacional como Sun Solaris), a solução *check-against-DSA* é usada. Ambas as soluções DSA e MD5sums sofrem do problema de que as assinaturas devem ser atualizadas regularmente.

Atualmente esse problema é resolvido fazendo novas versões do pacote Tiger, mas o mantenedor do pacote nem sempre pode fazer uma nova versão toda vez que um DSA é anunciado. Uma melhoria interessante, que ainda não está implementada, poderia fazer este trabalho próativamente. Isto é, fazer o download dos DSAs da web, construir a lista de DSAs e então rodar a verificação. Os DSAs são atualmente atualizados pelo mantenador do CVS local das fontes WML utilizadas para desenvolver `http://security.debian.org` (o servidor web).

Um programa para analisar sintaticamente os DSAs publicados, receber através de e-mail ou disponibilizar no security.debian.org, e então gerar o arquivo usado pelo `deb_checkadvisories` para confirmar vulnerabilidades seria bem-vindo. Envie-o como um relatório de bug para o pacote `tiger`.

Uma vez instalado, a verificação mencionada é definida pela configuração padrão do programa (veja `/etc/tiger/cronrc`):

```
# Check for Debian security measures every day at 1 AM
#
1 * *   deb_checkmd5sums deb_nopackfiles deb_checkadvisories
#
```

Existe uma verificação adicional que você pode querer acrescentar apesar de ainda não fazer parte dos scripts padrões do `cron`. O script `check_patches` funciona da seguinte maneira:

- execute `apt-get update`
- verifique se há novos pacotes disponíveis

Se você estiver rodando o Debian estável e adicionar a linha de fonte `apt` security.debian.org em `/etc/apt/sources.list` (como descrito em 'Executar uma atualização de segurança' on page 40), este script será capaz de informar se existem pacotes novos que devem ser instalados. Desde que somente os pacotes com configurações modificadas são atualizações de segurança, então você tem apenas tudo o que queria.

Claro que isso não funcionará se você estiver rodando a versão *testing* ou *sid/unstable*, já que atualmente, os novos pacote provavelmente têm mais funcionalidades que as atualização de segurança.

Você pode adicionar este script para realizar as verificações em um `cron` job (no arquivo de configuração) e no `tigercron` poderá enviar um email (para o endereço especificado na diretiva `Tiger_Mail_RCPT` em `/etc/tiger/tigerrc`) com os novos pacotes:

```
# Check for Debian security measures every day at 1 am
#
1 * *   deb_checkmd5sums deb_nopackfiles check_patches
#
```

9.1.4 Outros métodos para atualizações de segurança

Você também pode dar uma olhada em secpack (http://therapy.endorphin.org/secpack/) que é um programa não-oficial escrito por Fruhwirth Clemens e usado para fazer atualizações de segurança a partir do site security.debian.org com suporte a verificação de assinaturas.

9.1.5 Evite usar versões instáveis

Ao menos que você tenha tempo para aplicar patches de segurança toda vez que uma vulnerabilidade é descoberta, você *não* deve usar a versão instável do Debian para sistemas em produção. A principal razão para isto é que não há atualizações de segurança para a versão *unstable* (veja 'Como a segurança é tratada na `testing` e `unstable`?' on page 178).

O fato é que algumas questões relacionadas à segurança podem surgir na distribuição instável e *não* na *stable*. Isto porque novas funcionalidades são constantemente adicionadas às aplicações, assim como novas aplicações são incluídas sem serem totalmente testadas.

Para se fazer atualizações de segurança na versão *unstable*, você pode fazer uma atualização completa para nova versão (que atualiza muito mais do que somente os pacotes afetados). Embora existam algumas exceções, patches de segurança geralmente só são portadas para a versão *stable*. A idéia principal é que entre as atualizações, *nenhum código novo* deve ser adicionado, somente consertos para questões importantes.

9.1.6 Evite usar versões em teste

Se você estiver utilizando uma versão em *testing*, existem algumas questões relacionadas à disponibilidade das atualizações de segurança que devem ser levadas em conta:

- Quando um conserto de segurança é preparado, o Time de Segurança lança o patch para a versão *stable* (desde que a estável é geralmente algumas versões menor ou maior atrás). Os mantenedores de pacotes são responsáveis por preparar o patch para a versão *unstable*, geralmente baseado nos novos lançamentos. Algumas vezes as alterações acontecem quase ao mesmo tempo e em outras um dos lançamentos disponibiliza o conserto de segurança antes. Os pacotes para a distribuição *stable* são testados bem mais a fundo do que para a *unstable*, já que esta irá fornecer na maioria dos casos a última versão do lançamento (que pode incluir novos e desconhecidos bugs)

- Atualizações de segurança estão disponíveis para a versão *unstable* geralmente quando os mantenedores fazem um novo pacote e para a versão *stable* quando o Time de Segurança publica um DSA e faz um novo upload. Observe que nada disso altera a versão em *testing*.

- Se nenhum (novo) bug é detectado na versão *unstable* do pacote, ele passa para a versão em *testing* depois de algum tempo. Este tempo geralmente é de dez dias, embora

dependa de algumas coisas como a prioridade de upload e se o pacote está ou não bloqueado para entrar em teste por causa de dependências. Note que se o pacote estiver bloqueado, a prioridade de upload não afetará o tempo que ele leva para entrar na versão em teste.

Esse comportamento pode ser alterado conforme o estado de lançamento da distribuição. Quando uma distribuição está perto de ser lançada, o Time de Segurança ou os mantenedores dos pacotes devem fornecer atualizações de segurança diretamente para a versão em teste.

9.1.7 Atualizações automáticas no sistema Debian GNU/Linux

Primeiro de tudo, atualizações automáticas não são recomendadas, já que o administrador deve revisar os DSAs (alertas de segurança do Debian) e entender o impacto causado pela atualização de segurança no sistema.

Para atualizar o seu sistema automaticamente você deve:

- Configurar o `apt` para que os pacotes que você não queria atualizar continuem na mesma versão, usando o recurso de *pinning* do `apt` ou marcando-os como *hold* no `dpkg` ou `dselect`.

  Para fixar os pacotes em uma determinada versão, você deve editar o arquivo `/etc/apt/preferences` (veja `apt_preferences(5)`) e adicionar:

  ```
  Package: *
  Pin: release a=stable
  Pin-Priority: 100
  ```

  FIXME: verifcar se a configuração está OK.

- Você também pode usar o cron-apt como descrito em 'Verificando automaticamente por atualizações com o cron-apt' on page 142. Ative-o para instalar os pacotes baixados ou adicione uma entrada no `cron` para que a atualização seja feita diariamente, por exemplo:

  ```
  apt-get update && apt-get -y upgrade
  ```

  A opção `-y` faz com que o `apt` assuma 'sim' para todos os prompts que aparecerão durante a atualização. Em alguns casos, é melhor você usar a opção `--trivial-only` em vez de `--assume-yes` (equivalente a `-y`). [1]

- Configure o `cron` para que o `debconf` não faça nenhuma pergunta durante as atualizações, funcionando de forma não-interativa. [2]

[1] Você também pode optar por usar a opção `--quiet` (`-q`) para diminuir a quantidade de informações de saída do `apt-get`. Caso nenhum pacote esteja sendo instalado, nenhuma informação é mostrada na tela.

[2] Note que alguns pacotes podem *não* usar o `debconf` e a atualizações irão parar para que o usuário entre com alguma configuração.

- Verifique os resultados da execução do `cron`, que enviará um mail para o superusuário (ao menos que a variável de ambiente `MAILTO` seja alterada no script).

Uma alternativa mais segura seria usar a opção `-d` (ou `--download-only`), que irá fazer o download dos pacotes necessários mas não os instalará. Então se a execução do `cron` mostrar que o sistema precisa ser atualizado, esta atualização pode ser feita manualmente.

E para finalizar estas tarefas, o sistema deve ser configurado apropriadamente para fazer o download das atualizações de segurança como discutido no 'Executar uma atualização de segurança' on page 40.

Entretanto, isto não é recomendado para a versão *unstable* sem que haja uma análise cuidadosa, uma vez que pode tornar o seu sistema inutilizável se algum pacote importante que estiver com um bug sério for instalado. A *testing* é um pouco mais *segura* com relação a isto, já que os bugs sérios podem ser detectados antes do pacote ser movido para a versão em teste (embora, você *não* tenha atualizações de segurança disponíveis para todos).

Se você tem uma distribuição mista, isto é, uma instalação *stable* com alguns pacotes atualizados para a versão em *testing* ou *unstable*, você pode utilizar o recurso de *pinning* assim como a opção `--target-release` do `apt` para atualizar *somente* aqueles pacotes que devem ser atualizados. [3]

9.2 Faça verificações de integridade periódicas

A verificação de integridade é feita baseada na informação completa do sistema gerada depois da instalação (ex. o *snapshot* descrito em 'Fazendo um snapshot do sistema' on page 77) e deve ser feita de tempos em tempos. Com a verificação de integridade é possível detectar modificações no sistema de arquivos feitas por um intruso ou por algum erro do administrador do sistema.

As verificações de integridade devem ser, se possível, feitas offline [4] . Isto é, utilizar outro sistema operacional para fazer a verificação, evitando assim um falso senso de segurança (ex. falsos negativos) produzido por, por exemplo, rootkits instalados. A base de dados de integridade verificada pelo sistema também deve ser usada em uma mídia somente leitura.

Você deve considerar fazer a verificação online utilizando qualquer ferramenta de verificação de integridade do sistema de arquivos disponíveis (descrito em 'Verificando a integridade do sistema de arquivos' on page 71), se você não puder deixar o sistema fora do ar. Entretanto, algumas precauções devem ser levadas em conta como a utilização de uma base de dados da integridade somente para leitura e assegurar que a ferramenta de verificação de integridade (e o kernel do sistema operacional) não esteja sendo usada.

[3] Isso é uma prática comum, já que muitos usuários preferem manter o sistema estável, podendo atualizar alguns pacotes para a versão *unstable* para obter novas funcionalidades. Esta necessidade surge devido ao desenvolvimento de alguns projetos ser mais rápido que o tempo gasto entre os lançamentos da versão *stable* do Debian.

[4] Uma maneira fácil de fazer isso é utilizar um Live CD, tipo o Knoppix Std (`http://www.knoppix-std.org/`) que inclue ambas as ferramentas de verificação de arquivos e a base de dados de integridade do seu sistema.

Algumas das ferramentas citadas nesta seção, como `aide`, `integrit` ou `samhain` já estão preparadas para fazer revisões periódicas (através do crontab nas duas primeiras e através de um daemon standalone na `samhain`) e pode avisar o administrador por diferentes canais (geralmente e-mail, mas `samhain` também pode enviar pages, traps SNMP ou alertas do syslog) quando ocorrem alterações no sistema de arquivos.

Claro que se você for executar uma atualização do sistema, deve ser tirado novamente um snapshot para acomodar as alterações sofridas durante a atualização de segurança.

9.3 Configure um sistema de Detecção de Intrusão

O Debian GNU/Linux inclue ferramentas para detecção de intrusão, que é nada mais do que a prática de detectar atividades impróprias ou maliciosas no seu sistema local, ou outros sistemas que estejam na sua rede privada. Este tipo de defesa é importante se o sistema for altamente crítico ou você for realmente paranóico. Os tipos mais comuns de detecção de intrusão são detecção estatística de anomalias e detecção baseada em algum padrão.

Sempre tenha em mente que para melhorar a segurança do sistema com a instalação de uma dessas ferramentas, você deve ter um mecanismo de alertas e respostas elaborado. Detecção de intrusão é perda de tempo se você não for alertar ninguém.

Quando um ataque em particular for detectado, a maioria das ferramentas de detecção de intrusão irá tanto gerar um log do evento com o `syslogd` enviar um e-mail para o super-usuário (o destinatário geralmente é configurável). Um administrador precisa configurar propriamente as ferramentas para que falsos positivos não gerem alertas. Alertas também devem informar um ataque que pode estar acontecendo e ele não será útil, digamos, um dia depois que ocorrer. Então tenha certeza que existe uma política apropriada para tratar os alertas e que os mecanimos técnicos para implementar essa política sejam viáveis.

Uma fonte interessante de informações é CERT's Intrusion Detection Checklist (`http://www.cert.org/tech_tips/intruder_detection_checklist.html`)

9.3.1 Detecção de intrusão baseada em rede

As ferramentas de detecção de intrusão baseada em rede monitoram o tráfego em um segmento de rede e utilizam essas informações como fonte dos dados para serem analisados. Especificamente, os pacotes da rede são examinados, e eles são verificados para ver se existe uma certa assinatura de pacotes maliciosos.

O `Snort` é um sniffer de pacotes bastante flexível ou logger que detecta os ataques utilizando um dicionário de assinatura de ataques. Ele detecta uma variedade de ataques e probes, como estouro de buffer, varredores de portas stealth, ataques CGI, probes SMB e muito mais. O `Snort` também tem a capacidade de gerar alertas em tempo real. Você pode usar o `snort` tanto para uma série de máquinas na sua rede quanto para o seu próprio servidor. Ele é uma ferramenta que deve ser instalada em todos os roteadores para manter os

olhos na rede. Para instalá-lo basta usar o `apt-get install snort`, seguir as perguntas, e verificar o log. Para um arcabouço de segurança um pouco mais amplo, veja Prelude (http://www.prelude-ids.org).

O pacote `snort` do Debian tem diversas configurações de segurança ativadas por padrão. Entretanto, você deve customizar o programa tendo em mente os serviços particulares que você roda no seu sistema. Também seria interessante procurar algumas verificações específicas para estes serviços.

Nota: Os pacotes do snort disponíveis no woody não estão tão atualizados e podem até estar bugados (http://lists.debian.org/debian-devel/2003/debian-devel-200308/msg02105.html), você pode obter um backport (e assinatura) do Snort fornecido pelo mantenedor do pacote em http://people.debian.org/~ssmeenk/snort-stable-i386/.

Existem outras ferramentas mais simples que podem ser utilizadas para detectar ataques em rede. O `portsentry` é um pacote interessante que pode ajudar a descobrir varreduras contra seus hosts. Outras ferramentas como `ippl` ou `iplogger` também podem detectar alguns ataques IP (TCP e ICMP), mesmo que eles não forneçam os tipos de técnicas que o `snort` fornece.

Você pode testar qualquer uma dessas ferramentas com o pacote do Debian `idswakeup`, um script em shell que gera alarmes falsos e inclue muitas assinaturas de ataques comuns.

9.3.2 Detecção de intrusão baseada em host

A detecção de intrusão baseada em host envolve o carregamento de um software no sistema a ser monitorado e que utiliza arquivos de log e/ou os programas de auditoria de sistema como uma fonte de dados. Ele procura por processos suspeitos, monitora acesso ao host e pode até monitorar alterações em arquivos críticos do sistema.

O `tiger` é uma antiga ferramenta de detecção de intrusão que foi portado para o Debian desde a versão do Woddy. Ele fornece verificações de casos comuns relacionados a furo de segurança, como uso de força bruta nas senhas, problemas no sistema de arquivo, comunicação de processos e outras formas de comprometer o superusuário. Este pacote também inclue verificações de segurança específicas para o Debian como: verificações de MD5sums de arquivos instalados, localização de arquivos que não pertencem a nenhum pacote e análise de processos locais que estão em estado de escuta. A instalação padrão configura o `tiger` para rodar diariamente, gerando um relatório que é enviado para o superusuário sobre possíveis comprometimentos no sistema.

Ferramentas de análise de log, como `logcheck` também podem ser usadas para detectar tentativas de intrusão. Veja 'Usando e personalizando o `logcheck`' on page 61.

Em adição, pacotes que monitoram a integridade do sistema de arquivo (veja 'Verificando a integridade do sistema de arquivos' on page 71) podem ser perfeitamente úteis na detecção de anomalias em um ambiente seguro. É muito provável que uma intrusão efetiva irá modificar alguns arquivos no sistema de arquivo local para driblar a política de segurança local,

instalar Trojans, ou criar usuários. Tais eventos podem ser detectados com os programas para verificação de integridade do arquivo.

9.4 Evitando os rootkits

9.4.1 Loadable Kernel Modules (LKM)

Loadable kernel modules são arquivos contendo componentes carregados dinamicamente no kernel e são usados para expandir a funcionalidade do mesmo. O benefício principal de se usar módulos é a habilidade de adicionar dispositivos adicionas, como uma placa de rede Ethernet ou uma placa de som, sem ter que aplicar um patch no código-fonte e recompilar todo o kernel. Entretanto, os crackers vêm usando os LKMs para criar rootkits (knark e adore), abrindo backdoors nos sistemas GNU/Linux.

Os backdoors LKM estão cada vez mais sofisticados e mais difíceis de serem detectados que os rootkits tradicionais. Eles podem esconder processos, arquivos, diretórios e até mesmo conexões sem precisar modificar o código fonte dos binários. Por exemplo, um LKM malicioso pode forçar o kernel a esconder processos específicos do `procfs`, então mesmo uma cópia original do binário `ps` pode não listar informações precisas sobre os processos que estão rodando no sistema.

9.4.2 Detectando rootkits

Existem duas estratégias para defender seu sistema de rootkits LKM, a defesa pró-ativa e a reativa. O trabalho de detecção pode ser simples e fácil, ou difícil e cansativo, dependendo da estratégia escolhida.

Defesa pró-ativa

A vantagem para este tipo defesa é que ela previne qualquer dano ao sistema logo de início. Uma estratégia para esse tipo de defesa é conhecida como *pegar eles primeiro*, que é carregar na memória um módulo LKM designado para proteger o sistema de outros LKMs maliciosos. A segunda estratégia é remover algumas funcionalidades do próprio kernel. Por exemplo, você pode desabilitar a opção de carrergar módulos no kernel. Entretanto, note que existem rootkits que podem funcionar até mesmo neste caso. Alguns deles podem mexer com o `/dev/kmem` (memória do kernel) diretamente para torná-los indetectáveis.

O Debian GNU/Linux tem poucos pacotes que podem ser usados para montar uma defesa pró-ativa:

- `kernel-patch-2.4-lsm` - LSM é o arcabouço para os Módulos de Segurança do Linux.
- `lcap` - Uma interface amigável para remover as *funcionalidades* (controle de acesso) do kernel, fazendo o sistema mais seguro. Por exemplo, executando `lcap`

CAP_SYS_MODULE [5] irá remove a funcionalidade de carregar módulos (mesmo para o super-usuário). [6] Para mais informações sobre as funcionalidades do kernel você deve verificar a seção Kernel development (http://lwn.net/1999/1202/kernel.php3) de Jon Corbet na LWN (Dezembro 1999)

Se você realmente não precisa de muitos recursos do kernel no seu sistema GNU/Linux, você pode desabilitar o suporte aos módulos carregáveis durante a configuração do kernel. Para desabilitar este suporte, somente altere o CONFIG_MODULES=n durante o estágio de configuração da construção do seu kernel, ou no arquivo `.config`. Isto irá prevenir os rootkits LKM, mas você irá perder esta funcionalidade poderosa no kernel do Linux. Desabilitar a opção para carregar módulos no kernel pode muitas vezes sobrecarregar o kernel. Neste caso, é melhor deixar o kernel com o suporte.

Defesa reativa

A vantagem da defesa reativa é que ela não consome os recursos do sistema. Ela trabalha comparando a tabela de chamadas ao sistema com uma cópia autêntica conhecida, o arquivo em disco `System.map`. Claro que a defesa reativa somente notificará ao administrador do sistema depois que o sistema já estiver sido comprometido.

A detecção de alguns root-kits no Debian pode ser efetuada com o pacote `chkrootkit`. O programa Chkrootkit (http://www.chkrootkit.org) verifica por sinais de diversos rootkits conhecidos no sistema alvo, mas isto não deve ser um teste final.

Uma outra ferramenta auxiliar é o KSTAT (http://www.s0ftpj.org/en/site.html) (Kernel Security Therapy Anti Trolls) feita pelo grupo S0ftproject. O KSTAT busca na área de memória do kernel (`/dev/kmem`) informações sobre o host alvo para ajudar o administrador do sistema a encontrar e remover LKMs maliciosos.

9.5 Idéias Geniais/Paranóicas — o que você pode fazer

Esta é provavelmente a mais instável e divertida seção, apenas espero que algumas das ideias "duh, isso parece loucura" possam ser realizadas. A seguir algumas idéias para melhorar a segurança — talvez geniais, paranóicas, loucas ou até inspiradas dependendo do seu ponto de vista.

[5] Existem mais de 28 funcionalidades incluídas: CAP_BSET, CAP_CHOWN, CAP_FOWNER, CAP_FSETID, CAP_FS_MASK, CAP_FULL_SET, CAP_INIT_EFF_SET, CAP_INIT_INH_SET, CAP_IPC_LOCK, CAP_IPC_OWNER, CAP_KILL, CAP_LEASE, CAP_LINUX_IMMUTABLE, CAP_MKNOD, CAP_NET_ADMIN, CAP_NET_BIND_SERVICE, CAP_NET_RAW, CAP_SETGID, CAP_SETPCAP, CAP_SETUID, CAP_SYS_ADMIN, CAP_SYS_BOOT, CAP_SYS_CHROOT, CAP_SYS_MODULE, CAP_SYS_NICE, CAP_SYS_PACCT, CAP_SYS_PTRACE, CAP_SYS_RAWIO, CAP_SYS_RESOURCE, CAP_SYS_TIME, and CAP_SYS_TTY_CONFIG. Todas elas podem ser desativadas para melhorar a segurança do seu kernel.

[6] Você não precisa instalar o `lcap` para fazer isto, mas é melhor do que configurar o `/proc/sys/kernel/cap-bound` na mão.

- Brincando com o PAM. Como citado no artigo Phrack 56 PAM, a coisa legal do PAM é que "Você é limitado somente pelo o que pode imaginar". É verdade. Imagine efetuar login de root somente através de impressão digital ou verificação de retina ou cartão de criptografia (por que usei a conjunção OU em vez de E?).

- Gravação fascista de logs. Eu prefiro me referir à toda discussão anterior acima como um "esquema leve de logs". Se você quiser fazer um esquema real de logs, pegue uma impressora com papel de formulário contínuo, e envie todos os logs para ela. Parece engraçado, mas é realmente confiável e as informações não podem ser sobrescritas ou apagadas.

- Distribuição de CD. Essa idéia é muito simples de se realizar e oferece uma boa segurança. Crie uma distribuição Debian segura, com as regras de firewall apropriadas. Coloque ela em uma imagem ISO inicializável e grave em um CDROM. Agora você tem uma distribuição somente leitura, com mais ou menos 600 MB de espaço para os serviços. Tenha certeza de que todos os dados que devem ser escritos sejam feitos pela rede. É impossível para um intruso ter acesso de leitura/escrita no sistema, e qualquer alteração feita pelo intruso pode ser desfeita em uma reinicilização do sistema.

- Desabilite a capacidade de carregar módulos. Como discutido anteriormente, quando você desabilita o uso de módulos em tempo de compilação do kernel, muitos backdoors baseados em kernel ficam impossíveis de serem implementados, pois a maioria deles é baseada na instalação de módulos do kernel modificados.

- Grave os logs por um cabo serial. (contribuido por Gaby Schilders) Já que os servidores ainda têm portas serial, imagine ter um sistema de gravação de logs para um série de servidores. O sistema de logs é desconectado da rede, e conectado aos servidores via um multiplexador de porta serial (Cyclades ou algo do tipo). Agora faça com que todos os seus servidores gravem o log através da porta serial. A máquina de log vai somente aceitar o texto plano como entrada nas portas serial e escrever em um arquivo de log. Conecte um gravador de CD/DVD e grave o arquivo de log quando atingir a capacidade máxima da mídia.

- Altere as atribuições do arquivo usando `chattr`. (dica tirada do Tips-HOWTO, escrito por Jim Dennis). Depois de uma instalação limpa e configuração inicial, use o programa `chattr` com o atributo `+i` para que os arquivos não sejam modificados (o arquivo não pode ser apagado, renomeado, criado link ou escrito algo nele). Defina este atributo em todos os arquivos que estão em `/bin`, `/sbin/`, `/usr/bin`, `/usr/sbin`, `/usr/lib` e também nos arquivos do kernel no root. Você também pode fazer uma cópia de todos os arquivos do `/etc/`, usando o `tar` ou algo do tipo e marcar o arquivo comprimido como imutável.

  Esta estratégia irá ajudar a limitar o estrago que você poderá causar estando logado como root. Você não poderá sobrescrever arquivos por engano, nem deixar o sistema inoperante digitando por engano um espaço no comando `rm -fr` (você pode ainda fazer um monte de estragos no seus dados — mas suas bibliotecas e seus binários estarão seguros.)

  Esta estratégia também faz com que uma variedade de exploits de segurança e de negação de serviços (DoS) sejam difíceis ou impossíveis de serem realizados (já que a maio-

ria deles conta com a permissão de sobrescrever um arquivo através de algum programa SETUID que a princípio *não esteja fornecendo um comando shell arbitrário*).

Uma inconveniência desse tipo de estratégia aparece durante a compilação e instalação de alguns binários do sistema. Por outro lado, isso previne que um comando `make install` sobrescreva os arquivos. Quando você se esquece de ler o Makefile e executa um `chattr -i` nos arquivos a serem sobrescritos, (também nos diretórios nos quais serão adicionados os arquivos) - o comando make falha. Então você deve usar o comando `chattr` para desativar a flag de imutável e rodar o make novamente. Você também pode optar por mover os binários e as bibliotecas antigas para dentro de um diretório .old/ ou para um arquivo tar por exemplo.

Note que esta estratégia também impede que você atualize seu próprio sistema de pacotes, já que os arquivos que os pacotes a serem atualizados fornecem não podem ser sobrescritos. Você pode fazer um script ou usar outro mecanismo parecido para desativar a permissão de imutável em todos os binários antes de fazer um `apt-get update`.

- Você pode brincar um pouco com o cabeamento UTP cortando 2 ou 4 fios, tornando um cabo de tráfego unidirecional. Então use pacotes UDP para enviar informação para uma máquina de destino que atuaria como um servidor de log seguro ou até mesmo um sistema de armazenamento de cartões de crédito.

9.5.1 Construindo um honeypot

FIXME: É preciso de conteúdo mais específico para o Debian

Um honeypot é um sistema feito para auxiliar os administradores de sistemas a descobrir como os crackers sondam a máquina em busca de exploits. O sistema é configurado com a expectativa e objetivo de ser sondado, atacado e potencialmente invadido. Aprendendo as ferramentas e os métodos empregados pelo cracker, um administrador de sistema pode saber como melhor proteger seus sistemas e a rede.

Um sistema Debian GNU/Linux pode ser facilmente configurado como um honeypot, se você dedicar tempo para implementar e monitorá-lo. Simplesmente configure o servidor falso com um firewall e algumas ferramentas de detecção de intrusão de rede, coloque ele na Internet, e espere. Tome o cuidado de que se o sistema for invadido você seja imediatamente alertado (veja 'A importância dos logs e alertas' on page 60), desta forma você poderá tomar providências necessárias e paralizar a invasão quando tiver informações suficientes. Abaixo estão alguns dos pacotes e questões importantes quando estiver configurando seu honeypot:

- A tecnologia de firewall que irá usar (fornecidade pelo kernel do Linux).
- `syslog-ng`, útil para enviar logs do honeypot para um servidor syslog remoto.
- `snort`, para configurar a captura de todo o tráfego de rede de entrada para o honeypot e detectar os ataques.
- `osh`, um SETUID root, segurança aprimorada, shell restrita com sistema de log (veja o artigo de Lance Spitzner abaixo).

- Claro que todos os daemons serão usados por seu servidor honeypot falso (então *não* assegurar o honeypot).
- The Deception Toolkit, que utiliza um sistema de indução ao erro para reagir aos ataques. Homepage: Deception Toolkit (`http://all.net/dtk/dtk.html`)
- Verificadores de integridade (veja 'Verificando a integridade do sistema de arquivos' on page 71) e o Toolkit do Coroner (`tct`) para fazer auditorias pós-ataque.

Você pode ler mais sobre como construir honeypots no excelente artigo de Lanze Spitzner To Build a Honeypot (`http://www.net-security.org/text/articles/spitzner/honeypot.shtml`) (das séries "Know your Enemy"), ou de David Raikow Building your own honeypot (`http://www.zdnetindia.com/techzone/resources/security/stories/7601.htm`). O Projeto Honeynet (`http://project.honeynet.org/`) também fornece informações valiosas relacionadas à construção de honeypots e auditoria dos ataques feitos nelas.

Capítulo 10

Depois do comprometimento do sistema (resposta a incidentes)

10.1 Comportamento comum

Se você estiver fisicamente presente quando o ataque ocorrer, sua primeira obrigação é tirar a máquina da rede desconectando o cabo de rede da placa (se isso não for influenciar nas transações dos negócios). Desativando a rede na camada 1 é a única forma de manter o invasor longe da máquina comprometida (conselho sábio de Philip Hofmesiter).

Entretanto, alguns rootkits ou back doors são capazes de detectar este tipo de evento e reagir a ele. Ver um `rm -rf /` sendo executado quando você desativa a rede não é muito engraçado. Se você se nega a correr o risco e tem certeza que o sistema foi comprometido, você deve *desconectar o cabo de energia* (todos eles se existirem mais de um) e cruzar os dedos. Isso pode ser extremo mas, de fato, irá evitar qualquer bomba lógica que o invasor possa ter programado. Nesses casos, o sistema comprometido *não deve ser reiniciado*. Os discos rígidos também devem ser colocados em outro sistema para serem analisados, ou deve ser usado outro tipo de mídia (um CD-ROM) para inicializar o sistema e analisá-lo. Você *não* deve usar os discos de recuperação do Debian para inicializar o sistema, mas você *pode* utilizar o shell fornecido pelos discos de instalação (use Alt+F2 para acessá-lo) para analisar o sistema. [1]

O método mais recomendado para restaurar um sistema comprometido é utilizar um CDROM com todas as ferramentas (e módulos do kernel) necessárias para acessar o sistema. Você pode utilizar o pacote `mkinitrd-cd` para compilar tal CDROM[2]. Você também pode achar o CDROM FIRE (`http://biatchux.dmzs.com/`) útil, já que é um live CDROM com ferramentas para análise forense ideal neste tipo de situação. Não existe (ainda) uma ferramenta baseada no Debian como esta, nem uma maneira fácil de compilar o CDROM com pacotes

[1] Se você for aventureiro, você pode efetuar o logon no sistema e salvar as informações de todos os processos em execução (várias dessas informações estão em /proc/nnn/). É possível pegar todo código executável da memória, mesmo se o invasor tiver excluído os arquivos executáveis do disco. Então puxe o cabo de força.

[2]. De fato, esta é a ferramenta usada para compilar os CDROMs para o projeto Gibraltar (`http://www.gibraltar.at/`) (um firewall em um live-CD baseado na distribuição Debian).

específicos e com `mkinitrd-cd` (então você terá que ler a documentação fornecida com o programa para fazer seus próprios CDROMs).

Se você realmente quer consertar um sistema comprometido rapidamente, você deve tirar o sistema da sua rede e reinstalar todo o sistema operacional do zero. Claro, isto pode não ser efetivo porque você não saberá como o invasor comprometeu o sistema. Neste caso, você deve verificar tudo: firewall, integridade de arquivos, host de log, arquivos de log entre outros. Para mais informações do que fazer siga um guia, veja Sans' Incident Handling Guide (http://www.sans.org/y2k/DDoS.htm) ou CERT's Steps for Recovering from a UNIX or NT System Compromise (http://www.cert.org/tech_tips/root_compromise.html).

Algumas perguntas freqüentes de como lidar com um sistema Debian GNU/Linux estão disponíveis em 'Meu sistema é vulnerável! (Você tem certeza?)' on page 172.

10.2 Efetuando backup do sistema

Lembre-se que se você tem certeza de que o sistema foi comprometido você não pode confiar no software instalado ou em qualquer informação retornada por ele. Aplicações podem ser alteradas, módulos do kernel podem ser instalados e etc.

A melhor coisa a se fazer é uma cópia de backup completa do sistema de arquivo (usando o `dd`) depois de inicializar o sistema de uma mídia segura. Os CDROMs do Debian GNU/Linux podem ser utilizados para isto, já que eles fornecem um shell no console 2 quando a instalação é iniciada (acesse através do Alt+2 e pressione Enter). Do shell, efetue o backup das informações para outro host se possível (talvez um servidor de arquivos de rede através de NFS/FTP). Então qualquer análise da invasão ou reinstalação pode ser feita enquanto o sistema comprometido está off-line.

Se você tiver certeza de que um módulo do kernel com trojan comprometeu o sistema, você pode usar a imagem do kernel do CDROM do Debian no modo *rescue*. Inicie o GNU/Linux no modo *single user* para que nenhum outro processo com trojan seja executado depois do kernel.

10.3 Contate seu CERT local

O CERT (Computer and Emergency Response Team) é uma organização que pode te ajudar a recuperar o sistema comprometido. Existem CERTs espalhados por todo o mundo [3] e você

[3] Esta é a lista de alguns CERTS, para uma lista completa veja o FIRST Member Team information (http://www.first.org/about/organization/teams/index.html) (FIRST significa Forum of Incident Response and Security Teams): AusCERT (http://www.auscert.org.au) (Austrália), UNAM-CERT (http://www.unam-cert.unam.mx/) (México) CERT-Funet (http://www.cert.funet.fi) (Finlândia), DFN-CERT (http://www.dfn-cert.de) (Alemanha), RUS-CERT (http://cert.uni-stuttgart.de/) (Alemanha), CERT-IT (http://idea.sec.dsi.unim.it) (Itália), JPCERT/CC (http://www.jpcert.or.jp/) (Japão), UNINETT CERT (http://cert.uninett.no) (Noruega), HR-CERT (http://www.cert.hr) (Croácia) CERT Polskay (http://www.cert.pl) (Polônia), RU-CERT (http://www.cert.ru) (Rússia), SI-CERT (http://www.arnes.si/si-cert/) (Eslovênia) IRIS-CERT (http://www.rediris.es/cert/) (Espanha), SWITCH-CERT (http://www.switch.ch/cert/) (Suiça), TWCERT/CC (http://www.cert.org.tw) (Taiwan), e CERT/CC (http://www.cert.org) (US).

deve contatar seu CERT local caso ocorra algum incidente de segurança que comprometa seu sistema. As pessoas do CERT local são orientadas à ajudá-los.

Fornecer informações sobre os incidentes de segurança para o CERT local (ou o centro de coordenação do CERT), mesmo que você não precise de assistência, pode ajudar os outros a determinar se uma vulnerabilidade está disseminada na Internet e indicar que novas ferramentas de combate ao worm estão sendo utilizadas. Estas informações são usadas para fornecer à comunidade da Internet alertas sobre as atividades atuais dos incidentes de segurança (http://www.cert.org/current/), e para publicar notas sobre incidentes (http://www.cert.org/incident_notes/) e até mesmo alertas de segurança (http://www.cert.org/advisories/). Para informações mais detalhadas de como (e porquê) relatar um incidente leia o CERT's Incident Reporting Guidelines (http://www.cert.org/tech_tips/incident_reporting.html).

Você pode usar mecanismos menos formais se precisar de ajuda na recuperação de um sistema comprometido ou quiser discutir informações do incidente. Estes mecanismos incluem a lista de discussão sobre incidentes (http://marc.theaimsgroup.com/?l=incidents) e a lista de discussão sobre intrusos (http://marc.theaimsgroup.com/?l=intrusions).

10.4 Análise forense

Se você deseja recolher mais informações do ataque, o pacote `tct` (O Coroner's Toolkit de Dan Farmer e Wietse Venema) contém utilitários que realizam uma análise 'póstuma' do sistema. O `tct` permite que o usuário colete informações sobre arquivos excluídos, processos em execução e muito mais. Veja a documentação para mais informações. Você também pode conferir os pacotes similares Sleuthkit and Autopsy (http://www.sleuthkit.org/) desenvolvidos por Brian Carrier.

Algumas outras ferramentas que podem ser usadas para análise forense também são fornecidas pela distribuição Debian:

- `Fenris`.
- `Strace`.
- `Ltrace`.

Qualquer um desses pacotes podem ser usados para analisar binários anômalos (como os backdoors) para determinar como eles funcionam e o que eles fazem no sistema. Outras ferramentas comuns são o `ldd` (no pacote `libc6`), `strings` e `objdump` (ambos no pacote `binutils`).

Se você tentar fazer uma análise forense de um sistema comprometido com backdoors ou binários suspeitos, você deve fazê-la em um ambiente seguro (por exemplo em uma imagem `bochs` ou `flex86`, ou em um ambiente `chroot` utilizando um usuário com poucos privilégios). Caso contrário seu próprio sistema pode ser comprometido também!

Também, lembre-se que a análise forense deve ser feita sempre na cópia de backup dos dados, *nunca* nos dados originais, em caso dos dados serem alterados durante a análise e as evidências serem perdidas.

FIXME: This paragraph will hopefully provide more information about forensics in a Debian system in the coming future.

FIXME: talk on how to do a debsums on a stable system with the MD5sums on CD and with the recovered file system restored on a separate partition.

FIXME add pointers to forensic analysis papers (like the Honeynet's reverse challenge or David Dittirch's papers (http://staff.washington.edu/dittrich/).

Capítulo 11

Questões feitas com freqüência (FAQ)

Este capítulo introduz algumas das questões mais freqüentes da lista Debian security. Você deverá lê-las antes de postar lá ou senão as pessoas lhe dirão RTFM.

11.1 Tornando o sistema operacional Debian mais seguro

11.1.1 A Debian é mais segura que X?

Um sistema é tão seguro quanto um administrador é capaz de faze-lo. A instalação padrão dos serviços da Debian tenta ser *secura*, mas pode não ser paranóica como outros sistemas operacionais que instalam todos os serviços *desativados por padrão*. Em qualquer caso, o administrador de sistemas precisa adaptar a segurança do sistema a sua política de segurança local.

Para uma coleção de dados envolvendo vulnerabilidades de segurança de muitos sistemas operacionais, veja http://securityfocus.com/vulns/stats.shtml. Estes dados são úteis? O site lista diversos fatores a considerar quando estiver interpretando dados, e alerta que os dados não podem ser usados para comparar vulnerabilidades de um sistema operacional versus outro. [1] Também, tenha em mente que algumas das vulnerabilidades reportadas via bugs com relação a Debian, se aplicam somente ao repositório *unstable* (área de desenvolvimento).

A Debian é mais segura que as outras distribuições Linux (tal como Red Hat, SuSE...)?

Realmente não existem muitas diferenças entre as distribuições Linux, com exceção da instalação básica e do sistema de gerenciamento de pacotes. A maioria das distribuições compar-

[1] Neste exemplo, baseado nos dados da Securityfocus, pode ser visto que o Windows NT é mais seguro que o Linux, o que é uma afirmação questionável. Apesar de tudo, as distribuições do Linux geralmente oferecem mais aplicações comparadas ao Windows NT da Microsoft. Estas situações de *contagem de vulnerabilidades* são melhor descritas em Why Open Source Software / Free Software (OSS/FS)? Look at the Numbers! (http://www.dwheeler.com/oss_fs_why.html#security) por David A. Wheeler

tilham muitos dos aplicativos, com a diferença básica nas versões em que estes aplicativos são oferecidos com o lançamento da distribuição estável. Por exemplo, o kernel, Bind, Apache, OpenSSH, XFree, gcc, zlib, etc. são todos idênticos entre as distribuições de Linux.

Por exemplo, a Red Hat foi infeliz e ofereceu quando 1.2.3 era a atual, que em seguida foram encontrados problemas de segurança. Na Debian, por outro lado, foi sortuda e forneceu 1.2.4 que já possui a correção da falha. Este foi o caso no grande problema do rpc.statd (http://www.cert.org/advisories/CA-2000-17.html) diversos anos atrás.

Existe muita colaboração entre os respectivos times de segurança das maiores distribuições Linux. Atualizações de segurança conhecidas são raramente, se existirem, deixadas de lado por desenvolvedores de uma distribuição. O conhecimento de uma vulnerabilidade de segurança nunca é mantida isolada do conhecimento de desenvolvedores de outra distribuição, pois as correções são normalmente coordenadas com o autor ou através do CERT (http://www.cert.org). Como um resultado, as atualizações necessárias de segurança são geralmente lançadas ao mesmo tempo e a segurança relativa de diferentes distribuições são bem parecidas.

Uma das principais vantagens da Debian com relação a segurança é a facilidade de atualizações do sistema através do uso do `apt`. Aqui existem muitos outros aspectos da segurança na Debian a serem considerados:

- A Debian fornece mais ferramentas de segurança que outras distribuições, veja 'Ferramentas de segurança no Debian' on page 131.
- A instalação padrão da Debian é pequena (menos funcionalidades), e assim mais segura. Outras distribuições, em nome da funcionalidade, tem a tendência de instalarem diversos serviços por padrão e algumas vezes não estão corretamente configurados (lembre-se dos worms Ramen ou Lion (http://www.sans.org/y2k/lion.htm)). A instalação da Debian não é limitada como o OpenBSD (não existem daemons ativos por padrão), mas tem um bom compromisso. [2]
- A Debian documenta as melhores práticas de segurança em documentos como este.

11.1.2 Existem muitas falhas no sistema de tratamento de falhas da Debian. Isto significa que é muito vulnerável?

A distribuição Debian conta com um número grande e crescente de pacotes de software, provavelmente mais do que os fornecidos por muitos sistemas operacionais proprietários. Quanto mais pacotes instalados, maior o potencial de falhas de segurança em um determinado sistema.

Mais e mais pessoas estão examinando o código fonte por problemas. Existem muitos alertas relacionados com a auditoria de código fonte dos maiores componentes de software incluídos

[2] Sem mencionar o fato que algumas distribuições, tal como a Red Hat ou Mandrake, também estão permitindo que o usuário selecione *perfis de segurança* ou usando assistentes para ajudar na configuração de *firewalls pessoais*.

na Debian. Desta forma, tais auditorias de software mostram brechas de segurança, elas são corrigidas e um aviso é enviado para listas tal como Bugtraq.

Falhas que estão presentes na distribuição Debian normalmente também afetam outros distribuidores e vendedores. Verifique a seção "Específico da Debian: yes/no" no topo de cada aviso de segurança (DSA).

11.1.3 A Debian possui qualquer certificação relacionada a segurança?

Resposta curta: não.

Resposta longa: certificação custa dinheiro (especialmente se for uma certificação de segurança *séria*), ninguém dedicou seus recursos para para certificar a Debian GNU/Linux em qualquer nível de, por exemplo, Critérios comuns (http://niap.nist.gov/cc-scheme/st/). Se estiver interessado em ter uma distribuição de GNU/Linux seguramente certificada, tente fornecer os recursos necessários para tornar isto possível.

Existem pelo menos duas distribuições de Linux certificadas em diferentes níveis EAL (http://en.wikipedia.org/wiki/Evaluation_Assurance_Level). Note que alguns dos testes CC estão sendo integrados no Linux Testing Project (http://ltp.sourceforge.net) que está disponível na Debian através do pacote `ltp`.

11.1.4 Existe algum programa de fortalecimento para a Debian?

Sim. Bastille Linux (http://www.bastille-unix.org), originalmente orientado para outras distribuições de Linux (Red Hat e Mandrake), atualmente funciona com a Debian. Alguns passos estão sendo feitos para integrar as alterações feitas com a versão do autor no pacote da Debian, tendo o nome de `bastille`.

Algumas pessoas, no entanto, acreditam que uma ferramenta de fortalecimento não elimina a necessidade de se ter uma boa administração.

11.1.5 Eu desejo executar o serviço XYZ, qual eu devo escolher?

Um dos grandes potenciais da Debian é a grande variedade de escolhas disponíveis entre pacotes que oferecem a mesma funcionalidade (servidores de DNS, servidores de e-mail, servidores ftp, servidores web, etc.). Isto pode confundir o administrador novato ao tentar determinar que pacote é o mais adequado para você. O melhor para uma determinada situação depende de um balanceamento entre suas características e necessidades de segurança. Aqui estão algumas questões que devem ser feitas a você mesmo quando decidir entre pacotes parecidos:

- Existem um maintainer do código fonte do programa? Quando foi o último lançamento?
- O pacote está maduro? o número de versão realmente *não* mostra sua maturidade. Tente analisar o histórico de atualizações do software.

- Este programa é atormentado por falhas? Tem avisos de segurança relacionados a ele?
- Este programa oferece todas as funcionalidades que precisa? ele oferece mais do que você realmente precisa?

11.1.6 Como eu posso tornar o serviço XYZ mais seguro na Debian?

Você encontrará informações neste documento sobre como tornar alguns serviços (FTP, Bind) mais seguros na Debian GNU/Linux. Para serviços não cobertos aqui, verifique a documentação do programa, ou informações gerais sobre o Linux. Muitas das regras de segurança para sistemas Unix também se aplicam a Debian. Na maioria dos casos, o método para tornar um serviço X mais seguro na Debian é parecido com torná-lo mais seguro em qualquer outra distribuição de Linux (ou Unix, nesta importância).

11.1.7 Como posso remover todos os banners de serviços?

Se não gosta que os usuários que se conectam ao seu serviço de POP3 recebam informações sobre seu sistema (por exemplo), você pode querer remover (ou alterar) o banner que este serviço mostra para os usuários. [3] Fazer isto depende do programa que está executando para um determinado serviço. Por exemplo, no `postfix`, você poderá ajustar o banner SMTP no arquivo `/etc/postfix/main.cf`:

```
smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
```

Outros softwares não são fáceis de serem alterados. O `OpenSSH` precisará ser recompilado para alterar a versão que ele exibe. Tenha cuidado para não remover a primeira parte do banner (`SSH-2.0`), pois os clientes utilizam para identificar que protocolo é suportado por seu pacote.

11.1.8 Todos os pacotes da Debian são seguros?

O time de segurança da Debian não tem a possibilidade de analisar todos os pacotes incluídos na Debian procurando por vulnerabilidades de segurança em potencial, pois não existem recursos para auditar o código fonte de todo o projeto. No entanto, a Debian se beneficia da auditoria de código fonte feita por desenvolvedores que criam o programa.

Como um fato de importância, um desenvolvedor da Debian pode distribuir um Trojan em um pacote e não existe possibilidade de verificar isto. Até mesmo se for introduzido na estrutura da distribuição, seria impossível identificar todas as situações onde o trojan seria executado. Este é o motivo porque a Debian vem com a cláusula de licença *"sem garantias"*.

[3] Note que isto é "segurança pela obscuridade" e provavelmente este esforço não valerá a pena em longo termo.

No entanto, os usuários da Debian podem ter confiança no fato que que código estável tem uma audiência ampla e a maioria dos problemas foram descobertos durante o uso. A instalação de versões não testadas de programas em sistemas críticos é algo não recomendado (se não puder fornecer a auditoria de código necessária). Em qualquer caso, se for descoberta uma vulnerabilidade de segurança introduzida na distribuição, o processo usado para incluir pacote (usando assinaturas digitais) se certifica que o problema pode ser rastreado até o desenvolvedor. O projeto Debian não tem examinado isto levemente.

11.1.9 Porque alguns arquivos de logs/configuração tem permissão de leitura para qualquer um, isto não é inseguro?

É claro, você pode alterar as permissões padrões da Debian em seu sistema. A política atual relacionada com arquivos de log e configuração é que eles sejam lidos por todos *a não ser* que eles contenham informações sensíveis.

Tenha cuidado se fizer estas alterações pois:

- Alguns processos podem não ser capazes de gravar arquivos de log se restringir suas permissões.
- Alguns aplicativos podem deixar de funcionar se o arquivo de configuração que eles dependem não puder ser lido. Por exemplo, se você remover a permissão de leitura para todos do `/etc/samba/smb.conf`, o `smbclient` deixará de funcionar se for executado por um usuário normal.

FIXME: Verificar se isto está escrito na Política. Alguns pacotes (i.e. daemons de ftp) parecem forçar permissões diferentes.

11.1.10 Porque o /root/ (ou UsuarioX) tem permissões 755?

Como fato de importância, as mesmas questões são válidas para qualquer outro usuário. Como a instalação da Debian não coloca *qualquer* arquivo sob aquele diretório, não existe informações sensíveis a serem protegidas lá. Se você sentir que estas permissões são muito largas para seu sistema, considere altera-las para 750. Para os usuários, leia 'Limitando acesso a outras informações de usuários' on page 57.

A lista de discussão Debian security thread (`http://lists.debian.org/debian-devel/2000/debian-devel-200011/msg00783.html`) tem mais sobre este assunto.

11.1.11 Após instalar o grsec/firewall, comecei a receber muitas mensagens de console! como removê-las?

Se estiver recebendo mensagens de console e configurou o `/etc/syslog.conf` para redirecioná-las ou para arquivos ou para um TTY especial, você pode ver mensagens sendo direcionadas para a console.

O nível de registro padrão do console para qualquer kernel é 7, o que significa que qualquer mensagem que tem prioridade menor aparecerá no console. Normalmente, os firewalls (a regra LOG) e algumas outras ferramentas de segurança registram eventos em uma prioridade menor que esta, e assim, são enviadas diretamente para a console.

Para reduzir as mensagens enviadas para a console, você pode usar a opção `dmesg` (`-n`, veja `dmesg(8)`), que examina e *controla* o buffer do kernel. Para alterar isto após a próxima reinicialização, altere o `/etc/init.d/klogd` de:

```
KLOGD=""
```

para:

```
KLOGD="-c 4"
```

Use um número menor para `-c` se estiver ainda vendo as mensagens. Uma descrição dos diferentes níveis de logs podem ser encontrados no arquivo `/usr/include/sys/syslog.h`:

```
#define LOG_EMERG       0       /* o sistema está inutilizável */
#define LOG_ALERT       1       /* uma ação deve ser tomada imediatamente *
#define LOG_CRIT        2       /* condições críticas */
#define LOG_ERR         3       /* condições de erro */
#define LOG_WARNING     4       /* condições de alerta */
#define LOG_NOTICE      5       /* condição normal mas significante */
#define LOG_INFO        6       /* informativas */
#define LOG_DEBUG       7       /* mensagens a nível de depuração */
```

11.1.12 Usuários e grupos do sistema operacional

Todos os usuários do sistema são necessários?

Sim e não. A Debian vem com alguns usuários pré-definidos (identificação de usuários (UID) < 99 como descritos na Debian Policy (`http://www.debian.org/doc/debian-policy/`) ou `/usr/share/doc/base-passwd/README`) para facilitar a instalação de alguns serviços que requerem que sejam executados sob um usuário/UID apropriado. Se não tem a intenção de instalar novos serviços, você pode seguramente remover estes usuários que não são donos de qualquer arquivo em seu sistema e não executam qualquer serviço. Em qualquer caso, o comportamento padrão é que UIDs de 0 a 99 são reservadas para a Debian, e UIDs de 100 a 999 são criados por pacotes na instalação (e apagados quando o pacote e suas configurações são removidos do sistema).

Para encontrar facilmente que usuários não são donos de arquivos no sistema, execute o seguinte comando (execute-o como root, pois um usuário comum pode não ter permissões suficiente para entrar através de alguns diretórios sensíveis):

```
cut -f 1 -d : /etc/passwd | \
while read i; do find / -user "$i" | grep -q . && echo "$i"; done
```

Estes usuários são fornecidos pelo pacote `base-passwd`. Olhe em sua documentação por mais informações sobre como estes usuários são manipulados pelo sistema Debian. A lista de usuários padrões (com o grupo correspondente) segue:

- root: O root é (tipicamente) o superusuário.
- daemon: Alguns daemons não privilegiados que precisam gravar em arquivos no disco são executados como daemon.daemon (e.g., `portmap`, `atd`, provavelmente outros). Os daemons que não precisam ser donos de quaisquer arquivos são executados sob nobody.nogroup e daemons mais complexos ou com segurança em mente são executados como usuários dedicados. O usuário do daemon é prático para daemons instalados localmente.
- bin: mantido por razões históricas.
- sys: mesmo que bin. No entanto. /dev/vcs* e `/var/spool/cups` tem como donos o grupo sys.
- sync: O interpretador de comandos do usuário sync é `/bin/sync`. Assim se sua senha for ajustada para algo fácil de adivinhar (tal como “”), qualquer um pode fazer sync no sistema pela console, até mesmo se não possuir uma conta.
- games: Muitos jogos são SETGID para games assim eles podem gravar seus arquivos de pontuações. Isto é explicado na policy.
- man: O programa man (algumas vezes) é executado como usuário man, assim ele poderá gravar páginas de manuais em `/var/cache/man`
- lp: Usado por daemons de impressão.
- mail: Caixas de correios em `/var/mail` tem como dono o grupo mail, como explicado pela policy. O usuário e grupo também são usados para outros propósitos por vários MTA’s.
- news: Vários servidores de notícias e outros programas associados (tal como o `suck`) utilizam usuário e grupo news de várias formas. Os arquivos no spool de notícias tem freqüentemente como dono o usuário e grupo news. Os programas tais como `inews` que são usados para postar notícias tipicamente usam SETGID para o grupo news.
- uucp: O usuário e grupo uucp são usados pelo subsistema UUCP. Ele é dono do spool e arquivos de configuração. Usuários no grupo uucp podem executar o uucico.
- proxy: Assim como o daemon, este usuário e grupo são usados por alguns daemons (especificamente, daemons de proxy) que precisam de identificação de usuários dedicadas para ser dono de arquivos. Por exemplo, o grupo proxy é usado pelo `pdnsd` e `squid` para serem executados como o usuário proxy.

- majordom: `Majordomo` tem uma UID estaticamente alocada em sistemas Debian por razões históricas. Ele não é instalado em novos sistemas.
- postgres: Os bancos de dados do `Postgresql` tem como dono este usuário e grupo. Todos os arquivos sob `/var/lib/postgresql` tem como dono este usuário para forçar segurança de forma apropriada.
- www-data: Alguns servidores web são executados sob www-data. O conteúdo web *não* deve ter como dono este usuário, ou um servidor web comprometido poderia ser capaz de regravar um site de internet. Dados gravados por servidores web, incluindo arquivos de logs, terão que ter como dono www-data.
- backup: Assim as responsabilidades de backup/restauração podem ser localmente delegadas para alguém sem permissões completas de usuário root.
- operator: O operador é historicamente (e praticamente) a única conta de "usuário" que pode efetuar login remotamente, e não depende do NIS/NFS.
- list: Os arquivos de listas de discussões e dados tem como dono este usuário e grupo. Alguns programas de listas de discussões podem ser executadas também sobe este usuário.
- irc: Usado por daemons de irc. É necessário um usuário alocado estaticamente somente por causa de um bug no `ircd`, que faz SETUID()s de si mesmo para a UID especificada na inicialização.
- gnats.
- nobody, nogroup: Daemons que não tem necessidade de serem donos de quaisquer arquivos são executados sob o usuário nobody e grupo nogroup. Assim, nenhum arquivo existente no sistema devem ter como donos este usuário ou grupo.

Outros grupos que não tem um usuário associado:

- adm: O grupo adm é usado para tarefas de monitoramento do sistema. Os membros deste grupo podem ler a maioria dos arquivos de log em `/var/log` e podem usar o xconsole. Historicamente, o `/var/log` foi `/usr/adm` (e depois `/var/adm`), isto explica o nome do grupo.
- tty: Os dispositivos TTY tem como dono este grupo. Eles são usados pelas ferramentas write e wall para permitir escrever para pessoas conectadas em outras TTYs.
- disk: Acesso direto a disco. Muito equivalente ao acesso root.
- kmem: /dev/kmem e arquivos similares são lidos por este grupo. Isto é mais uma relíquia do BSD, mas alguns programas que precisam de acesso de leitura direto a memória do sistema podem fazer SETGID para o grupo kmem.
- dialout: Acesso direto e completo a portas seriais. Membros deste grupo podem reconfigurar o modem, discar para qualquer lugar, etc.

- dip: O nome do grupo vem de "Dial-up IP", e membros que pertencem ao grupo dip podem usar ferramentas como o `ppp`, `dip`, `wvdial`, etc. para realizar uma conexão. Os usuários neste grupo não podem reconfigurar o modem, mas podem executar programas para fazerem uso dele.
- fax: Permite que membros usem programas de fax para ler/enviar faxes.
- voice: Voicemail, útil para sistemas que usam modens como secretárias eletrônicas.
- cdrom: Este grupo pode ser usado localmente para dar ao grupo de usuários acesso a unidade de CDROM.
- floppy: Este grupo pode ser usado localmente par dar a um grupo de usuários acesso a unidade de disquetes.
- tape: Este grupo pode ser usado localmente para dar a um grupo de usuários acesso a uma unidade de fita.
- sudo: Membros dentro deste grupo não precisam digitar sua senha quando estiverem fazendo o uso do `sudo`. Veja `/usr/share/doc/sudo/OPTIONS`.
- audio: Este grupo pode ser usado localmente para dar a um grupo de usuários acesso a um dispositivo de audio.
- src: Este grupo é dono de código fonte, incluindo arquivos em `/usr/src`. Ele pode ser usado para dar a um usuário a habilidade de gerenciar código fonte do sistema.
- shadow: O arquivo `/etc/shadow` é lido por este grupo. Alguns programas que precisam ser capazes de acessar o arquivo tem SETGID ajustados para shadow.
- utmp: Este grupo pode gravar para o arquivo `/var/run/utmp` e similares. Programas que precisam se capazes de gravar para ele usam SETGID para utmp.
- video: Este grupo é usado localmente para dar a um conjunto de usuários permissões de acesso a dispositivos de vídeo.
- staff: Permite que usuários adicionem modificações locais ao sistema (`/usr/local`, `/home`) sem necessidade de privilégios de usuário root. Compare com o grupo "adm", que é mais relacionado a segurança/monitoramento.
- users: Enquanto usuários de sistemas Debian usam seus grupos privados de sistema por padrão (cada usuário tem seu próprio grupo), alguns preferem usar um grupo de sistema mais tradicional, no qual cada usuário é membro de seu grupo.

Quais são as diferenças entre os grupos adm e staff?

Componentes do grupo "adm" são geralmente administradores e neste grupo as permissões os permitem ler arquivos de log sem utilizar `su`. O grupo "staff" são geralmente administradores junior e de suporte, permitindo que trabalhem em `/usr/local` e criarem diretórios em `/home`.

11.1.13 Porque existe um novo grupo quando adiciono um novo usuário? (ou porque a Debian cria um novo grupo para cada usuário?)

O comportamento padrão na Debian é que cada usuário tem seu próprio e privado grupo. O esquema tradicional do UN*X coloca todos os usuários no grupo *users*. Grupos adicionais foram criados e usados para restringir o acesso a arquivos compartilhados associados com diferentes diretórios de projetos. O gerenciamento de arquivos se torna difícil quando apenas um usuário trabalha em múltiplos projetos, porque quando alguém cria um arquivo, ele é associado com o grupo primário do grupo que ele pertence (e.g. "users").

O método da Debian resolve este problema associando a cada usuário seu próprio grupo; assim com a máscara apropriada (0002) e o bit SETGID ajustado em um diretório determinado de projetos, o grupo correto é automaticamente designado para arquivos criados naquele diretório. Isto facilita a vida de pessoas que trabalham em múltiplos projetos, porque elas não terão que alterar os grupos e umasks quando estiverem trabalhando em arquivos compartilhados.

Você pode, no entanto, alterar este comportamento modificando o `/etc/adduser.conf`. Altere a variável *USERGROUPS* para "no", assim um novo grupo não será criado quando o novo usuário for criado. Também, altere *USERS_GID* para a identificação de grupo a que os usuários pertencem.

11.1.14 Questões relacionadas a serviços e portas abertas

Porque todos os serviços são ativados durante a instalação?

Esta é simplesmente uma aproximação do problema de sendo, de um lado, consciente de segurança e por outro lado amigável ao usuário. De forma contrária a OpenBSD, que desativa todos os serviços a não ser que sejam ativados pelo administrador, a Debian GNU/Linux ativa todos os serviços instalados a não ser que sejam desativados (veja 'Desabilitando daemons de serviço' on page 33 para mais informações). Afinal, você instalou o serviço, não foi?

Existem muitas discussões nas listas de discussões da Debian (ambas na debian-devel e na debian-security) com relação a qual é a melhor estratégia para a instalação padrão. No entanto, no momento em que isto foi escrito (Março de 2002), ainda não existia um consenso.

Posso remover o `inetd`?

O `Inetd` não é fácil de remover pois o pacote `netbase` depende do pacote que o fornece (`netkit-inetd`). Se deseja removê-lo, você poderá ou desativá-lo (veja 'Desabilitando daemons de serviço' on page 33) ou remover o pacote usando o pacote `equivs`.

Porque eu tenho a porta 111 aberta?

A porta 111 é usada pelo portmapper sunrpc e é instalada por padrão como parte do sistema de instalação básico da Debian, pois não existe a necessidade de saber quando o programa do

usuário precisa do RPC para funcionar adequadamente. Em qualquer caso, ele é mais usado pelo NFS. Se não precisar dele, remova-o como explicado na seção 'Tornando serviços RPC mais seguros' on page 102.

Em versões do pacote portmap maiores que a 5-5 você poderá ter o portmapper instalado mas escutando somente em localhost (modificando o `/etc/default/portmap`)

Para que a porta 113 (`identd`) é usada?

O serviço ident é um serviço de autenticação que identifica o dono de uma conexão TCP/IP para o servidor remoto que está aceitando a conexão. Tipicamente, quando um usuário se conecta ao servidor remoto, o `inetd` do sistema remoto envia uma requisição à porta 113 para procurar informações sobre o dono. É freqüentemente usada em servidores de e-mails, FTP e IRC, e também podem ser usadas para descobrir que usuário em seu sistema local está atacando um sistema remoto.

Existem discussões extensivas relacionadas a segurança do `identd` (Veja mailing list archives (http://lists.debian.org/debian-security/2001/debian-security-200108/msg00297.html)). Em geral, o `identd` é mais útil em um sistema multi-usuário que em uma estação de trabalho simples. Se não tiver um uso para ele, desative-o, assim você não estará deixando um serviço aberto para o mundo lá fora. Se decidir fazer um firewall na porta do ident, *por favor* use a política reject e não a deny, caso contrário uma conexão para o servidor usando o `identd` travará até que o tempo limite expire (veja questões relacionadas a reject ou deny (http://logi.cc/linux/reject_or_deny.php3)).

Tenho serviços usando a porta 1 e 6, o que são e como posso removê-las?

Se executar o comando `netstat -an` e receber como retorno:

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
raw        0      0 0.0.0.0:1               0.0.0.0:*               7
-
raw        0      0 0.0.0.0:6               0.0.0.0:*               7
-
```

Você *não* está vendo processos escutando na porta TCP/UDP 1 e 6. De fato, você está vendo um processo escutando em um soquete *cru* pelos protocolos 1 (ICMP) e 6 (TCP). Tal comportamento é normal para Trojans e alguns sistemas de detecção de intrusão como o `iipl`, `iplogger` e `portsentry`. Se tiver estes pacotes simplesmente os remova. Se não tiver, tente executar a opção `-p` do netstat (processo) para ver que processo é dono destas portas.

Encontrei a porta XYZ aberta, posso fechá-la?

Sim, com certeza. As portas que está deixando abertas devem aderir a política individual do seu site com relação a serviços públicos disponíveis para outras redes. Verifique se estão sendo abertas pelo `inetd` (veja 'Desabilitando o `inetd` ou seus serviços' on page 34) ou instalando pacotes individuais e tome as medidas apropriadas (i.e, configure o inetd, remova o pacote, evite executá-lo na inicialização).

Removendo serviços do `/etc/services` ajudará a tornar minha máquina mais segura?

Não o `/etc/services` somente oferece o mapeamento entre um nome virtual e um número dado de porta. A remoção de nomes deste arquivo (geralmente) não evitará que os serviços sejam iniciados. Alguns daemons podem não ser executados se o `/etc/services` for modificado mas isto não é a norma. Para desativar apropriadamente o serviço, veja 'Desabilitando daemons de serviço' on page 33.

11.1.15 Assuntos comuns relacionados a segurança

Perdi minha senha e não posso acessar o sistema!

Os passos que precisa fazer para se recuperar disto depende se aplicou ou não os procedimentos necessários para limitar o acesso ao `lilo` e da BIOS do seu sistema.

Se limitou ambos, precisará desativar a configuração de BIOS que somente lhe permite inicializar através do disco rígido antes de prosseguir. Se tiver também perdido a senha da sua BIOS, você terá que resetar a sua BIOS abrindo o computador e removendo manualmente a bateria que mantém os dados da BIOS>

Assim que permitir a inicialização através da unidade de CD-ROM ou ativação da unidade de disquete, faça o seguinte:

- Inicialize através de um disquete de recuperação e inicie o kernel
- Vá até o console virtual (Alt+F2)
- Monte o disco rígido onde o sistema de arquivos raíz (`/`) está
- Edite o arquivo `/etc/shadow` (o disquete de recuperação da Debian 2.2 vem com o editor `ae` e a Debian 3.0 vem com o `nano-tiny` que é similar ao `vi`) e altere a linha:

  ```
  root:asdfjgl29gl0341274075:XXXX:X:XXXX:X::: (X=um número qualquer)
  ```

  para:

  ```
  root::XXXX:X:XXXX:X:::
  ```

Isto removerá a senha de root perdida, contida no primeiro campo separado por dois pontos após o nome do usuário. Salve o arquivo, reinicie o sistema e faça login como usuário root usando uma senha em branco. Lembre-se de adicionar uma nova senha. Isto funcionará a menos que tenha configurado o sistema de forma mais restrita, ou seja, não permitindo que usuários utilizem senhas em branco ou não permitindo o login do usuário root através do console.

Se adicionou estas características, você precisará entrar em modo monousuário. Se o LILO foi restringido, será necessário re-executar o `lilo` após alterar a senha de root acima. Este truque é necessário pois seu `/etc/lilo.conf` precisa ser mexido devido ao sistema de arquivos raíz (/) ser um disco ram e não um disco rígido real.

Assim que a restrição do LILO for removida, tente o seguinte:

- Pressione as teclas Alt, shift e Control antes do sistema terminar o processo de inicialização, assim você terá acesso ao aviso de comandos do LILO.
- Digite `linux single`, `linux init=/bin/sh` ou `linux 1` na linha de comandos.
- Isto lhe dará um aviso de comandos do shell em modo monousuário (ele perguntará por uma senha, mas você já a conhece)
- Remonte sua partição raíz (/) usando o comando mount.

```
# mount -o remount,rw /
```

- Altere a senha do usuário root com o comando `passwd` (como você é o superusuário, o sistema não perguntará a senha anterior).

11.1.16 Como posso configurar um serviço para meus usuários sem lhes dar uma conta de acesso ao shell?

Por exemplo, se você quer configurar um serviço POP, você não precisará definir uma conta para cada usuário que esteja usando. É melhor configurar uma autenticação baseada em diretório através de um serviço externo (como Radius, LDAP ou banco de dados SQL). Apenas instale a biblioteca PAM apropriada (`libpam-radius-auth`, `libpam-ldap`, `libpam-pgsql` ou `libpam-mysql`), leia a documentação (para iniciantes, veja 'Autenticação do Usuário: PAM' on page 47) e configure o serviço que será ativado pelo PAM para usar o método de autenticação que escolheu. Isto é feito editando-se os arquivos sob o diretório `/etc/pam.d/` para seu serviço e modificando o

```
auth    required    pam_unix_auth.so shadow nullok use_first_pass
```

para, por exemplo, ldap:

```
auth    required    pam_ldap.so
```

No caso de diretórios LDAP, alguns serviços oferecem esquemas LDAP que devem ser incluídos em seu diretório e são necessários para a utilização de autenticação LDAP. Se estiver usando um banco de dados relacional, uma dica útil é usar a cláusula *where* quando estiver configurando os módulos do PAM. Por exemplo, se tiver um banco de dados com os seguintes atributos na tabela:

```
(user_id, user_name, realname, shell, password, UID, GID, homedir, sys, pop
```

Tornando os serviços campos de atributos boleanos, você poderá usa-los para permitir ou negar acesso a diferentes serviços apenas inserindo as linhas apropriadas nos seguintes arquivos:

- `/etc/pam.d/imap:where=imap=1`.
- `/etc/pam.d/qpopper:where=pop=1`.
- `/etc/nss-mysql*.conf:users.where_clause = user.sys = 1;`.
- `/etc/proftpd.conf:SQLWhereClause "ftp=1"`.

11.2 Meu sistema é vulnerável! (Você tem certeza?)

11.2.1 O scanner de vulnerabilidade X diz que meu sistema Debian é vulnerável!

Muitos scanners de avaliação de vulnerabilidades indicarão falso positivos quando forem usados em sistemas Debian, pois podem somente usar checagem de versões para determinar se uma determinada versão de pacote é vulnerável, mas realmente não testam a vulnerabilidade de segurança propriamente dita. Pois a Debian não muda os números de versões quando corrige um pacote (muitas vezes a correção feita em versões novas são reproduzidas nas atuais), algumas ferramentas tendem a achar que um sistema Debian atualizado está vulnerável, quando não está.

Se você acha que o seu sistema está atualizado com patches de segurança, você pode querer usar as referências cruzadas com o banco de dados de vulnerabilidades publicados com os DSAs (veja 'Debian Security Advisories' on page 114) para afastar a possibilidade de falsos positivos, se a ferramenta que estiver usando inclui referências do CVE.

11.2.2 Eu vi um ataque em meus logs de sistema. Meu sistema foi comprometido?

Um traço de ataque nem sempre significa que seu sistema foi comprometido, e você deverá fazer os passos tradicionais para determinar se o sistema está comprometido (veja 'Depois do comprometimento do sistema (resposta a incidentes)' on page 155). Também, note que o fato de ver os ataques nos logs pode significar que seu sistema está vulnerável a ele (um invasor determinado pode ter usado outras vulnerabilidades que não sejam a que você viu, no entanto).

11.2.3 Eu vi algumas linhas estranhas "MARK" em meus logs: Eu fui comprometido?

Você pode achar as seguintes linhas nos seus logs de sistema:

```
Dec 30 07:33:36 debian -- MARK --
Dec 30 07:53:36 debian -- MARK --
Dec 30 08:13:36 debian -- MARK --
```

Isto não indica qualquer tipo de comprometimento e os usuários que estão mudando de versão da Debian devem achar isto estranho. Se o seu sistema não tem uma carga alta (ou muitos serviços ativos), estas linhas devem aparecer entre seus logs. Isto é uma indicação que seu daemon do `syslogd` está sendo executado de forma apropriada. Texto extraído da página de manual `syslogd(8)`:

```
-m intervalo
       O syslogd registra uma marca de horário regularmente. O
       intervalo padrão entre duas linhas -- MARK -- é de 20 minutos.
       Isto pode ser alterado com esta opção.
       O intervalo de zero, desativa totalmente este recurso.
```

11.2.4 Encontrei usuários usando o "su" em meus logs: Eu fui comprometido?

Você pode encontrar linhas em seus logs como:

```
Apr  1 09:25:01 server su[30315]: + ??? root-nobody
Apr  1 09:25:01 server PAM_unix[30315]: (su) session opened for user nobody
```

Não se preocupe muito. Verifique para ver se estas mensagens são devido a tarefas do `cron` (normalmente `/etc/cron.daily/find` ou `logrotate`):

```
$ grep 25 /etc/crontab
25 6    * * *   root    test -e /usr/sbin/anacron || run-parts --report
/etc/cron.daily
$ grep nobody /etc/cron.daily/*
find:cd / && updatedb --localuser=nobody 2>/dev/null
```

11.2.5 Encontrei um possível "SYN flooding" em meus logs: Estou sob um ataque?

Se ver linhas como estas em seus logs:

```
May 1 12:35:25 linux kernel: possible SYN flooding on port X. Sending cooki
May 1 12:36:25 linux kernel: possible SYN flooding on port X. Sending cooki
May 1 12:37:25 linux kernel: possible SYN flooding on port X. Sending cooki
May 1 13:43:11 linux kernel: possible SYN flooding on port X. Sending cooki
```

Verifique se existe um número alto de conexões ao servidor usando o `netstat`, por exemplo:

```
linux:~# netstat -ant | grep SYN_RECV | wc -l
   9000
```

Isto é uma indicação de ataque de negação de serviço (denial of service - DoS) contra a porta X do seu sistema (mais provável contra um serviço público tal como um servidor web ou servidor de e-mails). Você deverá ativar os SynCookies TCP em seu kernel, veja 'Configurando Syncookies' on page 73. Note, no entanto, que um ataque DoS pode sobrecarregar sua rede até mesmo se você puder parar de fazê-lo travar seus sistemas (devido ao número de descritores de arquivos sendo reduzidos, o sistema pode parar de responder até que o tempo limite de algumas conexões se esgote). O único método efetivo de parar este ataque é contactar seu provedor de rede.

11.2.6 Encontrei seções de root estranhas em meus logs: Eu fui comprometido?

Se ver estes tipos de entradas em seu arquivo `/var/log/auth.log`:

```
May 2 11:55:02 linux PAM_unix[1477]: (cron) session closed for user root
May 2 11:55:02 linux PAM_unix[1476]: (cron) session closed for user root
May 2 12:00:01 linux PAM_unix[1536]: (cron) session opened for user root by
(UID=0)
May 2 12:00:02 linux PAM_unix[1536]: (cron) session closed for user root
```

Estas são devido a uma tarefa do `cron` sendo executada (neste exemplo, a cada cinco minutos). Para determinar que programa é responsável por estas tarefas, verifique as tarefas nos diretórios: `/etc/crontab`, `/etc/cron.d`, `/etc/crond.daily` e do root `crontab` sob `/var/spool/cron/crontabs`.

11.2.7 Sofri uma invasão, o que faço?

Existem diversos passos que deve fazer no caso de uma invasão:

- Verifique se o seu sistema está atualizado com as atualizações de segurança de vulnerabilidades publicadas. Se o seu sistema estiver vulnerável, as chances do sistema estar de fato comprometido são maiores. As chances crescem mais se a vulnerabilidade foi conhecida durante algum tempo, pois normalmente existem mais atividades com relação a vulnerabilidades antigas. Aqui está um link para As 20 maiores Vulnerabilidades de Segurança (http://www.sans.org/top20/).

- Leia este documento, especialmente a seção 'Depois do comprometimento do sistema (resposta a incidentes)' on page 155
- Peça assistência. Você deverá usar a lista de discussão debian-security para perguntar sobre como recuperar/corrigir seu sistema.
- Notifique seu CERT (`http://www.cert.org`) local (caso ele exista, caso contrário você deverá considerar o contato direto com o CERT). Isto pode ou não ajudar você, mas, pelo menos, informará o CERT de ataques que estejam acontecendo. Esta informação é muito valiosa em determinar que ferramentas e ataques estão sendo usados pela comunidade *chapéu preto*.

11.2.8 Como posso rastrear um ataque?

Olhando os logs (caso não tenham sido mexidos) usando sistemas de detecção de intrusão (veja 'Configure um sistema de Detecção de Intrusão' on page 147), `traceroute`, `whois` e ferramentas parecidas (incluindo análise forense), você pode ser capaz de detectar um ataque até a sua origem. O método que pode reagir a esta informação depende solenemente de sua política de segurança e o que *você* considera um ataque. Um scan remoto é um ataque? É um teste de vulnerabilidade um ataque?

11.2.9 O programa X na Debian é vulnerável, o que fazer?

Primeiro, leve um momento para se certificar se a vulnerabilidade foi anunciada em listas de discussões de segurança públicas (como a Bugtraq) ou outros fóruns. O time da Debian Security se mantém atualizada com estas listas, assim elas também deverão ter conhecimento do problema. Não faça qualquer outra ações se você ver um anúncio em `http://security.debian.org`.

Caso nenhuma informação tenha sido publicada, por favor envie um e-mail sobre o(s) pacote(s) afetado(s), assim como uma descrição detalhada da vulnerabilidade (código que comprova isto também é válido) para team@security.debian.org (`mailto:team@security.debian.org`). Isto lhe colocará em contato com o time de segurança da Debian.

11.2.10 O número de versão de um pacote indica que eu ainda estou usando uma versão vulnerável!

Ao invés de atualizar para uma versão nova, a Debian adapta as correções para a versão que é fornecida com o lançamento estável. A razão disto é para ter certeza que o lançamento estável altere o mínimo possível, assim as coisas não alterarão ou quebrarão de forma inesperada como resultado de uma correção de falha. Você pode verificar se está executando uma versão segura de pacote olhando nos logs de alterações do pacote ou comparando seu número de versão exato (versão do autor - traço- lançamento da Debian) com o número de versão indicado no aviso de segurança da Debian.

11.2.11 Programas específicos

`proftpd` é vulnerável ao ataque de negação de serviço.

Adicione `DenyFilter \*.*/` em seu arquivo de configuração, e para mais informações veja `http://www.proftpd.org/critbugs.html`.

Após instalar o `portsentry` muitas portas são abertas

Este é simplesmente o método como o `portsentry` funciona. Ele abre cerca de vinte portas não usadas para tentar identificar port scans.

11.3 Questões relacionadas ao time de segurança da Debian

Esta informação foi derivada de Debian Security FAQ (`http://www.debian.org/security/faq`). Este texto inclui as informações de 19 de Novembro e oferece algumas outras questões comuns perguntadas na lista de discussão debian-security.

11.3.1 O que é um Aviso de Segurança da Debian (Debian Security Advisory - DSA)?

É a informação enviada pelo Time de segurança da Debian (veja abaixo) com relação a descoberta e correção de uma vulnerabilidade relacionada a segurança em um pacote disponível na Debian GNU/Linux. DSAs assinados são enviados à lista de discussões públicas (debian-security-announce) e postados no web site da Debian (ambos na página inicial e na área de segurança (`http://www.debian.org/security/`)).

OS DSAs incluem informações sobre o pacote afetado, o problema de segurança descoberto e onde obter pacotes atualizados (com seus respectivos cálculos MD5).

11.3.2 As assinaturas nos avisos de segurança da Debian não são verificados corretamente!

É mais provável que este problema esteja sendo causado por algo em sua máquina. A lista debian-security-announce (`http://www.debian.org/security/faq`) tem um filtro que somente permite postagem de mensagens de um dos membros do time de segurança da Debian.

É mais provável que algumas peças do software de e-mail estejam alterando as mensagens, quebrando assim a assinatura. Tenha certeza que seu programa não faça qualquer encodificação ou decodificação MIME ou conversão de tab/espaços.

Acusados conhecidos são fetchmail (com a opção mimedecode ativada), formail (somente do procmail 3.14) e o evolution.

11.3.3 Como a segurança é tratada na Debian?

Assim que o time de segurança recebe a notificação de um incidente, um dos membros revisa e considera o impacto no lançamento estável da Debian (i.e. se é vulnerável ou não). Se o seu sistema é vulnerável, nós trabalharemos para corrigir o problema. O mantenedor do pacote também é contactado, caso ele já não tenha contactado o time de segurança. Finalmente, a correção é testada e novos pacotes são preparados, que então são compilados em todas as arquiteturas estáveis e após isto feito o upload. Após isto feito, um aviso de segurança é publicado.

11.3.4 Porque vocês estão trabalhando em uma versão antiga daquele pacote?

A regra de conduta mais importante quando criar um novo pacote que corrige um problema de segurança é fazer menos alterações possíveis. Nossos usuários e desenvolvedores se preocupam com o exato comportamento de um lançamento quando é feito, assim qualquer alteração que nós fazemos, pode possivelmente tornar o programa não funcional no sistema de alguém. Isto é especialmente verdadeiro no caso de bibliotecas: tenha certeza de nunca alterar a interface de aplicação do programa (API) ou a interface de aplicação do Binário (ABI), não importa quanto pequena a alteração seja.

Isto significa que não é uma boa solução mover para uma nova versão do autor do pacote, ao invés disto as alterações importantes devem ser feitas na versão atual (backportadas). Geralmente os autores ajudam se necessário, senão o time de segurança da Debian poderá ser capaz de ajudar.

Em alguns casos não é possível adaptar uma atualização de segurança para uma versão antiga, por exemplo, quando foi necessária a alteração de uma grande quantidade de código fonte. Se isto acontecer, é necessário mover para uma nova versão do autor, mas isto deve ser coordenado de forma muito pró ativa com o time de segurança.

11.3.5 Qual é a política para um pacote corrigido aparecer em security.debian.org?

Quebras de segurança na distribuição estável garante um pacote em security.debian.org. Qualquer outra coisa não. O tamanho do comprometimento não é o problema real aqui. Normalmente o time de segurança preparará pacotes juntos com o mantenedor do pacote. Fornecendo os rastros dos testes de alguém (confiável) sobre o problema e tendo todos os pacotes necessários compilados e enviados para o time de segurança, até mesmo problemas de segurança simples farão o pacote ser enviado para security.debian.org. Por favor, veja baixo.

11.3.6 O número de versão de um pacote indica que eu ainda estou usando uma versão vulnerável!

Ao invés de atualizar para uma nova versão, nós adaptamos as correções para a versão estável que é fornecida com o lançamento estável. A razão para fazermos isto é para ter certeza que

a versão estável mude o mínimo possível assim as coisas não serão alteradas ou quebrarão de forma inesperada como resultado de um problema de segurança. Você poderá verificar se está executando uma versão segura de um pacote olhando nos logs de alterações do pacote (changelog), ou comparando seu número de versão exato com o número de versão indicado no aviso de segurança da Debian (DSA).

11.3.7 Como a segurança é tratada na `testing` e `unstable`?

A resposta curta é: não é. Os lançamentos testing e unstable estão movendo rapidamente objetos e o time de segurança não possui os recursos necessários para suportá-las apropriadamente. Se desejar ter um servidor seguro (e estável) você é fortemente encorajado para permanecer usando a stable (estável). No entanto, as secretárias de segurança tentarão corrigir problemas na testing e unstable após terem sido corrigidos na stable (distribuição estável).

Em alguns casos, no entanto, o repositório unstable (instável) recebe correções de segurança de forma rápida, porque estas correções geralmente são disponibilizadas de forma rápida para o autor (outras versões, como as que estão no repositório stable, geralmente precisam ser adaptadas).

11.3.8 Eu uso uma versão antiga da Debian, ela é suportada pelo time de segurança?

Não. Infelizmente o time de segurança da Debian não pode tomar conta de ambos os lançamentos estáveis (oficialmente, também a unstable) e outros lançamentos antigos. No entanto, você poderá esperar por atualizações de segurança por um período limitado de tempo (normalmente alguns meses) imediatamente seguindo o lançamento de uma nova distribuição da Debian.

11.3.9 Porque não existem mirrors oficiais de security.debian.org?

O propósito de security.debian.org é tornar atualizações de segurança rapidamente disponíveis quanto possível. Os mirrors adicionariam uma complexidade extra que não é necessária e causariam frustração caso não estivessem sendo atualizados.

11.3.10 Eu vi o DSA 100 e DSA 102, o que aconteceu com o DSA 101?

Diversos distribuidores (a maioria de GNU/Linux, mas também de BSD e derivados) coordenam avisos de segurança para alguns incidentes e concordam em ter uma limite de tempo particular de lançamento, assim todos os distribuidores são capazes de lançar um aviso em conjunto. Isto foi decidido com a intenção de não existirem discriminações entre alguns distribuidores que precisam de mais tempo (e.g. quando o distribuidor passou pacotes através de grandes testes de qualidade ou precisa manter o suporte a diversas arquiteturas ou distribuições binários). Nosso próprio time de segurança também prepara avisos de forma pró

ativa. Toda vez que estiver acontecendo, outros problemas de segurança serão analisados antes de um aviso ser lançado, e assim deixando alguns números de avisos de lado temporariamente.

11.3.11 Como posso contactar o time de segurança?

Informações de segurança podem ser enviadas para security@debian.org (mailto:security@debian.org), que é lida por todos os desenvolvedores da Debian. Se tiver informações sensíveis, por favor use team@security.debian.org (mailto:team@security.debian.org) que é lida somente por membros. Caso a mensagem puder ser encriptada pela chave de contato do time de segurança da Debian (key ID 0x363CCD95 (http://pgpkeys.pca.dfn.de:11371/pks/lookup?search=0x363CCD95op=vindex)).

11.3.12 Qual é a diferença entre security@debian.org e debian-security@lists.debian.org?

Quando envia uma mensagem para security@debian.org, ela é enviada apara a lista de discussão de desenvolvedores (debian-private). Todos os desenvolvedores da Debian estão inscritos nesta lista e as postagens são mantidas privadas (i.e. não são arquivadas no site público da internet). A lista de discussão pública, debian-security@lists.debian.org, é aberta para qualquer pessoa que deseja se inscrever (http://www.debian.org/MailingLists/) e existem arquivos que podem ser pesquisados disponíveis aqui (http://lists.debian.org/search.html).

11.3.13 Como posso contribuir com o time de segurança da Debian?

- Contribuindo com este documento, corrigindo parágrafos marcados com FIXME ou fornecendo novos conteúdos. A documentação é importante e reduz a carga de perguntas de assuntos simples. A tradução desta documentação em outros idiomas também é de grande ajuda.
- Empacotando aplicativos que são úteis para a checagem e fortalecimento de um sistema Debian GNU/Linux. Se não for um desenvolvedor, envie uma falha sobre o WNPP (http://www.debian.org/devel/wnpp/) e pergunte pelo software que acha que poderia ser útil, mas que atualmente não é fornecido.
- Audite os programas na Debian ou resolva bugs de segurança e reporte assuntos para security@debian.org. Trabalhar em outros projetos como o Projeto de Auditoria e Segurança do Kernel do Linux (http://kernel-audit.sourceforge.net/) ou o Projeto de Segurança e Auditoria do Linux (http://www.lsap.org/) também aumenta a segurança da Debian GNU/Linux, pois as contribuições eventualmente também ajudarão aqui.

Em todos os casos, po favor revise cada problema antes de enviá-lo para security@debian.org. Se for capaz de fornecer patches, isto aceleraria o processo. Não redirecione mensagens de

listas de bugtraq, pois eles já foram recebidos. O fornecimento de informações adicionais, no entanto, é sempre uma ótima idéia.

11.3.14 quem compõe o time de segurança?

O time de segurança da Debian é composto de cinco membros e duas secretárias. O time de segurança por si mesmo recomenda pessoas para que façam parte do time.

11.3.15 O time de segurança verifica cada novo pacote que entra na Debian?

Não, o time de segurança da Debian não verifica cada pacote e não existe um método de checagem automático (lintian) para detectar novos pacotes maliciosos, pois estas tarefas são quase impossíveis de serem detectadas automaticamente. Mantenedores, no entanto, são completamente responsáveis pelos pacotes que adicionam na Debian, e todos os pacotes são primeiramente assinados por um desenvolvedor autorizado. O desenvolvedor tem a responsabilidade de analisar a segurança de todos os pacotes que ele mantém.

11.3.16 Quanto tempo a Debian levará para resolver a vulnerabilidade XXXX?

O time de segurança da Debian trabalha rapidamente para enviar avisos e produzir pacotes corrigidos para o repositório estável assim que uma vulnerabilidade é descoberta. Um relatório públicado na lista de discussão debian-security (http://lists.debian.org/debian-security/2001/debian-security-200112/msg00257.html) mostrou que no ano de 2001, houve uma média de 35 dias para corrigir problemas relacionados a segurança. No entanto, 50% dos problemas foram solucionados em um intervalo de 10 dias, e 15% dos problemas foram corrigidos no *mesmo dia* quando o aviso foi lançado.

No entanto, quando perguntam esta questão as pessoas tendem a se esquecer que:

- Os DSAs não são enviados até que:
  - os pacotes estejam disponíveis para *todas* as arquiteturas suportadas pela Debian (o que leva muito tempo para pacotes que são partes do núcleo do sistema, especialmente considerando o número de arquiteturas suportadas pelo lançamento estável).
  - novos pacotes são constantemente testados para ter certeza que nenhuma nova falha foi introduzida
- Os pacotes devem ser disponibilizados antes do DSA ser enviado (na queue incoming ou nos mirrors).
- O Debian é um projeto baseado em trabalho voluntário.
- A Debian é licenciada com uma cláusula “sem garantias”.

Se quiser uma análise mais precisa do tempo que o time de segurança leva para trabalhar em vulnerabilidades, você deverá considerar que os novos DSAs (veja 'Debian Security Advisories' on page 114) publicados no website de segurança (http://security.debian.org), e os metadados usado para gerá-los, incluem links para bancos de dados de vulnerabilidades. Você poderá baixar os fontes do servidor web (a partir do CVS (http://cvs.debian.org)) ou usar as páginas HTML para determinar o tempo que a Debian levou para corrigir a vulnerabilidade e co-relacionar estes dados com bancos de dados públicos.

Apêndice A

Passo-a-passo do processo de fortalecimento

Abaixo está uma pós-instalação, um procedimento passo-a-passo para tornar no sistema Debian 2.2 GNU/Linux mais seguro. Esse procedimento é uma alternativa para tornar os serviços de redes mais seguros. Será mostrado o processo completo do que deve ser feito durante a configuração. Também, veja 'Checklist de configuração' on page 187.

- Instale o sistema, levando em conta as informações sobre o particionamento que foi citada anteriormente neste documento. Depois da instalação básica, vá à instalação personalizada. Não selecione os pacotes de tarefa. Selecione senhas no formato shadow.
- Usando `dselect`, exclua todos os pacotes desnecessários, exceto os selecionados, antes de proceder com o [I]nstall. Mantenha um número reduzido de pacotes para o sistema.
- Atualize todos os softwares para a última versão disponível dos pacotes em security.debian.org como explicado anteriormente em 'Executar uma atualização de segurança' on page 40.
- Implementar as sugestões apresentadas neste manual com relação às cotas de usuários, definições de login e `lilo`
- Fazer uma lista de serviços que estão rodando no seu sistema. Tente:

```
$ ps -aux
$ netstat -pn -l -A inet
# /usr/sbin/lsof -i | grep LISTEN
```

  Você precisará instalar o `lsof-2.2` para o terceiro comando acima funcionar (execute como super-usuário). Você deve estar ciente de que o `lsof` pode traduzir a palavra LISTEN para suas configurações de localização.

- Para excluir serviços desnecessários, primeiro determine qual pacote fornece o serviço e como ele é inicializado. Isto pode ser feito verificando os programas que escutam no soquete. O shell script abaixo, que utiliza os programas `lsof` e `dpkg`, faz isso:

```
#!/bin/sh
# FIXME: this is quick and dirty; replace with a more robust script s
for i in `sudo lsof -i | grep LISTEN | cut -d " " -f 1 |sort -u` ; do
 pack=`dpkg -S $i |grep bin |cut -f 1 -d : | uniq`
 echo "Service $i is installed by $pack";
 init=`dpkg -L $pack |grep init.d/ `
 if [ ! -z "$init" ]; then
  echo "and is run by $init"
 fi
done
```

- Se você encontrar algum serviço desnecessário, exclua o pacote associado (com `dpkg -purge`), ou desabilite a inicialização automática durante a fase de boot usando o comando `update-rc.d` (veja 'Desabilitando daemons de serviço' on page 33).

- Para os serviços inetd (iniciados pelo superdaemon), verifique quais serviços estão ativados em `/etc/inetd.conf` através de:

```
$ grep -v "^#" /etc/inetd.conf | sort -u
```

  Então desative estes serviços desnecessários comentando a linha referente em `/etc/inetd.conf`, excluindo o pacote ou utilizando o comando `update-inetd`.

- Se você utiliza serviços wrapped (aqueles que utilizam `/usr/sbin/tcpd`), verifique se os arquivos `/etc/hosts.allow` e `/etc/hosts.deny` são configurados de acordo com sua política de serviço.

- Se o servidor usa mais que uma interface externa, dependendo do seu serviço, você pode limitar o serviço para escutar em uma interface específica. Por exemplo, se você quiser somente acesso interno para o FTP, você deve configurar o daemon FTP para escutar somente na sua interface de gerência, não em todas interfaces (i.e, 0.0.0.0:21).

- Reinicie o computador, ou troque o modo de single user para multiuser usando os comandos:

```
$ init 1
(....)
$ init 2
```

- Então verifique agora os serviços que estão disponíveis, e se necessário, repita os passos acima.

- Agora instale os serviços necessários, se não tiver feito isso ainda, e os configure corretamente.
- Use o comando shell abaixo para determinar com que usuário cada serviço disponível está sendo executado:

```
$ for i in `/usr/sbin/lsof -i |grep LISTEN |cut -d " " -f 1 |sort -u`
> do user=`ps -ef |grep $i |grep -v grep |cut -f 1 -d " "` ; \
> echo "Service $i is running as user $user"; done
```

  Considere alterar esses serviços para um usuário/grupo específico e talvez até enjaulá-los (`chroot`'ing) para aumentar nível de segurança. Você pode fazer isto alterando os scripts de inicialização em `/etc/init.d`. A maioria dos serviços no Debian usa o `start-stop-daemon` com as opções (`--change-uid` e `--chroot`) para fazer isso. Uma observação com relação ao enjaulamento (`chroot`'ing) dos serviços: você precisa colocar todos os arquivos instalados pelo pacote (use dpkg -L) que fornece o serviço, assim como qualquer pacote dependente, na jaula `chroot`. Informações sobre a configuração de um ambiente `chroot` para o programa `ssh` podem ser encontrada em 'Ambiente `chroot` para `SSH`' on page 207.
- Repita os passos acima para certificar que somente os serviços desejados estejam rodando e esteja sendo usada a combinação de usuário/grupo correta.
- Teste os serviços instalados para ver se estão funcionando corretamente.
- Verifique o sistema usando um vulnerability assessment scanner (tipo o `nessus`), para determinar as vulnerabilidades no sistema (i.e., mal-configuração, serviços antigos e desnecessários).
- Instale ferramentas de detecção de intrusão de rede e host como `snort` e `logsentry`.
- Repita o passo de varredura da rede e verifique se os sistemas de detecção de intrusão estão funcionando corretamente.

Para paranóia real, também considere o seguinte:

- Adicione as capacidades de firewall do sistema, conexões de entrada só devem ser feitas para os serviços oferecidos e limite as conexões de saída somente para aqueles que são autorizados.
- Verifique novamente a instalação com uma nova vulnerability assessment usando um varredor de rede.
- Usando um varredor de rede, verifique as conexões de saídas do sistema para um host remoto e certifique-se que as conexões indesejadas sejam estabelecida.

FIXME: este procedimento engloba o fortalecimento de serviços, mas não o fortalecimento a nível de usuário, incluindo informações sobre verificação de permissões de usuários, arquivos SETUID e congelamento de alterações no sistema utilizando o sistema de arquivo ext2.

Apêndice B

Checklist de configuração

Este apêndice retrata resumidamente os pontos de outras seções neste manual em um checklist no formato. A idéia é disponibilizar um sumário para a pessoa que já leu o manual buscar uma informação rapidamente. Existem outros checklists bons disponíveis, incluindo o Securing Linux Step by Step (http://seifried.org/security/os/linux/20020324-securing-linux-step-by-step.html) de Kurt Seifried e CERT's Unix Security Checklist (http://www.cert.org/tech_tips/usc20_full.html).

FIXME: Isso é baseado na versão 1.4 do manual e talvez precise de atualização.

- Limite o acesso físico e as capacidade de inicialização
  - Ative a senha de BIOS
  - Desative a inicialização por floppy/cdrom/...
  - Configure uma senha para o LILO ou GRUB (`/etc/lilo.conf` ou `/boot/grub/menu.lst`, respectivamente); verifique se o arquivo de configuração do LILO ou GRUB está protegido contra gravação.
  - Não permita a inicialização MBR pelo disquete sobrescrevendo a MBR (talvez não?)
- Particionamento
  - Separe os dados de escrita do usuário, dados que não são do sistema, e dados que são trocados rapidamente em tempo de execução para suas próprias partições
  - Configure as opções de mount `nosuid`, `noexec`, `nodev` em `/etc/fstab` na partições ext2 como `/tmp`.
- Higiene de senhas e segurança no login
  - Configure uma senha segura para o super-usuário
  - Ative o MD5 e o shadow de senha
  - Instale e use o PAM

    * Adicione suporte MD5 para o PAM e tenha certeza que (falando de forma generalizada) as entradas nos arquivos em `/etc/pam.d/` que garantem acesso à máquina tenham o segundo campo configurado como `requisite` ou `required`.
    * Modifique o `/etc/pam.d/login` para permite somente logins locais para o super-usuário.
    * Também marque tty:s autorizado em `/etc/security/access.conf` e geralmente configure este arquivo para limitar ao máximo possível o login do super-usuário.
    * Adicione o módulo pam_limits.so se você deseja configurar os limites por usuários
    * Modifique `/etc/pam.d/passwd`: configure o tamanho mínimo para as senhas (6 caracteres talvez) e ative o MD5
    * Adicione o grupo wheel para `/etc/group` se desejar; adicione a entrada pam_wheel.so group=wheel para `/etc/pam.d/su`
    * Para controles customizados por usuários, utilize o módulo pam_listfile.so
    * Tenha um arquivo `/etc/pam.d/other` e o configure com um grau de segurança reforçado
  - Configure limites em `/etc/security/limits.conf` (note que `/etc/limits` não é usado se você já estiver usando o PAM)
  - Aumente a segurança em `/etc/login.defs`; também, se você ativar o MD5 e/ou PAM, tenha certeza de fazer também as alterações correspondentes aqui, também
  - Desative o acesso ftp ao super-usuário em `/etc/ftpusers`
  - Desative login de rede ao super-usuário; use o `su(1)` ou `sudo(1)`. (considere instalar o `sudo`)
  - Usar o PAM para reforçar barreiras adicionais aos logins?

- Outras questões de segurança local

  - Modificações no kernel (veja 'Configurando características de rede do kernel' on page 72)
  - Patches no Kernel (veja 'Adicionando patches no kernel' on page 64)
  - Tighten up log file permissions (`/var/log/{last,fail}log`, Apache logs)
  - Certifique-se que a verificação SETUID está ativada em `/etc/checksecurity.conf`
  - Considere configurar alguns arquivos de logs como somente append e os arquivo de configuração imutáveis, usando o comando chattr (somente para arquivos ext2)
  - Configurar a integridade de arquivo (veja 'Verificando a integridade do sistema de arquivos' on page 71). Instale `debsums`
  - Efetuar o log de tudo em uma impressora local?
  - Gravar suas configurações em um CD inicializável e boof off?
  - Desativar os módulos do kernel?

- Limitar acesso a rede
  - Instale e configure `ssh` (sugiro PermitRootLogin No em `/etc/ssh/sshd_config`, PermitEmptyPasswords No; note outras sugestões também no texto)
  - Considere desativar ou excluir `in.telnetd`
  - Geralmente, desative serviços desnecessários em `/etc/inetd.conf` usando o comando `update-inetd -disable` (ou desative `inetd` completamente, ou use o um substituto como `xinetd` ou `rlinetd`)
  - Desative outros serviços de rede desnecessários; mail, ftp, DNS, WWW etc não devem estar sendo executados se você não precisa deles e monitore-os regularmente.
  - Para aqueles serviços que você precisa, não use os programas mais comuns, procure por versões mais seguras distribuídas com o Debian (ou de outras fontes). Seja lá o que você for parar de executar, tenha certeza que você entende os riscos.
  - Configure jaula `chroot` para usuários externos e daemons.
  - Configure firewall e tcpwrappers (i.e. `hosts_access(5)`); note o truque para `/etc/hosts.deny` no texto.
  - Se você executa o ftp, configure seu servidor ftpd sempre para executar enjaulado para o diretório home dos usuários
  - Se você executa X, desative a autenticação xhost e use-o com `ssh`; melhor ainda, se puder desative o X (adicione -nolisten tcp para a linha de comando do X e desligue o XDMCP no `/etc/X11/xdm/xdm-config` configurando requestPort para 0)
  - Desative acesso externo para as impressoras
  - Use tunelamento para qualquer sessão IMAP ou POP através do SSL ou `ssh`; instale stuneel se você quer fornecer este serviços para usuários de mail externos
  - Configure um host de log e configure as outras máquinas para enviar logs para esse host (`/etc/syslog.conf`)
  - Torne seguro o BIND, Sendmail, e outros daemons complexos (execute-os com uma jaula`chroot`; execute como um pseudo-usuário não root)
  - Instale o snort ou uma ferramenta similar para log.
  - Faça sem NIS ou RPC se puder (desative portmap).
- Políticas de segurança
  - Eduque os usuários sobre os porquês e comos de suas políticas. Quando você proíbe algo que está disponível regularmente em outros sistemas, forneça uma documentação que explique como obter resultados similares através de outros meios mais seguros.
  - Proíba o uso de protocolos que utilizam senhas em texto plano (`telnet`, `rsh` e similares; ftp, imap, http, ...).
  - Proíba programas que usam SVGAlib.
  - Use cotas de disco.

- Mantenha-se informado sobre questões relacionadas à segurança
  - Inscreva-se em listas de discussão sobre segurança
  - Configure `apt` para atualização de segurança – adicione no arquivo `/etc/apt/sources.list` uma entrada (ou entradas) para http://security.debian.org/debian-security
  - Também lembre-se de executar periodicamente os comandos `apt-get update ; apt-get upgrade` (talvez instalar como um job no `cron`?) como explicado em 'Executar uma atualização de segurança' on page 40.

Apêndice C

Configurando um IDS stand-alone

Você pode facilmente configurar um sistema Debian dedicado como um IDS stand-alone utilizando o `snort`.

Algumas linhas gerais:

- Instale um sistema Debian base e não selecione nenhum pacote adicional.
- Faça o download e manualmente (com dpkg) instale os pacotes necessários (veja a lista de pacotes instalados abaixo).
- Baixe e instale o ACID (Analysis Console for Intrusion Databases).

ACID está atualmente empacotado para o Debian como `acidlab`. Ele fornece uma interface WWW gráfica para o snort. Ele também pode ser baixado de `http://www.cert.org/kb/acid/`, `http://acidlab.sourceforge.net` ou `http://www.andrew.cmu.edu/~rdanyliw/snort/`. Você também pode querer ler o Snort Statistics HOWTO (`http://www.tldp.org/HOWTO/Snort-Statistics-HOWTO/index.html`).

Este sistema deve ser configurado com pelo menos duas interfaces de rede; uma interface conectada ao gerenciamento da LAN (para acessar os resultados e suporte do sistema), e outra interface sem nenhum endereço IP anexada ao segmento de rede a ser analisado.

O arquivo padrão `/etc/network/interfaces` do Debian utilizado normalmente para configurar placas de redes não pode ser usado, já que os programas `ifup` e `ifdown` esperam um endereço IP. Em vez disso, simplesmente use `ifconfig eth0 up`.

Além da instalação ordinária, `acidlab` também depende dos pacotes `php4` e `apache` entre outros. Baixe os seguintes pacotes (Note: as versões devem variar dependendo da distribuição do Debian que você esteja usando, esta lista é do Debian *woody* Setembro de 2001):

```
ACID-0.9.5b9.tar.gz
adduser_3.39_all.deb
apache-common_1.3.20-1_i386.deb
```

```
apache_1.3.20-1_i386.deb
debconf_0.9.77_all.deb
dialog_0.9a-20010527-1_i386.deb
fileutils_4.1-2_i386.deb
klogd_1.4.1-2_i386.deb
libbz2-1.0_1.0.1-10_i386.deb
libc6_2.2.3-6_i386.deb
libdb2_2.7.7-8_i386.deb
libdbd-mysql-perl_1.2216-2_i386.deb
libdbi-perl_1.18-1_i386.deb
libexpat1_1.95.1-5_i386.deb
libgdbmg1_1.7.3-27_i386.deb
libmm11_1.1.3-4_i386.deb
libmysqlclient10_3.23.39-3_i386.deb
libncurses5_5.2.20010318-2_i386.deb
libpcap0_0.6.2-1_i386.deb
libpcre3_3.4-1_i386.deb
libreadline4_4.2-3_i386.deb
libstdc++2.10-glibc2.2_2.95.4-0.010703_i386.deb
logrotate_3.5.4-2_i386.deb
mime-support_3.11-1_all.deb
mysql-client_3.23.39-3_i386.deb
mysql-common_3.23.39-3.1_all.deb
mysql-server_3.23.39-3_i386.deb
perl-base_5.6.1-5_i386.deb
perl-modules_5.6.1-5_all.deb
perl_5.6.1-5_i386.deb
php4-mysql_4.0.6-4_i386.deb
php4_4.0.6-1_i386.deb
php4_4.0.6-4_i386.deb
snort_1.7-9_i386.deb
sysklogd_1.4.1-2_i386.deb
zlib1g_1.1.3-15_i386.deb
```

Pacotes instalados (dpkg -l):

```
ii  adduser        3.39
ii  ae             962-26
ii  apache         1.3.20-1
ii  apache-common  1.3.20-1
ii  apt            0.3.19
ii  base-config    0.33.2
ii  base-files     2.2.0
ii  base-passwd    3.1.10
ii  bash           2.03-6
```

```
ii  bsdutils       2.10f-5.1
ii  console-data   1999.08.29-11.
ii  console-tools  0.2.3-10.3
ii  console-tools- 0.2.3-10.3
ii  cron           3.0pl1-57.2
ii  debconf        0.9.77
ii  debianutils    1.13.3
ii  dialog         0.9a-20010527-
ii  diff           2.7-21
ii  dpkg           1.6.15
ii  e2fsprogs      1.18-3.0
ii  elvis-tiny     1.4-11
ii  fbset          2.1-6
ii  fdflush        1.0.1-5
ii  fdutils        5.3-3
ii  fileutils      4.1-2
ii  findutils      4.1-40
ii  ftp            0.10-3.1
ii  gettext-base   0.10.35-13
ii  grep           2.4.2-1
ii  gzip           1.2.4-33
ii  hostname       2.07
ii  isapnptools    1.21-2
ii  joe            2.8-15.2
ii  klogd          1.4.1-2
ii  ldso           1.9.11-9
ii  libbz2-1.0     1.0.1-10
ii  libc6          2.2.3-6
ii  libdb2         2.7.7-8
ii  libdbd-mysql-p 1.2216-2
ii  libdbi-perl    1.18-1
ii  libexpat1      1.95.1-5
ii  libgdbmg1      1.7.3-27
ii  libmm11        1.1.3-4
ii  libmysqlclient 3.23.39-3
ii  libncurses5    5.2.20010318-2
ii  libnewt0       0.50-7
ii  libpam-modules 0.72-9
ii  libpam-runtime 0.72-9
ii  libpam0g       0.72-9
ii  libpcap0       0.6.2-1
ii  libpcre3       3.4-1
ii  libpopt0       1.4-1.1
ii  libreadline4   4.2-3
ii  libssl09       0.9.4-5
ii  libstdc++2.10  2.95.2-13
```

```
ii  libstdc++2.10- 2.95.4-0.01070
ii  libwrap0       7.6-4
ii  lilo           21.4.3-2
ii  locales        2.1.3-18
ii  login          19990827-20
ii  makedev        2.3.1-46.2
ii  mawk           1.3.3-5
ii  mbr            1.1.2-1
ii  mime-support   3.11-1
ii  modutils       2.3.11-13.1
ii  mount          2.10f-5.1
ii  mysql-client   3.23.39-3
ii  mysql-common   3.23.39-3.1
ii  mysql-server   3.23.39-3
ii  ncurses-base   5.0-6.0potato1
ii  ncurses-bin    5.0-6.0potato1
ii  netbase        3.18-4
ii  passwd         19990827-20
ii  pciutils       2.1.2-2
ii  perl           5.6.1-5
ii  perl-base      5.6.1-5
ii  perl-modules   5.6.1-5
ii  php4           4.0.6-4
ii  php4-mysql     4.0.6-4
ii  ppp            2.3.11-1.4
ii  pppconfig      2.0.5
ii  procps         2.0.6-5
ii  psmisc         19-2
ii  pump           0.7.3-2
ii  sed            3.02-5
ii  setserial      2.17-16
ii  shellutils     2.0-7
ii  slang1         1.3.9-1
ii  snort          1.7-9
ii  ssh            1.2.3-9.3
ii  sysklogd       1.4.1-2
ii  syslinux       1.48-2
ii  sysvinit       2.78-4
ii  tar            1.13.17-2
ii  tasksel        1.0-10
ii  tcpd           7.6-4
ii  telnet         0.16-4potato.1
ii  textutils      2.0-2
ii  update         2.11-1
ii  util-linux     2.10f-5.1
ii  zlib1g         1.1.3-15
```

Apêndice D

Configurando uma ponte firewall

Esta informação foi contribuição de Francois Bayart para ajudar os usuário a configurar um Linux como ponte/firewall com o kernel 2.4.x e `iptables`. Patches do kernel não são mais necessários, uma vez que o código passou a fazer parte do kernel do Linux.

Para configurar o kernel com o suporte necessário, execute `make menuconfig` ou `make xconfig`. Na seção *Networking options*, ative as seguintes opções:

```
[*] Network packet filtering (replaces ipchains)
[ ]   Network packet filtering debugging (NEW)
<*> 802.1d Ethernet Bridging
[*]   netfilter (firewalling) support (NEW)
```

Cuidado: você deve desativar isso se você quiser aplicar algumas regras de firewall ou o `iptables` não funcionará:

```
[ ]   Network packet filtering debugging (NEW)
```

Próximo passo, adicione as opções corretas na seção *IP: Netfilter Configuration*. Então, compile e instale o kernel. Se você quiser fazer isso no *jeito do Debian*, instale o `kernel-package` e execute `make-kpkg` para criar um pacote Debian customizado do kernel que possa ser instalado no servidor usando o dpkg. Uma vez que o novo kernel é compilado e instalado, instale o pacote `bridge-utils`.

Quando estes passos forem feitos, você pode completar a configuração de sua ponte. A próxima seção apresenta duas possíveis configurações para a ponte, cada uma com um mapa de rede hipotético e os comandos necessários.

D.1 Uma ponte fornecendo capacidades de NAT e firewall

A primeira configuração usa a ponte como um firewall com tradução de endereços de rede (NAT) que protege o servidor e os clientes da rede interna. Um diagrama da configuração da rede é mostrado abaixo:

```
Internet ---- router ( 62.3.3.25 ) ---- bridge (62.3.3.26 gw 62.3.3.25 / 192.
                                          |
                                          |
                                          |---- WWW Server (62.3.3.27 gw 62.3
                                          |
                                          |
                                         LAN --- Zipowz (192.168.0.2 gw 192.1
```

Os seguintes comandos mostram como esta ponte pode ser configurada.

```
# Create the interface br0
/usr/sbin/brctl addbr br0

# Add the Ethernet interface to use with the bridge
/usr/sbin/brctl addif br0 eth0
/usr/sbin/brctl addif br0 eth1

# Start up the Ethernet interface
/sbin/ifconfig eth0 0.0.0.0
/sbin/ifconfig eth1 0.0.0.0

# Configure the bridge ethernet
# The bridge will be correct and invisible ( transparent firewall ).
# It's hidden in a traceroute and you keep your real gateway on the
# other computers. Now if you want you can config a gateway on your
# bridge and choose it as your new gateway for the other computers.

/sbin/ifconfig br0 62.3.3.26 netmask 255.255.255.248 broadcast 62.3.3.32

# I have added this internal IP to create my NAT
ip addr add 192.168.0.1/24 dev br0
/sbin/route add default gw 62.3.3.25
```

D.2 Uma ponte fornecendo capacidades de firewall

Uma segunda possível configuração é um sistema que funciona como um firewall transparente para a LAN com um espaço de endereços IP públicos.

```
Internet ---- router (62.3.3.25) ---- bridge (62.3.3.26)
                                          |
                                          |
                                          |---- WWW Server (62.3.3.28 gw 62.3.3
                                          |
```

```
                                                |
                                                |---- Mail Server (62.3.3.27 gw 62.3.
```

Os seguintes comando mostram como esta ponte pode ser configurada.

```
# Create the interface br0
/usr/sbin/brctl addbr br0

# Add the Ethernet interface to use with the bridge
/usr/sbin/brctl addif br0 eth0
/usr/sbin/brctl addif br0 eth1

# Start up the Ethernet interface
/sbin/ifconfig eth0 0.0.0.0
/sbin/ifconfig eth1 0.0.0.0

# Configure the bridge Ethernet
# The bridge will be correct and invisible ( transparent firewall ).
# It's hidden in a traceroute and you keep your real gateway on the
# other computers. Now if you want you can config a gateway on your
# bridge and choose it as your new gateway for the other computers.

/sbin/ifconfig br0 62.3.3.26 netmask 255.255.255.248 broadcast 62.3.3.32
```

Se você seguir as rotas para o Linux Mail Server, não enxergará a ponte. Se você quiser acessar a ponte com o `ssh`, você deve ter um gateway ou acessar um outro servidor, como o "Mail Server", e então conectar à ponte através de uma placa de rede interna.

D.3 Regras básicas do IPtables

As regras básicas a seguir podem ser usadas em qualquer uma das duas configurações mostradas acima.

```
iptables -F FORWARD
iptables -P FORWARD DROP
iptables -A FORWARD -s 0.0.0.0/0.0.0.0 -d 0.0.0.0/0.0.0.0 -m state --state
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT

# Some funny rules but not in a classic Iptables sorry ...
# Limit ICMP
# iptables -A FORWARD -p icmp -m limit --limit 4/s -j ACCEPT
# Match string, a good simple method to block some VIRUS very quickly
# iptables -I FORWARD -j DROP -p tcp -s 0.0.0.0/0 -m string --string "cmd.e
```

```
# Block all MySQL connection just to be sure
iptables -A FORWARD -p tcp -s 0/0 -d 62.3.3.0/24 --dport 3306 -j DROP

# Linux Mail Server Rules

# Allow FTP-DATA ( 20 ) , FTP ( 21 ) , SSH ( 22 )
iptables -A FORWARD -p tcp -s 0.0.0.0/0 -d 62.3.3.27/32 --dport 20:22 -j AC

# Allow the Mail Server to connect to the outside
# Note: This is *not* needed for the previous connections
# (remember: stateful filtering) and could be removed.
iptables -A FORWARD -p tcp -s 62.3.3.27/32 -d 0/0 -j ACCEPT

# WWW Server Rules

# Allow HTTP ( 80 ) connections with the WWW server
iptables -A FORWARD -p tcp -s 0.0.0.0/0 -d 62.3.3.28/32 --dport 80 -j ACCEP

# Allow HTTPS ( 443 ) connections with the WWW server
iptables -A FORWARD -p tcp -s 0.0.0.0/0 -d 62.3.3.28/32 --dport 443 -j ACCE

# Allow the WWW server to go out
# Note: This is *not* needed for the previous connections
# (remember: stateful filtering) and could be removed.
iptables -A FORWARD -p tcp -s 62.3.3.28/32 -d 0/0 -j ACCEPT
```

Apêndice E

Exemplo de script para alterar a instalação padrão do Bind.

Este script automatiza o procedimento para alterar a instalação padrão do servidor de nome `bind` de forma que ele *não* execute como superusuário. Ele irá criar usuário e grupos que serão usados para o servidor de nome. Utilize-o com bastante cuidado já que o script não foi testado exaustivamente.

```
#!/bin/sh
# Change the default Debian bind configuration to have it run
# with a non-root user and group.
#
# WARN: This script has not been tested thoroughly, please
# verify the changes made to the INITD script

# (c) 2002 Javier Fernandez-Sanguino Peña
#
#    This program is free software; you can redistribute it and/or modify
#    it under the terms of the GNU General Public License as published by
#    the Free Software Foundation; either version 1, or (at your option)
#    any later version.
#
#    This program is distributed in the hope that it will be useful,
#    but WITHOUT ANY WARRANTY; without even the implied warranty of
#    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
#    GNU General Public License for more details.
#
#     Please see the file 'COPYING' for the complete copyright notice.
#

restore() {
# Just in case, restore the system if the changes fail
```

```
 echo "WARN: Restoring to the previous setup since I'm unable to properly c
 echo "WARN: Please check the $INITDERR script."
 mv $INITD $INITDERR
 cp $INITDBAK $INITD
}


USER=named
GROUP=named
INITD=/etc/init.d/bind
INITDBAK=$INITD.preuserchange
INITDERR=$INITD.changeerror
START="start-stop-daemon --start --quiet --exec /usr/sbin/named -- -g $GROU
AWKS="awk ' /start-stop-daemon --start/ { print \"$START\"; noprint = 1; };

[ `id -u` -ne 0 ] && {
 echo "This program must be run by the root user"
 exit 1
}

RUNUSER=`ps -eo user,fname |grep named |cut -f 1 -d " "`

if [ "$RUNUSER" = "$USER" ]
then
 echo "WARN: The name server running daemon is already running as $USER"
 echo "ERR:  This script will not many any changes to your setup."
 exit 1
fi
if [ ! -f $INITD ]
then
 echo "ERR:  This system does not have $INITD (which this script tries to c
 RUNNING=`ps -eo fname |grep named`
  [ -z "$RUNNING" ] && \
     echo "ERR:  In fact the name server daemon is not even running (is it
  echo "ERR:  No changes will be made to your system"
 exit 1
fi

# Check if named group exists
if [ -z "`grep $GROUP /etc/group`" ]
then
 echo "Creating group $GROUP:"
 addgroup $GROUP
else
 echo "WARN: Group $GROUP already exists. Will not create it"
fi
```

```
# Same for the user
if [ -z "`grep $USER /etc/passwd`" ]
then
 echo "Creating user $USER:"
 adduser --system --home /home/$USER \
 --no-create-home --ingroup $GROUP \
 --disabled-password --disabled-login $USER
else
 echo "WARN: The user $USER already exists. Will not create it"
fi

# Change the init.d script

# First make a backup (check that there is not already
# one there first)
if [ ! -f $INITDBAK ]
then
 cp $INITD $INITDBAK
fi

# Then use it to change it
cat $INITDBAK |
eval $AWKS > $INITD

echo "WARN: The script $INITD has been changed, trying to test the changes.
echo "Restarting the named daemon (check for errors here)."

$INITD restart
if [ $? -ne 0 ]
then
 echo "ERR:  Failed to restart the daemon."
 restore
 exit 1
fi

RUNNING=`ps -eo fname |grep named`
if [ -z "$RUNNING" ]
then
 echo "ERR:  Named is not running, probably due to a problem with the chang
 restore
 exit 1
fi

# Check if it's running as expected
RUNUSER=`ps -eo user,fname |grep named |cut -f 1 -d " "`
```

```
if [ "$RUNUSER" = "$USER" ]
then
 echo "All has gone well, named seems to be running now as $USER."
else
 echo "ERR:  The script failed to automatically change the system."
 echo "ERR:  Named is currently running as $RUNUSER."
 restore
 exit 1
fi

exit 0
```

O script anterior, execute-o no `bind` customizado do Woody (Debian 3.0), irá produzir o arquivo initd abaixo depois de criar o usuário e grupo 'named':

```
#!/bin/sh

PATH=/sbin:/bin:/usr/sbin:/usr/bin

test -x /usr/sbin/named || exit 0

start () {
 echo -n "Starting domain name service: named"
 start-stop-daemon --start --quiet \
     --pidfile /var/run/named.pid --exec /usr/sbin/named
 echo "."
}

stop () {
 echo -n "Stopping domain name service: named"
 # --exec doesn't catch daemons running deleted instances of named,
 # as in an upgrade.  Fortunately, --pidfile is only going to hit
 # things from the pidfile.
 start-stop-daemon --stop --quiet  \
     --pidfile /var/run/named.pid --name named
 echo "."
}

case "$1" in
    start)
 start
    ;;

    stop)
 stop
```

```
    ;;

    restart|force-reload)
 stop
 sleep 2
 start
    ;;

    reload)
 /usr/sbin/ndc reload
    ;;

    *)
 echo "Usage: /etc/init.d/bind {start|stop|reload|restart|force-reload}" >&
 exit 1
    ;;
esac

exit 0
```

Apêndice F

Atualização de segurança protegida por um firewall

Depois de uma instalaçõa padrão, o sistema ainda poderá ter algumas vulnerabilidades de segurança. Ao menos que você baixe as atualizações para os pacotes vulneráveis em outro computador (ou você tenha espelhado security.debian.org para uso local), o sistema deverá ter acesso à Internet para os downloads.

Entretanto, na medida que você se conecta à Internet estará expondo seu sistema. Se um de seus serviços locais estiver vulnerável, poderá ser comprometido mesmo antes de finalizar as atualizações! Isso pode ser paranóico, mas as análises do Projeto Honeynet (http://www.honeynet.org) têm mostrado que sistemas podem ser comprometidos em menos de três dias, mesmo que o sistema não seja conhecido publicamento (i.e., não está publicado nos registros DNS).

Quando estiver fazendo uma atualização em um sistema não protegido por um mecanismo externo como firewall, é possível configurar seu firewall local para restringir conexões envolvendo somente as próprias atualizações de segurança. O exemplo abaixo mostra como configurar estas capacidades de firewall, que permitem somente conexões do security.debian.org, registrando todas as outras que são negadas.

FIXME: add IP address for security.debian.org (since otherwise you need DNS up to work) on /etc/hosts.

FIXME: test this setup to see if it works properly

FIXME: this will only work with HTTP URLs since ftp might need the ip_conntrack_ftp module, or use passive mode.

```
# iptables -F
# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
```

```
Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
# iptables -P INPUT DROP
# iptables -P FORWARD DROP
# iptables -P OUTPUT DROP
# iptables -A OUTPUT -d security.debian.org --dport 80 -j ACCEPT
# iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
# iptables -A INPUT -p icmp -j ACCEPT
# iptables -A INPUT -j LOG
# iptables -A OUTPUT -j LOG
# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0          state RELATED,E
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0
LOG        all  --  anywhere             anywhere           LOG level warni

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy DROP)
target     prot opt source               destination
ACCEPT     80   --  anywhere             security.debian.org
LOG        all  --  anywhere             anywhere           LOG level warni
```

Apêndice G

Ambiente chroot para SSH

Criar um ambiente restrito para SSH é um trabalho duro, devido às suas dependências e pelo fato que, diferente dos outros serviços, o SSH fornece shell remoto aos usuários. Então, você também deve considerar as aplicações que serão permitidas aos usuários neste ambiente. Se você criar esta estrutura de arquivos em, por exemplo /var/chroot/ssh, poderia inicializar o servidor ssh enjaulado com o comando:

```
# chroot /var/chroot/ssh /sbin/sshd -f /etc/sshd_config
```

G.1 Configurando automaticamente o ambiente (a maneira fácil)

Você pode facilmente criar um ambiente restrito com o pacote makejail, já que ele automaticamente segue as trilhas do servidor daemon (com strace) e faz com que ele execute em um ambiente restrito.

A vantagem de programas que automaticamente geram um ambiente chroot é que eles são capazes de copiar qualquer pacote para o ambiente chroot (mesmo seguindo as dependências do pacote e certificar que foi completada). Então, fornecer as aplicações dos usuários é bem mais fácil.

Para configurar o ambiente usando os exemplos fornecidos pelo makejail, use o comando:

```
# makejail /usr/share/doc/makejail/examples/sshd.py
```

Leia o arquivo exemplo para ver que outras mudanças devem ser feitas para o ambiente. Algumas dessas mudanças, como copiar os diretórios home do usuário, não podem ser feitas automaticamente. Também limite a exposição de informações sensíveis, copiando os dados de um certo número de usuários dos arquivos /etc/shadow ou /etc/group.

O seguinte exemplo de ambiente tem sido (levemente) testado, foi construído com o arquivo de configuração fornecido no pacote e inclue o pacote fileutils:

```
.
|-- bin
|   |-- ash
|   |-- bash
|   |-- chgrp
|   |-- chmod
|   |-- chown
|   |-- cp
|   |-- csh -> /etc/alternatives/csh
|   |-- dd
|   |-- df
|   |-- dir
|   |-- fdflush
|   |-- ksh
|   |-- ln
|   |-- ls
|   |-- mkdir
|   |-- mknod
|   |-- mv
|   |-- rbash -> bash
|   |-- rm
|   |-- rmdir
|   |-- sh -> bash
|   |-- sync
|   |-- tcsh
|   |-- touch
|   |-- vdir
|   |-- zsh -> /etc/alternatives/zsh
|   `-- zsh4
|-- dev
|   |-- null
|   |-- ptmx
|   |-- pts
|   |-- ptya0
(...)
|   |-- tty
|   |-- tty0
(...)
|   `-- urandom
|-- etc
|   |-- alternatives
|   |   |-- csh -> /bin/tcsh
|   |   `-- zsh -> /bin/zsh4
|   |-- environment
|   |-- hosts
|   |-- hosts.allow
```

```
|   |-- hosts.deny
|   |-- ld.so.conf
|   |-- localtime -> /usr/share/zoneinfo/Europe/Madrid
|   |-- motd
|   |-- nsswitch.conf
|   |-- pam.conf
|   |-- pam.d
|   |   |-- other
|   |   `-- ssh
|   |-- passwd
|   |-- resolv.conf
|   |-- security
|   |   |-- access.conf
|   |   |-- chroot.conf
|   |   |-- group.conf
|   |   |-- limits.conf
|   |   |-- pam_env.conf
|   |   `-- time.conf
|   |-- shadow
|   |-- shells
|   `-- ssh
|       |-- moduli
|       |-- ssh_host_dsa_key
|       |-- ssh_host_dsa_key.pub
|       |-- ssh_host_rsa_key
|       |-- ssh_host_rsa_key.pub
|       `-- sshd_config
|-- home
|   `-- userX
|-- lib
|   |-- ld-2.2.5.so
|   |-- ld-linux.so.2 -> ld-2.2.5.so
|   |-- libc-2.2.5.so
|   |-- libc.so.6 -> libc-2.2.5.so
|   |-- libcap.so.1 -> libcap.so.1.10
|   |-- libcap.so.1.10
|   |-- libcrypt-2.2.5.so
|   |-- libcrypt.so.1 -> libcrypt-2.2.5.so
|   |-- libdl-2.2.5.so
|   |-- libdl.so.2 -> libdl-2.2.5.so
|   |-- libm-2.2.5.so
|   |-- libm.so.6 -> libm-2.2.5.so
|   |-- libncurses.so.5 -> libncurses.so.5.2
|   |-- libncurses.so.5.2
|   |-- libnsl-2.2.5.so
|   |-- libnsl.so.1 -> libnsl-2.2.5.so
```

```
|   |-- libnss_compat-2.2.5.so
|   |-- libnss_compat.so.2 -> libnss_compat-2.2.5.so
|   |-- libnss_db-2.2.so
|   |-- libnss_db.so.2 -> libnss_db-2.2.so
|   |-- libnss_dns-2.2.5.so
|   |-- libnss_dns.so.2 -> libnss_dns-2.2.5.so
|   |-- libnss_files-2.2.5.so
|   |-- libnss_files.so.2 -> libnss_files-2.2.5.so
|   |-- libnss_hesiod-2.2.5.so
|   |-- libnss_hesiod.so.2 -> libnss_hesiod-2.2.5.so
|   |-- libnss_nis-2.2.5.so
|   |-- libnss_nis.so.2 -> libnss_nis-2.2.5.so
|   |-- libnss_nisplus-2.2.5.so
|   |-- libnss_nisplus.so.2 -> libnss_nisplus-2.2.5.so
|   |-- libpam.so.0 -> libpam.so.0.72
|   |-- libpam.so.0.72
|   |-- libpthread-0.9.so
|   |-- libpthread.so.0 -> libpthread-0.9.so
|   |-- libresolv-2.2.5.so
|   |-- libresolv.so.2 -> libresolv-2.2.5.so
|   |-- librt-2.2.5.so
|   |-- librt.so.1 -> librt-2.2.5.so
|   |-- libutil-2.2.5.so
|   |-- libutil.so.1 -> libutil-2.2.5.so
|   |-- libwrap.so.0 -> libwrap.so.0.7.6
|   |-- libwrap.so.0.7.6
|   `-- security
|       |-- pam_access.so
|       |-- pam_chroot.so
|       |-- pam_deny.so
|       |-- pam_env.so
|       |-- pam_filter.so
|       |-- pam_ftp.so
|       |-- pam_group.so
|       |-- pam_issue.so
|       |-- pam_lastlog.so
|       |-- pam_limits.so
|       |-- pam_listfile.so
|       |-- pam_mail.so
|       |-- pam_mkhomedir.so
|       |-- pam_motd.so
|       |-- pam_nologin.so
|       |-- pam_permit.so
|       |-- pam_rhosts_auth.so
|       |-- pam_rootok.so
|       |-- pam_securetty.so
```

```
|       |-- pam_shells.so
|       |-- pam_stress.so
|       |-- pam_tally.so
|       |-- pam_time.so
|       |-- pam_unix.so
|       |-- pam_unix_acct.so -> pam_unix.so
|       |-- pam_unix_auth.so -> pam_unix.so
|       |-- pam_unix_passwd.so -> pam_unix.so
|       |-- pam_unix_session.so -> pam_unix.so
|       |-- pam_userdb.so
|       |-- pam_warn.so
|       `-- pam_wheel.so
|-- sbin
|   `-- start-stop-daemon
|-- usr
|   |-- bin
|   |   |-- dircolors
|   |   |-- du
|   |   |-- install
|   |   |-- link
|   |   |-- mkfifo
|   |   |-- shred
|   |   |-- touch -> /bin/touch
|   |   `-- unlink
|   |-- lib
|   |   |-- libcrypto.so.0.9.6
|   |   |-- libdb3.so.3 -> libdb3.so.3.0.2
|   |   |-- libdb3.so.3.0.2
|   |   |-- libz.so.1 -> libz.so.1.1.4
|   |   `-- libz.so.1.1.4
|   |-- sbin
|   |   `-- sshd
|   `-- share
|       |-- locale
|       |   `-- es
|       |       |-- LC_MESSAGES
|       |       |   |-- fileutils.mo
|       |       |   |-- libc.mo
|       |       |   `-- sh-utils.mo
|       |       `-- LC_TIME -> LC_MESSAGES
|       `-- zoneinfo
|           `-- Europe
|               `-- Madrid
`-- var
    `-- run
        |-- sshd
```

```
        `-- sshd.pid

27 directories, 733 files
```

G.2 Aplicando patch no SSH para ativar a funcionalidade do chroot

O sshd do Debian não permite restringir as operações do usuário através do servidor, já que falta uma função chroot que o programa comercial sshd2 inclue (usando 'ChrootGroups' ou 'ChrootUsers', veja sshd2_config(5)). Entretanto, existe um patch disponível para adicionar esta funcionalidade que pode ser baixado em Bug report 139047 (http://bugs.debian.org/139047) O patch pode ser incluído nos lançamentos futuros do pacote OpenSSH. Emmanuel Lacour tem os pacotes deb do ssh com este recurso em http://debian.home-dn.net/woody/ssh/. De qualquer forma é recomendável compilar o programa.

Uma descrição de todos os passos necessários podem ser encontrada em http://mail.incredimail.com/howto/openssh/ (apesar de ser direcionada para usuários RedHat 7.2, quase todos deles são aplicáveis para o Debian). Depois de aplicar o patch, modifique o arquivo /etc/passwd alterando o caminho do home dos usuários (com o token especial /./):

```
joeuser:x:1099:1099:Joe Random User:/home/joe/./:/bin/bash
```

Isto irá restringir *ambos* o acesso remoto ao shell, como também a cópia remota através do canal ssh.

Tenha certeza de ter todos os binários e bibliotecas necessárias dentro do caminho que está enjaulado para os usuários. Estes arquivos devem pertencer ao root para evitar tampering pelo usuário (como sair da jaula chroot'ed). Um exemplo possível inclue:

```
./bin:
total 660
drwxr-xr-x    2 root     root         4096 Mar 18 13:36 .
drwxr-xr-x    8 guest    guest        4096 Mar 15 16:53 ..
-r-xr-xr-x    1 root     root       531160 Feb  6 22:36 bash
-r-xr-xr-x    1 root     root        43916 Nov 29 13:19 ls
-r-xr-xr-x    1 root     root        16684 Nov 29 13:19 mkdir
-rwxr-xr-x    1 root     root        23960 Mar 18 13:36 more
-r-xr-xr-x    1 root     root         9916 Jul 26  2001 pwd
-r-xr-xr-x    1 root     root        24780 Nov 29 13:19 rm
lrwxrwxrwx    1 root     root            4 Mar 30 16:29 sh -> bash

./etc:
total 24
drwxr-xr-x    2 root     root         4096 Mar 15 16:13 .
```

```
drwxr-xr-x    8 guest    guest        4096 Mar 15 16:53 ..
-rw-r--r--    1 root     root           54 Mar 15 13:23 group
-rw-r--r--    1 root     root          428 Mar 15 15:56 hosts
-rw-r--r--    1 root     root           44 Mar 15 15:53 passwd
-rw-r--r--    1 root     root           52 Mar 15 13:23 shells

./lib:
total 1848
drwxr-xr-x    2 root     root         4096 Mar 18 13:37 .
drwxr-xr-x    8 guest    guest        4096 Mar 15 16:53 ..
-rwxr-xr-x    1 root     root        92511 Mar 15 12:49 ld-linux.so.2
-rwxr-xr-x    1 root     root      1170812 Mar 15 12:49 libc.so.6
-rw-r--r--    1 root     root        20900 Mar 15 13:01 libcrypt.so.1
-rw-r--r--    1 root     root         9436 Mar 15 12:49 libdl.so.2
-rw-r--r--    1 root     root       248132 Mar 15 12:48 libncurses.so.5
-rw-r--r--    1 root     root        71332 Mar 15 13:00 libnsl.so.1
-rw-r--r--    1 root     root        34144 Mar 15 16:10
libnss_files.so.2
-rw-r--r--    1 root     root        29420 Mar 15 12:57 libpam.so.0
-rw-r--r--    1 root     root       105498 Mar 15 12:51 libpthread.so.0
-rw-r--r--    1 root     root        25596 Mar 15 12:51 librt.so.1
-rw-r--r--    1 root     root         7760 Mar 15 12:59 libutil.so.1
-rw-r--r--    1 root     root        24328 Mar 15 12:57 libwrap.so.0

./usr:
total 16
drwxr-xr-x    4 root     root         4096 Mar 15 13:00 .
drwxr-xr-x    8 guest    guest        4096 Mar 15 16:53 ..
drwxr-xr-x    2 root     root         4096 Mar 15 15:55 bin
drwxr-xr-x    2 root     root         4096 Mar 15 15:37 lib

./usr/bin:
total 340
drwxr-xr-x    2 root     root         4096 Mar 15 15:55 .
drwxr-xr-x    4 root     root         4096 Mar 15 13:00 ..
-rwxr-xr-x    1 root     root        10332 Mar 15 15:55 env
-rwxr-xr-x    1 root     root        13052 Mar 15 13:13 id
-r-xr-xr-x    1 root     root        25432 Mar 15 12:40 scp
-rwxr-xr-x    1 root     root        43768 Mar 15 15:15 sftp
-r-sr-xr-x    1 root     root       218456 Mar 15 12:40 ssh
-rwxr-xr-x    1 root     root         9692 Mar 15 13:17 tty

./usr/lib:
total 852
drwxr-xr-x    2 root     root         4096 Mar 15 15:37 .
drwxr-xr-x    4 root     root         4096 Mar 15 13:00 ..
```

```
-rw-r--r--    1 root     root       771088 Mar 15 13:01
libcrypto.so.0.9.6
-rw-r--r--    1 root     root        54548 Mar 15 13:00 libz.so.1
-rwxr-xr-x    1 root     root        23096 Mar 15 15:37 sftp-server
```

G.3 Ambiente feito a mão (a maneira difícil)

É possível criar um ambiente, usando o método de tentativa e erro, seguindo a execução do servidor `sshd` e arquivos de log para determinar os arquivos necessários. O seguinte ambiente, contribuído por José Luis Ledesma, é uma listagem amostral do arquivos que estão no ambiente `chroot` para o `ssh`: [1]

```
.:
total 36
drwxr-xr-x 9 root root 4096 Jun 5 10:05 ./
drwxr-xr-x 11 root root 4096 Jun 3 13:43 ../
drwxr-xr-x 2 root root 4096 Jun 4 12:13 bin/
drwxr-xr-x 2 root root 4096 Jun 4 12:16 dev/
drwxr-xr-x 4 root root 4096 Jun 4 12:35 etc/
drwxr-xr-x 3 root root 4096 Jun 4 12:13 lib/
drwxr-xr-x 2 root root 4096 Jun 4 12:35 sbin/
drwxr-xr-x 2 root root 4096 Jun 4 12:32 tmp/
drwxr-xr-x 2 root root 4096 Jun 4 12:16 usr/
./bin:
total 8368
drwxr-xr-x 2 root root 4096 Jun 4 12:13 ./
drwxr-xr-x 9 root root 4096 Jun 5 10:05 ../
-rwxr-xr-x 1 root root 109855 Jun 3 13:45 a2p*
-rwxr-xr-x 1 root root 387764 Jun 3 13:45 bash*
-rwxr-xr-x 1 root root 36365 Jun 3 13:45 c2ph*
-rwxr-xr-x 1 root root 20629 Jun 3 13:45 dprofpp*
-rwxr-xr-x 1 root root 6956 Jun 3 13:46 env*
-rwxr-xr-x 1 root root 158116 Jun 3 13:45 fax2ps*
-rwxr-xr-x 1 root root 104008 Jun 3 13:45 faxalter*
-rwxr-xr-x 1 root root 89340 Jun 3 13:45 faxcover*
-rwxr-xr-x 1 root root 441584 Jun 3 13:45 faxmail*
-rwxr-xr-x 1 root root 96036 Jun 3 13:45 faxrm*
-rwxr-xr-x 1 root root 107000 Jun 3 13:45 faxstat*
-rwxr-xr-x 1 root root 77832 Jun 4 11:46 grep*
-rwxr-xr-x 1 root root 19597 Jun 3 13:45 h2ph*
-rwxr-xr-x 1 root root 46979 Jun 3 13:45 h2xs*
```

[1] Observe que não existem arquivos SETUID. Isso torna mais difícil para usuários remotos fugir o ambiente `chroot`. Entretanto, isso também previne que os usuários alterem suas senhas, já que o programa `passwd` não pode modificar os arquivos `/etc/passwd` ou `/etc/shadow`.

```
-rwxr-xr-x 1 root root 10420 Jun 3 13:46 id*
-rwxr-xr-x 1 root root 4528 Jun 3 13:46 ldd*
-rwxr-xr-x 1 root root 111386 Jun 4 11:46 less*
-r-xr-xr-x 1 root root 26168 Jun 3 13:45 login*
-rwxr-xr-x 1 root root 49164 Jun 3 13:45 ls*
-rwxr-xr-x 1 root root 11600 Jun 3 13:45 mkdir*
-rwxr-xr-x 1 root root 24780 Jun 3 13:45 more*
-rwxr-xr-x 1 root root 154980 Jun 3 13:45 pal2rgb*
-rwxr-xr-x 1 root root 27920 Jun 3 13:46 passwd*
-rwxr-xr-x 1 root root 4241 Jun 3 13:45 pl2pm*
-rwxr-xr-x 1 root root 2350 Jun 3 13:45 pod2html*
-rwxr-xr-x 1 root root 7875 Jun 3 13:45 pod2latex*
-rwxr-xr-x 1 root root 17587 Jun 3 13:45 pod2man*
-rwxr-xr-x 1 root root 6877 Jun 3 13:45 pod2text*
-rwxr-xr-x 1 root root 3300 Jun 3 13:45 pod2usage*
-rwxr-xr-x 1 root root 3341 Jun 3 13:45 podchecker*
-rwxr-xr-x 1 root root 2483 Jun 3 13:45 podselect*
-r-xr-xr-x 1 root root 82412 Jun 4 11:46 ps*
-rwxr-xr-x 1 root root 36365 Jun 3 13:45 pstruct*
-rwxr-xr-x 1 root root 7120 Jun 3 13:45 pwd*
-rwxr-xr-x 1 root root 179884 Jun 3 13:45 rgb2ycbcr*
-rwxr-xr-x 1 root root 20532 Jun 3 13:45 rm*
-rwxr-xr-x 1 root root 6720 Jun 4 10:15 rmdir*
-rwxr-xr-x 1 root root 14705 Jun 3 13:45 s2p*
-rwxr-xr-x 1 root root 28764 Jun 3 13:46 scp*
-rwxr-xr-x 1 root root 385000 Jun 3 13:45 sendfax*
-rwxr-xr-x 1 root root 67548 Jun 3 13:45 sendpage*
-rwxr-xr-x 1 root root 88632 Jun 3 13:46 sftp*
-rwxr-xr-x 1 root root 387764 Jun 3 13:45 sh*
-rws--x--x 1 root root 744500 Jun 3 13:46 slogin*
-rwxr-xr-x 1 root root 14523 Jun 3 13:46 splain*
-rws--x--x 1 root root 744500 Jun 3 13:46 ssh*
-rwxr-xr-x 1 root root 570960 Jun 3 13:46 ssh-add*
-rwxr-xr-x 1 root root 502952 Jun 3 13:46 ssh-agent*
-rwxr-xr-x 1 root root 575740 Jun 3 13:46 ssh-keygen*
-rwxr-xr-x 1 root root 383480 Jun 3 13:46 ssh-keyscan*
-rwxr-xr-x 1 root root 39 Jun 3 13:46 ssh_europa*
-rwxr-xr-x 1 root root 107252 Jun 4 10:14 strace*
-rwxr-xr-x 1 root root 8323 Jun 4 10:14 strace-graph*
-rwxr-xr-x 1 root root 158088 Jun 3 13:46 thumbnail*
-rwxr-xr-x 1 root root 6312 Jun 3 13:46 tty*
-rwxr-xr-x 1 root root 55904 Jun 4 11:46 useradd*
-rwxr-xr-x 1 root root 585656 Jun 4 11:47 vi*
-rwxr-xr-x 1 root root 6444 Jun 4 11:45 whoami*
./dev:
total 8
```

```
drwxr-xr-x 2 root root 4096 Jun 4 12:16 ./
drwxr-xr-x 9 root root 4096 Jun 5 10:05 ../
crw-r--r-- 1 root root 1, 9 Jun 3 13:43 urandom
./etc:
total 208
drwxr-xr-x 4 root root 4096 Jun 4 12:35 ./
drwxr-xr-x 9 root root 4096 Jun 5 10:05 ../
-rw------- 1 root root 0 Jun 4 11:46 .pwd.lock
-rw-r--r-- 1 root root 653 Jun 3 13:46 group
-rw-r--r-- 1 root root 242 Jun 4 11:33 host.conf
-rw-r--r-- 1 root root 857 Jun 4 12:04 hosts
-rw-r--r-- 1 root root 1050 Jun 4 11:29 ld.so.cache
-rw-r--r-- 1 root root 304 Jun 4 11:28 ld.so.conf
-rw-r--r-- 1 root root 235 Jun 4 11:27 ld.so.conf~
-rw-r--r-- 1 root root 88039 Jun 3 13:46 moduli
-rw-r--r-- 1 root root 1342 Jun 4 11:34 nsswitch.conf
drwxr-xr-x 2 root root 4096 Jun 4 12:02 pam.d/
-rw-r--r-- 1 root root 28 Jun 4 12:00 pam_smb.conf
-rw-r--r-- 1 root root 2520 Jun 4 11:57 passwd
-rw-r--r-- 1 root root 7228 Jun 3 13:48 profile
-rw-r--r-- 1 root root 1339 Jun 4 11:33 protocols
-rw-r--r-- 1 root root 274 Jun 4 11:44 resolv.conf
drwxr-xr-x 2 root root 4096 Jun 3 13:43 security/
-rw-r----- 1 root root 1178 Jun 4 11:51 shadow
-rw------- 1 root root 80 Jun 4 11:45 shadow-
-rw-r----- 1 root root 1178 Jun 4 11:48 shadow.old
-rw-r--r-- 1 root root 161 Jun 3 13:46 shells
-rw-r--r-- 1 root root 1144 Jun 3 13:46 ssh_config
-rw------- 1 root root 668 Jun 3 13:46 ssh_host_dsa_key
-rw-r--r-- 1 root root 602 Jun 3 13:46 ssh_host_dsa_key.pub
-rw------- 1 root root 527 Jun 3 13:46 ssh_host_key
-rw-r--r-- 1 root root 331 Jun 3 13:46 ssh_host_key.pub
-rw------- 1 root root 883 Jun 3 13:46 ssh_host_rsa_key
-rw-r--r-- 1 root root 222 Jun 3 13:46 ssh_host_rsa_key.pub
-rw-r--r-- 1 root root 2471 Jun 4 12:15 sshd_config
./etc/pam.d:
total 24
drwxr-xr-x 2 root root 4096 Jun 4 12:02 ./
drwxr-xr-x 4 root root 4096 Jun 4 12:35 ../
lrwxrwxrwx 1 root root 4 Jun 4 12:02 other -> sshd
-rw-r--r-- 1 root root 318 Jun 3 13:46 passwd
-rw-r--r-- 1 root root 546 Jun 4 11:36 ssh
-rw-r--r-- 1 root root 479 Jun 4 12:02 sshd
-rw-r--r-- 1 root root 370 Jun 3 13:46 su
./etc/security:
total 32
```

```
drwxr-xr-x 2 root root 4096 Jun 3 13:43 ./
drwxr-xr-x 4 root root 4096 Jun 4 12:35 ../
-rw-r--r-- 1 root root 1971 Jun 3 13:46 access.conf
-rw-r--r-- 1 root root 184 Jun 3 13:46 chroot.conf
-rw-r--r-- 1 root root 2145 Jun 3 13:46 group.conf
-rw-r--r-- 1 root root 1356 Jun 3 13:46 limits.conf
-rw-r--r-- 1 root root 2858 Jun 3 13:46 pam_env.conf
-rw-r--r-- 1 root root 2154 Jun 3 13:46 time.conf
./lib:
total 8316
drwxr-xr-x 3 root root 4096 Jun 4 12:13 ./
drwxr-xr-x 9 root root 4096 Jun 5 10:05 ../
-rw-r--r-- 1 root root 1024 Jun 4 11:51 cracklib_dict.hwm
-rw-r--r-- 1 root root 214324 Jun 4 11:51 cracklib_dict.pwd
-rw-r--r-- 1 root root 11360 Jun 4 11:51 cracklib_dict.pwi
-rwxr-xr-x 1 root root 342427 Jun 3 13:46 ld-linux.so.2*
-rwxr-xr-x 1 root root 4061504 Jun 3 13:46 libc.so.6*
lrwxrwxrwx 1 root root 15 Jun 4 12:11 libcrack.so -> libcrack.so.2.7*
lrwxrwxrwx 1 root root 15 Jun 4 12:11 libcrack.so.2 -> libcrack.so.2.7*
-rwxr-xr-x 1 root root 33291 Jun 4 11:39 libcrack.so.2.7*
-rwxr-xr-x 1 root root 60988 Jun 3 13:46 libcrypt.so.1*
-rwxr-xr-x 1 root root 71846 Jun 3 13:46 libdl.so.2*
-rwxr-xr-x 1 root root 27762 Jun 3 13:46 libhistory.so.4.0*
lrwxrwxrwx 1 root root 17 Jun 4 12:12 libncurses.so.4 -> libncurses.so.4.2*
-rwxr-xr-x 1 root root 503903 Jun 3 13:46 libncurses.so.4.2*
lrwxrwxrwx 1 root root 17 Jun 4 12:12 libncurses.so.5 -> libncurses.so.5.0*
-rwxr-xr-x 1 root root 549429 Jun 3 13:46 libncurses.so.5.0*
-rwxr-xr-x 1 root root 369801 Jun 3 13:46 libnsl.so.1*
-rwxr-xr-x 1 root root 142563 Jun 4 11:49 libnss_compat.so.1*
-rwxr-xr-x 1 root root 215569 Jun 4 11:49 libnss_compat.so.2*
-rwxr-xr-x 1 root root 61648 Jun 4 11:34 libnss_dns.so.1*
-rwxr-xr-x 1 root root 63453 Jun 4 11:34 libnss_dns.so.2*
-rwxr-xr-x 1 root root 63782 Jun 4 11:34 libnss_dns6.so.2*
-rwxr-xr-x 1 root root 205715 Jun 3 13:46 libnss_files.so.1*
-rwxr-xr-x 1 root root 235932 Jun 3 13:49 libnss_files.so.2*
-rwxr-xr-x 1 root root 204383 Jun 4 11:33 libnss_nis.so.1*
-rwxr-xr-x 1 root root 254023 Jun 4 11:33 libnss_nis.so.2*
-rwxr-xr-x 1 root root 256465 Jun 4 11:33 libnss_nisplus.so.2*
lrwxrwxrwx 1 root root 14 Jun 4 12:12 libpam.so.0 -> libpam.so.0.72*
-rwxr-xr-x 1 root root 31449 Jun 3 13:46 libpam.so.0.72*
lrwxrwxrwx 1 root root 19 Jun 4 12:12 libpam_misc.so.0 ->
libpam_misc.so.0.72*
-rwxr-xr-x 1 root root 8125 Jun 3 13:46 libpam_misc.so.0.72*
lrwxrwxrwx 1 root root 15 Jun 4 12:12 libpamc.so.0 -> libpamc.so.0.72*
-rwxr-xr-x 1 root root 10499 Jun 3 13:46 libpamc.so.0.72*
-rwxr-xr-x 1 root root 176427 Jun 3 13:46 libreadline.so.4.0*
```

```
-rwxr-xr-x 1 root root 44729 Jun 3 13:46 libutil.so.1*
-rwxr-xr-x 1 root root 70254 Jun 3 13:46 libz.a*
lrwxrwxrwx 1 root root 13 Jun 4 12:13 libz.so -> libz.so.1.1.3*
lrwxrwxrwx 1 root root 13 Jun 4 12:13 libz.so.1 -> libz.so.1.1.3*
-rwxr-xr-x 1 root root 63312 Jun 3 13:46 libz.so.1.1.3*
drwxr-xr-x 2 root root 4096 Jun 4 12:00 security/
./lib/security:
total 668
drwxr-xr-x 2 root root 4096 Jun 4 12:00 ./
drwxr-xr-x 3 root root 4096 Jun 4 12:13 ../
-rwxr-xr-x 1 root root 10067 Jun 3 13:46 pam_access.so*
-rwxr-xr-x 1 root root 8300 Jun 3 13:46 pam_chroot.so*
-rwxr-xr-x 1 root root 14397 Jun 3 13:46 pam_cracklib.so*
-rwxr-xr-x 1 root root 5082 Jun 3 13:46 pam_deny.so*
-rwxr-xr-x 1 root root 13153 Jun 3 13:46 pam_env.so*
-rwxr-xr-x 1 root root 13371 Jun 3 13:46 pam_filter.so*
-rwxr-xr-x 1 root root 7957 Jun 3 13:46 pam_ftp.so*
-rwxr-xr-x 1 root root 12771 Jun 3 13:46 pam_group.so*
-rwxr-xr-x 1 root root 10174 Jun 3 13:46 pam_issue.so*
-rwxr-xr-x 1 root root 9774 Jun 3 13:46 pam_lastlog.so*
-rwxr-xr-x 1 root root 13591 Jun 3 13:46 pam_limits.so*
-rwxr-xr-x 1 root root 11268 Jun 3 13:46 pam_listfile.so*
-rwxr-xr-x 1 root root 11182 Jun 3 13:46 pam_mail.so*
-rwxr-xr-x 1 root root 5923 Jun 3 13:46 pam_nologin.so*
-rwxr-xr-x 1 root root 5460 Jun 3 13:46 pam_permit.so*
-rwxr-xr-x 1 root root 18226 Jun 3 13:46 pam_pwcheck.so*
-rwxr-xr-x 1 root root 12590 Jun 3 13:46 pam_rhosts_auth.so*
-rwxr-xr-x 1 root root 5551 Jun 3 13:46 pam_rootok.so*
-rwxr-xr-x 1 root root 7239 Jun 3 13:46 pam_securetty.so*
-rwxr-xr-x 1 root root 6551 Jun 3 13:46 pam_shells.so*
-rwxr-xr-x 1 root root 55925 Jun 4 12:00 pam_smb_auth.so*
-rwxr-xr-x 1 root root 12678 Jun 3 13:46 pam_stress.so*
-rwxr-xr-x 1 root root 11170 Jun 3 13:46 pam_tally.so*
-rwxr-xr-x 1 root root 11124 Jun 3 13:46 pam_time.so*
-rwxr-xr-x 1 root root 45703 Jun 3 13:46 pam_unix.so*
-rwxr-xr-x 1 root root 45703 Jun 3 13:46 pam_unix2.so*
-rwxr-xr-x 1 root root 45386 Jun 3 13:46 pam_unix_acct.so*
-rwxr-xr-x 1 root root 45386 Jun 3 13:46 pam_unix_auth.so*
-rwxr-xr-x 1 root root 45386 Jun 3 13:46 pam_unix_passwd.so*
-rwxr-xr-x 1 root root 45386 Jun 3 13:46 pam_unix_session.so*
-rwxr-xr-x 1 root root 9726 Jun 3 13:46 pam_userdb.so*
-rwxr-xr-x 1 root root 6424 Jun 3 13:46 pam_warn.so*
-rwxr-xr-x 1 root root 7460 Jun 3 13:46 pam_wheel.so*
./sbin:
total 3132
drwxr-xr-x 2 root root 4096 Jun 4 12:35 ./
```

```
drwxr-xr-x 9 root root 4096 Jun 5 10:05 ../
-rwxr-xr-x 1 root root 178256 Jun 3 13:46 choptest*
-rwxr-xr-x 1 root root 184032 Jun 3 13:46 cqtest*
-rwxr-xr-x 1 root root 81096 Jun 3 13:46 dialtest*
-rwxr-xr-x 1 root root 1142128 Jun 4 11:28 ldconfig*
-rwxr-xr-x 1 root root 2868 Jun 3 13:46 lockname*
-rwxr-xr-x 1 root root 3340 Jun 3 13:46 ondelay*
-rwxr-xr-x 1 root root 376796 Jun 3 13:46 pagesend*
-rwxr-xr-x 1 root root 13950 Jun 3 13:46 probemodem*
-rwxr-xr-x 1 root root 9234 Jun 3 13:46 recvstats*
-rwxr-xr-x 1 root root 64480 Jun 3 13:46 sftp-server*
-rwxr-xr-x 1 root root 744412 Jun 3 13:46 sshd*
-rwxr-xr-x 1 root root 30750 Jun 4 11:46 su*
-rwxr-xr-x 1 root root 194632 Jun 3 13:46 tagtest*
-rwxr-xr-x 1 root root 69892 Jun 3 13:46 tsitest*
-rwxr-xr-x 1 root root 43792 Jun 3 13:46 typetest*
./tmp:
total 8
drwxr-xr-x 2 root root 4096 Jun 4 12:32 ./
drwxr-xr-x 9 root root 4096 Jun 5 10:05 ../
./usr:
total 8
drwxr-xr-x 2 root root 4096 Jun 4 12:16 ./
drwxr-xr-x 9 root root 4096 Jun 5 10:05 ../
lrwxrwxrwx 1 root root 7 Jun 4 12:14 bin -> ../bin//
lrwxrwxrwx 1 root root 7 Jun 4 11:33 lib -> ../lib//
lrwxrwxrwx 1 root root 8 Jun 4 12:13 sbin -> ../sbin//
```

Apêndice H

Ambiente chroot para Apache

H.1 Introduçao

O utilitário `chroot` é muitas vezes usado para enjaular um daemon dentro de uma estrutura restrita. Voce pode usá-lo para isolar um serviço do outro, desta forma um problema de segurança em um pacote de software específico não interfere em todo o servidor. A utilização do script `makejail` torna a configuração e atualização da árvore enjaulada muito mais fácil.

FIXME: Apache também pode ser enjaulado usando `http://www.modsecurity.org` que está disponível em `libapache-mod-security` (para Apache 1.x) e `libapache2-mod-security` (para Apache 2.x).

H.1.1 Licença

H.2 Instalando o servidor

Este procedimento foi testado no Debian GNU/Linux 3.0 (Woody) com `makejail` 0.0.4-1 (em Debian/testing).

- Efetue o login como `root` e crie um novo diretório para jaula:

```
$ mkdir -p /var/chroot/apache
```

- Crie um novo usuário e novo grupo. O servidor Apache enjaulado irá executar com este usuário/grupo, que não é utilizado para mais nada no sistema. Neste exemplo, ambos usuário e grupo são chamados de `chrapach`.

```
$ adduser --home /var/chroot/apache --shell /bin/false \
--no-create-home --system --group chrapach
```

FIXME: é preciso um novo usuário? (Apache já executa como usuário apache)

- Instale o Apache normalmente no Debian: `apt-get install apache`
- Configure o Apache (por exemplo defina seus subdomínios e etc.). No arquivo de configuração `/etc/apache/httpd.conf`, altere as opções *Group* e *User* para `chrapach`. Reinicie o Apache e tenha certeza que o servidor está funcionando corretamente. Agora, pare o daemon do Apache.
- Instale o `makejail` (disponível agora no Debian/testing). Você também deve instalar `wget` e `lynx`, pois eles serão usados pelo `makejail` para testar o servidor enjaulado: `apt-get install makejail wget lynx`
- Copie o arquivo de configuração de exemplo para o Apache para o diretório `/etc/makejail`:

```
# cp /usr/share/doc/makejail/examples/apache.py /etc/makejail/
```

- Edite o arquivo `/etc/makejail/apache.py`. Você precisa alterar as opções *chroot*, *users* e *groups*. Para executar esta versão do `makejail`, você também pode adicionar a opção `packages`. Veja a documentação do makejail (http://www.floc.net/makejail/current/doc/). Veja o exemplo mostrado abaixo:

```
chroot="/var/chroot/apache"
testCommandsInsideJail=["/usr/sbin/apachectl start"]
processNames=["apache"]
testCommandsOutsideJail=["wget -r --spider http://localhost/",
                         "lynx --source https://localhost/"]
preserve=["/var/www",
          "/var/log/apache",
          "/dev/log"]
users=["chrapach"]
groups=["chrapach"]
packages=["apache", "apache-common"]
userFiles=["/etc/password",
           "/etc/shadow"]
groupFiles=["/etc/group",
            "/etc/gshadow"]
forceCopy=["/etc/hosts",
           "/etc/mime.types"]
```

FIXME: algumas opções parecem não funcionar corretamente. Por exemplo, `/etc/shadow` e `/etc/gshadow` não são copiados, visto que `/etc/password` e `/etc/group` são copiados em vez de serem filtrados.

- Crie a árvore da jaula: `makejail /etc/makejail/apache.py`
- Se `/etc/password` e `/etc/group` forem copiados completamente, digite:

```
$ grep chrapach /etc/passwd > /var/chroot/apache/etc/passwd
$ grep chrapach /etc/group > /var/chroot/apache/etc/group
```

  para substituí-los com as cópias filtradas.

- Copie as páginas e os logs do site Web dentro da jaula. Estes arquivos não são copiados automaticamente (veja a opção *preserve* no arquivo de configuração do `makejail`).

```
# cp -Rp /var/www /var/chroot/apache/var
# cp -Rp /var/log/apache/*.log /var/chroot/apache/var/log/apache
```

- Edite o script de inicialização para que o daemon de logging do sistema também ouça do socket `/var/chroot/apache/dev/log`. No arquivo `/etc/init.d/sysklogd`, substitua: `SYSLOGD=""` com `SYSLOGD=" -a /var/chroot/apache/dev/log"` e reinicie o daemon (`/etc/init.d/sysklogd restart`).
- Edite o script de inicialização do Apache (`/etc/init.d/apache`). Você pode precisar fazer algumas alterações no script de inicialização padrão para que ele funcione apropriadamente com a árvore enjaulada. Como:
  - configure uma nova variável *CHRDIR* no início do arquivo;
  - edite as seções *start*, *stop*, *reload*, etc.;
  - adicione uma linha para montar e desmontar o sistema de arquivo `/proc` que está dentro da jaula.

```
#! /bin/bash
#
# apache Start the apache HTTP server.
#

CHRDIR=/var/chroot/apache

NAME=apache
PATH=/bin:/usr/bin:/sbin:/usr/sbin
DAEMON=/usr/sbin/apache
SUEXEC=/usr/lib/apache/suexec
PIDFILE=/var/run/$NAME.pid
CONF=/etc/apache/httpd.conf
```

```
APACHECTL=/usr/sbin/apachectl

trap "" 1
export LANG=C
export PATH

test -f $DAEMON || exit 0
test -f $APACHECTL || exit 0

# ensure we don't leak environment vars into apachectl
APACHECTL="env -i LANG=${LANG} PATH=${PATH} chroot $CHRDIR $APACHECTL"

if egrep -q -i "^[[:space:]]*ServerType[[:space:]]+inet" $CONF
then
    exit 0
fi

case "$1" in
  start)
    echo -n "Starting web server: $NAME"
    mount -t proc proc /var/chroot/apache/proc
    start-stop-daemon --start --pidfile $PIDFILE --exec $DAEMON \
 --chroot $CHRDIR
    ;;

  stop)
    echo -n "Stopping web server: $NAME"
    start-stop-daemon --stop --pidfile "$CHRDIR/$PIDFILE" --oknodo
    umount /var/chroot/apache/proc
    ;;

  reload)
    echo -n "Reloading $NAME configuration"
    start-stop-daemon --stop --pidfile "$CHRDIR/$PIDFILE" \
 --signal USR1 --startas $DAEMON --chroot $CHRDIR
    ;;

  reload-modules)
    echo -n "Reloading $NAME modules"
    start-stop-daemon --stop --pidfile "$CHRDIR/$PIDFILE" --oknodo \
 --retry 30
    start-stop-daemon --start --pidfile $PIDFILE \
 --exec $DAEMON --chroot $CHRDIR
    ;;

  restart)
```

```
    $0 reload-modules
    exit $?
    ;;

  force-reload)
    $0 reload-modules
    exit $?
    ;;

  *)
    echo "Usage: /etc/init.d/$NAME {start|stop|reload|reload-modules|fo
    exit 1
    ;;
esac

if [ $? == 0 ]; then
 echo .
 exit 0
else
 echo failed
 exit 1
fi
```

FIXME: should the first Apache process be run as another user than root (i.e. add –chuid chrapach:chrapach)? Cons: chrapach will need write access to the logs, which is awkward.

- Substitua no `/etc/logrotate.d/apache` o `/var/log/apache/*.log` com `/var/chroot/apache/var/log/apache/*.log`
- Inicialize o Apache (`/etc/init.d/apache start`) e verifique o que está sendo reportado no log da jaula (`/var/chroot/apache/var/log/apache/error.log`). Se a sua configuração for mais complexa (exemplo: se também utiliza PHP e MySQL), alguns arquivos provavelmente estarão faltando. Se estes arquivos não são copiados automaticamente pelo `makejail`, você pode listá-los com a opção *forceCopy* (para copiar os arquivos diretamente) ou *packages* (para copiar pacotes completos e suas dependências) no arquivo de configuração `/etc/makejail/apache.py`.
- Digite `ps aux | grep apache` para ter certeza que o Apache está rodando. Você deve ver algo do tipo:

```
root 180 0.0 1.1 2936 1436 ? S 04:03 0:00 /usr/sbin/apache
chrapach 189 0.0 1.1 2960 1456 ? S 04:03 0:00 /usr/sbin/apache
chrapach 190 0.0 1.1 2960 1456 ? S 04:03 0:00 /usr/sbin/apache
chrapach 191 0.0 1.1 2960 1456 ? S 04:03 0:00 /usr/sbin/apache
chrapach 192 0.0 1.1 2960 1456 ? S 04:03 0:00 /usr/sbin/apache
chrapach 193 0.0 1.1 2960 1456 ? S 04:03 0:00 /usr/sbin/apache
```

- Certifique-se que os processos do Apache estão sendo executados na jaula chroot procurando no sistema de arquivo `/proc`: `ls -la /proc/process_number/root/.` onde *process_number* é um dos PID listados acima (por exemplo: segunda coluna; PID 189). As entradas para a árvore restrita devem ser listadas:

```
drwxr-sr-x 10 root staff 240 Dec 2 16:06 .
drwxrwsr-x 4 root staff 72 Dec 2 08:07 ..
drwxr-xr-x 2 root root 144 Dec 2 16:05 bin
drwxr-xr-x 2 root root 120 Dec 3 04:03 dev
drwxr-xr-x 5 root root 408 Dec 3 04:03 etc
drwxr-xr-x 2 root root 800 Dec 2 16:06 lib
dr-xr-xr-x 43 root root 0 Dec 3 05:03 proc
drwxr-xr-x 2 root root 48 Dec 2 16:06 sbin
drwxr-xr-x 6 root root 144 Dec 2 16:04 usr
drwxr-xr-x 7 root root 168 Dec 2 16:06 var
```

  Para automatizar este teste, você pode digitar:`ls -la /proc/`cat /var/chroot/apache/var/run/apache.pid`/root/.`

  FIXME: Add other tests that can be run to make sure the jail is closed?

A razão pela qual eu gosto disso é que a configuração da jaula não é tão complicada e o servidor pode ser atualizado em somente duas linhas:

```
apt-get update && apt-get install apache
makejail /etc/makejail/apache.py
```

H.3 Veja também

Se você está procurando por mais informações você pode considerar as referências que foram utilizadas para fazer este tutorial:

- makejail homepage (`http://www.floc.net/makejail/`) , this program was written by Alain Tesio)
- Chrooting daemons and system processes HOWTO (`http://www.nuclearelephant.com/papers/chroot.html`) by Jonathan, Network Dweebs, 21/10/2002