Red Hat Enterprise Linux 6

Guia de Seguran?a

Um Guia para Proteger o Red Hat Enterprise Linux

Edição 1.5

Last Updated: 2017-10-29

Red Hat Enterprise Linux 6 Guia de Seguran?a

Um Guia para Proteger o Red Hat Enterprise Linux
Edição 1.5

Nota Legal

Resumo

Este documento assiste usuários e administradores no aprendizado dos processos e práticas de proteção de estações de trabalho e servidores contra a invasão remota e local, exploração e atividades mal-intencionadas. Focado no Red Hat Enterprise Linux mas detalhando conceitos e técnicas válidas para todos os sistemas Linux, este guia detalha o planejamento e ferramentas envolvidos na criação de um ambiente de informática protegido para o centro de dados, local de trabalho e lar. Com conhecimento administrativo adequado, vigilância e ferramentas, os sistemas com Linux podem ser tanto funcionais como protegidos da maioria das invasões comuns e métodos de explorações.

Índice

CAPÍTULO 1. VISÃO GERAL DA SEGURANÇA

Por causa da dependência crescente em poderosas redes de computadores para auxiliar empresas e manter registro de nossas informações pessoais, indústrias inteiras foram formadas em torno da prática das redes e segurança da informática. Organizações têm solicitado o conhecimento e habilidades de profissionais de segurança para propriamente auditar sistemas e criar soluções que se encaixam dentro das necessidades operacionais dessas organizações. Pela razão que a maioria das organizações estão cada vez mais dinâmicas por natureza, seus funcionários acessam recursos de TI da empresa críticos localmente e remotamente, por isso a necessidade de ambientes de computação seguros têm se tornado mais evidente.

Infelizmente, muitas organizações (e também usuários individuais) deixam a segurança em segundo plano, um processo que é esquecido perde-se aumento do poder, produtividade, conveniencia, facilidade de uso e questões de orçamento. Uma implementação de segurança apropriada é muitas vezes colocada em prática tarde demais — *depois* de uma invasão não autorizada já tiver ocorrido. Tomar medidas apropriadas antes de se conectar a uma rede não confiável, como a Internet é uma meio efetivo de impedir tentativas de intrusão.

NOTA

Este documento faz diversas referências à arquivos no diretório **`/lib`**. Se estiver usando os sistemas 64 bits, alguns dos arquivos mencionados podem estar localizados no **`/lib64`**.

1.1. INTRODUÇÃO À SEGURANÇA

1.1.1. O que é Segurança de Computadores?

Segurança de computadores é um termo geral que cobre uma grande área da informática e processamento da informação. Indústrias que dependem dos sistemas de computadores e redes para conduzir diariamente negociações e acessar informações críticas, consideram seus dados como uma importante parte de seus bens gerais. Diversos termos e métricas entraram em nosso vocabulário de negócios, tais como o custo total da posse (TCO - Total Cost of Ownership), retorno sobre investimento (ROI - Return on Investment), e qualidade de serviço (QoS - Quality of Service). Usando estas métricas, indústrias podem calcular aspectos como a integridade dos dados e alta disponibilidade (HA - High-Availability) como parte de seus planos e custos do gerenciamento de processos. Em algumas indústrias, como a de comércio eletrônico, a disponibilidade e confiabilidade dos dados podem significar a diferença entre sucesso e fracasso.

1.1.1.1. Como a Segurança de Computadores começou?

A segurança da informação tem evoluído ao longo dos anos devido à crescente dependência em redes públicas para não expor informações pessoais, financeiras e outras informações restritas. Existem muitas ocorrências tais como os casos Mitnick [1] e Vladimir Levin [2] que fizeram com que organizações de todas as áreas repensassem a maneira que lidam com a informação, incluindo sua transmissão e exposição. A popularidade da internet foi um dos mais importantes desenvolvimentos que levou a um esforço intensificado da segurança dos dados.

Um número crescente de pessoas estão usando seus computadores pessoais para obter acesso aos recursos que a internet oferece. Da pesquisa e obtenção de informação ao correio eletrônico e transações comerciais, a internet tem sido considerada um dos mais importantes desenvolvimentos do século 20.

A internet e seus primeiros protocolos, entretanto, foram desenvolvidos como um sistema *baseado em*

confiança. Ou seja, o Protocolo de Internet (IP) não foi desenvolvido para ser propriamente seguro. Não existem padrões de segurança aprovados construídos na pilha de comunicações TCP/IP, deixando-o aberto para usuários potencialmente maliciosos e processos na rede. Desenvolvimentos modernos têm feito a comunicação na internet mais segura, mas ainda existem diversos incidentes que ganham atenção nacional e nos alertam para o fato de que nada é completamente seguro.

1.1.1.2. A Segurança Hoje

Em fevereiro de 2000, um ataque de negação de serviço (DDoS) foi feito em vários dos principais sites de alto tráfego na internet. O ataque fez que sites como yahoo.com, cnn.com, amazon.com, fbi.gov entre outros ficassem completamente fora de alcance dos usuários normais, já que o ataque afetou roteadores por várias horas com enormes pacotes de dados ICMP, também conhecidos como *ping flood*. O ataque foi feito por invasores desconhecidos usando programas especialmente criados e totalmente disponíveis que escanearam servidores de rede vulneráveis e instalaram aplicativos clientes chamados *Trojans* nesses servidores e agendaram um ataque inundando os sites vítimas e os tornando indisponíveis. Muitos culparam esse ataque devido à deficiências fundamentais na maneira que roteadores e os protocolos usados são estruturados para aceitar todo o tráfego dos dados, sem importar de onde ou qual o propósito dos pacotes são enviados.

Em 2007, uma violação de dados explorando as fraquezas amplamente conhecidas do WEP (Wired Equivalent Privacy), protocolo de encriptação sem fio, resultou no roubo de mais de 45 milhões de números de cartão de crédito de uma instituição financeira global.[3]

Em um incidente separado, os registros de pagamentos de mais de 2.2 milhões de pacientes armazenados em uma fita de backup foram roubados do banco dianteiro de um carro de entregas [4]

Atualmente, estima-se que 1.4 bilhões de pessoas usam ou usaram a internet no mundo todo. [5] Ao mesmo tempo:

- Em qualquer dia, existem aproximadamente 225 importantes incidências de brecha de segurança reportadas ao CERT Coordination Center at Carnegie Mellon University.[6]
- O número de incidentes reportados no CERT pulou de 52,658 em 2001, 82,094 em 2002 e para 137,529 em 2003.[7]
- De acordo com o FBI, crimes relacionados à computadores custou às empresas americanas $67.2 bilhões de dólares em 2006.[8]

De uma pesquisa global feita em 2009 sobre segurança e profissionais da tecnologia da informação, "Porque a Segurança Importa Agora"[9], conduzida pela *CIO Magazine*, alguns resultados interessantes são:\n\t\n

- Apenas 23% dos que responderam possuem políticas para o uso das tecnologias de Web 2.0. Estas tecnologias, como Twitter, Facebook e LinkedIn podem oferecer uma maneira conveniente para empresas e individuais se comunicarem e colaborar, entretanto elas abrem novas vulnerabilidades, primariamente o vazamento de dados confidenciais.
- Mesmo durante a recente crise financeira de 2009, a pesquisa constatou que os orçamentos com segurança estavam no mesmo nível ou em crescimento em relação aos anos anteriores (aproximadamente 2 de cada 3 participantes esperam aumentar ou manter o mesmo nível). Isto é uma boa notícia e reflete a importância que organizações estão dando na segurança da informação hoje.

Estes resultados reforçam a realidade que segurança de computadores se tornou um gasto quantificável e justificável nos orçamentos de TI. Organizações que requerem integridade dos dados e alta

disponibilidade evocam as habilidades dos administradores de sistemas, desenvolvedores e engenheiros para garantir uma confiança de seus sistemas, serviços e informações 24x7. Ser vítima de usuários maliciosos, processos ou ataques coordenados é uma ameaça direta ao sucesso da organização.

Infelizmente, a segurança da rede e de sistemas podem ser uma proposta difícil, exigindo um conhecimento complexo de como a empresa encara, usa, manipula e transmite suas informações. Compreender a forma como uma organização (e as pessoas que compõem a organização) conduz os negócios é fundamental para implementação de um plano de segurança apropriado.

1.1.1.3. Padronização da Segurança

Empresas de todas as áreas confiam em regulamentos e regras que são definidas por entidades de padronização como a American Medical Association (AMA) ou o Institute of Electrical and Electronics Engineers (IEEE). Os mesmos ideais são verdadeiros para a segurança da informação. Muitos consultores de segurança e fornecedores concordam com um modelo de padronização da segurança conhecido como CIA, ou *Confidentiality, Integrity, and Availability*. Este modelo de três camadas é um componente geralmente aceito para avaliar riscos de informações sensíveis e estabelecer política de segurança. O seguinte descreve o modelo CIA em maiores detalhes:

- Confidencialidade — Informações sensíveis devem estar disponíveis somente para um conjunto de indivíduos pré definidos. Transmissão não autorizada e uso da informação devem ser restritos. Por exemplo, a confidencialidade da informação garante que uma informação pessoal ou financeira não seja obtida por um indivíduo não autorizado para propósitos maliciosos tais como roubo de identidade ou fraude de crédito.

- Integridade — As informações não devem ser alteradas de modo a torna-las incompletas ou incorretas. Usuários não autorizados devem ser restritos da possibilidade de modificar ou destruir informações sensíveis.

- Disponibilidade — As informações devem ser acessíveis a usuários autorizados em qualquer momento que seja necessário. Disponibilidade é uma garantia que a informação pode ser obtida com acordos de frequência e pontualidade. Isto é frequentemente medido em termos de porcentagens e definido formalmente em Acordos de Nível de Serviço (SLAs) usados por provedores de serviços de redes e seus clientes corporativos.

1.1.2. SELinux

O Red Hat Enterprise Linux inclui uma melhoria ao kernel do Linux chamado SELinux, que implementa uma arquitetura de Controle de Acesso Obrigatório (MAC - Mandatory Access Control) que fornece um nível de controle refinado sobre arquivos, processos, usuários e aplicações no sistema. Uma discussão detalhada sobre o SELinux está além do objetivo deste documento; entretanto, para mais informações sobre o SELinux e seu uso no Red Hat Enterprise Linux, consulte o Guia do Usuário SELinux do Red Hat Enterprise Linux. Para mais informações sobre configurar e rodar serviços que são protegidos pelo SELinux, consulte o Guia do SELinux Gerenciando Serviços Confinados. Outros recursos disponíveis para o SELinux estão listados no Capítulo 8, *Referências*.

1.1.3. Controles de Segurança

A segurança de computadores é frequentemente dividida em três categorias principais distintas, comumente referidas como *controles*:

- Físico

- Técnico

- Administrativo

Estas três categorias amplas definem os objetivos principais de uma implementação de segurança apropriada. Dentro destes controles estão sub categorias que detalham mais os controles e como implementa-las.

1.1.3.1. Controles Físicos

Controle Físico é a implementação de medidas de segurança em uma estrutura definida usada para deter ou previnir acesso não autorizado à material sensível. Exemplos de controles físicos são:

- Cameras de vigilância de circuito interno
- Sistemas de alarmes térmicos ou de movimento
- Guardas de Segurança
- IDs com fotos
- Portas de aço bloqueadas com parafusos sem cabeça
- Biometria (inclui impressão digital, voz, rosto, íris, manuscrito e outros métodos automatizados usados para reconhecer indivíduos)

1.1.3.2. Controles Técnicos

Controles técnicos usam técnologia como uma base para controlar o acesso e uso de dados sensíveis através de uma estrutura física e sobre uma rede. Controles técnicos são de alcance abrangente e incluem tecnologias como:

- Criptografia
- Cartões Smart
- Autenticação de rede
- Listas de Controle de Acesso (ACLs)
- Software de auditoria de integridade de arquivos

1.1.3.3. Controles Administrativos

Controles administrativos definem os fatores humanos de segurança. Eles envolvem todos o níveis de pessoas dentro de uma organização e determinam quais usuários possuem acesso a quais recursos e informações por tais meios como:

- Treinamento e conscientização
- Prevenção de desastres e planos de recuperação
- Estratégias de recrutamento de pessoal e de separação
- Registro de pessoal e de contabilidade

1.1.4. Conclusão

Agora que você aprendeu sobre as origens, motivos e aspectos da segurança, você achará mais fácil de determinar o plano de ação apropriado ao Red Hat Enterprise Linux. É importante saber quais fatores e condições compoem a segurança a fim de planejar e implementar uma estratégia apropriada. Com esta informação em mente, o processo pode ser formalizado e o caminho se torna mais claro à medida que você se aprofunda nos detalhes do processo de segurança.

1.2. AVALIAÇÃO DE VULNERABILIDADE

Tendo tempo, recursos e motivação, um invasor pode invadir praticamente qualquer sistema. Todos os procedimentos de segurança e tecnologias atualmente disponíveis não podem garantir que quaisquer sistemas estejam completamente seguros contra intrusão. Roteadores podem ajudar a proteger gateways na internet. Firewalls ajudam a proteger as bordas da rede. Redes Privadas Virtuais seguramente transmitem dados em um fluxo criptografado. Os sistemas de detecção de intrusão lhe avisam de atividade maliciosa. Entretanto, o sucesso de cada uma dessas tecnologias é dependente de uma variedade de variáveis, incluindo:

- A perícia do pessoal responsável pela configuração, monitoramento e manutenção das tecnologias.
- A habilidade de corrigir e atualizar serviços e kernels rapidamente e eficientemente.
- A habilidade daqueles responsáveis em manter constante vigilância sobre a rede.

Dado o estado dinâmico de sistemas de dados e tecnologias, proteger recursos corporativos pode ser muito complexo. Devido a esta complexidade, muitas vezes é difícil de encontrar recursos para todos os seus sistemas. Enquanto é possível ter um pessoal com conhecimentos em muitas áreas da segurança da informação em um alto nível, é difícil de reter empregados que são especialistas em mais do que algumas poucas áreas. Isto é principalmente pelo motivo que cada assunto da área da segurança da informação requer constante atenção e foco. A segurança da informação não fica parada.

1.2.1. Pensando Como o Inimigo

Suponha que você administra uma rede corporativa. Essas redes comumente compreendem de sistemas operacionais, aplicações, servidores, monitores de rede, firewalls, sistemas de detecção de intrusão e mais. Agora imagine tentar manter atualizados cada um desses mencionados. Dada a complexidade dos softwares de hoje e ambientes de rede, explorações e bugs são uma certeza. Manter-se atualizado com correções e atualizações para uma rede inteira pode ser uma tarefa difícil em uma grande organização com sistemas heterogêneos.

Combine o requerimento de experiência com a tarefa de manter-se atualizado, é inevitável que incidentes adversos ocorram, sistemas sejam violados, dados se corrompem e serviços são interrompidos.

Para aprimorar as tecnologias de segurança e auxiliar na proteção de sistemas, você deve pensar como um invasor e avaliar a segurança de seus sistemas verificando os pontos fracos. Avaliações de vulnerabilidade preventivas em seus próprios sistemas e recursos de rede podem revelar problemas potenciais que podem ser endereçados antes de um invasor explora-la.

A avaliação de vulnerabilidade é uma auditoria interna de seu sistema e segurança de sistema; os resultados dos quais indicam a confidencialidade, integridade e disponibilidade de sua rede (como explicado na Seção 1.1.1.3, “Padronização da Segurança”). Tipicamente, a avaliação da vulnerabilidade começa com uma fase de reconhecimento, durante o qual os dados importantes sobre os sistemas alvos e os recursos são reunidos. Esta fase leva à fase de prontidão do sistema, onde o alvo é checado com todas as vulnerabilidades conhecidas. A fase de prontidão culmina na fase do relatório, onde os resultados são classificados em categorias de alto, médio e baixo risco e métodos para melhorar a segurança (ou para minimizar o risco de vulnerabilidade) do alvo são discutidos.

Se você tivesse que realizar uma avaliação de vulnerabilidade de sua casa, você verificaria cada porta para ver se estão fechadas e trancadas. Você também checaria cada janela, tendo certeza que estão completamente fechadas e trancam corretamente. Este mesmo conceito se aplica à sistemas, redes e dados eletrônicos. Usuários maliciosos são os ladrões e vândalos de seus dados. Foque nas ferramentas, mentalidade e motivações e você poderá então reagir rapidamente às ações.

1.2.2. Definindo a Avaliação e Testes

Avaliações de vulnerabilidade podem ser divididas em dois tipos: *olhar de fora para dentro* e *olhar ao redor internamente*.

Quando realizar uma avaliação de vulnerabilidade olhando de fora para dentro, você está tentando comprometer seus sistemas a partir do lado de fora. Estando externo à sua empresa, lhe possibilita ter uma visão do invasor. Você vê o que o invasor vê — IPs roteáveis publicamente, endereços, sistemas em seu *DMZ*, interfaces externas de seu firewall e mais. DMS significa "zona demilitarizada", que corresponde a um computador ou sub-rede que fica entre a rede interna confiável, como uma LAN privada corporativa e uma rede externa não confiável, como a internet pública. Tipicamente, o DMZ contém dispositivos acessíveis ao tráfego de internet, como servidores web (HTTP), servidores FTP, servidores SMTP (e-mail) e servidores DNS.

Quando você realizar uma avaliação de vulnerabilidade de olhar ao redor internamente, você possui uma vantagem já que você está dentro e seu estado é elevado à confiável. Este é o ponto de vista que você e seus colegas de trabalho possuem uma vez autenticados a seus sistemas. Você vê servidores de impressão, servidores de arquivos e outros recursos.

Existem diferenças notáveis entre os dois tipos de avaliação de vulnerabilidade. Sendo interno à sua empresa lhe dá mais privilégios do que um externo. Na maioria das organizações, a segurança é configurada para manter invasores fora. Muito pouco é feito para proteger a parte interna da organização (como firewalls de departamentos, controles de acesso de nível de usuários e procedimentos de autenticação para recursos internos). Tipicamente, existem muito mais recursos quando olhar internamente já que a maioria dos sistemas são internos à uma empresa. Uma vez que você está fora da empresa, seu estado é não confiável. Os sistemas e recursos disponíveis para você externamente são normalmente muito limitados.

Considere a diferença entre avaliações de vulnerabilidade e *testes de penetração*. Pense em uma avaliação de vulnerabilidade como o primeiro passo de um teste de invasão. As informações obtidas no teste são usadas para testes. Onde a avaliação é feita para verificar por brechas e vulnerabilidades potenciais, o teste de penetração na verdade tenta explorar as descobertas.

Avaliar a infraestrutura de rede é um processo dinâmico. A segurança, tanto de informação quanto física são dinâmicas. Realizar uma avaliação mostra uma visão geral, que pode transformar falsos positivos e falsos negativos.

Administradores de segurança são somente tão bons quanto as ferramentas que eles usam e o conhecimento que eles retém. Tome quaisquer das ferramentas de avaliação atualmente disponíveis, rode-as em seu sistema e é quase uma garantia que haverão falsos positivos. Seja pelo defeito de um programa ou erro do usuário, o resultado é o mesmo. A ferramenta poderá encontrar vulnerabilidades que em realidade não existem (falsos positivos); ou ainda pior, a ferramenta pode não encontrar vulnerabilidades que na verdade existem (falsos negativos).

Note que a diferença é definida entre a avaliação de vulnerabilidade e o teste de invasão, tome os resultados da avaliação e as revise cuidadosamente antes de conduzir um teste de invasão como parte de sua nova abordagem de boas práticas.

ATENÇÃO

Tentar explorar vulnerabilidades de recursos em produção poderá ter efeitos adversos à produtividade e eficiência de seus sistemas e rede.

A seguinte lista examina alguns dos benefícios de realizar avaliações de vulnerabilidade.

- Cria um foco pró ativo na segurança da informação
- Encontra potenciais explorações antes que os invasores as encontrem
- Resulta em sistemas sendo atualizados e corrigidos
- Promove o crescimento e ajuda no desenvolvimento das habilidades dos funcionários
- Reduz perdas financeiras e publicidade negativa

1.2.2.1. Estabelece uma Metodologia

Para ajudar na seleção de ferramentas para uma avaliação de vulnerabilidade, é útil estabelecer uma metodologia de avaliação de vulnerabilidade. Infelizmente, não há no momento uma metodologia pré definida ou aprovada pela indústria neste momento; entretanto, o bom senso e boas práticas podem atuar como um guia suficiente.

O que é um alvo? Estamos olhando em um servidor ou em uma rede inteira e tudo dentro dessa rede? Estamos externos ou internos à empresa? As respostas à estas questões são importantes conforme elas ajudam a determinar não somente quais ferramentas escolher mas também a maneira a qual elas são usadas.

Para aprender mais sobre estabelecer metodologias, consulte os seguintes websites:

- http://www.isecom.org/osstmm/ *The Open Source Security Testing Methodology Manual* (OSSTMM)
- http://www.owasp.org/ *The Open Web Application Security Project*

1.2.3. Avaliando as Ferramentas

Uma avaliação pode se iniciar usando alguma ferramenta de coleta de informações. Quando avaliar a rede inteira, mapeie a estrutura primeiro para encontrar os hosts em execução. Uma vez localizados, examine cada host individualmente. Focar nestes hosts requer um outro conjunto de ferramentas. Conhecer quais ferramentas usar pode ser um passo crucial para encontrar vulnerabilidades.

Assim como qualquer aspecto do dia a dia, existem muitas ferramentas diferentes que realizam a mesma tarefa. Este conceito se aplica também para realizar avaliações de vulnerabilidade. Existem ferramentas especificas para sistemas operacionais, aplicações e mesmo redes (baseadas nos protocolos usados). Algumas ferramentas são grátis, outras não. Algumas ferramentas são intuitivas e fáceis de usar, enquanto outras são ocultas e mal documentadas mas possuem recursos que outras ferramentas não possuem.

Encontrar as ferramentas certas, pode ser uma tarefa difícil e no final, a experiência é que conta. É

possível montar um laboratório de testes e tentar o máximo de ferramentas que você puder, anotando os pontos fortes e fracos de cada uma. Revise o arquivo LEIAME ou página man da ferramenta. Adicionalmente, procure na internet para mais informações, como artigos, guias passo a passo ou mesmo listas de e-mails específicas da ferramenta.

As ferramentas discutidas abaixo são apenas um pequeno exemplo das ferramentas disponíveis.

1.2.3.1. Escaneando Hosts com o Nmap

O Nmap é uma ferramenta popular que pode ser usada para determinar o desenho de uma rede. O Nmap está disponível por muitos anos e é provavelmente a ferramenta mais usada para coletar informações. Uma excelente página man está incluída que fornece descrições detalhadas de suas opções e usos. Os administradores podem usar o Nmap em uma rede para encontrar sistemas de host e portas abertas nesses sistemas.

O Nmap é um competente primeiro passo na avaliação de vulnerabilidade. Você pode mapear todos os hosts dentro de sua rede e mesmo colocar uma opção que permite o Nmap tentar identificar os sistema operacional rodando em um determinado host. O Nmap é uma boa base para estabelecer uma política de uso de dispositivos seguros e restringir serviços não usados.

1.2.3.1.1. Usando o Nmap

O Nmap pode ser executado a partir da prompt do shell digitando o comando **nmap** seguido pelo nome do host ou endereço de IP da máquina a ser escaneada.

```
nmap foo.example.com
```

Os resultados de um escaneamento básico (que pode levar alguns minutos, dependendo de onde o host está localizado e outras condições de rede) devem ser similar ao seguinte:

```
Interesting ports on foo.example.com:
Not shown: 1710 filtered ports
PORT    STATE  SERVICE
22/tcp  open   ssh
53/tcp  open   domain
80/tcp  open   http
113/tcp closed auth
```

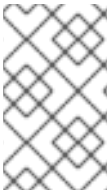
O Nmap testa as portas de comunicação de rede mais comuns para escutar ou aguardar por serviços. Esta informação pode ser útil para um administrador que quer terminar serviços não usados ou desnecessários.

Para mais informações sobre usar o Nmap, consulte a página web oficial na seguinte URL:

http://www.insecure.org/

1.2.3.2. Nessus

O Nessus é um escaner de segurança de serviço completo. A arquitetura de plug in do Nessus permite usuários personaliza-lo para seus sistemas e redes. Como em qualquer escaner, o Nessus é somente tão bom quanto a assinatura de banco de dados em que ele está. Felizmente, o Nessus é frequentemente atualizado e contém recursos de relatório completos, escaneamento de host e busca de vulnerabilidades em tempo real. Se lembre que podem existir falsos positivos e falsos negativos, mesmo em uma ferramenta tão poderosa e frequentemente atualizada como o Nessus.



NOTA

O cliente Nessus e o software do servidor requerem uma subscrição para serem usados. Ela foi incluída neste documento como uma referencia à usuários que podem estar interessados em usar esta aplicação popular.

Para mais informações sobre o Nessus, consulte o web site oficial no seguinte endereço:

http://www.nessus.org/

1.2.3.3. Nikto

Nikto é um excelente escaner de script de interface de gateway comum (CGI). O Nikto não somente verifica por vulnerabilidades CGI mas o faz de uma maneira evasiva, para então enganar sistemas de detecção de intrusão. O Nikto vem com uma documentação completa que deve ser cuidadosamente revisada antes de rodar o programa. Se você tiver servidores web rodando scripts CGI, o Nikto pode ser um excelente recurso para checar a segurança destes servidores.

Mais informações sobre o Nikto podem ser encontradas no seguinte endereço:

http://cirt.net/nikto2

1.2.3.4. Antecipar Suas Futuras Necessidades

Dependendo de seus alvos e recursos, existem muitas ferramentas disponíveis. Existem ferramentas para redes sem fio, redes Novell, sistemas Windows, sistemas Linux e outros mais. Outra parte essencial de realizar avaliações podem incluir revisar a segurança física, revista de pessoal ou avaliação de rede PBX/voz. Novos conceitos, como *war walking* e *wardriving*, que envolve o escaneamento dos perímetros de suas estruturas físicas de sua empresa por vulnerabilidades da rede sem fio, são alguns conceitos que você deve investigar e se necessário, incorporar suas tarefas. Imaginação e exposição são os únicos limites para planejar e conduzir avaliações de vulnerabilidade.

1.3. INVASORES E VULNERABILIDADES

Para planejar e implementar uma boa estratégia de segurança, primeiro esteja atento a alguns dos questões que determinaram e motivaram invasores a explorar e comprometer sistemas. Entretanto, antes de detalhar essas questões, a terminologia usada para identificar um ataque deve ser definida.

1.3.1. Uma Rápida História sobre Hackers

O significado moderno do termo *hacker* tem origem por volta dos anos 60 e o Tech Model Railroad Club do Instituto de Tecnologia de Massachussetts (MIT), que desenvolvia conjuntos de trens em larga escala e com complexos detalhes. Hacker era nome usado pelos membros do clube que descobriam um truque ou uma maneira de resolver um problema.

O termo hacker então começou a ser usado para tudo, desde viciados em computadores até programadores de talento. Uma característica comum entre a maioria dos hackers é a vontade de explorar em detalhes como sistemas de computadores e redes funcionam com pouca ou nenhuma motivação externa. Desenvolvedores de software de código aberto frequentemente consideram a si próprios e seus colegas serem hackers e usam a palavra como um termo de respeito.

Tipicamente, hackers seguem uma forma de *ética hacker* que diz que a busca por informações e perícia são essenciais e compartilhar esse conhecimento é uma função dos hackers à comunidade. Durante essa busca por conhecimento, alguns hackers desfrutam os desafios acadêmicos de contornar controles

de segurança em sistemas de computadores. Por este motivo, a imprensa muitas vezes usa o termo hacker para descrever aqueles que ilicitamente acessam sistemas e redes com intenções criminais, maliciosas ou sem escrúpulos. O termo mais preciso para este tipo de hacker de computadores é *cracker* — um termo criado por hackers na década de 80 para diferenciar as duas comunidades.

1.3.1.1. Tons de cinza

Dentro da comunidade de indivíduos que encontram e exploram vulnerabilidades em sistemas e redes existem diversos grupos distintos. Estes grupos são frequentemente descritos pela tonalidade do chapéu que eles usam quando realizam suas investigações de segurança e esta tonalidade é o indicativo de sua intenção.

O *hacker de chapéu branco* é aquele que testa redes e sistemas para examinar seu desempenho e determinam o quanto vulneráveis elas são à uma intrusão. Normalmente, hackers de chapéu branco invadem o próprio sistema ou sistemas de um cliente que especificamente os contratou para este propósito de auditar a segurança. Pesquisadores acadêmicos e consultores de segurança profissionais são dois exemplos de hackers de chapéu branco.

Um *hacker de chapéu preto* é o sinônimo de um cracker. Em geral, cracker são menos focados em programação e no lado acadêmico de invadir sistemas. Eles muitas vezes confiam em programas de invasão e exploram vulnerabilidades conhecidas em sistemas para revelar informações sensíveis para ganho pessoal ou causar danos no sistema alvo ou rede.

O *hacker de chapéu cinza*, por outro lado, possui as habilidades e intenções de um hacker de chapéu branco na maioria das situações mas usa seu conhecimento para propósitos menos nobres em certas ocasiões. Um hacker de chapéu cinza pode ser reconhecido como um hacker de chapéu branco que veste o chapéu preto as vezes para realizar seus próprios planos.

Hackers de chapéu cinza tipicamente concordam com outra forma de ética hacker, que diz que é aceitável invadir sistemas desde que o hacker não cometa roubo ou brechas de confidencialidade. Alguns discutem entretanto que o ato de invadir um sistema não é propriamente ético.

Independente da intenção do invasor, é importante conhecer as fraquezas que um cracker pode querer explorar. O restante deste capítulo é focado neste assunto.

1.3.2. Ameaças à Segurança de Rede

Procedimentos mal feitos na cofniguração dos seguintes aspectos da rede podem aumentar o risco de ataque.

1.3.2.1. Arquiteturas Inseguras

Uma rede mal configurada é o ponto de entrada para usuários não autorizados. Deixar uma rede local aberta, baseada em confiança, vulnerável à internet altamente insegura é tanto como deixar a porta entreaberta em um bairro de alta criminalidade — nada pode acontecer por um certo tempo, mas *eventualmente* alguém irá explorar a oportunidade.

1.3.2.1.1. Redes de Transmissão

Administradores de Sistemas frequentemente erram na percepção da importância do hardware de rede em seus esquemas de segurança. Hardware simples como hubs e roteadores dependem de transmissão ou princípios fora do switch; que significa, toda vez que um nó transmite dados pela rede para um nó recipiente, o hub ou o roteador envia pacotes de dados até que o nó recipiente receba e

processe os dados. Este método é o mais vulnerável para falsificação de endereços do address resolution protocol (*ARP*) ou media access control (*MAC*) por ambos invasores externos e usuários não autorizados no host local.

1.3.2.1.2. Servidores Centralizados

Outra armadilha de rede potencial é o uso de computação centralizada. Uma medida comum para cortar custos em muitas empresas é consolidar todos os serviços a uma única máquina poderosa. Isto pode ser conveniente já que é mais fácil de gerenciar e custa consideravelmente menos do que configurações de servidores múltiplos. Entretanto, um servidor centralizado apresenta um único ponto de falha na rede. Se o servidor central estiver comprometido, pode tornar a rede completamente sem uso ou pior, inclinado à manipulação de dados ou roubo. Nestas situações, um servidor central se torna uma porta aberta que permite acesso à rede inteira.

1.3.3. Ameaças à Segurança do Servidor

A segurança do servidor é tão importante quanto a segurança de rede porque os servidores muitas vezes possuem uma grande parcela de informações vitais de uma organização. Se um servidor estiver comprometido, todo o seu conteúdo pode se tornar disponível para o cracker roubar ou manipular à vontade. As seções seguintes detalham alguns dos problemas principais.

1.3.3.1. Serviços não usados e Portas Abertas

A instalação completa do Red Hat Enterprise Linux 6 possui mais de 1000 aplicativos e pacotes de bibliotecas. Entretanto, a maioria dos administradores de servidores optam por não instalar todos os pacotes da distribuição, preferindo em vez disso instalar os pacotes básicos, incluindo diversas aplicações do servidor.

Uma ocorrência comum entre administradores de sistemas é instalar o sistema operacional sem prestar atenção a quais programas estão atualmente sendo instalados. Isso pode ser problemático porque serviços desnecessários podem ser instalados, configurados com definições padrões e possivelmente serem ativados. Isto pode fazer com que serviços não requeridos como Telnet, DHCP ou DNS, sejam executados no servidor ou estação de trabalho sem o administrador perceber, podendo causar tráfego não requerido no servidor ou mesmo, um caminho potencial ao sistema para crackers. Consulte a Seção 2.2, “Segurança do Servidor” para informações sobre fechar portas e desativar serviços não usados.

1.3.3.2. Serviços sem Correção

A maioria dos aplicativos que estão incluídos em uma instalação padrão são software estáveis, completamente testados. Estando em uso em ambientes de produção por muitos anos, seus códigos têm sido completamente refinados e muitos bugs foram encontrados e corrigidos.

Entretanto, não existe um software perfeito e existe sempre espaço para mais refinamento. Além disso, um software novo muitas vezes não é testado tão rigorosamente como se pode esperar, pelo motivo de sua chegada recente aos ambientes de produção ou porque pode não ser tão popular quanto outros softwares de servidor.

Desenvolvedores e administradores de sistemas muitas vezes encontram bugs exploráveis em aplicações de servidor e publicam a informação em websites de registro de bugs e relacionados à segurança como a lista de emails Bugtraq (http://www.securityfocus.com) ou o web site Computer Emergency Response Team (CERT) (http://www.cert.org). Apesar destes mecanismos serem uma maneira efetiva de alertar a comunidade sobre vulnerabilidades de segurança, é uma decisão dos administradores do sistema de consertar seus sistemas prontamente. Isto é particularmente verdade que crackers possuem acesso a estes mesmos serviços de registro de vulnerabilidades e usarão a

informação para invadir sistemas sem correção sempre que puderem. Uma boa administração de sistema requer vigilância, registro de bugs constante e manutenção do sistema apropriada para garantir um ambiente de computação mais seguro.

Consulte a Seção 1.5, “Atualizações de Segurança” para mais informações sobre manter um sistema atualizado.

1.3.3.3. Administração Desatenta

Administradores que falham ao corrigir seus sistemas são uma das maiores ameaças à segurança dos servidores. De acordo com o *SysAdmin, Audit, Network, Security Institute* (*SANS*), a causa primária de vulnerabilidade de segurança dos computadores é "atribuir pessoas destreinadas para manter a segurança e não fornecer treinamento ou tempo suficiente para fazer o trabalho."[10] Isto se aplica tanto para administradores inexperientes quanto a administradores confiantes ou desmotivados.

Alguns administradores erram em não corrigir seus servidores e estações de trabalho, enquanto outros erram por não verificar as mensagens de log do kernel do sistema ou tráfego de rede. Outro erro comum é quando senhas padrões ou chaves para serviços não são alteradas. Por exemplo, alguns bancos de dados possuem senhas de administração padrões porque os desenvolvedores do sistema presumem que os administradores do sistema mudarão essas senhas imediatamente após a instalação. Se um administrador de banco de dados não mudar esta senha, mesmo um cracker inexperiente pode usar a senha padrão totalmente conhecida para ganhar privilégios administrativos ao banco de dados. Este são apenas alguns poucos exemplos de como uma administração desatenta pode levar ao comprometimento de servidores.

1.3.3.4. Serviços Essencialmente Inseguros

Mesmo as mais vigilantes organizações podem ser vítimas às vulnerabilidades se os serviços de rede que escolheram são inerentemente inseguros. Por exemplo, existem muitos serviços desenvolvidos que supõem estar sob redes confiáveis; entretanto, esta suposição termina tão logo quando o serviço se torna disponível na internet — que é inerentemente não confiável.

Uma categoria de serviços de rede inseguros são aqueles que requerem nomes de usuários e senha sem criptografia para autenticação. Telnet e FTP são dois desses tipos de serviços. Se um software de rastreamento de pacotes estiver monitorando o tráfego entre o usuário remoto e o tal serviço, nomes de usuário e senhas podem ser facilmente interceptados.

Naturalmente, tais serviços podem também ser vítimas mais facilmente do que a área da segurança chama de ataque *man-in-the-middle*. Neste tipo de ataque, um cracker redireciona o tráfego de rede enganando o servidor de nomes na rede para apontar para a máquina do cracker ao contrário do servidor correto. Uma vez que alguém abre uma sessão remota no servidor, a máquina do invasor age como um canal, ficando no meio entre o serviço remoto e o usuário capturando informações. Desta maneira, o cracker pode pegar senhas administrativas e dados brutos sem o servidor ou usuário perceberem.

Outra categoria de serviços inseguros incluem sistemas de arquivos de rede e serviços de informação como NFS ou NIS, que são desenvolvidor explicitamente para uso em LAN mas são infelizmente, extendidos para incluir WANs (para usuários remotos). O NFS não possui, por padrão, qualquer autenticação ou mecanismos de segurança configurados para previnir um cracker de montar o compartilhamento NFS e acessar qualquer informação contida nele. O NIS, também, possui informações vitais que devem ser conhecidas por cada computador na rede, incluindo senhas e permissões de arquivos, dentro de um banco de dados ASCII ou DBM (derivado ASCII) em texto puro. Um cracker que ganha acesso a este banco de dados pode então acessar cada conta de usuário na rede incluindo a conta do administrador.

Por padrão, o Red Hat Enterprise Linux é lançado com todos os serviços desativados. Entretanto, já que

administradores muitas vezes são forçados a usar estes serviços, cuidados na configuração é essencial. Consulte a Seção 2.2, “Segurança do Servidor” para mais informações sobre configurar serviços de uma maneira segura.

1.3.4. Ameaças à Estação de Trabalho e Segurança no PC doméstico.

Estações de trabalho e PCs domésticos podem não serem tão inclinados à ataques quanto em redes ou servidores, mas já que frequentemente possuem dados sensíveis, tais como informações de cartões de crédito, eles são alvos de crackers de sistemas. Estações de trabalho podem também ser cooptadas sem o conhecimento do usuário e usadas como máquinas "escravos" em ataques coordenados. Por estas razões, conhecer as vulnerabilidades de uma estação de trabalho pode tirar os usuários a dor de cabeça de reinstalar o sistema operacional, ou pior, se recuperar de roubo de dados.

1.3.4.1. Senhas Ruins

Senhas ruins são uma das maneiras mais fáceis para um invasor ganhar acesso a um sistema. Para mais informações em como evitar essas armadilhas quando criar uma senha, consulte a Seção 2.1.3, “Segurança da Senha”.

1.3.4.2. Aplicações Clientes Vulneráveis

Apesar de um administrador poder ter um servidor totalmente seguro e com correções, isto não significa que usuários remotos estão seguros ao acessa-lo. Por exemplo, se o servidor oferece Telnet ou serviços FTP para uma rede pública, um invasor pode capturar os nomes de usuários e senha em texto puro conforme eles passam pela rede e então usar as informações da conta para acessar a estação de trabalho do usuário remoto.

Mesmo quando usar protocolos seguros, como SSH, um usuário remoto pode estar vulnerável a certos ataques se eles não manterem suas aplicações clientes atualizadas. Por exemplo, clientes v.1 SSH são vulneráveis á um ataque X-forwarding a partir de servidores SSH. Uma vez conectados ao servidor, o invasor pode silenciosamente capturar qualquer digitação e cliques do mouse feitos no cliente sobre a rede. Este problema foi consertado no protocolo SSH v.2, mas é parte do usuário acompanhar quais aplicações possuem tais vulnerabilidades e atualiza-las conforme necessário.

A Seção 2.1, “Segurança da Estação de Trabalho” discute em mais detalhes quais passos os administradores e usuários domésticos devem tomar para limitar a vulnerabilidade das estações de trabalho dos computadores.

1.4. EXPLORAÇÕES COMUNS E ATAQUES

Tabela 1.1, “Explorações Comuns” detalha algumas das explorações mais comuns e pontos de entrada usados por invasores para acessar recursos de rede organizacionais. A chave para estas explorações comuns são explicações de como elas são realizadas e como os administradores podem proteger adequadamente sua rede contra tais ataques.

Tabela 1.1. Explorações Comuns

Exploração	Descrição	Notas

Exploração	Descrição	Notas
Senhas Nulas ou Padrão	Deixar as senhas administrativas em branco ou usar um conjunto de senhas padrões definidas pelo fabricante do produto. Isto é mais comum em hardwares como routers e firewalls, embora alguns serviços aplicados no LInux possam conter senhas de administração padrão (embora o Red Hat Enterprise Linux não é distribuído com eles).	Geralmente associado ao hardware de rede, tais como os equipamentos de routers, firewalls, VPNs e armazenamento de rede anexado (NAS). Comum em muitas legacias de sistemas operacionais, especialmente aqueles que agrupam serviços (como o UNIX e Windows). Os administradores as vezes criam contas de usuários privilegiadas às pressas e deixam a senha em branco, criando um ponto de entrada perfeito para usuários mal-intencionados que descobrem a conta.
Chaves Compartilhadas Padrão	Serviços seguros as vezes empacotam chaves de segurança padrão para o desenvolvimento ou avaliação para propósitos de testes. Se estas chaves são deixadas como estão e colocadas em um ambiente de produção na Internet, *todos* os usuários com as mesmas chaves padrões possuem acesso ao recurso de chave compartilhada e qualquer informação confidencial que ele contenha.	O mais comum em pontos de acesso wireless e equipamentos de servidor seguro pré-configurados.
IP Spoofing	Uma máquina remota que age como um nó em sua rede local, encontra vulnerabilidades em seus servidores e instala um programa de backdoor (porta-dos-fundos) ou um trojan horse para obter controle sob seus recursos de rede.	Spoofing é um tanto difícil, pois ele envolve que o atacante prediga os números da sequência do TCP/IP para coordenar uma conexão aos sistemas alvo, mas diversas ferramentas estão disponíveis para ajudar os invasores a explorar tal vulnerabilidade. Depende do sistema alvo que está executando os serviços (tal como **`rsh`**, **`telnet`**, FTP entre outros) que usam as técnicas de autenticação *sem criptografia*, que não são recomendadas quando comparadas ao PKI ou outras formas de autenticação criptografadas usadas em **`ssh`** ou SSL/TLS.

Exploração	Descrição	Notas
Eavesdropping (Interceptação)	Uma coleta dos dados que passam entre dois nós ativos em uma rede, interceptando a conexão entre os dois nós.	Este tipo de ataque funciona mais com protocolos de transmissão de texto simples como o Telnet, FTP, e transferências de HTTP. Invasores remotos devem ter acesso ao sistema comprometido em uma LAN para realizar tal ataque. Geralmente o atacante usou um ataque ativo (como o IP spoofing ou man-in-the-middle) para comprometer um sistema na LAN. Medidas preventivas incluem serviços com troca de chave criptográfica, senhas de uma vez, ou autenticação criptografada para prevenir que senhas sejam roubadas. Uma criptografia forte durante a transmissão também é recomendada.

Exploração	Descrição	Notas
Vulnerabilidades de Serviços	Um atacante encontra um defeito ou um furo em um serviço executado sob a Internet; através desta vulnerabilidade, o atacante compromete todo o sistema e qualquer dado que ele possa conter, e pode possivelmente comprometer outros sistemas na rede.	Os serviços baseados em HTTP como o CGI são vulneráveis à execução de comando remoto e até mesmo acesso de shell interativo. Mesmo se o serviço HTTP executasse um usuário não privilegiado como "nobody", informações como arquivos de configuração e mapas de rede poderiam ser lidos ou o atacante pode iniciar uma negação de ataque de serviço que drena os recursos do sistema ou o torna indisponível para outros usuários. Os serviços as vezes podem conter vulnerabilidades que passam despercebidas durante o desenvolvimento e teste; estas vulnerabilidades (tais como *buffer overflows*, onde atacantes quebram um sistema usando valores arbitrários que preenchem o buffer de memória de um aplicativo, (dando ao atacante o pedido de comando interativo do qual eles podem executar comandos arbitrários) pode fornecer controle administrativo completo ao atacante. Administradores devem ter certeza de que os serviços não são executados como root, e devem estar atentos às correções e atualizações de erratas para aplicativos de fabricantes ou organizações de segurança como o CERT e CVE.

Exploração	Descrição	Notas
Vulnerabilidades de Aplicativos	Atacantes encontram falhas em desktops e aplicativos de estações de trabalho (tal como clientes de email) e executam códigos arbitrários, implementam trojan horses para comprometimento futuro ou quebra de sistemas. Explorações futuras podem ocorrer se a estação de trabalho comprometida possui privilégios administrativos no resto da rede.	As estações de trabalho e desktops têm mais tendência a serem exploradas pois os funcionários não possuem o conhecimento ou experiência de evitar ou detectar o comprometimento. É crucial informar indivíduos dos riscos que correm quando instalam softwares não autorizados ou abrir anexos de emails não solicitados. Medidas de segurança podem ser implementadas para que clientes de email não abram automaticamente ou executem anexos. Além disso, a atualização automática de software da estação de trabalho via Red Hat Network ou outros serviços de gerenciamento de sistemas podem aliviar a carga da implementação de segurança de máquina em máquina.
Ataques Denial of Service (DoS)	O atacante ou grupo de atacantes coordenam contra uma rede de uma empresa ou recursos de servidor enviando pacotes não autorizados ao host alvo (tanto o servidor, router ou estação de trabalho). Isto força o recurso a se tornar indisponível para usuários legítimos.	O caso de ataque DoS mais reportado nos E.U.A. ocorreu em 2000. Diversos sites do governo e comerciais com alto índice de tráfego se tornaram indisponíveis por um ataque coordenado de inundação de pings, usando diversos sistemas comprometidos com conexões de banda larga agindo como *zombies*, ou nós de broadcast redirecionados. Pacotes fonte são geralmente forjados (assim como retransmitidos), tornado a investigação da verdadeira fonte de ataque um tanto difícil. Avanços nos filtros de ingresso (IETF rfc2267) usando o **`iptables`** e Network Intrusion Detection Systems como o **`snort`** assistem administradores no rastreamento e previnem ataques distribuídos do DoS.

1.5. ATUALIZAÇÕES DE SEGURANÇA

Conforme as vulnerabilidades de segurança são descobertas, o software afetado deve ser atualizado para limitar quaisquer riscos potenciais de segurança. Se o software é parte de um pacote dentro de uma distribuição Red Hat Enterprise Linux que é suportada atualmente, a Red Hat está comprometida a

lançar atualizações de pacotes que consertam as vulnerabilidades assim que possível. Muitas vezes, anúncios sobre uma exploração de segurança são acompanhados de uma correção (ou código fonte que conserta o problema). Esta correção é então aplicada ao pacote Red Hat Enterprise Linux e testada e lançada como uma errata de atualização. Entretanto, se um anúncio não inclui uma correção, um desenvolvedor primeiro trabalha com o mantenedor do software para consertar o problema. Uma o problema é consertado, o pacote é testado e lançado como uma errata de atualização.

Se uma errata de atualização é lançada para um software usado em seus sistema, é altamente recomendado que você atualize os pacotes afetados assim que possível para minimizar o período de tempo que seu sistema está potencialmente vulnerável.

1.5.1. Atualizando Pacotes

Quando atualizar o software em um sistema, é importante baixar a atualização de uma fonte confiável. Um invasor pode facilmente reconstruir um pacote com o mesmo número de versão do pacote suposto para consertar o problema mas com uma exploração de segurança diferente e lança-lo na internet. Se isso acontecer, use medidas de segurança como verificar arquivos contra os RPMs originais não detecta as explorações. Assim, é muito importante de somente baixar os RPMs de fontes confiáveis, como a Red Hat e checar a assinatura do pacote para verificar sua integridade.

NOTA

O Red Hat Enterprise Linux inclui um ícone conveniente no painel que mostra alertas vísiveis quando há uma atualização disponível.

1.5.2. Verificando Pacotes Assinados

Todos os pacotes do Red Hat Enterprise Linux são assinados com a chave *GPG* da Red Hat. O GPG significa GNU Privacy Guard, ou GnuPG, um pacote de software livre usado para garantir a autenticidade dos arquivos distribuídos. Por exemplo, uma chave privada (chave secreta) trava o pacote enquanto a chave pública destrava e verifica o pacote. Se a chave pública distribuída pelo Red Hat Enterprise Linux não corresponder com a chave privada durante a verificação do RPM, o pacote pode ter sido alterado e portanto não pode ser confiável.

O utilitário RPM dentro do Red Hat Enterprise Linux 6 automaticamente tenta verificar a assinatura GPG de um pacote RPM antes de instala-lo. Se a chave GPG da Red Hat não está instalada, instale-a a partir de uma locação estática e segura, como o CD-ROM ou DVD de instalação da Red Hat.

Assumindo que o disco é montado no **/mnt/cdrom**, use o seguinte comando para importa-lo ao *keyring* (um banco de dados de chaves confiáveis no sistema):

```
rpm --import /mnt/cdrom/RPM-GPG-KEY
```

Para exibir uma lista de todas as chaves instaladas para verificação do RPM, execute o seguinte comando:

```
rpm -qa gpg-pubkey*
```

O resultado será similar ao seguinte:

```
gpg-pubkey-db42a60e-37ea5438
```

Para exibir detalhes sobre uma chave específica, use o comando **rpm -qi** seguido do resultado do comando anterior, como neste exemplo:

```
rpm -qi gpg-pubkey-db42a60e-37ea5438
```

É extremamente importante verificar a assinatura dos arquivos RPM antes de instala-los para garantir que eles não foram alterados da fonte original dos pacotes. Para verificar todos os pacotes baixados de uma vez, use o seguinte comando:

```
rpm -K /tmp/updates/*.rpm
```

Para cada pacote, se a chave GPG é verifica com sucesso, o comando retorna **gpg OK**. Se caso não, tenha certeza que está usando a chave pública Red Hat correta e também verificar a fonte do conteúdo. Pacotes que não passam verificações GPG não devem ser instalados, já que podem ter sido alterados por um terceiro.

Depois de verificar a chave GPG e baixar todos os pacotes associados com o relatório de errada, instale os pacotes como root no prompt do shell.

1.5.3. Instalando Pacotes Assinados

A instalação para a maioria dos pacotes podem ser feitas seguramente (exceto pacotes do kernel) emitindo o seguinte comando:

```
rpm -Uvh /tmp/updates/*.rpm
```

Para os pacotes do kernel, use o seguinte comando:

```
rpm -ivh /tmp/updates/<kernel-package>
```

Substitua o *<kernel-package>* no exemplo anterior com o nome do RPM kernal.

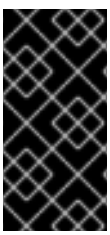
Uma vez que a máquina foi seguramente reinicializada usando o novo kernel, o kernel antigo pode ser removido usando o seguinte comando:

```
rpm -e <old-kernel-package>
```

Substitua o *<old-kernel-package>* do exemplo anterior com o nome do RPM do kernel antigo.

NOTA

Não é um requerimento que o kernel antigo seja removido. O carregador de boot padrão, o GRUB, permite múltiplos kernels serem instalados, e então escolhidos de um menu no momento da inicialização.



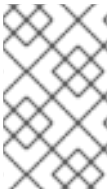
IMPORTANTE

Antes de instalar quaisquer erratas de segurança, tenha certeza de ler as instruções especiais contidas no relatório da errata e as execute de acordo. Consulte a Seção 1.5.4, “Aplicando as Mudanças” para instruções gerais sobre aplicar as mudanças feitas por uma atualização de errata.

1.5.4. Aplicando as Mudanças

Depois de baixar e instalar erratas de segurança e atualizações, é importante parar o uso do software

antigo e iniciar o uso do novo software. Como isto é feito, depende do tipo de software que foi atualizado. A seguinte lista relaciona as categorias gerais dos softwares e fornece instruções para usar as versões atualizadas depois de uma atualização de pacote.



NOTA

No geral, reinicializar o sistema é a maneira mais certa para garantir que a última versão do pacote de software é usada, esta opção não é sempre requerida, ou disponível para o administrador do sistema.

Aplicativos

Aplicativos do espaço do usuário são quaisquer programas que podem ser iniciados por um usuário do sistema. Tipicamente, tais aplicativos são usados somente quando um usuário, script ou tarefa automatizada os rodam e estes não persistem por longos períodos de tempo.

Uma vez que determinado aplicativo de espaço de usuário é atualizado, pare quaisquer instâncias do aplicativo no sistema e rode o programa de novo para usar a versão atualizada.

Kernel

O kernel é o componente de software principal do sistema operacional Red Hat Enterprise Linux. Ele gerencia o acesso à memória, ao processador e aos periféricos e também a todas as tarefas agendadas.

Por causa de seu papel central, o kernel não pode ser reiniciado sem também parar o computador. Portanto, uma versão atualizada do kernel não pode ser usada até que o sistema seja reinicializado.

Bibliotecas Compartilhadas

Bibliotecas compartilhadas são unidades de códigos, como o **glibc**, que são usados por um número de aplicações e serviços. Aplicações utilizando uma biblioteca compartilhada tipicamente carrega o código compartilhado quando a aplicação é inicializada, então quaisquer aplicações usando a biblioteca atualizada deve ser parada e reiniciada.

Para determinar quais aplicativos em execução se ligam a uma determinada biblioteca, use o comando **lsof** como no exemplo seguinte:

```
lsof /lib/libwrap.so*
```

Este comando retorna uma lista de todas os programas que usam os TCP Wrappers para controle de acesso ao host. Portanto, qualquer programa listado deve ser parado e reiniciado se o pacote **tcp_wrappers** estiver atualizado.

Serviços SysV

Serviços SysV são programas de servidores persistentes iniciados durante o processo de boot. Exemplos de serviços SysV incluem o **sshd**, **vsftpd**, e **xinetd**.

Pela razão que estes programas normalmente persistem na memória pelo tempo que a máquina é inicializada, cada serviço de atualização SysV deve ser parado e reiniciado depois que o pacote é atualizado. Isto pode ser feito usando **Services Configuration Tool** (Ferramenta de Configuração de Serviços) ou se autenticando no shell root e digitando o comando **/sbin/service** conforme no exemplo seguinte:

```
/sbin/service <service-name> restart
```

No exemplo anterior, substitua o *<service-name>* com o nome do serviço, como o **sshd**.

Serviços `xinetd`

Serviços controlados pelo super serviço **`xinetd`** somente roda quando há uma conexão ativa. Exemplos de serviços controlados pelo **`xinetd`** incluem o Telnet, IMAP e POP3.

Pela razão que novas instâncias destes serviços são iniciadas pelo **`xinetd`** cada vez que um novo pedido é recebido, as conexões que ocorrem depois de uma atualização são manuseadas pelo software atualizado. Entretanto se existem conexões ativas no momento que o serviço controlado pelo **`xinetd`** é atualizado, eles são atendidos pela versão antiga do software.

Para terminar as instâncias antigas de um determinado serviço controlado pelo **`xinetd`**, atualize o pacote para o serviço e então pare todos os processos atualmente em execução. Para determinar se o processo está rodando, use o comando **`ps`** e então o **`kill`** ou **`killall`** para parar instâncias atuais do serviço.

Por exemplo, se uma errata de segurança dos pacotes **`imap`** é lançada, atualize os pacotes e então digite o seguinte comando como root no prompt do shell:

```
ps aux | grep imap
```

Este comando retorna todas sessões ativas do IMAP. Sessões individuais podem então ser terminadas digitando o seguinte comando:

```
kill <PID>
```

Se isto falhar para terminar a sessão, use o seguinte comando então:

```
kill -9 <PID>
```

Nos exemplos anteriores, substitua o *<PID>* com o número de identificação do processo (encontrado na segunda coluna do comando **`ps`**) para uma sessão de IMAP.

Para terminar todas as sessões IMAP ativas, digite o seguinte comando:

```
killall imapd
```

---

[1] http://law.jrank.org/pages/3791/Kevin-Mitnick-Case-1999.html

[2] http://www.livinginternet.com/i/ia_hackers_levin.htm

[3] http://www.theregister.co.uk/2007/05/04/txj_nonfeasance/

[4] http://www.fudzilla.com/content/view/7847/1/

[5] http://www.internetworldstats.com/stats.htm

[6] http://www.cert.org

[7] http://www.cert.org/stats/cert_stats.html

[8] http://news.cnet.com/Computer-crime-costs-67-billion,-FBI-says/2100-7349_3-6028946.html

[9] http://www.cio.com/article/504837/Why_Security_Matters_Now

[10] http://www.sans.org/resources/errors.php

CAPÍTULO 2. PROTEGENDO SUA REDE

2.1. SEGURANÇA DA ESTAÇÃO DE TRABALHO

Proteger um ambiente Linux se inicia com a estação de trabalho. Seja trancando uma máquina pessoal ou protegendo um sistema corporativo, uma política de segurança bem feita começa com o computador individual. Uma rede de computadores é tão segura quanto seu nó mais vulnerável.

2.1.1. Avaliando a Segurança da Estação de Trabalho

Quando avaliar a segurança de uma estação de trabalho do Red Hat Enterprise Linux, considere o seguinte:

- *Segurança da BIOS e do carregador de Boot*— É possível que um usuário não autorizado acesse a máquina fisicamente e inicialize como um usuário único ou modo de recuperação sem uma senha?
- *Segurança da Senha* — Quão seguros estão as senhas das contas de usuários na máquina?
- *Controles Administrativos* — Quem possuirá uma conta no sistema e quanto controle administrativo possuirão?
- *Serviços de Redes Disponíveis* — Quais serviços estão escutando por pedidos da rede e eles deveriam estar rodando?
- *Firewalls Pessoais* — Qual tipo de firewall seria necessário?
- *Ferramentas de Comunicação com Segurança Avançada* — Quais ferramentas devem ser usadas para se comunicar entre estações de trabalho e quais deveriam ser evitadas?

2.1.2. Segurança da BIOS e do Carregador de Boot

A proteção por senha da BIOS (ou equivalente à BIOS) e do carregador de boot podem prevenir que usuários não autorizados que possuem acesso físico aos sistemas de realizarem o boot usando mídias removíveis ou obter privilégios root através do modo de usuário único. As medidas de segurança que você deve tomar para se proteger de tais ataques dependem tanto da sensibilidade da informação na estação de trabalho e da localização da máquina.

Por exemplo, se uma máquina é usada em uma exposição e não possui nenhuma informação sensível, então pode não ser crítico prevenir tais ataques. Entretanto, se um notebook de um empregado com chaves pessoais SSH e sem encriptação da rede da empresa é deixado desacompanhado na mesma exposição, isso poderia ser uma brecha de segurança maior com ramificações em toda a empresa.

Se a estação de trabalho está localizada em um lugar onde somente pessoas autorizadas ou de confiança possuem acesso, então proteger a BIOS ou o carregador do boot pode não ser necessário.

2.1.2.1. Senhas da BIOS

As duas razões primárias para proteger a BIOS de um computador com senha são[11]:

1. *Prevenindo Mudanças às Configurações da BIOS* — Se um invasor possui acesso à BIOS, ele pode configura-la para fazer o boot de um disquete ou CD-ROM. Sendo possível entrar no modo de recuperação ou modo de usuário único, que permite iniciar processos arbitrários no sistema ou copiar dados sensíveis.

2. *Prevenindo Inicialização do Sistema* — Algumas BIOS permitem proteção por senha do processo de boot. Quando ativados, um invasor é forçado a entrar com uma senha antes que a BIOS inicie o carregador de boot.

Porque os métodos para configurar a senha da BIOS podem variar entre os fabricantes de computador, consulte o manual do computador para instruções especificas.

Se você esquecer a senha da BIOS, ela pode ser zerada com os jumpers na placa mãe ou retirando a bateria do CMOS. Por esta razão, é uma boa prática trancar a caixa do computador se possível. Entretanto, consulte o manual do computador ou da placa mãe antes de tentar desconectar a bateria do CMOS.

2.1.2.1.1. Protegendo Plataformas que não são X86

Outras arquiteturas usam programas diferentes para realizar tarefas de baixo nível mais ou menos equivalentes à essas das BIOS em sistemas x86. Por exemplo, computadores Intel® Itanium™ usam a *Extensible Firmware Interface* (*EFI*) shell.

Para instruções sobre proteger com senha programas como a BIOS em outras arquiteturas, consulte as instruções do fabricante.

2.1.2.2. Senhas do Carregador de Boot

As razões primárias para proteger com senha um carregador de boot de Linux são as seguintes:

1. *Impedir Acesso ao Modo de Usuário Único* — Se invasores podem inicializar o sistema no modo de usuário único, eles são logados automaticamente como root sem serem questionados pela senha root.

2. *Impedir Acesso ao Console GRUB* — Se a máquina usa o GRUB como seu carregador de boot, um invasor pode usar a interface do editor GRUB para mudar sua configuração ou pegar informações usando o comando **`cat`**.

3. *Impedir Acesso à Sistemas Operacionais Inseguros* — Se o sistema possui sistema de boot duplo, um invasor pode selecionar um sistema operacional no momento do boot (por exemplo, o DOS), que ignora controles de acesso e permissões de arquivos.

O Red Hat Enterprise Linux 6 é lançado com o carregador de boot do GRUB na plataforma x86. Para uma visão detalhada do GRUB, consulte o Guia de Instalação da Red Hat.

2.1.2.2.1. Protegendo o GRUB com senha

Você pode configurar o GRUB para tratar os dois primeiros problemas listados na Seção 2.1.2.2, “Senhas do Carregador de Boot” adicionando uma senha direcionada ao seu arquivo de configuração. Para fazer isso, primeiro adicione uma senha forte, abra o shell, autentique-se como root e então digite o seguinte comando:

```
/sbin/grub-md5-crypt
```

Quando questionado, digite a senha do GRUB e pressione **`Enter`**. Isto retorna um hash MD5 da senha.

Depois, edite o arquivo de configuração do GRUB **`/boot/grub/grub.conf`**. Abra o arquivo e abaixo da linha **`timeout`** na seção principal do documento, adicione a seguinte linha:

```
password --md5 <password-hash>
```

Substitua *<password-hash>* com o valor retornado pelo **/sbin/grub-md5-crypt**[12].

A próxima vez que o sistema inicializar, o menu do GRUB impede o acesso ao editor ou interface de comando sem primeiro pressionar **p** seguido pela senha do GRUB.

Infelizmente, esta solução não previne que um invasor inicialize por um sistema operacional não seguro em um ambiente de boot duplo. Para isto, uma parte diferente do arquivo **/boot/grub/grub.conf** deve ser editada.

Busque pela linha **title** do sistema operacional que você quer proteger e adicione uma linha com a diretiva **lock** imediatamente embaixo dela.

Para o sistema DOS, a estrofe deve iniciar similarmente ao seguinte:

```
title DOS lock
```

ATENÇÃO

A linha **password** deve estar presente na seção principal do arquivo **/boot/grub/grub.conf** para este método para funcionar propriamente. Caso contrário, um invasor pode acessar a interface do editor GRUB e remover a linha bloqueada.

Para criar uma senha diferente para um kernel em particular ou sistema operacional, adicione a linha **lock** à estrofe, seguida pela linha da senha.

Cada estrofe protegida com uma senha única deve iniciar com as linhas similares ao exemplo seguinte:

```
title DOS lock password --md5 <password-hash>
```

2.1.3. Segurança da Senha

As senhas são o método primário que o Red Hat Enterprise Linux usa para verificar a identidade do usuário. Isto é porque a segurança de senha é tão importante para a proteção do usuário, da estação de trabalho e da rede.

Por motivos de segurança, o programa de instalação configura o sistema para usar o *Secure Hash Algorithm 512* (*SHA512*) e senhas shadow. É altamente recomendado que você não altere essas configurações.

Se senhas shadow não são selecionadas durante a instalação, todas as senhas são armazenadas como um hash de uma via no arquivo de leitura pública **/etc/passwd**, que faz o sistema vulnerável à ataques de quebra de senhas offline. Se um invasor pode ter acesso à máquina como um usuário normal, ele pode copiar o arquivo **/etc/passwd** para sua própria máquina e rodar quaisquer programas de quebra de senha. Se houver uma senha insegura no arquivo, é somente um questão de tempo até que o invasor a descubra.

Senhas shadow eliminam este tipo de ataque armazenando as senhas hash no arquivo **/etc/shadow**, que pode ser lido somente pelo usuário root.

Isto força um invasor em potencial tentar quebrar senhas remotamente, autenticando-se em um serviço de rede na máquina, como SSH ou FTP. Este tipo de ataque de força bruta é muito mais lento e deixa rastros óbvios, já que centenas de tentativas de login são gravadas nos arquivos do sistema. É claro que se o invasor iniciar um ataque no meio da noite em um sistema com senhas fracas, o invasor pode ter ganhado o acesso antes do amanhecer e editado os arquivos de log para encobrir seus rastros.

Além das considerações de formato e armazenamento existe a questão do conteúdo. A coisa mais importante que um usuário pode fazer para proteger sua conta contra um ataque de quebras de senha é criar uma senha forte.

2.1.3.1. Criando Senhas Fortes

Quando criar uma senha segura, é uma boa idéia seguir essas diretrizes:

- *Não Use Somente Palavras ou Números* — Nunca use somente números ou palavras em uma senha.

  Alguns exemplos inseguros incluem o seguinte:

  - 8675309
  - juan
  - hackme

- *Não Use Palavras Reconhecíveis* — Palavras como nomes próprios, palavras de dicionário ou mesmo termos de programas de televisão ou novelas devem ser evitados, mesmo se finalizados com números.

  Alguns exemplos inseguros incluem o seguinte:

  - john1
  - DS-9
  - mentat123

- *Não Use Palavras em Línguas Estrangeiras* — Programas de quebra de senha muitas vezes tentam listas de palavras que incluem dicionários de muitas línguas. Contar com línguas estrangeiras para senhas não é seguro.

  Alguns exemplos inseguros incluem o seguinte:

  - cheguevara
  - bienvenido1
  - 1dumbKopf

- *Não Use Terminologia Hacker* — Se você acha que é elite porque você usa terminologia hacker — também conhecida como a escrita l337 (LEET) — em sua senha, pense novamente. Muitas listas de palavras incluem a escrita LEET.

  Alguns exemplos inseguros incluem o seguinte:

  - H4X0R

- 1337

- *Não Use Informações Pessoais* — Evite usar qualquer informação pessoal em suas senhas. Se o invasor conhece sua identidade, a tarefa em adivinhar sua senha se torna mais fácil. A seguir está uma lista de tipos de informações a serem evitadas quando criar uma senha:

  Alguns exemplos inseguros incluem o seguinte:

  - Seu Nome
  - Nomes de animais de estimação
  - Os nomes dos membros da família
  - Qualquer data de aniversário
  - Seu número de telefone ou código postal

- *Não Inverta Palavras Reconhecíveis* — Verificadores de senhas bons sempre invertem as palavras comuns, então inverter uma senha fraca não a faz mais segura.

  Alguns exemplos inseguros incluem o seguinte:

  - R0X4H
  - nauj
  - 9-DS

- *Não Escreva Sua Senha* — Nunca guarde sua senha em papel. É mais seguro memoriza-la.

- *Não Use a Mesma Senha para Todas as Máquinas* — É importante fazer senhas separadas para cada máquina. Desta maneira se um sistema é comprometido, todas suas máquinas não estarão imeditatamente em risco.

As seguintes diretrizes lhe ajudarão a criar uma senha forte:

- *Faça a Senha com no Mínimo 8 Dígitos* — Quanto mais longa a senha, melhor. Se usar senhas MD5, ela deve ser de 15 ou mais dígitos . Com senha DES, use a extensão máxima (oito caracteres).

- *Misture Maiúsculas e Minúsculas* — O Red Hat Enterprise Linux diferencia maiúsculas de minúsculas, então misture as letras para aumentar a força da senha.

- *Misture Letras e Números* — Adicionar números às senhas, especialmente quando adicionados no meio (não só no início ou no final), pode aumentar a força da senha.

- *Inclua Caracteres Não Alfa Numéricos* — Caracteres especiais como &, $, e > podem melhorar muito a força de uma senha (isto não é possível se usar senhas DES).

- *Escolha uma Senha que Você pode se Lembrar* — A melhor senha do mundo não faz nada se você não lembra-la; use acrônimos ou outros dispositivos mnemônicos para ajudar a memorizar senhas.

Com estas regras, pode parecer difícil criar uma senha que atenda todos esses critérios de uma senha eficiente e evitar as peculiaridades de uma ruim. Felizmente, existem alguns passos que você pode tomar para gerar uma senha segura, fácil de lembrar.

2.1.3.1.1. Metodologia para Criação de Senhas Seguras

Existem muitos métodos que pessoas usam para criar senhas seguras. Um dos métodos mais populares envolvem acrônimos. Por exemplo:

- Pense em uma frase fácil de se lembrar, como esse em inglês:

  "over the river and through the woods, to grandmother's house we go."

- Depois, transforme a frase em um acrônimo (incluindo a pontuação).

  otrattw,tghwg.

- Adicione complexidade substituindo números e símbolos por letras no acrônimo. Por exemplo, substitua **7** pelo **t** e o símbolo (@) pelo **a**:

  o7r@77w,7ghwg.

- Adicione mais complexidade colocando em maiúsculo pelo menos uma letra, tal como **H**.

  o7r@77w,7gHwg.

- *Finalmente, não use o exemplo da senha acima em qualquer sistema, nunca.*

Enquanto criar senhas seguras é imperativo, gerencia-las propriamente também é importante, especialmente para administradores de sistemas dentro de grandes organizações.

2.1.3.2. Criando Senhas de Usuários Dentro de Uma Organização

Se uma organização possui um grande número de usuários, os administradores de sistema possuem duas opções básicas disponíveis para forçar o uso de boas senhas. Eles podem criar senhas para o usuário ou eles podem deixar os usuários criarem suas próprias senhas e verificando que as senhas são de qualidade aceitável.

Criar senhas para os usuários garante que as senhas sejam boas, mas se torna uma tarefa assustadora conforme a organização cresce. Também aumenta o risco dos usuários escreverem suas senhas em papel.

Por estas razões, a maioria dos administradores de sistemas preferem que os usuários criem suas próprias senhas, mas ativamente verificar que as senhas sejam boas e em alguns casos, forçar os usuários a muda-las periodicamente através de expiração de senha.

2.1.3.2.1. Forçando Senhas Fortes

Para proteger a rede de intrusões é uma boa idéia para os administradores de sistema verificar que as senhas usadas dentro de uma organização sejam fortes. Quando os usuários são perguntados para criar ou mudar senhas, eles podem usar a aplicação de linha de comando **passwd**, que é o *Pluggable Authentication Modules* (*PAM*) checando se a senha é muito curta ou de outra maneira fácil de quebrar. Esta verificação é realizada usando o modulo PAM **pam_cracklib.so**. Já que o PAM é personalizável, é possível adicionar mais de um verificador de integridade de senhas, como **pam_passwdqc** (disponível em http://www.openwall.com/passwdqc/) ou escrever um módulo novo. Para uma lista de módulos PAM disponíveis, consulte o http://www.kernel.org/pub/linux/libs/pam/modules.html. Para mais informações sobre o PAM, consulte *Gerenciando Sign-On Únicos e Cartões Smart*.

A verificação de senha que é realizada no momento de sua criação não descobre senhas ruins tão efetivamente quanto rodar um programa de quebra de senhas.

Muitas programas de quebra de senhas estão disponíveis e que rodam no Red Hat Enterprise Linux, apesar de que nenhum está no pacote do sistema operacional. Abaixo está uma lista breve de alguns dos programas mais populares de quebra de senha:

- ***John The Ripper*** — Um rápido de flexível programa de quebra de senhas. Permite o uso de múltiplas listas de palavras e é capaz de quebrar senhas por força bruta. Está disponível online em http://www.openwall.com/john/.

- ***Crack*** — Talvez o mais conhecido programa de quebra de senhas, o**Crack** é também muito rápido, embora não tão fácil de usar como o **John The Ripper**. Pode ser encontrado em http://www.crypticide.com/alecm/security/crack/c50-faq.html.

- ***Slurpie*** — O **Slurpie** é similar ao **John The Ripper** e o **Crack**, mas é desenhado para rodar em múltiplos computadores simultaneamente, criando um ataque de quebra de senhas distribuído. Ele pode ser encontrado online junto com uma série de outras ferramentas de avaliação de ataque à segurança em http://www.ussrback.com/distributed.htm.

ATENÇÃO

Sempre obtenha uma autorização escrita antes de tentar quebrar senhas dentro de uma organização.

2.1.3.2.2. Frases Secretas

Frases secretas e senhas são um pilar na segurança na maioria dos sistemas de hoje. Infelizmente, tais técnicas como biometria e autenticação de dois fatores ainda não se tornaram o caminho principal em muitos sistemas. Se senhas serão usadas para proteger um sistema, então o uso de frases secretas devem ser consideradas. Frases secretas são mais longas do que senhas e fornecem melhor proteção do que uma senha mesmo quando implementadas sem caracteres padrões como números e símbolos.

2.1.3.2.3. Expiração de Senha

Expiração de senha é uma outra técnica usada por administradores de sistemas para se defender contra senhas ruins dentro de uma organização. Expiração de senha significa que depois de um especificado período (normalmente 90 dias), o usuário é questionado para criar uma nova senha. A teoria por trás disto é que se um usuário é forçado a mudar sua senha periodicamente, uma senha descoberta é somente útil ao invasor por um período de tempo limitado. A desvantagem da expiração de senha, entretanto, é que usuários são mais inclinados a escrever suas senhas no papel.

Existem dois programas primários usados para especificar a expiração no Red Hat Enterprise Linux: o comando **chage** ou a aplicação **User Manager** (**system-config-users**).

A opção **-M** do comando **chage** especifica o número máximo de dias que a senha é válida. Por exemplo, para definir que a senha de um usuário expire em 90 dias, use o seguinte comando:

```
chage -M 90 <username>
```

No comando acima, substitua o *<username>* com o nome do usuário. Para desativar a expiração de senha, é normal usar um valor de **99999** depois da opção **-M** (isto é equivale a mais ou menos 273 anos).

Você pode também usar o comando **chage** em modo interativo para modificar múltiplas expirações de senhas e detalhes de conta. Use o comando seguinte para entrar no modo interativo:

```
chage <username>
```

O exemplo seguinte é exemplo de sessão interativa usando este comando:

```
[root@myServer ~]# chage davido
Changing the aging information for davido
Enter the new value, or press ENTER for the default
Minimum Password Age [0]: 10
Maximum Password Age [99999]: 90
Last Password Change (YYYY-MM-DD) [2006-08-18]:
Password Expiration Warning [7]:
Password Inactive [-1]:
Account Expiration Date (YYYY-MM-DD) [1969-12-31]:
[root@myServer ~]#
```

Consulte a página man sobre o chage para mais informações sobre as opções disponíveis.

Você pode também usar a aplicação gráfica **User Manager** (Gerenciador de Usuários) para criar políticas de expiração de senhas, conforme a seguir. Nota: você precisa de privilégios de Administrador para realizar este procedimento.

1. Clique no menu **System** (Sistema) no painel, depois em **Administration** (Administração) e então clique em **Users and Groups** (Usuários e Grupos) para exibir o Gerenciador de Usuários. Alternativamente, digite o comando **system-config-users** no shell.

2. Clique na aba **Users** (Usuários) e selecione o usuário requerido na lista de usuários.

3. Clique em **Properties** (Propriedades) na barra de ferramentas para exibir as caixa de diálogo das Propriedades do Usuário (ou escolha **Properties** (Propriedades) no menu **File** (Arquivo)).

4. Clique na aba **Password Info** (Informações de Senha) e marque a caixa de verificação **Enable password expiration** (Ativar expiração de senha).

5. Digite o valor requerido no campo **Days before change required** (Dias antes da mudança requerida) e clique **OK**.

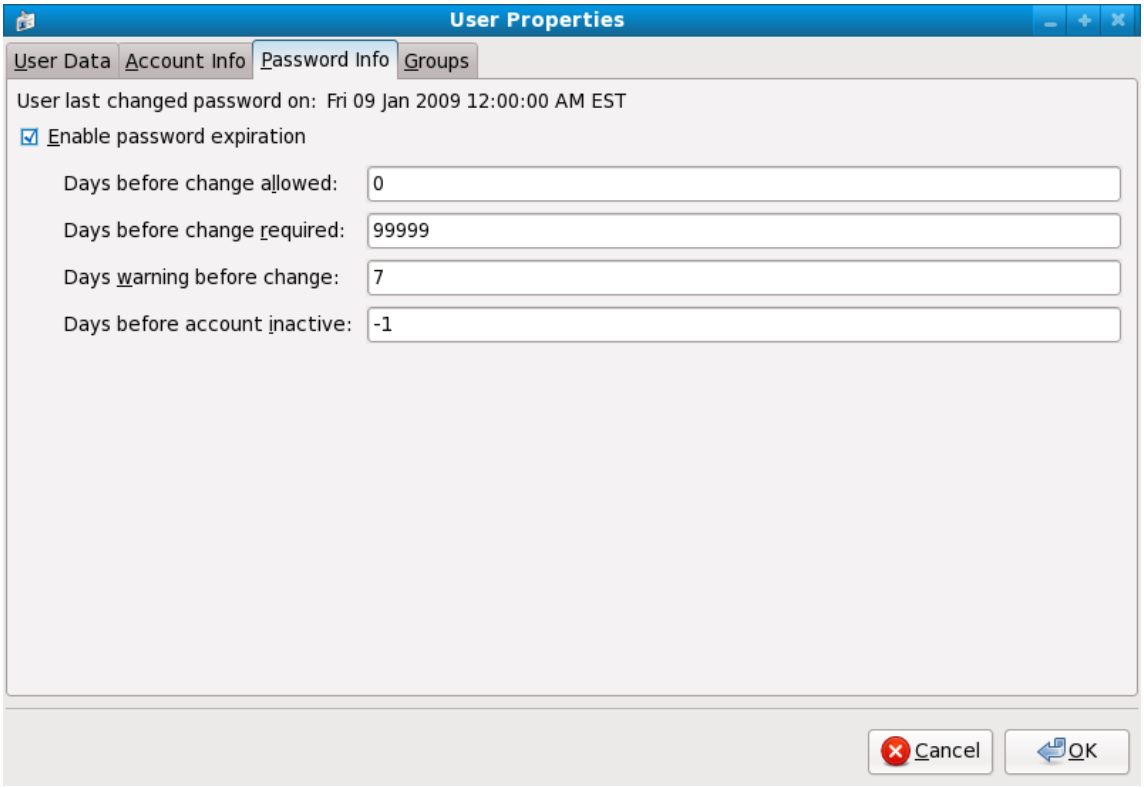
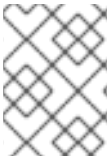


Figura 2.1. Especificando as opções de expiração de senha

2.1.4. Controles Administrativos

Quando administrar uma máquina doméstica, o usuário deve realizar algumas tarefas como usuário root ou adquirindo privilégios root pelo programa *setuid*, como o **sudo** ou **su**. Um programa setuid é um que opera com a ID de usuário (*UID*) do proprietário do programa em vez do usuário operar o programa. Tais programas são denotados por um **s** na seção proprietário de uma lista de formato longo, como no exemplo a seguir:

```
-rwsr-xr-x 1 root root 47324 May 1 08:09 /bin/su
```



NOTA

O **s** pode ser maiúsculo ou minusculo. Se aparecer como maiúsculo, significa que o bit de permissão subjacente não foi definido.

Para os administradores de sistemas de uma organização, entretanto, as escolhas podem ser feitas como o quanto de acesso administrativo os usuários dentro da organização podem ter em suas máquinas. Através do módulo PAM chamado **pam_console.so**, algumas atividades normalmente reservadas somente para o usuário root, como reinicialização e montagem de mídia removíveis são permitidas para o primeiro usuário que logar no console físico (consulte *Gerenciando Sign-On Únicos e Cartões Smart* para mais informações sobre o módulo **pam_console.so**). Entretanto, outras tarefas importantes de administração do sistema, como alterar definições de rede, configurar um mouse novo ou montar dispositivos de rede não são possíveis sem privilégios administrativos. Como resultado, administradores de sistema devem decidir quanta acessibilidade os usuários em sua rede devem receber.

2.1.4.1. Permitindo Acesso Root

Se os usuários dentro de uma organização são confiáveis e entendedores de informática, então permitir

acesso root a eles pode não ser um problema. Permitindo acesso root aos usuários significa que atividades menores, como adicionar dispositivos ou configurar interfaces de rede, podem ser manuseadas pelos usuários individuais, deixando os administradores de sistema livres para lidar com a segurança da rede e outras questões importantes.

Por outro lado, dar acesso root a usuários individuais podem levar às seguintes questões:

- *Desconfiguração da Máquina* — Usuários com acesso root pode desconfigurar suas máquinas e necessitar de assistência para resolver o problema. Ainda pior, eles podem abrir brechas de segurança sem perceber.

- *Executando Serviços Inseguros* — Usuários com acesso root podem rodar servidores inseguros em suas máquinas, como FTP ou Telnet, colocando nomes de usuários e senhas potencialmente em risco. Estes serviços transmitem estas informações pela a rede em texto puro.

- *Executar Anexos de Email como Root* — Ainda que raro, vírus em e-mails que afetam o Linux existem. O único momento entretanto que eles são uma ameaça é quando eles são executados pelo usuário root.

2.1.4.2. Desabilitando Acesso ao Root

Se um administrador não está confortável em permitir que os usuários autentiquem-se como root por estas e outras razões, a senha root deve ser mantida em segredo e acessar o nível de execução um ou modo de usuário único deve ser desativado através de proteção de senha do carregador de boot (consulte a Seção 2.1.2.2, “Senhas do Carregador de Boot” para mais informações sobre este tópico).

A Tabela 2.1, “Métodos para Desabilitar a Conta Root” descreve maneiras que um administrador pode assegurar ainda mais que logins root sejam desativados:

Tabela 2.1. Métodos para Desabilitar a Conta Root

Método	Descrição	Efeitos	Não afeta

Método	Descrição	Efeitos	Não afeta
Alternado o shell do root.	Edite o arquivo **/etc/passwd** e mude o shell de **/bin/bash** para **/sbin/nologin**.	Impede acesso ao shell do root e registra nos logs quaisquer tentativas. Os seguintes programas são impedidos de acessar a conta root: · **login** · **gdm** · **kdm** · **xdm** · **su** · **ssh** · **scp** · **sftp**	Programas que não requerem o uso do shell, como clientes FTP, clientes de email e muitos programas setuid. Os seguintes programas *não* são impedidos de acessar a conta root: · **sudo** · clientes FTP · clientes de E-mail
Desativar o acesso root por qualquer dispositivo de console (tty).	Um arquivo **/etc/securetty** impede o login de root em quaisquer dispositivos anexados ao computador.	Impede acesso à conta root pelo console ou rede. Os programas seguintes são impedidos de acessar a conta root: · **login** · **gdm** · **kdm** · **xdm** · Outros serviços de rede que abram o tty	Programas que não logam como root mas realizam tarefas administrativas através do setuid ou outros mecanismos. Os seguintes programas *não* são impedidos de acessar a conta root: · **su** · **sudo** · **ssh** · **scp** · **sftp**

Método	Descrição	Efeitos	Não afeta
Desativando logins root SSH.	Edite o arquivo **`/etc/ssh/sshd_config`** e defina o parâmetro **`PermitRootLogin`** para **`no`**.	Impede o acesso root pela suíte de ferramentas OpenSSH. Os seguintes programas são impedidos de acessar a conta root: · **`ssh`** · **`scp`** · **`sftp`**	Isto somente impede o acesso root à suítes de ferramentas OpenSSH.
Use o PAM para limitar acesso root aos serviços.	Edite o arquivo para o serviço alvo no diretório **`/etc/pam.d/`**. Tenha certeza que o **`pam_listfile.so`** é requerido para autenticação. [a]	Impede o acesso root aos serviços de rede que estão atentos ao PAM. Os seguintes serviços são impedidos de acessar a conta root: · clientes FTP · clientes de E-mail · **`login`** · **`gdm`** · **`kdm`** · **`xdm`** · **`ssh`** · **`scp`** · **`sftp`** · Quaisquer serviços atentos ao PAM.	Programas e serviços que não estão atentos ao PAM.

[a] Consulte a Seção 2.1.4.2.4, “Desativando o Root de usar o PAM” para detalhes.

2.1.4.2.1. Desativando o Shell do Root

Para impedir usuários de logar diretamente como root, o administrador do sistema pode definir o shell da conta root para **/sbin/nologin** no arquivo **/etc/passwd**. Isso impede o acesso à conta root através de comando que requerem um shell, como os comandos **su** e o **ssh**

IMPORTANTE

Programas que não requerem acesso ao shell, como clientes de e-mail ou o comando **sudo**, podem ainda acessar a conta root.

2.1.4.2.2. Desativando Logins Root

Para limitar ainda mais o acesso à conta root, os administradores pode desativar os logins root no console editando o arquivo **/etc/securetty**. Este arquivo lista todos os dispositivos em que o usuário root é permitido logar. Se o arquivo não existir, o usuário root pode se autenticar através de qualquer dispositivo de comunicação no sistema, seja pelo console ou uma interface de rede bruta. Isto é perigoso, porque um usuário pode se autenticar em sua máquina pela rede. Por padrão, o arquivo **/etc/securetty** do Red Hat Enterprise Linux somente permite ao usuário root se autenticar no console fisicamente anexado à máquina. Para impedir o root de se autenticar, remova o conteúdo deste arquivo, digitando o seguinte comando:

```
echo > /etc/securetty
```

ATENÇÃO

Um arquivo em branco **/etc/securetty** *não* impede o usuário root de se autenticar remotamente usando a suíte de ferramentas OpenSSH porque o console não é aberto até depois da autenticação.

2.1.4.2.3. Desativando Logins Root SSH

Logins root pelo protocolo SSH são desativados por padrão no Red Hat Enterprise Linux 6; entretanto, se esta opção foi ativada, ela pode ser desativada novamente editando o arquivo de configuração do daemon SSH (**/etc/ssh/sshd_config**). Mude a linha que contém:

```
PermitRootLogin yes
```

para ficar como:

```
PermitRootLogin no
```

Para essas mudanças terem efeito, o daemon SSH deve ser reiniciado. Isto pode ser feito pelo seguinte comando:

```
kill -HUP `cat /var/run/sshd.pid`
```

2.1.4.2.4. Desativando o Root de usar o PAM

O PAM, pelo módulo **/lib/security/pam_listfile.so**, permite uma grande flexibilidade em negar contas específicas. O administrador pode usar este módulo para referenciar uma lista de usuários que não estão permitidos de se autenticar. Abaixo está um exemplo de como o módulo é usado pelo servidor FTP **vsftpd** no arquivo de configuração do PAM **/etc/pam.d/vsftpd**, o caractere **** no final da primeira linha no exemplo seguinte *não* é necessário se a diretiva está em uma linha):

```
auth required /lib/security/pam_listfile.so item=user \
sense=deny file=/etc/vsftpd.ftpusers onerr=succeed
```

Isto instrui o PAM para consultar o arquivo **/etc/vsftpd.ftpusers** e negar acesso ao serviço para qualquer usuário listado. O administrador pode mudar o nome deste arquivo e manter listas separadas para cada serviço ou usar uma lista central para negar acesso à múltiplos serviços.

Se o administrador quer negar acesso à múltiplos serviços, uma linha similar pode ser adicionada aos arquivos de configuração do PAM, como **/etc/pam.d/pop** e o **/etc/pam.d/imap** para clientes de e-mail ou o **/etc/pam.d/ssh** para cliente SSH.

Para mais informações sobre o PAM, consulte *Gerenciando Sign-On Únicos e Cartões Smart*.

2.1.4.3. Limitando o Acesso Root

Mais do que completamente negar acesso ao usuário root, o administrador pode querer permitir acesso somente por programas setuid, como o **su** ou **sudo**.

2.1.4.3.1. O Comando su

Quando um usuário executa o comando **su**, ele é questionado pela senha root e depois da autenticação, recebe a linha de comando do root.

Uma vez autenticado pelo comando **su**, o usuário se torna o usuário root e possui acesso administrativo absoluto ao sistema[13]. Além disso, uma vez que um usuário se tornou root, é possível para ele usar o comando **su** para mudar para qualquer outro usuário no sistema sem ser questionado por uma senha.

Pela razão deste programa ser tão poderoso, os administradores dentro de uma organização podem desejar limitar o acesso a este comando.

Uma das maneiras mais simples de fazer isso é adicionar usuários ao grupo administrativo especial chamado *wheel*. Para fazer isso, digite o seguinte comando como root:

```
usermod -G wheel <username>
```

No comando anterior, substitua o *<username>* com o nome de usuário que você quer adicionar ao grupo **wheel**.

Você pode também usar o **Gerenciador de Usuários** para modificar afiliações de grupos, conforme a seguir. Nota: você precisa de privilégios de Administrador para realizar este procedimento.

1. Clique no menu **System** (Sistema) no painel, depois em **Administration** (Administração) e então clique em **Users and Groups** (Usuários e Grupos) para exibir o Gerenciador de Usuários. Alternativamente, digite o comando **system-config-users** no shell.

2. Clique na aba **Users** (Usuários) e selecione o usuário requerido na lista de usuários.

3. Clique em **Properties** (Propriedades) na barra de ferramentas para exibir as caixa de diálogo das Propriedades do Usuário (ou escolha **Properties** (Propriedades) no menu **File** (Arquivo)).

4. Clique na aba **Groups** (Grupos), marque a caixa de seleção para o grupo wheel, e então clique em **OK**. Veja a Figura 2.2, “Adicionando usuários ao grupo "wheel".”

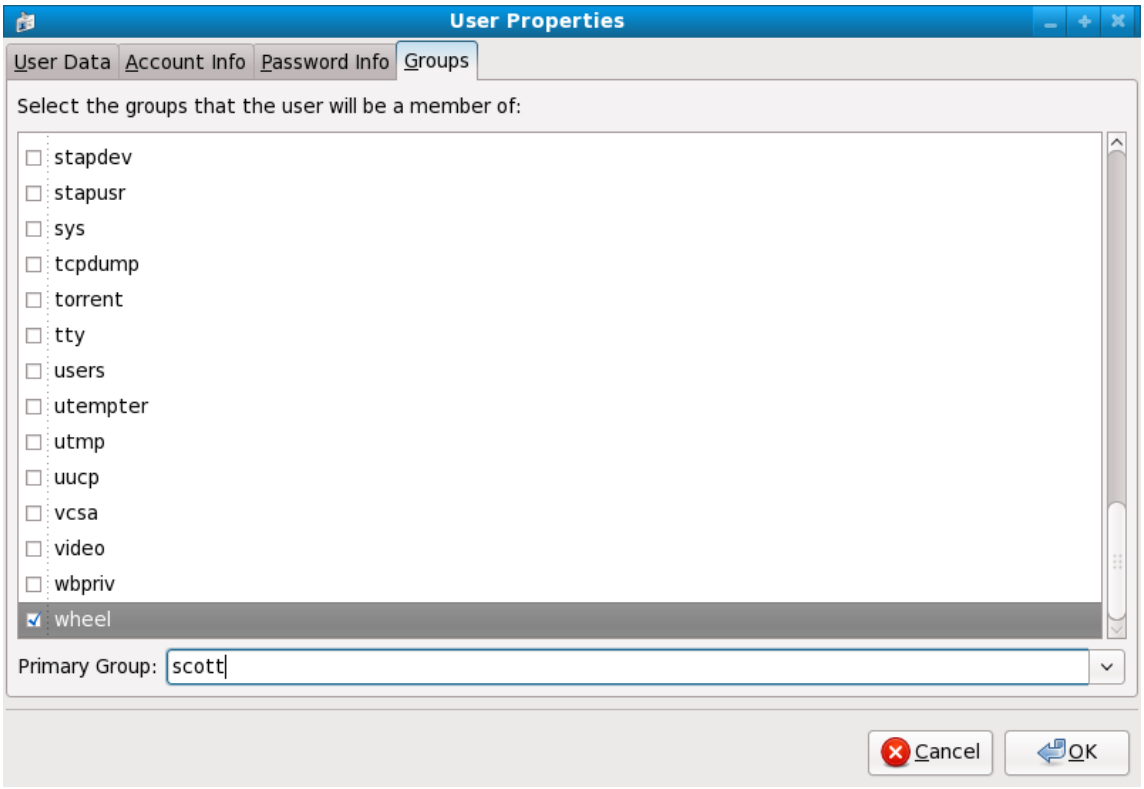


Figura 2.2. Adicionando usuários ao grupo "wheel".

Abra o arquivo de configuração do PAM para o **su** (**/etc/pam.d/su**) em um editor de textos e remova o comentário # na seguinte linha:

```
auth  required /lib/security/$ISA/pam_wheel.so use_uid
```

Esta mudança significa que somente membros do grupo administrativo **wheel** podem usar este programa.

NOTA

O usuário root é parte do grupo **wheel** por padrão.

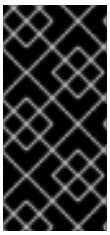
2.1.4.3.2. O Comando sudo

O comando **sudo** oferece outra abordagem para dar aos usuários o acesso administrativo. Quando usuários confiáveis precedem um comando administrativo com o **sudo**, eles são questionados *pela própria* senha. Então, quando eles são autenticados e assumindo que o comando é permitido, o comando administrativo é executado como se ele fosse o usuário root.

O formato básico do comando **sudo** é como a seguir:

```
sudo <command>
```

No exemplo acima, o *<command>* seria substituido por um comando normalmente reservado para o usuário root, tal como o **mount**.



IMPORTANTE

Usuários do comando **sudo** devem ter cuidade e sair do comando antes de deixar a máquina, já que o sudo pode executar o comando novamente sem ter a senha pedida dentro de um período de cinco minutos. Esta configuração pode ser alterada pelo arquivo de configuração, **/etc/sudoers**.

O comando **sudo** permite um alto grau de flexibilidade. Por exemplo, somente usuários listados no arquivo de configuração **/etc/sudoers** são permitidos usar o comando **sudo** e o comando é executado no shell do *usuário*, não no shell do root. Isto significa que o shell do root pode ser completamente desativado, como mostrado na Seção 2.1.4.2.1, “Desativando o Shell do Root”.

O comando **sudo** também fornece um registro de auditoria abrangente. Cada autenticação feita é registrada no arquivo **/var/log/messages** e o comando digitado junto com o nome do usuário é registrado no arquivo **/var/log/secure**.

Outra vantagem do comando **sudo** é que um administrador pode permitir usuários diferentes acessar comandos especificos baseados em suas necessidades.

Administradores querendo editar o arquivo de configuração do **sudo**, **/etc/sudoers**, devem usar o comando **visudo**.

Para dar a alguém privilégios administrativos completo, digite **visudo** e adicione uma linha similar à seguinte na seção de especificação de privilégio do usuário:

```
juan ALL=(ALL) ALL
```

Este exemplo declara que o usuário, **juan**, pode usar **sudo** de qualquer host e executar qualquer comando.

O exemplo abaixo ilustra a possível granularidade quando configurar o **sudo**:

```
%users localhost=/sbin/shutdown -h now
```

Este exemplo declara que qualquer usuário possa emitir o comando **/sbin/shutdown -h now** desde que ele seja emitido a partir do console.

A página man do **sudo** possui uma lista detalhada de opções para este arquivo.

2.1.5. Serviços de Rede Disponíveis

Enquanto o acesso do usuário aos controles administrativos é uma questão importante para administradores do sistema dentro de uma organização, monitorar quais serviços de rede estão ativos é de suma importância para qualquer um que administra ou opera um sistema Linux.

Muitos serviços no Red Hat Enterprise Linux 6 se comportam como servidores de rede. Se um serviço de rede estiver rodando em uma máquina então a aplicação do servidor (chamada *daemon*), está aguardando por conexões em uma ou mais portas da rede. Cada um destes servidores devem ser tratados como potencias portas de ataque.

2.1.5.1. Riscos ao Serviços

Serviços de rede podem impor muitos riscos para sistemas Linux. Abaixo segue uma lista de algumas das questões primárias:

- *Denial of Service Attacks (DoS)* — Inunda um serviço com pedidos, um ataque de negação de serviço pode fazer um sistema inutilizável conforme ele tenta registrar e responder cada pedido.
- *Distributed Denial of Service Attack (DDoS)* — Um tipo de ataque DoS que usa múltiplas máquinas comprometidas (muitas vezes milhares ou mais) para direcionar um ataque coordenado à um serviço, inundando com pedidos e fazendo o serviço inutilizável.
- *Script Vulnerability Attacks* — Se um servidor estiver usando scripts para executar ações dentro do servidor, como servidores web geralmente fazem, um invasor pode atacar os scripts impropriamente. Estes ataques de vulnerabilidade de script podem levar à uma condição de buffer overflow ou permitir ao invasor alterar arquivos no sistema.
- *Buffer Overflow Attacks* — Serviços que conectam às portas numeradas de 0 a 1023 devem rodar como um usuário administrativo. Se a aplicação tiver um buffer overflow explorável, um invasor poderia ganhar acesso ao sistema como o usuário executando o deaemon. Pela razão que buffer overflow existe, os crackers usam ferramentas automatizadas para identificar sistemas com vulnerabilidades e uma vez que ganharam acesso, eles usam rootkits automatizados para manter o acesso ao sistema.

NOTA

A ameaça das vulnerabilidades do buffer overflow é minimizada no Red Hat Enterprise Linux pelo *ExecShield*, uma segmentação de memória executável e tecnologia de proteção suportada pelos processadores únicos e múltiplos do kernel compatíveis com x86. O ExecShield reduz o risco de buffer overflow separando a memória virtual em segmentos executáveis e não executáveis. Qualquer código de programa que tenta executar fora do segmento executável (tal como um código malicioso injetado a partir de uma exploração de buffer overflow) aciona um defeito de segmentação e o termina.

O Execshield também inclui suporte para a tecnologia *No eXecute* (NX) em plataformas AMD64 e tecnologia *eXecute Disable* (XD) em Itanium e sistemas Intel® 64. Estas tecnologias trabalham em conjunto com o ExecShield para impedir códigos maliciosos de rodar na porção de memória virtual executável com uma granularidade de 4KB de código executável, baixando o risco de ataque a partir de explorações stealthy buffer overflow.

IMPORTANTE

Para limitar a exposição de ataques na rede, todos os serviços que não são usados devem ser desligados.

2.1.5.2. Identificando e Configurando Serviços

Para aumentar a segurança, a maioria dos serviços de rede instalados com o Red Hat Enterprise Linux são desligados por padrão. Existem, entretanto, algumas exceções notáveis:

- **cupsd** — O servidor de impressão do Red Hat Enterprise Linux.
- **lpd** — Um servidor de impressão alternativo.

- **xinetd** — Um super servidor que controla conexões de uma variedade de servidores subordinados, tais como **gssftp** e **telnet**.
- **sendmail** — O Sendmail *Mail Transport Agent* (MTA) é ativado por padrão, mas somente escuta por conexões do localhost.
- **sshd** — O servidor OpenSSH que é um substituto seguro para o Telnet.

Quando determinar se deixar ou não estes serviços rodando, é melhor usar o bom senso com cautela. Por exemplo, se uma impressora não está disponível, não deixe o **cupsd** rodando. O mesmo é verdadeiro para o **portmap**. Se você não montar os volumes NFSv3 ou usar o NIS (o serviço **ypbind**), então o **portmap** deve ser desativado.

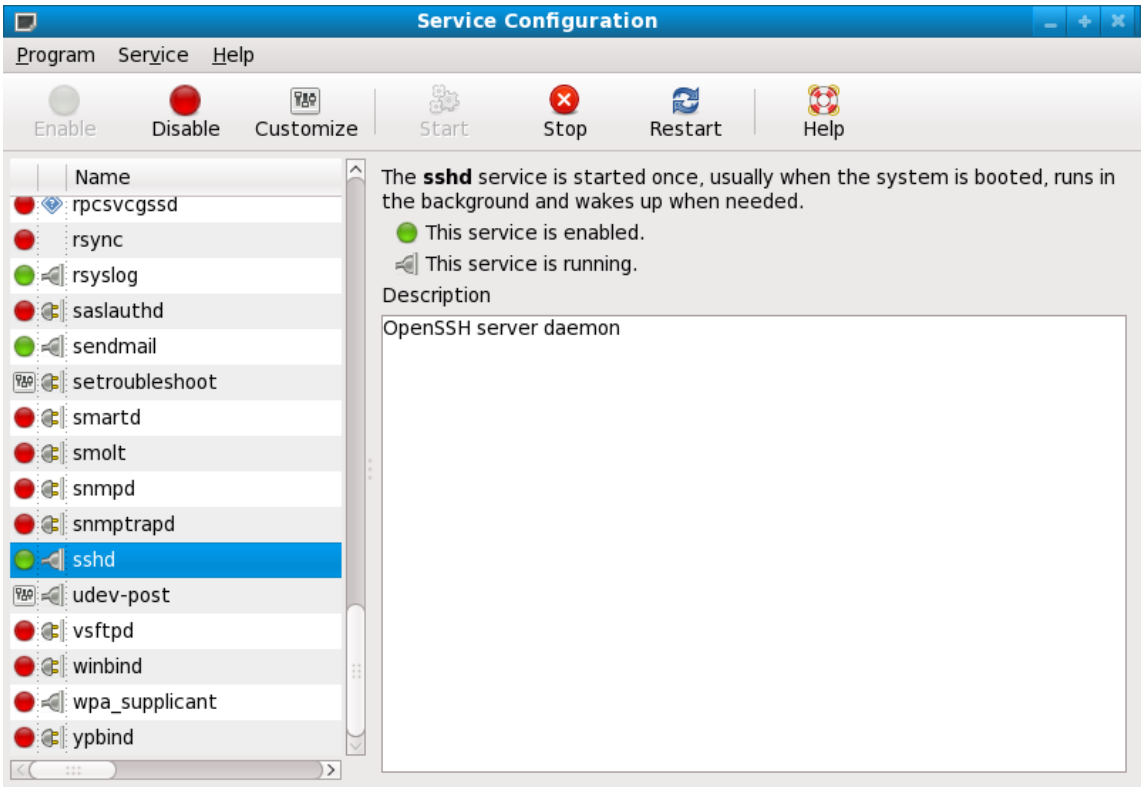


Figura 2.3. Ferramentas de Configuração de Serviços

Se não estiver certo do propósito para um serviço em particular, a **Ferramenta de Configuração de Serviços** possui um campo de descrição, ilustrado na Figura 2.3, "**Ferramentas de Configuração de Serviços**", que fornece informações adicionais.

Verificando quais serviços de rede estão disponíveis para iniciar no momento de boot é somente uma parte da história. Você deve também verificar quais portas estão abertas e escutando. Consulte a Seção 2.2.8, "Verificando Quais Portas Estão Escutando" para mais informações.

2.1.5.3. Serviços Inseguros

Potencialmente, qualquer serviço de rede é inseguro. Isto explica porque desligar serviços não utilizados é tão importante. Explorações de serviços são rotineiramente revelados e corrigidos, sendo muito importante atualizar regularmente pacotes associados com quaisquer serviços de rede. Consulte a Seção 1.5, "Atualizações de Segurança" para mais informações.

Alguns protocolos de rede são inerentemente mais inseguros que outros. Estes incluem quaisquer serviços que:

- *Transmitir Nomes de Usuários e Senhas Sobre uma Rede Sem Encriptação*— Muitos protocolos antigos, como Telnet e FTP não fazem encriptação da sessão de autenticação e deveriam ser evitados sempre que possível.

- *Transmitir Dados Sensíveis Sobre uma Rede Sem Encriptação* — Muitos protocolos transmitem dados sobre uma rede sem encriptação. Estes protocolos incluem Telnet, FTP, HTTP e SMTP. Muitos sistemas de arquivos de rede, como NFS e SMB também transmitem informações sobre a rede sem encriptação. É a responsabilidade do usuário limitar quais tipos de dados são transmitidos quando usar estes protocolos.

  Serviços de Dump de Memória Remota, como o **netdump**, transmitem os conteúdos de memória sobre a rede sem encriptação. Dumps de memória podem conter senhas ou, ainda pior, entradas de banco de dados e outras informações sensíveis.

  Outros serviços como o **finger** e **rwhod** revelam informações sobre os usuários do sistema.

Exemplos de serviços inerentemente inseguros incluem **rlogin**, **rsh**, **telnet** e **vsftpd**.

Todos os logins remotos e programas de shell (**rlogin**, **rsh**, e **telnet**) devem ser evitados em favor do SSH. Consulte a Seção 2.1.7, “Ferramentas de Comunicação Avançadas de Segurança” para mais informações sobre o **sshd**.

O FTP não é inerentemente perigoso à segurança do sistema como shells remotos, mas servidores FTP deve ser cuidadosamente configurados e monitorados para evitar problemas. Consulte a Seção 2.2.6, “Protegendo o FTP” para mais informações sobre proteger servidores FTP.

Serviços que devem ser cuidadosamente implementados e colocados em firewall incluem:

- **finger**
- **authd** (este era chamado **identd** em versões anteriores do Red Hat Enterprise Linux.)
- **netdump**
- **netdump-server**
- **nfs**
- **rwhod**
- **sendmail**
- **smb** (Samba)
- **yppasswdd**
- **ypserv**
- **ypxfrd**

Mais informações sobre proteger serviços de rede estão disponíveis na Seção 2.2, “Segurança do Servidor”.

A próxima seção discute ferramentas para configurar um firewall simples.

2.1.6. Firewalls Pessoais

Depois de que os serviços de redes *necessários* são configurados, é importante implementar um firewall.

IMPORTANTE

Você deve configurar os serviços necessários e implementar um firewall *antes* de conectar à internet ou qualquer outra rede que você não confiar.

Os firewalls impedem pacotes de rede de acessar a interface de rede de sistema. Se um pedido é feito a uma porta que está bloqueada por um firewall, o pedido é ignorado. Se um serviço está escutando em uma dessas portas bloqueadas, ele não recebe os pacotes e está efetivamente desativado. Por esta razão, cuidado deve ser tomado quando configurar um firewall para bloquear o acesso às portas que não estão em uso, enquanto não bloquear acesso às portas usadas pelos serviços configurados.

Para a maioria dos usuários, a melhor ferramenta para configurar um firewall simples é a ferramenta de configuração de firewall gráfica que vem junto com o Red Hat Enterprise Linux: a **Firewall Configuration Tool** (`system-config-firewall`) . Esta ferramenta cria regras abrangentes de `iptables` para um firewall de uso geral usando uma interface de painel de controle.

Consulte a Seção 2.5.2, “Configuração de Firewall Básica” para mais informações sobre usar esta aplicação e suas opções disponíveis.

Para usuários avançados e administradores de servidor, configurar manualmente um firewall com o `iptables` é provavelmente a melhor opção. Consulte a Seção 2.5, “Firewalls” para mais informações. Consulte a Seção 2.6, “IPTables” para um guia compreensivo do comando `iptables`.

2.1.7. Ferramentas de Comunicação Avançadas de Segurança

Conforme o tamanho e popularidade da internet cresceu, também cresceram as ameaças de interceptação de comunicação. Ao passar dos anos, ferramentas tem sido desenvolvidas para encriptar comunicações conforme elas são transferidas na rede.

O Red Hat Enterprise Linux 6 vem com duas ferramentas básicas que usam um alto nível de algoritmos de criptografia baseados em criptografia de chave pública para proteger as informações conforme elas viajam na rede.

- *OpenSSH* — Uma implementação grátis do protocolo SSH para encriptação de comunicação de rede.
- *Gnu Privacy Guard (GPG)* — Uma implementação grátis da aplicação de encriptação PGP (Pretty Good Privacy) para dados.

O OpenSSH é uma maneira segura de acessar uma máquina remota e substituir serviços antigos, sem criptografia como o `telnet` e `rsh`. O OpenSSH inclui um serviço de rede chamado `sshd` e três aplicações clientes de linha de comando:

- `ssh` — Um cliente de acesso remoto de console seguro.
- `scp` — Um comando de cópia remota seguro.
- `sftp` — Um cliente de pseudo ftp seguro que permite sessões de transferência de arquivos seguros.

Consulte o Seção 3.6, “Secure Shell (Shell Segura)” para mais informações relacionadas sobre o OpenSSH.

IMPORTANTE

Apesar do serviço **sshd** ser inerentemente seguro, o serviço *deve* ser mantido atualizado para impedir ameaças de segurança. Consulte a Seção 1.5, “Atualizações de Segurança” para mais informações.

O GPG é uma maneira de proteger comunicações de e-mail privadas. Ele pode ser usado tanto para enviar e-mails com dados sensíveis em redes públicas quanto proteger dados sensíveis em discos rígidos.

2.2. SEGURANÇA DO SERVIDOR

Quando um sistema é usado como um servidor em uma rede pública, ele se torna um alvo para ataques. Endurecendo o sistema e trancando serviços é, portanto, de suma importância para o administrador do sistema.

Antes de se aprofundar em questões específicas, revise as seguintes dicas gerais para aumentar a segurança do servidor:

- Mantém todos os serviços atualizados, para protege-los contra as últimas ameaças.
- Use protocolos seguros sempre que possível.
- Sirva somente um tipo de serviço de rede por máquina sempre que possível.
- Monitore todos os servidores cuidadosamente por atividades suspeitas.

2.2.1. Assegure os Serviços com TCP Wrappers e xinetd

Os *TCP Wrappers* fornecem controle de acesso à uma variedade de serviços. A maioria dos serviços de rede modernos, como SSH, Telnet e FTP fazem uso dos TCP Wrappers, que fazem guarda entre pedidos de entrada e os serviços solicitados.

Os benefícios oferecidos pelos TCP Wrappers são reforçados quando usados em conjunto com o **xinetd**, um super servidor que fornece acesso adicional, registro de logs, associação, redirecionamento e controle de utilização de recursos.

NOTA

É uma boa idéia usar as regras de firewall iptables em conjunto com os TCP Wrappers e **xinetd** para criar redundância dentro dos controles de acesso de serviço. Consulte a Seção 2.5, “Firewalls” para mais informações sobre implementar firewalls com comandos iptables.

As subseções a seguir pressupõem um conhecimento básico de cada tópico e foco em opções de segurança específicas.

2.2.1.1. Aumentando a Segurança com TCP Wrappers

Os TCP Wrappers são capazes de muito mais do que negar acesso á serviços. Esta seção ilustra como eles podem ser usados para enviar banners de conexão, avisos de ataque de determinados hosts e aumenta a funcionalidade de registro de log. Consulte a página man **hosts_options** para informações

sobre a funcionalidade dos TCP Wrappers e idioma de controle. Consulte a página man **xinetd.conf** disponível online em http://linux.die.net/man/5/xinetd.conf para os sinalizadores disponíveis, que agem como opções que você pode aplicar em um serviço.

2.2.1.1.1. TCP Wrappers e Banners de Conexão

Exibir um banner apropriado quando os usuários conectam é uma boa maneira de fazer que os potencias invasores saibam que o administrador do sistema está vigilante. Você pode também controlar qual informação sobre o sistema é apresentado aos usuários. Para implementar um banner de TCP Wrappers para um serviço, use a opção **banner**.

Este exemplo implementa um banner para o **vsftpd**. Para iniciar, crie um arquivo de banner. Ele pode estar em qualquer lugar no sistema, mas ele deve ter o mesmo nome como o daemon. Para este exemplo, este arquivo é chamado **/etc/banners/vsftpd** e contém a seguinte linha:

```
220-Hello, %c
220-All activity on ftp.example.com is logged.
220-Inappropriate use will result in your access privileges being removed.
```

O token **%c** fornece uma variedade de informações sobre o cliente, tais como o nome de usuário e hostname ou nome de usuário e endereço de IP para fazer a conexão ainda mais intimidadora.

Para este banner ser mostrado às conexões de entrada, adicione a seguinte linha ao arquivo **/etc/hosts.allow**:

```
vsftpd : ALL : banners /etc/banners/
```

2.2.1.1.2. TCP Wrappers e Avisos de Ataques

Se um determinado host ou rede tiver sido detectada atacando o servidor, os TCP Wrappers podem ser usados para avisar o administrador de ataques subsequentes daquele host ou rede usando a diretiva **spawn**.

Neste exemplo, pressuponha que um cracker da rede 206.182.68.0/24 foi detectado tentando atacar o servidor. Coloque a seguinte linha no arquivo **/etc/hosts.deny** para negar quaisquer tentativas de conexão dessa rede, e registrar as tentativas em log em um arquivo especial:

```
ALL : 206.182.68.0 : spawn /bin/echo `date` %c %d >>
/var/log/intruder_alert
```

O token **%d** fornece o nome do serviço que o invasor estava tentando acessar.

Para permitir a conexão e registra em log, coloque a diretiva **spawn** no arquivo **/etc/hosts.allow**.

NOTA

Por causa que a diretiva **spawn** executa qualquer comando shell, é uma boa idéia criar um script especial para notificar o administrador ou executar uma cadeia de comandos no evento de um determinado cliente tentar se conectar ao servidor.

2.2.1.1.3. Os TCP Wrappers e Registro de Log Avançado

Se certos tipos de conexões são mais preocupantes que outras, o nível de log pode ser elevado para esse serviço usando a opção **severity**.

Neste exemplo, pressuponha que qualquer um tentando se conectar à porta 23 (a porta Telnet) em um servidor FTP seja um invasor. Para simbolizar isso, coloque um sinalizador **emerg** nos arquivos de log em vez do sinalizador padrão, **info** e negue a conexão.

Para fazer isso, coloque a seguinte linha no **/etc/hosts.deny**:

```
in.telnetd : ALL : severity emerg
```

Isto usa a facilidade de registro de log padrão **authpriv**, mas eleva a prioridade do valor padrão de **info** para **emerg**, que posta mensagens de log diretamente ao console.

2.2.1.2. Aumentando a Segurança com o xinetd

Esta seção foca no uso do **xinetd** para definir um serviço de interceptação (trap) e usa-lo para controlar os níveis de recursos disponíveis para qualquer serviço **xinetd** dado. Definindo limites de recursos disponíveis para serviços pode ajudar impedir ataques DoS (*Denial of Service*). Consulte as páginas man do **xinetd** e do **xinetd.conf** para uma lista de opções disponíveis.

2.2.1.2.1. Configurando uma Interceptação (Trap)

Um recurso importante do **xinetd** é sua habilidade de adicionar hosts à uma lista **no_access** global. Hosts nesta lista são conexões subsequentes negadas a serviços gerenciados pelo **xinetd** por um período especificado ou até que o **xinetd** seja reiniciado. Você pode fazer isto usando o atributo **SENSOR**. Esta é uma fácil maneira de bloquear hosts tentando escanear as portas neste servidor.

O primeiro passo para definir um **SENSOR** é escolher um serviço que você não planeja usar. Para este exemplo, o Telnet é usado.

Edite o arquivo **/etc/xinetd.d/telnet** e mude as linhas das **flags** (sinalizadores) para:

```
flags           = SENSOR
```

Adicione a seguinte linha:

```
deny_time       = 30
```

Isto nega qualquer outra tentativa de conexão à esta porta por esse host por 30 minutos. Outros valores aceitáveis para o atributo **deny_time** são o FOREVER, que mantém o banimento em efeito até que o **xinetd** seja reiniciado, e o NEVER, que permite a conexão e registra isso em log.

Finalmente, a última linha deve ser:

```
disable         = no
```

Isto ativa a própria interceptação.

Enquanto usar o **SENSOR** é uma boa maneira para detectar e parar conexões de hosts indesejáveis, ela possui desvantagens:

- Não funciona contra escaneamentos stealth.

- Um invasor que sabe que o **SENSOR** está rodando pode montar um ataque Dos (Denial of Service) contra determinados hosts, falsificando seus endereços IP e se conectando à porta proibida.

2.2.1.2.2. Controlando Recursos de Servidor

Um outro recurso importante do **xinetd** é a habilidade de configurar limites de serviços sobre seu controle.

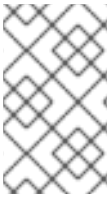
Isso pode ser feito usando as seguintes diretivas:

- **cps = <number_of_connections> <wait_period>** — Limita a taxa de conexões de entrada. Esta diretiva leva dois argumentos:
  - **<number_of_connections>** — O número de conexões por segundo para manuseio. Se a taxa de conexões de entrada é maior que isso, o serviço é temporariamente desativado. O valor padrão é cinquenta (50).
  - **<wait_period>** — O número de segundos para esperar antes da reativação do serviço depois que ele foi desativado. O intervalo padrão é dez (10) segundos.
- **instances = <number_of_connections>** — Especifica o número total de conexões permitidas a um serviço. Esta diretiva aceita tanto um valor de número inteiro ou **UNLIMITED** (ilimitado).
- **per_source = <number_of_connections>** — Especifica o número de conexões permitidas a um serviço por cada host. Esta diretiva aceita tanto um valor de número ou **UNLIMITED** (ilimitado).
- **rlimit_as = <number[K|M]>** — Especifica a quantidade de espaço de endereço de memória que o serviço pode ocupar em kilobytes ou megabytes. Esta diretiva aceita tanto um valor de número inteiro ou **UNLIMITED**.
- **rlimit_cpu = <number_of_seconds>** — Especifica o período de tempo em segundos que um serviço pode ocupar na CPU. Esta diretiva aceita tanto um integrador de número inteiro ou **UNLIMITED** (limitado).

Usando estas diretivas pode ajudar a impedir qualquer serviço **xinetd** único de sobrecarregar o sistema, resultando em um DoS (denial of service).

2.2.2. Protegendo o Portmap

O serviço **portmap** é um daemon de atribuição de porta dinâmica para serviços RPC como NIS e NFS. Ele possui uma mecanismo de autenticação fraco e possui a habilidade de atribuir uma grande variedade de portas para os serviços que controla. Por estas razões, ele é difícil de proteger.



NOTA

Protegendo o **portmap** somente afeta as implementações NFSv2 e NFSv3, já que o NFSv4 não o requer mais. Se você planeja implementar um servidor NFSv2 ou NFSv3, então o **portmap** é requerido e as seguintes seções se aplicam.

Se rodar os serviços RPC, siga estas regras básicas.

2.2.2.1. Proteja o portmap com TCP Wrappers

É importante usar os TCP Wrappers para limitar quais redes ou hosts têm acesso ao serviço **portmap** desde que ele não possua uma forma de autenticação embutida.

Além disso, use *somente* endereços IP quando limitar acesso ao serviço. Evite usar hostnames, já que eles podem ser falsificados por envenenamento de DNS e outros métodos.

2.2.2.2. Proteger o portmap com o iptables

Para restringir ainda mais o acesso ao serviço **portmap**, é uma boa idéia adicionar regras de iptables ao servidor e restringir acesso às redes especificas.

Abaixo estão dois exemplos de comandos iptables. O primeiro permite conexões TPC à porta 111 (usada pelo serviço **portmap**) da rede 192.168.0.0/24. A segunda permite conexões TCP à mesma porta do local host. Isto é necessário para o serviço **sgi_fam** usado pelo **Nautilus**. Todos os outros pacotes são largados.

```
iptables -A INPUT -p tcp ! -s 192.168.0.0/24 --dport 111 -j DROP
iptables -A INPUT -p tcp -s 127.0.0.1  --dport 111 -j ACCEPT
```

Para similarmente limitar tráfego UDP, use o seguinte comando.

```
iptables -A INPUT -p udp ! -s 192.168.0.0/24  --dport 111 -j DROP
```

NOTA

Consulte a Seção 2.5, “Firewalls” para mais informações sobre implementar firewalls com comandos iptables.

2.2.3. Protegendo o NIS

A *Network Information Service* (NIS) é um serviço RPC, chamado **ypserv**, que é usado em conjunto com o **portmap** e outros serviços relacionados para distribuir mapas de nomes de usuários, senhas e outras informações sensíveis a qualquer computador alegando estar dentro do domínio.

Um servidor NIS compreende de diversas aplicações. Elas incluem o seguinte:

- **/usr/sbin/rpc.yppasswdd** — Também chamado serviço **yppasswdd**, este daemon permite usuários alterar suas senhas NIS.
- **/usr/sbin/rpc.ypxfrd** — Também chamado serviço **ypxfrd**, este daemon é responsável por transferências de mapas NIS pela rede.
- **/usr/sbin/yppush** — Esta aplicação propaga bancos de dados NIS para múltiplos servidores NIS.
- **/usr/sbin/ypserv** — Este é o daemon do servidor NIS.

O NIS é um pouco inseguro para os padrões de hoje. Ele não possui mecanismos de autenticação e transmite todas suas informações pela rede sem criptografia, incluindo senhas hash. Como um resultado, cuidado extremo deve ser tomado quando configurar uma rede que usa o NIS. Isto é mais complicado pelo fato que a configuração padrão do NIS é inerentemente insegura.

É recomendado que qualquer um que esteja planejando implementar um servidor NIS, primeiro proteja o serviço **portmap** conforme descrito na Seção 2.2.2, “Protegendo o Portmap”, em seguida, aborde as seguintes questões, como planejar a rede.

2.2.3.1. Planeje a Rede Cuidadosamente

Por causa que o NIS transmite informações sensíveis sem criptografia pela rede, é importante que o serviço seja rodado por trás de um firewall e em uma rede segmentada e segura. Sempre que a informação NIS é transmitida por uma rede insegura, há riscos de ser interceptada. Um planejamento de rede cuidadoso pode ajudar a impedir grandes brechas na segurança.

2.2.3.2. Use um Nome de Domínio NIS e Hostname como se fosse uma Senha.

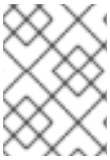
Qualquer máquina dentro de um domínio NIS pode usar comandos para extrair informações do servidor sem autenticação, desde que o usuário saiba o hostname DNS do servidor NIS e o nome de domínio NIS.

Por exemplo, se alguém tanto se conectar com um laptop na rede ou invadir a rede externamente (e fazer um spoof de um endereço IP), o seguinte comando revela o mapa **/etc/passwd**:

```
ypcat -d <NIS_domain> -h <DNS_hostname> passwd
```

Se um invasor é o usuário root, ele pode obter o arquivo **/etc/shadow** digitando o seguinte comando:

```
ypcat -d <NIS_domain> -h <DNS_hostname> shadow
```



NOTA

Se o Kerberos é usado, o arquivo **/etc/shadow** não é armazenado dentro de um mapa NIS.

Para fazer o acesso aos mapas NIS mais difícil para um invasor, crie uma string aleatória para o hostname DNS, tal como **o7hfawtgmhwg.domain.com**. Da mesma maneira, crie um nome de domínio NIS aleatório diferente. Isto deixa mais difícil para um invasor acessar o servidor NIS.

2.2.3.3. Edite o Arquivo /var/yp/securenets

Se o arquivo **/var/yp/securenets** está em branco ou não existe (como é o caso depois da instalação padrão), o NIS escuta todas as redes. Uma das primeiras coisas a fazer é colocar os pares de mascara de rede/rede no arquivo para que o **ypserv** somente responda aos pedidos da rede apropriada.

Abaixo segue um modelo de entrada de um arquivo **/var/yp/securenets**:

```
255.255.255.0     192.168.0.0
```

ATENÇÃO

Nunca inicie um servidor NIS pela primeira vez sem criar o arquivo **/var/yp/securenets**.

Esta técnica não fornece proteção contra um ataque de spoof de IP, mas ao menos impões limites em quais redes o servidor NIS serve.

2.2.3.4. Atribua Portas Estáticas e Use Regras iptables

Todos os servidores relacionados ao NIS podem ser atribuídos à portas específicas exceto o **rpc.yppasswdd** — o daemon que permite usuários mudar suas senhas de login. Atribuir portas aos outros dois daemons do servidor NIS **rpc.ypxfrd** and **ypserv**, permite a criação de regras de firewall para proteger ainda mais o daemons do servidor NIS de invasores.

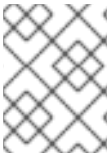
Para fazer isso, adicione as seguintes linhas ao **/etc/sysconfig/network**:

```
YPSERV_ARGS="-p 834" YPXFRD_ARGS="-p 835"
```

As seguintes regras de iptables podem então ser usadas para forçar quais redes o servidor escuta nestas portas:

```
iptables -A INPUT -p ALL ! -s 192.168.0.0/24  --dport 834 -j DROP
iptables -A INPUT -p ALL ! -s 192.168.0.0/24  --dport 835 -j DROP
```

Isto significa que o servidor somente permite conexões às portas 834 e 835 se os pedidos chegam da rede 192.168.0.0/24, sem importar o protocolo.



NOTA

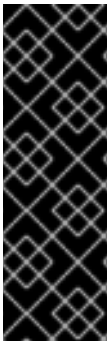
Consulte a Seção 2.5, “Firewalls” para mais informações sobre implementar firewalls com comandos iptables.

2.2.3.5. Use a Autenticação Kerberos

Uma das questões a considerar quando o NIS é usado para autenticação é que sempre que um usuário se autentica em uma máquina, uma senha hash do mapa **/etc/shadow** é enviado pela rede. Se um invasor ganha acesso a um domínio NIS e intercepta o tráfego de rede, ele pode coletar nomes de usuários e senhas hash. Com tempo suficiente, um programa de quebra de senha pode adivinhar senhas fracas e um invasor pode ter acesso a uma conta válida na rede.

Desde que o Kerberos usa criptografia de chave secreta, nenhuma senha hash é enviada pela rede, fazendo o sistema muito mais seguro. Consulte *Gerenciando Cartões Single Sign-On e Smart Cards* para mais informações sobre o Kerberos.

2.2.4. Protegendo o NFS



IMPORTANTE

A versão do NFS incluída no Red Hat Enterprise Linux 6, o NFSv4, não requer mais o serviço **portmap** conforme explicado na Seção 2.2.2, “Protegendo o Portmap”. O tráfego de NFS agora utiliza o TCP em todas as versões, em vez do UDP, e requer isso quando usar o NFSv4. O NFSv4 agora inclui o usuário Kerberos e a autenticação de grupo como parte do módulo do kernel **RPCSEC_GSS**. Informações sobre o **portmap** é ainda incluída, desde que o Red Hat Enterprise Linux 6 suporta o NFSv2 e o NFSv3, e ambos utilizam o **portmap**.

2.2.4.1. Planeje a Rede Cuidadosamente

Agora que o NFSv4 possui a habilidade de transmitir todas as informações criptografadas usando o Kerberos pela rede, é importante que o serviço esteja configurado corretamente se ele está por trás de um firewall ou em uma rede segmentada. O NFSv2 e NFSv3 ainda transmitem dados sem segurança e isso deve ser levado em consideração. Um projeto de rede pensado nessas considerações podem impedir brechas na segurança.

2.2.4.2. Atenção aos Erros de Sintaxe

O servidor NFS determina quais sistemas de arquivos exportar e para quais hosts exportar estes diretórios consultando o arquivo **/etc/exports**. Cuidado para não adicionar espaços fora de lugar quando editar este arquivo.

Por exemplo, a linha seguinte no arquivo **/etc/exports** compartilha o diretório **/tmp/nfs/** ao host **bob.example.com** com permissões de leitura/escrita.

```
/tmp/nfs/     bob.example.com(rw)
```

A linha seguinte no arquivo **/etc/exports**, por outro lado, compartilha o mesmo diretório do host **bob.example.com** com permissões de leitura somente e compartilha com o *world* com permissões de leitura/escrita devido ao carácter espaço único depois do hostname.

```
/tmp/nfs/     bob.example.com (rw)
```

É uma boa prática verificar quaisquer compartilhamentos de NFS configurados usando o comando **showmount** para verificar o que está sendo compartilhado:

```
showmount -e <hostname>
```

2.2.4.3. Não Use a Opção no_root_squash

Por padrão, o compartilhamento NFS muda o usuário root para usuário **nfsnobody**, uma conta de usuário sem previlégios. Isto muda o proprietário de todos os arquivos criados pelo root para **nfsnobody**, que impede o upload de programas com o setuid definido.

Se o **no_root_squash** é usado, os usuários root remotos são capazes de mudar quaisquer arquivos no sistema de arquivos compartilhados e deixar aplicações infectadas por trojans para que outros usuários as executem sem saber.

2.2.4.4. Configuração de Firewall NFS

As portas usadas para o NFS são atribuídas dinamicamente pelo rpcbind, que pode causar problemas quando criar regras de firewall. Para simplificar este processo, use o arquivo */etc/sysconfig/nfs* para especificar quais portas serão usadas.

- **MOUNTD_PORT** — Porta UDP e TCP para o mountd (rpc.mountd)
- **STATD_PORT** — Porta UDP e TCP para o status (rpc.statd)
- **LOCKD_TCPPORT** — Porta TCP para o nlockmgr (rpc.lockd)
- **LOCKD_UDPPORT** — Porta UDP para o nlockmgr (rpc.lockd)

Números de Porta especificadas não devem ser usados por qualquer outro serviço. Configure seu firewall para permitir os números de portas especificadas e também a porta 2049 UDP e TCP (NFS)

Rode o comando **rpcinfo -p** no servidor NFS para ver quais portas e programas RPC estão sendo usados.

2.2.5. Protegendo o Servidor HTTP Apache

O Servidor HTTP Apache é um dos serviços mais estáveis e seguros que vem com o Red Hat Enterprise Linux. Um grande número de opções e técnicas estão disponíveis para proteger o Servidor HTTP Apache — muitas para serem descritas em detalhes aqui. A seção seguinte explica brevemente boas práticas quando rodar o Servidor HTTP Apache.

Sempre verifique que quaisquer scripts rodando no sistema funcionam conforme pretendido *antes* de coloca-los em produção. Também assegure-se que somente o usuário root possui permissões de escrita a quaisquer diretórios que contém os scripts ou CGIs. Para fazer isso, rode o seguinte comando como usuário root:

1. ```
   chown root <directory_name>
   ```
2. ```
   chmod 755 <directory_name>
   ```

Administradores de sistemas devem ter cuidados quando usar as seguintes opções de configuração (definidos em **/etc/httpd/conf/httpd.conf**):

FollowSymLinks

Esta diretiva está ativada por padrão, então seja cauteloso quando criar links simbólicos ao documento root do Servidor Web. Por exemplo, não é uma boa idéia fornecer um link simbolico para o **/**.

Indexes

Esta diretiva está ativada por padrão, mas pode não ser desejável. Para impedir visitantes de navegar pelos arquivos do servidor, remova esta diretiva.

UserDir

A diretiva **UserDir** está desativada por padrão por causa que ela pode confirmar a presença de uma conta de usuário no sistema. Para ativar a navegação de diretórios no servidor, use as seguintes diretivas:

```
UserDir enabled
UserDir disabled root
```

Estas diretivas ativam a navegação de diretórios de usuários para todos os diretórios de usuários menos o **/root/**. Para adicionar usuários na lista de contas desativadas, adicione um lista separada por espaços dos usuários na linha **UserDir disabled**.

IMPORTANTE

Não remova a diretiva **IncludesNoExec**. Por padrão, o módulo *Server-Side Includes* (SSI) não pode executar comandos. É recomendado que você não mude estas definições a menos que seja absolutamente necessário, já que poderia potencialmente ativar um invasor para executar comandos no sistema.

2.2.6. Protegendo o FTP

O *File Transfer Protocol* (FTP) é um protocolo TCP antigo desenvolvido para transferir arquivos em uma rede. Pela razão que todas as transações com o servidor, incluindo a autenticação do usuário, não são criptografadas, é considerado um protocolo inseguro e deve ser configurado cuidadosamente.

O Red Hat Enterprise Linux fornece três servidores FTP.

- **gssftpd** — Um daemon de FTP baseado em **xinetd** e atento ao Kerberos que não transmite informações de autenticação na rede.
- **Red Hat Content Accelerator** (**tux**) — Um servidor Web do kernel com capacidades de FTP.
- **vsftpd** — Um implementação orientada à segurança, autônoma do serviço de FTP.

As seguintes diretrizes de segurança são para definir o serviço FTP **vsftpd**

2.2.6.1. Banner de Saudação do FTP

Antes de enviar um nome de usuário e senha, todos os usuários recebem um banner de saudação. Por padrão este banner inclui informações de versão úteis para os invasores que tentam identificar as fraquezas do sistema.

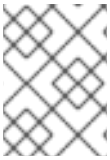
Para mudar o banner de saudação do **vsftpd**, adicione a seguinte diretiva ao arquivo **/etc/vsftpd/vsftpd.conf**:

```
ftpd_banner=<insert_greeting_here>
```

Substitua o *<insert_greeting_here>* na diretiva acima com o texto da mensagem de saudação.

Para banners com múltiplas linhas, é melhor usar um arquivo de banner. Para simplificar o gerenciamento de banners múltiplos, coloque todos os banners em um novo diretório chamado **/etc/banners/**. O arquivo de banner para conexões FTP neste exemplo é o **/etc/banners/ftp.msg**. Abaixo está um exemplo de como o arquivo deve ser:

```
######### # Hello, all activity on ftp.example.com is logged. #########
```



NOTA

Não é necessário iniciar cada linha do arquivo com **220** como especificado na Seção 2.2.1.1.1, “TCP Wrappers e Banners de Conexão”.

Para referenciar este arquivo de banner de saudação para o **vsftpd**, adicione a seguinte diretiva ao arquivo **/etc/vsftpd/vsftpd.conf**:

```
banner_file=/etc/banners/ftp.msg
```

É também possível enviar banners adicionais às conexões de entrada usando os TCP Wrappers conforme escrito Seção 2.2.1.1.1, “TCP Wrappers e Banners de Conexão”.

2.2.6.2. Acesso Anônimo

A presença do diretório **/var/ftp/** ativa a conta anônima.

A maneira mais fácil para criar este diretório é instalar o pacote **vsftpd**. Este pacote estabelece uma árvore de diretórios para usuários anônimos e configura as permissões nos diretórios para somente leitura para os usuários anônimos.

Por padrão o usuário anônimo não pode escrever em quaisquer diretórios.

ATENÇÃO

Se ativar o acesso anônimo a um servidor FTP, esteja atento onde os dados sensíveis estão armazenados.

2.2.6.2.1. Upload Anônimo

Para permitir usuários anônimos fazer upload de arquivos, é recomendado que um diretório de escrita somente seja criado dentro do **/var/ftp/pub/**.

Para fazer isso, digite o seguinte comando:

```
mkdir /var/ftp/pub/upload
```

Depois, mude as permissões para que então os usuários anônimos não possam ver os conteúdos dos diretórios:

```
chmod 730 /var/ftp/pub/upload
```

Um formato de listagem longa do diretório deve se parecer com:

```
drwx-wx---    2 root     ftp          4096 Feb 13 20:05 upload
```

ATENÇÃO

Administradores que permitem usuários anônimos ler e escrever nos diretórios muitas vezes possuem seus servidores transformados em um repositório de software roubado.

Adicionalmente, sob o **vsftpd**, adicione a seguinte linha ao arquivo **/etc/vsftpd/vsftpd.conf**:

```
anon_upload_enable=YES
```

2.2.6.3. Contas de Usuários

Por causa que o FTP transmite transmite nomes de usuários e senhas não criptografados por redes desprotegidas para autenticação, é uma boa idéia negar aos usuários do sistema o acesso ao servidor com suas contas de usuário.

Para desativar todas as contas de usuários no **vsftpd**, adicione a seguinte diretiva ao **/etc/vsftpd/vsftpd.conf**:

```
local_enable=NO
```

2.2.6.3.1. Restringindo Contas de Usuários

Para desativar o acesso FTP para contas específicas ou grupos de contas específicos, como o usuário root e aqueles com privilégios **sudo**, a maneira mais fácil para usar um arquivo de lista PAM conforme descrito na Seção 2.1.4.2.4, “Desativando o Root de usar o PAM”. O arquivo de configuração PAM para o **vsftpd** é **/etc/pam.d/vsftpd**.

É também possível desativar contas de usuários dentro de cada serviço diretamente.

Para desativar contas de usuários específicas no **vsftpd**, adicione o nome de usuário ao **/etc/vsftpd/ftpusers**

2.2.6.4. Usar o TCP Wrappers para Controla Acesso

Use o TCP Wrapper para controlar o acesso tanto do daemon FTP conforme descrito na Seção 2.2.1.1, “Aumentando a Segurança com TCP Wrappers”.

2.2.7. Protegendo o Sendmail

O Sendmail é um Mail Transfer Agent (MTA) que usa o Simple Mail Transfer Protocol (SMTP) para entregar mensagens eletrônicas entre outros MTAs e enviar emails para clientes ou agentes de entrega. Apesar de que muitos MTAs são capazes de criptografar o tráfego entre eles, a maioria não o faz, então enviar emails sobre quaisquer redes públicas é considerado um forma insegura de comunicação.

É recomendado que qualquer um planejando implementar um servidor Sendmail aborde as seguintes questões.

2.2.7.1. Limitando um DoS (Denial of Service Attack)

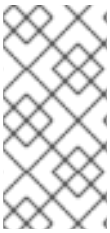
Devido à natureza do email, um determinado invasor pode causar um flood no servidor com emails bem facilmente e causar uma negação de serviço. Configurar limites nas seguintes diretivas no **/etc/mail/sendmail.mc**, a efetividade do tais ataques é limitada.

- **confCONNECTION_RATE_THROTTLE** — O número de conexões que o servidor pode receber por segundo. Por padrão, o Sendmail não limita o número de conexões. Se um limite é definido e alcançado, as próximas conexões são atrasadas.
- **confMAX_DAEMON_CHILDREN** — O número máximo de processos filhos que podem ser gerados pelo servidor. Por padrão, o Sendmail não atribui um limite ao número de processos filhos. Se um limite é definido e alcançado, as próximas conexões são atrasadas.
- **confMIN_FREE_BLOCKS** — O número mínimo de blocos livres que devem estar disponíveis para o servidor para aceitar mail. O padrão é 100 blocos.
- **confMAX_HEADERS_LENGTH** — O tamanho máximo aceitável (em bytes) para um cabeçalho de mensagens.
- **confMAX_MESSAGE_SIZE** — O tamanho máximo aceitável (em bytes) para uma mensagem única.

2.2.7.2. O NFS e Sendmail

Nunca coloque o diretório de mail spool, **/var/spool/mail/**, em um volume NFS compartilhado.

Por causa que o NFSv2 e NFSv3 não mantêm controle sobre as IDs de usuários e grupos, dois ou mais usuários podem ter o mesmo UID e receberem e lerem os emails um do outro.



NOTA

Com o NFSv4 usando o Kerberos, isto não acontece, já que o módulo do kernel **SECRPC_GSS** não utiliza uma autenticação baseada em UID. Entretanto, é ainda considerado uma boa prática *não* colocar o diretório de mail spool em volumes NFS não compartilhados.

2.2.7.3. Usuários somente de Mail

Para ajudar a impedir que um usuário local explore o servidor Sendmail, é melhor para os usuários de mail somente acessar o servidor Sendmail usando um programa de email. Contas shell no servidor de mail não devem ser permitidos e todos os usuários shell no arquivo **/etc/passwd** devem ser definidos para **/sbin/nologin** (com a exceção possível do usuário root).

2.2.8. Verificando Quais Portas Estão Escutando

Depois de configurar serviços de rede, é importante prestar atenção a quais portas estão na realidade escutando nas interfaces de rede do sistema. Quaisquer portas abertas podem ser evidências de uma invasão.

Existem duas abordagens básicas para listar as portas que estão escutando na rede. A abordagem menos confiável é solicitar a pilha de rede usando comandos como **netstat -an** ou **lsof -i**. Este método é menos confiável desde que estes programas não se conectam à máquina a partir da rede, mas é melhor verificar para ver o que está rodando no sistema. Por esta razão, estas aplicações são

alvos frequentes de substituições por invasores. Crackers tentam cobrir seus rastros se eles abrirem portas de rede não autorizadas substituindo o **netstat** e o **lsof** com suas próprias, versões modificadas.

Uma maneira mais confiável de checar quais portas estão escutando na rede é usar um escaner de porta como o **nmap**.

O seguinte comando emitido a partir do console determina quais portas estão escutando pelas conexões TCP da rede:

```
nmap -sT -O localhost
```

O resultado deste comando aparece como a seguir:

```
Starting Nmap 4.68 ( http://nmap.org ) at 2009-03-06 12:08 EST
Interesting ports on localhost.localdomain (127.0.0.1):
Not shown: 1711 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
111/tcp   open  rpcbind
113/tcp   open  auth
631/tcp   open  ipp
834/tcp   open  unknown
2601/tcp  open  zebra
32774/tcp open  sometimes-rpc11
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.17 - 2.6.24
Uptime: 4.122 days (since Mon Mar  2 09:12:31 2009)
Network Distance: 0 hops
OS detection performed. Please report any incorrect results at
http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.420 seconds
```

Este resultado mostra que o sistema está rodando o **portmap** por causa da presença do serviço **sunrpc** . Entretanto, há também um serviço misterioso na porta 834. Para checar se a porta é associada com o a lista oficial de serviços conhecidos, digite:

```
cat /etc/services | grep 834
```

Este comando retorna nenhum resultado para a porta 834. Devido ao formato do comando, o resultado para outras portas (1834, 2834, and 3834) serão mostrados. Isto indica que enquanto a porta 834 está na variação reservada (significando 0 até 1023) e requer o acesso root para abrir, ele não é associado com um serviço conhecido.

Depois, verifique por informações sobre a porta usando o **netstat** ou **lsof**. Para verificar pela porta 834 usando o **netstat**, use o seguinte comando:

```
netstat -anp | grep 834
```

O comando retorna o seguinte resultado:

```
tcp   0    0 0.0.0.0:834    0.0.0.0:*   LISTEN   653/ypbind
```

A presença da porta aberta no **netstat** é animadora por causa que se um cracker abrir uma porta clandestinamente em um sistema hackeado, não é susceptível de permitir que esta seja revelada através deste comando. Também, a opção **[p]** revela o ID do processo (PID) do serviço que abriu a porta. Neste caso, a porta aberta pertence ao **ypbind** (NIS), que é um serviço RPC manuseado em conjunto com o serviço **portmap**.

O comando **lsof** revela informações similares ao **netstat** já que ele é também capaz de ligar portas abertas à serviços:

```
lsof -i | grep 834
```

A porção relevante do resultado deste comando é:

```
ypbind      653        0    7u  IPv4       1319                 TCP
*:834 (LISTEN)
ypbind      655        0    7u  IPv4       1319                 TCP
*:834 (LISTEN)
ypbind      656        0    7u  IPv4       1319                 TCP
*:834 (LISTEN)
ypbind      657        0    7u  IPv4       1319                 TCP
*:834 (LISTEN)
```

Estas ferramentas revelam uma grande parte sobre o estado dos serviços rodando na máquina. Estas ferramentas são flexíveis e podem fornecer uma riqueza de informações sobre os serviços de rede e configuração. Consulte as páginas man do **lsof**, **netstat**, **nmap**, e **services** para mais informações.

2.3. TCP WRAPPERS E XINETD

Controlar o acesso aos serviços de rede é uma das tarefas de segurança mais importantes para um administrador de servidor. O Red Hat Enterprise Linux fornece diversas ferramentas para este propósito. Por exemplo, um firewall baseado em **iptables** filtra pacotes de rede indesejados dentro da pilha de rede do kernel. Para serviços de rede que os utilizam, os *TCP Wrappers* adicionam uma camada a mais de proteção definindo quais hosts são ou não são permitidos conectar aos serviços de rede "*envolvidos (wrapped)*". Um tipo de serviço de rede wrapped é o **xinetd** *super server*. O serviço é chamado super server porque ele controla conexões de um subconjunto de serviços de rede e refina ainda mais o controle de acesso.

A Figura 2.4, “Controle de Acesso para Serviços de Rede” é uma ilustração básica de como estas ferramentas trabalham juntas para proteger os serviços de rede.

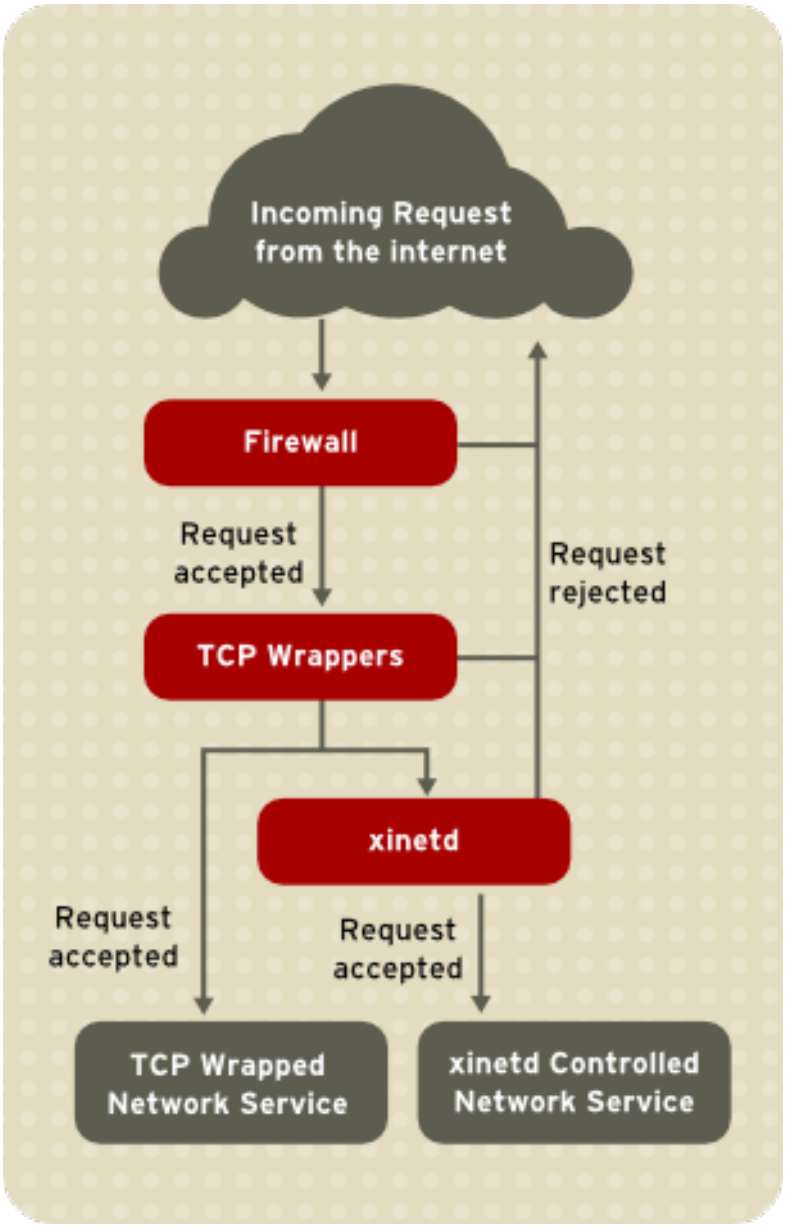


Figura 2.4. Controle de Acesso para Serviços de Rede

Este capítulo foca na função de TCP Wrappers e **xinetd** no controle de acesso aos serviços de rede e revisa como estas ferramentas podem ser usadas para melhorar tanto a autenticação e gerenciamento de utilização. Consulte a Seção 2.6, "IPTables" para informações sobre usar firewalls com **iptables**.

2.3.1. TCP Wrappers

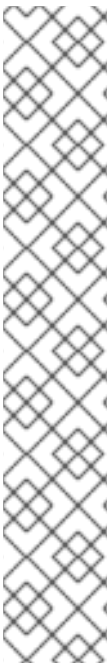
Os pacotes de TCP Wrappers (**tcp_wrappers** e **tcp_wrappers-libs**) são instalados por padrão e fornecem controle de acesso baseado no host para serviços de rede. O componente mais importante do pacote é a biblioteca **/lib/libwrap.a** ou **/lib64/libwrap.a**. Em termos gerais, um serviço TCP wrapped é um que foi compilado para a biblioteca **libwrap.a**.

Quando uma tentativa de conexão é feita a um serviço TCP Wrapper, o serviço primeiro referencia os arquivos de acesso ao host (**/etc/hosts.allow** e **/etc/hosts.deny**) para determinar se o cliente é permitido ou não se conectar. Na maioria dos casos, ele então usa o syslog daemon (**syslogd**) para gravar o nome do cliente solicitante e do serviço solicitado em **/var/log/secure** ou **/var/log/messages**.

Se um cliente é permitido se conectar, os TCP Wrappers liberam o controle da conexão para o serviço solicitado e não participa mais na comunicação entre o cliente e o servidor.

Além de controlar o acesso e autenticação, os TCP Wrappers podem executar comandos para interagir com o cliente antes de negar ou liberar controle da conexão ao serviço de rede solitictado.

Pela razão que os TCP Wrappers são uma adição valiosa a qualquer arsenal de ferramentas de segurança de um administrador de servidor, a maiorias dos serviços de rede dentro do Red Hat Enterprise Linux são ligados à biblioteca **`libwrap.a`**. Algumas dessas aplicações incluem **`/usr/sbin/sshd`**, **`/usr/sbin/sendmail`**, and **`/usr/sbin/xinetd`**.



NOTA

Para determinar se um serviço de rede binária está ligado ao **`libwrap.a`**, digite o seguinte comando como usuário root:

```
ldd <binary-name> | grep libwrap
```

Substitua o *<binary-name>* com o nome do serviço de rede binário.

Se o comando retornar diretamente ao prompt sem nenhum resultado, então o serviço de rede *não* é ligado ao **`libwrap.a`**.

O exemplo seguinte indica que o **`/usr/sbin/sshd`** está ligado ao **`libwrap.a`**:

```
[root@myServer ~]# ldd /usr/sbin/sshd | grep libwrap
        libwrap.so.0 => /lib/libwrap.so.0 (0x00655000)
[root@myServer ~]#
```

2.3.1.1. Vantagens do TCP Wrappers

Os TCP Wrappers fornecem as seguintes vantagens sobre outras técnicas de controle de serviço de rede:

- *Transparências tanto aos clientes e serviços de rede wrapped* — Ambos clientes em conexão e o serviço de rede wrapped não estão atentos que os TCP Wrappers estão em uso. Usuários legítimos são autenticados e conectados ao serviço solicitado enquanto conexões de clientes banidos falham.

- *Gerenciamento Centralizado de múltiplos protocolos* — Os TCP Wrappers operam separadamente dos serviços de rede que protegem, permitindo muitas aplicações de servidor compartilhar um conjunto comum de arquivos de configurações de controle de acesso, para um gerenciamento mais simples.

2.3.2. Arquivos de Configuração dos TCP Wrappers

Para determinar se um cliente é permitido se conectar a um serviço, os TCP Wrappers referenciam os seguintes dois arquivos, que são comumente referidos como arquivos *de acesso de hosts*:

- **`/etc/hosts.allow`**

- **`/etc/hosts.deny`**

Quando um serviço TCP wrapped recebe uma solicitação de um cliente, ele realiza os seguintes passos:

1. *Ele faz referencia ao* ***/etc/hosts.allow*** — O serviço TCP wrapper analisa sequencialmente o arquivo **/etc/hosts.allow** e aplica a primeira regra especificada para aquele serviço. Se ele encontra uma regra correspondente, ele permite a conexão. Se não, vai para o próximo passo.

2. *Ele faz referência ao* ***/etc/hosts.deny*** — O serviço TCP wrapper analisa sequencialmente o arquivo **/etc/hosts.deny**. Se ele encontrar uma regra correspondente, ele nega a conexão. Se não, garante acesso ao serviço.

A seguir estão pontos importantes a considerar quando usar TCP Wrappers para proteger serviços de rede:

- Como as regras de acesso no **hosts.allow** são aplicadas primeiro, elas têm preferência sobre as regras especificadas no **hosts.deny**. Portanto, se o acesso a um serviço é permitido no **hosts.allow**, uma regra negando acesso ao mesmo serviço **hosts.deny** é ignorado.

- As regras em cada arquivo são lidas de cima para baixo e a primeira regra correspondente para um determinado serviço é a única aplicada. A ordem das regras é extremamente importante.

- Se nenhuma regra para o serviço é encontrada em qualquer arquivo ou nenhum arquivo existe, acesso ao serviço é garantido.

- Os serviços TCP wrappers não fazem cache das regras dos arquivos de acesso dos hosts, então quaisquer mudanças no **hosts.allow** ou **hosts.deny** acontecem imediatamente, sem reiniciar os serviços de rede.

ATENÇÃO

Se a última linha do arquivo de acesso ao host não é um carácter de nova linha (criado ao se pressionar o **Enter**), a última regra no arquivo falha e um erro é registrado tanto no logs **/var/log/messages** ou **/var/log/secure**. Isto é também o caso para uma regra que se extende por múltiplas linhas sem usar a barra invertida. O seguinte exemplo ilustra a porção relevante de uma mensagem de log para uma falha de regra devido a uma dessas circunstâncias:

```
warning: /etc/hosts.allow, line 20: missing newline or line
too long
```

2.3.2.1. Formatando Regras de Acesso

Os formatos para ambos **/etc/hosts.allow** e **/etc/hosts.deny** são idênticos. Cada regra deve estar em sua própria linha. Linhas em branco ou linhas que iniciam com um hash (#) são ignoradas.

Cada regra usa o seguinte formato básico para controlar acesso aos serviços de rede:

```
<daemon list>: <client list> [: <option>: <option>: ...]
```

- *<daemon list>* — Uma lista separada por vírgula de nomes de processos (*não* os nomes dos serviços) ou o carácter coringa **ALL**. A lista daemon também aceita operadores (consulte a Seção 2.3.2.1.4, “Operadores”) para permitir maior flexibilidade.

- *<client list>* — Uma lista separada por vígula de nomes de host, endereços IP de host, modelos especiais ou carácteres coringa que identificam os hosts afetados pela regra. A lista de clientes também aceita operadores listados na Seção 2.3.2.1.4, “Operadores” para permitir maior flexibilidade.

- *<option>* — Uma ação opcional ou lista separada por dois pontos de ações realizadas quando a regra é acionada. Campos opcionais suportam expansões, dão inicio à comandos shell, permitem ou negam acesso e alteram o comportamento de logging.

NOTA

Mais informações sobre alguns dos termos acima podem ser encontrados em outras partes deste guia:

- Seção 2.3.2.1.1, “Carácteres Coringa (wildcards)”
- Seção 2.3.2.1.2, “Modelos”
- Seção 2.3.2.2.4, “Expansões”
- Seção 2.3.2.2, “Campos de Opção”

A seguir há um exemplo básico de regra de acesso ao host:

```
vsftpd : .example.com
```

Esta regra instrui o TCP Wrapper para aguardar por conexões do daemon FTP (**vsftpd**) de qualquer host no domínio **example.com**. Se esta regra aparecer no **hosts.allow**, a conexão é aceita. Se esta regra aparecer no **hosts.deny**, a conexão é rejeitada.

O próximo exemplo de regra de acesso à host é mais complexa e usa dois campos de opções:

```
sshd : .example.com  \ : spawn /bin/echo `/bin/date` access
denied>>/var/log/sshd.log \ : deny
```

Note que cada campo de opção é precedido com uma barra inversa (\). O uso da barra invertida previne falha da regra devido ao comprimento.

Este exemplo de regra declara que se uma conexão ao daemon SSH (**sshd**) é tentada a partir de um host no domínio **example.com**, executa o comando **echo** para anexar a tentativa ao arquivo de log especial e nega a conexão. Pelo motivo que a diretiva opcional **deny** é usada, esta linha nega o acesso mesmo se ele aparecer no arquivo **hosts.allow**. Consulte a Seção 2.3.2.2, “Campos de Opção” para uma visão mais detalhada das opções disponíveis.

2.3.2.1.1. Carácteres Coringa (wildcards)

Carácteres coringa permitem aos TCP Wrappers corresponder mais facilmente grupos de daemons ou hosts. Eles são usados mais frequentemente no campo de lista de clientes das regras de acesso.

Os seguintes carácteres coringa estão disponíveis:

- **ALL** — Corresponde a tudo. Ele pode ser usado para ambas listas de daemons e lista de clientes.
- **LOCAL** — Corresponde a qualquer host que não contém um ponto (.), como o localhost.
- **KNOWN** — Corresponde a qualquer host onde o hostname e endereço de host são conhecidos ou onde o usuário é conhecido.
- **UNKNOWN** — Corresponde a qualquer host onde o hostname ou endereço de host são desconhecidos ou onde o usuário é desconhecido.
- **PARANOID** — Corresponde a qualquer host onde o hostname não corresponde ao endereço de host.

IMPORTANTE

Os coringas **KNOWN**, **UNKNOWN** e **PARANOID** devem ser usados com cuidado, por causa que eles dependem em um servidor DNS em funcionamento para que a operação seja correta. Qualquer disruptura na resolução de nome pode impedir usuários legítimos de ganhar acesso ao serviço.

2.3.2.1.2. Modelos

Modelos podem ser usados no campo de cliente das regras de acesso para mais precisamente especificar grupos de hosts clientes.

A seguir há uma lista de modelos comuns para serem usadas no campo do cliente:

- *Hostname iniciando com ponto (.)* — Colocar um ponto no início de um hostname corresponde a todos os hosts compartilhando os componentes listados do nome. O exemplo a seguir se aplica a qualquer host dentro do domínio **example.com**:

```
ALL : .example.com
```

- *Endereço de IP terminando com um ponto (.)* — Colocar um ponto no final do endereço IP corresponde a todos os hosts compartilhando os grupos numéricos iniciais de um endereço de IP. O exemplo seguinte se aplica a qualquer host dentro da rede **192.168.x.x**:

```
ALL : 192.168.
```

- *endereço de IP /par máscara de rede* — Expressões de máscara de rede podem também ser usadas como um modelo para controlar acesso a um grupo particular de endereços IP. O exemplo seguinte se aplica a qualquer host com uma variação de endereço de **192.168.0.0** até **192.168.1.255**:

```
ALL : 192.168.0.0/255.255.254.0
```

IMPORTANTE

Quando trabalhar com espaço de endereço IPv4, a extensão do endereço/prefixo (*prefixlen*) declarações de par (notação CIDR) não são suportadas. Somente regras IPv6 podem usar este formato.

- *[endereço IPv6]/par prefixlen* — pares [net]/prefixlen podem também ser usados como um modelo para controlar acesso a um grupo particular de endereços IPv6. O exemplo seguinte se aplicaria a qualquer host com uma faixa de **`3ffe:505:2:1::`** até **`3ffe:505:2:1:ffff:ffff:ffff:ffff`**:

```
ALL : [3ffe:505:2:1::]/64
```

- *O asterisco (*)* — Asteriscos podem ser usados para corresponder grupos inteiros de hostnames ou endereços de IP, desde que eles não sejam misturados em uma lista de clientes contendo outros tipos de modelos. O exemplo seguinte se aplicaria a qualquer host dentro do domínio **`example.com`**:

```
ALL : *.example.com
```

- *A barra (/)* — Se uma lista de clientes iniciar com uma barra (/), ela é tratada como um nome de arquivo. Isto é útil se regras especificando números grandes de hosts são necessárias. O exemplo seguinte se refere aos TCP Wrappers para o arquivo **`/etc/telnet.hosts`** para todas as conexões Telnet:

```
in.telnetd : /etc/telnet.hosts
```

Outro exemplo, modelos menos usados também são aceitos pelos TCP Wrappers. Consulte a página man 5 **`hosts_access`** para mais informações.

ATENÇÃO

Seja cuidadoso quando usar hostnames e nomes de domínio. Invasores podem usar uma variedade de truques para burlar uma resolução de nomes precisa. Além disso, disruptura no serviço DNS impede que mesmo usuários autorizados usem os serviços de rede. É melhor, portanto usado endereços de IP sempre que possível.

2.3.2.1.3. Portmap e TCP Wrappers

A implementação do **`Portmap`** de TCP Wrappers não suporta busca de hosts, que siginifca que o **`portmap`** não pode usar hostnames para identificar hosts. Consequentemente, regras de controle de acesso para portmap no **`hosts.allow`** ou **`hosts.deny`** devem usar endereços de IP ou a palavra chave **`ALL`**, para especificar os hosts.

Mudanças nas regras de controle de acesso do **`portmap`** podem não ter efeito imediatamente. Você pode necessitar reiniciar o serviço **`portmap`**.

Serviços usados amplamente, como NIS e NFS, dependem do **`portmap`** para operar, então esteja atento à estas limitações.

2.3.2.1.4. Operadores

No presente, regras de controle de acesso aceitam um operador, o **`EXCEPT`**. Ele pode ser usado em ambas listas daemon e listas clientes para uma regra.

O operador **EXCEPT** permite exceções específicas para abranger correspondências com a mesma regra.

No exemplo seguinte de um arquivo **hosts.allow**, todos os hosts **example.com** são permitidos se conectarem a todos os serviços, exceto **cracker.example.com**:

```
ALL: .example.com EXCEPT cracker.example.com
```

Em um outro exemplo de um arquivo **hosts.allow**, clientes da rede **192.168.0.*x*** podem usar todos os serviços exceto para o FTP:

```
ALL EXCEPT vsftpd: 192.168.0.
```

NOTA

Organizacionalmente, é muito mais fácil evitar os operadores **EXCEPT**. Isto permite a outros administradores rapidamente escanear os arquivos apropriados para ver quais hosts tem acesso permitido ou negado aos serviços sem ter de classificar operadores **EXCEPT**.

2.3.2.2. Campos de Opção

Além das regras básicas que permitem e negam os acessos, a implementação Red Hat Enterprise Linux dos TCP Wrappers suportam extensões à linguagem de controle de acesso pelos *campos de opções*. Usando estes campos de opções em regras de acesso aos hosts, os administradores podem realizar uma variedade de tarefas como alterar o comportamento do log, consolidar controle de acesso e realizar comandos shell.

2.3.2.2.1. Registro de Logs

Campos de opção permitem aos administradores facilmente mudar a facilidade de log e nível de prioridade para uma regra usando a diretiva **severity**.

No exemplo seguinte, conexões ao daemon SSH de qualquer host no domínio **example.com** são registradas à facilidade **authpriv syslog** (por causa que nenhum valor de facilidade é especificado) com uma prioridade de **emerg**:

```
sshd : .example.com : severity emerg
```

Também é possível especificar uma facilidade usando a opção **severity**. O seguinte exemplo registra em log qualquer tentativa de conexão SSH por hosts do domínio **example.com** à facilidade **local0** com a prioridade de **alert**:

```
sshd : .example.com : severity local0.alert
```

NOTA

Na prática, este exemplo não funciona até que o syslog daemon (**syslogd**) esteja configurado para registrar no log a facilidade **local0**. Consulte a página man **syslog.conf** para informações sobre configurar facilidades de log personalizadas.

2.3.2.2.2. Controle de Acesso

Campos de Opções também permitem administradores permitir ou negar hosts explicitamente em uma regra simples adicionando a diretiva **allow** ou **deny** como a opção final.

Por exemplo, as seguintes duas regras permitem conexões SSH a partir do **client-1.example.com**, mas negam conexões do **client-2.example.com**:

```
sshd : client-1.example.com : allow
sshd : client-2.example.com : deny
```

Permitindo controle de acesso numa base por regra, o campo de opção permite administradores consolidar todas as regras de acesso em um arquivo único: tanto **hosts.allow** ou **hosts.deny**. Alguns administratores consideram isso uma maneira fácil de organizar regras de acesso.

2.3.2.2.3. Comandos Shell

Campos de Opção permitem regras de acesso para realizar comandos shell pelas duas diretivas a seguir:

- **spawn** — Realiza um comando shell como um processo filho. Esta diretiva pode realizar tarefas como usar o **/usr/sbin/safe_finger** para obter mais informações sobre o cliente solicitante ou criar arquivos especiais de log usando o comando **echo**.

  No exemplo seguinte, clientes que tentam acessar os serviços Telnet a partir do domínio **example.com** são silenciosamente registrados ao log de um arquivo especial:

  ```
  in.telnetd : .example.com \
   : spawn /bin/echo `/bin/date` from %h>>/var/log/telnet.log \
   : allow
  ```

- **twist** — Substitui o serviço solicitado com o comando especificado. Esta diretiva é frequentemente usada para configurar armadilhas para invasores (também chamados "honey pots" (potes de mel). Ele também pode ser usado para enviar mensagens para clientes em conexão. A diretiva **twist** deve ocorrer no final da linha da regra.

  No exemplo seguinte, clientes que tentam acessar os serviços FTP do domínio **example.com** recebem uma mensagem usando o comando **echo**:

  ```
  vsftpd : .example.com \
   : twist /bin/echo "421 This domain has been black-listed. Access
  denied!"
  ```

Para mais informações sobre opções do comando shell, consulte a página man **hosts_options**.

2.3.2.2.4. Expansões

As expansões, quando usadas em conjunto com as diretivas **spawn** e **twist**, fornecem informações sobre o cliente, servidor e processos envolvidos.

A seguir está uma lista de expansões suportadas:

- **%a** — Retorna o endereço IP do cliente.

- **%A** — Retorna o endereço IP do servidor.
- **%c** — Retorna uma variedade de informações dos clientes, como o nome de usuário e hostname ou o nome de usuário e endereço IP.
- **%d** — Retorna o nome do processo daemon.
- **%h** — Retorna o hostname do cliente (ou endereço IP, se o hostname estiver indisponível).
- **%H** — Retorna o hostname do servidor (ou endereço IP, se o hostname estiver indisponível).
- **%n** — Retorna o hostname do cliente. Se indisponível, **unknown** é mostrado. Se o hostname do cliente e endereço de host não correpondem, **paranoid** é mostrado.
- **%N** — Retorna o hostname do servidor. Se indisponível, **unknown** é mostrado. Se o hostname do servidor e endereço de host não correspondem, **paranoid** é mostrado.
- **%p** — Retorna o ID do processo daemon.
- **%s** — Retorna vários tipos de informações do servidor, como processo daemon e o host ou endereço IP do servidor.
- **%u** — Retorna o nome de usuário. Se indisponível, **unknown** é mostrado.

O seguinte exemplo de regra usa uma expansão em conjunção com o comando **spawn** para identificar o host do cliente em um arquivo de log personalizado.

Quando conexões ao daemon SSH (**sshd**) são tentadas de um host no domínio **example.com** , execute o comando **echo** para registrar as tentativas, incluindo o hostname do cliente (usando a expansão **%h**), a um arquivo especial:

```
sshd : .example.com  \
 : spawn /bin/echo `/bin/date` access denied to %h>>/var/log/sshd.log \
 : deny
```

Similarmente, as expansões podem ser usadas para personalizar mensagens de volta ao cliente. No exemplo seguinte, clientes que tentam acessar os serviços FTP do domínio **example.com** são informados que eles foram banidos do servidor:

```
vsftpd : .example.com \
: twist /bin/echo "421 %h has been banned from this server!"
```

Para uma explicação completa das expansões disponíveis, tanto como opções de controle de acesso, consulte a seção 5 das páginas man para **hosts_access** (**man 5 hosts_access**) e a página man para **hosts_options**.

Consulte a Seção 2.3.5, “Recursos Adicionais” para mais informações sobre TCP Wrappers.

2.3.3. xinetd

O daemon **xinetd** é um *super serviço* do TCP Wrapper que controla acesso de um sub conjuntos de serviços de redes populares, incluindo FTP, IMAP e Telnet. Ele também fornece opções de configuração especificas de serviço para controle de acesso, registro de log melhorados, vinculação, redirecionamento, e controle de utilização de recursos.

Quando um cliente tentar se conectar a uma rede de serviços controlados pelo **xinetd**, o super serviço recebe o pedido e verifica por quaisquer regras de controle de acesso dos TCP Wrappers.

Se o acesso é permitido, o **xinetd** verifica que a conexão é permitida sobre suas próprias regras de acesso para esse serviço. Ele também verifica que o serviço é capaz de ter mais recursos atribuídos a ele e não quebra nenhuma das regras definidas.

Se todas estas condições são atendidas (que significa, o acesso é permitido ao serviço; o serviço não atingiu seu limite de recursos; e o serviço não quebrou nenhuma das regras definidas), o **xinetd** então inicia uma instância do serviço solicitado e passa o controle da conexão à ele. Depois da conexão ter sido estabelecida, o **xinetd** não faz mais parte da comunicação entre o cliente e o servidor.

2.3.4. Arquivos de Configuração xinetd

Os arquivos de configuração do **xinetd** são como a seguir:

- **/etc/xinetd.conf** — O arquivo de configuração global do **xinetd**.
- **/etc/xinetd.d/** — O diretório contendo todos os arquivos especificos do serviço.

2.3.4.1. O arquivo /etc/xinetd.conf

O arquivo **/etc/xinetd.conf** contém definições de configurações gerais que afetam todos os serviços sob o controle do **xinetd**. Ele é lido quando o serviço **xinetd** é primeiramente iniciado, então para as mudanças de configuração terem efeito, você precisa reiniciar o serviço **xinetd**. O seguinte é um modelo do arquivo **/etc/xinetd.conf**:

```
defaults
{
  instances               = 60
  log_type                = SYSLOG authpriv
  log_on_success          = HOST PID
  log_on_failure          = HOST
  cps                     = 25 30
}
includedir /etc/xinetd.d
```

Estas linhas controlam os seguintes aspectos do **xinetd**:

- **instances** — Especifica o número máximo de pedidos simultâneos que o **xinetd** pode processar.
- **log_type** — Configura o **xinetd** para usar a facilidade de log **authpriv**, que grava entradas no log para o arquivo **/var/log/secure**. Adicionar uma diretiva como **FILE /var/log/xinetdlog** criaria um arquivo de log personalizado chamado **xinetdlog** no diretório **/var/log/**.
- **log_on_success** — Configura o **xinetd** para registrar no log tentativas de conexão com sucesso. Por padrão, o endereço IP do host remoto e o ID de processo do servidor que processo o pedido são gravados,
- **log_on_failure** — Configura o **xinetd** para logar tentativas de conexão com falhas ou se a conexão foi negada.

- **cps** — Configura o **xinetd** para permitir não mais que 25 conexões por segundo para qualquer serviço dado. Se este limite é excedido, o serviço é suspenso por 30 segundos.
- **includedir /etc/xinetd.d/** — Inclui opções declaradas nos arquivos de configuração de serviço específico localizados no diretório **/etc/xinetd.d/**. Consulte a Seção 2.3.4.2, “O Diretório /etc/xinetd.d/” para mais informações.

NOTA

Muitas vezes, as configurações do **log_on_success** e **log_on_failure** no **/etc/xinetd.conf** são também modificadas nos arquivos de configuração de serviços específicos. Mais informações podem entretanto aparecer no arquivo de log do serviço do que o **/etc/xinetd.conf** pode indicar. Consulte a Seção 2.3.4.3.1, “Opções de Registro de Log” para mais informações.

2.3.4.2. O Diretório /etc/xinetd.d/

O diretório **/etc/xinetd.d/** contém os arquivos de configuração para cada serviço gerenciado pelo **xinetd** e os nomes dos arquivos são correlacionados ao serviço. Como ocorre com o **xinetd.conf**, este diretório é somente leitura quando o serviço é iniciado. Para quaisquer mudanças terem efeito, o administrador deve reiniciar o serviço **xinetd**.

O formato dos arquivos no diretório **/etc/xinetd.d/** usam as mesmas convenções como o **/etc/xinetd.conf**. A razão primária que a configuração para cada serviço é armazenada em um arquivo separado é fazer a personalização mais fácil e menos suscetíveis de afetar outros serviços.

Para obter um melhor entendimento de como estes arquivos são estruturados, considere o arquivo **/etc/xinetd.d/krb5-telnet**:

```
service telnet
{
  flags           = REUSE
  socket_type     = stream
  wait            = no
  user            = root
  server          = /usr/kerberos/sbin/telnetd
  log_on_failure  += USERID
  disable         = yes
}
```

Estas linhas controlam vários aspectos do serviço **telnet**:

- **service** — Especifica o nome do serviço, normalmente um dos listados no arquivo **/etc/services**.
- **flags** — Define um número qualquer de atributos para a conexão. O **REUSE** instrui o **xinetd** para reusar p socket para a conexão Telnet.

NOTA

O sinalizador **REUSE** está obsoleto. Todos os serviços agora usam implicitamente o sinalizador **REUSE**.

- **socket_type** — Define o tipo de socket de rede para **stream**.
- **wait** — Especifica se o serviço é single-threaded (**yes**) ou multi-threaded (**no**).
- **user** — Especifica qual ID de usuário o processo o rodará.
- **server** — Especifica qual binário executável a dar início.
- **log_on_failure** — Especifica os parâmetros de registro de log para o **log_on_failure** além daqueles já definidos em **xinetd.conf**.
- **disable** — Especifica se o serviço está inativo (**yes**) ou ativo (**no**).

Consulte a página man **xinetd.conf** para mais informações sobre estas opções e seu uso.

2.3.4.3. Aterando Arquivos de Configuração xinetd

Uma variedade de diretivas está disponível para serviços protegidos pelo **xinetd**. Esta seção destaca algumas das opções mais comumente usadas.

2.3.4.3.1. Opções de Registro de Log

As seguintes opções de log estão disponíveis para ambos **/etc/xinetd.conf** e os arquivos específicos de serviço dentro do diretório **/etc/xinetd.d/**.

A seguir está uma lista de algumas das opções usadas mais comuns para registro de log:

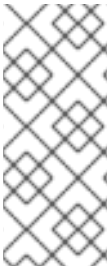
- **ATTEMPT** — Registra em log que uma tentativa com falha foi feita (**log_on_failure**).
- **DURATION** — Registra em log o período de tempo que o serviço é usado por um sistema remoto (**log_on_success**).
- **EXIT** — Registra em log o estado de saída ou sinal de término do serviço (**log_on_success**).
- **HOST** — Registra em log o endereço de IP do host remoto (**log_on_failure** e **log_on_success**).
- **PID** — Registra em log o ID do processo do servidor recebendo o pedido (**log_on_success**).
- **USERID** — Registra em log o usuário remoto usando o método definido em RFC 1413 para todos os serviços de stream multi-threaded (**log_on_failure** e **log_on_success**).\n\t\n

Para uma lista completa de opções de log, consulte a página man **xinetd.conf**.

2.3.4.3.2. Opções de Controle de Acesso

Usuários dos serviços **xinetd** podem escolher usar as regras de acesso de hosts TCP Wrappers, fornecer controle de acesso pelos arquivos de configuração **xinetd** ou uma mistura de ambos. Consulte a Seção 2.3.2, “Arquivos de Configuração dos TCP Wrappers” para mais informações sobre os arquivos de controle de acesso de hosts TCP Wrappers.

Esta seção discute o uso do **xinetd** para controlar o acesso aos serviços.



NOTA

Diferente dos TCP Wrappers, mudanças nos controle de acesso somente tem efeito se o administrador **xinetd** reiniciar o serviço **xinetd**.

Também, diferentemente dos TCP Wrappers, o controle de acesso pelo **xinetd** somente afeta serviços controlados pelo **xinetd**.

O controle de acesso de hosts **xinetd** difere do método usado pelos TCP Wrappers. Enquanto os TCP Wrappers colocam toda a configuração de acesso dentro de dois arquivos **/etc/hosts.allow** e **/etc/hosts.deny**, o controle de acesso do **xinetd** é encontrado em cada um dos arquivos de configuração do serviço no diretório **/etc/xinetd.d/**.

As seguintes opções de acesso de hosts são suportadas pelo **xinetd**:

- **only_from** — Permite somente os hosts especificados usarem o serviço.
- **no_access** — Bloqueia os hosts listados de usar o serviço.
- **access_times** — Especifica o período de tempo quando um determinado serviço pode ser usado. O período de tempo deve ser declarado no formato 24 horas, HH:MM-HH:MM.

As opções **only_from** e **no_access** podem usar uma lista de endereços de IP ou nomes de host, ou pode especificar uma rede inteira. Como os TCP Wrappers, combinar o controle de acesso **xinetd** com a configuração de log avançada pode aumentar a segurança bloqueando pedidos dos hosts banidos enquanto grava a verbosidade de cada tentativa de conexão.

Por exemplo, o arquivo seguinte **/etc/xinetd.d/telnet** pode ser usado para bloquear o acesso Telnet de um grupo de rede particular e restringe o intervalo de tempo de total que mesmo usuários permitidos podem se logar:

```
service telnet
{
  disable         = no
  flags           = REUSE
  socket_type     = stream
  wait            = no
  user            = root
  server          = /usr/kerberos/sbin/telnetd
  log_on_failure  += USERID
  no_access       = 172.16.45.0/24
  log_on_success  += PID HOST EXIT
  access_times    = 09:45-16:15
}
```

Neste exemplo, quando um sistema cliente da rede **172.16.45.0/24**, como **172.16.45.2**, tenta acessar o serviço Telnet, ele recebe a seguinte mensagem:

```
Conexão fechada por um host externo.
```

Alem disso, as tentativas de login deles são registradas no **/var/log/messages** conforme a seguir:

```
Sep  7 14:58:33 localhost xinetd[5285]: FAIL: telnet address
from=172.16.45.107
```

```
Sep  7 14:58:33 localhost xinetd[5283]: START: telnet pid=5285
from=172.16.45.107
Sep  7 14:58:33 localhost xinetd[5283]: EXIT: telnet status=0 pid=5285
duration=0(sec)
```

Quando usar os TCP Wrappers em conjunto com os controles de acesso **xinetd** , é importante entender o relacionamento entre os dois mecanismos de controle de acesso.

A seguir está uma sequencia de eventos seguidos pelo **xinetd** quando um cliente solicita uma conexão:

1. O daemon **xinetd** acessa as regras de acesso de hosts do TCP Wrappers usando uma chamada da biblioteca **libwrap.a**. Se uma regra de negação corresponde ao cliente, a conexão é passada para o **xinetd**.

2. O daemon **xinetd** verifica suas próprias regras de controle de acesso tanto para o serviço **xinetd** e o serviço solicitado. Se uma regra de negação corresponde ao cliente, a conexão é despejada. Caso contrário , o **xinetd** inicia uma instância do serviço solicitado e passa o controle da conexão a esse serviço.

IMPORTANTE

Cuidado deve ser tomado quando usar os controles de acesso do TCP Wrappers em conjunto com os controles de acesso **xinetd**. Um erro de configuração pode causar efeitos indesejáveis,

2.3.4.3.3. Opções de Associação e Redirecionamento

Os arquivos de configuração do serviço para o **xinetd** suporta a associação do serviço a um endereço de IP e redireciona pedidos de entrada desse serviço para outro endereço IP, hostname ou porta.

A associação é controlada com o a opção **bind** nos arquivos de configuração de serviço específicos e liga o serviço a um endereço IP no sistema. Quando este é configurado, a opção **bind** somente permite pedidos ao endereço de IP correto para acessar o serviço. Você pode usar este método para associar diferentes serviços à interfaces de redes diferentes baseadas em requerimentos.

Isto é particularmente útil para sistemas com múltiplos adaptadores de rede ou com múltiplos endereços IP. Em tal sistema, serviços inseguros (por exemplo, Telnet), pode ser configurado para escutar somente à interface conectada à uma rede privada e não à interface conectada à internet.

A opção **redirect** aceita um endereço IP ou hostname seguido por um número de porta. Ela configura o serviço para redirecionar quaisquer pedidos para este serviço ao host e número de porta especificados. Este recurso pode ser usado para apontar para um outro número de porta no mesmo sistema, redirecionar o pedido para um endereço IP diferente na mesma máquina, deslocar o pedido para um sistema e número de porta totalmente diferentes, ou qualquer combinação destas opções. Um usuário conectando a um certo serviço em um sistema pode portanto ser reencaminhado a um outro sistema sem ruptura.

O daemon **xinetd** é capaz de realizar este redirecionamento reproduzindo um processo que permanece vivo pela duração da conexão entre a máquina cliente solicitante e o host que fornece o serviço, transferindo dados entre os dois sistemas.

As vantagens das opções **bind** e **redirect** são mais claramente evidentes quando elas são usadas em conjunto. Associando um serviço à um endereço IP determinado em um sistema e então redirecionar os pedidos para este serviço a uma segunda máquina que somente a primeira máquina

pode ver, um sistema interno pode ser usado para fornecer serviços para uma rede totalmente diferente. Alternativamente, estas opções podem ser usadas para limitar a exposição de um determinado serviço em uma máquina multihome para endereços IP conhecidos, tanto quanto redireciona quaisquer pedidos para esse serviço para uma outra máquina especialmente configurada para este propósito.

Por exemplo, considere um sistema que é usado como um firewall com estas configurações para o serviço Telnet:

```
service telnet
{
  socket_type  = stream
  wait   = no
  server   = /usr/kerberos/sbin/telnetd
  log_on_success  += DURATION USERID
  log_on_failure  += USERID
  bind                  = 123.123.123.123
  redirect              = 10.0.1.13 23
}
```

As opções **bind** e **redirect** neste arquivo garantem que o serviço Telnet nesta máquina é ligado a um endereço IP externo (**123.123.123.123**), o que está na internet. Além disso, qualquer pedido pelo serviço Telnet enviado ao **123.123.123.123** é redirecionado pelo segundo adaptador de rede para um endereço Ip interno (**10.0.1.13**) que somente o firewall e sistemas internos podem acessar. O firewall então envia a comunicação entre os dois sistemas e o sistema em conexão pensa que está conectado ao **123.123.123.123** quando na verdade está conectado a uma máquina diferente.

Este recurso é particularmente útil para usuários com conexões de banda larga e somente um endereço IP fixo. Quando usar o Network Address Translation (NAT), os sistemas por trás da máquina de gateway, que estão usando endereços de IP somente internos, não estão disponíveis fora do sistema de gateway. Entretanto, quando certos serviços controlados pelo **xinetd** são configurados com as opções **bind** e **redirect**, a máquina gateway pode agir como uma proxy entre sistemas do lado de fora e uma máquina interna particular configurada para fornecer o serviço. Além disso, os vários controles de acesso **xinetd** e opções de log estão também disponíveis para proteção adicional.

2.3.4.3.4. Opções de Gerenciamento de Recursos

O daemon **xinetd** pode adicionar um nível básico de proteção contra ataques Dos (Denial of Service). A seguir está uma lista de diretivas que podem auxiliar na limitação da efetividade desses ataques:

- **per_source** — Define o número máximo de instâncias de um serviço por endereço IP de origem. Ele aceita somente números inteiros como um argumento e pode ser usado em ambos **xinetd.conf** e nos arquivos de configuração de serviço específicos no diretório **xinetd.d/**.

- **cps** — Define o número máximo de conexões por segundo. Esta diretiva leva dois argumentos de números inteiros separados por um espaço em branco. O primeiro argumento é o número máximo de conexões permitidas para o serviço por segundo. O segunto argumento é o número de segundos que o **xinetd** deve esperar antes de re ativar o serviço. Ele aceita somente números inteiros como argumentos e pode ser usado tanto no arquivo **xinetd.conf** ou nos arquivos de configuração de serviço específicos no diretório **xinetd.d/**.

- **max_load** — Define o uso de CPU ou limite da média de carregamento para um serviço. Ele aceita um argumento de número de ponto flutuante.

A média de carregamento é uma medida aproximada de quantos processos estão ativos em um determinado momento. Veja os comandos **uptime**, **who**, e **procinfo** para mais informações sobre a média de carregamento.

Existem mais opções de gerenciamento de recursos disponíveis para o **xinetd**. Consulte a página man **xinetd.conf** para maiores informações.

2.3.5. Recursos Adicionais

Mais informações sobre TCP Wrappers e **xinetd** estão disponíveis na documentação dos sistemas e na internet.

2.3.5.1. Documentação Instalada dos TCP Wrappers

A documentação que está no seu sistema é um bom lugar para iniciar a busca por opções adicionais de configuração para os TCP Wrappers, **xinetd** e o controle de acesso.

- **/usr/share/doc/tcp_wrappers-<*version*>/** — Este diretório contém um arquivo **README** que discute como os TCP Wrappers funcionam e os riscos variados de spoofing do hostname e endereço de host existentes.
- **/usr/share/doc/xinetd-<*version*>/** — Este diretório contém um arquivo **README** que discute aspectos de controle de acesso e um arquivo **sample.conf** com várias idéias para modificar os arquivos de configuração de serviço específicos no diretório **/etc/xinetd.d/**.
- As páginas man relacionadas dos TCP Wrappers e **xinetd** — Uma variedade de páginas man existem para várias aplicações e arquivos de configuração envolvidos com os TCP Wrappers e **xinetd**. A seguir estão algumas das mais importantes das páginas man:

  Aplicações do Servidor

  - **man xinetd** — A página man para o **xinetd**.

  Arquivos de Configurações

  - **man 5 hosts_access** — A página man para os arquivos de controle de acesso ao host dos TCP Wrappers.
  - **man hosts_options** — A página man para os campos de opções dos TCP Wrappers.
  - **man xinetd.conf** — A página man que lista as opções de configuração do **xinetd**.

2.3.5.2. Web sites úteis sobre TCP Wrappers

- http://www.docstoc.com/docs/2133633/An-Unofficial-Xinetd-Tutorial — Um tutorial completo que discute as diferentes maneiras de otimizar os arquivos de configuração do **xinetd** para atender objetivos de segurança específicos.

2.3.5.3. Livros Relacionados

- *Hacking Linux Exposed* por Brian Hatch, James Lee, and George Kurtz; Osbourne/McGraw-Hill — Uma excelente fonte sobre segurança com informações sobre TCP Wrappers e **xinetd**.

2.4. REDES PRIVADAS VIRTUAIS (VPNS)

Organizações com diversos escritórios satélites muitas vezes se conectam entre si por linhas dedicadas para eficiência e proteção dos dados. Por exemplo, muitas empresas usam frame relay ou linhas *Asynchronous Transfer Mode* (ATM) como uma solução de rede ponto a ponto para ligar um escritório com os outros. Isto pode ser uma alternativa cara, especialmente para pequenas e médias empresas (SMBs) que querem expandir sem pagar os altos custos associados com o nível corporativo e circuitos digitais dedicados.

Para atender a essa necessidade, as *Virtual Private Networks* (VPNs) foram desenvolvidas. Seguindo os mesmos princípios funcionais dos circuitos dedicados, as VPNs permitem comunicações digitais seguras entre duas partes (ou redes), criando uma *Wide Area Network* (WAN) a partir da *Local Area Networks* (LANs) existentes. O ponto onde ela difere do frame relay ou ATM é no meio de transporte. As VPNs transmitem por IP usando datagramas como uma camada de transporte, a fazendo um canal seguro através da internet para um destino pretendido. A maioria das implemtentações de softwares VPN grátis incorporam métodos de encriptação padrão abertos para máscaras mais os dados em transito.

Algumas organizações empregam soluções de hardware de VPN para aumentar a segurança, enquanto outras usam implementações de software ou baseadas em protocolos. Diversos fornecedores fornecem soluções de hardware de VPN, como Cisco, Nortel, IBM e Checkpoint. Existe uma solução de VPN baseada em software grátis para o Linux chamada FreeS/Wan que utiliza uma implementação padronizada *Internet Protocol Security* (IPsec). Estas soluções de VPN, independente se são baseadas em hardware ou software, agem como roteadores especializados que existem entre a conexões de IPs de um escritório ao outro.

2.4.1. Como uma VPN funciona?

Quando um pacote é transmitido de um cliente, ele envia através do roteador VPN ou gateway, que adiciona uma *Authentication Header* (AH) para roteamento e autenticação. Os dados são então encriptados e finalmente inclusos com um *Encapsulating Security Payload* (ESP). Este mais tarde constitui a descriptografia e instruções de manuseio.

O roteador VPN de recebimento retira a informação do cabeçalho, descriptografa os dados e encaminha ao destino (tanto uma estação de trabalho ou outro nó na rede). Usando uma conexão rede à rede, o nó receptor na rede local recebe os pacotes já descriptografados e prontos para processamento. O processo de criptografia/descriptografia em uma conexão VPN rede à rede é transparente ao nó local.

Com um nível tão elevado de segurança, um invasor não deve somente interceptar um pacote, mas descriptografa-lo também. Invasores que empregam um ataque man-in-the-middle entre um servidor e cliente devem também ter acesso a pelo menos uma das chaves privadas para sessões de autenticação. Porque elas empregam camadas de autenticação e criptografia, as VPNs são um meio seguro e efetivo de conectar múltiplos nós remotos para agir como uma intranet unificada.

2.4.2. Openswan

2.4.2.1. Visão Geral

Visão Geral

O Openswan é uma implementação de código aberto, no nível de kernel IPsec, disponível no Red Hat Enterprise Linux. Ele emprega protocolos de estabelecimento de chave IKE (Internet Key Exchange) v1 e v2, implementados como daemons no nível de usuário. O estabelecimento de chave manual é também possível via comandos **`ip xfrm`**, entretanto isto não é recomendado.

Suporte Criptográfico

O Openswan possui uma biblioteca embutida de criptografia, entretanto também suporta uma biblioteca NSS (Network Security Services) , que é totalmente suportada e requerida para cumprimento de segurança FIPS. Mais informações sobre o FIPS (Federal Information Processing Standard) pode ser encontrado na Seção 7.2, “Federal Information Processing Standard (FIPS)”.

Instalação

Rode o comando **`yum install openswan`** para instalar o Openswan.

2.4.2.2. Configuração

Localizações

Este seção lista e explica diretórios e arquivos importantes usados para configurar o Openswan.

- **`/etc/ipsec.d`** - diretório principal. Armazena os arquivos relacionados do Openswan.
- **`/etc/ipsec.conf`** - arquivo de configuração principal. Mais arquivos de configuração **`*.conf`** podem ser criados no **`/etc/ipsec.d`** para configurações individuais.
- **`/etc/ipsec.secrets`** - arquivos secretos principais. Mais arquivos **`*.secrets`** podem ser criados no **`/etc/ipsec.d`** para configurações individuais.
- **`/etc/ipsec.d/cert*.db`** - Arquivos de banco de dados certificados. O padrão antigo do arquivo de banco de dados NSS é **`cert8.db`**. A partir do Red Hat Enterprise Linux 6 em diante, o banco de dados NSS sqlite são usados no arquivo **`cert9.db`**.
- **`/etc/ipsec.d/key*.db`** - Arquivos de banco de dados chave. O padrão antigo do banco de dados NSS é **`key3.db`**. A partir do Red Hat Enterprise Linux 6 em diante, os bancos de dados sqlite NSS são usados no arquivo **`key4.db`**.
- **`/etc/ipsec.d/cacerts`** - Localização dos Certificate Authority (CA).
- **`/etc/ipsec.d/certs`** - Localização dos certificados do usuário. Não necessário quando usar o NSS.
- **`/etc/ipsec.d/policies`** - Políticas dos grupos. As políticas podem ser definidas como *block*, *clear*, *clear-or-private*, *private*, *private-or-clear*.
- **`/etc/ipsec.d/nsspassword`** - O arquivo de senha NSS. Este arquivo não existe por padrão e é requerido se o banco de dados NSS em uso é criado com uma senha.

Parâmetros de Configuração

Esta seção lista algumas das opções de configuração disponíveis, a maioria escritas em **`/etc/ipsec.conf`**.

- **`protostack`** - define qual pilha de protocolo é usada. A opção padrão no Red Hat Enterprise Linux 6 é *netkey*. Outros valores válidos são *auto*, *klips* e *mast*.
- **`nat_traversal`** - define se a solução NAT para conexões é aceita. O padrão é não.
- **`dumpdir`** - define a localização para despejo dos arquivos core.
- **`nhelpers`** - Quando usar o NSS, define o número de segmentos usados para criptografar operações. Quando não estiver usando o NSS, define o número de processos usados para operações de criptografia.

- **virtual_private** - sub-redes permitidas para a conexão cliente. A faixa que pode existir por detrás de um roteador NAT a que um cliente se conecta.
- **plutorestartoncrash** - definido para sim por padrão.
- **plutostderr** - caminho para o log de erro do pluto. Aponta para a localização do syslog por padrão.
- **connaddrfamily** - pode ser definido tanto para ipv4 ou ipv6.

Mais detalhes sobre a configuração Openswan podem ser encontradas na página man **ipsec.conf(5)**.

2.4.2.3. Comandos

Esta seção explica e dá exemplos de alguns dos comandos usados pelo Openswan.

NOTA

Como mostrado no exemplo a seguir, usar o **service ipsec start/stop** é o método recomendado para mudar o estado do serviço ipsec. Isto é também a técnica recomendada para iniciar e parar todos os outros serviços no Red Hat Enterprise Linux 6.

- Iniciando e parando o Openswan:
  - **ipsec setup start/stop**
  - **service ipsec start/stop**
- Adicionar/Deletar uma conexão:
  - **ipsec auto --add/delete <connection name>**
- Estabelecer/quebrar uma conexão
  - **ipsec auto --up/down <connection-name>**
- Gerando chaves RSA:
  - **ipsec newhostkey --configdir /etc/ipsec.d --password password --output /etc/ipsec.d/<name-of-file>**
- Checando políticas ipsec no Kernel:
  - **ip xfrm policy**
  - **ip xfrm state**
- Criando certificados auto assinados:
  - **certutil -S -k rsa -n <ca-cert-nickname> -s "CN=ca-cert-common-name" -w 12 -t "C,C,C" -x -d /etc/ipsec.d**
- Criando um certificado de usuário assinado pelo CA anterior:

- `certutil -S -k rsa -c <ca-cert-nickname> -n <user-cert-nickname> -s "CN=user-cert-common-name" -w 12 -t "u,u,u" -d /etc/ipsec.d`

2.4.2.4. Recursos Openswan

- http://www.openswan.org
- http://lists.openswan.org/pipermail/users/
- http://lists.openswan.org/pipermail/dev/
- http://www.mozilla.org/projects/security/pki/nss/
- O pacote Openswan-doc: HTML, exemplos, README.*
- README.nss

2.5. FIREWALLS

A segurança da informação geralmente significa um processo e não um produto. No entanto, implementações de segurança padrões geralmente implementam uma forma de mecanismo dedicada a controlar privilégios de acesso e restringir recursos de rede à usuários não autorizados, identificáveis e rastreáveis. O Red Hat Enterprise Linux inclui diversas ferramentas que assistem administradores e engenheiros de segurança com problemas de controle de acesso em nível de rede.

Firewalls são um dos componentes principais de uma implementação de segurança de rede. Diversos fabricantes criam soluções de firewall, focando em todos os níveis do mercado: desde usuários domésticos protegendo um PC até soluções de banco de dados, protegendo informações corporativas vitais. \nOs Firewalls podem ser soluções de hardware sozinhas, como equipamentos de firewall da Cisco, Nokia e Sonicwall. Empresas como Checkpoint, McAfee e Symantec também desenvolveram soluções de firewall de software privado para mercado doméstico e corporativo.

Além das diferenças entre os firewalls de software e hardware, existem também diferenças na forma que os firewalls funcionam, que separam uma solução da outra. A Tabela 2.2, “Tipos de Firewall” detalha três tipos comuns de firewall e como eles funcionam:

Tabela 2.2. Tipos de Firewall

Método	Descrição	Vantagens	Desvantagens

Método	Descrição	Vantagens	Desvantagens
NAT	*Network Address Translation* (NAT) coloca sub-redes IP privadas atrás de um ou pequeno grupo de endereços IP, mascarando todas as requisições em uma fonte ao invés de diversas. O kernel do Linux possui a funcionalidade NAT embutida através do subsistema do kernel Netfilter.	· Pode ser configurado de forma transparente em máquinas em uma LAN · A proteção de muitas máquinas e serviços atrás de um ou mais endereços IP externos simplifica as tarefas de administração. · Restrição de acesso ao usuário de e para uma LAN pode ser configurado ao abrir e fechar portas no fireway/gateway do NAT.	· Não é possível evitar atividades mal-intencionadas depois que os usuários se conectarem à um serviço fora do firewall.
Filtro de Pacote	Um firewall de filtro de pacote lê cada pacote de dados que passa por uma LAN. Ele pode ler e processar os pacotes por informações de cabeçalho e filtra o pacote baseado em conjuntos de regras programáveis implementada por um administrador de firewall. O kernel do Linux possui uma funcionalidade de filtro de pacotes embutida através do subsistema do kernel, o Netfilter.	· Personalizável através do utilitário front-end `iptables` · Não requer qualquer personalização no lado do cliente, pois todas as atividades de rede são filtradas no nível do roteador ao invés do nível de aplicativo. · Como os pacotes não são transmitidos através de uma proxy, o desempenho de rede é mais rápido devido à conexão direta do cliente para host remoto.	· Não é possível filtrar pacotes para firewalls de proxy de conteúdo. · Processa pacotes na camada de protocolos, mas não pode filtrar pacotes na camada do aplicativo. · Arquiteturas de rede complexas podem dificultar o estabelecimento das regras de filtragem de pacote, principalmente se em par com o *mascaramento de IP* ou sub-redes locais e redes DMZ.

Método	Descrição	Vantagens	Desvantagens
Proxy	Os firewalls de proxy filtram todas as requisições de um certo protocolo ou tipo de clientes LAN para uma máquina de proxy, que então faz essas requisições para a Internet em nome do cliente local. Uma máquina proxy age como um buffer entre os usuários remotos mal-intencionados e as máquinas clientes de rede internas.	· Fornece aos administradores controle sob quais aplicativos e protocolos funcionam foram da LAN · Alguns servidores proxy podem fazer o cache dos dados mais acessados localmente, ao invés de precisar usar a conexão de internet para requisitá-los. Isto ajuda a reduzir o consumo de banda larga. · Os serviços proxy podem ser autenticados e monitorados de perto, permitindo um controle maior do uso de recursos na rede.	· Proxies são geralmente aplicativos específicos (HTTP, Telnet, etc.), ou protocolos restritos (a maioria dos proxies funcionam somente com os serviços conectados ao TCP) · Os serviços de aplicativos não podem ser executados por trás de um proxy, portanto seus servidores de aplicativo devem usar uma forma separada de segurança de rede. · Os proxies podem se tornar um gargalo na rede, pois todas as requisições e transmissões são passadas através de uma fonte ao invés de diretamente de um cliente à um serviço remoto.

2.5.1. Netfilter e IPTables

O kernel do Linux apresenta um subsistema de rede poderoso chamado *Netfilter*. O subsistema Netfilter fornece um filtro de pacotes com ou sem estados assim como o NAT e serviços de mascaramento do IP. O Netfilter também tem a habilidade de *desmembrar* informações de cabeçalho de IP para roteamento avançado e gerenciamento do estado de conexão. O Netfilter é controlado usando a ferramenta **`iptables`**.

2.5.1.1. Visão Geral do IPTables

O poder e flexibilidade do Netfilter é implementado usando a ferramenta de administração do **`iptables`**, uma ferramenta de linha de comando semelhante à sintaxe de seu precedente, **`ipchains`**, o qual o Netfilter/iptables substituiu no kernel 2.4 e posteriores do Linux.

O **`iptables`** usa o subsistema do Netfilter para aprimorar a conexão de rede, inspeção e processamento. O **`iptables`** apresenta autenticação avançada, ações pré e pós roteamento, tradução de endereço de rede e encaminhamento de porta, todos em uma interface de linha de comando.

Esta seção fornece uma visão geral do **`iptables`**. Para mais informações detalhadas, consulte a Seção 2.6, “IPTables”.

2.5.2. Configuração de Firewall Básica

Assim como uma parede corta fogo (firewall) tenta evitar o alastre do fogo, o firewall do computador tenta evitar que softwares mal-intencionados se alastrem em seu computador. Ele também ajuda a evitar que usuários não autorizados acessem seu computador.

Em uma instalação padrão do Red Hat Enterprise Linux, um firewall existe entre seu computador ou rede e qualquer rede não confiável, por exemplo a Internet. Ele determina qual serviço em seu computador que os usuários remotos podem acessar. Um firewall configurado adequadamente pode aumentar a segurança de seu sistema de forma significativa. Recomenda-se que você configure um firewall para qualquer sistema Red Hat Enterprise Linux com uma conexão da Internet.

2.5.2.1. Firewall Configuration Tool

Durante a tela de **`Configuração do Firewall`** da instalação do Red Hat Enterprise Linux, você tem a opção para habilitar um firewall básico assim como permitir dispositivos específicos, serviços de entrada e portas.

Após a instalação, você pode mudar esta preferência usando o **Firewall Configuration Tool**.

Para iniciar este aplicativo, use o seguinte comando:

```
[root@myServer ~] # system-config-firewall
```

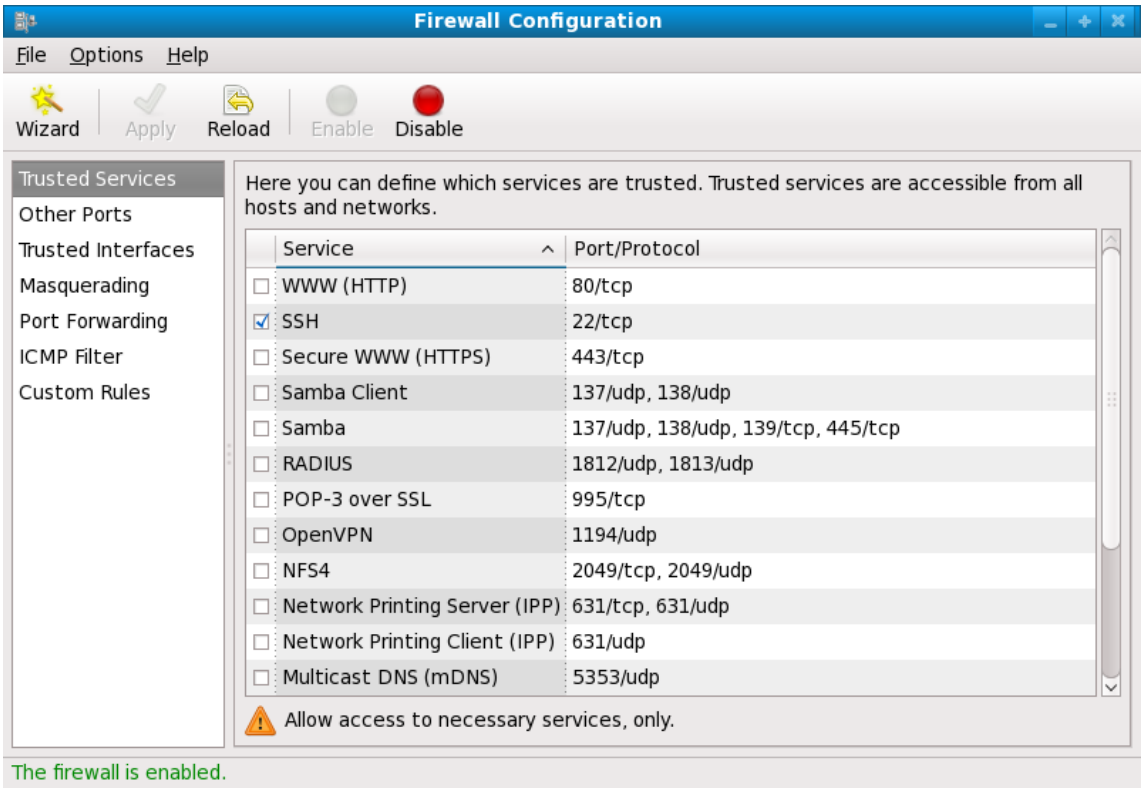


Figura 2.5. Firewall Configuration Tool

NOTA

O **Firewall Configuration Tool** somente configura um firewall básico. Se o sistema precisar de regras mais complexas, consulte a Seção 2.6, “IPTables” para mais detalhes sobre como configurar regras de configurações específicas de **`iptables`**.

2.5.2.2. Habilitando e desabilitando o Firewall

Selecione uma das seguintes opções para o firewall:

- **Disabled** — Desabilitar o firewall fornece acesso completo ao seu sistema e não faz verificação de segurança. Isto deve ser selecionado somente se você estiver executando em uma rede confiável (não a Internet) ou precisar configurar um firewall padronizado usando a ferramenta de linha de comando iptables.

ATENÇÃO

As configurações do Firewall e quaisquer regras de firewall padronizada é armazenada no arquivo **/etc/sysconfig/iptables**. Se você escolher **Disabled** e clicar em **OK**, estas configurações e regras do firewall serão perdidas.

- **Enabled** — Esta opção configura o sistema para rejeitar conexões de entrada que não estão em reposta com requisições externas, tais como respostas de DNS ou requisições de DHCP. Se for necessário o acesso à serviços vindos desta máquina, você poderá escolher permitir serviços específicos através do firewall.

  Se você estiver conectando seu sistema à Internet, mas não planeja executar um servidor, esta é a escolha mais rápida.

2.5.2.3. Serviços Confiáveis

Habilitar opções na lista de **Serviços confiáveis** permite que o serviço especificado passe através do firewall.

WWW (HTTP)

O protocolo HTTP é usado pelo Apache (e outros servidores da Web) para servir as páginas da Web. Se você planeja tornar seu servidor publicamente disponível, selecione esta caixa. Esta opção não é necessária para visualizar páginas localmente ou para desenvolver páginas da Web. Este serviço requer que o pacote **httpd** seja instalado.

Habilitar o **WWW (HTTP)** não abrirá uma porta para o HTTPS, a versão SSL do HTTP. Se este serviço for necessário, selecione o item **Secure WWW (HTTPS)**.

FTP

O protocolo FTP é usado para transferir arquivos entre máquinas em uma rede. Se você planeja tornar seu FTP publicamente disponível, selecione este item. Este serviço requer que o pacote **vsftpd** seja instalado.

SSH

Secure Shell (SSH) é um conjunto de ferramentas para se autenticar e executar comandos em uma máquina remota. Para permitir acesso remoto à esta máquina via ssh, selecione este item. Este serviço requer que o pacote **openssh-server** seja instalado.

Telnet

A Telnet é um protocolo para autenticação em máquinas remotas. As comunicações da Telnet são descriptografadas e não fornecem nenhuma segurança contra o snooping de rede. Permitir acesso de entrada da Telnet não é recomendado. Para permitir acesso remoto à máquina via telnet, selecione este item. Este serviço requer que o pacote **telnet-server** seja instalado.

Mail (SMTP)

O SMTP é um protocolo que permite hosts remotos se conectarem diretamente à sua máquina para a entrega de correio. Você não precisa habilitar este serviço se você coletar seu correio de um servidor ISP usando o POP3 ou IMAP, ou se você utilizar uma ferramenta como o **fetchmail**. Para permitir a entrega de correio para sua máquina, selecione esta caixa. Note que um servidor SMTP configurado inadequadamente pode permitir máquinas remotas usar seu servidor para enviar spam.

NFS4

O Network File System (NFS) é um protocolo de compartilhamento de arquivos geralmente usado em sistemas *NIX. A Versão 4 deste protocolo é mais segura do que seus precedentes. Se você quiser compartilhar arquivos ou diretórios em seu sistema com outros usuários de rede, selecione esta caixa.

Samba

O Samba é uma implementação do protocolo de rede SMB de propriedade da Microsoft. Se você precisar compartilhar arquivos, diretórios ou impressoras conectadas localmente com máquinas Microsoft Windows, selecione esta caixa.

2.5.2.4. Outras Portas

O **Firewall Configuration Tool** inclui uma seção **Other ports** para especificar portas padrão IP como sendo confiáveis pelo **iptables**. Por exemplo, para permitir que o IRC e protocolo de impressora da internet (IPP) passe pelo firewall, adicione o seguinte para a seção **Other ports**:

```
194:tcp,631:tcp
```

2.5.2.5. Salvando Configurações

Clique em **OK** para salvar as mudanças e habilitar ou desabilitar o firewall. Se o item **Habilitar o firewall** foi selecionado, as opções selecionadas são traduzidas para os comandos do **iptables** e gravadas no arquivo **/etc/sysconfig/iptables**. O serviço **iptables** também é iniciado para que o firewall seja ativado imediatamente após salvar as opções selecionadas. Se o **Disable firewall** foi selecionado, o arquivo **/etc/sysconfig/iptables** será removido e o serviço **iptables** será interrompido imediatamente.

As opções selecionadas são também gravadas no arquivo **/etc/sysconfig/system-config-firewall** para que as configurações possam ser restauradas na próxima vez que o aplicativo for iniciado. Não edite este arquivo manualmente.

Embora o firewall seja ativado imediatamente, o serviço **iptables** não é configurado para iniciar automaticamente durante a inicialização. Consulte a Seção 2.5.2.6, “Ativando o Serviço IPTables.” para mais informações.

2.5.2.6. Ativando o Serviço IPTables.

As regras do firewall são ativas somente se o serviço **iptables** estiver em execução. Para iniciar manualmente o serviço, use o seguinte comando:

```
[root@myServer ~] # service iptables restart
```

Para se certificar que o **iptables** inicia-se quando o sistema é inicializado, use o seguinte comando:

```
[root@myServer ~] # chkconfig --level 345 iptables on
```

2.5.3. Usando IPTables

O primeiro passo para usar o **iptables** é iniciar o serviço **iptables**. Use o seguinte comando para iniciar o serviço **iptables**:

```
[root@myServer ~] # service iptables start
```

NOTA

O serviço **ip6tables** pode ser delisgado se você desejar usar somente o serviço **iptables**. Se desativar o **ip6tables** lembre-se de desativar a rede IPv6 também. Nunca deixe um dispositivo de rede ativo sem um firewall.

Para forçar o **iptables** a iniciar por default quando o sistema for inicializado, use o seguinte comando:

```
[root@myServer ~] # chkconfig --level 345 iptables on
```

Para forçar o **iptables** a iniciar sempre que o sistema inicializar em runlevel 3, 4, ou 5.

2.5.3.1. Sintaxe de Comandos do IPTables

O exemplo a seguir de comando **iptables** ilustra a sintaxe de comando básica:

```
[root@myServer ~ ] # iptables -A <chain> -j <target>
```

A opção **-A** especifica a regra a ser adicionada ao *<chain>*. Cada corrente é comprimida em um ou mais *regras*, e é portanto também conhecida como um *ruleset*.

As três correntes embutidas são INPUT, OUTPUT, e FORWARD. Estas correntes são permanentes e não podem ser removidas. A corrente especifica o ponto no qual o pacote é manipulado.

A opção **-j *<target>*** especifica o alvo da regra; ex: o que fazer se o pacote coincide com a regra. Exemplos de alvos embutidos são ACCEPT, DROP e REJECT.

Consulte o página man do **iptables** para mais informações sobre as correntes disponíveis, opções e alvos.

2.5.3.2. Políticas de Firewall Básicas

Estabelecer políticas de firewall básicas cria uma fundação para construir regras definidas por usuários detalhadas.

Cada corrente **iptables** contém uma política padrão e zero ou mais regras que funcionam em conjunto com a política padrão para definir o conjunto de regras (ruleset) geral para o firewall.

A política padrão para uma corrente pode ser DROP ou ACCEPT. Os administradores preocupados com a segurança geralmente implementam uma política padrão DROP, e somente permitem pacotes específicos de acordo com cada caso. Por exemplo, as seguintes políticas bloqueiam todos os pacotes de entrada e saída em uma gateway de rede:

```
[root@myServer ~ ] # iptables -P INPUT DROP
[root@myServer ~ ] # iptables -P OUTPUT DROP
```

Também é recomendado à qualquer tráfego de rede de *pacotes enviados* — que deva ser roteado do firewall para seu nó de destino — seja negado também, para clientes internos restritos de exposições inadvertentes à internet. Para fazer isto, use a seguinte regra:

```
[root@myServer ~ ] # iptables -P FORWARD DROP
```

Depois que você estabeleceu as políticas padrões para cada corrente, você pode criar e salvar regras para sua rede particular e requerimentos de segurança.

As seguintes seções descrevem como salvar regras de iptables e delimitar algumas regras que possam implementar durante a construção de seu firewall iptables.

2.5.3.3. Salvando e Restaurando as Regras IPTables

Mudanças para o **iptables** são transitórias; se o sistema for reinicializado ou se o serviço **iptables** for reiniciado, as regras serão removidas automaticamente e redefinidas. Para salvar as regras para que sejam carregadas quando o serviço **iptables** é iniciado, use o seguinte comando:

```
[root@myServer ~ ] # service iptables save
```

As regras são armazenadas no arquivo **/etc/sysconfig/iptables** e são aplicadas sempre que o serviço é iniciado ou quando a máquina é reinicializada.

2.5.4. Filtros de IPTables Comuns

Impedir invasores remotos de acessar uma LAN é o aspecto mais importante de segurança de rede. A integridade da LAN deve ser protegida de usuários remotos mal-intencionados durante o uso de regras de firewall.

No entanto, com uma política padrão definida para bloquear todos os pacotes enviados, entrada e saída, é impossível para o firewall/gateway e usuários internos da LAN de se comunicarem uns com os outros ou com recursos externos.

Para permitir que usuários realizem funções relacionadas à rede e usar os aplicativos de rede, administradores devem abrir certas portas para comunicação.

Por exemplo, para permitir acesso à porta 80 *no firewall* adicione a seguinte regra:

```
[root@myServer ~ ] # iptables -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
```

Esta regra permite que usuários naveguem websites que comunicam usando a porta padrão 80. Para permitir acesso à websites seguros (por exemplo, https://www.example.com/), você também precisará fornecer acesso à porta 443, como se segue:

```
[root@myServer ~ ] # iptables -A INPUT -p tcp -m tcp --dport 443 -j ACCEPT
```

IMPORTANTE

Ao criar um conjunto de regras **iptables**, a ordem é importante.

Se uma regra especificar que qualquer pacote da sub-rede 192.168.100.0/24 seja despejada, e seja seguido de uma regra que permite pacotes do 192.168.100.13 (o qual está dentro da sub-rede que foi despejada), então a segunda regra é ignorada.

A regra para permitir pacotes do 192.168.100.13 deve preceder a regra que despeja o restante da sub-rede.

Para inserir uma regra em um local específico em uma corrente existente, use a opção **-I**. Por exemplo:

```
[root@myServer ~ ] # iptables -I INPUT 1 -i lo -p all -j ACCEPT
```

Esta regra é inserida como a primeira regra na corrente INPUT para permitir tráfego de dispositivo loopback local.

Pode haver vezes que você requer acesso remoto à LAN. Serviços seguros, por exemplo SSH, podem ser usados para conexão remota criptografada à serviços LAN.

Administradores com recursos baseados em PPP (como os bancos de modem ou contas ISP em massa) acesso discado pode ser usado para evitar barreiras de firewall de forma segura. Como são conexões diretas, as conexões de modem se encontram geralmente atrás de um firewall/gateway.

No entanto, para usuários remotos com conexões de banda larga, são reservados casos especiais. Você pode configurar o **iptables** para aceitar conexões de clientes remotos SSH. Por exemplo, as regras a seguir permitem acesso SSH remoto:

```
[root@myServer ~ ] # iptables -A INPUT -p tcp --dport 22 -j ACCEPT
[root@myServer ~ ] # iptables -A OUTPUT -p tcp --sport 22 -j ACCEPT
```

Estas regras permitem acesso de entrada e saída para um sistema individual, como um PC único conectado diretamente à Internet ou à um firewall/gateway. No entanto, eles não permitem que nós por detrás de firewall/gateway acessem estes serviços. Para permitir acesso da LAN à estes serviços, use *Network Address Translation* (NAT) com as regras de filtro do **iptables**.

2.5.5. FORWARD e Regras NAT

A maioria dos ISPs fornecem somente um número limitado de endereços IP roteáveis publicamente à empresa que eles servem.

Os administradores devem portanto encontrar formas alternativas de compartilhar acesso à serviços de Internet sem fornecer endereços IP públicos à todos os nós na LAN. O uso de endereços IP privados é a forma mais comum de permitir que todos os nós em uma LAN acessem adequadamente serviços de rede internos e externos.

Roteadores de Limite (como os firewalls) podem receber transmissões de entrada de Internet e rotear pacotes para o nós da LAN pretendido. Ao mesmo tempo, os firewalls/gateways podem também rotear requisições de saída de um nó de LAN para serviço de Internet remoto.

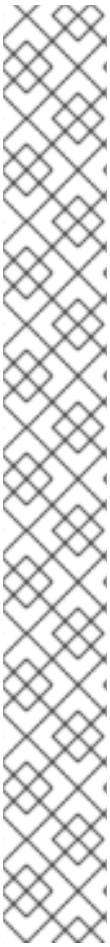
Este encaminhamento de tráfego de rede pode se tornar perigoso, especialmente com a disponibilidade de ferramentas modernas de intrusão que podem fazer o spoof de endereços IP *internos* e fazer com que a máquina remota do invasor aja como um nó em sua LAN.

Para evitar isto, o **iptables** fornece políticas de roteação e encaminhamento que podem ser implementadas para impedir o uso anormal dos recursos de rede.

A corrente **FORWARD** permite que um administrador controle onde os pacotes podem ser roteados dentro de uma LAN. Por exemplo, para permitir o encaminhamento de uma LAN inteira (considerando-se que o firewall/gateway receba um endereço IP na eth1), use as regras a seguir:

```
[root@myServer ~ ] # iptables -A FORWARD -i eth1 -j ACCEPT
[root@myServer ~ ] # iptables -A FORWARD -o eth1 -j ACCEPT
```

Esta regra dá acesso à sistemas por detrás de firewall/gateway à rede interna. O gateway roteia pacotes de um nó de LAN ao seu nó de destino, passando todos os pacotes por seu dispositivo **eth1**.



NOTA

Por padrão, a política IPv4 nos kernels do Red Hat Enterprise Linux desabilitam o suporte de encaminhamento de IP. Isto evita que máquinas rodando o Red Hat Enterprise Linux funcionem como roteadores de limite dedicado. Para habilitar o encaminhamento de IP, use o seguinte comando:

```
[root@myServer ~ ] # sysctl -w net.ipv4.ip_forward=1
```

Esta mudança de configuração é válilda somente para a sessão atual; não persiste através de reinicializações ou reinício de serviço de rede. Para definir o encaminhamento de IP permanentemente, edite o arquivo **/etc/sysctl.conf** como a seguir:

Localize a linha a seguir:

```
net.ipv4.ip_forward = 0
```

Edite-o para fica como a seguir:

```
net.ipv4.ip_forward = 1
```

Use o seguinte comando para permitir mudanças no arquivo **sysctl.conf**:

```
[root@myServer ~ ] # sysctl -p /etc/sysctl.conf
```

2.5.5.1. Postrouting e Mascaramento de IP

Aceitar pacotes encaminhados via dispositivo de IP interno do firewall permite que os nós da LAN se comuniquem entre si; no entanto eles ainda não poderão se comunicar fora da Internet.

Para permitir nós da LAN com endereços IP privados se comunicarem com redes externas públicas, configure o firewall para *IP masquerading*, o qual mascara requisições dos nós da LAN com endereços IP do dispositivo externo do firewall (neste caso eth0):

```
[root@myServer ~ ] # iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Esta regra usa a tabela de coincidências do pacote NAT (**-t nat**) e especifica a corrente POSTROUTING embutida para o NAT (**-A POSTROUTING**) no dispositivo de rede externo do firewall (**-o eth0**).

O POSTROUTING permite que os pacotes sejam alterados a medida que deixam o dispositivo externo do firewall.

O alvo **-j MASQUERADE** é especificado para mascarar o endereço IP de um nó com o endereço IP externo do firewall/gateway.

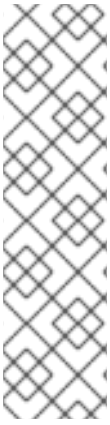
2.5.5.2. Pre roteamento

Se você possuir um servidor em sua rede interna que você queira disponibilizar externamente, você pode usar o alvo **-j DNAT** da corrente PREROUTING no NAT para especificar um endereço IP de destino e porta para onde pacotes de entrada requisitando uma conexão para seu serviço interno possam ser encaminhados

Por exemplo, se você quiser encaminhar requisições HTTP de entrada para seu Servidor Apache HTTP no 172.31.0.23, use o seguinte comando:

```
[root@myServer ~ ] # iptables -t nat -A PREROUTING -i eth0 -p tcp --dport
80 -j DNAT --to 172.31.0.23:80
```

Esta regra especifica que a tabela nat use a corrente PREROUTING embutida para enviar requisições HTTP de entrada exclusivamente para o endereço IP de destino listado do 172.31.0.23.



NOTA

Se você possuir uma política padrão de DROP em sua corrente FORWARD, você precisa adicionar uma regra para enviar à todas as requisições HTTP de entrada para que o roteamento de destino NAT seja possível. Para fazer isto, use o seguinte comando:

```
[root@myServer ~ ] # iptables -A FORWARD -i eth0 -p tcp --dport
80 -d 172.31.0.23 -j ACCEPT
```

Esta regra encaminha todas as requisições de HTTP do firewall para o destino pretendido; o servidor Apache HTTP por detrás do firewall.

2.5.5.3. DMZs e IPTables

Você pode criar as regras de comando **iptables** para rotear o tráfego para certas máquinas, tal como o servidor dedicado HTTP ou FTP, em um *demilitarized zone* (DMZ). Um DMZ é uma sub-rede local especial dedicada para fornecer serviços em uma portador público, tal como a Internet.

Por exemplo, para estabelecer uma regra para requisições de entrada HTTP de roteamento para um servidor HTTP dedicado em 10.0.4.2 (fora da classe do 192.168.1.0/24 da LAN), o NAT usa a tabela **PREROUTING** para encaminhar os pacotes para o destino apropriado:

```
[root@myServer ~ ] # iptables -t nat -A PREROUTING -i eth0 -p tcp --dport
80 -j DNAT --to-destination 10.0.4.2:80
```

Com este comando, todas as conexões para a porta 80 de fora da LAN são roteadas para o servidor HTTP em uma rede separada do resto da rede interna. Esta forma de segmentação de rede pode ser mais segura do que permitir conexões de HTTP para uma máquina na rede.

Se o servidor HTTP for configurado para aceitar conexões seguras, então a porta 443 deve ser encaminhada também.

2.5.6. Softwares Maliciosos e Spoof de Endereços IP

Regras mais elaboradas podem ser criadas para controlar acesso à sub-redes específicas ou até nós específicos, dentro de uma LAN. Você pode também restringir certos aplicativos duvidosos ou programas tais como trojans, worms e outros vírus de clientes/servidores de contatar seus servidores.

Por exemplo, alguns trojans escaneiam redes procurando serviços na portas de 31337 até 31340 (chamadas de portas *elite* na terminologia dos crackers).

Como não há mais serviços legítimos que se comunicam via estas portas não padrão, bloqueá-las pode diminuir efetivamente as chances que nós potencialmente infectados em sua rede se comuniquem de forma independente com seus servidores mestre remotos.

As seguintes regras derrubam todo o tráfego do TCP que tentam usar a porta 31337:

```
[root@myServer ~ ] # iptables -A OUTPUT -o eth0 -p tcp --dport 31337 --
sport 31337 -j DROP
[root@myServer ~ ] # iptables -A FORWARD -o eth0 -p tcp --dport 31337 --
sport 31337 -j DROP
```

Você também pode bloquear conexões de fora que tentam fazer spoof de variações de endereço IP privados para infiltrar sua LAN.

Por exemplo, se sua LAN usa a classe 192.168.1.0/24, você pode criar uma regra que instrui o dispositivo de rede de internet (por exemplo , eth0) para derrubar todos os pacotes naquele dispositivo com um endereço em sua classe de IP de LAN.

Pelo motivo que é recomendado rejeitar pacotes encaminhados como política padrão, qualquer outro endereço IP forjado ao dispositivo que lida com dados externos (eth0) é rejeitado automaticamente.

```
[root@myServer ~ ] # iptables -A FORWARD -s 192.168.1.0/24 -i eth0 -j DROP
```

NOTA

Existe uma distinção entre o **DROP** e alvos **REJECT**ao lidar com as regras *adicionadas*.

O alvo **REJECT** nega acesso e retorna um erro de **conexão negada** para usuários que tentam se conectar ao serviço. O alvo **DROP**, como o nome implica, despeja o pacote sem qualquer aviso.

Os administradores podem usar sua própria discrição ao usar estes alvos. No entanto, para evitar confusão do usuário e tentativa de continuar se conectar, o alvo **REJECT** é recomendado.

2.5.7. IPTables e Rastreamento de Conexão

Você pode inspecionar e restringir conexões à serviços baseados em seus *estados de conexão.* Um módulo dentro do **iptables** usa um método chamado *rastreamento de conexão* para armazenar informações sobre conexões de entrada. Você pode permitir ou negar acesso baseado nos seguintes estados de conexão:

- **NEW** — Um pacote que requer uma nova conexão, tal como uma requisição HTTP.
- **ESTABLISHED** — Um pacote que é parte de uma conexão existente.
- **RELATED** — Um pacote que está requisitando uma nova conexão mas é parte de uma conexão existente. Por exemplo, o FTP usa a porta 21 para estabelecer uma conexão, mas os dados são transferidos em uma porta diferente (geralmente a porta 20).
- **INVALID** — Um pacote que não é parte de nenhuma conexão na tabela de rastreamento de conexão.

Você pode usar a funcionalidade stateful (com estado) da conexão **iptables** rastreando com qualquer protocolo de rede, até mesmo se o próprio protocolo é stateless (sem estado) assim como o UDP. O exemplo a seguir mostra uma regra que usa o rastreamento de conexão para enviar somente os pacotes que são associados com uma conexão estabelecida:

```
[root@myServer ~ ] # iptables -A FORWARD -m state --state
ESTABLISHED,RELATED -j ACCEPT
```

2.5.8. IPv6

A introdução do Protocolo de Internet de próxima geração, chamado IPv6 expande além do endereço do limite de 32 bits do IPv4 (ou IP). O IPv6 suporta endereços de 128 bits e portadores de redes que aceitam IPv6 são portanto capazes de controlar um número maior de endereços roteáveis do que o IPv4.

O Red Hat Enterprise Linux suporta as regras do firewall IPv6 usando o subsistema do Netfilter 6 e o comando **ip6tables**. No Red Hat Enterprise Linux 6, ambos serviços IPv4 e IPv6 são habilitados por padrão.

A sintaxe do comando **ip6tables** é idêntica ao **iptables** em todos os aspectos exceto que ele suporta os endereços 128 bits. Por exemplo, use o seguinte comando para habilitar as conexões SSH em um servidor de rede consciente do IPv6:

```
[root@myServer ~ ] # ip6tables -A INPUT -i eth0 -p tcp -s
3ffe:ffff:100::1/128 --dport 22 -j ACCEPT
```

Para mais informações sobre a rede IPv6, consulte a Página de informações do IPv6 em http://www.ipv6.org/.

2.5.9. Recursos Adicionais

Existem diversos aspectos do firewall e do subsistema Linux Netfilter que talvez não sejam vistos neste capítulo. Para mais informações, consulte os recursos a seguir.

2.5.9.1. Documentação de Firewall Instalada

- Consulte a Seção 2.6, “IPTables” para mais informações detalhadas sobre o comando **iptables** incluindo definições para diversas opções de comando.
- A página man do **iptables** contém um breve sumário de diversas opções.

2.5.9.2. Websites de Firewall Úteis

- http://www.netfilter.org/ — A homepage oficial do Netfilter e do projeto **iptables**.
- http://www.tldp.org/ — O Projeto de Documentação do Linux contém diversos guias úteis relacionados à criação de firewall e administração.
- http://www.iana.org/assignments/port-numbers — A lista oficial de portas de serviços comuns e registrados atribuídos pelo Internet Assigned Numbers Authority.

2.5.9.3. Documentação Relacionada

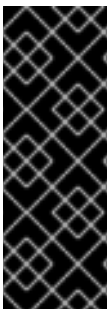
- *Red Hat Linux Firewalls*, by Bill McCarty; Red Hat Press — uma referência compreensiva para construção de redes e firewalls de servidor usando uma tecnologia de filtro de pacote de código aberto como o Netfilter e o **iptables**. Ele inclui tópicos que cobrem análise de logs de firewall, desenvolvimento de regras de firewall e padronização de seu firewall usando diversas ferramentas gráficas.
- *Linux Firewalls*, by Robert Ziegler; New Riders Press — Contém informações ricas sobre a construção de firewalls usando o 2.2 kernel **ipchains**, como também o Netfilter e **iptables**. Também são tratados tópicos de segurança adicionais como os problemas de acesso remoto e sistemas de detecção de intrusão.

2.6. IPTABLES

Incluídos com o Red Hat Enterprise Linux estão ferramentas avançadas para *filtragem de pacotes* de rede — o processo de controlar pacotes de rede conforme entram, se movem e saem da pilha de rede dentro do kernel. As versões do kernel anteriores à 2.4 que confiavam no **ipchains** para filtragem de pacotes e usavam listas de regras aplicadas ao pacote em cada passo do processo de filtragem. O kernel 2.4 introduziu o **iptables** (também chamado de *netfilter*), o qual é semelhante ao **ipchains** mas expande o alcance e controle disponíveis para filtrar pacotes de rede.

Este capítulo foca no conhecimento básico de filtragem de pacote, explica diversas opções disponíveis com os comandos **iptables**, e explica como regras de filtragem podem ser preservadas entre as reinicializações de sistema.

Consulte a Seção 2.6.6, “Recursos Adicionais” para instruções sobre como construir regras de **iptables** e instalar um firewall baseado nestas regras.



IMPORTANTE

O mecanismo de firewall padrão no kernel 2.4 e versões posteriores é o **iptables**, mas o **iptables** não pode ser usado se o **ipchains** já estiver sendo executado. Se o **ipchains** estiver presente durante a inicialização, o kernel emite um erro e não inicia o **iptables**.

A funcionalidade do **ipchains** não foi afetada por estes erros.

2.6.1. Filtro de Pacote

O kernel Linux usa o serviço do **Netfilter** para filtrar pacotes, permitindo que alguns deles sejam recebidos ou passados pelo sistema enquanto outros são interrompidos. Este serviço é embutido no kernel do Linux e possui três *tabelas* ou *listas de regras* embutidos, como se segue:

- **filter** — A tabela padrão para manipular pacotes de rede.

- **`nat`** — Usado para alterar pacotes que criam uma nova conexão e usado para o *Network Address Translation* (*NAT*).
- **`mangle`** — Usado para tipos específicos de alteração de pacote.

Cada tabela possui um grupo de *correntes* (chains), que correspondem às ações realizadas no pacote pelo **`netfilter`**.

As correntes (chains) embutidas para a tabela do **`filtragem`** são estas:

- *INPUT* — Se aplica aos pacotes de rede que são direcionados para o host.
- *OUTPUT* — Se aplica ao pacotes de rede gerados localmente.
- *FORWARD* — Se aplica aos pacotes de rede roteados no host.

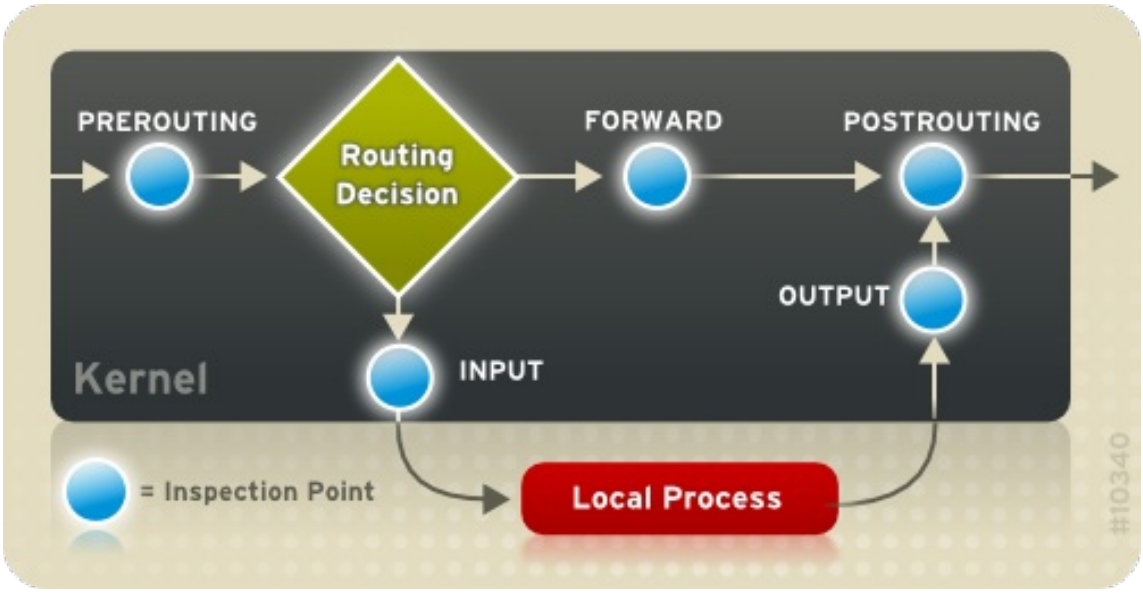
As correntes embutidas para a tabela **`nat`** são estas:

- *PREROUTING* — Altera os pacotes de rede quando chegam.
- *OUTPUT* — Altera os pacotes de rede gerados localmente antes de serem enviados..
- *POSTROUTING* — Altera pacotes de rede antes de serem enviados.

As chains (correntes) embutidas para a tabela **`mangle`** são estas:

- *INPUT* — Altera pacotes de rede alvo para a máquina.
- *OUTPUT* — Altera os pacotes de rede gerados localmente antes de serem enviados..
- *FORWARD* — Altera pacotes de rede roteados pela máquina.
- *PREROUTING* — Altera pacotes de entrada antes de serem roteados.
- *POSTROUTING* — Altera pacotes de rede antes de serem enviados.

Cada pacote de rede recebido por ou enviado de um sistema Linux está sujeito à ao menos uma tabela. No entanto, um pacote pode estar sujeito à diversas regras dentro de cada tabela antes de emergir no final da corrente. A estrutura e propósito destas regras podem variar, mas elas geralmente procuram identificar um pacote que vem ou vai à um endereço IP específico, ou conjunto de endereços, ao usar um determinado protocolo e serviço de rede. A imagem a seguir apresenta como os pacotes são examinados pelo subsistema do iptables:



NOTA

Por padrão, as regras de firewall são salvas nos arquivos **`/etc/sysconfig/iptables`** ou **`/etc/sysconfig/ip6tables`** .

O serviço **`iptables`** inicia antes dos serviços relacionados ao DNS quando um sistema Linux é inicializado. Isto significa que as regras do firewall podem somente fazer referência à endereços de IP numéricos (por exemplo, 192.168.0.1). Nomes do domínio (por exemplo, host.example.com) em tais regras, produzem erros.

Seja qual for o destino, quando os pacotes coincidem com uma regra específica em uma das tabelas, um *alvo* ou ação é aplicado à eles. Se a regra especifica um alvo **`ACCEPT`** para um pacote coincidente, o pacote pula o restante das verificações de regras e é permitido que continue com seu destino. Se uma regra especifica um alvo **`DROP`**, aquele pacote é recusado ter acesso ao sistema e nada é retornado ao host que enviou ou pacote. Se uma regra especifica um alvo **`QUEUE`**, o pacote é passado ao espaço de usuário. Se uma regra especifica o alvo opcional **`REJECT`**, o pacote é despejado, mas um pacote de erro é enviado ao originador do pacote.

Toda corrente possui uma política padrão para **`ACCEPT`**, **`DROP`**, **`REJECT`**, ou **`QUEUE`**. Se nenhuma destas regras na corrente se aplicar ao pacote, então o pacote será lidado de acordo com a política padrão.

O comando **`iptables`** configura estas tabelas, assim como instala tabelas se necessário.

2.6.2. Opções de Comando para IPTables.

Regras para filtrar pacotes são criadas usando o comando **`iptables`**. Se os aspectos a seguir de um pacote são usados geralmente como um critério:

- *Packet Type* — Especifica os tipos de pacotes que o comando filtra.
- *Packet Source/Destination* — Especifica quais pacotes o comando filtra baseado na fonte ou destino do pacote.
- *Target* — Especifica qual ação é tomada nos pacotes que coincidem com o critério acima.

Consulte a Seção 2.6.2.4, “Opções de Coincidência de IPTables” e a Seção 2.6.2.5, “Opções de Alvo” para mais informações sobre opções específicas que se referem à estes aspectos de um pacote.

As opções usadas com as regras **iptables** específicas devem ser agrupadas de forma lógica, baseadas no propósito e condições da regra geral, para a regra ser válida. O restante desta seção explica opções mais utilizadas para o comando **iptables**.

2.6.2.1. Estrutura das Opções do Comando IPTables

Muitos comandos **iptables** possuem a seguinte estrutura:

```
 iptables [-t <table-name>] <command> <chain-name> \ <parameter-1>
<option-1> \ <parameter-n> <option-n>
```

<table-name> — Especifica qual tabela a regra se aplica. Se omitido, a tabela **filter** será usada.

<command> — Especifica a ação a realizar, tal como adicionar ou remover uma regra.

<chain-name> — Especifica a corrente a editar, criar ou remover.

<parameter>-*<option>* pairs — Os parâmetros e opções associadas que especificam como processar um pacote que coincide com a regra.

O extensão e complexidade de um comando **iptables** pode mudar de forma significante, dependendo do seu propósito.

Por exemplo, um comando que remove uma regra de uma corrente pode ser bastante curto:

iptables -D <*chain-name*> <*line-number*>

Em contraste, um comando que adiciona uma regra que filtra pacotes de uma sub-rede específica usando uma variedade de parâmetros e opções específicos, podem ser um tanto longos. Ao construir os comandos **iptables**, é importante lembrar que alguns parâmetros e opções requerem mais parâmetros e opções para construir uma regra válida. Isto pode produzir um efeito cascata, com os parâmetros adicionais que requerem ainda mais parâmetros. Até que cada parâmetro e opção que requerem outro conjunto de opções sejam atendidos, a regra não é válida.

Digite **iptables -h** para visualizar uma lista compreensiva de estruturas de comando **iptables**.

2.6.2.2. Opções de Comando

As Opções de Comando instruem o **iptables** a realizar uma ação específica. Somente uma opção de comando é permitida por comando **iptables**. Com a exceção do comando help, todos os comandos são escritos em letras maiúsculas.

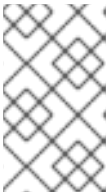
Os comandos **iptables** são:

- **-A** — Adiciona a regra ao final da corrente especificada. Oposto à opçao **-I** descrita abaixo, esta opção não toma um argumento inteiro. Ele sempre adiciona a regra ao final da corrente especificada.

- **-D <integer> | <rule>** — Remove uma regra em uma corrente específica pelo número (como o **5** para a quinta regra em uma corrente), ou por uma especificação de regra. A especificação de regra deve coincidir exatamente uma regra existente.

- **-E** — Renomeia uma corrente definida por usuário. Uma corrente definida por usuário é qualquer corrente exceto as padrões, correntes pré-existentes. (consulte a opção **-N** abaixo para informações sobre como criar correntes definidas por usuário). Isto é uma mudança de

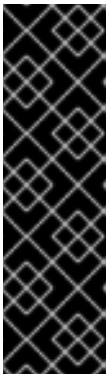
aparência e não afeta a estrutura da tabela.



NOTA

Se você tentar renomear uma das correntes padrões, o sistema irá reportar um erro de **`Correspondência não encontrada`** (Match not found). Você não poderá renomear correntes padrões.

- **`-F`** — Libera a corrente selecionada, o qual remove efetivamente todas as regras na corrente. Se não for especificada nenhuma corrente, este comando liberará todas as regras de cada corrente.
- **`-h`** — Fornece uma lista de estruturas de comando, assim como um sumário rápido de parâmetros de comando e opções.
- **`-I [<integer>]`** — Insere a regra na corrente especificada em um ponto especificado por um argumento inteiro definido por um usuário. Se nenhum argumento é especificado a regra é inserida no topo da corrente.



IMPORTANTE

Como notado acima, a ordem de regras em uma corrente determina quais regras se aplicam à quais pacotes. Isto é importante lembrar quando adicionar regras usando tanto a opção **`-A`** ou **`-I`**.

Isto é especialmente importante ao adicionar regras usando o **`-I`** com um argumento inteiro. Se você especificar um número existente ao adicionar uma regra em uma corrente, o **`iptables`** adiciona a nova regra *antes* (ou acima) da regra existente.

- **`-L`** — Lista todas as regras na corrente especificada após o comando. Para listar todas as regras em todas as correntes na tabela de **`filtragem`** padrão, não especifica uma corrente ou tabela. Caso contrário a sintaxe deve ser usada para listar as regras em uma corrente específica em uma tabela específica:

```
iptables -L <chain-name> -t <table-name>
```

Opções adicionais para a opção de comando **`-L`**, que fornece números de regra e permite mais verbosidade nas descrições da regra, são descritas na Seção 2.6.2.6, “Opções de Listagem”.

- **`-N`** — Cria uma nova corrente com um nome de usuário específico. O nome de corrente deve ser único, caso contrário uma mensagem de erro é exibida.
- **`-P`** — Estabelece a política padrão para a corrente especificada, para que quando pacotes passam por uma corrente inteira sem coincidir uma regra, eles são enviados para o alvo especificado, tal como ACCEPT ou DROP.
- **`-R`** — Substitui uma regra na corrente especificada. O número de regra deve ser especificado após o nome da corrente. A primeira regra em uma corrente corresponde à regra número um.
- **`-X`** — Remove uma corrente de usuário específico. Você não pode remover uma corrente embutida.

- **-Z** — Estabelece os contadores de byte e pacote em todas as correntes para uma tabela para zero.

2.6.2.3. Opções de Parâmetro de IPTables

Certos comandos do **iptables**, incluindo aqueles usados para adicionar, remover, inserir ou substituir regras dentro de uma corrente específica, requer diversos parâmetros para construir uma regra de filtro de pacote.

- **-c** — Redefine os contadores para uma regra específica. Este parâmetro aceita as opções **PKTS** e **BYTES** para especificar quais contadores redefinir.

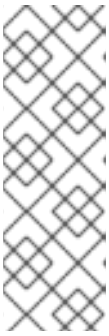
- **-d** — Estabelece o hostname de destino, endereço IP, ou rede de um pacote que coincide a regra. Quando coincidir uma rede, o seguinte formato de endereço IP/netmasks são suportados:

  - ***N.N.N.N/M.M.M.M*** — Onde *N.N.N.N* é a classe de endereço IP e *M.M.M.M* é o netmask.

  - ***N.N.N.N/M*** — Onde *N.N.N.N* é o endereço IP e *M* é o bitmask.

- **-f** — Aplique estas regras somente à pacotes fragmentados.

  Você pode usar este caractere de ponto de exclamação (**!**) antes deste parâmetro para especificar que somente os pacotes desfragmentados são coincidentes.



NOTA

Distinguir entre os pacotes fragmentados e desfragmentados é uma boa prática, apesar dos pacotes fragmentados serem uma parte padrão do protocolo IP.

Inicialmente criado para permitir que pacotes IP viajem sob redes com tamanhos de estruturas diferentes, estas fragmentações de dias são mais usadas para gerar ataques de DoS usando pacotes mal-formados. Também vale notar que a fragmentação retira totalmente a permissão.

- **-i** — Define a interface de rede de entrada, como **eth0** ou **ppp0**. Com o **iptables**, este parâmetro opcional pode ser usado somente com as correntes INPUT e FORWARD quando usado com a tabela **filter** e a corrente PREROUTING com as tabelas **nat** e **mangle**.

  Este parâmetro também suporta as seguintes opções especiais:

  - Caractere de ponto de exclamação (**!**) — Reverte a diretiva, o que significa que qualquer interface especificada é excluída desta regra.

  - Caractere de mais (**+**) — Um caractere curinga usado para coincidir todas as interfaces que coincidem com a faixa especificada. Por exemplo, o parâmetro **-i eth+** aplicaria esta regra à qualquer interface de Ethernet mas excluiria qualquer outra interface, tal como **ppp0**.

  Se o parâmetro **-i** é usado mas nenhuma interface é especificada, então todas as interfaces são afetadas pela regra.

- **-j** — Pula para o alvo especificado quando um pacote coincide uma regra particular.

  Os alvos padrões são **ACCEPT**, **DROP**, **QUEUE**, e **RETURN**.

Opções estendidas também estão disponíveis através de módulos carregados pelo padrão com o Red Hat Enterprise Linux **iptables** pacote RPM. Alvos válidos nestes módulos incluem **LOG**, **MARK**, e **REJECT**, entre outros. Consulte a página man **iptables** para obter mais informações sobre estes e outros alvos.

Esta opção também pode ser usada para direcionar um pacote que coincide com uma regra em particular para uma corrente definida por um usuário fora da corrente atual, então outras regras possam ser aplicadas ao pacote.

Se não for especificado nenhum alvo, o pacote passa pela regra sem nenhuma ação. O contador para esta regra, no entanto, aumenta por um.

- **-o** — Define a interface de rede de saída para uma regra. Esta opção é válida somente para correntes OUTPUT e FORWARD na tabela **filter**, e a corrente POSTROUTING nas tabelas **nat** e **mangle**. Este parâmetro aceita as mesmas opções que o parâmetro de interface de rede de entrada (**-i**).

- **-p <protocol>** — Define o protocolo IP afetado pela regra. Este pode ser tanto o **icmp**, **tcp**, **udp**, ou **all**, ou pode ser um valor numérico, representando um destes ou um protocolo diferente. Você pode usar qualquer protocolo listado no arquivo **/etc/protocols**.

  O protocolo "**all**" significa que a regra se aplica à todos os protocolos suportados. Se nenhum protocolo for listado com esta regra, ele se torna retorna ao padrão "**all**".

- **-s** — Define a fonte para um pacote específico usando a mesma sintaxe como parâmetro de destino (**-d**).

2.6.2.4. Opções de Coincidência de IPTables

Protocolos de rede diferentes fornecem opções de coincidência especializadas que podem ser configuradas para coincidir um pacote específico usando aquele protocolo. No entanto, o protocolo deve primeiro ser especificado no comando **iptables**. Por exemplo, **-p <*protocol-name*>** habilita as opções para o protocolo especificado. Note que você também pode usar o ID de protoloco, ao invés do nome do protocolo. Consulte os seguintes exemplos, cada dos quais tem o mesmo efeito:

```
iptables -A INPUT -p icmp --icmp-type any -j ACCEPT
```

```
iptables -A INPUT -p 5813 --icmp-type any -j ACCEPT
```

Definições de serviço são fornecidas no arquivo **/etc/services**. Para definição legível, recomenda-se que você use os nomes de serviços ao invés dos números de porta.

ATENÇÃO

Proteja o arquivo **/etc/services** para prevenir edição não autorizada. Se este arquivo for editável, invasores podem usá-lo para habilitar portas em sua máquina que você tenha fechado. Para proteger este arquivo, digite os seguintes comandos como usuário root:

```
[root@myServer ~]# chown root.root /etc/services
[root@myServer ~]# chmod 0644 /etc/services
[root@myServer ~]# chattr +i /etc/services
```

Isto evita que o arquivo seja renomeado, removido ou ter links apontados para ele.

2.6.2.4.1. Protocolo TCP

Estas opções de coincidência estão disponíveis para o protocolo TCP (**-p tcp**):

- **--dport** — Define a porta de destino para o pacote.

  Para configurar esta opção, use um nome de serviço de rede (tal como o www ou smtp); um número de porta; ou uma classe de números de porta.

  Para especificar uma classe de números de porta, separe os dois números com dois pontos (**:**). Por exemplo: **-p tcp --dport 3000:3200**. A classe válida maior é **0:65535**.

  Use um ponto de exclamação (**!**) antes da opção **--dport** para coincidir todos os pacotes que *não* usam aquele serviço ou porta de rede.

  Para navegar pelos nomes e aliases de serviços de rede e números de porta que eles usam, visualize o arquivo **/etc/services**.

  A opção de coincidência do **--destination-port** é sinônima da **--dport**.

- **--sport** — Define a porta fonte do pacote usando as mesmas opções que **--dport**. A opção de coincidência **--source-port** é sinônimo da **--sport**.

- **--syn** — Se aplica à todos os pacotes TCP criados para iniciar a comunicação, geralmente chamada de *pacotes SYN*. Quaisquer pacotes que carreguem um bloco de dados não são tocados.

  Use um ponto de exclamação (**!**) antes da opção **--syn** para coincidir com todos os pacotes que não são SYN.

- **--tcp-flags <tested flag list> <set flag list>** — Permite que pacotes TCP que possuem específicos bits (sinalizadores) definidos, coincidirem com uma regra.

  A opção de coincidência **--tcp-flags** aceita dois parâmetros. O primeiro parâmetro é a máscara; uma lista separada por vírgula de sinalizadores a serem examinados no pacote. O segundo parâmetro é uma lista separada por vírgula de sinalizadores que devem ser definidos para a regra que coincidir.

Os possíveis sinalizadores são:

- **ACK**
- **FIN**
- **PSH**
- **RST**
- **SYN**
- **URG**
- **ALL**
- **NONE**

Por exemplo, uma regra **iptables** que contenha a seguinte especificação somente coincide com pacotes TCP que possuem o sinalizador SYN definido e os sinalizadores ACK e FIN não definidos:

```
--tcp-flags ACK,FIN,SYN SYN
```

Use o ponto de exclamação (**!**) antes do **--tcp-flags** para reverter o efeito da opção de coincidência.

- **--tcp-option** — Tenta coincidir com as opções de TCP específicas que podem ser definidas dentro de um pacote particular. Esta opção coincidente também pode ser revertida com o ponto de exclamação (**!**).

2.6.2.4.2. Protocolo UDP

Estas opções de coincidência estão disponíveis para o protocolo UDP (**-p udp**):

- **--dport** — Especifica a porta de destino do pacote UDP, usando o nome de serviço, número de porta ou classe de número de portas. A opção de coincidência **--destination-port** é sinônima de **--dport**.
- **--sport** — Especifica a porta fonte do pacote UDP, usando o nome de serviço, número de porta, ou classe de números de porta. A opção coincidente **--source-port** é sinônima com **--sport**.

Para as opções **--dport** e **--sport**, para especificar uma classe de números de portas, separe os dois números com dois pontos (:). Por exemplo: **-p tcp --dport 3000:3200**. A maior classe válida aceitável é 0:65535.

2.6.2.4.3. Protocolo ICMP

As opções de coincidência a seguir estão disponíveis para o Internet Control Message Protocol (ICMP) (**-p icmp**):

- **--icmp-type** — Define o nome e número do tipo de ICMP para coincidir com a regra. Uma lista de nomes ICMP válidos pode ser recuperada digitando o comando **iptables -p icmp -h**.

2.6.2.4.4. Módulos de Opção de Coincidência Adicional

Opções de coincidência adicionais estão disponíveis através de módulos carregados pelo comando **iptables**.

Para usar um módulo de opção de coincidência, carregue o módulo pelo nome usando o **-m *<module-name>***, onde *<module-name>* é o nome do módulo.

Muitos módulos estão disponíveis por padrão. Você também pode criar módulos para fornecer funcionalidade adicional.

Segue uma lista parcial dos módulos mais usados:

- módulo **limit** — Coloca limites em quantos pacotes são coincididos em uma regra específica.

  Quando usado em conjunto com o alvo **LOG**, o módulo **limit** pode evitar uma inundação de pacotes coincidentes de encher o log do sistema com mensagens repetitivas ou usar os recursos do sistema.

  Consulte a Seção 2.6.2.5, “Opções de Alvo” para mais informações sobre o alvo **LOG**.

  O módulo **limit** habilita as seguintes opções:

  - **--limit** — Define o número máximo de coincidências para um período de tempo específico, especificado como um par **<*value*>/<*period*>**. Por exemplo, usando o **--limit 5/hour** permite-se cinco coincidências de regras por hora.

    Períodos podem ser especificados em segundos, minutos, horas ou dias.

    Se um modificador de número e tempo não forem utilizados, o valor padrão de **3/hour** é assumido.

  - **--limit-burst** — Define um limite em um número de pacotes capazes de coincidir uma regra em uma vez.

    Esta opção é especificada como um número inteiro e deve ser usada em conjunto com a opção **--limit**.

    Se não for especificado nenhum valor, o valor padrão de cinco (5) será assumido.

- módule **state** — Habilita coincidência de estado.

  O módulo **state** habilita as seguintes opções:

  - **--state** — coincide um pacote com os seguintes estados de conexões:

    - **ESTABLISHED** — O pacote coincidente é associado com outros pacotes em uma conexão estabelecida. Você precisa aceitar este estado se você quiser manter uma conexão entre um cliente e um servidor.

    - **INVALID** — O pacote coincidente não pode ser ligado à uma conexão conhecida.

    - **NEW** — O pacote coincidente está tanto criando uma nova conexão ou é parte de uma conexão de duas vias ainda não vista. Você precisa aceitar este estado se você quiser permitir novas conexões para um serviço.

- **RELATED** — O pacote coincidente inicia uma nova conexão relacionada de alguma forma a uma conexão existente. Um exemplo disto é o FTP que usa uma conexão para o controle de tráfego (porta 21), e uma conexão separada para a transferência de dados (porta 20).

Estes estados de conexão podem ser usados em conjunto um com o outro separando-os com com vírgulas, tais como **`-m state --state INVALID,NEW`**.

- **`mac`** module — Habilita coincidência do endereço MAC de hardware.

  O módulo **`mac`** habilita a seguinte opção:

  - **`--mac-source`** — Coincide um endereço MAC de placa de interface de rede que enviou o pacote. Para excluir um endereço MAC de uma regra, coloque um ponto de exclamação (**`!`**) antes da opção de coincidência **`--mac-source`**.

Consulte a página man **`iptables`** para mais opções de coincidência disponíveis através de módulos.

2.6.2.5. Opções de Alvo

Quando um pacote coincidiu uma regra específica, a regra pode direcionar o pacote para diversos alvos diferentes que determinam a ação apropriada. Cada corrente possui um alvo padrão, que é usado se nenhuma das regras na corrente coincidir um pacote ou se nenhuma das regras que coincidem com o pacote especificarem um alvo.

Seguem os alvos padrões:

- ***`<user-defined-chain>`*** — Uma corrente definida pelo usuário dentro da tabela. Nomes de correntes definidas pelo usuário devem ser únicas. Este alvo passa o pacote à corrente especificada.
- **`ACCEPT`** — Permite o pacote passar para seu destino ou para outra corrente.
- **`DROP`** — Despeja o pacote sem responder ao requisitante. O sistema que enviou o pacote não é notificado da falha.
- **`QUEUE`** — O pacote é enfileirado para manuseio por um aplicativo do espaço de usuário.
- **`RETURN`** — Para a verificação de pacote contra as regras na corrente atual. Se o pacote com um alvo **`RETURN`** coincide com uma regra em uma corrente chamada de outra corrente, o pacote é retornado à primeira corrente para retomar a verificação de regra onde ele parou. Se a regra **`RETURN`** for usada em uma corrente embutida e o pacote não puder mover para sua corrente anterior, o alvo padrão para a corrente atual será usado.

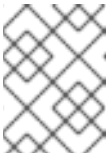
Além disso, as extensões estão disponíveis que permitem outros alvos a serem especificados. Estas extensões são chamadas de módulos alvo ou módulos de opção coincidente e se aplicam mais às tabelas específicas e situações. Consulte a Seção 2.6.2.4.4, “Módulos de Opção de Coincidência Adicional” para obter mais informações sobre módulos de opção de coincidência.

Muitos módulos de alvo estendidos existem, a maioria deles só se aplica à tabelas específicas ou situações. Alguns dos módulos de alvo mais populares incluídos por padrão no Red Hat Enterprise Linux são:

- **`LOG`** — Registra em log todos os pacotes que coincidem com esta regra. Como os pacotes são autenticados pelo kernel, o arquivo **`/etc/syslog.conf`** determina onde estas entradas de log são escritas. Por padrão, elas são colocadas no arquivo **`/var/log/messages`**.

Opções adicionais podem ser usadas após o alvo **LOG** para especificar a forma na qual a autenticação ocorre:

- **--log-level** — Define o nível de prioridade de um evento de log. Consulte a página man **syslog.conf** para obter uma lista de níveis de prioridade.
- **--log-ip-options** — Registra em log qualquer opção definida no cabeçalho de um pacote IP.
- **--log-prefix** — Coloca uma faixa de até 29 caracteres antes da linha do log quando ela é escrita. Isto é útil para escrever filtros de syslog para usar em conjunto com registro em log dos pacotes.



NOTA

Devido à um problema com esta opção, você precisa adicionar um espaço à direita ao valor *log-prefix* .

- **--log-tcp-options** — Registra em log qualquer opção definida no cabeçalho de um pacote de TCP.
- **--log-tcp-sequence** — Escreve o número sequencial de TCP para o pacote no log.

- **REJECT** — Envia um pacote de erro de volta ao sistema remoto e despeja o pacote.

  O alvo **REJECT** aceita **--reject-with <*type*>** (onde *<type>* é o tipo de rejeição) permitindo mais informações detalhadas a serem retornadas com pacote de erro. A mensagem **port-unreachable** é o tipo de erro padrão dado se nenhuma opção for utilizada. Consulte a página man **iptables** para obter uma lista completa de opções **<*type*>**.

Outras extensões de alvo, incluindo diversos que são úteis para o mascaramento do IP usando a tabela **nat**, ou com a alteração do pacote usando a tabela **mangle**, podem ser encontradas na página man do **iptables** .

2.6.2.6. Opções de Listagem

O comando da lista padrão, **iptables -L [<chain-name>]**, fornece uma visão geral básica das correntes atuais. As opções adicionais fornecem mais informações:

- **-v** — Exibe resultado de verbosidade, tal como o número de pacotes e bytes que cada corrente processou, o número de pacotes e bytes que cada regra coincidiu e quais as interfaces que se aplicam à regra específica.
- **-x** — Expande números para seus valores exatos. Em um sistema ocupado, o número de pacotes e bytes processados por uma corrente específica ou regra pode ser abreviada para **Kilobytes**, **Megabytes** (Megabytes) ou **Gigabytes**. Esta opção força o número completo a ser exibido.
- **-n** — Exibe os endereços IP e os números de porta em formato numérico, ao invés do hostname padrão e o formato de serviço de rede.
- **--line-numbers** — Lista regras em cada corrente próxima à ordem numérica na corrente. Esta opção é útil para quando tentar remover a regra específica em uma corrente ou localizar onde inserir uma regra dentro de uma corrente.

- **-t <table-name>** — Especifica um nome de tabela. Se omitido, torna-se tabela de filtro por padrão.

2.6.3. Salvando Regras de IPTables

Regras criadas com o comando **iptables** são armazenadas na memória. Se o sistema for reiniciado antes de salvar o conjunto de regras do **iptables**, todas as regras serão perdidas. Para as regras do netfilter persistirem através da inicialização do sistema, elas precisam ser salvas. Para salvar as regras netfilter, digite o seguinte comando como root:

```
/sbin/service iptables save
```

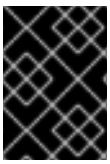
Este executa o script init do **iptables** que roda o programa **/sbin/iptables-save** e escreve a configuração do **iptables** atual em **/etc/sysconfig/iptables**. O arquivo **/etc/sysconfig/iptables** existente é salvo como **/etc/sysconfig/iptables.save**.

A próxima vez que o sistema inicializar, o script init do **iptables** reaplica as regras salvas no **/etc/sysconfig/iptables** usando o comando **/sbin/iptables-restore**.

Se por um lado é uma boa idéia testar uma nova regra **iptables** antes de submetê-la ao arquivo **/etc/sysconfig/iptables**, por outro lado é possível copiar as regras **iptables** para o arquivo a partir de outra versão do sistema deste arquivo. Isto fornece uma forma rápida de distribuir conjuntos de regras **iptables** para multiplicar máquinas.

Você também pode salvar as regras iptables em arquivos separados para a distribuição, backup ou outros propósitos. Para salvar suas regras de iptables, digite o seguinte comando como root:

```
 [root@myServer ~]# iptables-save > <filename>where <filename>é um nome de
usuário definido para seu conjunto de regras.
```



IMPORTANTE

Caso distribua o arquivo **/etc/sysconfig/iptables** à outras máquinas, digite **/sbin/service iptables restart** para as novas regras tomarem efeito.

NOTA

Note a diferença entre o *comando* **iptables**(**,/sbin/iptables**), o qual é usado para manipular as tabelas e correntes que constituem a funcionalidade do **iptables** e o *serviço* **iptables**, (**/sbin/service iptables**), o qual é usado para habilitar e desabilitar o próprio serviço **iptables**.

2.6.4. Scripts de Controle de IPTables

Existem dois métodos básicos para controlar o **iptables** no Red Hat Enterprise Linux:

- **Firewall Configuration Tool** (**system-config-firewall**) — Uma interface gráfica para criar, ativar e salvar regras de firewall básicas. Consulte a Seção 2.5.2, “Configuração de Firewall Básica” para mais informações.

- **/sbin/service iptables <option>** — Usado para manipular diversas funções de **iptables** usando seu initscript. As seguintes opções estão disponíveis:

- **start** — se um firewall é configurado (ou seja, o **/etc/sysconfig/iptables** existe), todos os **iptables** em execução são interrompidos completamente e depois iniciados usando o comando **/sbin/iptables-restore**. Esta opção funciona somente se o módulo do kernel **ipchains** não for carregado. Para verificar se o módulo foi carregado, digite o seguinte comando como usuário root:

  ```
  [root@MyServer ~]# lsmod | grep ipchains
  ```

  Se este comando retornar nenhum resultado, significa que o módulo não foi carregado. Se necessário, use o comando **/sbin/rmmod** para remover o módulo.

- **stop** — Se o firewall estiver rodando, as suas regras na memória serão liberadas e todos os módulos do iptables e auxiliares serão descarregados.

  Se a diretiva **IPTABLES_SAVE_ON_STOP** no arquivo de configuração do **/etc/sysconfig/iptables-config** é modificado de seu valor padrão para **yes**,as regras atuais serão salvas em **/etc/sysconfig/iptables** e quaisquer regras existentes serão movidas para o arquivo **/etc/sysconfig/iptables.save**.

  Refer to Seção 2.6.4.1, “Arquivo de Configuração de Scripts de Controle do IPTables” para mais informações sobre o arquivo **iptables-config**.

- **restart** — Se um firewall estiver sendo executado, as regras do firewall na memória serão liberadas, e o firewall será iniciado novamente se estiver configurado no **/etc/sysconfig/iptables**. Esta opção funciona somente se o módulo do kernel **ipchains** não for carregado.\n\t\n

  Se a diretiva **IPTABLES_SAVE_ON_RESTART** no arquivo de configuração **/etc/sysconfig/iptables-config** for modificada de seu valor padrão para **yes**, as regras atuais serão salvas em **/etc/sysconfig/iptables** e quaisquer regras existentes serão movidas para o arquivo **/etc/sysconfig/iptables.save**.

  Refer to Seção 2.6.4.1, “Arquivo de Configuração de Scripts de Controle do IPTables” para mais informações sobre o arquivo **iptables-config**.

- **status** — Exibe o status do firewall e lista todas as regras ativas.

  A configuração padrão para esta opção exibe os endereços IP em cada regra. Para exibir as informações do domínio e hostname, edite o arquivo **/etc/sysconfig/iptables-config** e mude o valor de **IPTABLES_STATUS_NUMERIC** para **no**. Consulte a Seção 2.6.4.1, “Arquivo de Configuração de Scripts de Controle do IPTables” para obter mais informações sobre o arquivo **iptables-config**.

- **panic** — Ribera todas as regras do firewall. A política de todas as tabelas configuradas está definida para **DROP**.

  Esta opção pode ser útil se um servidor estiver comprometido. Ao invés de desconectar fisicamente da rede ou fechar o sistema, você pode usar esta opção para interromper todos os tráfegos de rede futuros mas deixar a máquina em um estado pronto para análise ou outros investigações.

- **save** — Salva as regras do firewall em **/etc/sysconfig/iptables** usando **iptables-save**. Consulte a Seção 2.6.3, “Salvando Regras de IPTables” para mais informações.

NOTA

Para usar os mesmos comandos initscript para controlar o netfilter para o IPv6, substitua o **ip6tables** pelo **iptables** nos comandos **/sbin/service** listados nesta seção. Para mais informações sobre o IPv6 e netfilter, consulte a Seção 2.6.5, “IPTables e IPv6”.

2.6.4.1. Arquivo de Configuração de Scripts de Controle do IPTables

O comportamento do initscripts do **iptables** é controlado pelo arquivo de configuração **/etc/sysconfig/iptables-config**. Esta é uma lista das diretivas contidas neste arquivo:

- **IPTABLES_MODULES** — Especifica uma lista separada por espaços dos módulos de **iptables** adicionais para carregar quando um firewall é ativado. Estes podem incluir a conexão de rastreamento e auxiliares do NAT.
- **IPTABLES_MODULES_UNLOAD** — Descarrega módulos na reinicialização e pára. Esta diretiva aceita os seguintes valores:
  - **yes** — O valor padrão. Esta opção deve ser definida para alcançar um estado correto para um reinício ou parada de firewall.
  - **no** — Esta opção deve ser somente definida se existirem problemas para descarregar os módulos do netfilter.
- **IPTABLES_SAVE_ON_STOP** — Salva as regras atuais do firewall em **/etc/sysconfig/iptables** quando um firewall é interrompido. Esta diretiva aceita os seguintes valores:
  - **yes** — Salva as regras existentes em **/etc/sysconfig/iptables** quando o firewall é interrompido, movendo a versão anterior para o arquivo **/etc/sysconfig/iptables.save**.
  - **no** — O valor padrão. Não salva regras existentes quando o firewall for interrompido.
- **IPTABLES_SAVE_ON_RESTART** — Salva regras atuais de firewall quando o firewall é reinicializado. Esta diretiva aceita os seguintes valores:
  - **yes** — Salva as regras existentes em **/etc/sysconfig/iptables** quando o firewall é reiniciado, movendo a versão anterior para o arquivo **/etc/sysconfig/iptables.save**.
  - **no** — O valor padrão. Não salva regras existentes quando o firewall é reiniciado.
- **IPTABLES_SAVE_COUNTER** — Salva e recupera todos os pacotes e contadores de bytes em todas as correntes e regras. Esta diretiva aceita os seguintes valores:
  - **yes** — Salva os valores do contador.
  - **no** — O valor default. Não salva os valores do contador.
- **IPTABLES_STATUS_NUMERIC** — Fornece resultado de endereços IP em forma númerica ao invés do domínio ou hostnames. Esta diretiva aceita os seguintes valores:
  - **yes** — O valor padrão. Retorna somente os endereços IP dentro do resultado do status.
  - **no** — Retorna o domínio ou hostnames dentro do resultado do status.

2.6.5. IPTables e IPv6

Se o pacote **iptables-ipv6** estiver instalado, o netfilter no Red Hat Enterprise Linux pode filtrar a próxima geração de protocolo de Internet IPv6. O comando usado para manipular o netfilter do IPv6 é **ip6tables**.

A maioria das diretivas para este comando são idênticas àquelas usadas para o **iptables**, exceto a tabela **nat** que não é suportada ainda. Isto significa que ainda não é possível realizar a tarefa de tradução do endereços de rede IPv6, tal como o mascaramento e encaminhamento de porta.

As regras para **ip6tables** são salvas no arquivo **/etc/sysconfig/ip6tables**. Regras anteriores salvas pelos initscripts **ip6tables** são salvas no arquivo **/etc/sysconfig/ip6tables.save**.

As opções de configuração para o script init do **ip6tables** estão armazenadas no **/etc/sysconfig/ip6tables-config**, e os nomes para cada diretiva variam muito pouco dos equivalentes do **iptables** .

Por exemplo, a diretiva **iptables-config IPTABLES_MODULES**: o equivalente no arquivo **ip6tables-config** é **IP6TABLES_MODULES**.

2.6.6. Recursos Adicionais

Consulte os seguintes recursos para informações adicionais sobre o pacote de filtro com **iptables**.\n\t\n

- Seção 2.5, “Firewalls” — Contém um capítulo sobre o papel de firewalls dentro da estratégia de segurança geral assim como as estratégias para construir as regras do firewall.

2.6.6.1. Documentação instaladas da IP Tables

- **man iptables** — Contém uma descrição do **iptables** assim como uma lista compreensiva de alvos, opções e extensões coincidentes.

2.6.6.2. Websites de IPtables Úteis

- http://www.netfilter.org/ — O home do projeto netfilter/iptables. Contém informações diversas sobre o **iptables**, incluindo guias FAQ de Rusty Russell, o mantedor do firewall Linux IP. Os documentos HOWTO neste site tratam sobre tais conceitos de rede básicos, filtro de pacote de kernel e configurações do NAT.

- http://www.linuxnewbie.org/nhf/Security/IPtables_Basics.html — Uma introdução para a forma que pacotes se movem através do kernel do Linux, mais uma introdução para construção básica de comandos **iptables**.

---

[11] Já que as BIOS de sistemas diferem entre fabricantes, algumas podem não suportar proteção por senha de qualquer tipo, enquanto outras podem suportar um certo tipo mas não o outro.

[12] O GRUB também aceita senhas sem encriptação, mas é recomendado que um hash MD5 seja usado para segurança adicional

[13] Este acesso é ainda sujeito às restrições impostas pelo SELinux, se ativado

CAPÍTULO 3. CRIPTOGRAFIA

Existem dois tipos principais de dados que devem ser protegidos: dados parados e dados ativos. Estes tipos diferentes de dados são protegidos de formas semelhantes usando tecnologia semelhante mas as implementações podem ser completamente diferentes. Nenhuma implementação de proteção pode prevenir os métodos possíveis de se comprometer com as mesmas informações que possam conter nos dados parados e ativo em determinados períodos diferentes.

3.1. DADOS PARADOS

Dados parados são dados armazenados em um disco rígido, fita, CD, DVD, disco ou outra forma de mídia. A maior ameaça de informação é de roubo. Os laptops em aeroportos, CDs enviados por correio, e fitas de backup que são deixadas em locais errados são alguns exemplos de eventos onde dados foram comprometidos através de roubo. Se os dados estivessem criptografados na mídia, você não teria que se preocupar tanto sobre o comprometimento destes dados.

3.2. CRIPTOGRAFIA DE DISCO CHEIO

Disco cheio ou criptografia de partição é uma das melhores formas de proteger seus dados. Não só protege cada arquivo como também protege armazenamento temporário que possa conter partes destes arquivos . A criptografia de disco cheio irá proteger todos os seus arquivos para que você não tenha que se preocupar com a seleção do que você precisa preencher e possivelmente um arquivo que esteja faltando.

Red Hat Enterprise Linux 6 suporta originalmente a criptografia LUKS. O LUKS criptografas todas as suas partições do disco rígido para que enquanto o computador esteja desligado, seus dados sejam protegidos. Isto também protege seu computador à tentativas de ataques para usar um modo de usuário único para se autenticar em seu computador ou obter acesso.

As soluções de criptografia de disco cheio, assim como LUKS, protege somente dados quando seu computador estiver desligado. Depois que o computador estiver ligado e o LUKS houver descriptografado o disco, os arquivos no disco estarão disponíveis para qualquer um que tivesse acesso normalmente à ele. Para proteger seus arquivos quando o computador estiver ligado, use a criptografia de disco cheio junto com outra solução como a criptografia baseada em arquivo. Lembre-se também de trancar seu computador sempre que estiver longe dele. Uma boa dica para evitar intrusos é obter um descanso de tela protegido por uma frase-senha, que seja ativado após alguns minutos de ociosidade.

3.3. CRIPTOGRAFIA BASEADO EM ARQUIVO

O GnuPG (GPG) é uma versão de fonte aberta de PGP que permite que você se autentique e/ou criptografe um arquivo ou uma mensagem de email. Isto é útil para manter a integridade da mensagem ou arquivo e também protege a confidencialidade das informações contidas dentro do arquivo ou do email. No caso do email, o GPG fornece proteção dupla. Ele não só fornece a proteção de Dados Parados como também a proteção de Dados Ativos, depois da mensagem ter sido enviada pela rede.

A criptografia baseada em arquivo pretende proteger um arquivo após ele ter deixado seu computador, tal como quando você envia um CD por correio. Algumas soluções de criptografia baseada em arquivo deixarão rastros dos arquivos criptografados que permitirá que um atacante que tenha acesso físico ao seu computador recupere-os sob algumas circunstâncias. Para proteger este conteúdo destes arquivos contra intrusos que possam acessar seu computador, use a criptografia baseada em arquivo junto com outra solução como uma criptografia de disco cheio.

3.4. DADOS ATIVOS

Dados ativos são dados que são transmitidos via rede. A maior ameaça aos dados ativos ẽ a intercepção e alteração. Seu nome de usuário e senha nunca devem ser transmitidos via rede sem a proteção, pois pode ser interceptado e usado por outra pessoa que se passe por você ou obter acesso à informações confidenciais. Outras informações privadas como informações de conta bancária, devem também ser protegidas ao serem transmitidas via rede. Se a sessão de rede foi criptografada então você não precisará se preocupar com o comprometimento dos dados enquanto estiverem sendo enviados.

Dados ativos são especialmente vulneráveis à atacantes pois o atacante não precisa estar perto do computador que contém os dados armazenados, eles só precisam estar em algum local no caminho dele. Os túneis de criptografia podem proteger dados no caminho das comunicações.

3.5. VIRTUAL PRIVATE NETWORKS (REDE PRIVADA VIRTUAL)

Virtual Private Networks (Rede Privada Virtual - VPN) fornece túneis criptografados entre computadores ou redes de computadores em todas as portas. Com um VPN, todo o tráfego de rede de clientes é encaminhado para o servidor através do túnel criptografado. Isto significa que o cliente é logicamente na mesma rede que o servidor está conectado via VPN. Os VPNs são muito comuns e simples de usar e instalar.

3.6. SECURE SHELL (SHELL SEGURA)

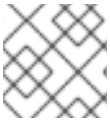
Secure Shell (SSH) é um protocolo de rede potente usado para comunicar com outro sistema sob um canal seguro. As transmissões sob SSH são criptografadas e protegidas de intercepção. A autenticação criptográfica pode também ser utilizada para fornecer um método de autenticação melhor ao invés de nomes de usuários tradicionais e senhas.

O SSH é fácil de ativar. Simplesmente iniciando o serviço sshd, o sistema irá começar a aceitar conexões e permitirá acesso ao sistema quando um nome de usuário correto e senha for fornecido durante o processo de conexão. A porta padrão TCP para o serviço SSH é 22, no entanto isto pode mudar ao modificar o arquivo de configuração */etc/ssh/sshd_config* e reiniciando o serviço. Este arquivo também contém outras opções de configuração para o SSH.

Secure Shell (SSH) também fornece túneis criptografados entre computadores mas somente usando uma única porta. O encaminhamento da Porta pode ser feito pelo túnel SSH e o tráfego será criptografado quando passar por este túnel mas o uso da porta que encaminha não é tão rápido quando o VPN.

3.7. OPENSSL PADLOCK ENGINE

O VIA PadLock Engine está disponível em alguns processadores VIA C3 (Nehemia), e permite criptografia e descriptografia de hardware extremamente rápida.



NOTA

Não existe suporta para o VIA Padlock em sistemas de 64-bit.

Para ativá-lo, edite **`/etc/pki/tls/openssl.cnf`** e adicione o seguinte no início de cada arquivo:

```
openssl_conf = openssl_init
```

Depois adicione o seguinte no final do arquivo:

```
[openssl_init]
engines = openssl_engines

[openssl_engines]
padlock = padlock_engine

[padlock_engine]
default_algorithms = ALL
dynamic_path = /usr/lib/openssl/engines/libpadlock.so
init = 1
```

Para verificar se o módulo está ativado, aplique este comando:

```
# openssl engine -c -tt
```

Para testar a velocidade, aplique este comando:

```
# openssl speed aes-128-cbc
```

Para testar a velocidade do OpenSSH você pode aplicar um comando como este:

```
# dd if=/dev/zero count=100 bs=1M | ssh -c aes128-cbc
localhost "cat >/dev/null"
```

Você pode encontra mais informações sobre o PadLock VIA nas seguintes URLs: http://www.logix.cz/michal/devel/padlock/ e http://www.via.com.tw/en/initiatives/padlock/.

3.8. LUKS DISK ENCRYPTION

Linux Unified Key Setup-on-disk-format (ou LUKS) permite que você criptografe partições em seu computador Linux. Isto é muito importante em relação aos computadores móveis e mídias removíveis. O LUKS permite que chaves de usuários múltiplos descriptografem uma chave master que é usada para criptografia em massa de partição.

3.8.1. Implementação do LUKS no Red Hat Enterprise Linux

O Red Hat Enterprise Linux 6 usa o LUKS para realizar criptografia do sistema de arquivos . Por padrão, a opção de criptografar o sistema de arquivo não é selecionada durante a instalação. Se você selecionar a opção para criptografar seu disco rígido, você precisará inserir uma senha que será solicitada todas as vezes que você inicializar seu computador. Esta senha "desbloqueia" a chave de criptografia em massa que é usada para descriptografar sua partição. Se você escolher modificar a tabela de partição padrão, você poderá escolher quais partições você quer criptografar. Isto é definido nas configurações de tabela da partição.

A cífra padrão usada para o LUKS (consulte o **`cryptsetup --help`**) é aes-cbc-essiv:sha256 (ESSIV - Encrypted Salt-Sector Initialization Vector). Note que o programa de instalação, **Anaconda**, usa o modo XTS por padrão (aes-xts-plain64). O tamanho da chave padrão para o LUKS é 256 bits. O tamanho da chave padrão para o LUKS com o **Anaconda** (XTS mode) é 512 bits. Cífras que estão disponíveis são:

- AES - Advanced Encryption Standard - FIPS PUB 197
- Twofish (A 128-bit Block Cipher)
- Serpent

- cast5 - RFC 2144
- cast6 - RFC 2612

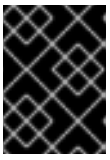
3.8.2. Criptografando Diretórios Manualmente

ATENÇÃO

Ao seguir este procedimento você removerá todos os dados da partição que você está criptografando. Você IRÁ PERDER todas as informações! Certifique-se de criar um backup de seus dados em uma fonte externa antes de iniciar este procedimento!

3.8.3. Instruções Passo-a-Passo

1. entre no runlevel 1: **`telinit 1`**
2. desmonte seu /home existente: **`umount /home`**
3. Se isto falhar, use o **`fuser`** para encontrar e eliminar processos se apoderando do /home: **`fuser -mvk /home`**
4. verifique se o /home não está mais montado: **`cat /proc/mounts | grep home`**
5. Preencha sua partição com dados aleatórios: **`dd if=/dev/urandom of=/dev/VG00/LV_home`** Este processo leva horas para ser concluído.



IMPORTANTE

O processo, no entanto, é crucial para ter uma boa proteção contra tentativas de quebrar a criptografia. Deixe executando durante a noite.

6. inicialize sua partição: **`cryptsetup --verbose --verify-passphrase luksFormat /dev/VG00/LV_home`**
7. abra o dispositivo criptografado recentemente: **`cryptsetup luksOpen /dev/VG00/LV_home home`**
8. confirme que se encontra lá: **`ls -l /dev/mapper | grep home`**\n\t\n
9. crie um sistema de arquivos: **`mkfs.ext3 /dev/mapper/home`**\n\t\n
10. monte-o: **`mount /dev/mapper/home /home`**
11. verifique sua visibilidade: **`df -h | grep home`**
12. adicione o seguinte ao /etc/crypttab: **`home /dev/VG00/LV_home none`**

13. edite seu /etc/fstab, removendo a entrada antiga para /home e adicionando **`/dev/mapper/home /home ext3 defaults 1 2`**

14. restaure o conteúdo de segurança SELinux: **`/sbin/restorecon -v -R /home`**

15. reinicialize: **`shutdown -r now`**

16. A entrada em /etc/crypttab faz com que seu computador solicite sua senha **`luks`** na inicialização

17. Autentique-se como root e recupere seu backup

3.8.4. O que você acaba de concluir.

Parabéns, você criou uma partição criptografada para manter todos os seus dados com segurança enquanto seu computador estiver desligado.

3.8.5. Links de interesse

Para informações adicionais sobre o LUKS ou criptografia de disco rígido sob o Red Hat Enterprise Linux visite um dos seguintes links:

- LUKS home page
- LUKS/cryptsetup FAQ
- LUKS - Linux Unified Key Setup
- HOWTO: Creating an encrypted Physical Volume (PV) using a second hard drive and pvmove

3.9. USANDO O GNU PRIVACY GUARD (GNUPG)

O GPG é usado para identificar você e suas comunicações, incluindo aquelas com pessoas que você não conhece. O GPG permite qualquer pessoa lendo um e-mail com assinatura GPG verificar sua autenticidade. Em outras palavras, o GPG permite a alguém estar razoavelmente certo de que a comunicação assinada por você é realmente sua. O GPG é útil porque ajuda a prevenir terceiros de alterar o código ou interceptar conversas e alterar a mensagem.

3.9.1. Criando chaves GPG no GNOME

Instale o utilitário Seahorse, que deixa o gerenciamento de chave GPG mais fácil. A partir do menu principal, selecione **`Sistema > Administração > Adicionar/Remover Software`** e aguarde pelo PackageKit iniciar. Digite **`Seahorse`** na caixa de texto e selecione Buscar. Marque a caixa próxima ao pacote "seahorse" e selecione "Aplicar" para adicionar o software. Você pode também instalar o **`Seahorse`** na linha de comando com o comando **`su -c "yum install seahorse"`**.

Para criar uma chave, do menu "Aplicativos > Acessórios" selecione "Senhas e Chaves de Criptografia", que inicia a aplicação **`Seahorse`**. Do menu "Arquivo" selecione "Novo" então "Chave PGP". Então clique em "Continuar". Digite o nome inteiro, endereço de email e um comentário opcional descrevendo quem você é (exemplo: (e.g.: John C. Smith, jsmith@example.com, O cara). Clique "Criar". Uma janela é mostrada pedindo a frase secreta para a chave. Escolha uma frase secreta forte mas também fácil de lembrar. Clque "Ok" e a chave será criada.

ATENÇÃO

Se você esquecer a frase secreta, a chave não poderá ser usada e quaisquer dados criptografados usando essa chave serão perdidos.

Para encontrar sua ID de chave GPG, olhe na coluna "Key ID" próxima à chave recém criada. Na maioria dos casos, se você for perguntado pela ID da chave, você deve prefixar "0x" à ID da chave, como em "0x6789ABCD". Você deve fazer um backup de sua chave privada e armazena-la em um lugar seguro.

3.9.2. Criando Chaves GPG no KDE

Inicie o programa KGpg do menu principal selecionando Aplicativos > Utilitários > Ferramentas de Criptografia. Se você nunca usou o KGpg antes, o programa lhe ajuda no processo de criar seu próprio par de chaves GPG. Um caixa de diálogo aparecerá pedindo para você criar um novo par de chaves. Digite seu nome, endereço de e-mail e um comentário opcional. Você pode escolher um período de expiração para sua chave, tanto quanto a força da chave (número de bits) e algorítimos. A próxima caixa de diálogo pede pela frase secreta. Neste momento, sua chave aparece na janela principal do **KGpg**.

ATENÇÃO

Se você esquecer a frase secreta, a chave não poderá ser usada e quaisquer dados criptografados usando essa chave serão perdidos.

Para encontrar sua ID de chave GPG, olhe na coluna "Key ID" próxima à chave recém criada. Na maioria dos casos, se você for perguntado pela ID da chave, você deve prefixar "0x" à ID da chave, como em "0x6789ABCD". Você deve fazer um backup de sua chave privada e armazena-la em um lugar seguro.

3.9.3. Criando chaves GPG Usando a Linha de Comando

Use o seguinte comando no shell: **gpg --gen-key**

Este comando gera um par de chaves que consiste de uma chave pública e uma privada. Outras pessoas usam sua chave pública para se autenticar e/ou descriptografar suas comunicações. Distribua sua chave pública o máximo possível, especialmente para pessoas que você sabe que receberão de você comunicações autênticas, tal como uma mail list.

Uma série de perguntas lhe direcionam no processo. Pressione **Enter** para atribuir um valor padrão se quiser. A primeira questão pede para você selecionar o tipo de chave que você prefere:

Por favor selecione qual tipo de chave você quer: (1) DSA e ElGamal (padrão) (2) DSA (apenas assinatura) (4) RSA (apenas assinatura). Sua seleção? Na maioria dos casos, o padrão é a escolha correta. Uma chave DSA/ElGamal lhe permite não somente assinar comunicações mas também criptografar arquivos.

Próximo, escolha o tamanho da chave: o tamanho mínimo é 768 bits, o tamanho padrão é 1024 bits e o máximo sugerido é 2048 bits. Qual tamanho de chave escolher? (1024). Novamente, o padrão é suficiente para a maioria dos usuários e representa um nível de segurança extremamente forte.

A seguir, escolha quando a chave irá expirar. É uma boa idéia escolher uma data de expiração em vez de usar o padrão, que é "nenhum". Se por exemplo, o endereço de e-mail na chave se tornar inválido, uma data de expiração avisará os outros para parar de usar essa chave pública.

Por favor especifique o tempo que a chave deve ser válida. 0 = a chave não expira, d = a chave expira em n dias, w = a chave expira em n semanas, m = a chave expira em n meses, y = a chave expira em n anos. A chave é valida por? (0)

Digitando o valor **1y**, por exemplo, faz a chave válida por 1 ano. (Você pode alterar essa data de expiração depois que a chave é gerada, se você mudar de idéia).

Antes do programa **gpg** perguntar por informações de assinatura, a seguinte pergunta aparece: **Está correto (s/n)?** Digite **s** para terminar o processo.

A seguir, digite seu nome e endereço de e-mail. Lembre-se que este processo é sobre autentica-lo como uma pessoa real. Por esta razão, inclua seu nome real. Não use apelidos ou códigos, já que esses disfarçam ou ofuscam sua identidade.

Digite seu endereço de e-mail real para sua chave GPG. Se você escolher um endereço de e-mail falso, será mais difícil para os outros encontrarem sua chave pública. Isto dificulta a autenticação de suas comunicações. Se você estiver usando essa chave GPG para [[DocsProject/SelfIntroduction| self-introduction]] em uma mail list, por exemplo, digite o endereço de e-mail que você usa nessa lista.

Use o campo de comentários para incluir apelidos e outras informações. (Algumas pessoas usam chaves diferentes para diferentes propósitos e identificam cada chave com um comentário, tal como "Office" ou "Open Source Projects.")

No prompt de confirmação, digite a letra O para continuar se todas as entradas estão corretas ou use as outras opções para consertar quaisquer problemas. Finalmente, digite uma frase secreta para sua chave secreta. O programa **gpg** pede para você digitar sua frase secreta duas vezes para assegurar que não houve erros de digitação.

Finalmente, o **gpg** gera dados aleatórios para fazer sua chave a mais única possível. Mova seu mouse, digite chaves aleatórias ou realize outras tarefas no sistema durante este passo para acelerar o processo. Uma vez que este passo estiver terminado, suas chaves estão completas e prontas para uso:

```
pub  1024D/1B2AFA1C 2005-03-31 John Q. Doe <jqdoe@example.com>
Key fingerprint = 117C FE83 22EA B843 3E86  6486 4320 545E 1B2A FA1C
sub  1024g/CEA4B22E 2005-03-31 [expires: 2006-03-31]
```

A chave de impressão digital é um atalho de "assinatura" para sua chave. Ela permite que você confirme aos outros que receberam a sua chave pública real, sem qualquer adulteração. Você não precisa anotar esta impressão digital. Para mostrar a impressão digital em qualquer momento, use este comando, substituindo com seu e-mail: **gpg --fingerprint jqdoe@example.com**

Sua "ID de chave GPG" consiste de 8 dígitos hex identificando a chave pública. No exemplo acima, a ID de chave GPG é 1B2AFA1C. Na maioria dos casos, se você for perguntado pela ID da chave, você deve prefixar "0x" na ID da chave, como em "0x1B2AFA1C".

ATENÇÃO

Se você esquecer a frase secreta, a chave não poderá ser usada e quaisquer dados criptografados usando essa chave serão perdidos.

3.9.4. Sobre Criptografia de Chave Pública

1. Wikipedia - Public Key Cryptography
2. HowStuffWorks - Encryption

CAPÍTULO 4. PRINCÍPIOS GERAIS DA SEGURANÇA DE INFORMAÇÃO

Os seguintes princípios gerais fornecem uma visão geral de boas práticas de segurança:

- criptografar todos os dados transmitidos via rede para ajudar a prevenir os ataques man-in-the-middle e eavesdropping. É importante criptografar as informações de autenticação, como senhas.
- minimizar a quantidade de software instalado e serviços de execução.
- use software de melhoria de segurança e ferramentas, por exemplo, Security-Enhanced Linux (SELinux) for Mandatory Access Control (MAC), Netfilter iptables para filtragem de pacotes (firewall), e GNU Privacy Guard (GnuPG) para arquivos criptografados.
- se possível, execute cada serviço de rede em um sistema separado para minimizar o risco de um serviço comprometido sendo usado para comprometer outros serviços.
- manter contas de usuário: criar e reforçar uma política de senha forte; remover contas de usuários sem uso.
- reveja o sistema diariamente e logs de aplicativos. Por padrão, os logs de sistema relevante a segurança são gravados em **`/var/log/secure`** e **`/var/log/audit/audit.log`**. Nota: o envio de logs ao servidor de log dedicado ajuda a prevenir atacantes de modificar com facilidade logs locais para evitar a detecção.
- nunca autentique-se como usuário root, a menos que absolutamente necessário. Recomenda-se que os administradores usem o **`sudo`** para executar comandos como root quando requerido. Os usuários capazes de executar o **`sudo`** são especificados em **`/etc/sudoers`**. Use o utilitário **`visudo`** para editar o **`/etc/sudoers`**.

4.1. DICAS, GUIAS E FERRAMENTAS

O National Security Agency (NSA) dos Estados Unidos, fornece guia de hardening e dicas para muitos sistemas operacionais diferentes, para ajudar agências governamentais, comércios e indivíduos a protegerem seus sistemas contra ataques. Os seguintes guias (em formato PDF) fornecem diretrizes para o Red Hat Enterprise Linux 6:

- Hardening Tips for the Red Hat Enterprise Linux 5
- Guia para Proteger Configuração do Red Hat Enterprise Linux 5

NOTA

São fornecidas referências ao guia de hardening do Red Hat Enterprise Linux 5 neste documento até que os guias de hardening do Red Hat Enterprise Linux 6 esteja disponíveis. Enquanto isso, por favor note que os guias de hardening do Red Hat Enterprise Linux 5 podem não se aplicar totalmente ao Red Hat Enterprise Linux 6.

O Defense Information Systems Agency (DISA) fornece documentação, checklists, e testes para ajudar a proteger seu sistema (Information Assurance Support Environment). O UNIX SECURITY TECHNICAL IMPLEMENTATION GUIDE (PDF) é um guia bastante específico para a segurança do UNIX, um conhecimento avançado do UNIX e Linux é recomendado antes de ler este guia.

O DISA Unix Security Checklist fornece a coleção de documentos e checklists, classificado entre propriedades corretas e modos para arquivos de sistema, para controle de reparos.

CAPÍTULO 5. INSTALAÇÃO SEGURA

A segurança inicia-se na primeira vez que você coloca o CD ou DVD em seu drive de disco para instalar o Red Hat Enterprise Linux. Configurar seu sistema de forma segura desde o início torna-o mais fácil de implementar configurações de segurança adicional mais tarde.

5.1. PARTIÇÕES DE DISCO

O NSA recomenda criar partições separadas para o /boot, /, /home, /tmp, e /var/tmp. As razões para cada um diferem e trataremos de cada partição.

/boot - Esta partição é a primeira partição que é lida pelo sistema durante a inicialização. O carregador de inicialização e imagens do kernel que são usadas para inicializar seu sistema em Red Hat Enterprise Linux são armazenadas nesta partição. Esta partição não deve ser criptografada. Se esta partição for incluída em / e essa partição for criptografada ou se tornar indisponível, seu sistema não poderá inicializar.

/home - Quando os dados de usuário (/home) são armazenados em / ao invés de serem armazenados em uma partição separada, a partição pode ficar cheia, fazendo com que o sistema operacional se torne instável. Da mesma forma, ao fazer o upgrade de seu sistema para uma próxima versão do Red Hat Enterprise Linux é muito mais fácil manter seus dados na partição /home, pois não será sobrescrito durante a instalação. Se a partição root (/) for corrompida, seus dados podem se perder para sempre. Ao usar uma partição separada, adicionará proteção contra a perda de dados. Você também pode escolher esta partição para fazer backups frequentes.

/tmp e /var/tmp - Tanto o diretório /tmp quanto o /var/tmp são usados para armazenar dados que não precisam ser armazenados por um longo período de tempo. No entanto, se estes diretórios ficarem sobrecarregados com muitos dados, consumirá seu espaço de armazenamento. Se isto acontecer e estes diretórios forem armazenados dentro do / então seu sistema poderá se tornar indisponível e travar. Por esta razão, mover estes diretórios para dentro de suas próprias partições é uma ótima idéia.

5.2. USE A CRIPTOGRAFIA DA PARTIÇÃO LUKS

Durante o processo de instalação será apresentada ao usuário uma opção para criptografar suas partições. O usuário deve fornecer uma senha frase que será a chave para desbloquear a chave de criptografia em massa que será usada para proteger os dados de partição.

CAPÍTULO 6. MANUTENÇÃO DO SOFTWARE

A manutenção do software é extremamente importante para manter um sistema seguro. É vital corrigir um software assim que a atualização esteja disponível para impedir que invasores usem as brechas para se infiltrar em seu sistema.

6.1. INSTALE O MÍNIMO DE SOFTWARE

A melhor prática é instalar somente pacotes que você usará porque cada instalação de software em sua máquina pode possivelmente conter uma vulnerabilidade. Se você estiver instalando a partir de uma mídia de DVD, use a oportunidade de selecionar exatamente os pacotes que você quiser instalar durante a instalação. Quando você achar que precisa de um outro pacote, você pode sempre adiciona-lo ao sistema mais tarde.

6.2. PLANEJE E CONFIGURE ATUALIZAÇÕES DE SEGURANÇA

Todos os softwares contém bugs. Frequentemente, estes bugs podem resultar em uma vulnerabilidade que pode expor seu sistema à usuários maliciosos. Sistemas sem correção são uma causa comum de instrusão em computadores. Você deve ter um plano para instalar correções de segurança em uma maneira agendada para impedir essas vulnerabilidades para que então não possam ser exploradas.

Para usuários domésticos, atualizações de segurança devem ser instaladas assim que possível. Configurar instalações automáticas das atualizações de segurança é uma maneira de não ter que ficar lembrando, mas possui um pequeno risco que alguma coisa pode causar um conflito com sua configuração ou com outro software no sistema.

Para usuários domésticos avançados ou de empresas, atualizações de segurança devem ser testadas e agendadas para instalação. Controles adicionais precisarão ser usados para proteger o sistema durante o período entre o lançamento da correção e sua instalação no sistema. Estes controles dependem da vulnerabilidade, mas podem incluir regras adicionais de firewall, o uso de firewalls externos ou mudanças nas configurações do software.

6.3. AJUSTANDO ATUALIZAÇÕES AUTOMÁTICAS

O Red Hat Enterprise Linux é configurado para aplicar todas as atualizações em uma programação diária. Se você quiser mudar como seu sistema instala as atualizações, você deve fazer pelo '"Software Update Preferences"' (Preferências de Atualização de Software). Você pode mudar a programação, os tipos de atualizações a serem aplicadas ou notificá-lo sobre atualizações disponíveis.

No Gnome, você pode encontrar controles de suas atualizações em **`System -> Preferences -> Software Updates`**. No KDE está localizado em **`Applications -> Settings -> Software Updates`**.

6.4. INSTALE PACOTES ASSINADOS DE REPOSITÓRIOS BEM CONHECIDOS

Pacotes de Software são publicados nos repositórios. Todos os repositórios bem conhecidos suportam assinatura de pacotes. Assinatura de Pacotes usam tecnologia de chave pública para provar que o pacote foi publicado pelo repositório e não foi alterado desde que a assinatura foi aplicada. Isto fornece certa proteção contra instalar o software que pode ter side maliciosamente modificado depois que o pacote foi criado mas antes de você te-lo baixado.

Usar muitos repositórios, repositórios não confiáveis, ou repositórios sem assinaturas de pacotes possui um risco maior de colocar um código malicioso ou vulnerável em seu sistema. Tenha cautela quando adicionar repositório ao yum/atualização de software.

CAPÍTULO 7. PADRÕES FEDERAIS E REGULAMENTAÇÃO

7.1. INTRODUÇÃO

Para manter os níveis de segurança, é possível que sua empresa se esforce para atender as medidas federais e especificações de segurança industrial, padrões e regulamentações. Este capítulo descreve alguns dos padrões e regulamentações.

7.2. FEDERAL INFORMATION PROCESSING STANDARD (FIPS)

The Federal Information Processing Standard (FIPS) Publicaton 140-2, é um padrão de segurança de computação, desenvolvido pelo Governo dos EUA e força de trabalho industrial para validar a qualidade de módulos criptográficos. As publicações FIPS (incluindo o 140-2) podem ser encontradas nas seguintes URL: http://csrc.nist.gov/publications/PubsFIPS.html. Observe que durante o processo de escrita deste, o Publication 140-3 se encontra em estado de Rascunho, e pode não representar o padrão completo. O padrão FIPS fornece quatro (4) *níveis* de segurança, para assegurar cobertura adequada de indústrias diferentes, implementações de módulos criptográficos e tamanhos e requerimentos organizacionais. Estes níveis são descritos abaixo:

- Nível 1 - O Nível de Segurança 1 fornece o nível mais baixo de segurança. Os requerimentos de segurança básicos são especificados por um módulo criptográfico (ex.: deve ser usado ao menos um algorítimo aprovado ou função de segurança Aprovada). Não é requerido nenhum mecanismo de segurança física específica em um módulo criptográfico de Segurança Nível 1, além dos requerimentos básicos para os componentes de grau de produção. Um exemplo de um Nível 1 de Segurança de módulo criptográfico é a placa de criptografia de um computador pessoal (PC).

- Nível 2 - O Nível de Segurança 2 aumenta os mecanismos de segurança física de um módulo criptográfico de Nível de Segurança 1, adicionando os requerimentos para cobertura ou selo tamper-evident ou para bloqueios pick-resistant em coberturas removíveis ou portas do módulo. A cobertura ou selo tamper-evident são colocadas em um módulo criptográfico para que a cobertura ou selo dese quebrada para obter acesso físico às chaves criptográficas de texto simples e parâmetros de segurança críticos (CSPs) dentro do módulo. Os selos tamper-evident ou bloqueios pick-resistant são colocados em coberturas ou portas para proteger contra acesso físico não autorizado.

- Nível 3 - Além dos mecanismos de segurança físicos tamper-evident requeridos no Nível de Segurança 2, o Nível de Segurança 3 tenta prevenir o intruso de obter acesso so CSPs mantido dentro do módulo criptográfico. Os mecanismos de segurança física requeridos no Nível de Segurança 3 podem ter alta possiblidade de detectar e responder à tentativas contra acesso físico, uso ou modificação de módulo criptográfico. Os mecanismos de segurança física podem incluir o uso de conteúdo forte e detecção de intromissão/ circuito de reposta que zera todos os CSPs de texto simples quando a cobertura/portas removíveis de módulo criptográfico estiverem abertos.

- Nível 4 - Nível de Segurança 4 fornece alto nível de segurança, definido neste padrão. Neste nível de segurança, os mecanismos de segurança física fornece um envelope completo de proteção ao redor de módulo criptográfico com a intenção de detectar e responder à todas as tentativas de acesso físico não autorizados. A penetração do conteúdo de módulo criptografico de qualquer direção possui uma alta probabilidade de ser detectado, resultando em zeroização de todos os CSPs de texto simples. Os módulos criptográficos de Nível de Segurança 4 são úteis para a operação em ambientes fisicamente desprotegidos.

Consulte o padrão completo FIPS 140-2 em: http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf para mais detalhes sobre estes níveis e outras especificações do padrão FIPS.

7.3. NATIONAL INDUSTRIAL SECURITY PROGRAM OPERATING MANUAL (NISPOM)

O NISPOM (também chamado de DoD 5220.22-M), como um componente do National Industrial Security Program (NISP), estabelece uma série de procedimentos e requerimentos para todos os contratantes do governo em relação a informações classificadas. O NISPOM atual é datado em 28 de Fevereiro de 2006. O documento NISPOM pode ser baixado da seguinte URL:https://www.dss.mil/GW/ShowBinary/DSS/isp/fac_clear/download_nispom.html.

7.4. PAYMENT CARD INDUSTRY DATA SECURITY STANDARD (PCI DSS)

Em https://www.pcisecuritystandards.org/about/index.shtml: *O PCI Security Standards Council é um forum global aberto, lançado em 2006, responsável pelo desenvolvimento, gerenciamento, educação e consciência do PCI Security Standards, incluindo o Data Security Standard (DSS).*

Você pode baixar o padrão PCI DSS a partir de https://www.pcisecuritystandards.org/security_standards/pci_dss.shtml.

7.5. GUIA DE IMPLEMENTAÇÃO TÉCNICO DE SEGURANÇA

O Guia de Implementação Técnico de Segurança ou STIG é uma metodologia para instalação e manutenção segura padronizada do software e hardware do computador.

Consulte a seguinte URL para obter uma lista dos guias disponíveis: http://iase.disa.mil/stigs/stig/index.html.

CAPÍTULO 8. REFERÊNCIAS

As referências a seguir são apontadores de informações adicionais que são relevantes ao SELinux e Red Hat Enterprise Linux mas além do escopo deste guia. Note que devido ao rápido desenvolvimento do SELinux, um pouco deste material pode ser aplicado somente em lançamentos específicos do Red Hat Enterprise Linux.

Livros

SELinux by Example

Mayer, MacMillan, and Caplan

Prentice Hall, 2007

Tutorial e Ajuda

Entendendo e Padronizando o Apache HTTP SELinux Policy

http://docs.fedoraproject.org/selinux-apache-fc3/

Tutorial e conversas com Russel Coker

http://www.coker.com.au/selinux/talks/ibmtu-2004/

Generic Writing SELinux policy HOWTO

http://www.lurking-grue.org/writingselinuxpolicyHOWTO.html

Base de Conhecimento Red Hat

http://kbase.redhat.com/

Informações Gerais

Website Principal do NSA SELinux

http://www.nsa.gov/selinux/

NSA SELinux FAQ

http://www.nsa.gov/selinux/info/faq.cfm

Fedora SELinux FAQ

http://docs.fedoraproject.org/selinux-faq/

SELinux NSA's Open Source Security Enhanced Linux

http://www.oreilly.com/catalog/selinux/

Tecnologia

Uma Visão Geral de Classes de Objetos e Permissões

http://www.tresys.com/selinux/obj_perms_help.html

Integrando Suporte Flexível para Políticas de Segurança no Sistema Operacional Linux (um histórico de Implementação do Flask no linux)

http://www.nsa.gov/research/_files/selinux/papers/selsymp2005.pdf

Implementando o SELinux como um Linux Security Module

http://www.nsa.gov/research/_files/publications/implementing_selinux.pdf

Uma configuração de política para o Linux Security-Enhanced

http://www.nsa.gov/research/_files/selinux/papers/policy/policy.shtml

Comunidade

Guia de Usuário do Fedora SELinux

http://docs.fedoraproject.org/en-US/Fedora/13/html/Security-Enhanced_Linux/

Fedora SELinux Managing Confined Services Guide

http://docs.fedoraproject.org/en-US/Fedora/13/html/Managing_Confined_Services/

Página da Comunidade do SELinux

http://selinuxproject.org/

IRC

irc.freenode.net, #selinux, #fedora-selinux, #security

HIstórico

Histórico breve do Flask

http://www.cs.utah.edu/flux/fluke/html/flask.html

Histórico completo do Fluke

http://www.cs.utah.edu/flux/fluke/html/index.html

APÊNDICE A. PADRÕES DE CRIPTOGRAFIA

A.1. CRIPTOGRAFIA SINCRONIZADA

A.1.1. Advanced Encryption Standard - AES

Na criptografia, o Advanced Encryption Standard (AES) é uma criptografia padrão adotada pelo governo dos E.U.A. O padrão consiste em três blocos de cífras, AES-128, AES-192 e AES-256, adotado por uma coleção maior originalmente publicada como Rijndael. Cada cífra AES possui um tamanho de bloco de 128 bites, com tamanhos de chaves de 128, 192 e 256 bites, e assim por diante. As cífras AES foram analisadas extensivamente e agora são utilizadas mundialmente, como foi o caso com seu precedente, o Data Encryption Standard (DES).[14]\n\t\n

A.1.1.1. Uso do AES

A.1.1.2. Histórico do AES

O AES foi anunciado pelo National Institute of Standards and Technology (NIST-Instituto Nacional de Padrões e Tecnologia) no dia 26 de Novembro de 2001 após 5 anos de processo de padronização no qual quinze modelos foram apresentados e avaliados antes que o Rijndael fosse selecionado como o mais adequado (veja o processo do Advanced Encryption Standard para obter mais detalhes). Ele foi efetivado com padrão no dia 26 de Maio de 2002. Está disponível em diversos pacotes de criptografia diferentes. O AES é a primeira cífra aberta acessível ao público pelo NSA para informações secretas (veja Segurança do AES, abaixo). [15]

A cífra Rijndael foi desenvolvida pelos critografadores Belgos Joan Daemen e Vincent Rijmen, e submetido por eles para o processo de seleção do AES. O Rijndael (pronuncia-se [rɛinda:l]) é uma palavra-valise do nome de dois inventores. [16]

A.1.2. Data Encryption Standard - DES

O Data Encryption Standard (DES) é uma cífra em bloco (uma forma de criptografia secreta compartilhada) que foi selecionada pelo National Bureau of Standards como um Padrão de Processamento de Informações Federal (FIPS - Federal Information Processing Standard) para os Estados Unidos em 1976 e o qual foi utilizado internacionalmente. Ele é baseado em algorítmos de chave simétrica que usam chaves de 56 bits. O algorítimo gerou controvérsias inicialmente com elementos de modelo classificados, um tamanho de chave relativamente pequeno, e esteve sob suspeita de conter uma backdoor (porta-dos-fundos) do National Security Agency (NSA). O DES consequentemente, foi criado sob uma análise detalhada acadêmica que motivou a compreensão moderna de cífras de bloco e suas criptoanálises. [17]

A.1.2.1. Uso do DES

A.1.2.2. Histórico do DES

O DES é considerado inseguro por muitos aplicativos. Isto se deve ao fato do tamanho da chave de 56 bits ser muito pequeno; em Janeiro, 1999, distributed.net e o Electronic Frontier Foundation colaboraram com a quebra pública de uma chave do DES em 22 horas e 15 minutos (veja a cronologia). Existem também alguns resultados analíticos que demonstraram fraqueza teorética na cífra, embora sejam impossíveis de se montar na prática. Acredita-se que o algorítmo seja praticamente seguro na forma de Triple DES, embora existam ataques teoréticos. Recentemente, a cífra foi substituída pela cífra Advanced Encryption Standard (AES).[18]

Em algumas documentações, foi feita uma distinção entre o DES como um padrão e o DES algorítimo que é referido como o DEA o Data Encryption Algorithm). Na fala, o "DES" é soletrado como uma abreviação (/ˌdiːˌiːˈɛs/), ou pronunciado como um acronismo de uma sílaba (/ˈdɛz/).[19]

A.2. CRIPTOGRAFIA DE CHAVE PÚBLICA

A criptografia de chave pública é uma forma de criptografia, empregada por muitos algorítimos criptográficos e criptosistemas, cujas características distintas são o uso de algorítimos de chave assimétricos ao invés de ou além de algorítimos de chaves simétricas. O uso das técnicas de criptografia pública de chave privada, muitos métodos de proteção de comunicação ou autenticação de mensagens antes desconhecidas se tornaram práticas. Elas não requerem uma troca inicial segura de uma ou mais chaves secretas como é requerido ao utilizar algoríticmos de chave simétricas. Ele pode também ser usado para criar assinaturas digitais. [20]

Criptografia de chave pública é fundamental e uma tecnologia amplamente utilizada mundialmente, e é a forma que contém tais padrões de Internet como Segurança de Camada de Transporte (TLS) (sucessor do SSL), PGP e GPG. [21]

A técnica distinta usada na criptografia de chave pública é o uso de algorítmos de chave assimétricos, onde a chave usada para criptografar uma mensagem não é a mesma que a chave usada para descriptografá-la. Cada usuário possui um par de chaves criptográficas, uma chave pública e uma chave privada. A chave privada é mantida em segredo, enquanto a chave pública pode ser distribuída amplamente. As mensagens são criptografadas com a chave pública do rementente e pode ser dscriptografada somente com a chave privada correpondente. As chaves são relacionadas matematicamente, mas a chave privada não pode ser derivada (ou seja, em prática atual ou projetada) de uma chave pública. Foi a descoberta de alguns algorítmos que revolucionaou a prática de criptografia iniciando-se nos meados de 1970. [22]

Oposto a isto, os algorítimos de chave simétrica, variações que foram usadas por alguns milhares de anos, usam uma chave secreta única compartilhada pelo transmissor e remetente (que deve também manter a chave privada, além de levar em conta a ambiguidade das terminologias comuns) para a criptografia e descriptografia. para usar um esquema de criptografia simétrica, o transmissor e remetente devem compartilhar a chave antes. [23]

Como os algorítimos de chave simétricas são quase muito menos intensivos na informática, é comum trocar uma chave usando o algorítimo de troca de chave e transmitir dados usando aquela chave e um algorítimos de chave simétrica. A família de esquemas do PGP e o SSL/TLS fazem isto, por exemplo, e são portanto chamados de critosistemas híbridos. [24]

A.2.1. Diffie-Hellman

A troca de chave Diffie–Hellman (D-H) é um protocolo criptográfico que permite que duas partes, que não se conhecem préviamente, estabeleçam juntas uma chave secreta compartilhada sob um canal de comunicações inseguro. Esta chave pode então ser usada para criptografar comunicações subsequentes usando uma cífra de chave simétrica. [25]

A.2.1.1. Histórico do Diffie-Hellman

O esquema foi publicado inicialmente pelo Whitfield Diffie e Martin Hellman em 1976, embora tenha emergido mais tarde que havia sido inventado separadamente alguns anos antes dentro do GCHQ, a agência de inteligência de sinais Britânicos, por Malcolm J. Williamson mas foi mantida em classificado. Em 2002, Hellman sugeriu que o algorítimo fosse chamado de chave de troca Diffie-Hellman-Merkle reconhecendo a contribuição de Ralph Merkle para a invensão da criptografia de chave pública (Hellman, 2002). [26]

Embora o acordo da Diffie-Hellman seja um protocolo de acordo de chave anônimo (não autenticado), ele fornece a base para uma variedade de protocolos autenticados, e é usado para fornecer segredo perfeito nos modos efêmeros da Segurança de Camada de Transporte, (referido como o EDH ou DHE dependendo da cífra que se adeque). [27]\n\n

A Patente 4,200,770 dos E.U.A, agora expirada, descreve o algorítimo e dá crédito ao Hellman, Diffie e Merkle como inventores. [28]

A.2.2. RSA

Na criptografia, o RSA (que significa Rivest, Shamir e Adleman que primeiro descreveram-no publicamente, veja abaixo) é um algorítimo para criptografia de chave pública. É o primeiro algorítimo conhecido como adequado para assinaturas, assim como criptografia, e foi o primeiro grande avanço em criptografia de chave pública. O RSA é usado amplamente em protocolos de comércio eletrônico, e acredita-se que é seguro, considerando as chaves longas e o uso de implementações atualizadas.

A.2.3. DSA

O DSA (Digital Signature Algorithm) é um padrão para assinaturas digitais padrão, um padrão de governo federal dos Estados Unidos para assinaturas digitais. O DSA é somente para assinaturas e não é um algoritimo de criptografia. [29]

A.2.4. SSL/TLS

O Transport Layer Security (TLS) é seu precedente, o Secure Sockets Layer (SSL), são protocolos criptográficos que fornecem segurança para as comunicações sob a rede, tais como a Internet. O TLS e SSL criptografam os segmentos de conexões de rede do Transport Layer do começo ao fim.

Diversas versões dos protocolos são utilizadas amplamente em aplicativos como o web browsing, correio eletrônico, fax via Internet, mensagem instantânea e voice-over-IP (VoiP). [30]

A.2.5. Cramer-Shoup Cryptosystem

O sistema Cramer-Shoup é um algorítimos de criptografia assimétrica, e foi o primeiro esquema eficiente que provou ser seguro em ataques de textos de cifras escolhidos como adaptáveis, usando presunções criptográficas padrão. Sua segurança é baseada em intractabilidade computacional (presumido amplamente, mas ainda não foi provado) de presunções de do Diffie-Hellman. Desenvolvido por Ronald Cramer e Victor Shoup, em 1998, é uma estensão maleável, o Cramer-Shoup adiciona elementos para assegurar a falta de maleabilidade mesmo contra atacantes munidos de recursos. Esta não maleabilidade é alcançada através do uso da função de hash resistente à colisão e outras tecnologias, resultando em texto cífra que é duas vezes maior do que em Elgamal. [31]

A.2.6. ElGamal Encryption

Na criptografia, o sistema de criptografia ElGamal é um algorítimo de criptografia assimétrica para criptografia de chave pública, que é baseada no acordo de chave do Diffie-Hellman. Foi descrito por Taher Elgamal em 1985. A criptografia do ElGamal é usada no software livre GNU Privacy Guard, versões recentes do PGP, e outros criptosistemas. [32]

---

[14] "Advanced Encryption Standard." *Wikipedia.* 14 November 2009 http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

[15] "Advanced Encryption Standard." *Wikipedia.* 14 November 2009 http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

[16] "Advanced Encryption Standard." *Wikipedia.* 14 November 2009 http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

[17] "Data Encryption Standard." *Wikipedia.* 14 November 2009 http://en.wikipedia.org/wiki/Data_Encryption_Standard

[18] "Data Encryption Standard." *Wikipedia.* 14 November 2009 http://en.wikipedia.org/wiki/Data_Encryption_Standard

[19] "Data Encryption Standard." *Wikipedia.* 14 November 2009 http://en.wikipedia.org/wiki/Data_Encryption_Standard

[20] "Public-key Encryption." *Wikipedia.* 14 November 2009 http://en.wikipedia.org/wiki/Public-key_cryptography

[21] "Public-key Encryption." *Wikipedia.* 14 November 2009 http://en.wikipedia.org/wiki/Public-key_cryptography

[22] "Public-key Encryption." *Wikipedia.* 14 November 2009 http://en.wikipedia.org/wiki/Public-key_cryptography

[23] "Public-key Encryption." *Wikipedia.* 14 November 2009 http://en.wikipedia.org/wiki/Public-key_cryptography

[24] "Public-key Encryption." *Wikipedia.* 14 November 2009 http://en.wikipedia.org/wiki/Public-key_cryptography

[25] "Diffie-Hellman." *Wikipedia.* 14 November 2009 http://en.wikipedia.org/wiki/Diffie-Hellman

[26] "Diffie-Hellman." *Wikipedia.* 14 November 2009 http://en.wikipedia.org/wiki/Diffie-Hellman

[27] "Diffie-Hellman." *Wikipedia.* 14 November 2009 http://en.wikipedia.org/wiki/Diffie-Hellman

[28] "Diffie-Hellman." *Wikipedia.* 14 November 2009 http://en.wikipedia.org/wiki/Diffie-Hellman

[29] "DSA." *Wikipedia.* 24 February 2010 http://en.wikipedia.org/wiki/Digital_Signature_Algorithm

[30] "TLS/SSl." *Wikipedia.* 24 February 2010 http://en.wikipedia.org/wiki/Transport_Layer_Security

[31] "Cramer-Shoup cryptosystem." *Wikipedia.* 24 February 2010 http://en.wikipedia.org/wiki/Cramer–Shoup_cryptosystem

[32] "ElGamal encryption" *Wikipedia.* 24 February 2010 http://en.wikipedia.org/wiki/ElGamal_encryption

APÊNDICE B. HISTÓRICO DE REVISÃO

Revisão 1.5-3.35.402 **Fri Oct 25 2013** **Rüdiger Landmann**
Rebuild with Publican 4.0.0

Revisão 1.5-3.35 **August 7 2012** **Ruediger Landmann**
Rebuild for Publican 3.0

Revisão 1.5-3 **2012-07-18** **Anthony Towns**
Rebuild for Publican 3.0

Revisão 1.5-1 **Apr 19 2010** **Scott Radvan**
Reparos mínimos, construção final para Beta

Revisão 1.4-1 **Mar 5 2010** **Scott Radvan**
Revisão QE e Atualizações

Revisão 1.3-1 **Feb 19 2010** **Scott Radvan**
Enviar para área de teste pronto para revisão