**Introdução**

4.0.1

**Por que devo cursar este módulo?**

Linux é um sistema operacional de código aberto que é rápido, poderoso e altamente personalizável. Ele é construído para uso em rede como cliente ou servidor. O Linux é bem amado por uma grande comunidade de usuários, incluindo pessoal de segurança cibernética.

Estude este módulo para aprender algumas informações básicas sobre o sistema operacional Linux e aprender algumas habilidades Linux. As habilidades do Linux são altamente desejáveis na profissão de operações de segurança cibernética.

4.0.2

**O que vou aprender neste módulo?**

**Título do módulo:** Visão geral do Linux

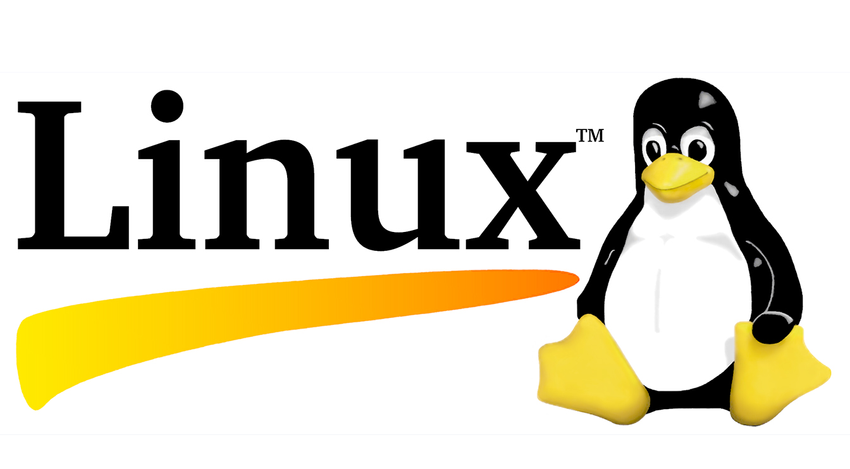
**Objetivo do módulo:** Implementar segurança básica do Linux.

|  |  |
| --- | --- |
| **Título do Tópico** | **Objetivo do Tópico** |
| Linux Básico | Explicar por que as habilidades do Linux são essenciais para o monitoramento e a investigação de segurança de rede. |
| Trabalhando no Linux Shell | Usar o Linux Shell para manipular arquivos de texto. |
| Servidores e clientes Linux | Explicar como funcionam as redes client-server. |
| Administração Básica do Servidor | Explique como um administrador Linux localiza e manipula arquivos de log de segurança. |
| O sistema de arquivos Linux | Gerenciar o sistema de arquivos e as permissões do Linux. |
| Trabalhando na GUI do Linux | Explicar os componentes básicos da GUI do Linux. |
| Trabalhando em um host Linux | Usar ferramentas para detectar malware em um host do Linux. |

# Linux Básico

4.1.1

## O que é Linux?



Linux é um sistema operacional criado em 1991. O Linux é de código aberto, rápido, confiável e pequeno. Ele requer muito poucos recursos de hardware para ser executado e é altamente personalizável. Ao contrário de outros sistemas operacionais, como Windows e Mac OS X, Linux foi criado, e atualmente é mantido por, uma comunidade de programadores. Linux faz parte de várias plataformas e pode ser encontrado em dispositivos de qualquer lugar, desde “relógios de pulso a supercomputadores”.

Outro aspecto importante do Linux é que ele foi projetado para ser conectado à rede, o que torna muito mais simples escrever e usar aplicativos baseados em rede. Como o Linux é open source, qualquer pessoa ou empresa pode obter o código-fonte do kernel, inspecioná-lo, modificá-lo e recompilá-lo à vontade. Eles também estão autorizados a redistribuir o programa com ou sem encargos.

Uma distribuição Linux é o termo usado para descrever pacotes criados por diferentes organizações. Distribuições Linux (ou distros) incluem o kernel Linux com ferramentas personalizadas e pacotes de software. Embora algumas dessas organizações possam cobrar por seu suporte de distribuição Linux (voltado para empresas baseadas em Linux), a maioria delas também oferece sua distribuição gratuitamente sem suporte. Debian, Red Hat, Ubuntu, CentOS e SUSE são apenas alguns exemplos de distribuições Linux.

4.1.2

## O valor do Linux

O Linux é frequentemente o sistema operacional escolhido no Centro de Operações de Segurança (SOC). Estas são algumas das razões para escolher o Linux:

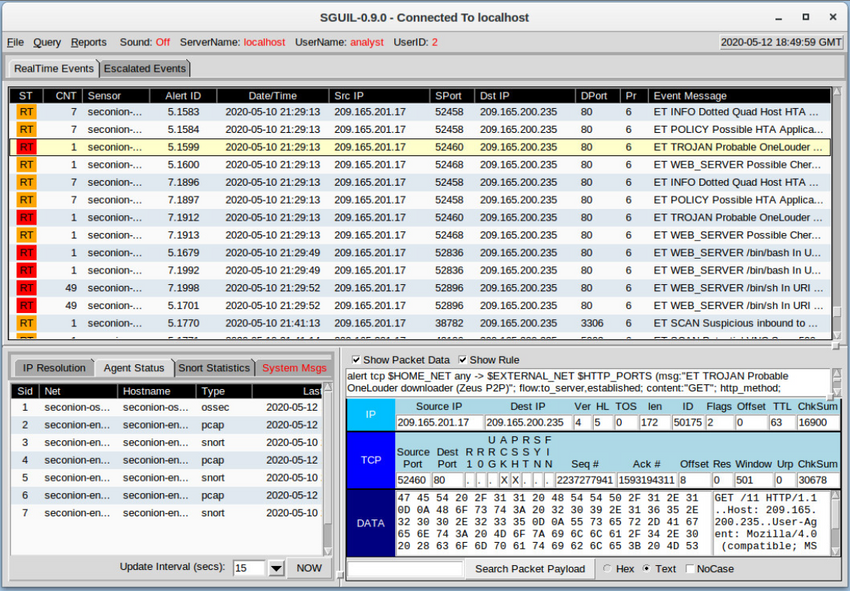
* **Linux é open source** - Qualquer pessoa pode adquirir Linux gratuitamente e modificá-lo para atender a necessidades específicas. Essa flexibilidade permite que analistas e administradores criem um sistema operacional especificamente para análise de segurança.
* **A CLI do Linux é muito poderosa** - Embora uma GUI torne muitas tarefas mais fáceis de executar, ela adiciona complexidade e requer mais recursos de computador para executar. A CLI (Command Line Interface) Linux é extremamente poderosa e permite que os analistas executem tarefas não apenas diretamente em um terminal, mas também remotamente.
* **O usuário tem mais controle sobre o sistema operacional** - O usuário administrador no Linux, conhecido como o usuário root, ou superusuário, tem poder absoluto sobre o computador. Ao contrário de outros sistemas operacionais, o usuário raiz pode modificar qualquer aspecto do computador com algumas teclas pressionadas. Essa capacidade é especialmente valiosa quando se trabalha com funções de baixo nível, como a pilha de rede. Ele permite que o usuário raiz tenha controle preciso sobre a maneira como os pacotes de rede são manipulados pelo sistema operacional.
* **Ele permite um melhor controle de comunicação de rede** - Controle é uma parte inerente do Linux. Como o sistema operacional pode ser ajustado em praticamente todos os aspectos, é uma ótima plataforma para a criação de aplicativos de rede. Esta é a mesma razão pela qual muitas grandes ferramentas de software baseadas em rede estão disponíveis apenas para Linux.

4.1.3

## Linux no SOC

A flexibilidade fornecida pelo Linux é um ótimo recurso para o SOC. Todo o sistema operacional pode ser adaptado para se tornar a plataforma de análise de segurança perfeita. Por exemplo, os administradores podem adicionar apenas os pacotes necessários ao sistema operacional, tornando-o simples e eficiente. Ferramentas de software específicas podem ser instaladas e configuradas para funcionar em conjunto, permitindo que os administradores criem um computador personalizado que se encaixe perfeitamente no fluxo de trabalho de um SOC.

A figura mostra o Sguil, que é o console do analista de segurança cibernética em uma versão especial do Linux chamada Security Onion. O Security Onion é um conjunto de ferramentas de código aberto que trabalham juntas para análise de segurança de rede. Trabalharemos com Security Onion mais tarde neste curso.



A tabela lista algumas ferramentas que são freqüentemente encontradas em um SOC.

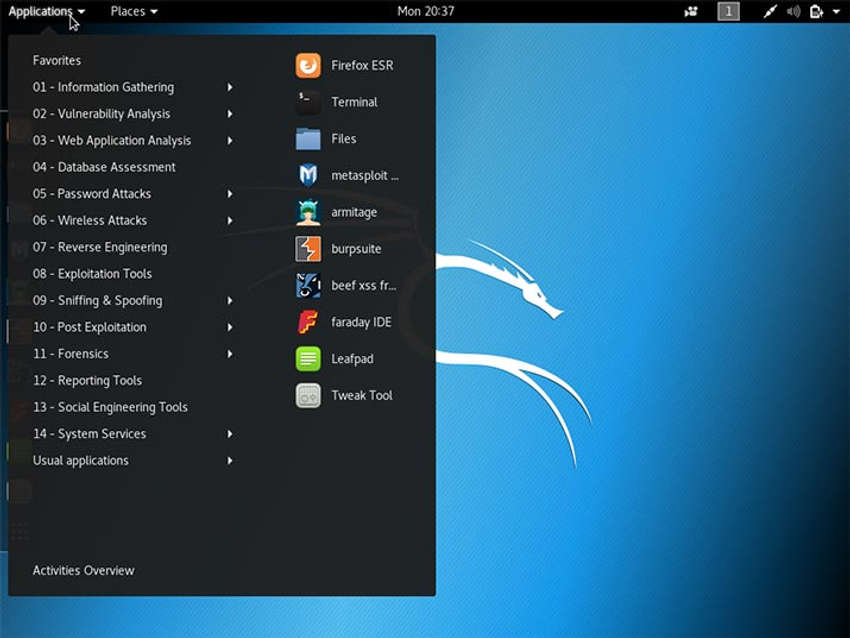
| **Ferramenta SOC** | **Descrição** |
| --- | --- |
| **Software de captura de pacotes de rede** | * Uma ferramenta crucial para um analista de SOC, pois permite observar e entender cada detalhe de uma transação de rede. * Wireshark é uma ferramenta popular de captura de pacotes. |
| **Ferramentas de análise de malware** | Essas ferramentas permitem que os analistas executem e observem com segurança a execução de malware sem o risco de comprometer o sistema subjacente. |
| **Sistemas de detecção de intrusão (IDSs)** | * Essas ferramentas são usadas para monitoramento e inspeção de tráfego em tempo real. * Se qualquer aspecto do tráfego atualmente em fluxo corresponder a qualquer uma das regras estabelecidas, uma ação predefinida será executada. |
| **Firewalls** | Este software é usado para especificar, com base em regras predefinidas, se o tráfego tem permissão para entrar ou sair de uma rede ou dispositivo. |
| **Gerenciadores de log** | * Os arquivos de log são usados para registrar eventos. * Como uma rede pode gerar um número muito grande de entradas de log, o software do gerenciador de logs é empregado para facilitar o monitoramento de log. |
| **Segurança das informações e gerenciamento de eventos (SIEM)** | Os SIEMs fornecem análise em tempo real de alertas e entradas de log geradas por dispositivos de rede, como IDSs e firewalls. |
| **Sistemas de emissão de bilhetes** | A atribuição de tíquetes de tarefa, edição e gravação é feita através de um sistema de gerenciamento de tíquetes. Os alertas de segurança são frequentemente atribuídos a analistas por meio de um sistema de emissão de bilhetes. |

4.1.4

## Ferramentas Linux

Além das ferramentas específicas de SOC, os computadores Linux que são usados no SOC geralmente contêm ferramentas de teste de penetração. Também conhecido como Pentesting, um teste de penetração é o processo de procurar vulnerabilidades em uma rede ou computador atacando-o. Geradores de pacotes, scanners de porta e explorações de prova de conceito são exemplos de ferramentas Pentesting.

Kali Linux é uma distribuição Linux agrupa muitas ferramentas de penetração juntas em uma única distribuição Linux. Kali contém uma grande variedade de ferramentas. A figura mostra uma captura de tela do Kali Linux. Observe todas as principais categorias de ferramentas de teste de penetração.



# Trabalhando no Linux Shell

4.2.1

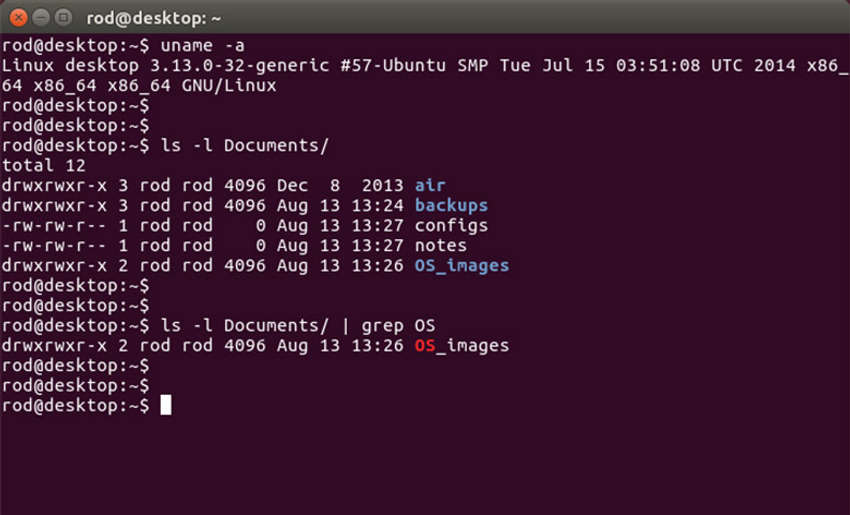
## O Shell do Linux

No Linux, o usuário se comunica com o SO usando a CLI ou a GUI. O Linux geralmente inicia na GUI por padrão. Isso oculta a CLI do usuário. Uma maneira de acessar a CLI a partir da GUI é por meio de um aplicativo de emulador de terminal. Esses aplicativos fornecem acesso do usuário ao CLI e geralmente são nomeados como uma variação da palavra "terminal". No Linux, emuladores de terminal populares são Terminator, eterm, xterm, konsole e gnome-terminal.

Fabrice Bellard criou JSLinux que permite que uma versão emulada do Linux seja executada em um navegador. Procure por ele na internet. Abra um console Linux em JSLinux e digite o **ls** comando para listar o conteúdo do diretório atual. Mantenha a guia aberta se você gostaria de experimentar alguns dos outros comandos discutidos neste capítulo.

A figura mostra o gnome-terminal, um emulador de terminal Linux popular.

**Observação**: Os termos shell, console, janela do console, terminal da CLI e janela do terminal são frequentemente usados de forma intercambiável.



4.2.2

## Comandos básicos

Os comandos Linux são programas criados para executar uma tarefa específica. Use o **man** comando (abreviação para manual) para obter documentação sobre comandos. Como exemplo, **man ls** fornece documentação sobre o **ls** comando a partir do manual do usuário.

Como os comandos são programas armazenados no disco, quando um usuário digita um comando, o shell deve encontrá-lo no disco antes que ele possa ser executado. O shell procurará comandos digitados pelo usuário em diretórios específicos e tentará executá-los. A lista de diretórios verificados pelo shell é chamada de caminho. O caminho contém muitos diretórios comumente usados para armazenar comandos. Se um comando não estiver no caminho, o usuário deve especificar sua localização, ou o shell não será capaz de encontrá-lo. Os usuários podem facilmente adicionar diretórios ao caminho, se necessário.

Para invocar um comando através do shell, basta digitar seu nome. O shell tentará encontrá-lo no caminho do sistema e executá-lo.

A tabela lista comandos básicos do Linux e suas funções.

| **Comando** | **Descrição** |
| --- | --- |
| **mv** | Move ou renomeia arquivos e diretórios |
| **chmod** | Modifica as permissões de arquivos |
| **chown** | Altera o dono de um arquivo |
| **dd** | Copia os dados de uma entrada para uma saída |
| **pwd** | Exibe o nome do diretório atual |
| **ps** | Lista os processos que estão atualmente em execução no sistema |
| **su** | Simula um login como outro usuário ou para se tornar um superusuário |
| **sudo** | Executa um comando como um superusuário, por padrão, ou outro usuário nomeado |
| **grep** | Usado para pesquisar cadeias de caracteres específicas em um arquivo ou outras saídas de comando. Para pesquisar através da saída de um comando anterior, **grep** deve ser canalizado no final do comando anterior. |
| **ifconfig** | Usado para exibir ou configurar informações relacionadas à placa de rede. Se emitido sem parâmetros, o **ifconfig** exibirá a configuração atual da (s) placa (s) de rede. Observação: Embora ainda esteja amplamente em uso, esse comando está obsoleto. Use o endereço **IP** em vez disso. |
| **apt-obter** | Usado para instalar, configurar e remover pacotes no Debian e seus derivados. Nota: **apt-get** é um front-end de linha de comando amigável para o **dpkg**, o gerenciador de pacotes do Debian. O combo **dpkg** e **apt-get** é o sistema de gerenciador de pacotes padrão em todas as derivadas Debian Linux, incluindo Raspbian. |
| **iwconfig** | Usado para exibir ou configurar informações relacionadas à placa de rede sem fio. Semelhante ao **ifconfig**, o **iwconfig** exibirá informações sem fio quando emitido sem parâmetros. |
| **shutdown** | Desliga o sistema, o **desligamento** pode ser instruído a executar uma série de tarefas relacionadas ao encerramento, incluindo reiniciar, parar, colocar em suspensão ou expulsar todos os usuários conectados no momento. |
| **passwd** | Usado para alterar a senha. Se nenhum parâmetro for fornecido, **passwd** altera a senha do usuário atual. |
| **cat** | Usado para listar o conteúdo de um arquivo e espera o nome do arquivo como parâmetro. O **comando** cat geralmente é usado em arquivos de texto. |
| **man** | Usado para exibir a documentação de um comando específico. |

**Nota**: Supõe-se que o usuário tenha as permissões adequadas para executar o comando. As permissões de arquivos no Linux são abordadas posteriormente neste capítulo.

4.2.3

## Comandos de arquivo e diretório

Muitas ferramentas de linha de comando estão incluídas no Linux por padrão. Para ajustar a operação do comando, os usuários podem passar parâmetros e opções junto com o comando. A tabela lista alguns dos comandos mais comuns relacionados a arquivos e diretórios.

| **Comando** | **Descrição** |
| --- | --- |
| **ls** | Exibe os arquivos dentro de um diretório |
| **cd** | Muda o diretório atual |
| **mkdir** | Cria um diretório no diretório atual |
| **cp** | Copia arquivos da origem para o destino |
| **mv** | Move os arquivos para um diretório diferente |
| **rm** | Remove arquivos |
| **grep** | Pesquisa cadeias de caracteres específicas em um arquivo ou outras saídas de comandos |
| **cat** | Lista o conteúdo de um arquivo e espera o nome do arquivo como parâmetro |

4.2.4

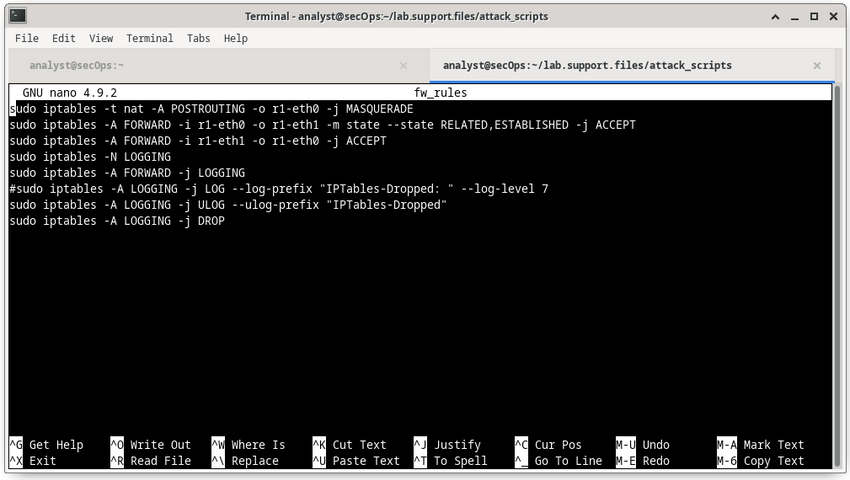
## Trabalhando com Arquivos de Texto

Linux tem muitos editores de texto diferentes, com vários recursos e funções. Alguns editores de texto incluem interfaces gráficas, enquanto outros são apenas ferramentas de linha de comando. Cada editor de texto inclui um conjunto de recursos projetado para suportar um tipo específico de tarefa. Alguns editores de texto se concentram no programador e incluem recursos como realce de sintaxe, parênteses e verificação de parênteses e outros recursos focados na programação.

Embora os editores de texto gráficos sejam convenientes e fáceis de usar, os editores de texto baseados em linha de comando são muito importantes para os usuários do Linux. O principal benefício dos editores de texto baseados em linha de comando é que eles permitem a edição de arquivos de texto a partir de um computador remoto.

Considere o seguinte cenário: um usuário deve executar tarefas administrativas em um computador Linux, mas não está sentado na frente desse computador. Usando SSH, o usuário inicia um shell remoto para o computador remoto. Sob o shell remoto baseado em texto, a interface gráfica não está disponível, o que torna impossível confiar em ferramentas como editores de texto gráficos. Neste tipo de situação, programas baseados em texto são cruciais.

A figura mostra **nano**, um editor de texto de linha de comando popular. O administrador está editando regras de firewall. Editores de texto são frequentemente usados para configuração e manutenção do sistema no Linux.



Devido à falta de suporte gráfico, nano (ou GNU nano) só pode ser controlado com o teclado. Por exemplo, **CTRL+O** salva o arquivo atual; **CTRL+W** abre o menu de pesquisa. GNU nano usa uma barra de atalho de duas linhas na parte inferior da tela, onde os comandos para o contexto atual são listados. Pressione **CTRL+G** para obter a tela de ajuda e uma lista completa de comandos.

4.2.5

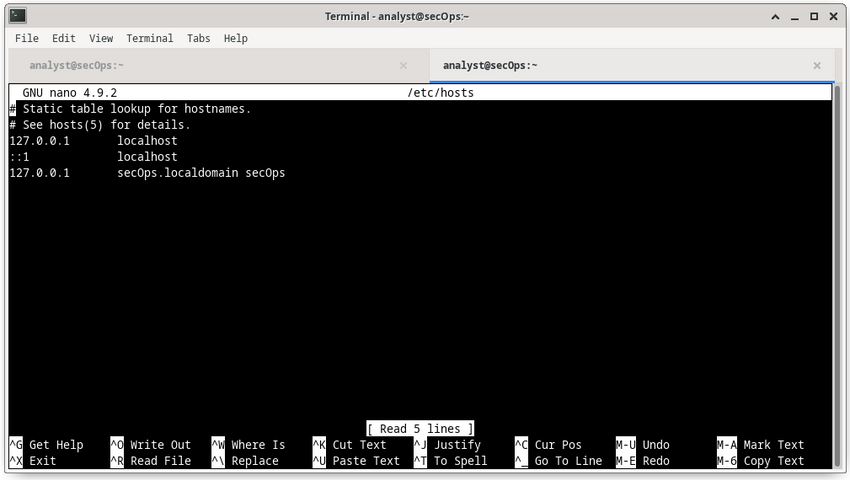
## A importância dos arquivos de texto no Linux

No Linux, tudo é tratado como um arquivo. Isso inclui a memória, os discos, o monitor e os diretórios. Por exemplo, do ponto de vista do sistema operacional, mostrar informações na tela significa gravar no arquivo que representa o dispositivo de exibição. Não deve ser surpresa que o próprio computador esteja configurado através de arquivos. Conhecidos como arquivos de configuração, eles geralmente são arquivos de texto usados para armazenar ajustes e configurações para aplicativos ou serviços específicos. Praticamente tudo no Linux depende de arquivos de configuração para funcionar. Alguns serviços têm não um, mas vários arquivos de configuração.

Os usuários com níveis de permissão adequados podem usar editores de texto para alterar o conteúdo dos arquivos de configuração. Depois que as alterações são feitas, o arquivo é salvo e pode ser usado pelo serviço ou aplicativo relacionado. Os usuários podem especificar exatamente como querem que qualquer aplicativo ou serviço se comporte. Quando iniciados, serviços e aplicativos verificam o conteúdo de arquivos de configuração específicos para ajustar seu comportamento de acordo.

Na figura, o administrador abriu o arquivo de configuração do host **nano** para edição. O arquivo host contém mapeamentos estáticos de endereços IP do host para nomes. Os nomes servem como atalhos que permitem a conexão com outros dispositivos usando um nome em vez de um endereço IP. Somente o superusuário pode alterar o arquivo host.

**Observação**: O administrador usou o comando \*\*sudo nano /etc/hosts\*\* para abrir o arquivo. O comando \*\*sudo\*\* (abreviação de “superusuário do”) invoca o privilégio de superusuário para usar o editor de texto nano para abrir o arquivo host.



4.2.6

## Laboratório — Trabalhando com arquivos de texto na CLI

Neste laboratório, você se familiarizará com editores de texto de linha de comando Linux e arquivos de configuração.

4.2.7

## Laboratório — Familiarizando-se com o Linux Shell

Neste laboratório, você usará a linha de comando do Linux para gerenciar arquivos e diretórios e realizar algumas tarefas administrativas básicas.

# Servidores e clientes Linux

4.3.1

## Uma Introdução às Comunicações Cliente-Servidor

Os servidores são computadores com software instalado que lhes permite fornecer serviços aos clientes em toda a rede. Existem muitos tipos de serviços. Alguns fornecem recursos externos, como arquivos, mensagens de e-mail ou páginas da Web para clientes mediante solicitação. Outros serviços executam tarefas de manutenção, como gerenciamento de registros, gerenciamento de memória, varredura de disco e muito mais. Cada serviço exige um software de servidor separado. Por exemplo, o servidor na figura usa um software de servidor de arquivos para fornecer aos clientes a capacidade de recuperar e enviar arquivos.

As comunicações cliente-servidor são discutidas com mais detalhes posteriormente no curso.

A imagem é um diagrama que mostra um servidor enviando arquivos para um cliente. Os recursos são armazenados no servidor. Um cliente é uma combinação de hardware/software que as pessoas usam diretamente. Arquivos são transferidos do servidor para o cliente.

Arquivos são transferidos do servidor para o cliente.DownloadClienteServidorRedeOs recursos são armazenados no servidor.Um cliente é uma combinação de hardware/software que as pessoas usam diretamente.

4.3.2

## Servidores, serviços e suas portas

Para que um computador possa ser o servidor de vários serviços, as portas são usadas. Uma porta é um recurso de rede reservado usado por um serviço. Um servidor é dito estar “escutando” em uma porta quando ele se associou a essa porta.

Embora o administrador possa decidir qual porta usar com qualquer serviço específico, muitos clientes são configurados para usar uma porta específica por padrão. É prática comum deixar o serviço em execução em sua porta padrão. A tabela lista algumas portas comumente usadas e seus serviços. Estes também são chamados de “portas bem conhecidas”.

| **Porta** | **Descrição** |
| --- | --- |
| **20/21** | File Transfer Protocol (FTP) |
| **22** | Secure Shell (SSH) |
| **23** | Serviço de login remoto Telnet |
| **25** | Protocolo SMTP |
| **53** | Domain Name System (DNS) |
| **67/68** | Protocolo de Configuração Dinâmica de Host (DHCP) |
| **69** | Protocolo de Transferência Trivial de Arquivo (TFTP) |
| **80** | Protocolo HTTP |
| **110** | Protocolo POP3 (Post Office Protocol - Protocolo dos Correios) |
| **123** | Network Time Protocol (NTP) |
| **143** | Protocolo IMAP |
| **161/162** | Protocolo de Gerenciamento Simples de Rede (SNMP) |
| **443** | HTTP seguro (HTTPS) |

As portas e seus usos nas comunicações de rede são discutidos mais detalhadamente no final do curso.

4.3.3

## Clientes

Os clientes são programas ou aplicativos projetados para se comunicar com um tipo específico de servidor. Também conhecidos como aplicativos cliente, os clientes usam um protocolo bem definido para se comunicar com o servidor. Os navegadores da Web são clientes da Web que são usados para se comunicar com servidores Web por meio do Hyper Text Transfer Protocol (HTTP) na porta 80. O cliente FTP (File Transfer Protocol) é um software usado para se comunicar com um servidor FTP. A figura mostra um cliente fazendo upload de arquivos para um servidor.

A imagem é um diagrama que mostra um cliente carregando arquivos para um servidor. Um cliente é uma combinação de hardware/software que as pessoas usam diretamente. Os arquivos são carregados do cliente para o armazenamento no servidor. Os recursos são armazenados no servidor.

Os arquivos são carregados do cliente para o armazenamento no servidor.UploadClienteServidorRedeOs recursos são armazenados no servidor.Um cliente é uma combinação de hardware/software que as pessoas usam diretamente.

4.3.4

## Laboratório - Servidores Linux

Neste laboratório, você usará a linha de comando do Linux para identificar servidores que estão sendo executados em um computador.

# Administração Básica do Servidor

4.4.1

## Arquivos de configuração de serviço

No Linux, os serviços são gerenciados usando arquivos de configuração. As opções comuns nos arquivos de configuração são o número da porta, a localização dos recursos hospedados e os detalhes da autorização do cliente. Quando o serviço é iniciado, ele procura seus arquivos de configuração, carrega-os na memória e se ajusta de acordo com as configurações nos arquivos. As modificações do arquivo de configuração geralmente exigem a reinicialização do serviço antes que as alterações entrem em vigor.

Como os serviços geralmente exigem privilégios de superusuário para serem executados, os arquivos de configuração de serviço geralmente exigem privilégios de superusuário para editar.

A saída do comando mostra uma parte do arquivo de configuração do Nginx, que é um servidor web leve para Linux.

[analyst@secOps ~]$ **cat /etc/nginx/nginx.conf**

#user html;

worker\_processes 1;

#error\_log logs/error.log;

#error\_log logs/error.log notice;

#error\_log logs/error.log info;

#pid logs/nginx.pid;

events {

worker\_connections 1024;

}

http {

include mime.types;

default\_type application/octet-stream;

#log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '

# '$status $body\_bytes\_sent "$http\_referer" '

# '"$http\_user\_agent" "$http\_x\_forwarded\_for"';

#access\_log logs/access.log main;

A próxima saída do comando mostra o arquivo de configuração para o protocolo de tempo de rede (NTP).

[analyst@secOps ~]$ **cat /etc/ntp.conf**

# Please consider joining the pool:

#

# http://www.pool.ntp.org/join.html

#

# For additional information see:

# - https://wiki.archlinux.org/index.php/Network\_Time\_Protocol\_daemon

# - http://support.ntp.org/bin/view/Support/GettingStarted

# - the ntp.conf man page

# Associate to Arch's NTP pool

server 0.arch.pool.ntp.org

server 1.arch.pool.ntp.org

server 2.arch.pool.ntp.org

server 3.arch.pool.ntp.org

# By default, the server allows:

# - all queries from the local host

# - only time queries from remote hosts, protected by rate limiting and kod

restrict default kod limited nomodify nopeer noquery notrap

restrict 127.0.0.1

restrict ::1

# Location of drift file

[analyst@secOps ~]$

A saída do último comando mostra o arquivo de configuração do Snort, um sistema de detecção de intrusões (IDS) baseado em Linux.

[analyst@secOps ~]$ **cat /etc/snort/snort.conf**

#--------------------------------------------------

# VRT Rule Packages Snort.conf

#

# For more information visit us at:

# http://www.snort.org Snort Website

# http://vrt-blog.snort.org/ Sourcefire VRT Blog

#

# Mailing list Contact: snort-sigs@lists.sourceforge.net

# False Positive reports: fp@sourcefire.com

# Snort bugs: bugs@snort.org

#

# Compatible with Snort Versions:

# VERSIONS : 2.9.9.0

#

# Snort build options:

# OPTIONS : --enable-gre --enable-mpls --enable-targetbased --enable-ppm --enable-perfprofiling --enable-zlib --enable-active-response --enable-normalizer --enable-reload --enable-react --enable-flexresp3

<output omitted>

###################################################

# Step #1: Set the network variables. For more information, see README.variables

###################################################

# Setup the network addresses you are protecting

###ipvar HOME\_NET any

###ipvar HOME\_NET [192.168.0.0/24,192.168.1.0/24]

ipvar HOME\_NET [209.165.200.224/27]

# Set up the external network addresses. Leave as "any" in most situations

ipvar EXTERNAL\_NET any

Não há regra para um formato de arquivo de configuração; é a escolha do desenvolvedor do serviço. No entanto, o formato **option = value** é freqüentemente usado. Por exemplo, na última saída do comando, a variável **ipvar** é configurada com várias opções. A primeira opção, HOME\_NET, tem o valor 209.165.200.224/27. O caractere hash (**#**) é usado para indicar comentários.

4.4.2

## Fortalecimento de Dispositivos (hardering)

O fortalecimento (hardering) do dispositivo envolve a implementação de métodos comprovados de proteção do dispositivo e proteção de seu acesso administrativo. Alguns desses métodos envolvem a manutenção de senhas, a configuração de recursos avançados de login remoto e a implementação de login seguro com SSH. A definição de funções administrativas em termos de acesso é outro aspecto importante da proteção dos dispositivos de infraestrutura, pois nem todo o pessoal de tecnologia da informação deve ter o mesmo nível de acesso aos dispositivos de infraestrutura.

Dependendo da distribuição Linux, muitos serviços são habilitados por padrão. Alguns desses recursos estão habilitados por motivos históricos, mas não são mais necessários. Parar esses serviços e garantir que eles não iniciem automaticamente no momento da inicialização é outra técnica de fortalecimento (hardering) do dispositivo.

As atualizações do sistema operacional também são extremamente importantes para manter um dispositivo reforçado. Novas vulnerabilidades são descobertas todos os dias. Os desenvolvedores de SO criam e emitem correções e patches regularmente. Um computador atualizado é menos provável que seja comprometido.

A seguir estão as práticas recomendadas básicas para o fortalecimento (hardering) do dispositivo.

* Garantir a segurança física
* Minimizar pacotes instalados
* Desativar serviços não utilizados
* Usar SSH e desabilitar o login da conta raiz por SSH
* Manter o sistema atualizado
* Desativar a detecção automática de USB
* Aplicar senhas fortes
* Forçar mudanças de senha periódicas
* Manter os usuários de reutilizarem senhas antigas

Muitas outras etapas existem e muitas vezes dependem do serviço ou do aplicativo.

4.4.3

## Logs de serviço de monitoramento

Arquivos de log são os registros que um computador armazena para manter o controle de eventos importantes. Kernel, serviços e eventos de aplicativos são todos registrados em arquivos de log. É muito importante que um administrador revise periodicamente os logs de um computador para mantê-lo saudável. Ao monitorar arquivos de log do Linux, um administrador obtém uma visão clara do desempenho do computador, status de segurança e quaisquer problemas subjacentes. A análise do arquivo de log permite que um administrador proteja contra problemas futuros antes que eles ocorram.

No Linux, os arquivos de log podem ser categorizados como:

* Logs de aplicativos
* Logs de eventos
* Registros de serviço
* Logs do sistema

Alguns logs contêm informações sobre daemons que estão sendo executados no sistema Linux. Um daemon é um processo em segundo plano que é executado sem a necessidade de interação do usuário. Por exemplo, o System Security Services Daemon (SSSD) gerencia o acesso remoto e a autenticação para recursos de logon único.

A tabela lista alguns arquivos de log populares do Linux e suas funções

| **Arquivo de log do Linux** | **Descrição** |
| --- | --- |
| **/var/log/mensagens** | * Este diretório contém logs genéricos de atividade do computador. * Ele é usado principalmente para armazenar mensagens informativas e não críticas do sistema. * Em computadores baseados em Debian, o diretório /var/log/syslog serve a mesma finalidade. |
| **/var/log/auth.log** | * Este arquivo armazena todos os eventos relacionados à autenticação em computadores Debian e Ubuntu. * Qualquer coisa que envolva o mecanismo de autorização do usuário pode ser encontrada neste arquivo. |
| **/var/log/secure** | * Este diretório é usado por computadores RedHat e CentOS em vez de /var/log/auth.log. * Ele também rastreia logins sudo, logins SSH e outros erros registrados pelo SSSD. |
| **/var/log/boot.log** | * Este arquivo armazena informações relacionadas à inicialização e mensagens registradas durante o processo de inicialização do computador. |
| **/var/log/dmesg** | * Este diretório contém mensagens de buffer do anel do kernel. * Informações relacionadas a dispositivos de hardware e seus drivers são registradas aqui. * É muito importante porque, devido à sua natureza de baixo nível, sistemas de registro como syslog não estão sendo executados quando esses eventos ocorrem e, portanto, muitas vezes não estão disponíveis para o administrador em tempo real. |
| **/var/log/kern.log** | * Este arquivo contém informações registradas pelo kernel. |
| **/var/log/cron** | * Cron é um serviço usado para agendar tarefas automatizadas no Linux e este diretório armazena seus eventos. * Sempre que uma tarefa agendada (também chamada de trabalho cron) é executada, todas as informações relevantes, incluindo status de execução e mensagens de erro, são armazenadas aqui. |
| **/var/log/mysqld.log** ou **/var/log/mysql.log** | * Este é o arquivo de log do MySQL. * Todas as mensagens de depuração, falha e sucesso relacionadas ao processo mysqld e ao daemon mysqld\_safe são registradas aqui. * As distribuições RedHat, CentOS e Fedora Linux armazenam logs MySQL em /var/log/mysqld.log, enquanto Debian e Ubuntu mantêm o log no arquivo /var/log/mysql.log. |

A saída do comando mostra uma parte do arquivo de **/var/log/messages** log. Cada linha representa um evento registrado. Os carimbos de data/hora no início das linhas marcam o momento em que o evento ocorreu.

[analyst@secOps ~]$ **sudo cat /var/log/messages**

Mar 20 15:28:45 secOps kernel: Linux version 4.15.10-1-ARCH (builduser@heftig-18961) (gcc version 7.3.1 20180312 (GCC)) #1 SMP PREEMPT Thu Mar 15 12:24:34 UTC 2018

Mar 20 15:28:45 secOps kernel: Command line: BOOT\_IMAGE=/boot/vmlinuz-linux root=UUID=07c6b457-3f39-4ddf-bfd8-c169e8a877b2 rw quiet

Mar 20 15:28:45 secOps kernel: KERNEL supported cpus:

Mar 20 15:28:45 secOps kernel: Intel GenuineIntel

Mar 20 15:28:45 secOps kernel: AMD AuthenticAMD

Mar 20 15:28:45 secOps kernel: Centaur CentaurHauls

Mar 20 15:28:45 secOps kernel: x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'

Mar 20 15:28:45 secOps kernel: x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'

Mar 20 15:28:45 secOps kernel: x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'

Mar 20 15:28:45 secOps kernel: x86/fpu: xstate\_offset[2]: 576, xstate\_sizes[2]: 256

Mar 20 15:28:45 secOps kernel: x86/fpu: Enabled xstate features 0x7, context size is 832 bytes, using 'standard' format.

Mar 20 15:28:45 secOps kernel: e820: BIOS-provided physical RAM map:

Mar 20 15:28:45 secOps kernel: BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff] usable

Mar 20 15:28:45 secOps kernel: BIOS-e820: [mem 0x000000000009fc00-0x000000000009ffff] reserved

Mar 20 15:28:45 secOps kernel: BIOS-e820: [mem 0x00000000000f0000-0x00000000000fffff] reserved

Mar 20 15:28:45 secOps kernel: BIOS-e820: [mem 0x0000000000100000-0x000000003ffeffff] usable

Mar 20 15:28:45 secOps kernel: BIOS-e820: [mem 0x000000003fff0000-0x000000003fffffff] ACPI data

Mar 20 15:28:45 secOps kernel: BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fff] reserved

Mar 20 15:28:45 secOps kernel: BIOS-e820: [mem 0x00000000fee00000-0x00000000fee00fff] reserved

Mar 20 15:28:45 secOps kernel: BIOS-e820: [mem 0x00000000fffc0000-0x00000000ffffffff] reserved

Mar 20 15:28:45 secOps kernel: NX (Execute Disable) protection: active

Mar 20 15:28:45 secOps kernel: random: fast init done

Mar 20 15:28:45 secOps kernel: SMBIOS 2.5 present.

Mar 20 15:28:45 secOps kernel: DMI: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006

Mar 20 15:28:45 secOps kernel: Hypervisor detected: KVM

Mar 20 15:28:45 secOps kernel: e820: last\_pfn = 0x3fff0 max\_arch\_pfn = 0x400000000

Mar 20 15:28:45 secOps kernel: MTRR: Disabled

Mar 20 15:28:45 secOps kernel: x86/PAT: MTRRs disabled, skipping PAT initialization too.

Mar 20 15:28:45 secOps kernel: CPU MTRRs all blank - virtualized system.

4.4.4

## Laboratório — Localizando arquivos de log

Neste laboratório, você vai se familiarizar com a localização e manipulação de arquivos de log do Linux.

# O sistema de arquivos Linux

4.5.1

## Os tipos de sistema de arquivos no Linux

Existem muitos tipos diferentes de sistemas de arquivos, variando em propriedades de velocidade, flexibilidade, segurança, tamanho, estrutura, lógica e muito mais. Cabe ao administrador decidir qual tipo de sistema de arquivos melhor se adequa ao sistema operacional e aos arquivos que ele armazenará.

A tabela lista alguns tipos de sistema de arquivos comumente encontrados e suportados pelo Linux.

| **Sistema de arquivos Linux** | **Descrição** |
| --- | --- |
| **ext2 (segundo sistema de arquivos estendido)** | * ext2 era o sistema de arquivos padrão em várias distribuições Linux principais até ser suplantado pelo ext3. * Quase totalmente compatível com ext2, ext3 também suporta registro no diário (veja abaixo). * O ext2 ainda é o sistema de arquivos escolhido para mídia de armazenamento baseada em flash porque sua falta de diário aumenta o desempenho e minimiza o número de gravações. * Como os dispositivos de memória flash têm um número limitado de operações de gravação, a minimização das operações de gravação aumenta a vida útil do dispositivo. * No entanto, os kernels Linux contemporâneos também suportam ext4, um sistema de arquivos ainda mais moderno, com melhor desempenho e que também pode operar em um modo sem diário. |
| **ext3 (terceiro sistema de arquivos estendido)** | * ext3 é um sistema de arquivo registrado projetado para melhorar o sistema de arquivo ext2 existente. * Um diário, o principal recurso adicionado a ext3, é uma técnica usada para minimizar o risco de danos ao sistema de arquivos no caso de perda repentina de energia. * Os sistemas de arquivos mantém um registro (ou diário) de todas as alterações do sistema de arquivos prestes a ser feitas. * Se o computador falhar antes da conclusão da alteração, o diário pode ser usado para restaurar ou corrigir quaisquer eventuais problemas criados pela falha. * O tamanho máximo do arquivo em sistemas de arquivos ext3 é 32 TB. |
| **ext4 (quarto sistema de arquivos estendido)** | * Projetado como um sucessor de ext3, ext4 foi criado com base em uma série de extensões para ext3. * Enquanto as extensões melhoram o desempenho do ext3 e aumentam o tamanho dos arquivos suportados, os desenvolvedores do kernel Linux estavam preocupados com problemas de estabilidade e se opuseram a adicionar as extensões ao ext3 estável. * O projeto ext3 foi dividido em dois; um mantido como ext3 e seu desenvolvimento normal e o outro, denominado ext4, incorporou as extensões mencionadas. |
| **NFS (Network File System)** | * NFS é um sistema de arquivos baseado em rede, permitindo acesso a arquivos pela rede. * Do ponto de vista do usuário, não há diferença entre acessar um arquivo armazenado localmente ou em outro computador da rede. * O NFS é um padrão aberto, o que permite que qualquer pessoa o implemente. |
| **CDFS (Compact Disc File System)** | O CDFS foi criado especificamente para mídia de disco óptico. |
| **Sistema de troca de arquivos** | * O sistema de arquivos de troca é usado pelo Linux quando fica sem RAM. * Tecnicamente, é uma partição de swap que não tem um sistema de arquivos específico, mas é relevante para a discussão do sistema de arquivos. * Quando isso acontece, o kernel move o conteúdo inativo da RAM para a partição de troca no disco. * Embora as partições de permuta (também conhecidas como espaço de permuta) possam ser úteis para computadores Linux com uma quantidade limitada de memória, elas não devem ser consideradas como uma solução primária. * A partição de permuta é armazenada no disco que tem velocidades de acesso muito mais baixas do que a RAM. |
| **HFS Plus ou HFS+ (Sistema de Arquivos Hierárquico Plus)** | * Um sistema de arquivos usado pela Apple em seus computadores Macintosh. * O kernel Linux inclui um módulo para montar HFS+ para operações de leitura-gravação. |
| **APFS (Sistema de Arquivos Apple)** | Um sistema de arquivos atualizado que é usado por dispositivos Apple. Ele fornece criptografia forte e é otimizado para unidades flash e de estado sólido. |
| **Master Boot Record (MBR)** | * Localizado no primeiro setor de um computador particionado, o MBR armazena todas as informações sobre a forma como o sistema de arquivos é organizado. * O MBR entrega rapidamente o controle a uma função de carregamento, que carrega o sistema operacional. |

Montagem é o termo usado para o processo de atribuição de um diretório a uma partição. Após uma operação de montagem bem-sucedida, o sistema de arquivos contido na partição é acessível através do diretório especificado. Neste contexto, o diretório é chamado de ponto de montagem para esse sistema de arquivos. Os usuários do Windows podem estar familiarizados com um conceito semelhante; a letra da unidade.

A saída do comando mostra a saída do comando **mount** emitido na VM Cisco CyberOps.

[analyst@secOps ~]$ **mount**

proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)

sys on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)

dev on /dev type devtmpfs (rw,nosuid,relatime,size=494944k,nr\_inodes=123736,mode=755)

run on /run type tmpfs (rw,nosuid,nodev,relatime,mode=755)

**/dev/sda1 on / type ext4 (rw,relatime)**

securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)

tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)

devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)

tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755)

cgroup2 on /sys/fs/cgroup/unified type cgroup2 (rw,nosuid,nodev,noexec,relatime,nsdelegate)

cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr,name=systemd)

pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)

none on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)

cgroup on /sys/fs/cgroup/rdma type cgroup (rw,nosuid,nodev,noexec,relatime,rdma)

cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,cpu,cpuacct)

cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,blkio)

cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,hugetlb)

cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset)

cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)

cgroup on /sys/fs/cgroup/pids type cgroup (rw,nosuid,nodev,noexec,relatime,pids)

cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)

cgroup on /sys/fs/cgroup/net\_cls,net\_prio type cgroup (rw,nosuid,nodev,noexec,relatime,net\_cls,net\_prio)

cgroup on /sys/fs/cgroup/perf\_event type cgroup (rw,nosuid,nodev,noexec,relatime,perf\_event)

cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,freezer)

systemd-1 on /proc/sys/fs/binfmt\_misc type autofs (rw,relatime,fd=29,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe\_ino=11792)

debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime)

tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime)

hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,pagesize=2M)

mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime)

tmpfs on /tmp type tmpfs (rw,nosuid,nodev)

Quando emitido sem opções, **mount** retorna a lista de sistemas de arquivos atualmente montados em um computador Linux. Embora muitos dos sistemas de arquivos mostrados estejam fora do escopo deste curso, observe o sistema de arquivos raiz (destacado). O sistema de arquivos raiz é representado pelo símbolo “/” e contém todos os arquivos no computador por padrão. Também é mostrado na saída que o sistema de arquivos raiz foi formatado como ext4 e ocupa a primeira partição da primeira unidade (/dev/sda1).

4.5.2

## Funções do Linux e permissões de arquivo

No Linux, a maioria das entidades do sistema são tratadas como arquivos. Para organizar o sistema e impor limites dentro do computador, o Linux usa permissões de arquivo. As permissões de arquivo são incorporadas à estrutura do sistema de arquivos e fornecem um mecanismo para definir permissões em cada arquivo. Cada arquivo no Linux carrega suas permissões de arquivo, que definem as ações que o proprietário, o grupo e outros podem executar com o arquivo. Os direitos de permissão possíveis são Ler, Gravar e Executar. O comando **ls** com o **-l** parâmetro lista informações adicionais sobre o arquivo.

Considere a saída do comando **ls -l** na saída do comando.

[analyst@secOps ~]$ **ls -l space.txt**

-rwxrw-r-- 1 analyst staff 253 May 20 12:49 space.txt

(1)(2)(3)(4)(5)(6)(7)

[analyst@secOps ~]$

A saída fornece muitas informações sobre o arquivo space.txt.

O primeiro campo da saída exibe as permissões associadas a **space.txt** (**-rwxrw-r--**). As permissões de arquivo são sempre exibidas na ordem Usuário, Grupo e Outro.

O arquivo **space.txt** tem as seguintes permissões:

* O traço (-) significa que este é um arquivo. Para diretórios, o primeiro traço seria um “d”.
* O primeiro conjunto de caracteres é para permissão do usuário (O). O usuário, **analyst** do **rwx**, que possui o arquivo pode Ler, Gravar e eXECute o arquivo.
* O segundo conjunto de caracteres é para permissões de grupo (**rw-**). O grupo, **staff**, que possui o arquivo pode Ler e WRite para o arquivo.
* O terceiro conjunto de caracteres é para qualquer outra permissão de usuário ou grupo (\*\*). Qualquer outro usuário ou grupo no computador só pode ler o arquivo.

O segundo campo define o número de links rígidos para o arquivo (o número **1** após as permissões). Um link rígido cria outro arquivo com um nome diferente vinculado ao mesmo lugar no sistema de arquivos (chamado de inode). Isso está em contraste com um link simbólico, que é discutido na próxima página.

O terceiro e quarto campos exibem o usuário (**analyst**) e o grupo (**staff**) que possuem o arquivo, respectivamente.

O quinto campo exibe o tamanho do arquivo em bytes. O arquivo **space.txt** tem 253 bytes.

O sexto campo exibe a data e hora da última modificação.

O sétimo campo exibe o nome do arquivo.

A figura mostra uma divisão das permissões de arquivo no Linux.

A imagem mostra o comando: ls -l space.txt. A única linha de saída é ፦rwxrw-r— 1 analista equipe 253 Maio 20 12:49 space.txt. A segunda imagem destaca que a saída de permissão de arquivo consiste no primeiro caractere em que um - indica um arquivo ou a letra d que indica um diretório, as próximas três letras rwx indicam a permissão do usuário, as três letras seguintes rwx indicam a permissão do grupo e as três letras seguintes rwx indicar a permissão de outros usuários. A tabela abaixo da imagem tem 4 colunas: binário, octal, permissão e descrição. O binário 000 é 0 em octal, a permissão é listada como — o que significa que não há acesso. O binário 001 é 1 em octal, a permissão é listada como -x, o que significa executar apenas. O binário 010 é 2 em octal, a permissão é listada como -w- o que significa apenas escrever. O binário 011 é 3 em octal, a permissão é listada como -wx que significa escrever e executar. O binário 100 é 4 em octal, a permissão é listada foi râ€” o que significa somente leitura. O binário 101 é 5 em octal, a permissão é listada como r-x, o que significa ler e executar. O binário 110 é 6 em octal, a permissão é listada como rw- o que significa ler e escrever. O binário 111 é 7 em octal, a permissão é listada como rwx que significa ler, escrever e executar.

- or d r r r w x w w x

x

UsuárioGrupoOutrosBit de arquivo ou diretório: -  
indica que é um arquivo   
indica que é um diretório

Use valores octais para definir permissões.

| **Binário** | **Octal** | **Permissão** | **Descrição** |
| --- | --- | --- | --- |
| 000 | 0 | --- | Sem acesso |
| 001 | 1 | --x | Executar apenas |
| 010 | 2 | -w- | Somente escrita |
| 011 | 3 | -wx | Edição e execução |
| 100 | 4 | r-- | Somente leitura |
| 101 | 5 | r-x | Ler e Executar |
| 110 | 6 | rw- | Leitura e Escrita |
| 111 | 7 | rwx | Ler, escrever e executar |

Permissões de arquivo são uma parte fundamental do Linux e não podem ser quebradas. Um usuário tem apenas os direitos de um arquivo que as permissões de arquivo permitem. O único usuário que pode substituir a permissão de arquivo em um computador Linux é o usuário root. Como o usuário root tem o poder de substituir as permissões de arquivo, o usuário root pode gravar em qualquer arquivo. Como tudo é tratado como um arquivo, o usuário root tem controle total sobre um computador Linux. O acesso como root é muitas vezes necessário antes de realizar tarefas administrativas e de manutenção. Devido ao poder do usuário raiz, as credenciais raiz devem usar senhas fortes e não ser compartilhadas com ninguém além de administradores de sistema e outros usuários de alto nível.

4.5.3

## Links rígidos e links simbólicos

Um link rígido é outro arquivo que aponta para o mesmo local que o arquivo original. Use o comando **ln** para criar um link rígido. O primeiro argumento é o arquivo existente e o segundo argumento é o novo arquivo. Como mostrado na saída do comando, o arquivo **space.txt** é vinculado **space.hard.txt** e o campo de link agora mostra 2.

[analyst@secOps ~]$ **ln space.txt space.hard.txt**

[analyst@secOps ~]$

[analyst@secOps ~]$ **ls -l space\***

-rw-r--r-- 2 analyst analyst 239 May 7 18:18 space.hard.txt

-rw-r--r-- 2 analyst analyst 239 May 7 18:18 space.txt

[analyst@secOps ~]$

[analyst@secOps ~]$ **echo "Testing hard link" >> space.txt**

[analyst@secOps ~]$

[analyst@secOps ~]$ **ls -l space\***

-rw-r--r-- 2 analyst analyst 257 May 7 18:19 space.hard.txt

-rw-r--r-- 2 analyst analyst 257 May 7 18:19 space.txt

[analyst@secOps ~]$

[analyst@secOps ~]$ **rm space.hard.txt**

[analyst@secOps ~]$

[analyst@secOps ~]$ **more space.txt**

Space... The final frontier…

These are the voyages of the Starship Enterprise. Its continuing mission:

- To explore strange new worlds…

- To seek out new life; new civilizations…

- To boldly go where no one has gone before!

Testing hard link

[analyst@secOps ~]$

Ambos os arquivos apontam para o mesmo local no sistema de arquivos. Se você alterar um arquivo, o outro também será alterado. O comando **echo** é usado para adicionar algum texto ao **space.txt**. Observe que o tamanho do arquivo para ambos **space.txt** e **space.hard.txt** aumentou para 257 bytes. Se você excluir o space.hard.txt com o comando (remover) **rm**, o arquivo **space.txt** ainda existe, conforme verificado com o comando **more space.txt**.

Um link simbólico, também chamado de link simbólico ou link suave, é semelhante a um link rígido em que a aplicação de alterações ao link simbólico também mudará o arquivo original. Como mostrado na saída do comando abaixo, use o comando **ln** com a opção **-s** para criar um link simbólico.

[analyst@secOps ~]$ **echo "Hello World!" > test.txt**

[analyst@secOps ~]$

[analyst@secOps ~]$ **ln -s test.txt mytest.txt**

[analyst@secOps ~]$

[analyst@secOps ~]$ **echo "It's a lovely day!" >> mytest.txt**

[analyst@secOps ~]$

[analyst@secOps ~]$ **more test.txt**

Hello World!

It's a lovely day!

[analyst@secOps ~]$

[analyst@secOps ~]$ **rm test.txt**

[analyst@secOps ~]$

[analyst@secOps ~]$ **more mytest.txt**

more: stat of mytest.txt failed: No such file or directory

[analyst@secOps ~]$

[analyst@secOps ~]$ **ls -l mytest.txt**

lrwxrwxrwx 1 analyst analyst 8 May 7 20:17 mytest.txt -> test.txt

[analyst@secOps ~]$

Observe que a adição de uma linha de texto **test.txt** também adiciona a linha **mytest.txt**. No entanto, ao contrário de um link rígido, excluir o arquivo **text.txt** original significa que agora **mytext.txt** está vinculado a um arquivo que não existe mais, como mostrado com os comandos **more mytest.txt** e **ls -l mytest.txt** .

Embora links simbólicos tenham um único ponto de falha (o arquivo subjacente), links simbólicos têm vários benefícios sobre links rígidos:

* Localizar links físicos é mais difícil. Links simbólicos mostram a localização do arquivo original no comando **ls -l**, conforme mostrado na última linha de saída no comando anterior. (**mytest.txt -> test.txt**).
* Links rígidos são limitados ao sistema de arquivos no qual eles são criados. Links simbólicos podem ser vinculados a um arquivo em outro sistema de arquivos.
* Links rígidos não podem se vincular a um diretório porque o próprio sistema usa links rígidos para definir a hierarquia da estrutura de diretórios. No entanto, links simbólicos podem se vincular a diretórios.

4.5.4

## Laboratório - Navegando no Sistema de Arquivos Linux e Configurações de Permissão

Neste laboratório, você se familiarizará com sistemas de arquivos Linux.

# Trabalhando com a GUI Linux

4.6.1

## Sistema X Window

A interface gráfica presente na maioria dos computadores Linux é baseada no X Window System. Também conhecido como X ou X11, X Window é um sistema de janelas projetado para fornecer a estrutura básica para uma GUI. X inclui funções para desenhar e mover janelas no dispositivo de exibição e interagir com um mouse e teclado.

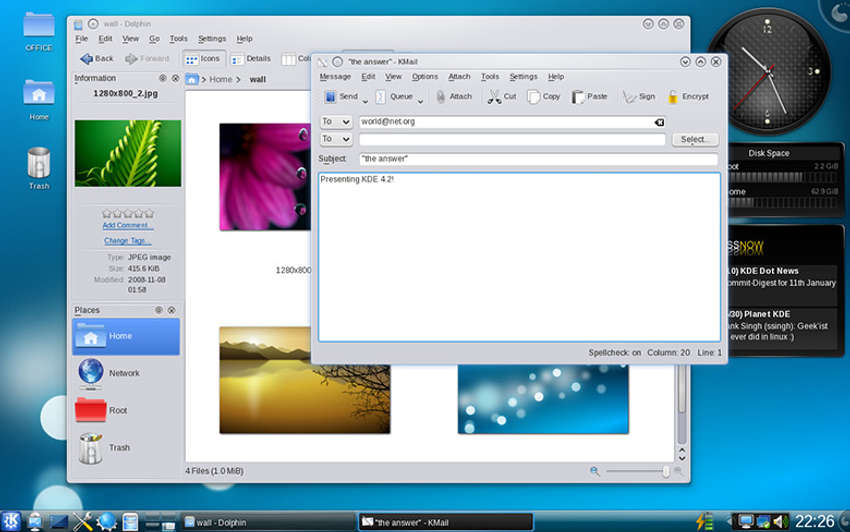
X funciona como um servidor que permite que um usuário remoto use a rede para se conectar, iniciar um aplicativo gráfico e ter a janela gráfica aberta no terminal remoto. Enquanto o aplicativo em si é executado no servidor, o aspecto gráfico dele é enviado por X pela rede e exibido no computador remoto.

Observe que X não especifica a interface do usuário, deixando para outros programas, como gerenciadores de janelas, definir todos os componentes gráficos. Essa abstração permite grande flexibilidade e personalização, pois componentes gráficos, como botões, fontes, ícones, bordas de janelas e esquemas de cores são definidos pelo aplicativo do usuário. Devido a essa separação, a GUI do Linux varia muito de distribuição para distribuição. Exemplos de gerenciadores de janelas são o Gnome e o KDE. Embora a aparência dos gerenciadores de janelas variem, os componentes principais ainda estão presentes.

A figura exibe o Gerenciador de Janelas do Gnome.



Esta figura mostra o Gerenciador de Windows do KDE.



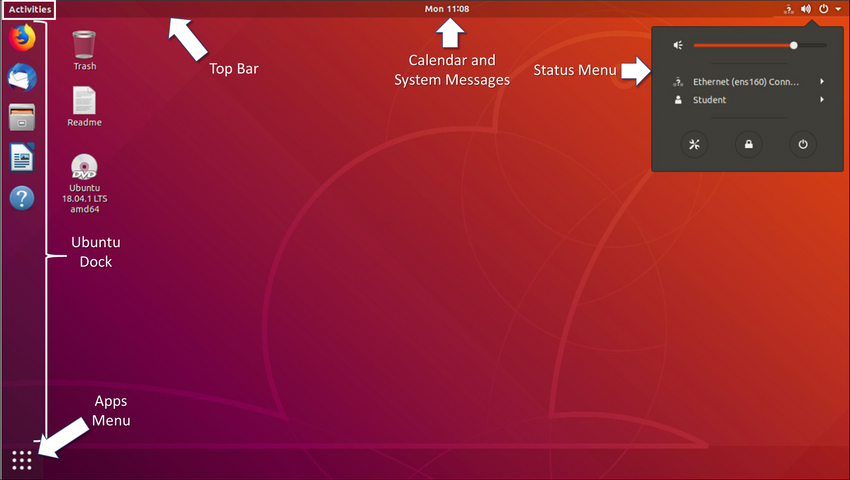
4.6.2

## A GUI do Linux

Embora um sistema operacional não exija uma GUI para funcionar, as GUIs são consideradas mais fáceis de usar do que a CLI. A GUI Linux como um todo pode ser facilmente substituída pelo usuário. Como resultado do grande número de distribuições Linux, este capítulo se concentra no Ubuntu ao cobrir o Linux, porque é uma distribuição muito popular e amigável.

Ubuntu Linux usa Gnome 3 como sua GUI padrão. O objetivo do Gnome 3 é tornar o Ubuntu ainda mais amigável. A tabela lista os principais componentes da interface do usuário do Unity.

A figura mostra a localização de alguns dos recursos do Ubuntu Gnome 3 Desktop.



| **Componente UI** | **Descrição** |
| --- | --- |
| **Menu Apps** | * O Menu Aplicativos mostra ícones para os aplicativos que estão instalados no sistema. * Um menu com o botão direito do mouse fornece atalhos que permitem iniciar ou configurar os aplicativos. * A caixa de pesquisa do sistema está disponível na Exibição de Atividades. |
| **Dock do Ubuntu** | * Este é um dock no lado esquerdo da tela que serve como um inicializador de aplicativos e alternador para favoritos de aplicativos. * Clique para iniciar um aplicativo e, quando o aplicativo estiver em execução, clique novamente para alternar entre aplicativos em execução. * Se mais de uma instância de um aplicativo estiver em execução, o Launcher exibirá todas as instâncias. * Clique com o botão direito do mouse em qualquer aplicativo hospedado no iniciador para ver detalhes sobre isso o aplicativo. |
| **Barra superior** | * Esta barra de menu multiuso contém um menu para o aplicativo que atualmente tem o foco. * Ele exibe a hora atual e indica se há novas mensagens do sistema. * Ele também fornece acesso à visualização da área de trabalho Atividade e ao Menu Status do sistema. |
| **Calendário e bandeja de mensagens do sistema** | * Clique no dia e hora para ver o calendário de compromissos completo e quaisquer mensagens atuais do sistema. * Acesse o calendário de compromissos daqui para criar novos compromissos. |
| **Atividades** | * Alterne para a visualização do aplicativo para alternar ou fechar os aplicativos em execução. * Uma ferramenta de pesquisa poderosa está disponível aqui para localizar aplicativos, arquivos e valores dentro dos arquivos. * Permite alternar entre espaços de trabalho. |
| **Menu de Status** | * Permite a configuração do adaptador de rede e outros dispositivos em execução. * O usuário atual pode fazer logoff ou alterar suas configurações. * Mudanças na configuração do sistema podem ser feitas aqui. * A estação de trabalho pode ser bloqueada ou desligada a partir daqui. |

# Trabalhando em um host Linux

4.7.1

## Instalação e execução de aplicativos em um host Linux

Muitos aplicativos de usuário final são programas complexos escritos em linguagens compiladas. Para ajudar no processo de instalação, o Linux geralmente inclui programas chamados gerenciadores de pacotes. Um pacote é o termo usado para se referir a um programa e todos os seus arquivos de suporte. Ao usar um gerenciador de pacotes para instalar um pacote, todos os arquivos necessários são colocados no local correto do sistema de arquivos.

Os gerenciadores de pacotes variam dependendo das distribuições do Linux. Por exemplo, **pacman** é usado pelo Arch Linux enquanto **dpkg** (pacote Debian) e **apt** (Advanced Packaging Tool) são usados em distribuições Debian e Ubuntu Linux. Para este curso, usaremos o gerenciador de **pacman** pacotes.

A saída do comando mostra a saída de alguns comandos **apt-get** usados nas distribuições Debian.

analyst@cuckoo:~$ **sudo apt-get update**

[sudo] password for analyst:

Hit:l http://us.archive.ubuntu.com/ubuntu xenial InRelease

Get:2 http://us.archive.ubuntu.com/ubuntu xenial-updates InRelease [102 kB]

Get:3 http://security.ubuntu.com/ubuntu xenial-security InRelease [102 kB]

Get:4 http://us.archive.ubuntu.com/ubuntu xenial-backports InRelease [102 kB]

Get:5 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 Packages [534 kB]

<output omitted>

Fetched 4,613 kB in 4s (1,003 kB/s)

Reading package lists... Done

analyst@cuckoo:~$

analyst@cuckoo:~$ **sudo apt-get upgrade**

Reading package lists Done

Building dependency tree

Reading state information... Done

Calculating upgrade... Done

The following packages have been kept back:

linux-generic-hwe-16.04 linux-headers-generic-hwe-16.04

linux-image-generic-hwe-16.04

The following packages will be upgraded:

firefox firefox-locale-en girl.2-javascriptcoregtk-4.0 girl.2-webkit2-4.0 libjavascriptcoregtk-4.0-18

libwebkit2gtk-4.0-37 libwebkit2gtk-4.0-37-gtk2 libxen-4.6 libxenstore3.0 linux-libc-dev logrotate openssh-client

qemu-block-extra qerau-kvm qemu-system-common qemu-system-x86 qemu-utils

O comando **apt-get update** é usado para obter a lista de pacotes do repositório de pacotes e atualizar o banco de dados de pacotes local. O comando **apt-get upgrade** é usado para atualizar todos os pacotes atualmente instalados para suas versões mais recentes.

4.7.2

## Mantendo o sistema atualizado

Também conhecido como patches, as atualizações do sistema operacional são lançadas periodicamente por empresas de sistema operacional para resolver quaisquer vulnerabilidades conhecidas em seus sistemas operacionais. Embora as empresas tenham programações de atualização, o lançamento de atualizações não programadas do sistema operacional pode acontecer quando uma vulnerabilidade importante é encontrada no código do sistema operacional. Os sistemas operacionais modernos alertarão o usuário quando atualizações estiverem disponíveis para download e instalação, mas o usuário pode verificar as atualizações a qualquer momento.

A tabela a seguir compara os comandos de distribuição Arch Linux e Debian / Ubuntu Linux para realizar operações básicas do sistema de pacotes.

| **Tarefa** | **Arch** | **Debian/Ubuntu** |
| --- | --- | --- |
| Instalar um pacote pelo nome | **pacman -S** | **apt install** |
| Remover um pacote pelo nome | **pacman -Rs** | **apt remover** |
| Update a local package | **pacman -Syy** | **apt-get update** |
| Atualize todos os pacotes atualmente instalados | **pacman -Syu** | **atualização do apt-get** |

Uma GUI do Linux também pode ser usada para verificar e instalar manualmente as atualizações. No Ubuntu, por exemplo, para instalar atualizações, clique em **Dash Search Box** ,**software updater** digite e clique no **ícone Atualizador de** Software, conforme mostrado na figura.

The image shows the output of the apt-get command.  Command: analyst@cuckoo:~$ sudo apt-get update  output: [sudo] password for analyst:  Hit:1 http://us.archive.ubuntu.com/ubuntu xenial InRelease Get:2 http://us.archive.ubuntu.com/ubuntu xenial-updates InRelease [102 kB], Get:3 http://security.ubuntu.com/ubuntu xenial-security InRelease [102 kB]Get:4 http://us.archive.ubuntu.com/ubuntu xenial-backports InRelease [102 kB] Get:5 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 Packages [534 kB] <output omitted> Fetched 4,613 kB in 4s (1,003 kB/s)Reading package lists... Done analyst@cuckoo:~$ sudo apt-get upgrade Reading package lists  Done Building dependency tree Reading state information... Done Calculating upgrade... Done The following packages have been kept back:
linux-generic-hwe-16.04 linux-headers-generic-hwe-16.04 linux-image-generic-hwe-16.04 The following packages will be upgraded: firefox firefox-locale-en girl.2-javascriptcoregtk-4.0 girl.2-webkit2-4.0 libjavascriptcoregtk-4.0-18 libwebkit2gtk-4.0-37 libwebkit2gtk-4.0-37-gtk2 libxen-4.6 libxenstore3.0 linux-libc-dev logrotate openssh-client qemu-block-extra qerau-kvm qemu-system-common qemu-system-x86 qemu-utils

4.7.3

## Processos e Forks

Um processo é uma instância em execução de um programa de computador. Os sistemas operacionais multitarefa podem executar muitos processos ao mesmo tempo.

Bifurcação é um método que o kernel usa para permitir que um processo crie uma cópia de si mesmo. Os processos precisam de uma maneira de criar novos processos em sistemas operacionais multitarefa. A operação bifurcação (fork) é a única maneira de fazer isso no Linux.

A bifurcação (fork) é importante por muitos motivos. Um deles está relacionado à escalabilidade do processo. Apache, um servidor web popular, é um bom exemplo. Ao se bifurcar, o Apache é capaz de atender a um grande número de solicitações com menos recursos do sistema do que um servidor baseado em processo único.

Quando um processo é bifurcado (fork), o processo do chamador se torna o processo pai, com o processo recém-criado referido como seu filho. Depois da bifurcação, os processos são, até certo ponto, processos independentes; eles têm IDs de processo diferentes, mas executam o mesmo código de programa.

A tabela lista três comandos usados para gerenciar processos.

| **Comando** | **Descrição** |
| --- | --- |
| **ps** | * Usado para listar os processos em execução no computador no momento em que é invocado. * Ele pode ser instruído a exibir processos em execução que pertencem ao usuário atual ou a outros usuários. * Embora listar processos não exija privilégios de root, eliminar ou modificar os processos de outros usuários exige. |
| **top** | * Usado para listar processos em execução, mas ao contrário do **ps**, **top** continua exibindo processos em execução dinamicamente. * Pressione **q** para sair do topo. |
| **kill** | * Usado para modificar o comportamento de um processo específico. * Dependendo dos parâmetros, **kill** removerá, reiniciará ou pausará um processo. * Em muitos casos, o usuário executará **ps** ou **top** antes de executar kill. * Isso é feito para que o usuário possa aprender o PID de um processo antes de executar kill. |

A saída do comando mostra a saída do comando **top** em um computador Linux.

[analyst@secOps ~]$ **top**

top - 11:29:16 up 0 min, 1 user, load average: 1.09, 0.31, 0.11

Tasks: 119 total, 1 running, 118 sleeping, 0 stopped, 0 zombie

%Cpu(s): 5.4 us, 2.0 sy, 0.0 ni, 87.4 id, 2.7 wa, 1.4 hi, 1.0 si, 0.0 st

MiB Mem : 982.8 total, 67.9 free, 765.8 used, 149.1 buff/cache

MiB Swap: 0.0 total, 0.0 free, 0.0 used. 39.3 avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND

729 analyst 20 0 2652376 284472 61076 S 2.7 28.3 0:06.75 Web Con+

570 analyst 20 0 2691388 215728 62404 S 2.0 21.4 0:06.99 firefox

357 root 20 0 267972 91960 18468 S 1.3 9.1 0:01.63 Xorg

461 analyst 20 0 322208 21000 7480 S 1.3 2.1 0:00.67 xfce4-p+

121 root 20 0 0 0 0 S 0.7 0.0 0:00.43 kswapd0

1 root 20 0 174376 4196 1688 S 0.3 0.4 0:00.66 systemd

294 root 20 0 245036 11876 868 S 0.3 1.2 0:00.34 python2+

539 analyst 20 0 150824 660 0 S 0.3 0.1 0:00.02 VBoxCli+

800 analyst 20 0 477768 18968 9800 S 0.3 1.9 0:00.30 xfce4-t+

2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthreadd

3 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu\_gp

4 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu\_par+

5 root 20 0 0 0 0 I 0.0 0.0 0:00.00 kworker+

6 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker+

7 root 20 0 0 0 0 I 0.0 0.0 0:00.00 kworker+

8 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 mm\_perc+

9 root 20 0 0 0 0 S 0.0 0.0 0:00.02 ksoftir+

[analyst@secOps ~]$

4.7.4

## Malware em um host Linux

O malware do Linux inclui vírus, cavalos de Tróia, worms e outros tipos de malware que podem afetar o sistema operacional. Devido a uma série de componentes de design, tais como estrutura do sistema de arquivos, permissões de arquivo e restrições de conta de usuário, sistemas operacionais Linux são geralmente considerados como melhor protegidos contra malware.

Embora indiscutivelmente mais protegido, o Linux não é imune a malware. Muitas vulnerabilidades foram encontradas e exploradas no Linux. Estes variam de software de servidor a vulnerabilidades de kernel. Os atacantes são capazes de explorar essas vulnerabilidades e comprometer o alvo. Como o Linux é de código aberto, correções e patches são frequentemente disponibilizados em poucas horas após a descoberta de tais problemas.

Se um programa malicioso for executado, ele causará danos, independentemente da plataforma. Um vetor de ataque Linux comum é seus serviços e processos. Vulnerabilidades são freqüentemente encontradas no código de servidor e processo em execução em computadores conectados à rede. Uma versão desatualizada do servidor web Apache poderia conter uma vulnerabilidade não corrigida que pode ser explorada por um invasor, por exemplo. Os invasores geralmente testam portas abertas para avaliar a versão e a natureza do servidor em execução nessa porta. Com esse conhecimento, os atacantes podem pesquisar se há algum problema conhecido com essa versão específica desse servidor específico para dar suporte ao ataque. Como acontece com a maioria das vulnerabilidades, manter o computador atualizado e fechar todos os serviços e portas não utilizados é uma boa maneira de reduzir as oportunidades de ataque em um computador Linux.

A saída do comando mostra um invasor usando o comando Telnet para testar a natureza e a versão de um servidor Web (porta 80).

analyst@secOps ~]$ **telnet 209.165.200.224 80**

Trying 209.165.200.224...

Connected to 209.165.200.224.

Escape character is ‘^]’.

<type anything to force an HTTP error response>

HTTP/1.1 400 Bad Request

Server: nginx/1.12.0

Date: Wed, 17 May 2017 14:27:30 GMT

Content-Type: text/html

Content-Length: 173

Connection: close

<html>

<head><title>400 Bad Request</title></head>

<body bgcolor="white">

<center><h1>400 Bad Request</h1></center>

<hr><center>nginx/1.12.0</center>

</body>

</html >

Connection closed by foreign host.

analyst@secOps ~]$

O invasor descobriu que o servidor em questão está executando o nginx versão 1.12.0. O próximo passo seria pesquisar vulnerabilidades conhecidas no código nginx 1.12.0.

**Nota**: Você aprenderá mais sobre esse ataque mais tarde no curso.

4.7.5

## Verificação de Rootkit

Um rootkit é um tipo de malware projetado para aumentar os privilégios de um usuário não autorizado ou conceder acesso a partes do software que normalmente não devem ser permitidas. Rootkits também são frequentemente usados para proteger uma porta traseira para um computador comprometido.

A instalação de um rootkit pode ser automatizada (feita como parte de uma infecção) ou um invasor pode instalá-lo manualmente após comprometer um computador. Um rootkit é destrutivo porque ele muda o código do kernel e seus módulos, alterando as operações mais fundamentais do próprio sistema operacional. Com um nível tão profundo de comprometimento, os rootkits podem ocultar a intrusão, remover quaisquer faixas de instalação e até mesmo adulterar ferramentas de diagnóstico e solução de problemas para que sua saída agora esconda a presença do rootkit. Embora algumas vulnerabilidades do Linux ao longo do histórico tenham permitido a instalação do rootkit através de contas de usuário regulares, a grande maioria dos compromissos do rootkit requer acesso root ou administrador.

Como a própria natureza do computador está comprometida, a detecção de rootkit pode ser muito difícil. Os métodos de detecção típicos geralmente incluem a inicialização do computador a partir de mídia confiável, como um CD ativo do sistema operacional de diagnóstico. A unidade comprometida é montada e, a partir do conjunto de ferramentas do sistema confiável, ferramentas de diagnóstico confiáveis podem ser iniciadas para inspecionar o sistema de arquivos comprometido. Os métodos de inspeção incluem métodos baseados em comportamento, varredura de assinatura, varredura de diferenças e análise de despejo de memória.

A remoção do Rootkit pode ser complicada e muitas vezes impossível, especialmente nos casos em que o rootkit reside no kernel; a reinstalação do sistema operacional geralmente é a única solução real para o problema. Os rootkits de firmware geralmente requerem substituição de hardware.

**chkrootkit** é um programa popular baseado em Linux projetado para verificar o computador em busca de rootkits conhecidos. É um script shell que usa ferramentas comuns do Linux, como **strings** e **grep** para comparar as assinaturas de programas principais. Ele também procura discrepâncias à medida que atravessa o sistema de arquivos /proc comparando as assinaturas encontradas lá com a saída de **ps**.

Embora útil, tenha em mente que os programas para verificar se há rootkits não são 100% confiáveis.

A saída do comando mostra a saída de **chkrootkit** em um Ubuntu Linux.

analyst@cuckoo:~$ **sudo ./chkrootkit**

[sudo] password for analyst:

ROOTDIR is ‘/’

Checking &apos;amd’... not found

Checking &apos;basename’... not infected

Checking &apos;biff’... not found

Checking &apos;chfn’... not infected

Checking &apos;chsh&apos;... not infected

Checking &apos;cron&apos;... not infected

Checking &apos;crontab’... not infected

Checking &apos;date&apos;... not infected

Checking &apos;du&apos;... not infected

Checking &apos;dirname&apos;... not infected

Checking &apos;echo&apos;... not infected

Checking &apos;egrep’... not infected

Checking &apos;env&apos;— not infected

Checking &apos;find&apos;... not infected

Checking &apos;fingerd&apos;... not found

Checking &apos;gpm&apos;... not found

Checking &apos;grep&apos;... not infected

Checking &apos;hdparm’... not infected

Checking &apos;su&apos;... not infected

Checking &apos;ifconfig&apos;... not infected

Checking &apos;inetd’... not tested

Checking &apos;inetdconf’... not found

4.7.6

## Comandos de piping

Embora as ferramentas de linha de comando geralmente sejam projetadas para executar uma tarefa específica e bem definida, muitos comandos podem ser combinados para executar tarefas mais complexas por uma técnica conhecida como piping. Nomeado após seu caractere definidor, o pipe (**|**), tubulação consiste em encadear comandos juntos, alimentando a saída de um comando na entrada de outro.

Por exemplo, o comando **ls** é usado para exibir todos os arquivos e diretórios de um determinado diretório. O comando **grep** compara pesquisas através de um arquivo ou texto procurando a string especificada. Se encontrado, **grep** exibe todo o conteúdo da pasta onde a string foi encontrada.

Os dois comandos, **ls** e **grep**, podem ser canalizado juntos para filtrar a saída de **ls**. Isto é mostrado na saída do comando **ls -l | grep host** e do comando **ls -l | grep file** .

[analyst@secOps ~]$ **ls -l**

total 40

drwxr-xr-x 2 analyst analyst 4096 Mar 22 2018 Desktop

drwxr-xr-x 3 analyst analyst 4096 April 2 14:44 Downloads

-rw-r--r-- 1 analyst analyst 9 May 20 10:51 hostfile1.txt

-rw-r--r-- 1 analyst analyst 9 May 20 10:51 hostfile2.txt

-rw-r--r-- 1 analyst analyst 9 May 20 10:52 hostfile3.txt

drwxr-xr-x 9 analyst analyst 4096 Jul 19 2018 lab.support.files

-rw-r--r-- 1 analyst analyst 19 May 20 10:53 mytest.com

-rw-r--r-- 1 analyst analyst 228844 May 20 10:54 rkhunter-1.4.6-1-any.pkg.tar.xz

drwxr-xr-x 2 analyst analyst 4096 Mar 21 2018 second\_drive

-rw-r--r-- 1 analyst analyst 257 May 20 10:52 space.txt

[analyst@secOps ~]$

[analyst@secOps ~]$ **ls -l | grep host**

-rw-r--r-- 1 analyst analyst 9 May 20 10:51 hostfile1.txt

-rw-r--r-- 1 analyst analyst 9 May 20 10:51 hostfile2.txt

-rw-r--r-- 1 analyst analyst 9 May 20 10:52 hostfile3.txt

[analyst@secOps ~]$

[analyst@secOps ~]$ **ls -l | grep file**

-rw-r--r-- 1 analyst analyst 9 May 20 10:51 hostfile1.txt

-rw-r--r-- 1 analyst analyst 9 May 20 10:51 hostfile2.txt

-rw-r--r-- 1 analyst analyst 9 May 20 10:52 hostfile3.txt

drwxr-xr-x 9 analyst analyst 4096 Jul 19 2018 lab.support.files

[analyst@secOps ~]$

4.7.7

## Vídeo - Aplicações, Rootkits e comandos de tubulação

Reproduza o vídeo para ver uma demonstração de instalação e atualização de aplicativos, verificação de um rootkit e uso de comandos de piping.

# Resumo básico do Linux

4.8.1

## O que aprendi neste módulo?

**Linux Básico**

O Linux é um sistema operacional de código aberto rápido, confiável e pequeno. Ele requer poucos recursos de hardware para ser executado e é altamente personalizável. Ele é projetado para ser usado em redes. O kernel Linux é distribuído por diferentes organizações com diferentes ferramentas e pacotes de software. Uma versão personalizada do Linux chamada Security Onion contém software e ferramentas que são projetados para uso em monitoramento de segurança de rede por analistas de segurança cibernética. Kali Linux é outra distribuição Linux personalizada que tem inúmeras ferramentas que são projetadas para testes de penetração de segurança de rede.

**Trabalhando no Linux Shell**

No Linux, o usuário se comunica com o sistema operacional por meio de uma GUI ou uma interface de linha de comando (CLI) ou shell. Se uma GUI estiver em execução, o shell é acessado através do aplicativo terminal, como xterm ou gnome terminal. Comandos Linux são programas que executam uma tarefa específica. O comando **man** , seguido de um comando específico, fornece documentação para esse comando. É importante conhecer pelo menos comandos básicos do Linux, comandos de arquivo e diretório e comandos para trabalhar com arquivos de texto. No Linux tudo é tratado se fosse um arquivo, incluindo a memória, discos, monitor e diretórios.

**Servidores e clientes Linux**

Servidores são computadores que possuem software instalado que lhes permite fornecer serviços a computadores cliente em toda a rede. Alguns serviços fornecem acesso a recursos externos, como arquivos, e-mail e páginas da Web, para clientes mediante solicitação. Outros serviços são executados internamente e executam tarefas como gerenciamento de logs, gerenciamento de memória ou varredura de disco. Para permitir que um computador forneça vários serviços, são utilizadas portas. Uma porta é um recurso de rede reservado que “escuta” solicitações de clientes. Embora o número da porta que é usado por um serviço possa ser configurado, a maioria dos serviços escuta em portas “conhecidas” padrão. Os aplicativos de software cliente são projetados para se comunicar com tipos específicos de servidores. Os navegadores da Web são projetados para se comunicar com servidores da Web usando o protocolo HTTP na porta 80. Clientes FTP se comunicam com servidores FTP para transferir arquivos.

**Administração Básica do Servidor**

No Linux, os servidores são gerenciados usando arquivos de configuração. Várias configurações podem ser modificadas e salvas em arquivos de configuração. Quando um serviço é iniciado, ele examina seu (s) arquivo (s) de configuração para saber como ele deve ser executado. Não há nenhuma regra para a forma como os arquivos de configuração são gravados. A formatação do arquivo de configuração depende do criador do software do servidor. Os dispositivos Linux devem ser protegidos usando métodos comprovados para proteger o dispositivo e o acesso administrativo. Isso é conhecido como fortalecimento (hardering) dos dispositivos. Uma maneira de fortalecer um dispositivo é manter senhas, configurar recursos de login aprimorados e implementar login remoto seguro com SSH. Também é muito importante manter o sistema operacional atualizado. Outras formas de reforçar um dispositivo são forçar alterações periódicas de senha, impor senhas fortes e impedir a reutilização de senhas. Finalmente, clientes e servidores Linux usam arquivos de log para registrar a operação do sistema e eventos importantes. Vários arquivos de log diferentes são mantidos, incluindo logs de aplicativos, logs de eventos, logs de serviço e logs do sistema. O servidor registra atividades que são conduzidas por usuários remotos que acessam serviços do sistema. É importante saber a localização de diferentes logs no sistema de arquivos Linux para que eles possam ser acessados e monitorados em busca de problemas.

**O sistema de arquivos Linux**

O Linux suporta vários sistemas de arquivos diferentes que variam de acordo com velocidade, flexibilidade, segurança, tamanho, estrutura, lógica e muito mais. Alguns dos sistemas de arquivos suportados pelo Linux são ext2, ext3, ext4, NFS e CDFS. Os sistemas de arquivos são montados em partições e acessados através de pontos de montagem ou diretórios. Letras de unidade do Windows são exemplos de pontos de montagem. O comando **mount** pode ser usado para exibir detalhes dos sistemas de arquivos que estão atualmente montados em um computador Linux. O sistema de arquivos raiz é representado pelo símbolo “**/**”. Ele contém todos os arquivos no computador por padrão. O Linux usa permissões de arquivo para controlar quem tem permissão para ter diferentes tipos de acesso a arquivos e diretórios. As permissões incluem leitura (read) (r), escrita (write) (w) e execução (execute) (x). Arquivos e diretórios têm permissões atribuídas a usuários, grupos e outros. As permissões para arquivos e pastas são exibidas com o comando **ls -l** . Esse comando também exibe os links para um arquivo. Links rígidos criam outro arquivo com um nome diferente que está vinculado ao mesmo local no sistema de arquivos. O proprietário do arquivo e o grupo para o arquivo também são exibidos juntamente com a data e a hora da última modificação do arquivo. Permissões de arquivo são recursos poderosos do sistema de arquivos Linux e não podem ser violadas. Apenas o usuário root pode substituir as permissões do arquivo. Devido ao poder do usuário root, o acesso root deve ser cuidadosamente controlado. Links rígidos são criados com o comando **ln** . Alterações em um dos arquivos vinculados também são feitas no arquivo original. Links simbólicos, ou links simbólicos, são semelhantes aos links rígidos em que uma alteração no arquivo vinculado é refletida no arquivo original. Links simbólicos têm várias vantagens sobre links rígidos.

**Trabalhando com Linux GUI**

O sistema X Windows, ou X11, é uma estrutura de software básica que inclui funções para criar, controlar e configurar uma GUI do Windows em uma interface de apontar e clicar. Diferentes fornecedores usam o sistema X Windows para criar diferentes GUIs do gerenciador de janelas para Linux. Exemplos de gerenciadores de janelas são o Gnome e o KDE. A distribuição Ubuntu Linux usa o Gnome 3 por padrão. A área de trabalho do Gnome 3 consiste no Menu Apps, Ubuntu Dock, Barra Superior, Tabuleiro Calendário e Mensagem do Sistema, na área Atividades e no Menu Status.

**Working on a Linux Host**

Para instalar aplicativos em hosts Linux, programas chamados gerenciadores de pacotes são usados. Os pacotes são aplicativos de software e todos os seus arquivos de suporte. Os gerenciadores de pacotes são extremamente úteis para instalar aplicativos de software complexos a partir de repositórios de pacotes centralizados acessíveis pela Internet. Diferentes distribuições Linux usam gerenciadores de pacotes diferentes. Por exemplo, Arch Linux usa **pacman**, Debian usa **dpkg** como gerenciador de pacotes base e **apt** para se comunicar com **dpkg**. Ubuntu também usa **apt**. Os comandos CLI do gerenciador de pacotes são usados para instalar, remover e atualizar pacotes de software. Os comandos de atualização atualizam todos os pacotes atualmente instalados. O gerenciamento de pacotes também pode ser realizado em uma GUI. Processos de software são instâncias de programas de computador em execução. Os sistemas operacionais multitarefa podem executar vários processos ao mesmo tempo. A bifurcação (fork) é um método que o kernel usa para permitir que um processo em execução copie a si mesmo. O comando **ps** lista os processos em execução, **top** exibe informações sobre processos em execução dinamicamente e **kill** é usado para remover, reiniciar ou pausar processos em execução. Embora o Linux seja considerado melhor protegido contra software malicioso (malware) do que outros sistemas operacionais, ele ainda é suscetível a cavalos de Tróia, worms e outros tipos de malware. O Linux é geralmente atacado através de seus serviços e processos. Software desatualizado é frequentemente vulnerável a ataques. Os atores de ameaças podem sondar um dispositivo em busca de portas abertas vinculadas a processos de servidor desatualizados. Com esse conhecimento, os ataques podem ser lançados. É importante manter o sistema operacional e seus componentes e aplicativos atualizados. O programa **chkrootkit** foi projetado para detectar malware rootkit. Rootkits são programas de malware de nível profundo que são muito difíceis de detectar e remover. Eles podem alterar a operação fundamental do próprio sistema operacional e podem ser usados para criar acesso não autorizado aos sistemas. Os comandos de tubulação usam o símbolo “|” para encadear diferentes comandos usando a saída de um comando como entrada para outro.