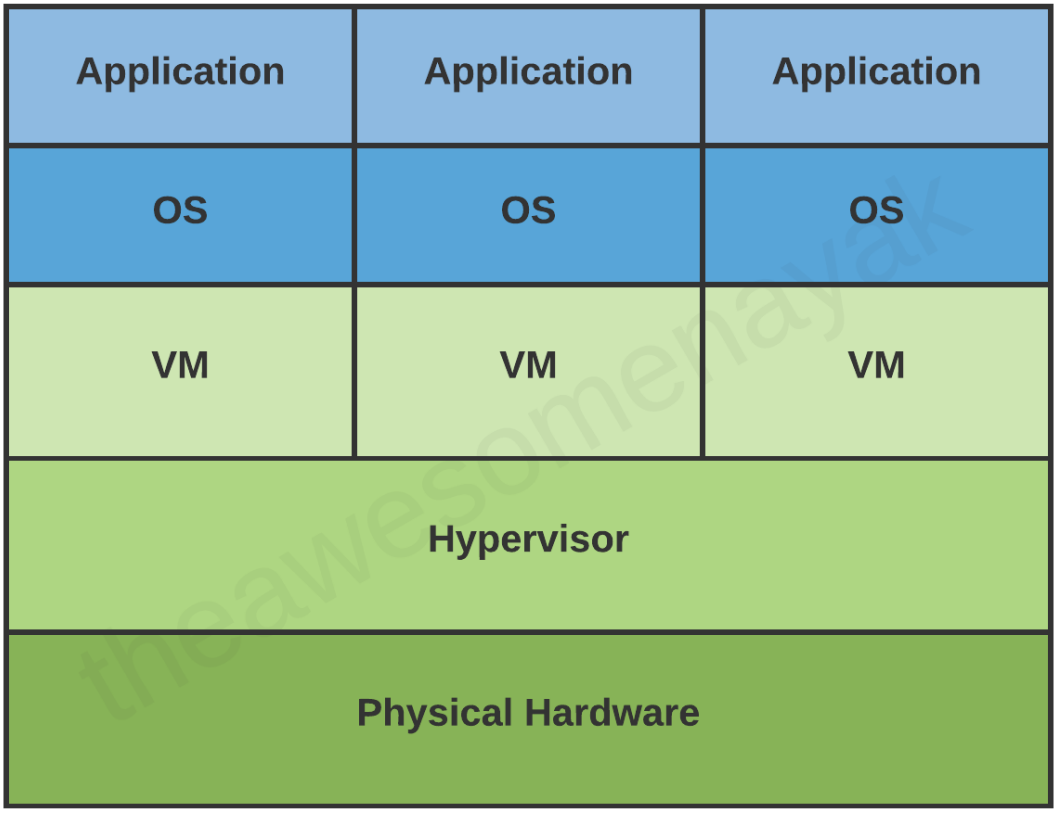
Introdução ao Linux para CTFs

Page • 1 backlink

Pré-Requisitos

Máquinas Virtuais



Máquinas virtuais (VMs) são emulações de sistemas computacionais que operam de maneira isolada dentro de um computador físico. Elas permitem a execução de sistemas operacionais (SO) e aplicativos como se estivessem em hardware dedicado, proporcionando um ambiente seguro e controlado. VMs são amplamente usadas para otimizar o uso de recursos de hardware, permitir a execução de múltiplos sistemas operacionais em um único host e facilitar o desenvolvimento e testes de software em ambientes replicáveis.

O funcionamento das VMs é gerenciado por um software chamado hypervisor, que pode ser do tipo 1 (bare-metal), executado diretamente no hardware, ou do

tipo 2, rodando sobre um sistema operacional host. O hypervisor aloca recursos de CPU, memória e armazenamento para cada VM, garantindo que elas funcionem de forma independente. Esse isolamento é crucial para a segurança e estabilidade, pois problemas em uma VM não afetam as outras nem o sistema host.

A utilização de VMs oferece diversas vantagens, como maior eficiência de recursos, flexibilidade na criação e modificação de ambientes e custos reduzidos com hardware. Não é preciso entrar em detalhes no momento sobre o seu funcionamento interno, porém é importante ressaltar que ela é uma ferramenta importantíssima para se ter no seu arsenal de conhecimentos.

Virtualbox: https://www.virtualbox.org/

Guia para instalação do Kali Linux no Virtualbox: https://www.kali.org/docs/virtualization/install-virtualbox-guest-vm/

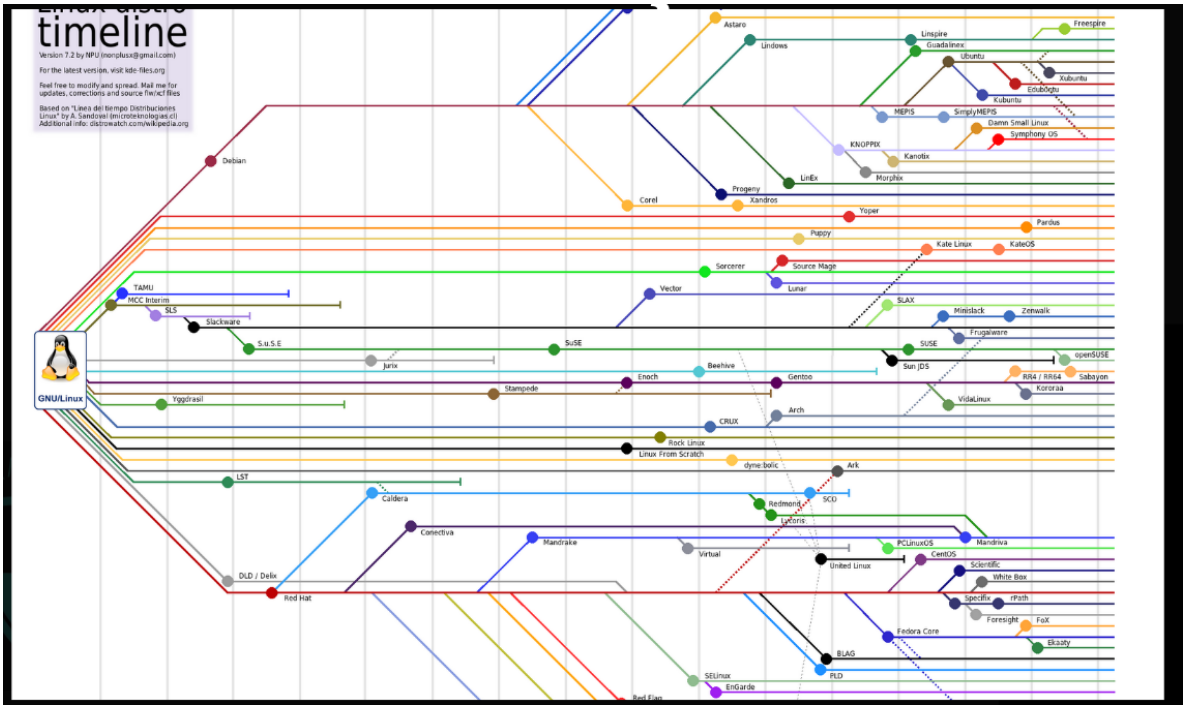
O que é o Linux?

Linux é um sistema operacional criado por Linux Torvalds em 1991, em conjunto com a Free Software Foundation que atribui a terminologia correta como sendo GNU/Linux.

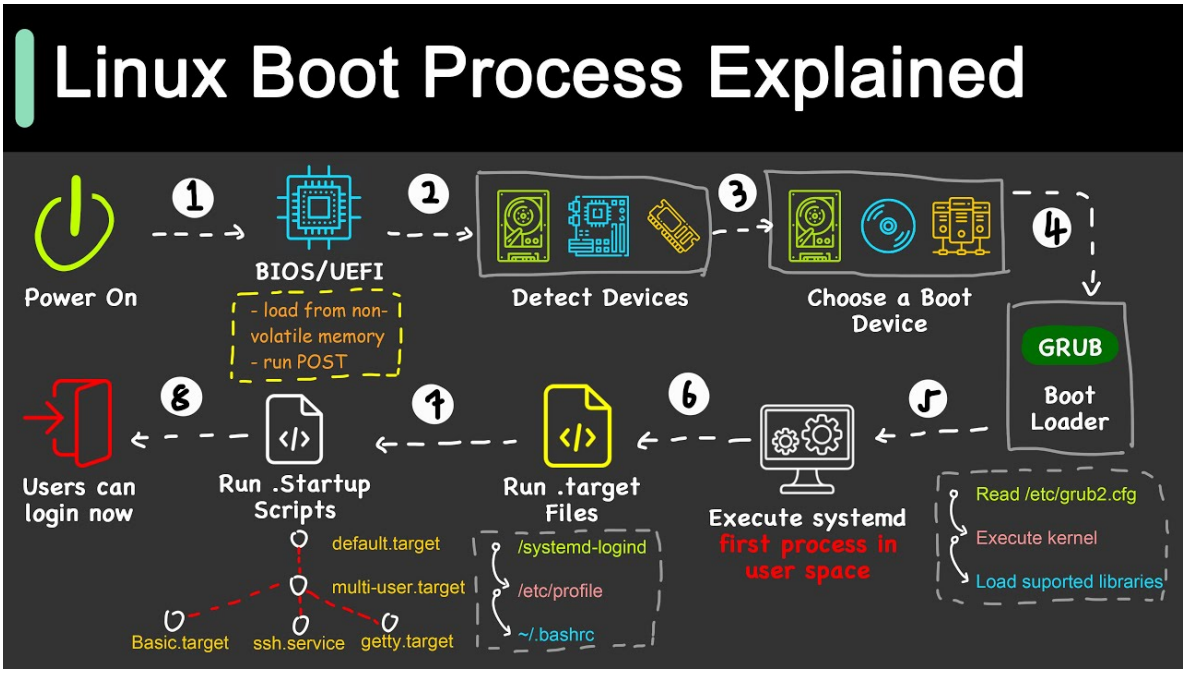
É considerado como um sistema operacional que possui certas características, como:

- Altamente Customizável
- Código Livre e Aberto (FOSS)
- Mantido pela Comunidade
- Estável e Seguro
- Multiplataforma

Uma atributo predominante é sua customização, isso é evidenciado pelas distruibuições que são versões específicas do Linux que incluem o seu kernel (quem faz a interface entre o hardware e o software, ele é responsável por controlar e gerenciar os recursos do sistema).



Processo de Inicialização e Estrutura do Linux



O processo de inicialização geralmente começa com a inicialização da BIOS/UEFI (firmware responsável por ligar o computador), ao ligar o computador ela realiza uma checagem de POST (Power-On Self Test) que é uma chacagem se os principais componentes do computador estão funcionando adequadamente.

Em seguida, o firmware carrega o bootloader, que é responsável por carregar o kernel do Linux na memória RAM. O kernel, ao ser carregado, assume o controle e realiza uma série de inicializações de hardware, incluindo a detecção e configuração de dispositivos, como processador, memória, placas de rede e gráficos.

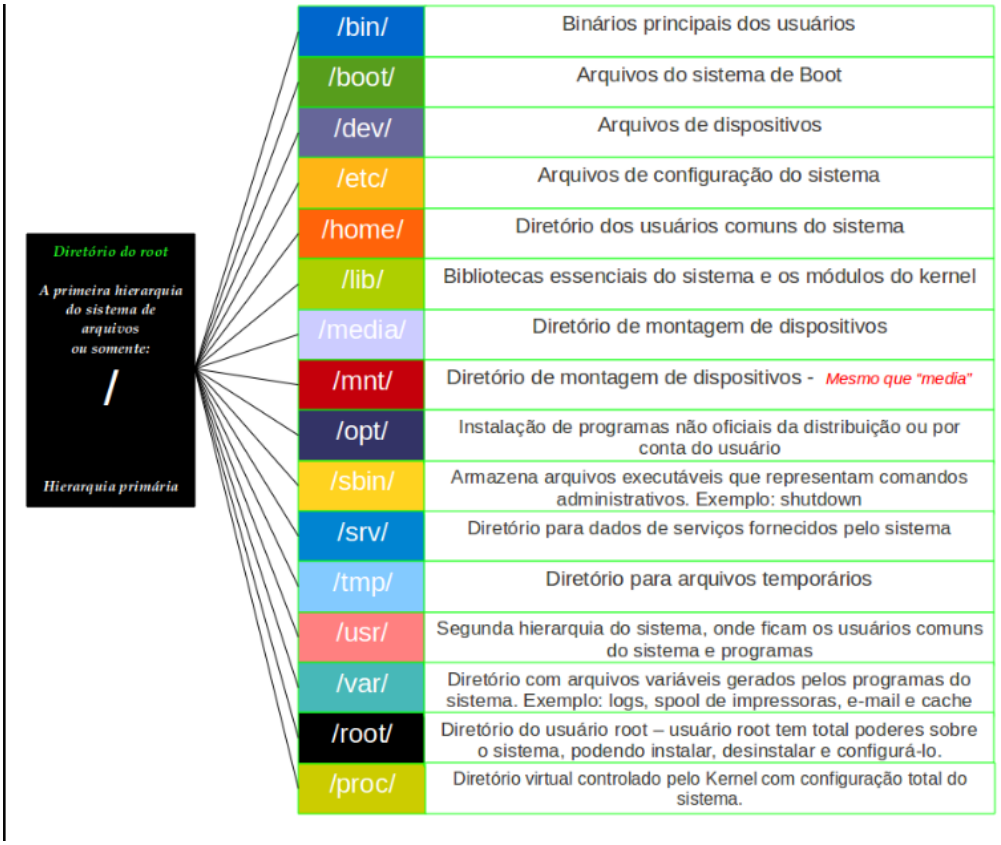
Além disso, o kernel monta o sistema de arquivos raiz, que contém todos os arquivos essenciais para o funcionamento do sistema operacional. Uma vez que o sistema de arquivos raiz esteja montado, o kernel inicia o init system, como o systemd, que é responsável por iniciar e gerenciar os processos e serviços essenciais do sistema, como o gerenciador de login, rede e outros.

Com os serviços essenciais iniciados, o sistema está pronto para que o usuário faça login. O gerenciador de login é exibido na tela, solicitando ao usuário que insira suas credenciais.

Após o login bem-sucedido, o ambiente de trabalho escolhido pelo usuário é iniciado, proporcionando assim a interação com o sistema operacional, onde o usuário pode executar aplicativos, acessar arquivos e realizar outras tarefas.

Este processo de inicialização é fundamental para que o sistema operacional Linux esteja pronto para uso.

Além de entender o seu processo de inicialização, é fundamental entender como é a estrutura de diretórios dado um sistema de arquivos. Com base na imagem abaixo, podemos explicar melhor sobre o que cada diretório tem por finalidade.



Shell Scripting e Terminal

Shell scripting é uma ferramenta poderosa para que possamos automatizar tarefas por meio da criação de scripts (sequência de comandos), por meio do shell (podendo ser acessada pelo terminal) que é a interface de linha de comando do sistema operacional.

Características

- Personalização e Extensibilidade com lógica condicional, loops e variáveis
- Facilidade no uso e aprendizado
- Flexibilidade e portabilidade em sistemas baseados em Unix
- Integração completa com o sistema operacional

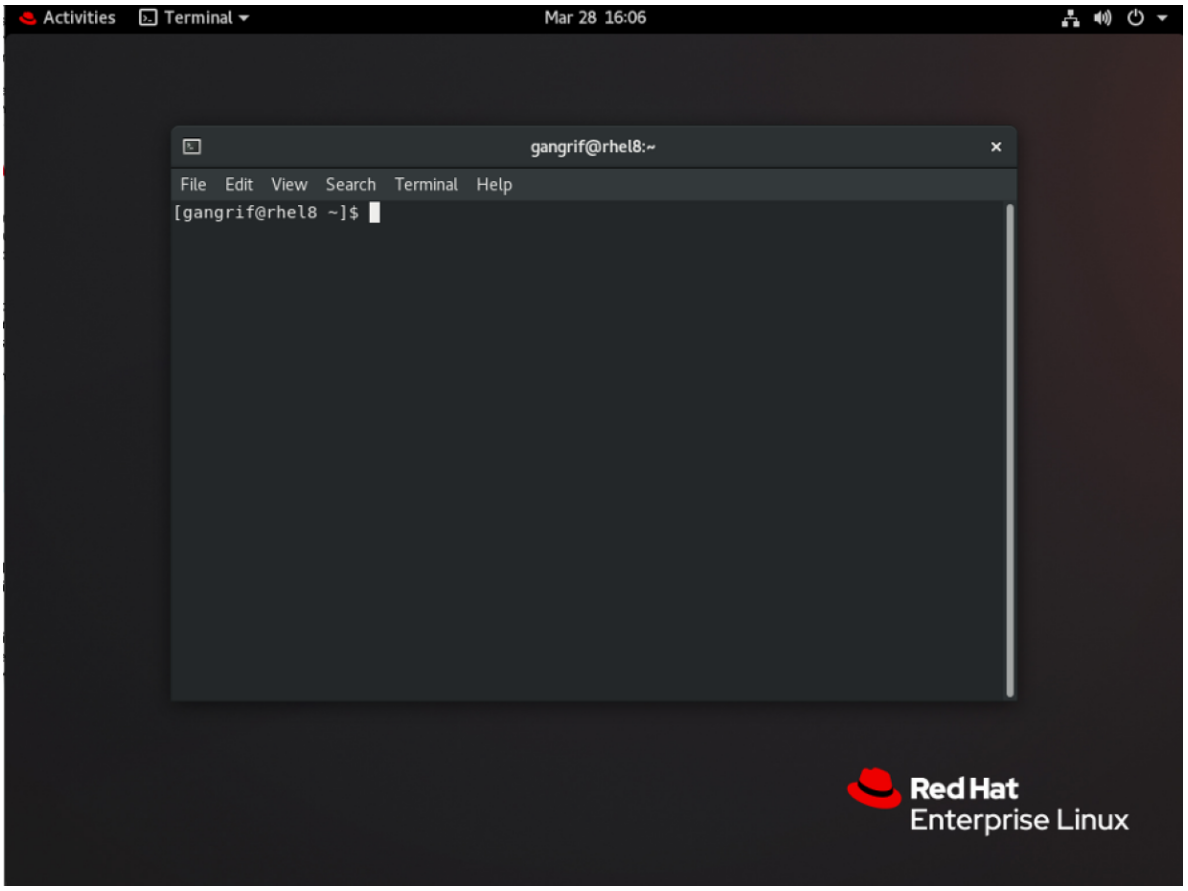
Bash

```bash
#!/bin/bash

# Este é um script de exemplo em shell scripting que demonstra as principais funcionalidades da linguagem.

# Definindo e exibindo uma variável
nome="Maria"
```

```
echo "Olá, $nome!"

# Solicitando entrada do usuário
echo "Por favor, digite sua idade:"
read idade

# Estrutura condicional para verificar se a idade é maior ou igual
a 18
if [ $idade -ge 18 ]; then
    echo "Você é maior de idade."
else
    echo "Você é menor de idade."
fi

# Loop for para exibir números de 1 a 5
echo "Contagem de 1 a 5:"
for i in {1..5}; do
    echo "$i"
done

# Função para calcular a soma de dois números
calcular_soma() {
    # Recebe dois parâmetros e retorna a soma
    local num1=$1
    local num2=$2
    local soma=$((num1 + num2))
    echo "A soma de $num1 e $num2 é: $soma"
}

# Chamando a função e passando dois números como argumentos
calcular_soma 10 5

# Redirecionamento de saída para um arquivo
echo "Escrevendo para um arquivo." > arquivo.txt

# Exibindo o conteúdo do arquivo
echo "Conteúdo do arquivo:"
cat arquivo.txt
```

Não se preocupe em dominar o shell scripting no momento, posteriormente ele será uma ferramenta de grande importância, por ora precisamos apenas nos habituar com os comandos do Linux.

Abaixo temos uma representação visual de uma janela do terminal, onde podemos inserir comandos.



Principais Comandos

Existem inúmeros comandos para se interagir no terminal Linux, aqui iremos apresentar os principais comandos utilizados nesse sistema operacional.

OBS.: Grande parte dos comandos podemos colocar argumentos após o seu nome, por exemplo, ls -a

Também é importante ressaltar que não é necessário decorar todos os comandos a princípio, inicialmente essa lista de comandos irá nos auxiliar como modo de consulta, caso seja necessário realizar alguma ação, referente a navegação do sistema, modificação de permissões, etc.

File Permissions

`chmod octal FILE`
Change permission of FILE

`4`
read (r)

`2`
write (w)

`1`
execute (x)

Compression

`tar cf FILE.tar files`
Tar files into FILE.tar

`tar xf FILE.tar`
Untar into current directory

`tar tf FILE.tar`
List the contents of an archive

`gzip FILE`
Compress FILE and rename to FILE.gz

`gzip -d FILE.gz`
Decompress FILE.gz

SSH

`ssh user@host`
Connect to host as user

`ssh -p port user@host`
Connect using port p

`ssh -D port user@host`
Connect and use bind port

Installation

`./configure`
Configure the source file

`make`
Compile the source code

`make install`
Install the compiled source code

Navigation

`ls`
List directory contents

`ls -alh`
Formatted long listing with hidden files

`cd DIR`
Change the current directory to the DIR directory

`cd ~`
Change current directory to $HOME

`cd /`
Change the current directory to the root directory

`cd ..`
Change to the parent of the current directory

`pwd`
Show name of current working directory

Searching and Sorting

`grep pattern FILE`
Search for pattern in FILE

`grep -r pattern DIR`
Search recursively for pattern in DIR

`command | grep pattern`
Search for the pattern in in the output of command

`find /dir/ -name name*`
Find files starting with name in dir

`find /dir/ -user name`
Find files owned by name in dir

`locate FILE`
Find all instances of FILE

`sort FILE`
Sort the content of FILE alphabetically

`sort -r FILE`
Sort in reverse

`sort -R FILE`
Sort randomly

Process Management

`top`
Show real time processes

`ps`
Report a snapshot of the current processes

`ps aux`
Show processes for all users

`kill [pid]`
Terminate a process

`bg`
Run jobs in the background

`fg`
Run jobs in the foreground

`fg n`
Bring job n to the foreground

`du`
Estimate file space usage

`du -sh`
Summarize file space usage and print sizes in human readable format

`free -h`
Display amount of free and used memory in the system

`whereis`
Locate the binary, source, and manual page files for a command

`which`
Locate a command

System Info

`date`
Print or set the system date and time

`cal`
Displays a calendar

`uptime`
Tell how long the system has been running

`w`
Show who is logged on and what they are doing

`whoami`
Print effective userid

`hostname`
Show host name

`uname -a`
Print system information

`cat /proc/cpuinfo`
Print the cpu info

`cat /proc/meminfo`
Print the memory information

Network

`ping host`
Ping host 'host'

`whois domain`
Get whois for domain

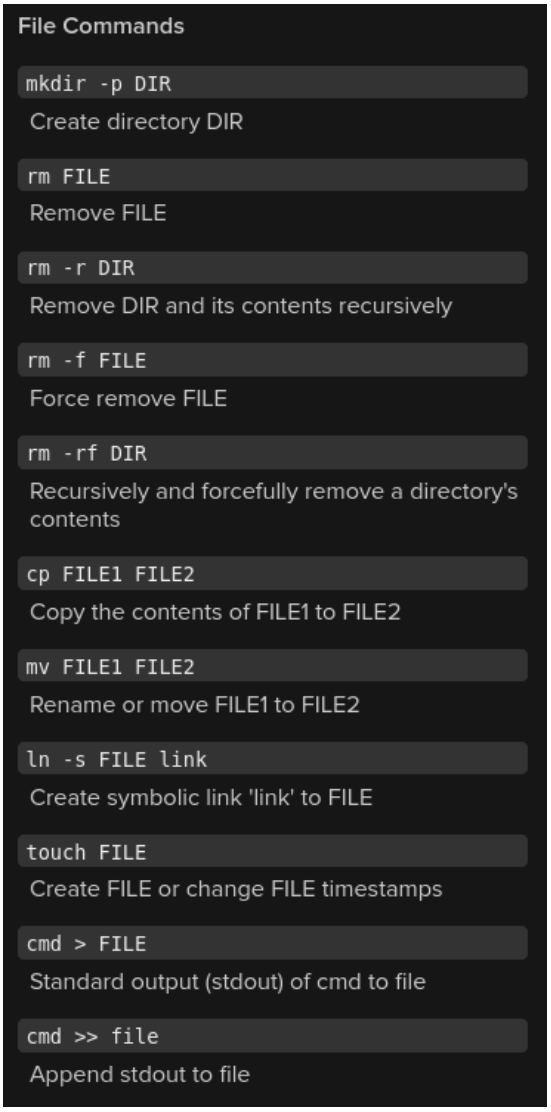
`dig domain`
Get DNS for domain

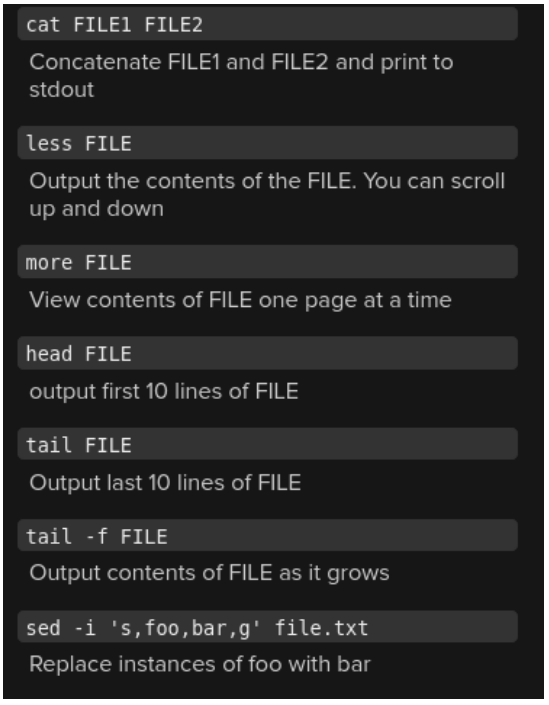
`dig -x host`
Reverse lookup host

`wget file`
Download file

`wget -c file`
Continue stopped download

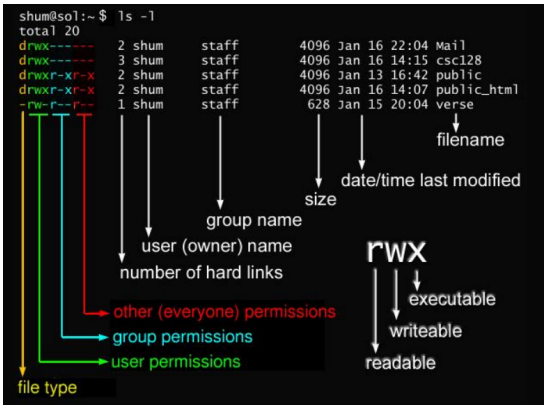
`wget -r url`
Recursively download files from url





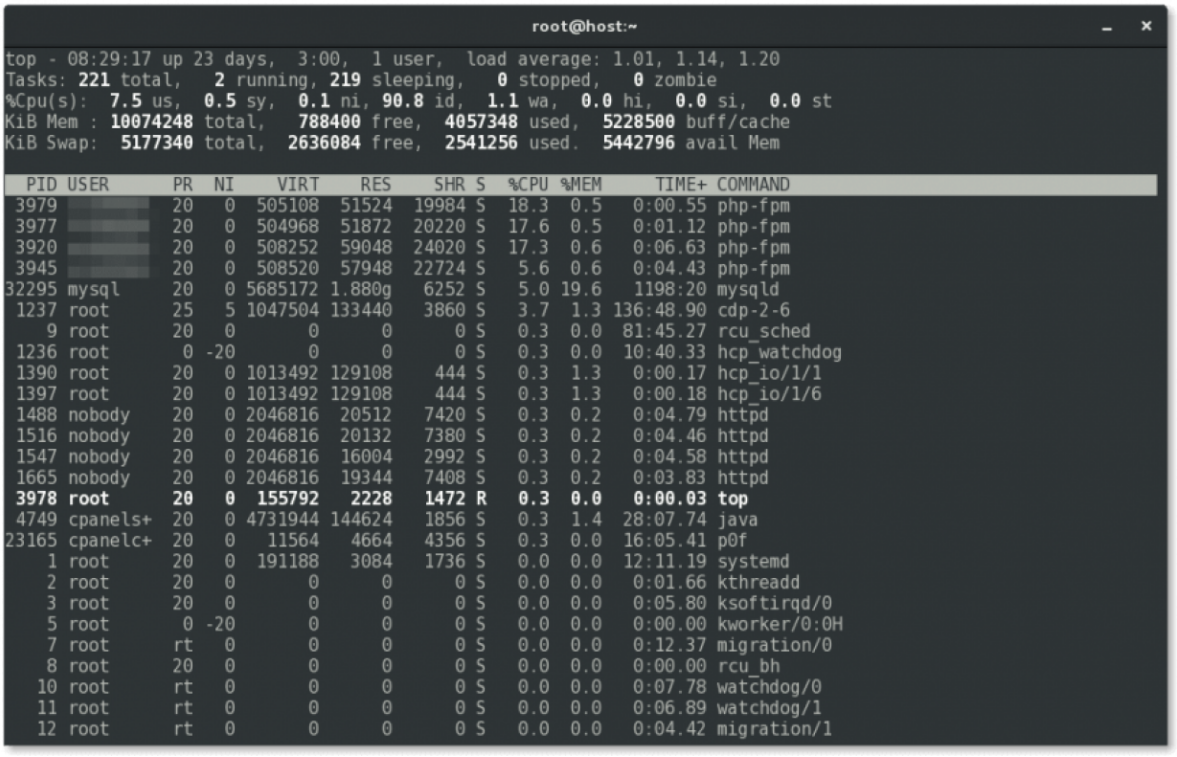
Um comando muito importante de ser abordado, que será muito útil em nossa jornada é o chmod, ele é responsável por modificar permissões de arquivos e diretórios. Para entender melhor ele, as imagens abaixo explicam sua estrutura básica. Para simplificar a explicação, suas permissões são divididas em operações de leitura, escrita e execução, podendo ser alteradas com base nos valores da tabela abaixo.

Octal	Decimal	Permission	Representation
000	0 (0+0+0)	No Permission	---
001	1 (0+0+1)	Execute	--x
010	2 (0+2+0)	Write	-w-
011	3 (0+2+1)	Write + Execute	-wx
100	4 (4+0+0)	Read	r--
101	5 (4+0+1)	Read + Execute	r-x
110	6 (4+2+0)	Read + Write	rw-
111	7 (4+2+1)	Read + Write + Execute	rwx



Monitoramento de Processos

Ao monitorar processos podemos obter informações importantes a respeito do uso de recursos e sua disponibilidade no sistema. Um programa utilizado para realizar esse monitoramento é o top (existem outras variações mais robustas como o btop e htop). Por meio dele é possível ter uma visão geral de aspectos de tracing no sistema.



- Nomenclatura
    - PID (Identificador de Processos)
    - USER (Login do Proprietário do Processo
    - PR (Prioridade do Processo)
    - NI
    - VIRT (Quantidade de Memória Virtual utilizada no
    - processo)
    - RES (Quantidade de Memória Física utilizada no
    - processo)
    - SHR (Quantidade de Memória Compartilhada pelo processos)

- S (Indica o Status do processo)
- S (Sleep)
- R (Running)
- Z (Zombie)
- TIME+ (Tempo Total de atividade do processo)
- %CPU (Porcentagem de CPU usada no processo)
- %MEM (Porcentagem de Memória Virtual usada pelo processo)

Complemento Teórico do Assunto (Importante!)

Aqui estão alguns recursos teóricos muito completos a respeito do assunto abordado nessa semana, fiquem a vontade para checa-los e levantar o que aprenderam durante nosso encontro semanal.

Qualquer dúvida estamos a disposição! Bons estudos!

- https://tryhackme.com/r/room/linuxmodules
- https://tryhackme.com/r/room/linuxfundamentalspart1
- https://tryhackme.com/r/room/linuxfundamentalspart2
- https://tryhackme.com/r/room/linuxfundamentalspart3
- https://academy.hackthebox.com/module/details/18 (Linux Fundamentals - HTB)
- https://academy.hackthebox.com/module/details/21 (Introduction to Bash Scripting)

Materiais

Vídeos e Playlists

- https://www.youtube.com/playlist?list=PLXoSGejyuQGqJEEyo2fY3SA-QCKlF2rxO (Curso Shell GNU)
- https://www.youtube.com/playlist?list=PLXoSGejyuQGrjEIS_tIJ7XYJTcc1ggQy- (Criação de scripts em Bash)

- https://youtu.be/3OawXnTELqA?list=PL_Px_tgmLSheuxBHmbJrIYJYUQQVZfste (general skills in ctfs)
- https://youtu.be/sWbUDq4S6Y8 (Introduction to Linux - freecodecamp)
- https://youtube.com/playlist?list=PLIhvC56v63IJIujb5cyE13oLuyORZpdkL (Linux for Hackers - Network Chuck)

Livros

- Linux Basics for Hackers: Getting Started with Networking, Scripting, and Security in Kali by OccupyTheWeb
- The Linux Command Line: A Complete Introduction by William E. Shotts Jr.
- Linux Basics for Hackers: Secure Kali Linux-Based Systems by OccupyTheWeb
- Mastering Kali Linux for Advanced Penetration Testing by Vijay Kumar Velu, Robert Beggs, Michael Beggs
- Linux Server Security: Hack and Defend by Chris Binnie

Sites

- https://blauaraujo.com/tgl/
- https://training.linuxfoundation.org/training/introduction-to-linux/

Plataformas de CTF que serão utilizadas

- https://tryhackme.com/
- https://www.hackthebox.com/
- https://picoctf.org/
- https://overthewire.org/wargames/

Desafios

War Games

O site OverTheWire WarGames possui uma série de desafios que visa auxliar no aprendizado em cibersegurança, a sua dificuldade vai escalando conforme os níveis, e conforme ela escala outros temas são abordados, como criptografia, exploração de binários e etc. Os desafios selecionados abaixo são focados em Linux

- https://overthewire.org/wargames/bandit/bandit0.html (level 0)

- https://overthewire.org/wargames/bandit/bandit1.html (level 1)
- https://overthewire.org/wargames/leviathan/ (level 2)

picoCTF

- https://play.picoctf.org/practice/challenge/147?category=5&page=1 (Obedient Cat)
- https://play.picoctf.org/practice/challenge/170?category=5&page=1 (Wave a flag)
- https://play.picoctf.org/practice/challenge/156?category=5&page=1 (Nice netcat...)
- https://play.picoctf.org/practice/challenge/163?category=5&page=1 (Static ain't always noise)
- https://play.picoctf.org/practice/challenge/176?category=5&page=1 (Tab, Tab, Attack)
- https://play.picoctf.org/practice/challenge/424?category=5&page=1 (Super SSH)
- https://play.picoctf.org/practice/challenge/189?category=5&page=1 (Magikarp Ground Mission)
- https://play.picoctf.org/practice/challenge/425?category=5&page=2 (Time Machine)
- https://play.picoctf.org/practice/challenge/34?category=5&page=2 (what's a net cat?)
- https://play.picoctf.org/practice/challenge/37?category=5&page=2 (strings it)
- https://play.picoctf.org/practice/challenge/85?category=5&page=2 (First Grep)
- https://play.picoctf.org/practice/challenge/347?category=5&page=3 (chrono)
- https://play.picoctf.org/practice/challenge/363?category=5&page=4 (Permissions)
- https://play.picoctf.org/practice/challenge/384?category=5&page=4 (useless)
- https://play.picoctf.org/practice/challenge/48?category=5&page=4 (plumbing)
- https://play.picoctf.org/practice/challenge/377?category=5&page=4 (Special)

- https://play.picoctf.org/practice/challenge/378?category=5&page=5 (Specialer)

TryHackMe

Obs.: Alguns desafios envolvem outros conceitos além do Linux em si.

Fáceis e Médio

- https://tryhackme.com/room/basicpentestingjt
- https://tryhackme.com/room/picklerick
- https://tryhackme.com/room/kenobi
- https://tryhackme.com/room/cowboyhacker
- https://tryhackme.com/room/overpass2hacked

Difíceis

- https://tryhackme.com/room/anonymousplayground
- https://tryhackme.com/room/yearofthepig
- https://tryhackme.com/room/seasurfer
- https://tryhackme.com/room/m4tr1xexitdenied
- https://tryhackme.com/room/plottedlms