Linux Structure

Linux, jaisa ki aap pehle se jaante honge, ek Operating System hai. Iska use Personal Computers, Servers aur yahan tak ki Mobile Devices mein bhi hota hai.

Lekin Cybersecurity mein Linux ek bahut important foundation hai. Yeh apni mazbooti (Robustness), flexibility aur Open-Source nature ke liye jaana jaata hai.

Is section mein hum Linux ke baare mein kuch important topics padhenge:

- Linux Structure
- Linux History
- Linux Philosophy
- Linux Architecture
- File System Hierarchy

Yeh sabhi topics kisi bhi Cybersecurity Professional ke liye bahut zaroori knowledge hain.

Aap ise aise samajh sakte hain jaise aap pehli baar koi nayi car chalana seekh rahe hain. Sabse pehle aapko car ki basic jaankari di jaati hai, jaise car kin cheezon se milkar bani hai aur ise is tarah kyun banaya gaya hai.

Shuruaat karne ke liye, sabse pehle samajhte hain ki Linux kya hai.

Linux ek Operating System (OS) hai, bilkul Windows, macOS, iOS aur Android ki tarah.

Operating System (OS) ek software hai jo computer ke saare Hardware Resources ko manage karta hai.

Yeh Software Applications aur Hardware Components ke beech communication karne mein madad karta hai.

Kuch doosre Operating Systems se alag, Linux ke bahut saare versions hote hain, jinhe aksar Distributions (Distros) kaha jaata hai.

Linux ke ye alag-alag versions different needs aur preferences ke hisaab se banaye jaate hain.

History

Linux ka pehla Kernel aur aakhir mein Linux Operating System (OS) banne ke peeche kai events hue the. Iski shuruaat 1970 mein hui, jab Ken Thompson aur Dennis Ritchie ne Unix Operating System release kiya. Us samay dono AT&T company mein kaam karte the.

1977 mein Berkeley Software Distribution (BSD) release hua, lekin kyunki ismein Unix ka code tha, jiska ownership AT&T ke paas tha, isliye ek legal case (lawsuit) hua, jiski wajah se BSD ka development limited ho gaya.

1983 mein Richard Stallman ne GNU Project shuru kiya. Unka goal ek free Unix-like Operating System banana tha. Unke kaam ke ek result ke roop mein GNU General Public License (GPL) banaya gaya.

Kai saalon tak doosre developers ke projects ek working aur free Kernel banane mein successful nahi ho paaye, jise widely adopt kiya ja sake. Aakhirkar Linux Kernel create hua.

Shuruaat mein Linux ek personal project tha, jise 1991 mein Finland ke ek student Linus Torvalds ne shuru kiya tha. Unka goal ek naya aur free Operating System Kernel banana tha.

Saalon ke dauran Linux Kernel mein bahut changes hue. Pehle Linux Kernel kuch hi files se bana tha, jo C programming language mein likhi gayi thi. Us samay iska license commercial distribution ko allow nahi karta tha.

Lekin aaj Linux Kernel ke latest version mein 23 million se bhi zyada source code lines hain (comments ko chhodkar). Yeh GNU General Public License v2 ke under licensed hai.

Linux 600 se bhi zyada Distributions (yaani Linux Kernel, supporting software aur libraries par based Operating Systems) mein available hai.

Kuch sabse popular aur well-known Distributions hain: Ubuntu, Debian, Fedora, OpenSUSE, elementary, Manjaro, Gentoo Linux, RedHat aur Linux Mint.

Generally, Linux ko doosre Operating Systems ke comparison mein zyada secure maana jaata hai. Past mein Linux Kernel mein kai vulnerabilities mili hain, lekin ab aisi vulnerabilities milne ki frequency kam hoti ja rahi hai.

Linux, Windows Operating Systems ke comparison mein Malware se kam affected hota hai aur ise bahut frequently update kiya jaata hai.

Linux bahut stable bhi hai aur generally end-user ko bahut achhi performance provide karta hai. Lekin beginners ke liye ise samajhna thoda difficult ho sakta hai aur Windows ke comparison mein iske liye Hardware Drivers bhi kam available hain.

Kyunki Linux free aur Open-Source hai, isliye koi bhi iska Source Code modify kar sakta hai aur ise commercially ya non-commercially distribute kar sakta hai.

Linux-based Operating Systems ka use Servers, Mainframes, Desktops, Embedded Systems (jaise Routers), Televisions, Video Game Consoles aur doosri devices mein hota hai.

Smartphones aur Tablets mein chalne wala Android Operating System bhi Linux Kernel par based hai. Isi wajah se Linux duniya mein sabse zyada install kiye jaane wale Operating Systems mein se ek hai.

Linux bhi Windows, iOS, Android aur macOS ki tarah ek Operating System hai.

OS ek Software hai jo hamare computer ke saare Hardware Resources ko manage karta hai. Iska matlab hai ki OS Software aur Hardware ke beech hone wale poore communication ko manage karta hai.

Iske alawa, Linux ki bahut saari alag-alag Distributions (Distros) bhi available hain. Yeh Windows Operating Systems ke alag-alag versions ki tarah hain.

Interactive instances ke through humein Pwnbox ka access milta hai, jo Parrot OS ka ek customized version hai.

Modules ke dauran hum isi OS ke saath kaam karenge.

Parrot OS ek Debian-based Linux Distribution hai, jiska main focus Security, Privacy aur Development par hai.

Linux ko ek aisi company ki tarah imagine karo jo achhe se chal rahi hai. Iske components dedicated employees ki tarah hain, jinki apni specific roles aur responsibilities hoti hain, taaki company ka kaam smoothly chalta rahe.

Architecture company ke organizational structure ki tarah hota hai. Yeh batata hai ki employees ko alag-alag departments mein kaise arrange kiya gaya hai aur woh efficiency aur productivity achieve karne ke liye ek-doosre se kaise communicate karte hain.

Philosophy company ki culture aur core values ki tarah hoti hai. Yeh guide karti hai ki employees individually aur teamwork mein kaise kaam karenge. Yeh Simplicity, Transparency aur Cooperation jaise principles ko promote karti hai, taaki sabhi milkar common goals achieve kar sakein.

Philosophy

Linux ki Philosophy simplicity, modularity aur openness par based hai.

Yeh chhote-chhote programs banane ko badhava deti hai, jo sirf ek specific kaam karte hain aur us kaam ko achhe se perform karte hain.

In programs ko alag-alag tarikon se combine karke complex operations perform kiye ja sakte hain.

Isse efficiency aur flexibility badhti hai.

Linux paanch basic principles ko follow karta hai

Principle	Description
Everything is a file	Linux mein sabhi system resources, jaise Hardware Devices, Processes aur Network Connections ko Files ke roop mein represent kiya jaata hai. Isse hum same standard tools aur commands ka use karke in files ko read aur write kar sakte hain.
Small, single-purpose programs	Linux mein bahut saare alag-alag tools available hain, jin par hum kaam karenge. In tools ko ek saath combine karke use kiya ja sakta hai.
Ability to chain programs together to perform complex tasks	Alag-alag tools ko aapas mein combine karke hum bade aur complex tasks perform kar sakte hain, jaise specific data results ko process ya filter karna.
Avoid captive user interfaces	Linux ko mainly Shell (ya Terminal) ke through kaam karne ke liye design kiya gaya hai, jisse user ko Operating System par zyada control milta hai.
Configuration data stored in a text file	Iska ek example **/etc/passwd** file hai, jisme system par registered sabhi users ki information store hoti hai.

Components

Component	Description
Bootloader	Yeh ek code hota hai jo Operating System ko start karne ke liye Booting Process ko guide karta hai. Parrot Linux mein GRUB Bootloader ka use hota hai.
OS Kernel	Kernel, Operating System ka main component hota hai. Yeh Hardware Level par system ke I/O Devices ke resources ko manage karta hai.
Daemons	Linux mein Background mein chalne wali services ko "Daemons" kaha jaata hai. Inka kaam important functions ko sahi tarah se chalana hota hai, jaise Scheduling, Printing aur Multimedia. Ye chhote programs computer ko boot karne ya login karne ke baad load hote hain.
OS Shell	Operating System Shell ya Command Language Interpreter (jise Command Line bhi kaha jaata hai), OS aur user ke beech ek interface hota hai. Is interface ke through user OS ko batata hai ki use kya karna hai. Sabse commonly used Shells hain: Bash, Tcsh/Csh, Ksh, Zsh aur Fish.
Graphics server	Yeh ek graphical subsystem (server) provide karta hai, jise "X" ya "X-server" kaha jaata hai. Yeh X-windowing system par graphical programs ko locally ya remotely run karne ki permission deta hai.
Window Manager	Ise Graphical User Interface (GUI) bhi kaha jaata hai. Iske kai options hain, jaise GNOME, KDE, MATE, Unity aur Cinnamon. Ek Desktop Environment mein usually kai applications hoti hain, jaise File Browser aur Web Browser. Inki madad se user Operating System ke important aur frequently used features aur services ko access aur manage kar sakta hai.

Utilities	Applications ya Utilities aise programs hote hain jo user ya kisi doosre program ke liye specific functions perform karte hain.

Linux Architecture

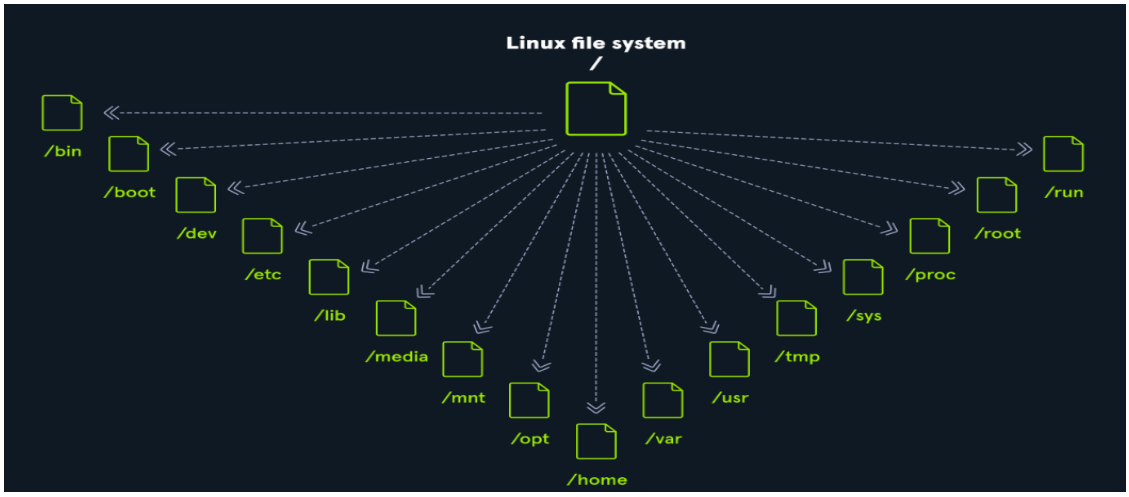
Linux Operating System ko alag-alag layers mein divide kiya ja sakta hai:

Layer	Description
Hardware	Computer ke physical devices, jaise RAM, Hard Drive, CPU aur doosre Peripheral Devices.
Kernel	Kernel, Linux Operating System ka main part hota hai. Iska kaam CPU, allocated memory aur accessed data jaise common computer hardware resources ko virtualize aur control karna hota hai. Kernel har Process ko apne alag Virtual Resources deta hai aur different Processes ke beech hone wale conflicts ko rokta hai ya kam karta hai.
Shell	Yeh ek Command-Line Interface (CLI) hota hai, jise Shell bhi kaha jaata hai. Ismein user commands enter kar sakta hai, jinke through Kernel ke functions execute hote hain.
System Utility	Yeh user ko Operating System ke sabhi available functions use karne ki facility provide karta hai.

File System Hierarchy

Linux Operating System ko ek tree-like structure (पेड़ जैसी संरचना) mein arrange kiya gaya hai aur iski documentation Filesystem Hierarchy Standard (FHS) mein di gayi hai.

Linux ko kuch standard top-level directories ke saath structure kiya gaya hai:



Path	Description (Hinglish)
/	Yeh Top-level directory hoti hai, jise Root Filesystem kaha jaata hai. Ismein Operating System ko boot karne ke liye zaroori files hoti hain. Ismein woh files bhi hoti hain jo doosre Filesystems ko boot karne ke liye required hoti hain. Boot hone ke baad, baaki sabhi Filesystems ko Root directory ke andar standard Mount Points par Subdirectories ke roop mein mount kiya jaata hai.
/bin	Ismein zaroori Command Binaries (commands chalane wali executable files) hoti hain.
/boot	Ismein Static Bootloader, Kernel Executable aur Linux OS ko boot karne ke liye zaroori files hoti hain.
/dev	Ismein Device Files hoti hain, jo system se connected har Hardware Device ko access karne mein help karti hain.
/etc	Ismein Local System Configuration Files hoti hain. Installed Applications ki Configuration Files bhi yahan save ho sakti hain.

/home	System ke har user ke liye yahan ek alag Subdirectory hoti hai, jismein user apna data store karta hai.
/lib	Ismein Shared Library Files hoti hain, jo system ko boot karne ke liye zaroori hoti hain.
/media	External Removable Media Devices, jaise USB Drives, ko yahan Mount kiya jaata hai.
/mnt	Yeh Regular Filesystems ko temporarily Mount karne ke liye use hone wala Mount Point hai.
/opt	Optional Files, jaise Third-party Tools, ko yahan save kiya ja sakta hai.
/root	Yeh Root User ki Home Directory hoti hai.
/sbin	Is Directory mein System Administration ke liye use hone wali Executable Files (Binary System Files) hoti hain.
/tmp	Operating System aur kai Programs temporary files store karne ke liye is Directory ka use karte hain. System Boot hone par generally is Directory ka data clear kar diya jaata hai. Kabhi-kabhi bina kisi warning ke bhi iske data ko delete kiya ja sakta hai.
/usr	Ismein Executable Files, Libraries, Man Files aur doosri files hoti hain.
/var	Is Directory mein Variable Data Files hoti hain, jaise Log Files, Email Inboxes, Web Application se related files, Cron Files aur doosri files.

Linux Distributions

Linux distributions – ya distros – aise operating systems hote hain jo Linux Kernel par based hote hain.

Inka use alag-alag purposes ke liye kiya jata hai, jaise servers, embedded devices, desktop computers aur mobile phones.

Linux distributions ek hi company ki alag-alag branches ya franchises ki tarah hote hain. Har distribution ko specific markets ya users ki preferences ko poora karne ke liye banaya jata hai.

Jis tarah sabhi branches mein dedicated employees (components), organizational structure (architecture) aur corporate culture (philosophy) same hote hain, usi tarah Linux distributions mein bhi kuch common cheezein hoti hain.

Lekin har distribution apne unique products aur services (software packages aur configurations) provide karta hai, taaki alag-alag users ki zarooraton ke according experience diya ja sake. Ye sab Linux ke ek hi brand aur values ke under kaam karte hain.

Har Linux distribution alag hota hai aur uske apne features, packages aur tools hote hain.

Kuch popular examples hain:

- **Ubuntu**
- **Fedora**
- **CentOS**
- **Debian**
- **Red Hat Enterprise Linux**

Bahut se users apne desktop computers ke liye Linux choose karte hain, kyunki ye free, open source aur highly customizable hai.

Ubuntu aur Fedora desktop Linux aur beginners ke liye do popular choices hain.

Linux ka use server operating system ke roop mein bhi bahut kiya jata hai, kyunki ye secure, stable aur reliable hai aur ismein frequent aur regular updates milte hain.

Aakhir mein, hum cybersecurity specialists bhi aksar Linux prefer karte hain, kyunki ye open source hai. Iska matlab hai ki iska source code available hota hai, jise check aur customize kiya ja sakta hai.

Isi customization ki wajah se hum apni zaroorat ke according Linux distribution ko optimize aur customize kar sakte hain aur zaroorat padne par use specific purposes ke liye configure kar sakte hain.

Hum in distros ka use har jagah kar sakte hain, jaise (web) servers, mobile devices, embedded systems, cloud computing aur desktop computing.

Cybersecurity specialists ke liye kuch sabse popular Linux distributions hain, lekin sirf yahi distributions nahi hain:

- ParrotOS
- Raspberry Pi OS
- BlackArch
- CentOS
- Ubuntu
- Debian
- BackBox
- Pentoo

Alag-alag Linux distributions ke beech main differences unmein included packages, user interface aur available tools hote hain.

Kali Linux cybersecurity specialists ke liye sabse popular distribution hai. Ismein security se related bahut saare tools aur packages available hote hain.

Ubuntu desktop users ke beech bahut popular hai, jabki Debian servers aur embedded systems ke liye popular hai.

Aakhir mein, Red Hat Enterprise Linux aur CentOS enterprise-level computing ke liye popular hain.

Debian

Debian ek widely used aur well-respected Linux distribution hai, jo apni stability aur reliability ke liye jaana jata hai.

Iska use alag-alag purposes ke liye kiya jata hai, jaise desktop computing, servers aur embedded systems.

Ye software updates aur security patches ko manage karne ke liye Advanced Package Tool (**apt**) package management system ka use karta hai.

Package management system system ko updated aur secure rakhne mein help karta hai. Ye security updates available hote hi automatically download aur install kar deta hai.

Is process ko manually execute kiya ja sakta hai ya automatically set up kiya ja sakta hai.

Debian ko seekhna doosre Linux distributions ke comparison mein thoda difficult ho sakta hai, lekin ise widely ek flexible aur customizable Linux distro maana jata hai.

Iski configuration aur setup thodi complex ho sakti hai, lekin ye humein apne system par bahut achha control provide karta hai, jo advanced users ke liye useful ho sakta hai.

Linux system par jitna zyada control hamare paas hota hai, system utna hi complex feel hone lagta hai.

Lekin doosre options aur possibilities ke comparison mein ye sirf aisa feel hota hai.

Agar hum Linux ko required depth ke saath nahi seekhte hain, toh ho sakta hai ki hum simple tasks aur processes ko configure karne mein zyada time laga dein, jabki kuch commands aur tools ko achhe se seekhkar hum ye kaam kam time mein kar sakte hain.

Hum ise **Filter Contents** aur **Find Files and Directories** sections mein dekhenge.

Stability aur reliability Debian ki sabse badi strengths hain.

Debian apne long-term support releases ke liye jaana jata hai, jo 5 saal tak updates aur security patches provide kar sakte hain.

Ye servers aur doosre systems ke liye bahut important ho sakta hai, jinhe 24/7 chalu rehna zaroori hota hai.

Debian mein kuch vulnerabilities bhi saamne aayi hain, lekin iski development community ne jaldi hi unke patches aur security updates release kiye hain.

Iske alawa, Debian security aur privacy ko lekar strong commitment rakhta hai aur iska security record bhi kaafi achha raha hai.

Debian ek versatile aur reliable Linux distribution hai, jiska use alag-alag purposes ke liye widely kiya jata hai.

Iski stability, reliability aur security ke prati commitment ise kai tarah ke kaamon ke liye ek achha choice banati hai, jisme cybersecurity bhi shamil hai.

Introduction to Shell

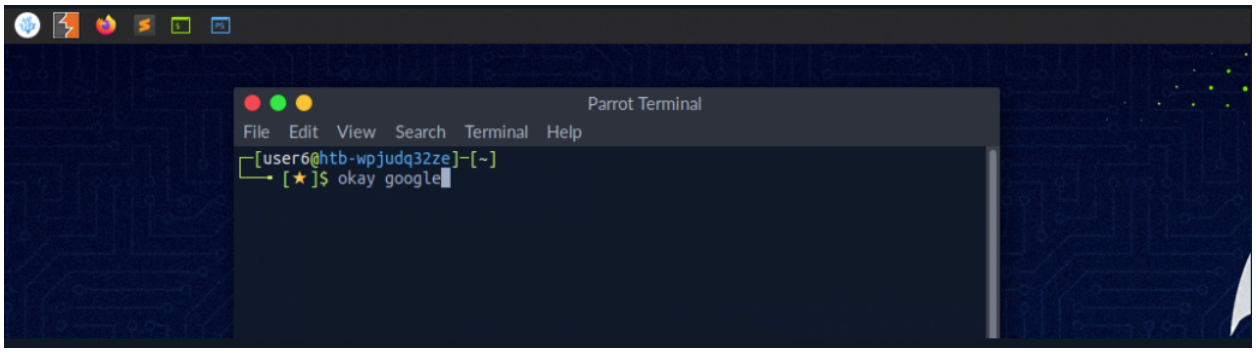
Linux Shell ko use karna seekhna bahut zaroori hai, kyunki bahut saare servers Linux par based hote hain.

In servers ka use aksar isliye kiya jata hai, kyunki Windows servers ke comparison mein Linux mein errors hone ki possibility kam hoti hai.

Example ke liye, web servers aksar Linux par based hote hain.

Operating system ko effectively control karne ke liye humein use karna aana chahiye. Iske liye Linux ke ek important part Shell ko samajhna aur usmein achhi knowledge hasil karna zaroori hai.

Jab hum pehli baar Windows se Linux par switch hue the, toh kya ye kuch aisa dikhta tha:



Linux terminal, jise **Shell** ya Command Line bhi kaha jata hai, ek text-based input/output (I/O) interface provide karta hai. Ye computer system mein user aur Kernel ke beech communication ka kaam karta hai.

Console shabd ka use bhi aam taur par hota hai, lekin iska matlab window nahi, balki text mode wali screen hoti hai.

Terminal window mein hum system ko control karne ke liye commands execute kar sakte hain.

Hum Shell ko ek text-based GUI ki tarah samajh sakte hain, jismein hum commands enter karke alag-alag kaam karte hain, jaise:

- Ek directory se doosri directory mein jana.
- Files ke saath kaam karna.
- System se information prapt karna.

Lekin Shell mein normal GUI ke comparison mein bahut zyada capabilities hoti hain

---

Terminal Emulators

Terminal Emulation ek software hai jo Terminal ke kaam karne ke tareeke ki nakal (emulate) karta hai.

Yeh Graphical User Interface (**GUI**) ke andar text-based programs ko use karne ki suvidha deta hai.

Kuch Command-Line Interfaces (**CLI**) bhi hote hain, jo ek terminal ke andar additional terminals ki tarah chalte hain.

Short mein: Terminal, Shell Interpreter tak pahunchne ke liye ek interface ka kaam karta hai.

Socho tum ek bade office building mein ho, jahan Shell ek main server room ki tarah hai, jo company ke saare data aur commands ko process karta hai.

Terminal ek Receptionist ki desk ki tarah hai, jo Server Room se communication karne ka ek madhyam hai.

Tum Receptionist (Terminal) ke paas jaakar instructions ya requests dete ho, aur woh unhe Server Room (Shell) tak pahunchata hai.

Ab maan lo tum office se door rehkar kaam kar rahe ho.

Terminal Emulation Software tumhare computer screen (GUI) par ek virtual Receptionist Desk ki tarah kaam karta hai. Iski madad se tum office mein physically present hue bina Server Room se communicate kar sakte ho.

Yeh asli Receptionist Desk ki tarah kaam karta hai aur tumhe graphical environment ke andar text-based programs aur commands use karne ki suvidha deta hai.

Iske alawa, **Command-Line Interfaces** (**CLI**), jo ek Terminal ke andar additional Terminals ki tarah chalte hain, aise hain jaise tumhari screen par ek saath kai virtual Receptionist Desks khule hon.

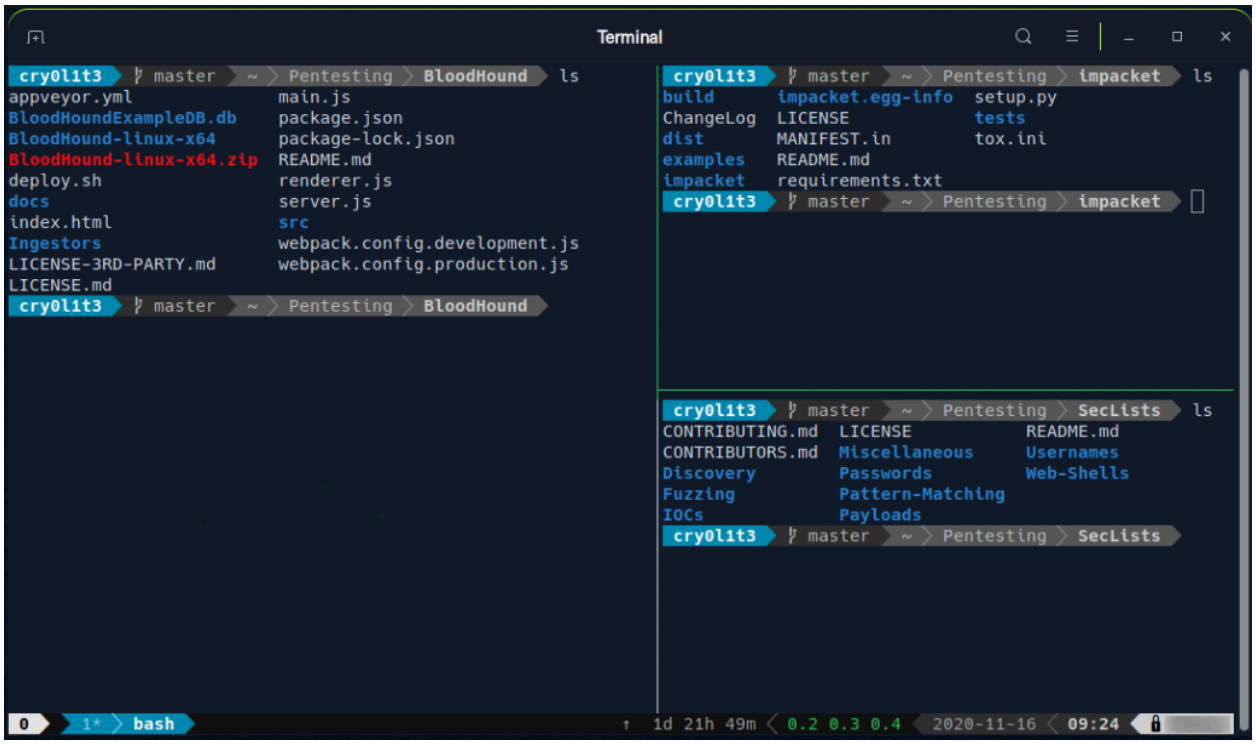
Har ek Desk tumhe alag-alag instructions Server Room tak bhejne ki permission deta hai, lekin sabhi ek hi main interface ke through kaam karte hain.

Asal mein, Terminal ek gateway ki tarah kaam karta hai, jiske through tum Shell ke dwara manage kiye jaane wale main operations se communicate aur unhe control kar sakte ho.

Terminal Emulators aur Multiplexers, Terminal ke useful extensions hain.

Ye humein Terminal ke saath kaam karne ke alag-alag tareeke aur functions provide karte hain, jaise Terminal ko ek window mein divide karna, multiple directories mein kaam karna, alag-alag workspaces banana aur bhi bahut kuch.

Tmux naam ke ek Multiplexer ka use kuch is tarah dikh sakta hai:



Shell

Linux mein sabse zyada use hone wala Shell **Bourne-Again Shell** (**BASH**) hai, jo GNU Project ka ek hissa hai.

GUI ke through hum jo bhi kaam karte hain, woh sab Shell ke through bhi kar sakte hain.

Shell humein Programs aur Processes ke saath interact karne ke bahut saare options deta hai, jisse hum information jaldi hasil kar sakte hain.

Iske alawa, chhoti ya badi Scripts ki madad se kai processes ko aasani se automate kiya ja sakta hai, jisse manual kaam bahut aasan ho jata hai.

Bash ke alawa bhi kai doosre Shells hote hain, jaise:

- **Tcsh/Csh**
- **Ksh**
- **Zsh**
- **Fish**

Prompt Description

Bash Prompt ko samajhna aasan hai. Default रूप से, yeh kuch information dikhata hai, jaise tumhara username (tum kaun ho), tumhare computer ka naam (hostname), aur woh folder/directory jismein tum abhi kaam kar rahe ho.

Yeh screen par dikhne wali text ki ek line hoti hai, jo batati hai ki system tumhare command dene ke liye ready hai.

Prompt ek nayi line par dikhai deta hai aur cursor (blink karne wali line ya box) uske turant baad hota hai, jo tumhare command type karne ka intezar karta hai.

User ko useful information dene ke liye Prompt ko customize bhi kiya ja sakta hai.

Iska format kuch is tarah dikh sakta hai:

```
<username>@<hostname><current working directory>$
```

Kisi user ki Home Directory ko tilde **(~)** symbol se dikhaya jata hai. Jab hum login karte hain, toh by default hum isi folder mein hote hain.

```
<username>@<hostname>[~]$
```

Dollar sign ($) yahan normal user ko represent karta hai. Jaise hi hum root user ke roop mein login karte hain, toh yeh symbol badalkar hash (#) ho jata hai aur kuch is tarah dikhta hai:

```
root@htb[/htb]#
```

Example ke liye, jab hum target system par Shell upload karke run karte hain, toh ho sakta hai humein username, hostname aur current working directory dikhai na de.

Aisa environment ke andar PS1 variable sahi tarah set na hone ki wajah se ho sakta hai.

Is situation mein humein neeche diye gaye Prompts dikhai denge:

Unprivileged - User Shell Prompt

```
$
```

Privileged - Root Shell Prompt

```
#
```

Linux systems mein **PS1** variable control karta hai ki Terminal mein tumhara Command Prompt kaisa dikhega.

Yeh ek template ki tarah hota hai, jo define karta hai ki jab bhi system command type karne ke liye ready hota hai, toh screen par kaunsa text dikhai dega.

PS1 variable ko customize karke tum Prompt mein apna username, computer ka naam, current folder aur colors ya special characters bhi dikha sakte ho.

Isse tum Command-Line Interface ko apni zaroorat ke hisaab se customize kar sakte ho, taaki woh zyada useful aur dekhne mein achha lage.

Username aur current folder jaisi basic information ke alawa, tum Command Prompt mein IP address, date, time aur last command successful hui ya fail, yeh bhi dikha sakte ho.

Yeh customization Penetration Testing ke dauraan bahut useful hoti hai, kyunki isse tum apne actions ko zyada achhe se track kar sakte ho.

Example ke liye, tum Prompt ko is tarah set kar sakte ho ki woh current working directory ka sirf naam dikhane ke bajay uska poora path dikhaye. Zaroorat padne par target ka IP address bhi add kar sakte ho.

script jaise tools ka use karke ya **.bash_history** file ko check karke (jo user ki Home Directory mein hoti hai), tum apne use kiye gaye sabhi commands ko record kar sakte ho aur unhe date aur time ke hisaab se organize kar sakte ho.

Isse Documentation aur Analysis karne mein madad milti hai.

Prompt ko Shell ki configuration file (**Bash** shell ke liye **.bashrc**) mein special characters aur variables ka use karke customize kiya ja sakta hai.

Example ke liye:

- **\u** ka use current username dikhane ke liye hota hai.
- **\h** ka use hostname dikhane ke liye hota hai.
- **\w** ka use current working directory dikhane ke liye hota hai.

Special Character	**Description (मतलब)**
\d	Date (Mon Feb 6)
\D{%Y-%m-%d}	Date (YYYY-MM-DD)
\H	Full hostname (computer ka poora naam)
\j	Shell dwara manage kiye ja rahe jobs ki sankhya
\n	Newline (nayi line)
\r	Carriage return
\s	Shell ka naam
\t	Current time 24-hour format (HH:MM:SS)
\T	Current time 12-hour format (HH:MM:SS)
\@	Current time

\u	Current username
\w	Current working directory ka poora path

Prompt ko customize karna Terminal ko apni zaroorat ke hisaab se personalize aur efficient banane ka ek useful tareeka hai.

Yeh Troubleshooting aur Problem-Solving mein bhi madad karta hai, kyunki yeh kisi bhi samay system ki important information aur uski current condition dikha sakta hai.

Prompt ko customize karne ke alawa, hum apne Terminal Environment ko alag-alag color schemes, fonts aur doosri settings ke saath bhi customize kar sakte hain, taaki kaam karne ka environment dekhne mein achha aur use karne mein aasan ho.

Lekin yahan bhi hum Windows GUI ki tarah hi dekhte hain. Hum ek specific naam wale computer par user ke roop mein login hote hain aur jab hum system mein navigate karte hain, toh humein pata hota hai ki hum kis directory mein hain.

Bash Prompt ko bhi apni zaroorat ke hisaab se customize aur change kiya ja sakta hai.

Bash Prompt ko customize karna is module ke scope se bahar hai. Lekin hum **bash-prompt-generator** aur **Powerline** ko dekh sakte hain, jo humein apni zaroorat ke hisaab se Prompt ko customize karne ki suvidha dete hain.

Getting Help

Linux ki structure, uske alag-alag Distributions aur Shell ke purpose ko achhe se samajhne ke baad, ab hum is knowledge ko practically use karne ke liye ready hain.

Ab hum Terminal mein directly Commands ka use karna seekhenge. Saath hi, jab humein koi unknown Command milegi, toh uski help kaise leni hai, yeh bhi seekhenge.

Humein aksar aise Tools milenge jinke Optional Parameters humein yaad nahi honge, ya aise Tools milenge jinhe humne pehle kabhi nahi dekha hoga.

Isliye yeh jaanna bahut zaroori hai ki hum un Tools ke baare mein khud information kaise hasil kar sakte hain.

Iske pehle do tareeke hain:

1. Man Pages
2. Help Functions

Kisi bhi Tool ko use karne se pehle uske baare mein achhe se jaankari lena hamesha achha hota hai.

Hum kuch aisi Tricks bhi seekhenge, jinki madad se hum un Tools ke saath aise kaam kar sakenge jo humein pehle possible nahi lagte the.

Man Pages mein humein Detailed Manuals milte hain, jinmein Tools ke baare mein poori jaankari aur detail mein explanations di hoti hain.

First Command

```
Toxicx01@htb[/htb]$ ls

cacert.der  Documents  Music     Public     Videos
Desktop     Downloads  Pictures  Templates
```

Linux aur Unix systems mein **ls** command ka use current folder ya kisi specified directory ke andar maujood files aur directories ki list dekhne ke liye kiya jata hai.

Iski madad se hum dekh sakte hain ki kisi folder ke andar kya-kya hai aur files ko zyada achhe se manage kar sakte hain.

Linux ki doosri commands ki tarah, **ls** command mein bhi additional options aur features hote hain, jinki madad se hum output ko filter ya format kar sakte hain, taaki humein wahi information dikhe jo hum chahte hain.

Kisi Tool ya Command ke available options jaanne ke liye help lene ke kai tareeke hote hain.

Unmein se ek tareeka **man** command ka use karna hai. Yeh commands ke Manual Pages dikhata hai aur unke use karne ke baare mein detailed information deta hai.

Syntax:

```
Toxicx01@htb[/htb]$ man <tool>
```

Let us have a look at an example and get help for the ls command:

Example:

```
Toxicx01@htb[/htb]$ man ls
```

```
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List  information  about  the FILEs (the current directory by default).
       Sort entries alphabetically if none of -cftuvSUX nor --sort  is  speci-
       fied.

       Mandatory  arguments  to  long  options are mandatory for short options
       too.

       -a, --all
              do not ignore entries starting with .

       -A, --almost-all
              do not list implied . and ..

       --author
 Manual page ls(1) line 1 (press h for help or q to quit)
```

Kuch examples dekhne ke baad, hum complete documentation ko padhe bina bhi optional parameters ko jaldi se dekh sakte hain.

Aisa karne ke hamare paas kai tareeke hain.

Syntax:

```
Toxicx01@htb[/htb]$ man <tool>
```

Example:

```shellsession
Toxicx01@htb[/htb]$ ls --help

Usage: ls [OPTION]... [FILE]...
List information about the FILEs (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.
  -a, --all                  do not ignore entries starting with .
  -A, --almost-all           do not list implied . and ..
      --author               with -l, print the author of each file
  -b, --escape               print C-style escapes for nongraphic characters
      --block-size=SIZE      with -l, scale sizes by SIZE when printing them;
                             e.g., '--block-size=M'; see SIZE format below

  -B, --ignore-backups       do not list implied entries ending with ~
  -c                         with -lt: sort by, and show, ctime (time of last
                             modification of file status information);
                             with -l: show ctime and sort by name;
                             otherwise: sort by ctime, newest first

  -C                         list entries by columns
<SNIP>
```

Kuch tools ya commands, jaise curl, --help ki jagah -h ka use karke help ka short version provide karte hain.

Syntax:

```
Toxicx01@htb[/htb]$ <tool> -h
```

Example:

```shellsession
Toxicx01@htb[/htb]$ curl -h

Usage: curl [options...] <url>
     --abstract-unix-socket <path> Connect via abstract Unix domain socket
     --anyauth       Pick any authentication method
 -a, --append        Append to target file when uploading
     --basic         Use HTTP Basic Authentication
     --cacert <file> CA certificate to verify peer against
     --capath <dir>  CA directory to verify peer against
 -E, --cert <certificate[:password]> Client certificate file and password
<SNIP>
```

Jaise hum dekh sakte hain, is example mein dono ke results mein koi difference nahi hai.

Ek aur tool jo shuruaat mein useful ho sakta hai, woh hai apropos.

Har manual page ke andar ek chhota description diya hota hai. Ye tool diye gaye keyword ko un descriptions mein search karta hai.

Syntax:

```
Toxicx01@htb[/htb]$ apropos <keyword>
```

Example:

```
Toxicx01@htb[/htb]$ apropos sudo

sudo (8)              - execute a command as another user
sudo.conf (5)         - configuration for sudo front end
sudo_plugin (8)       - Sudo Plugin API
sudo_root (8)         - How to run administrative commands
sudoedit (8)          - execute a command as another user
sudoers (5)           - default sudo security policy plugin
sudoreplay (8)        - replay sudo session logs
visudo (8)            - edit the sudoers file
```

Agar humein kisi lambe command ko samajhne mein problem hoti hai, toh help lene ke liye ek aur useful website hai:

https://explainshell.com/

Next, hum bahut saare commands ke baare mein padhenge, jinmein se kai commands aapke liye naye ho sakte hain.

Lekin ab aap jaan chuke hain ki kisi bhi aise command ke baare mein help kaise leni hai, jisse aap unfamiliar hain ya jiske options ko lekar aap sure nahi hain.

Saath hi, hum aapko encourage karte hain ki aap apni curiosity ke according naye tools ko explore karein.

Jitna time chahiye utna time lekar diye gaye tools ke saath practice aur experiment karein.

Ye time hamesha aapke liye useful rahega.

System Information

Ab, terminal aur shell ka use karne ki practical practice shuru karte hain, taaki aap inka use karne mein comfortable ho sakein.

Dhyan rakhein ki zaroorat padne par aap help lene ke liye hamesha -h, --help, ya man commands ka use kar sakte hain.

Kyunki hum alag-alag Linux systems par kaam karenge, isliye unki structure ko samajhna zaroori hai. Ismein system details, processes, network configurations, users/user settings aur directories ke saath unse jude parameters ko samajhna shamil hai.

Neeche kuch zaroori tools ki list di gayi hai, jo ye information collect karne mein madad karenge.

Inmein se zyadaatar tools pehle se hi installed hote hain.

Ye knowledge sirf roz ke Linux tasks ke liye hi zaroori nahi hai, balki Linux operating systems mein security configurations check karne, vulnerabilities (kamzoriyan) identify karne aur possible security risks ko rokne mein bhi important role play karti hai.

Command	Description
whoami	Current user ka username dikhata hai.
id	User ki identity (pehchaan) ki information deta hai.
hostname	Current host system ka naam set ya display karta hai.
uname	Operating system ke naam aur system hardware ki basic information dikhata hai.
pwd	Current working directory (aap abhi kis folder mein hain) ka naam batata hai.
ifconfig	Network interface ka IP address dekhne, set karne aur uski network settings configure karne ke kaam aata hai.
ip	Routing, network devices, interfaces aur tunnels ko dekhne ya manage karne ke kaam aata hai.
netstat	Network ki current status information dikhata hai.
ss	Network sockets ki information check karne ke kaam aata hai.
ps	System mein chal rahe processes (programs) ki status dikhata hai.
who	Kaun-kaun se users system mein login hain, ye dikhata hai.

env	Environment variables ko display karta hai ya unhe set karke command execute karta hai.
lsblk	Block devices (jaise HDD, SSD aur USB storage) ki list dikhata hai.
lsusb	System se connected USB devices ki list dikhata hai.
lsof	System mein open files ki list dikhata hai.
lspci	System se connected PCI devices ki list dikhata hai.

Logging In via SSH

Secure Shell (SSH) ek protocol hai, jo clients ko remote computers access karne aur un par commands ya actions execute karne ki permission deta hai.

Linux-based hosts aur servers ke saath-saath doosre Unix-like operating systems mein bhi SSH ek standard tool hai, jo pehle se installed hota hai.

Bahut saare administrators remote access ke through computer ko configure aur maintain karne ke liye SSH ka use karna pasand karte hain.

SSH ek purana aur achhe se tested protocol hai, jise use karne ke liye GUI (Graphical User Interface) ki zaroorat nahi hoti aur na hi ye GUI provide karta hai.

Isi wajah se ye bahut efficiently kaam karta hai aur bahut kam system resources use karta hai.

Hum aage ke sections aur module ke zyadaatar lab exercises mein isi tarah ke connection ka use karenge, taaki aap seekhi hui commands aur actions ko ek safe environment mein try kar sakein.

Hum apne target machines se connect hone ke liye neeche di gayi command ka use kar sakte hain:

```
Toxicx01@htb[/htb]$ ssh htb-student@[IP address]
```

Ab, chaliye us machine par kuch examples dekhte hain, jismein humne abhi login kiya hai.

Hostname

hostname command ko samajhna kaafi aasan hai. Ye bas us computer ka naam dikhata hai, jismein hum abhi login hain.

```
Toxicx01@htb[/htb]$ hostname

nixfund
```

Whoami

Is simple aur quick command ka use Windows aur Linux dono systems mein apna current username pata karne ke liye kiya ja sakta hai.

Security assessment ke dauraan, jab humein kisi host par Reverse Shell ka access milta hai, toh sabse pehle humein ye pata karna chahiye ki hum kis user ke roop mein system chala rahe hain.

Iske baad hum pata laga sakte hain ki us user ke paas koi special privileges (khaas permissions) ya access hai ya nahi.

```
cry0l1t3@htb[/htb]$ whoami

cry0l1t3
```

Id

id command, whoami command se thodi zyada information deta hai. Ye hamari current user ki group membership aur IDs ko display karta hai.

Penetration testers ke liye ye command useful hai, kyunki isse pata chalta hai ki kisi user ke paas kya-kya access ho sakta hai. Sysadmins bhi iska use account permissions aur group membership check karne ke liye karte hain.

Is output mein hackthebox group important hai, kyunki ye ek non-standard group hai.

adm group ka matlab hai ki user /var/log directory mein log files ko read kar sakta hai aur iske through sensitive information tak pahunch sakta hai.

sudo group khaas taur par important hai, kyunki iska matlab hai ki hamara user kuch ya sabhi commands powerful root user ke roop mein run kar sakta hai.

Sudo rights ki madad se hum apne privileges ko badha sakte hain (Privilege Escalation).

Ye sysadmin ke liye bhi ek signal ho sakta hai ki use permissions aur group memberships ko check karna chahiye aur user se woh access hata dena chahiye jiski use apne daily tasks karne ke liye zaroorat nahi hai.

```shellsession
cry0l1t3@htb[/htb]$ id

uid=1000(cry0l1t3) gid=1000(cry0l1t3)
groups=1000(cry0l1t3),1337(hackthebox),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev
```

Uname

Chaliye, ab uname command ko thoda aur detail mein samajhte hain.

Agar hum apne terminal mein man uname type karte hain, toh uname command ka man page (manual page) open ho jayega.

Ismein humein pata chalega ki is command ke saath kaun-kaun se options use kar sakte hain aur un options ko use karne par kya results milte hain.

NAME

uname - print system information

SYNOPSIS

uname [OPTION]...

DESCRIPTION

Print certain system information. With no OPTION, same as -s.

-a, --all
print all information, in the following order, except omit -p and -i if unknown:

-s, --kernel-name
print the kernel name

-n, --nodename
print the network node hostname

-r, --kernel-release
print the kernel release

-v, --kernel-version
print the kernel version

-m, --machine
print the machine hardware name

-p, --processor
print the processor type (non-portable)

-i, --hardware-platform
print the hardware platform (non-portable)

-o, --operating-system

uname -a command chalane se machine ki saari information ek specific order mein print hogi: Kernel name, Hostname, Kernel release, Kernel version, Machine hardware name aur Operating system.

-a flag, agar unknown hon, toh -p (Processor type) aur -i (Hardware platform) ki information ko show nahi karega.

```shellsession
cry0l1t3@htb[/htb]$ uname -a

Linux box 4.15.0-99-generic #100-Ubuntu SMP Wed Apr 22 20:32:56 UTC 2020 x86_64
x86_64 x86_64 GNU/Linux
```

Upar diye gaye command se hum dekh sakte hain ki kernel name **Linux** hai, hostname **box** hai, kernel release **4.15.0-99-generic** hai, kernel version **#100-Ubuntu SMP Wed Apr 22 20:32:56 UTC 2020** hai, aur isi tarah baaki information bhi milti hai.

Inmein se kisi bhi option ko alag se run karne par humein uski specific information mil jayegi, jise hum dekhna chahte hain.

Kernel Release Pata Karne Ke Liye Uname

Maan lo hum kernel release ko print karna chahte hain, taaki jaldi se potential kernel exploits (kernel ki kamzori ka fayda uthane wale programs) search kar sakein.

Is information ko paane ke liye hum **uname -r** type kar sakte hain.

```shellsession
cry0l1t3@htb[/htb]$ uname -r

4.15.0-99-generic
```

Is information ki madad se hum Google par "4.15.0-99-generic exploit" search kar sakte hain. Search karne par pehla result hi hamare liye useful dikhai deta hai.

Commands ko achhe se padhna aur samajhna bahut recommended hai ki ye kis kaam aate hain aur inse kya information mil sakti hai.

Bhale hi ye kaam thoda boring ho, lekin common commands ke manpages padhne se hum bahut kuch seekh sakte hain.

Ho sakta hai humein kuch aisi cheezon ke baare mein bhi pata chale, jinke baare mein humein pehle nahi pata tha ki kisi command se ye kaam bhi kiya ja sakta hai.

Ye information sirf Linux par kaam karne ke liye hi use nahi hoti, balki baad mein Linux system mein vulnerabilities (kamzoriyaan) aur misconfigurations (galat settings) dhoondhne ke liye bhi use hoti hai.

Ye kamzoriyaan Privilege Escalation mein madad kar sakti hain.

Neeche kuch optional exercises diye gaye hain, jinhe hum practice ke liye solve kar sakte hain. Inse humein alag-alag commands se familiar hone mein madad milegi.

Linux Exercises Ke Baare Mein

Linux OS aur uske commands ko padhne ke liye diye gaye exercises mein hamesha ye turant samajh nahi aayega ki aapko kya karna hai, aur ye bilkul normal hai. Aisa hona swabhavik hai.

Jaise aapne Learning Process module mein seekha hai, kuch naya seekhna kabhi-kabhi uncomfortable lag sakta hai aur stress bhi ho sakta hai.

Aap ise is tarah samajh sakte hain: jab aapne pehli baar car chalana seekha tha aur akele drive karni padi thi. Us samay stress hua tha, kyunki aapko ek saath bahut saari cheezon par dhyan dena tha.

Lekin ab experience ke saath car chalana aasan ho gaya hai, haalaanki ab aap utna naya nahi seekh rahe hain.

Isi tarah, is module mein bhi aapko kabhi-kabhi aisi situations milengi, jahan aapko samajh nahi aayega ki kya karna hai. Lekin ye bilkul theek hai.

Apni cybersecurity journey mein aapko aise moments baar-baar milenge. Ye ek positive sign hai ki aap kuch naya seekh rahe hain.

In challenges ko paar karne se aapki skills improve hoti hain, chahe aapne exercise ko abhi poori tarah solve na bhi kiya ho.

Yahi sabse important goal hai — seekhte hue aage badhna (Progress through Learning).

Exercises ko jaanbujhkar is tarah design kiya gaya hai ki dheere-dheere aap apni current knowledge se aage badhein aur aisi cheezein seekhein jo aapke liye bilkul nayi hain.

Ye progression jaanbujhkar rakha gaya hai, taaki practice karte rehne se aapka experience aur knowledge naturally badhta rahe.

Kabhi-kabhi ye process uncomfortable lag sakta hai, lekin growth ke liye ye zaroori hai.

Har naye challenge ke saath aap apni existing knowledge se aage badhenge. Aur lagataar practice karne se aapki understanding aur skills dheere-dheere improve hoti jayengi.

Practice karte rahiye. Dheere-dheere aap unknown situations ko handle karne mein zyada confident aur capable ban jayenge.

Q. Find out the machine hardware name and submit it as the answer.

Ans.

Q. What is the path to htb-student's home directory?

Ans.

Q. What is the path to the htb-student's mail?
Ans.

Q. Which shell is specified for the htb-student user?
Ans.

Q. Which kernel release is installed on the system? (Format: 1.22.3)
Ans.

Q. What is the name of the network interface that MTU is set to 1500?
Ans.

Navigation

Navigation bahut zaroori hai, bilkul waise hi jaise ek normal Windows user mouse ka use karta hai.

Navigation ki madad se hum apne system mein idhar-udhar ja sakte hain aur un directories aur files ke saath kaam kar sakte hain, jinki humein zaroorat hoti hai.

Iske liye hum alag-alag commands aur tools ka use karte hain, taaki kisi directory ya file ki information dekh sakein. Hum advanced options ka use karke output ko apni zaroorat ke hisaab se behtar bana sakte hain.

Kuch naya seekhne ka sabse achha tareeka hai uske saath experiment karna.

Yahan hum Linux mein navigation karna, files aur folders ko create, move, edit aur delete karna seekhenge. Saath hi, operating system mein files ko dhoondhna, alag-alag tarah ke redirects aur file descriptors ke baare mein bhi seekhenge.

Hum kuch shortcuts bhi seekhenge, jo shell mein hamare kaam ko aur aasan aur comfortable bana denge.

Hum recommend karte hain ki aap apni locally hosted VM mein in sabhi cheezon ki practice karein.

Dhyan rakhein ki apni VM ka snapshot zaroor bana lein, taaki agar galti se aapka system damage ho jaye, toh aap use pehle wali condition mein wapas la sakein.

Chaliye navigation se shuru karte hain. System mein idhar-udhar jaane se pehle humein pata karna hoga ki hum kis directory mein hain.

Hum **pwd** command se pata laga sakte hain ki hum abhi kis directory mein hain.

```
cry0l1t3@htb[~]$ pwd

/home/cry0l1t3
```

Bina kisi additional option ke iska use karne par ye sirf directories aur files ko dikhayega.

Lekin hum **-l** option bhi add kar sakte hain, jisse un directories aur files ke baare mein zyada information dikhai degi.

```
cry0l1t3@htb[~]$ ls -l

total 32
drwxr-xr-x 2 cry0l1t3 htbacademy 4096 Nov 13 17:37 Desktop
drwxr-xr-x 2 cry0l1t3 htbacademy 4096 Nov 13 17:34 Documents
drwxr-xr-x 3 cry0l1t3 htbacademy 4096 Nov 15 03:26 Downloads
drwxr-xr-x 2 cry0l1t3 htbacademy 4096 Nov 13 17:34 Music
drwxr-xr-x 2 cry0l1t3 htbacademy 4096 Nov 13 17:34 Pictures
drwxr-xr-x 2 cry0l1t3 htbacademy 4096 Nov 13 17:34 Public
drwxr-xr-x 2 cry0l1t3 htbacademy 4096 Nov 13 17:34 Templates
drwxr-xr-x 2 cry0l1t3 htbacademy 4096 Nov 13 17:34 Videos
```

Sabse pehle, hum current directory mein listed files aur directories ke dwara use kiye gaye total blocks (1024-byte) dekhte hain. Ye batata hai ki total kitni disk space use hui hai.

Iska matlab hai:

32 blocks × 1024 bytes/block = 32,768 bytes (yaani 32 KB) disk space use hui hai.

Iske baad, hum kuch columns dekhte hain, jo is tarah arranged hain:

Column Content	Description
`drwxr-xr-x`	Type and permissions
`2`	Number of hard links to the file/directory
`cry0l1t3`	Owner of the file/directory
`htbacademy`	Group owner of the file/directory
`4096`	Size of the file or the number of blocks used to store the directory information
`Nov 13 17:37`	Date and time
`Desktop`	Directory name

Lekin humein is folder ke andar maujood saari cheezein dikhai nahi dengi.

Ek directory mein hidden files bhi ho sakti hain, jinke naam ki shuruaat mein dot (.) laga hota hai (jaise **.bashrc** ya **.bash_history**).

Isliye, humein **ls -la** command ka use karna hoga, taaki directory ki saari files (List All) dekh sakein.

```
cry0l1t3@htb[~]$ ls -la

total 403188
drwxr-xr-x 2 cry0l1t3 htbacademy 4096 Nov 13 17:37 .bash_history
drwxr-xr-x 2 cry0l1t3 htbacademy 4096 Nov 13 17:37 .bashrc
...SNIP...
```

Kisi directory ke contents (andar ki files aur folders) dekhne ke liye zaroori nahi hai ki hum pehle us directory mein jaayein.

Hum **ls** command ka use karke us directory ka path bhi de sakte hain, jiske contents hum dekhna chahte hain.

```
cry0l1t3@htb[~]$ ls -l /var/

total 52
drwxr-xr-x  2 root root     4096 Mai 15 18:54 backups
drwxr-xr-x 18 root root     4096 Nov 15 16:55 cache
```

Hum directory mein jaane ke liye bhi yahi kaam kar sakte hain. Directories ke andar move karne ke liye hum **cd** command ka use karte hain.

Chalo, **/dev/shm** directory mein jaate hain.

Hum pehle **/dev** directory mein jaa sakte hain aur phir **/shm** mein. Lekin hum poora path ek saath bhi enter kar sakte hain aur seedhe wahan pahunch sakte hain.

```
cry0l1t3@htb[~]$ cd /dev/shm

cry0l1t3@htb[/dev/shm]$
```

Kyunki hum pehle Home directory mein the, isliye hum jaldi se us directory mein wapas ja sakte hain, jismein hum sabse aakhir mein the.

```
cry0l1t3@htb[/dev/shm]$ cd -

cry0l1t3@htb[~]$
```

Shell humein Auto-complete function bhi provide karta hai, jo navigation ko aasan banata hai.

Agar hum ab **cd /dev/s** type karein aur **[TAB]** key ko do baar press karein, toh humein **/dev/** directory ke andar s letter se shuru hone wali sabhi entries dikhai dengi.

```
cry0l1t3@htb[~]$ cd /dev/s [TAB 2x]

shm/ snd/
```

Agar hum letter **s** ke saath letter **h** jod dein, toh Shell input ko automatically complete kar dega, kyunki is directory mein **sh** letters se shuru hone wala koi doosra folder nahi hai.

Ab agar hum directory ke saare contents display karein, toh humein sirf neeche diye gaye contents dikhai denge.

```
cry0l1t3@htb[/dev/shm]$ ls -la /dev/shm

total 0
drwxrwxrwt  2 root root   40 Mai 15 18:31 .
drwxr-xr-x 17 root root 4000 Mai 14 20:45 ..
```

Pehli entry mein ek single dot (.) hai, jo us current directory ko show karta hai jismein hum abhi maujood hain.

Doosri entry mein two dots (..) hain, jo parent directory **/dev** ko represent karte hain.

Iska matlab hai ki hum neeche di gayi command ka use karke parent directory mein jaa sakte hain.

```
cry0l1t3@htb[/dev/shm]$ cd ..

cry0l1t3@htb[/dev]$
```

Kyunki hamara shell kuch records se bhar gaya hai, isliye hum **clear** command ka use karke shell ko saaf kar sakte hain.

Lekin sabse pehle, chalo dobara **/dev/shm** directory mein wapas chalte hain. Uske baad apne terminal ko saaf karne ke liye **clear** command execute karte hain.

```
cry0l1t3@htb[/dev]$ cd shm && clear
```

Apne terminal ko saaf karne ka ek aur tarika hai shortcut [Ctrl] + [L] ka use karna.

Hum arrow keys (↑ ya ↓) ka use karke command history mein aage-peeche ja sakte hain. Isse humein pehle use ki gayi commands dikhengi.

Lekin hum shortcut **[Ctrl] + [R]** ka use karke bhi command history mein search kar sakte hain aur us text ko type kar sakte hain jise hum dhoondh rahe hain.

Q. What is the name of the hidden "history" file in the htb-user's home directory?
Ans.

Q. What is the index number of the "sudoers" file in the "/etc" directory?
Ans.

Files aur Directories ke saath kaam karna

Linux mein files ke saath kaam karne aur Windows mein files ke saath kaam karne ke beech sabse bada difference ye hai ki hum un files ko kaise access aur manage karte hain.

Windows mein hum files ko dhoondhne, open karne aur edit karne ke liye aam taur par Explorer jaise graphical tools ka use karte hain.

Lekin Linux mein terminal ek powerful alternative deta hai, jahan hum commands ka use karke directly files ko access aur edit kar sakte hain.

Ye tarika sirf fast hi nahi, balki zyada efficient bhi hai, kyunki ismein hum vim ya nano jaise editors ki zaroorat ke bina bhi files ko interactively edit kar sakte hain.

Terminal ki efficiency ka reason ye hai ki:

- Hum sirf kuch commands ka use karke files ko access kar sakte hain.
- Regular Expressions (**regex**) ka use karke files mein chuninda changes kar sakte hain.
- Hum ek saath multiple commands run kar sakte hain.
- Commands ke output ko files mein save kar sakte hain.
- Bahut saari files ko ek saath automatically edit karne ka kaam kar sakte hain.

Ye sab kaam karne se bahut saara time bachta hai, khaaskar jab hum ek saath bahut saari files par kaam kar rahe hote hain.

Command-line ka ye tarika hamare workflow ko aasan aur fast banata hai. Isliye ye un kaamon ke liye bahut useful hai jinhe graphical interface se karne mein zyada time lagta hai.

Ab hum apne operating system ke content ko achhe se manage karne ke liye files aur directories ke saath kaam karna seekhenge.

Create, Move and Copy

Chalo, sabse pehle seekhte hain ki files ko create (banana), rename (naam badalna), move (ek jagah se doosri jagah le jaana), copy (nakal banana) aur delete (hatana) kaise karte hain.

Neeche di gayi commands execute karne se pehle, humein SSH ke through target machine se connect hona hoga (iske liye section ke bottom mein diye gaye connection instructions ka use karein).

Ab maan lo ki hum ek nayi file ya directory banana chahte hain. Iske liye syntax kuch is tarah hai:

Syntax :- touch

```
Toxicx01@htb[/htb]$ touch <name>
```

Syntax :- mkdir

```
Toxicx01@htb[/htb]$ mkdir <name>
```

Agla example mein, hum **info.txt** naam ki ek file aur **Storage** naam ki ek directory (folder) banayenge.

Inhe banane ke liye, hum upar diye gaye commands aur unke syntax ka use karenge.

Create an Empty File

```
Toxicx01@htb[/htb]$ touch info.txt
```

Create a Directory

```
Toxicx01@htb[/htb]$ mkdir Storage
```

Jab aap apne system ko organize karte hain, toh kabhi-kabhi ek directory (folder) ke andar multiple directories (folders) banane ki zarurat pad sakti hai.

Har directory ko alag-alag **mkdir** command se banana time-consuming (zyada samay lene wala) ho sakta hai.

Achhi baat ye hai ki **mkdir** command mein **-p** (parents) option hota hai.

Is option ki help se aap automatically parent directories (main folders) bhi create kar sakte hain.

```
Toxicx01@htb[/htb]$ mkdir -p Storage/local/user/documents
```

Parent directories create karne ke baad, hum tree tool ka use karke poore folder structure ko dekh sakte hain.

```
Toxicx01@htb[/htb]$ tree .

.
├── info.txt
└── Storage
    └── local
        └── user
            └── documents

4 directories, 1 file
```

Aap kisi specific directory (folder) ke andar directly file create kar sakte hain. Iske liye aapko us file ka path dena hoga jahan aap file store karna chahte hain.

Aap single dot (.) ka use karke bata sakte hain ki aap current directory (jis folder mein abhi hain) se kaam start karna chahte hain.

Ye current location mein kaam karne ka ek easy tareeka hai, kyunki ismein aapko poora path type karne ki zarurat nahi hoti.

Isliye, ek aur empty file banane ke liye command kuch is tarah hogi:

Create userinfo.txt

```
Toxicx01@htb[/htb]$ touch ./Storage/local/user/userinfo.txt
```

```
Toxicx01@htb[/htb]$ tree .

.
├── info.txt
└── Storage
    └── local
        └── user
            ├── documents
            └── userinfo.txt

4 directories, 2 files
```

mv command ka use karke hum files aur directories ko ek jagah se doosri jagah move (shift) kar sakte hain aur unka naam bhi change (rename) kar sakte hain.

Syntax :- mv

```
Toxicx01@htb[/htb]$ mv <file/directory> <renamed file/directory>
```

Sabse pehle, hum **info.txt** file ka naam badalkar **information.txt** rakhenge.

Uske baad, is file ko **Storage** directory (folder) ke andar move kar denge.

Rename File

```
Toxicx01@htb[/htb]$ mv info.txt information.txt
```

Ab hum current directory mein **readme.txt** naam ki ek file banayenge. Uske baad **information.txt** aur **readme.txt** dono files ko **Storage/** directory mein move kar denge.

Create readme.txt

```
Toxicx01@htb[/htb]$ touch readme.txt
```

Move Files to Specific Directory

```
Toxicx01@htb[/htb]$ mv information.txt readme.txt Storage/
```

```
Toxicx01@htb[/htb]$ tree .

.
└── Storage
    ├── information.txt
    ├── local
    │   └── user
    │       ├── documents
    │       └── userinfo.txt
    └── readme.txt

4 directories, 3 files
```

Maan lijiye hum **readme.txt** ko **local/** directory mein rakhna chahte hain. Toh hum specified paths ka use karke us file ko wahan copy kar sakte hain.

Copy readme.txt

```
Toxicx01@htb[/htb]$ cp Storage/readme.txt Storage/local/
```

Ab hum tree tool ka dobara use karke check kar sakte hain ki file wahan hai ya nahi.

```
Toxicx01@htb[/htb]$ tree .

.
└── Storage
    ├── information.txt
    ├── local
    │   ├── readme.txt
    │   └── user
    │       ├── documents
    │       └── userinfo.txt
    └── readme.txt

4 directories, 4 files
```

Linux mein basic file management commands ke alawa bhi files ke saath kaam karne ke kai powerful tareeke hain, jaise Redirection aur Text Editors ka use karna.

Redirection humein commands aur files ke beech input aur output ke flow ko control karne ki permission deta hai. Isse files create karna ya modify karna jaise kaam jaldi aur aasani se kiye ja sakte hain.

Aap interactive editing ke liye popular text editors jaise vim aur nano ka bhi use kar sakte hain.

Hum in methods ko aage ke sections mein detail mein samjhenge. Jaise-jaise aap in techniques se familiar hote jayenge, waise-waise aap apne system mein files create, edit aur manage karne ke aur bhi flexible tareeke seekhenge.

Optional Exercise:

Humne ab tak jo tools seekhe hain, unka use karke pata lagane ki koshish karein ki files aur directories ko delete kaise kiya jata hai.

Dhyan rakhein ki online research karna learning process ka ek important part hai — ye cheating nahi hai.

Abhi aapka test nahi liya ja raha hai, balki aapki knowledge build ki ja rahi hai.

Online solutions search karne se aapko alag-alag methods aur alternative approaches ke baare mein pata chal sakta hai. Isse aap cheezon ko behtar samajh sakte hain aur problems solve karne ke sabse efficient tareeke bhi discover kar sakte hain.

Q. What is the name of the last modified file in the "/var/backups" directory?
Ans.

Q. What is the inode number of the "shadow.bak" file in the "/var/backups" directory?
Ans.

Editing Files

Files aur directories create karna seekhne ke baad, ab hum in files ke saath kaam karna seekhenge.

Linux mein kisi file ko edit karne ke kai tareeke hain. Iske liye kuch sabse common text editors **Vi** aur **Vim** hain.

Lekin hum shuruaat **Nano** editor se karenge, jo comparatively kam use hota hai, lekin ise samajhna aasaan hai.

Nano ka use karke file create aur edit karne ke liye, editor launch karte waqt file ka naam pehle parameter ke roop mein specify kar sakte hain.

Example:

Maan lijiye humein **notes.txt** naam ki ek nayi file create karke open karni hai, toh hum neeche di gayi command ka use karenge:

```
Toxicx01@htb[/htb]$ nano notes.txt
```

Yeh command **Nano** editor ko open karegi, jisse aap turant **notes.txt** file ko edit karna shuru kar sakte hain.

Nano ka simple interface (jise "pager" bhi kaha gaya hai) ise text files ko jaldi edit karne ke liye ek achha option banata hai, khaaskar jab aap abhi shuruaat kar rahe hain.

Is this conversation helpful so far?

Nano Editor

```shellsession
  GNU nano 2.9.3                                  notes.txt

Here we can type everything we want and make our notes.


^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur
Pos     M-U Undo
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell    ^_ Go To
Line  M-E Redo
```

Neeche hum do lines dekhte hain, jinmein chhote descriptions diye gaye hain.

Caret (**^**) hamari **[CTRL]** key ko represent karta hai.

Example ke liye, agar hum **[CTRL + W]** press karte hain, toh Editor ke bottom mein ek **Search:** line dikhti hai, jahan hum us word ya words ko enter kar sakte hain jinhe hum search karna chahte hain.

Ab agar hum **we** word search karke **[ENTER]** press karte hain, toh Cursor us pehle word par chala jayega jo hamare search kiye gaye word se match karta hai.

```shellsession
GNU nano 2.9.3                                  notes.txt

Here we can type everything we want and make our notes.

Search:   notes
^G Get Help    M-C Case Sens  M-B Backwards  M-J FullJstify ^W Beg of Par  ^Y First
Line  ^P PrevHstory
^C Cancel      M-R Regexp     ^R Replace     ^T Go To Line  ^O End of Par  ^V Last
Line   ^N NextHstory
```

Cursor ke saath next matching word par jaane ke liye, hum dobara **[CTRL + W]** press karenge aur bina koi extra information enter kiye **[ENTER]** press karke confirm kar denge.

```shellsession
GNU nano 2.9.3                                  notes.txt

Here we can type everything we want and make our notes.

Search [we]:
^G Get Help    M-C Case Sens  M-B Backwards  M-J FullJstify ^W Beg of Par  ^Y First
Line  ^P PrevHstory
^C Cancel      M-R Regexp     ^R Replace     ^T Go To Line  ^O End of Par  ^V Last
Line   ^N NextHstory
```

Ab hum **[CTRL + O]** press karke file ko save kar sakte hain aur **[ENTER]** press karke file ke naam ko confirm kar sakte hain.

```shellsession
GNU nano 2.9.3                                  notes.txt

Here we can type everything we want and make our notes.

File Name to Write: notes.txt
^G Get Help    M-C Case Sens  M-B Backwards  M-J FullJstify ^W Beg of Par  ^Y First
Line  ^P PrevHstory
^C Cancel      M-R Regexp     ^R Replace     ^T Go To Line  ^O End of Par  ^V Last
Line   ^N NextHstory
```

File save karne ke baad, hum **[CTRL + X]** press karke Editor se bahar (exit) aa sakte hain.

Back on the shell

File ke andar kya likha hai, ye dekhne ke liye hum **cat** command ka use kar sakte hain.

```
Toxicx01@htb[/htb]$ cat notes.txt

Here we can type everything we want and make our notes.
```

Linux systems mein kai aisi files hoti hain jo Penetration Testers ke liye bahut faydemand ho sakti hain, kyunki Administrators kabhi-kabhi unki permissions ya security settings sahi tarike se configure nahi karte.

Aisi hi ek important file **/etc/passwd** hai. Is file mein system ke users ki zaroori information hoti hai, jaise unke usernames, User IDs (**UIDs**), Group IDs (**GIDs**) aur Home Directories.

Pehle **/etc/passwd** file mein password hashes bhi store hote the, lekin ab ye hashes usually **/etc/shadow** file mein store hote hain, jiski permissions zyada strict hoti hain.

Lekin agar **/etc/passwd** ya doosri important files ki permissions sahi tarike se set nahi ki gayi hain, toh sensitive information expose ho sakti hai ya Privilege Escalation ke chances ban sakte hain.

Penetration Testers ke roop mein, galat permissions wali files ko identify karna humein possible vulnerabilities ke baare mein important information de sakta hai, jinhe exploit kiya ja sakta hai.

Jaise:

- Weak user accounts.
- Galat file permissions.
- Aisi files ka access milna jo normally restricted honi chahiye.

System ki security ko properly assess karne ke liye in files ko samajhna bahut zaroori hai.

VIM

Vim ek open-source text editor hai, jiska use har tarah ke ASCII text ko edit karne ke liye kiya jata hai. Ye **Nano** ki tarah hi ek text editor hai.

Vim, purane **Vi** editor ka improved version (clone) hai.

Ye ek bahut hi powerful editor hai, jo mainly zaroori kaam par focus karta hai, yani text ko edit karna.

Text editing ke alawa jo doosre kaam hote hain, unke liye Vim external programs ka interface provide karta hai. Jaise **grep**, **awk**, **sed** etc.

Ye external programs apne-apne specific kaam ko zyada achhe tarike se handle kar sakte hain, kyunki unhe unhi kaamon ke liye banaya gaya hai. Ye kaam aksar editor ke andar directly banaye gaye functions se zyada achhe tarike se kar sakte hain.

In sabhi features ki wajah se Vim:

- Small aur Compact: Iska size chhota hota hai.
- Fast: Ye tezi se kaam karta hai.
- Powerful: Ismein bahut saare powerful features hote hain.
- Flexible: Ise alag-alag tarah ke kaamon ke liye use kiya ja sakta hai.
- Less Error-Prone: Ismein galti hone ki sambhavna kam hoti hai.

Vim yahan Unix principle ko follow karta hai:

Is principle ke according, bahut saare chhote-chhote special programs banaye jaate hain, jinhe achhe se test kiya gaya hota hai aur jo apne kaam mein proven hote hain.

Jab in programs ko ek saath combine kiya jata hai aur ye ek-doosre ke saath communicate karte hain, toh milkar ek flexible aur powerful system banate hain.

Vim

```
Toxicx01@htb[/htb]$ vim
```

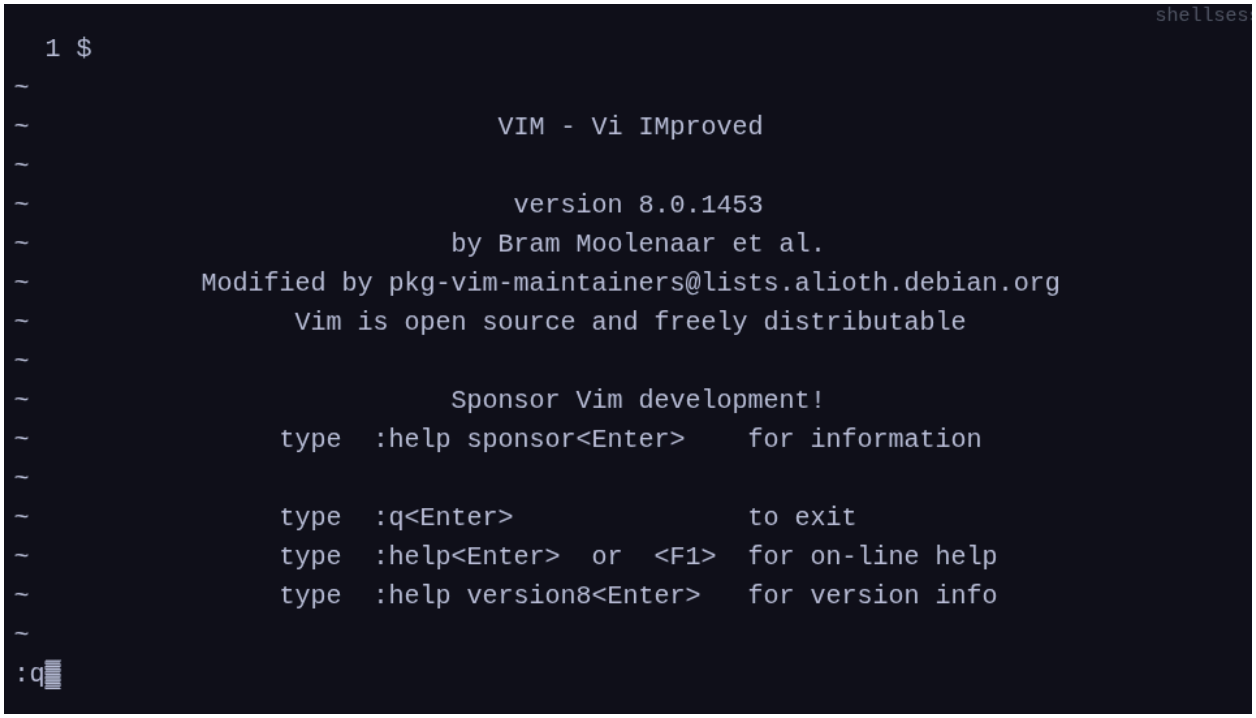
```
  1 $
~
~                           VIM - Vi IMproved
~
~                            version 8.0.1453
~                        by Bram Moolenaar et al.
~          Modified by pkg-vim-maintainers@lists.alioth.debian.org
~                Vim is open source and freely distributable
~
~                           Sponsor Vim development!
~               type  :help sponsor<Enter>    for information
~
~               type  :q<Enter>               to exit
~               type  :help<Enter>  or  <F1>  for on-line help
~               type  :help version8<Enter>   for version info
~
                                                              0,0-1
All
```

Nano ke comparison mein, **Vim** ek Modal Editor hai, jo text input aur command input ke beech difference samajh sakta hai.

Vim mein total 6 basic modes hote hain, jo hamare kaam ko aasaan banate hain aur isi wajah se Vim itna powerful editor hai.

Mode	Description
Normal	Normal mode mein hum jo bhi input dete hain, use Editor command maana jata hai. Is mode mein type kiye gaye characters editor ke buffer (jahan text temporarily store hota hai) mein add nahi hote, jaise ki baaki editors mein hote hain. Jab hum Vim editor open karte hain, toh usually hum Normal mode mein hi hote hain.
Insert	Is mode mein kuch exceptions ko chhodkar, hum jo bhi characters type karte hain, woh editor ke buffer mein insert ho jaate hain. Yani is mode ka use text likhne ke liye hota hai.
Visual	Visual mode ka use text ke kisi ek continuous part (lagataar text) ko select karne ke liye hota hai. Select kiya gaya text screen par highlight ho jata hai. Cursor ko move karke hum selected area ko badal sakte hain. Is highlighted text ko delete, copy ya replace bhi kar sakte hain.
Command	Is mode mein hum editor ke bottom mein single-line commands enter kar sakte hain. Iska use text ko sort karne, text ke kisi part ko replace karne ya delete karne jaise kaamon ke liye kiya ja sakta hai.
Replace	Replace mode mein hum jo naya text type karte hain, woh pehle se maujood text ke characters ko overwrite (unke upar likh) kar deta hai. Lekin agar cursor ki current position par overwrite karne ke liye koi purana character nahi bacha hai, toh naya text add ho jata hai.
Ex	Ye mode **Ex** text editor ke behaviour ko imitate karta hai, jo **Vim** ke purane editors mein se ek tha. Is mode mein hum ek ke baad ek multiple commands execute kar sakte hain, aur har command ke baad Normal mode mein wapas aane ki zaroorat nahi hoti.

Jab Vim editor open hota hai, toh hum **:** type karke Command mode mein ja sakte hain. Uske baad Vim ko close karne ke liye **q** type karte hain. Yani **:q** likhkar Vim ko band kar sakte hain.



Vim mein ek bahut achha feature hota hai, jiska naam vimtutor hai. Iski help se hum Vim editor ki practice kar sakte hain aur uske baare mein achhe se seekh sakte hain.

Shuruaat mein Vim thoda difficult aur complicated lag sakta hai, lekin aisa sirf kuch samay ke liye hi lagega.

Jab hum Vim ke saath comfortable ho jaate hain, toh isse kaam karne ki speed aur efficiency bahut zyada badh jaati hai.

Vim editor mein Tutor mode open karne ke do tarike hain:

1. Vim ke **Command** mode mein jaakar **:Tutor** type karein.
2. Shell mein **vimtutor** command type karein.

VimTutor

```
Toxicx01@htb[/htb]$ vimtutor
```

```
===============================================================================

    Vim is a very powerful editor that has many commands, too many to
    explain in a tutor such as this.  This tutor is designed to describe
    enough of the commands that you will be able to easily use Vim as
    an all-purpose editor.

    The approximate time required to complete the tutor is 25-30 minutes,
    depending upon how much time is spent with experimentation.

    ATTENTION:
    The commands in the lessons will modify the text.  Make a copy of this
    file to practice on (if you started "vimtutor" this is already a copy).

    It is important to remember that this tutor is set up to teach by
    use.  That means that you need to execute the commands to learn them
    properly.  If you only read the text, you will forget the commands!

    Now, make sure that your Caps-Lock key is NOT depressed and press
    the   j   key enough times to move the cursor so that lesson 1.1
    completely fills the screen.
```

Find Files and Directories

Humein jin files aur folders ki zaroorat hoti hai, unhe dhoondhna bahut zaroori hai.

Jab hum kisi Linux-based system ka access hasil kar lete hain, toh humein configuration files, users ya administrator ke banaye hue scripts aur doosri files aur folders ko dhoondhna zaroori hota hai.

Humein har ek folder ko manually browse karke check karne ki zaroorat nahi hai ki usmein kya hai aur use aakhri baar kab modify kiya gaya tha.

Is kaam ko aasaan banane ke liye hum kuch tools ka use kar sakte hain.

Which

Common tools mein se ek **which** hai.

Ye tool us file ya link ka path batata hai, jise execute (run) kiya jana chahiye.

Iski help se hum pata laga sakte hain ki cURL, netcat, wget, Python aur gcc jaise specific programs hamare operating system mein available hain ya nahi.

Ab hum apne interactive instance mein Python ko search karne ke liye **which** command ka use karte hain.

```
Toxicx01@htb[/htb]$ which python

/usr/bin/python
```

Agar hum jis program ko search kar rahe hain, woh system mein exist nahi karta hai, toh koi result display nahi hoga.

Find

Ek aur kaam aane wala tool **find** hai.

Files aur folders ko dhoondhne ke function ke alawa, is tool mein search results ko filter karne ka function bhi hota hai.

Hum file ke size ya date jaise filter parameters ka use kar sakte hain.

Hum ye bhi specify kar sakte hain ki humein sirf files search karni hain ya sirf folders.

Syntax - find

```
Toxicx01@htb[/htb]$ find <location> <options>
```

Chaliye, ek example dekhte hain ki multiple options ke saath aisi command kaisi dikhegi.

```shellsession
Toxicx01@htb[/htb]$ find / -type f -name *.conf -user root -size +20k -newermt
2020-03-03 -exec ls -al {} \; 2>/dev/null

-rw-r--r-- 1 root root 136392 Apr 25 20:29 /usr/src/linux-headers-5.5.0-1parrot1-
amd64/include/config/auto.conf
-rw-r--r-- 1 root root 82290 Apr 25 20:29 /usr/src/linux-headers-5.5.0-1parrot1-
amd64/include/config/tristate.conf
-rw-r--r-- 1 root root 95813 May  7 14:33 /usr/share/metasploit-
framework/data/jtr/repeats32.conf
-rw-r--r-- 1 root root 60346 May  7 14:33 /usr/share/metasploit-
framework/data/jtr/dynamic.conf
-rw-r--r-- 1 root root 96249 May  7 14:33 /usr/share/metasploit-
framework/data/jtr/dumb32.conf
-rw-r--r-- 1 root root 54755 May  7 14:33 /usr/share/metasploit-
framework/data/jtr/repeats16.conf
-rw-r--r-- 1 root root 22635 May  7 14:33 /usr/share/metasploit-
framework/data/jtr/korelogic.conf
-rwxr-xr-x 1 root root 108534 May  7 14:33 /usr/share/metasploit-
framework/data/jtr/john.conf
-rw-r--r-- 1 root root 55285 May  7 14:33 /usr/share/metasploit-
framework/data/jtr/dumb16.conf
-rw-r--r-- 1 root root 21254 May  2 11:59
/usr/share/doc/sqlmap/examples/sqlmap.conf
```

Ab chaliye, pichhli command mein use kiye gaye options ko thoda detail mein samajhte hain.

Agar hum mouse ko kisi particular option ke upar le jaate hain, toh ek chhoti window appear hoti hai, jismein us option ka explanation diya hota hai.

Aise explanations humein doosre modules mein bhi milenge. Agar hum abhi kisi tool se familiar nahi hain, toh ye explanations humein use samajhne mein help karenge.

Option	Description
-type f	Is option se hum search kiye jaane wale object ka type define karte hain. Yahan **f** ka matlab **file** hota hai.
-name *.conf	**-name** se hum us file ka naam batate hain jise hum search karna chahte hain. Yahan asterisk (*) ka matlab hai **.conf** extension wali sabhi files.
-user root	Ye option un sabhi files ko filter karta hai jinka owner **root** user hai.
-size +20k	Is option se hum filter kar sakte hain ki humein sirf 20 KiB se badi files hi dekhni hain.
-newermt 2020-03-03	Is option se hum date set karte hain. Ismein sirf wahi files dikhayi jayengi jo di gayi date ke baad create ya modify hui hain.
-exec ls -al {} \;	Ye option har search result par di gayi command ko execute karta hai. Curly brackets **{}** har result ke liye placeholder ka kaam karte hain. Backslash (\) agle character ko Shell dwara special meaning mein interpret hone se rokta hai, kyunki agar aisa nahi kiya gaya, toh semicolon (;) command ko wahin terminate kar dega.
2>/dev/null	Ye **STDERR** ko **null device** par redirect karta hai. Iska matlab hai ki terminal par koi error display nahi hoga. Is redirection ke baare mein hum agle section mein detail mein padhenge. Dhyan rahe, ye **find** command ka option nahi hai.

Locate

Apni files aur directories ko dhoondhne ke liye poore system mein baar-baar search karna kaafi time-consuming ho sakta hai, khaaskar jab humein alag-alag tarah ki searches karni hoti hain.

locate command humein poore system mein files aur directories ko jaldi search karne ka ek aasaan tarika provide karta hai.

find command ke comparison mein, **locate** ek local database ke saath kaam karta hai, jismein system mein maujood sabhi files aur folders ki information store hoti hai.

Hum neeche di gayi command ka use karke is database ko update kar sakte hain.

```
Toxicx01@htb[/htb]$ sudo updatedb
```

Ab agar hum **.conf** extension wali sabhi files ko search karte hain, toh hum dekhenge ki **find** command ke comparison mein **locate** command bahut jaldi results show karta hai.

```
Toxicx01@htb[/htb]$ locate *.conf

/etc/GeoIP.conf
/etc/NetworkManager/NetworkManager.conf
/etc/UPower/UPower.conf
/etc/adduser.conf
<SNIP>
```

Lekin, is tool mein utne zyada filter options available nahi hote, jitne hum find command mein use kar sakte hain.

Isliye humein hamesha ye sochna chahiye ki hamare kaam ke liye locate command use karna sahi rahega ya find command.

Ye poori tarah is baat par depend karta hai ki hum kya search karna chahte hain.

Q. What is the name of the config file that has been created after 2020-03-03 and is smaller than 28k but larger than 25k?

Ans.

Q. How many files exist on the system that have the ".bak" extension?

Ans.

Q. Submit the full path of the "xxd" binary.

Ans.

File Descriptors and Redirections

Unix/Linux operating systems mein File Descriptor (**FD**) ek reference hota hai, jise Kernel maintain karta hai. Yeh system ko Input/Output (**I/O**) operations manage karne mein help karta hai.

Yeh kisi open file, socket ya kisi bhi doosre I/O resource ke liye ek unique identifier (pehchaan) ki tarah kaam karta hai.

Windows-based operating systems mein ise File Handle kaha jata hai.

Simple words mein, File Descriptor system ka ek tarika hai jisse woh active I/O connections ko track karta hai, jaise kisi file se data read karna ya usmein data write karna.

Ise ek ticket number ki tarah samjho, jo tumhe coatroom mein apna coat jama karte waqt milta hai.

Yeh ticket (File Descriptor) tumhare coat (File ya Resource) ke saath tumhare connection ko represent karta hai. Jab bhi tumhe apna coat wapas chahiye hota hai (I/O perform karna), toh tum attendant (Operating System) ko ticket dikhate ho. Attendant ko pata hota hai ki tumhara coat kahan rakha hai (File Descriptor kis resource ko refer karta hai).

Ticket ke bina tumhare liye bahut saare rakhe hue coats mein se apna coat aasani se dhoondhna possible nahi hoga. Bilkul isi tarah, File Descriptor ke bina Operating System ko pata nahi chalega ki kis resource ke saath kaam karna hai.

Aage tum dekhoge ki File Descriptors itne important kyun hain aur aane wale examples ko samajhne ke liye inhe samajhna kyun zaroori hai.

By default, Linux mein pehle teen File Descriptors hote hain:

1. Input ke liye Data Stream

- **STDIN – 0**

2. Output ke liye Data Stream

- **STDOUT – 1**

3. Kisi error ke hone par usse related Output ke liye Data Stream

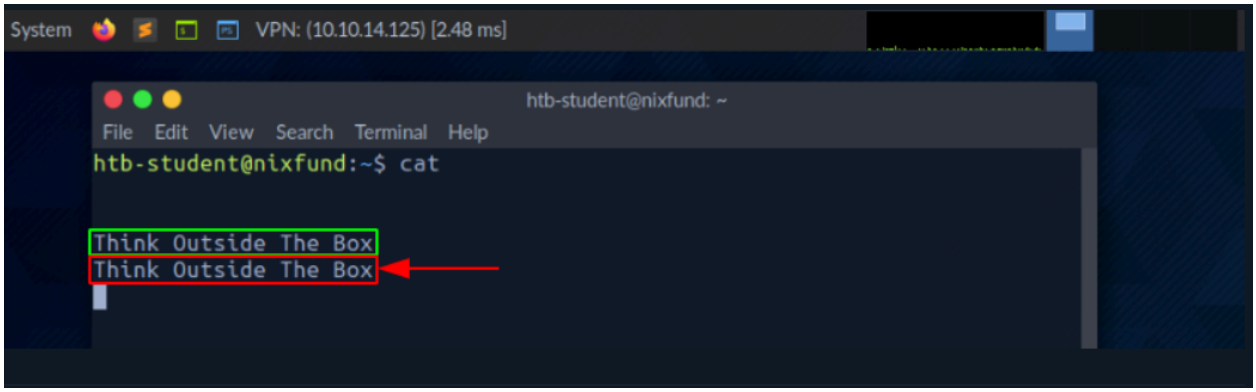
- **STDERR – 2**

STDIN and STDOUT

Aao cat command ke saath ek example dekhte hain.

Jab hum cat command run karte hain, toh hum running program ko apna standard input (STDIN - FD 0) dete hain, jise green colour se mark kiya gaya hai. Is example mein input "SOME INPUT" hai.

Jaise hi hum [ENTER] press karke apne input ko confirm karte hain, wahi input terminal par standard output (STDOUT - FD 1) ke roop mein wapas dikhai deta hai, jise red colour se mark kiya gaya hai.

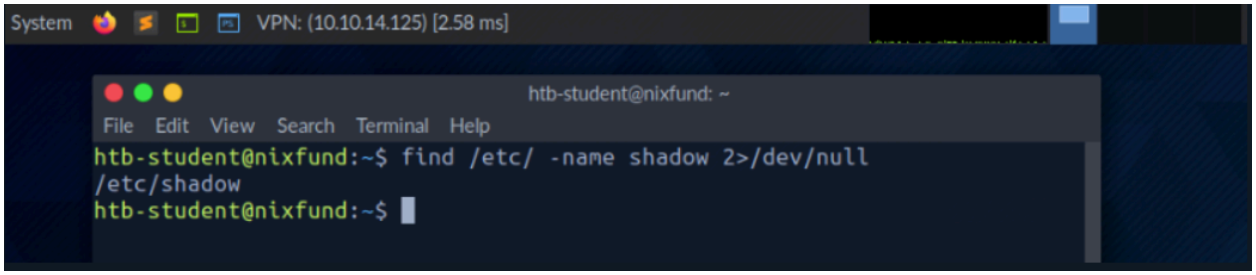


STDOUT and STDERR

Agale example mein, hum **find** command ka use karenge.

Ismein hum standard output (**STDOUT - FD 1**) ko green colour mein aur standard error (**STDERR - FD 2**) ko red colour mein dekhenge.

```
Toxicx01@htb[/htb]$ find /etc/ -name shadow 2>/dev/null
```



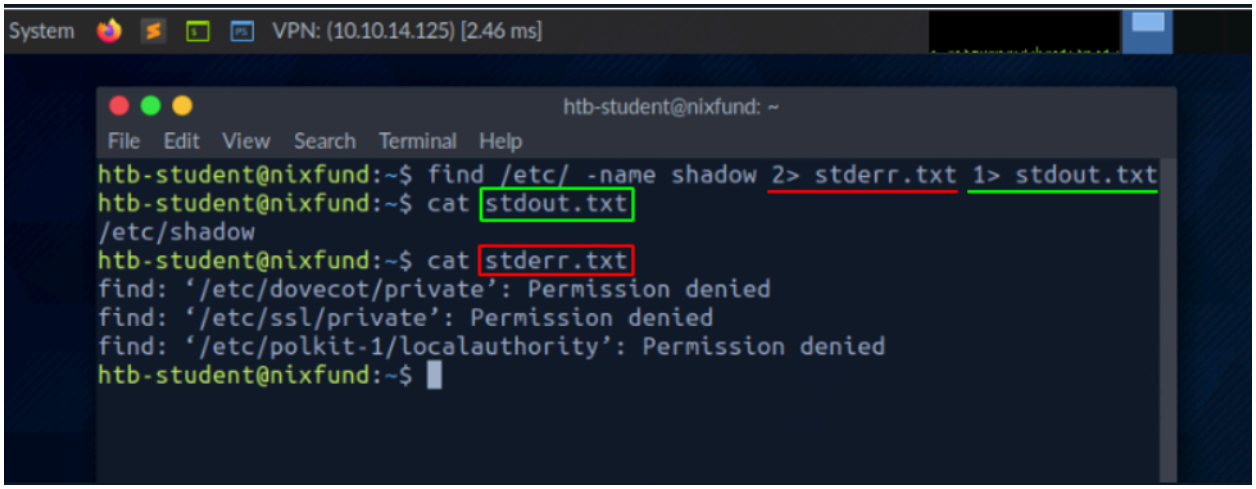
Redirect STDOUT and STDERR to Separate Files

Humein dhyan dena chahiye ki pichhle example mein humne greater-than sign (>) se pehle koi number use nahi kiya tha.

Aisa isliye tha kyunki humne pehle sabhi standard errors ko "null device" mein redirect kar diya tha. Is wajah se humein sirf standard output (**FD 1 - STDOUT**) hi milta hai.

Is process ko aur precise (sahi) banane ke liye, ab hum standard error (**FD 2 - STDERR**) aur standard output (**FD 1 - STDOUT**) ko alag-alag files mein redirect karenge.

```
Toxicx01@htb[/htb]$ find /etc/ -name shadow 2> stderr.txt 1> stdout.txt
```



Redirect STDIN

Jaisa ki hum pehle dekh chuke hain, File Descriptors ke saath greater-than character (>) ka use karke hum errors aur output ko redirect kar sakte hain.

Yeh kaam lower-than sign (**<**) ke saath bhi hota hai. Lekin lower-than sign ka use standard input (**FD 0 - STDIN**) ke liye kiya jata hai.

In characters ko hum ek direction (disha) ke roop mein samajh sakte hain, jo arrow ki tarah humein batate hain ki data kahan se (from where) aur kahan par (where to) redirect hona chahiye.

Hum **cat** command ka use karke file **stdout.txt** ke contents ko **STDIN** ke roop mein use karenge.

```
Toxicx01@htb[/htb]$ cat < stdout.txt
```

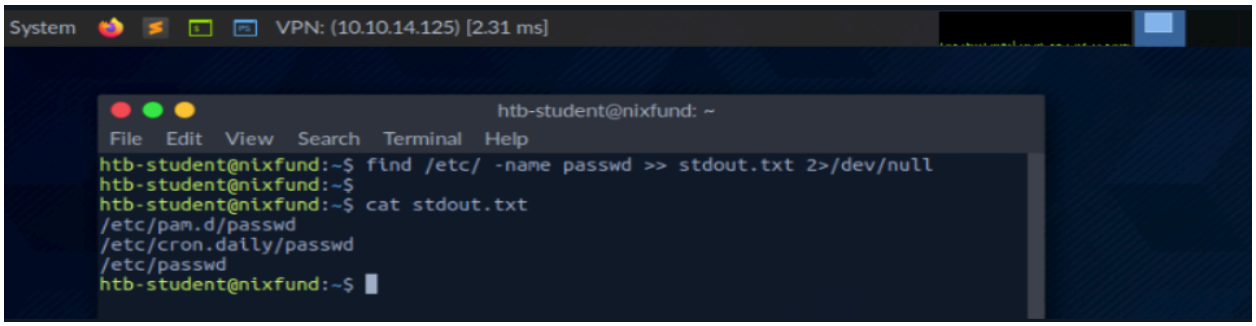
Redirect STDOUT and Append a File

Jab hum apne **STDOUT** ko redirect karne ke liye greater-than sign (**>**) ka use karte hain, toh agar file pehle se exist nahi karti hai, toh ek nayi file automatically create ho jaati hai.

Agar woh file pehle se exist karti hai, toh bina confirmation maange uska purana data overwrite (replace) kar diya jata hai.

Agar hum apne **STDOUT** ko existing file mein append (jodna) chahte hain, toh hum double greater-than sign (**>>**) ka use kar sakte hain.

```
Toxicx01@htb[/htb]$ find /etc/ -name passwd >> stdout.txt 2>/dev/null
```



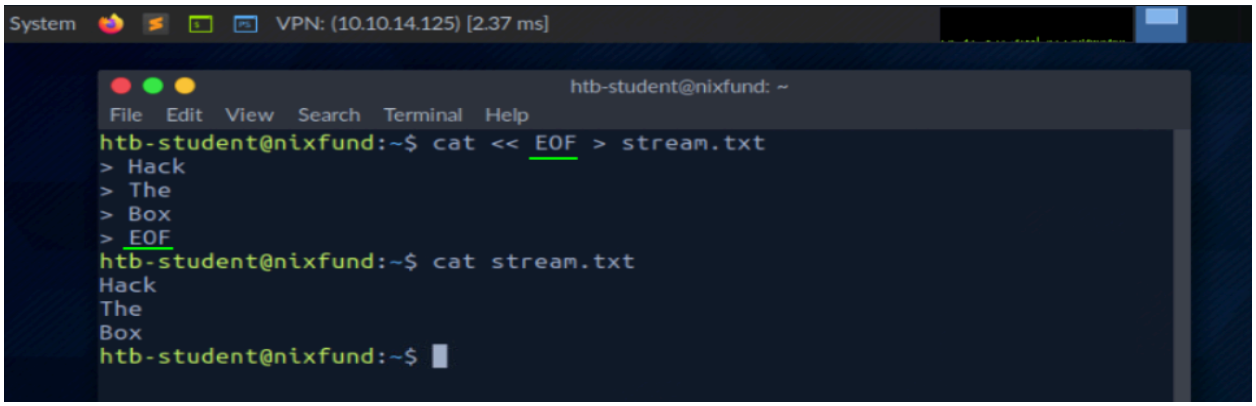
Redirect STDIN Stream to a File

Hum double lower-than characters (**<<**) ka use karke bhi apne standard input ko ek stream ke through add kar sakte hain.

Hum Linux system ki file ke End-Of-File (**EOF**) function ka use kar sakte hain, jo input ke end ko define karta hai.

Agale example mein, hum **cat** command ka use karenge, jo stream ke through diye gaye input ko read karega aur us input ko **stream.txt** naam ki file mein redirect kar dega.

```
Toxicx01@htb[/htb]$ cat << EOF > stream.txt
```



Pipes (|)

STDOUT ko redirect karne ka ek aur tarika pipes (**|**) ka use karna hai.

Pipes tab useful hote hain jab hum ek program ke **STDOUT** (output) ko doosre program ke through process karna chahte hain.

Iske liye sabse commonly used tools mein se ek **grep** hai, jiska use hum agale example mein karenge.

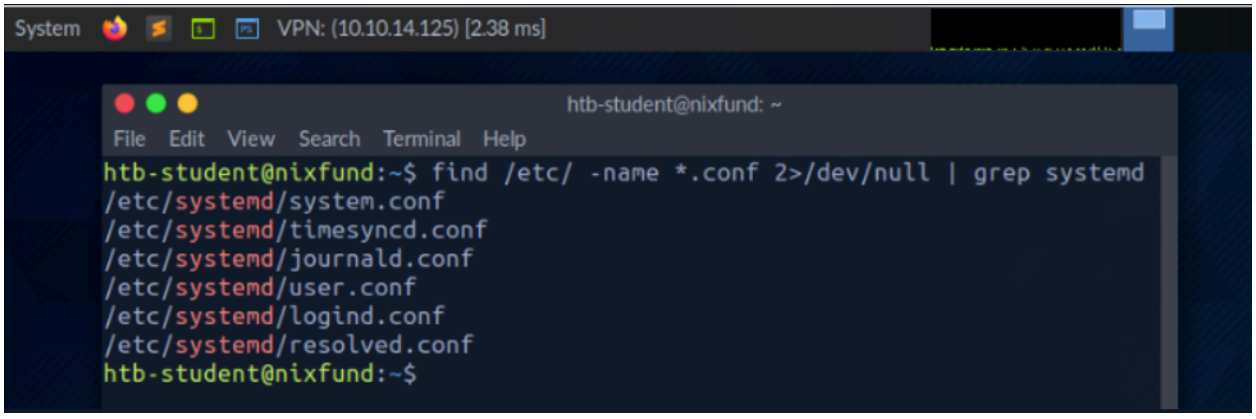
grep ka use hum apne define kiye gaye pattern ke according **STDOUT** ko filter karne ke liye karte hain.

Agale example mein, hum **find** command ka use karke **/etc/** directory ke andar **.conf** extension wali sabhi files ko search karenge.

Kisi bhi error ko **null device** (**/dev/null**) mein redirect kar diya jayega.

Uske baad, hum **grep** ka use karke results ko filter karenge aur specify karenge ki sirf wahi lines display honi chahiye jinmein **systemd** pattern present ho.

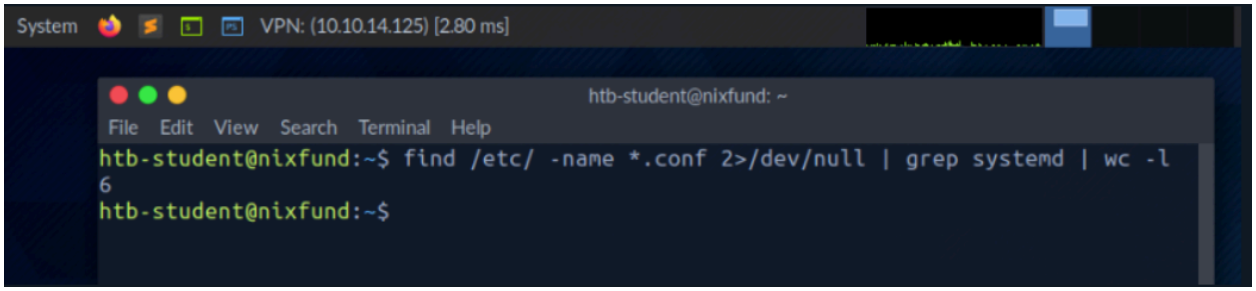
```
Toxicx01@htb[/htb]$ find /etc/ -name *.conf 2>/dev/null | grep systemd
```



Redirections sirf ek baar hi kaam nahi karte. Hum pehle se mile hue results ko kisi doosre program mein redirect karne ke liye bhi use kar sakte hain.

Agale example mein, hum **wc** naam ke tool ka use karenge, jo humein mile hue total results ki sankhya (count) ginega.

```
shellsess
Toxicx01@htb[/htb]$ find /etc/ -name *.conf 2>/dev/null | grep systemd | wc -l
```



Ab jab humein File Descriptors, Redirections aur Pipes ki basic understanding ho gayi hai, toh hum apne commands ko zyada efficiently arrange kar sakte hain, taaki humein bilkul wahi information mil sake jiski humein zaroorat hai.

Yeh knowledge humein input aur output ke flow ko files, processes aur system ke beech control karne ki permission deti hai, jisse hum data ko zyada effectively handle kar sakte hain.

In tools ka use karke hum apne tasks ko aasaan bana sakte hain, unnecessary steps se bach sakte hain aur files aur system resources ke saath zyada organized aur efficient tarike se kaam kar sakte hain.

Aakhir mein, isse operations ko manage karne mein hamari productivity aur accuracy dono improve hoti hain.

Q. How many files exist on the system that have the ".log" file extension?

Ans.

Q. How many total packages are installed on the target system?

Ans.

Filter Contents

Pichhle section mein humne dekha tha ki redirection ka use karke ek program ke output ko doosre program mein kaise bheja jata hai, taaki us output ko aage process kiya ja sake.

Ab baat karte hain ki bina kisi text editor ko open kiye, directly command line se files ko kaise read kar sakte hain.

Iske liye do powerful tools hain — **more** aur **less**.

Inhe Pagers kaha jata hai. Ye humein kisi file ka content ek baar mein ek screen par dekhne ki permission dete hain.

Dono tools ka kaam lagbhag ek jaisa hai, lekin inki functionality mein kuch differences hain, jinhe hum aage samjhenge.

more aur **less** ka use karke tum badi files mein aasani se scroll kar sakte ho, text search kar sakte ho aur bina file ko modify kiye aage ya peeche navigate kar sakte ho.

Yeh tab bahut useful hota hai jab tum badi log files ya text files ke saath kaam kar rahe hote ho, jo ek screen mein poori tarah fit nahi hoti hain.

Is section ka goal yeh seekhna hai ki content ko kaise filter karein aur pichhle commands se mile redirected output ko kaise handle karein.

Lekin filtering shuru karne se pehle, humein kuch zaroori tools aur commands ko samajhna hoga, jo filtering ko zyada efficient aur powerful banane ke liye specially design kiye gaye hain.

Commands ke output ko filter karna shuru karne se pehle, aao kuch basic tools ko samajhte hain, jo text ko efficiently search, sort aur process karne mein hamari help karenge.

Yeh tools tab bahut important hote hain jab hum bahut saare data ke saath kaam kar rahe hote hain ya aise tasks ko automate karna chahte hain jinmein information ko search, sort ya process karna hota hai.

Aao kuch examples dekhte hain, taaki samajh sakein ki ye tools practically kaise kaam karte hain.

More

```
Toxicx01@htb[/htb]$ cat /etc/passwd | more
```

Linux mein **/etc/passwd** file system ke users ke liye ek phone directory ki tarah hoti hai.

Ismein username, user ID, group ID, home directory aur user ke dwara use kiye jaane wale default shell jaisi details hoti hain.

Jab hum **cat** command ka use karke file ka content read karte hain aur us output ko **more** mein redirect karte hain, toh pehle bataya gaya Pager open ho jata hai.

Aur hum automatically file ke shuru (beginning) se content dekhna start karte hain.

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
<SNIP>
--More--
```

[Q] key press karke hum is **pager** se bahar (exit) aa sakte hain.

Hum dekhenge ki **pager** se bahar aane ke baad bhi output terminal mein bana rehta hai.

Less

Agar hum ab **less** tool ko dekhen, toh iske **man page** mein humein pata chalega ki ismein **more** ke comparison mein kaafi zyada features hain.

```
Toxicx01@htb[/htb]$ less /etc/passwd
```

Iska dikhne ka tarika lagbhag more jaisa hi hai.

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
<SNIP>
:
```

Jab hum **[Q]** key dabakar **less** ko close karte hain, toh hum dekhenge ki **more** ke unlike, jo output humne dekha tha, woh terminal mein nahi rehta hai.

Head

```
Toxicx01@htb[/htb]$ head /etc/passwd

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
```

Kabhi-kabhi humein file ke sirf kuch specific parts mein interest hota hai, jaise file ki shuruaat (beginning) ya end mein.

Agar hum file ki sirf pehli (first) kuch lines dekhna chahte hain, toh hum **head** tool ka use kar sakte hain.

By default, **head** di gayi file ya input ki pehli 10 lines print karta hai, agar hum koi alag number specify nahi karte hain.

Tail

Agar hum kisi file ya results ka sirf aakhiri (last) hissa dekhna chahte hain, toh hum head ke opposite tool **tail** ka use kar sakte hain.

tail by default file ki aakhiri 10 lines dikhata hai.

```
Toxicx01@htb[/htb]$ tail /etc/passwd

miredo:x:115:65534::/var/run/miredo:/usr/sbin/nologin
usbmux:x:116:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
rtkit:x:117:119:RealtimeKit,,,:/proc:/usr/sbin/nologin
nm-openvpn:x:118:120:NetworkManager
OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
nm-openconnect:x:119:121:NetworkManager OpenConnect
plugin,,,:/var/lib/NetworkManager:/usr/sbin/nologin
pulse:x:120:122:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
beef-xss:x:121:124::/var/lib/beef-xss:/usr/sbin/nologin
lightdm:x:122:125:Light Display Manager:/var/lib/lightdm:/bin/false
do-agent:x:998:998::/home/do-agent:/bin/false
user6:x:1000:1000:,,,:/home/user6:/bin/bash
```

In tools ke available options ko explore karna aur unke saath practice (experiment) karna hamare liye bahut faydemand hoga.

Sort

Hum jin Results aur Files ke saath kaam karte hain, woh aksar kisi proper order mein nahi hote.

Kai baar humein Results ko Alphabetical (A se Z) ya Numerical (chhote number se bade number) order mein arrange karna padta hai, taaki unhe samajhna aur dekhna aasaan ho.

Iske liye hum **sort** naam ke tool ka use kar sakte hain.

```
shellsession
Toxicx01@htb[/htb]$ cat /etc/passwd | sort

_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
cry0l1t3:x:1001:1001::/home/cry0l1t3:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
dovecot:x:114:117:Dovecot mail server,,,:/usr/lib/dovecot:/usr/sbin/nologin
dovenull:x:115:118:Dovecot login user,,,:/nonexistent:/usr/sbin/nologin
ftp:x:113:65534::/srv/ftp:/usr/sbin/nologin
games:x:5:60:games:/usr/games:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
htb-student:x:1002:1002::/home/htb-student:/bin/bash
<SNIP>
```

Ab hum dekh sakte hain ki output ab **root** se shuru nahi ho raha hai, balki Alphabetical order (A se Z) mein sorted hai.

Grep

Kai baar humein kuch specific information search karne ki zarurat hoti hai, jo hamare define kiye gaye patterns (Pattern) se match karti ho.

Is kaam ke liye sabse zyada use hone wale tools mein se ek grep hai. Ye pattern ke according information search karne ke liye kai powerful features provide karta hai.

Example: Hum grep ka use un users ko search karne ke liye kar sakte hain, jinka Default Shell /bin/bash set hai.

```
Toxicx01@htb[/htb]$ cat /etc/passwd | grep "/bin/bash"

root:x:0:0:root:/root:/bin/bash
mrb3n:x:1000:1000:mrb3n:/home/mrb3n:/bin/bash
cry0l1t3:x:1001:1001::/home/cry0l1t3:/bin/bash
htb-student:x:1002:1002::/home/htb-student:/bin/bash
```

Ye sirf ek example hai ki **grep** ka use karke pehle se define kiye gaye patterns ke according data ko efficiently filter kaise kar sakte hain.

Ek aur possibility hai ki hum kuch specific results ko exclude (हटा) kar sakte hain.

Iske liye **grep** ke saath **-v** option ka use kiya jata hai.

Agle example mein, hum un sabhi users ko exclude karenge jinhone standard shell ko disable kar rakha hai, jiska naam **/bin/false** ya **/usr/bin/nologin** hai.

```shellsession
Toxicx01@htb[/htb]$ cat /etc/passwd | grep -v "false\|nologin"

root:x:0:0:root:/root:/bin/bash
sync:x:4:65534:sync:/bin:/bin/sync
postgres:x:111:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
user6:x:1000:1000:,,,:/home/user6:/bin/bash
```

Ye sirf ek example hai ki **grep** ka use karke pehle se define kiye gaye patterns ke according data ko efficiently filter kaise kar sakte hain.

Ek aur possibility hai ki hum kuch specific results ko exclude (हटा) kar sakte hain.

Iske liye **grep** ke saath **-v** option ka use kiya jata hai.

Agle example mein, hum un sabhi users ko exclude karenge jinhone standard shell ko disable kar rakha hai, jiska naam **/bin/false** ya **/usr/bin/nologin** hai.

```shellsession
Toxicx01@htb[/htb]$ cat /etc/passwd | grep -v "false\|nologin"

root:x:0:0:root:/root:/bin/bash
sync:x:4:65534:sync:/bin:/bin/sync
postgres:x:111:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
user6:x:1000:1000:,,,:/home/user6:/bin/bash
```

Cut

Alag-alag characters ki help se specific results ko ek-doosre se separate kiya ja sakta hai. In characters ko Delimiters kehte hain.

Yahan ye samajhna useful hai ki hum kisi specific delimiter ko kaise hata sakte hain aur line mein kisi particular position par maujood word ko kaise display kar sakte hain.

Is kaam ke liye **cut** tool ka use kiya ja sakta hai.

Isliye hum **-d** option ka use karke delimiter ko colon character (:) set karte hain.

Aur **-f** option ki help se line mein us position ko define karte hain, jise hum output mein dekhna chahte hain.

```shellsession
Toxicx01@htb[/htb]$ cat /etc/passwd | grep -v "false\|nologin" | cut -d":" -f1

root
sync
postgres
mrb3n
cry0l1t3
htb-student
```

Tr

Kisi line mein maujood kuch specific characters ko apne diye gaye characters se replace karne ka ek aur tarika tr tool hai.

Isme pehle option mein hum define karte hain ki kis character ko replace karna hai.

Aur doosre option mein hum define karte hain ki uski jagah kaunsa character lagana hai.

Agle example mein, hum colon character (**:**) ko space se replace karenge.

```
Toxicx01@htb[/htb]$ cat /etc/passwd | grep -v "false\|nologin" | tr ":" " "

root x 0 0 root /root /bin/bash
sync x 4 65534 sync /bin /bin/sync
postgres x 111 117 PostgreSQL administrator,,, /var/lib/postgresql /bin/bash
mrb3n x 1000 1000 mrb3n /home/mrb3n /bin/bash
cry0l1t3 x 1001 1001  /home/cry0l1t3 /bin/bash
htb-student x 1002 1002  /home/htb-student /bin/bash
```

Column

Kyunki search results aksar clear tarike se dikhai nahi dete hain, isliye un results ko table ke form mein dikhane ke liye **column** tool ka use kiya jata hai.

Iske liye **-t** option ka use kiya jata hai.

```
Toxicx01@htb[/htb]$ cat /etc/passwd | grep -v "false\|nologin" | tr ":" " " |
column -t

root         x  0     0      root              /root               /bin/bash
sync         x  4     65534  sync              /bin                /bin/sync
postgres     x  111   117    PostgreSQL        administrator,,,
/var/lib/postgresql     /bin/bash
mrb3n        x  1000  1000   mrb3n             /home/mrb3n         /bin/bash
cry0l1t3     x  1001  1001   /home/cry0l1t3    /bin/bash
htb-student  x  1002  1002   /home/htb-student  /bin/bash
```

Awk

Jaisa ki humne notice kiya hoga, postgres user ki line mein ek column extra hai.

Aise results ko aasani se sort out karne ke liye awk programming ka use karna faydemand hota hai.

Iski help se hum kisi line ka pehla result ($1) aur aakhri result ($NF) display kar sakte hain.

```shellsession
Toxicx01@htb[/htb]$ cat /etc/passwd | grep -v "false\|nologin" | tr ":" " " | awk '{print $1, $NF}'

root /bin/bash
sync /bin/sync
postgres /bin/bash
mrb3n /bin/bash
cry0l1t3 /bin/bash
htb-student /bin/bash
```

Sed

Aise kai mauke aayenge jab hum poori file ya Standard Input mein kuch specific names ko change karna chahenge.

Iske liye hum **sed** naam ke Stream Editor tool ka use kar sakte hain.

Iska sabse common use text ko replace karna hai.

Yahan **sed** hamare define kiye gaye patterns ko Regular Expressions (Regex) ke form mein search karta hai aur unhe hamare define kiye gaye doosre pattern se replace karta hai.

Ab hum pichhle results ko hi lete hain aur maan lete hain ki hum word **bin** ko **HTB** se replace karna chahte hain.

Shuruaat mein **s** flag ka matlab Substitute command hota hai.

Uske baad hum us pattern ko specify karte hain jise replace karna hai.

Slash **(/)** ke baad, teesri position mein hum woh pattern likhte hain jisse replace karna hai.

Aakhir mein, hum **g** flag ka use karte hain, jiska matlab hai ki jitni bhi matching jagah hain, un sabhi ko replace karna hai.

```shellsession
Toxicx01@htb[/htb]$ cat /etc/passwd | grep -v "false\|nologin" | tr ":" " " | awk '{print $1, $NF}' | sed 's/bin/HTB/g'

root /HTB/bash
sync /HTB/sync
postgres /HTB/bash
mrb3n /HTB/bash
cry0l1t3 /HTB/bash
htb-student /HTB/bash
```

Wc

Aakhir mein, kai baar ye jaana useful hota hai ki humein kitne successful matches mile hain.

Lines ya characters ko manually count karne se bachne ke liye hum **wc** tool ka use kar sakte hain.

-l option ki help se hum specify karte hain ki sirf lines ko count kiya jaye.

```shellsession
Toxicx01@htb[/htb]$ cat /etc/passwd | grep -v "false\|nologin" | tr ":" " " | awk '{print $1, $NF}' | wc -l

6
```

Practice

Dhyan rakhein ki iske alawa bhi bahut saare tools available hain, jinhe aap apni learning journey ke dauran use kar sakte hain.

Aapko strongly recommend kiya jata hai ki alag-alag tasks ke liye doosre tools bhi explore karein, taaki aapki skills badh sakein. Ho sakta hai ki aapko kuch aise tools mil jaayein jo aapki personal preferences aur working style ke liye zyada useful hon.

Iski koi fixed limitation nahi hai, isliye alag-alag possibilities ko explore karein aur community mein share kiye gaye resources ka fayda uthayein.

Agar hum in tools aur unke functions se pehle familiar nahi hain, toh shuruaat mein itne saare tools ko samajhna thoda overwhelming lag sakta hai.

Isliye apna time lein aur in tools ke saath experiment karein.

Inke man pages (man <tool>) dekhein ya help command (<tool> -h / <tool> --help) ka use karein.

In sabhi tools se familiar hone ka sabse achha tarika Practice karna hai.

Inka jitna ho sake utna use karne ki koshish karein. Kuch hi samay mein hum intuitively bahut saari cheezon ko filter karna seekh jayenge.

Yahan kuch optional exercises di gayi hain, jinki help se hum apni filtering skills ko improve kar sakte hain aur Terminal aur commands se zyada familiar ho sakte hain.

Humein jis file par kaam karna hai, woh hamare target ki /etc/passwd file hai. Hum upar dikhaye gaye kisi bhi command ka use kar sakte hain.

Hamara goal sirf specific information ko filter karke display karna hai.

File ko read karein aur uske contents ko is tarah filter karein ki humein sirf ye information dikhai de:

Q. How many services are listening on the target system on all interfaces? (Not on localhost and IPv4 only)

Ans.

Q. Determine what user the ProFTPd server is running under. Submit the username as the answer.
Ans.

Q. Use cURL from your Pwnbox (not the target machine) to obtain the source code of the "https://www.inlanefreight.com" website and filter all unique paths (https://www.inlanefreight.com/directory" or "/another/directory") of that domain. Submit the number of these paths as the answer.
Ans.

Regular Expressions (RegEx)

Regular Expressions (**RegEx**) text ya files mein patterns search karne ke liye precise rules banane ki ek technique hai.

Ye humein data ko bahut accurately find, replace aur manipulate karne ki permission dete hain.

RegEx ko ek aise filter ki tarah samjho, jise hum apni zarurat ke hisaab se customize kar sakte hain. Iski help se hum text ke andar wahi cheezein search kar sakte hain, jo humein chahiye.

Chahe data ko analyze karna ho, input ko validate karna ho ya advanced searching karni ho, RegEx ka use kiya ja sakta hai.

RegEx ka basic concept ye hai ki ye characters aur symbols ka ek sequence hota hai, jo milkar ek search pattern banata hai.

In patterns mein aksar kuch special symbols ka use hota hai, jinhe Metacharacters kaha jata hai.

Ye metacharacters search ka structure define karte hain, na ki kisi normal text ko represent karte hain.

Example: Metacharacters ki help se hum define kar sakte hain ki humein digits (numbers), letters ya kisi particular pattern se match hone wale characters search karne hain.

RegEx kai programming languages aur tools mein available hai, jaise **grep** aur **sed**.

Is wajah se RegEx hamare toolkit ka ek versatile aur powerful tool hai.

Grouping

Regex humein kai doosri cheezon ke saath-saath apne desired search patterns ko group karne ki facility bhi provide karta hai.

Basic se, Regex teen alag-alag concepts follow karta hai, jinhe teen different types ke brackets se identify kiya jata hai:

Grouping Operators

No.	Operators	Description
1	**(a)**	Round brackets ka use Regex ke alag-alag parts ko group karne ke liye hota hai. Brackets ke andar hum aur bhi patterns define kar sakte hain, jinhe ek saath process kiya jana chahiye.
2	**[a-z]**	Square brackets ka use Character Classes define karne ke liye hota hai. Inke andar hum un characters ki list de sakte hain, jinhe search karna hai.
3	**{1,10}**	Curly brackets ka use Quantifiers define karne ke liye hota hai. Inke andar hum ek number ya range specify kar sakte hain, jo batata hai ki pichhle pattern ko kitni baar repeat karna hai.
4	**|**	Ise OR Operator bhi kaha jata hai. Ye result tab show karta hai jab diye gaye do expressions mein se koi ek match hota hai.
5	**.***	Ye AND Operator ki tarah kaam karta hai. Ye result tabhi display karta hai jab dono expressions present hon aur diye gaye order mein match karein.

Maan lo hum **OR** operator ka use karte hain.

Regex diye gaye search parameters mein se kisi ek ko search karta hai.

Agle example mein, hum un lines ko search karenge jinmein **my** ya **false** word present ho.

In operators ka use karne ke liye, **grep** mein **-E** option lagakar Extended Regex apply karna zaroori hai.

OR operator

```shellsession
cry0l1t3@htb:~$ grep -E "(my|false)" /etc/passwd

lxd:x:105:65534::/var/lib/lxd/:/bin/false
pollinate:x:109:1::/var/cache/pollinate:/bin/false
mysql:x:116:120:MySQL Server,,:/nonexistent:/bin/false
```

Kyunki diye gaye dono search parameters mein se koi ek har baar un teen lines mein maujood hai, isliye teeno lines display hoti hain.

Lekin agar hum **AND** operator ka use karte hain, toh same search parameters ke liye humein ek alag result milega.

AND operator

```
cry0l1t3@htb:~$ grep -E "(my.*false)" /etc/passwd

mysql:x:116:120:MySQL Server,,:/nonexistent:/bin/false
```

Basically, is command ke through hum keh rahe hain ki humein aisi line search karni hai, jismein **my** aur **false** dono present hon.

Iska ek simple example ye bhi hai ki hum **grep** ko do baar use kar sakte hain, jo kuch is tarah dikhega:

```
cry0l1t3@htb:~$ grep -E "my" /etc/passwd | grep -E "false"

mysql:x:116:120:MySQL Server,,:/nonexistent:/bin/false
```

Yahan kuch optional tasks diye gaye hain, jo aapko RegEx ki practice karne aur unhe zyada effectively handle karne ki ability improve karne mein help karenge.

In exercises mein aapke **Pwnbox** instance ki **/etc/ssh/sshd_config** file ka use kiya jayega. Isse aap kisi configuration file mein RegEx ke real-world applications ko explore kar sakenge.

In tasks ko complete karne se aapko patterns ke saath kaam karne, text search karne aur text ko manipulate karne ka practical experience milega.

No.	Translation
1.	Un sabhi lines ko show karo jinmein **#** character nahi hai.
2.	Un sabhi lines ko search karo jinmein koi aisa word hai jo **Permit** se start hota hai.
3.	Un sabhi lines ko search karo jinmein koi aisa word hai jo **Authentication** par end hota hai.
4.	Un sabhi lines ko search karo jinmein **Key** word present hai.
5.	Un sabhi lines ko search karo jo **Password** se start hoti hain aur jinmein **yes** present hai.
6.	Un sabhi lines ko search karo jo **yes** par end hoti hain.

Permission Management

Linux mein permissions keys ki tarah hoti hain, jo files aur directories ke access ko control karti hain.

Ye permissions users aur groups dono ko di jaati hain. Bilkul waise hi jaise kisi organization mein specific logon aur teams ko keys di jaati hain.

Har user ek se zyada groups ka member ho sakta hai. Kisi group ka member hone se user ko additional access rights milte hain, jinki help se woh files aur directories par specific actions perform kar sakta hai.

Har file aur directory ka ek owner (user) hota hai aur woh kisi ek group se associated hoti hai.

In files ki permissions owner aur group dono ke liye define hoti hain. Ye decide karti hain ki kaunse actions allowed hain, jaise reading (padhna), writing (likhna) ya executing (run karna).

Jab aap koi nayi file ya directory create karte hain, toh woh automatically aapki ho jaati hai aur us group se associated ho jaati hai jiske aap member hain. Ye bilkul waise hi hai jaise kisi company mein naya project automatically aapki team ki responsibility mein aa jata hai.

Aasaan shabdon mein, Linux permissions rules ya keys ka ek set hoti hain, jo decide karti hain ki kaun kisi resource ko access ya modify kar sakta hai. Isse poore system mein security bani rehti hai aur users ke beech proper collaboration hota hai.

Jab koi user Linux directory ke contents ko access karna chahta hai, toh ye bilkul aisa hai jaise andar jaane se pehle kisi darwaze ko unlock karna.

Kisi directory ke andar enter karne ya navigate karne ke liye user ke paas sahi key honi chahiye. Ye key directory par **execute** permission hoti hai.

Is permission ke bina, chahe directory ke contents user ko visible hon, phir bhi woh us directory ke andar enter nahi kar sakta ya uske through navigate nahi kar sakta.

Doosre shabdon mein, directory par **execute** permission hona bilkul hallway se hokar andar ke rooms tak jaane ki permission milne jaisa hai.

Ye permission aapko directory ke andar jaane aur uski structure ko explore karne ki ability deti hai. Lekin isse aap directory ke andar kya hai, use dekhne ya modify karne ki permission nahi milti.

Is permission ke bina user directory ke contents access nahi kar sakta aur use **Permission Denied** error message dikhega.

```shellsession
cry0l1t3@htb[/htb]$ ls -l

drw-rw-r-- 3 cry0l1t3 cry0l1t3   4096 Jan 12 12:30 scripts


cry0l1t3@htb[/htb]$ ls -al mydirectory/

ls: cannot access 'mydirectory/script.sh': Permission denied
ls: cannot access 'mydirectory/..': Permission denied
ls: cannot access 'mydirectory/subdirectory': Permission denied
ls: cannot access 'mydirectory/.': Permission denied
total 0
d????????? ? ? ? ?            ? .
d????????? ? ? ? ?            ? ..
-????????? ? ? ? ?            ? script.sh
d????????? ? ? ? ?            ? subdirectory
```

Yeh samajhna important hai ki kisi bhi directory ke andar jaane (traverse karne) ke liye Execute (x) permission zaroori hoti hai, chahe user ke paas kisi bhi level ka access ho.

Lekin directory par Execute permission hone ka matlab yeh nahi hai ki user uske andar ki files ko execute ya modify bhi kar sakta hai. Iska kaam sirf directory ke andar jaana aur uske content tak pahunchne ki permission dena hai.

- Directory ke andar ki files execute karne ke liye: User ke paas us particular file par Execute (x) permission honi chahiye.
- Directory ke content ko modify karne ke liye: Jaise files aur subdirectories create, delete ya rename karna, user ke paas directory par Write (w) permission honi chahiye.

Linux systems mein poora permission system Octal Number System par based hota hai. Basically, kisi file ya directory ko 3 tarah ki permissions di ja sakti hain:

- (r) – Read: Padhne ki permission.
- (w) – Write: Likhne ya badlav karne ki permission.
- (x) – Execute: Execute karne ki permission (directory ke case mein andar jaane ki permission).

Permissions ko **owner**, **group** aur **others** ke liye set kiya ja sakta hai, jaisa ki agle example mein unki corresponding permissions ke saath dikhaya gaya hai.

```
cry0l1t3@htb[/htb]$ ls -l /etc/passwd

- rwx rw- r--   1 root root 1641 May  4 23:42 /etc/passwd
- --- --- ---   |  |    |    |   |__________|
|  |   |   |    |  |    |    |        |_ Date
|  |   |   |    |  |    |    |__________ File Size
|  |   |   |    |  |    |_______________ Group
|  |   |   |    |  |____________________ User
|  |   |   |    |_________________________ Number of hard links
|  |   |   |_ Permission of others (read)
|  |   |_____ Permissions of the group (read, write)
|  |_________ Permissions of the owner (read, write, execute)
|____________ File type (- = File, d = Directory, l = Link, ... )
```

Permissions Change Karna

Hum **chmod** command ka use karke permissions ko modify (change) kar sakte hain. Iske liye permission group references (**u** – Owner, **g** – Group, **o** – Others, **a** – All users) aur permissions add ya remove karne ke liye **+** ya **-** ka use kar sakte hain.

Niche diye gaye example mein maan lete hain ki hamare paas **shell** naam ki ek file hai. Hum iski permissions change karna chahte hain, taaki yeh script us user ki ownership mein ho, executable na rahe aur sabhi users ke liye Read/Write permissions set ho jaayein.

```
cry0l1t3@htb[/htb]$ ls -l shell

-rwxr-x--x   1 cry0l1t3 htbteam 0 May  4 22:12 shell
```

Ab hum sabhi users ke liye **Read** permission apply kar sakte hain aur uska result dekh sakte hain.

```
cry0l1t3@htb[/htb]$ chmod a+r shell && ls -l shell

-rwxr-xr-x   1 cry0l1t3 htbteam 0 May  4 22:12 shell
```

Hum Octal value assignment ka use karke baaki sabhi users (others) ke liye sirf Read permission bhi set kar sakte hain.

```
cry0l1t3@htb[/htb]$ chmod 754 shell && ls -l shell

-rwxr-xr--   1 cry0l1t3 htbteam 0 May  4 22:12 shell
```

Aaiye, isse aur achhe se samajhne ke liye permission assignment se jude sabhi representations ko dekhte hain, taaki hum samajh sakein ki permissions ki calculation kaise hoti hai.

```
Binary Notation:               4 2 1  |  4 2 1  |  4 2 1
-------------------------------------------------------
Binary Representation:         1 1 1  |  1 0 1  |  1 0 0
-------------------------------------------------------
Octal Value:                     7    |    5    |    4
-------------------------------------------------------
Permission Representation:     r w x  |  r - x  |  r - -
```

Agar hum **Binary Notation** mein diye gaye values ko **Binary Representation** ke set bits ke saath add karte hain, toh humein **Octal Value** milti hai.

Permission Representation mein **Binary Representation** ke set bits ko 3 characters (**r**, **w**, **x**) ki madad se dikhaya jata hai, jisse set ki gayi permissions ko samajhna aasan ho jata hai.

Owner Change Karna

Kisi file ya directory ka Owner (malik) aur/ya Group assignment change karne ke liye hum **chown** command ka use kar sakte hain.

Iska syntax neeche diya gaya hai:

Syntax – chown

```
cry0l1t3@htb[/htb]$ chown <user>:<group> <file/directory>
```

Is example mein **shell** ki jagah kisi bhi file ya folder ka naam use kiya ja sakta hai.

```
cry0l1t3@htb[/htb]$ chown root:root shell && ls -l shell

-rwxr-xr--   1 root root 0 May  4 22:12 shell
```

Sticky Bit

Linux mein Sticky Bit files par ek lock ki tarah kaam karta hai, jo shared directories ke andar files ko protect karta hai.

Jab kisi directory par Sticky Bit set kiya jata hai, toh yeh security ki ek extra layer add karta hai. Yeh ensure karta hai ki sirf kuch specific users hi files ko modify ya delete kar sakein, bhale hi doosre users ke paas bhi us directory ka access ho.

Ek common workspace imagine karo, jahan bahut saare log enter kar sakte hain aur same tools use kar sakte hain, lekin har person ka apna ek drawer hai, jise sirf wahi person ya manager khol sakta hai.

Sticky Bit in drawers par lage lock ki tarah kaam karta hai, jo doosre logon ko unke contents ke saath chhed-chhad karne se rokta hai.

Ek shared directory mein iska matlab hai ki sirf file ka owner, directory ka owner ya root user (system administrator) hi files ko delete ya rename kar sakte hain. Doosre users abhi bhi directory ko access kar sakte hain, lekin un files ko modify nahi kar sakte jinke woh owner nahi hain.

Yeh feature shared environments mein bahut useful hota hai, jaise public directories, jahan multiple users ek saath kaam karte hain.

Sticky Bit set karne se important files galti se ya jaanbujhkar un logon ke dwara change nahi ki ja sakti, jinhe aisa karne ka adhikar nahi hai. Is tarah yeh collaborative workspaces ko secure rakhne mein madad karta hai.

```
cry0l1t3@htb[/htb]$ ls -l

drw-rw-r-t 3 cry0l1t3 cry0l1t3   4096 Jan 12 12:30 scripts
drw-rw-r-T 3 cry0l1t3 cry0l1t3   4096 Jan 12 12:32 reports
```

Is example mein hum dekh sakte hain ki dono directories par Sticky Bit set hai. Lekin **reports** folder mein uppercase `T` hai aur **scripts** folder mein lowercase **t** hai.

Agar Sticky Bit uppercase (**T**) mein hai, toh iska matlab hai ki baaki sabhi users ke paas **Execute** (**x**) permission nahi hai. Isliye woh folder ke contents nahi dekh sakte aur na hi us folder se koi program run kar sakte hain.

Lowercase Sticky Bit (**t**) ka matlab hai ki Sticky Bit ke saath **Execute** (**x**) permission bhi set hai.

User Management (User Management)

Effective User Management, Linux system administration ka ek fundamental (bunyadi) hissa hai.

Administrators ko aksar naye user accounts create karne padte hain ya existing users ko specific groups mein assign karna padta hai, taaki sahi access controls apply kiye ja sakein.

Iske alawa, alag permissions wale tasks ko complete karne ke liye kabhi-kabhi kisi doosre user ke roop mein commands execute karna bhi zaroori hota hai.

Example ke liye, kuch groups ko specific files ya directories ko dekhne ya modify karne ki special permissions hoti hain. Yeh system ki security aur integrity maintain karne ke liye zaroori hai.

Yeh capability humein machine ke andar locally aur bhi detailed information collect karne ki permission deti hai, jo troubleshooting ya auditing ke liye bahut important ho sakti hai.

Example:

Maan lo Alex naam ka ek naya employee tumhari company join karta hai aur use apne kaam karne ke liye Linux-based workstation diya jata hai.

Ek System Administrator ke roop mein, tumhe Alex ke liye ek user account create karna hoga aur use appropriate groups mein add karna hoga, taaki woh zaroori resources, jaise project files aur development tools ko access kar sake.

Iske alawa, kuch situations mein Alex ko apne tasks complete karne ke liye higher privileges ke saath ya kisi doosre user ke roop mein commands execute karne ki zaroorat bhi ho sakti hai.

Execution as a user

```
Toxicx01@htb[/htb]$ cat /etc/shadow

cat: /etc/shadow: Permission denied
```

/etc/shadow file ek bahut important system file hai, jo sabhi user accounts ke encrypted password ki information store karti hai.

Security reasons ki wajah se, is file ko sirf **root** user hi read aur write kar sakta hai, taaki sensitive authentication data ko unauthorized access se bachaya ja sake.

Aise tasks perform karne ke liye jinmein higher privileges ki zaroorat hoti hai, users **sudo** command ka use kar sakte hain.

sudo ka full form Superuser Do hai. Yeh command permitted users ko kisi doosre user ke security privileges ke saath commands execute karne ki permission deti hai. Usually yeh doosra user **superuser** ya **root** hota hai.

Isse users ko administrative tasks perform karne ke liye **root** user ke roop mein login karne ki zaroorat nahi padti.

System ki security maintain karne ke liye yeh ek best practice hai.

Hum **Linux Security** section mein **sudo** permissions ko aur detail mein samjhenge.

Execution as root

```
Toxicx01@htb[/htb]$ sudo cat /etc/shadow

root:<SNIP>:18395:0:99999:7:::
daemon:*:17737:0:99999:7:::
bin:*:17737:0:99999:7:::
<SNIP>
```

Yahan ek list di gayi hai, jo humein User Management ko behtar tareeke se samajhne aur usse deal karne mein madad karegi.

Command	Description
`sudo`	Execute command as a different user.
`su`	The `su` utility requests appropriate user credentials via PAM and switches to that user ID (the default user is the superuser). A shell is then executed.
`useradd`	Creates a new user or update default new user information.
`userdel`	Deletes a user account and related files.
`usermod`	Modifies a user account.
`addgroup`	Adds a group to the system.
`delgroup`	Removes a group from the system.
`passwd`	Changes user password.

User accounts, permissions aur authentication mechanisms kaise kaam karte hain, ise samajhne se hum vulnerabilities (kamzoriyan) identify kar sakte hain, misconfigurations ka fayda utha sakte hain aur kisi system ki security ka effectively assessment kar sakte hain.

User Management mein expertise hasil karne ka sabse effective tareeka hai ki ek controlled environment mein alag-alag commands aur unke various options ka use karke practice ki jaye.

Alag-alag commands ko try karne aur unki functionality explore karne ke liye free feel karo. Tum kya achieve karna chahte ho, yeh decide karne ke liye apni creativity ka use karo.

Jab tum in User Management tools ko pichhle sections mein seekhe gaye knowledge ke saath combine karoge, toh tumhe pata chalega ki tumne abhi tak kitna kuch seekh liya hai.

Linux system ke baare mein apni understanding ko practically apply karo:

- Naye user accounts create karo.
- In users ke liye files aur directories set up karo.
- Files select karo.
- Specific elements ko read aur filter karo.
- Unhe apne banaye hue naye users ki files aur directories mein redirect karo.

Jitna chaaho utna explore karo.

Tumhare target system mein aisi koi problem nahi hai jise fix nahi kiya ja sakta. Agar kuch galat ho bhi jata hai, toh tumhare paas target ko reset karke dobara shuru karne ka option hai.

Tab tak practice karo jab tak tum khud ko confident mehsoos na karne lago.

Q. Which option needs to be set to create a home directory for a new user using "useradd" command?

Ans.

Q. Which option needs to be set to lock a user account using the "usermod" command? (long version of the option)

Ans.

Q. Which option needs to be set to execute a command as a different user using the "su" command? (long version of the option)

Ans.

Package Management

Chahe hum System Administrator ke roop mein kaam kar rahe hon, apne ghar ke Linux machines ko maintain kar rahe hon, ya apni pasand ke Penetration Testing Distribution ko build, upgrade ya maintain kar rahe hon, Linux ke available Package Managers aur unka use karke packages ko Install, Update ya Remove karna achhe se samajhna bahut zaroori hai.

Packages aisi Archives hoti hain jinmein Software ke Binaries, Configuration Files, Dependencies ki Information aur Updates aur Upgrades ka record hota hai.

Zyadatar Package Management Systems ye features provide karte hain:

- Package downloading
- Dependency resolution
- Ek standard binary package format
- Installation aur configuration ke liye common locations
- System se related additional configuration aur functionality
- Quality control

Hum kai alag-alag Package Management Systems ka use kar sakte hain, jo different types ki files ko handle karte hain, jaise **.deb**, **.rpm** aur doosri files.

Package Management ke liye zaroori hai ki jo Software hum install karna chahte hain, woh uske corresponding Package ke roop mein available ho.

Aam taur par Linux Distributions ke under hi in Packages ko centrally create, provide aur maintain kiya jata hai.

Is tarah Software directly System ke saath integrate ho jata hai aur uski alag-alag directories System mein different locations par distribute ho jaati hain.

Package Management Software, Package ke andar se System mein installation ke liye zaroori changes ki information leta hai aur phir un changes ko apply karke Package ko successfully install karta hai.

Agar Package Management Software ko pata chalta hai ki kisi Package ko properly work karne ke liye kuch additional Packages ki zaroorat hai, jo abhi tak install nahi hue hain, toh ise Dependency kaha jata hai.

Aisi situation mein Package Management Software ya toh Administrator ko warning deta hai, ya Repository se missing Software ko dobara download karke pehle install karne ki koshish karta hai.

Agar kisi installed Software ko delete kar diya jata hai, toh Package Management System us Package ki information dobara leta hai, apni Configuration ke according usmein changes karta hai aur Files ko delete kar deta hai.

Is kaam ke liye hum alag-alag Package Management Programs ka use kar sakte hain.

Yahan aise hi kuch Programs ki list di gayi hai:

Command	Description
dpkg	**dpkg** ek tool hai jo Debian packages ko Install, Build, Remove aur Manage karne ke kaam aata hai. **dpkg** ka main aur zyada user-friendly front-end **aptitude** hai.
apt	**apt** Package Management System ke liye ek high-level Command-Line Interface provide karta hai.
aptitude	**aptitude**, **apt** ka ek alternative hai aur Package Manager ke liye ek high-level interface provide karta hai.

snap	**snap** packages ko Install, Configure, Refresh aur Remove karne ke kaam aata hai. Snaps ki help se Cloud, Servers, Desktops aur Internet of Things (IoT) ke liye latest Apps aur Utilities ko securely distribute kiya ja sakta hai.
gem	**gem**, RubyGems ka front-end hai. RubyGems, Ruby programming language ka standard Package Manager hai.
pip	**pip** ek Python Package Installer hai. Iska use un Python packages ko install karne ke liye recommend kiya jata hai jo Debian Archive mein available nahi hote. Ye Version Control Repositories ke saath kaam kar sakta hai (filhaal Git, Mercurial aur Bazaar repositories). Ye output ka detailed record rakhta hai aur installation shuru karne se pehle saari requirements download karke incomplete installation ko rokta hai.
git	**Git** ek fast, scalable aur distributed Revision Control System hai. Ismein bahut saare commands hote hain, jo high-level operations ke saath-saath system ke internal parts tak full access provide karte hain.

Experiment karne ke liye apni Virtual Machine (VM) ko locally setup karna highly recommended hai.

Chaliye, apni local VM mein thoda experiment karte hain aur kuch additional packages install karke ise extend karte hain.

Sabse pehle, **apt** ka use karke **git** install karte hai

Advanced Package Manager (APT)

Debian-based Linux Distributions mein **APT** Package Manager ka use hota hai.

Ek Package ek Archive File hoti hai, jismein multiple **.deb** files hoti hain.

dpkg utility ka use related **.deb** file se Programs ko install karne ke liye kiya jata hai.

APT Programs ko Update aur Install karna aasan bana deta hai, kyunki bahut saare Programs ki Dependencies hoti hain.

Jab hum kisi standalone **.deb** file se Program install karte hain, toh humein Dependency se related problems aa sakti hain. Aisi situation mein humein ek ya multiple additional Packages download aur install karne pad sakte hain.

APT is process ko aasan aur efficient bana deta hai, kyunki ye Program ko install karne ke liye zaroori saari Dependencies ko ek saath manage karta hai.

Har Linux Distribution Software Repositories ka use karti hai, jinhe frequently update kiya jata hai.

Jab hum kisi Program ko Update karte hain ya naya Program Install karte hain, toh System required Package ko search karne ke liye in Repositories ko check karta hai.

Repositories ko alag-alag categories mein divide kiya ja sakta hai:

- Stable
- Testing
- Unstable

Zyadatar Linux Distributions sabse stable ya Main Repository ka use karti hain.

Ise check karne ke liye hum **/etc/apt/sources.list** file ka content dekh sakte hain.

Parrot OS ke liye Repository List yahan hoti hai:

/etc/apt/sources.list.d/parrot.list

```shellsession
Toxicx01@htb[/htb]$ cat /etc/apt/sources.list.d/parrot.list

# parrot repository
# this file was automatically generated by parrot-mirror-selector
deb http://htb.deb.parrot.sh/parrot/ rolling main contrib non-free
#deb-src https://deb.parrot.sh/parrot/ rolling main contrib non-free
deb http://htb.deb.parrot.sh/parrot/ rolling-security main contrib non-free
#deb-src https://deb.parrot.sh/parrot/ rolling-security main contrib non-free
```

APT ek database ka use karta hai, jise APT Cache kaha jata hai.

Iska use hamare System mein installed Packages ke baare mein information provide karne ke liye kiya jata hai, bina Internet ke bhi.

Hum APT Cache mein search kar sakte hain. Example ke liye, hum **Impacket** se related sabhi Packages ko find kar sakte hain.

```shellsession
Toxicx01@htb[/htb]$ apt-cache search impacket

impacket-scripts - Links to useful impacket scripts examples
polenum - Extracts the password policy from a Windows system
python-pcapy - Python interface to the libpcap packet capture library (Python 2)
python3-impacket - Python3 module to easily build and dissect network protocols
python3-pcapy - Python interface to the libpcap packet capture library (Python 3)
```

Iske baad hum kisi Package ke baare mein additional information bhi dekh sakte hain.

```
Toxicx01@htb[/htb]$ apt-cache show impacket-scripts

Package: impacket-scripts
Version: 1.4
Architecture: all
Maintainer: Kali Developers <devel@kali.org>
Installed-Size: 13
Depends: python3-impacket (>= 0.9.20), python3-ldap3 (>= 2.5.0), python3-
ldapdomaindump
Breaks: python-impacket (<< 0.9.18)
Replaces: python-impacket (<< 0.9.18)
Priority: optional
Section: misc
Filename: pool/main/i/impacket-scripts/impacket-scripts_1.4_all.deb
Size: 2080
<SNIP>
```

Hum apne System mein installed sabhi Packages ki list bhi dekh sakte hain.

```
Toxicx01@htb[/htb]$ apt list --installed

Listing... Done
accountsservice/rolling,now 0.6.55-2 amd64 [installed,automatic]
adapta-gtk-theme/rolling,now 3.95.0.11-1 all [installed]
adduser/rolling,now 3.118 all [installed]
adwaita-icon-theme/rolling,now 3.36.1-2 all [installed,automatic]
aircrack-ng/rolling,now 1:1.6-4 amd64 [installed,automatic]
<SNIP>
```

Agar hamare System mein kuch Packages missing hain, toh hum unhe search kar sakte hain aur neeche diye gaye command ka use karke install kar sakte hain.

```shellsession
Toxicx01@htb[/htb]$ sudo apt install impacket-scripts -y

Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  impacket-scripts
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 2,080 B of archives.
After this operation, 13.3 kB of additional disk space will be used.
Get:1 https://euro2-emea-mirror.parrot.sh/mirrors/parrot rolling/main amd64
impacket-scripts all 1.4 [2,080 B]
Fetched 2,080 B in 0s (15.2 kB/s)
Selecting previously unselected package impacket-scripts.
(Reading database ... 378459 files and directories currently installed.)
Preparing to unpack .../impacket-scripts_1.4_all.deb ...
Unpacking impacket-scripts (1.4) ...
Setting up impacket-scripts (1.4) ...
Scanning application launchers
Removing duplicate launchers from Debian
Launchers are updated
```

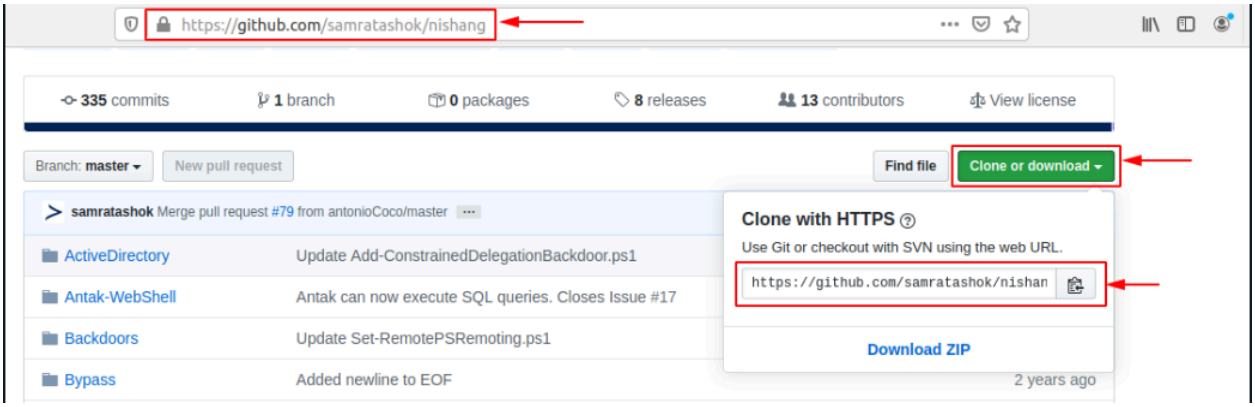
Git

Ab jab humne **git** install kar liya hai, toh hum iska use Github se useful tools download karne ke liye kar sakte hain.

Aisa hi ek project 'Nishang' ke naam se hai. Hum baad mein is project ke saath kaam karenge aur ise detail mein samjhenge.

Sabse pehle, humein **Nishang ke GitHub Repository** par jana hoga aur uska GitHub link copy karna hoga.

Iske baad git ka use karke hum ise download kar sakte hain.



Phir bhi, Project aur uski Scripts aur Lists ko download karne se pehle, humein ek alag folder create karna chahiye.

```
Toxicx01@htb[/htb]$ mkdir ~/nishang/ && git clone
https://github.com/samratashok/nishang.git ~/nishang

Cloning into '/opt/nishang/'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 1691 (delta 4), reused 6 (delta 2), pack-reused 1676
Receiving objects: 100% (1691/1691), 7.84 MiB | 4.86 MiB/s, done.
Resolving deltas: 100% (1055/1055), done.
```

DPKG

Hum Repositories se Programs aur Tools ko alag-alag bhi download kar sakte hain.

Is example mein, hum Ubuntu 18.04 LTS ke liye **strace** download karte hain.

```
shellsession
Toxicx01@htb[/htb]$ wget
http://archive.ubuntu.com/ubuntu/pool/main/s/strace/strace_4.21-1ubuntu1_amd64.deb

--2020-05-15 03:27:17--
http://archive.ubuntu.com/ubuntu/pool/main/s/strace/strace_4.21-1ubuntu1_amd64.deb
Resolving archive.ubuntu.com (archive.ubuntu.com)... 91.189.88.142, 91.189.88.152,
2001:67c:1562::18, ...
Connecting to archive.ubuntu.com (archive.ubuntu.com)|91.189.88.142|:80...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 333388 (326K) [application/x-debian-package]
Saving to: ‘strace_4.21-1ubuntu1_amd64.deb’

strace_4.21-1ubuntu1_amd64.deb       100%
[=====================================================================>] 325,57K  -
-.-KB/s    in 0,1s

2020-05-15 03:27:18 (2,69 MB/s) - ‘strace_4.21-1ubuntu1_amd64.deb’ saved
[333388/333388]
```

Ab hum Package ko install karne ke liye **apt** aur **dpkg** dono ka use kar sakte hain.

Kyuki hum pehle hi **apt** ke saath kaam kar chuke hain, isliye agle example mein hum **dpkg** ka use karenge.

```
Toxicx01@htb[/htb]$ sudo dpkg -i strace_4.21-1ubuntu1_amd64.deb

(Reading database ... 154680 files and directories currently installed.)
Preparing to unpack strace_4.21-1ubuntu1_amd64.deb ...
Unpacking strace (4.21-1ubuntu1) over (4.21-1ubuntu1) ...
Setting up strace (4.21-1ubuntu1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
```

Iske saath, humne tool ko successfully install kar liya hai aur ab check kar sakte hain ki kya ye properly kaam kar raha hai.

shellsession

```
Toxicx01@htb[/htb]$ strace -h

usage: strace [-CdffhiqrtttTvVwxxy] [-I n] [-e expr]...
              [-a column] [-o file] [-s strsize] [-P path]...
              -p pid... / [-D] [-E var=val]... [-u username] PROG [ARGS]
   or: strace -c[dfw] [-I n] [-e expr]... [-O overhead] [-S sortby]
              -p pid... / [-D] [-E var=val]... [-u username] PROG [ARGS]

Output format:
  -a column      alignment COLUMN for printing syscall results (default 40)
  -i             print instruction pointer at time of syscall
```

Service and Process Management

Services, jinhe Daemons bhi kaha jata hai, Linux System ke basic components hote hain. Ye background mein bina kisi direct user interaction ke silently run karte hain.

Ye important tasks perform karte hain, jo System ko properly chalu rakhte hain aur additional functionalities provide karte hain.

Generally, Services ko do types mein divide kiya ja sakta hai:

System Services

Ye internal services hoti hain jo System Startup ke time zaroori hoti hain.

Ye important hardware-related tasks perform karti hain aur Operating System ko properly work karne ke liye zaroori System Components ko initialize karti hain.

Ye car ke Engine aur Transmission System ki tarah hoti hain.

Jaise hi aap car ki ignition key turn karte ho, ye start ho jate hain aur car ko chalane ke liye zaroori hote hain.

Inke bina car nahi chal payegi.

User-Installed Services

Ye services users ke dwara add ki jati hain. Inmein generally Server Applications aur doosre Background Processes shamil hote hain, jo specific features ya capabilities provide karte hain.

Ye services car ke Air Conditioning ya GPS Navigation System ki tarah hoti hain.

Car ko chalane ke liye ye zaroori nahi hain, lekin ye functionality ko improve karti hain aur driver ki preferences ke according additional features provide karti hain.

Daemons ko aksar unke Program Names ke end mein `d` letter se identify kiya jata hai.

Examples:

- **sshd** (SSH daemon)
- **systemd**

Jaise ek car apna complete experience provide karne ke liye apne main components aur optional features dono par depend karti hai, waise hi Linux System efficiently work karne aur users ki needs ko पूरा karne ke liye System Services aur User-Installed Services dono ka use karta hai.

Generally, jab hum kisi Service ya Process ke saath kaam karte hain, toh hamare kuch main goals hote hain:

1. Kisi Service/Process ko Start ya Restart karna.
2. Kisi Service/Process ko Stop karna.
3. Dekhna ki kisi Service/Process mein kya ho raha hai ya pehle kya hua tha.
4. Boot ke time kisi Service/Process ko Enable ya Disable karna.
5. Kisi Service/Process ko Find karna.

Zyadatar modern Linux Distributions ne **systemd** ko apne Initialization System (Init System) ke roop mein adopt kar liya hai.

Boot Process ke dauran sabse pehle start hone wala Process **systemd** hota hai aur ise ek Process ID (**PID**) assign ki jati hai.

Linux System ke sabhi Processes ko ek **PID** assign ki jati hai aur unhe **/proc/** Directory ke andar dekha ja sakta hai.

/proc/ Directory mein har Process ke baare mein information hoti hai.

Processes ki ek Parent Process ID (**PPID**) bhi ho sakti hai, jo batati hai ki unhe kisi doosre Process (Parent Process) ne start kiya tha.

Aise Processes ko Child Processes kaha jata hai.

Systemctl

Apni VM par **OpenSSH** install karne ke baad, hum neeche diye gaye command ka use karke iski Service ko Start kar sakte hain.

```
Toxicx01@htb[/htb]$ systemctl start ssh
```

Service ko Start karne ke baad, ab hum check kar sakte hain ki kya ye bina kisi error ke run ho rahi hai.

```shellsession
Toxicx01@htb[/htb]$ systemctl status ssh

● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset:
enabled)
   Active: active (running) since Thu 2020-05-14 15:08:23 CEST; 24h ago
   Main PID: 846 (sshd)
   Tasks: 1 (limit: 4681)
   CGroup: /system.slice/ssh.service
           └─846 /usr/sbin/sshd -D

Mai 14 15:08:22 inlane systemd[1]: Starting OpenBSD Secure Shell server...
Mai 14 15:08:23 inlane sshd[846]: Server listening on 0.0.0.0 port 22.
Mai 14 15:08:23 inlane sshd[846]: Server listening on :: port 22.
Mai 14 15:08:23 inlane systemd[1]: Started OpenBSD Secure Shell server.
Mai 14 15:08:30 inlane systemd[1]: Reloading OpenBSD Secure Shell server.
Mai 14 15:08:31 inlane sshd[846]: Received SIGHUP; restarting.
Mai 14 15:08:31 inlane sshd[846]: Server listening on 0.0.0.0 port 22.
Mai 14 15:08:31 inlane sshd[846]: Server listening on :: port 22.
```

OpenSSH ko SysV Script mein add karne ke liye, taaki System Startup ke baad is Service ko run karne ka instruction mil sake, hum neeche diye gaye command se ise link kar sakte hain:

```
Toxicx01@htb[/htb]$ systemctl enable ssh

Synchronizing state of ssh.service with SysV service script with
/lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable ssh
```

Jab hum System ko Reboot karenge, toh OpenSSH Server automatically run ho jayega.

Hum ise **ps** naam ke ek Tool ki help se check kar sakte hain.

```
shellsession
Toxicx01@htb[/htb]$ ps -aux | grep ssh

root       846  0.0  0.1  72300  5660 ?        Ss   Mai14   0:00 /usr/sbin/sshd -D
```

Hum **systemctl** ka use karke sabhi Services ki list bhi dekh sakte hain.

```
Toxicx01@htb[/htb]$ systemctl list-units --type=service

UNIT                                                    LOAD   ACTIVE SUB
DESCRIPTION
accounts-daemon.service                                 loaded active running
Accounts Service
acpid.service                                           loaded active running
ACPI event daemon
apache2.service                                         loaded active running
The Apache HTTP Server
apparmor.service                                        loaded active exited
AppArmor initialization
apport.service                                          loaded active exited
LSB: automatic crash repor
avahi-daemon.service                                    loaded active running
Avahi mDNS/DNS-SD Stack
bolt.service                                            loaded active running
```

Ho sakta hai kisi error ki wajah se Services Start na ho paayein.
Problem ka pata lagane ke liye hum journalctl tool ka use karke Logs dekh sakte hain.

```
Toxicx01@htb[/htb]$ journalctl -u ssh.service --no-pager

-- Logs begin at Wed 2020-05-13 17:30:52 CEST, end at Fri 2020-05-15 16:00:14 CEST.
--
Mai 13 20:38:44 inlane systemd[1]: Starting OpenBSD Secure Shell server...
Mai 13 20:38:44 inlane sshd[2722]: Server listening on 0.0.0.0 port 22.
Mai 13 20:38:44 inlane sshd[2722]: Server listening on :: port 22.
Mai 13 20:38:44 inlane systemd[1]: Started OpenBSD Secure Shell server.
Mai 13 20:39:06 inlane sshd[3939]: Connection closed by 10.22.2.1 port 36444
[preauth]
Mai 13 20:39:27 inlane sshd[3942]: Accepted password for master from 10.22.2.1 port
36452 ssh2
Mai 13 20:39:27 inlane sshd[3942]: pam_unix(sshd:session): session opened for user
master by (uid=0)
Mai 13 20:39:28 inlane sshd[3942]: pam_unix(sshd:session): session closed for user
master
Mai 14 02:04:49 inlane sshd[2722]: Received signal 15; terminating.
```

Kill a Process

Ek Process ki following states ho sakti hain:

- Running: Process abhi run ho raha hai.
- Waiting: Process kisi Event ya System Resource ka wait kar raha hai.
- Stopped: Process ko rok diya gaya hai.
- Zombie: Process stop ho chuka hai, lekin abhi bhi Process Table mein uski entry मौजूद hai.

Processes ko control karne ke liye hum **kill**, **pkill**, **pgrep** aur **killall** commands ka use kar sakte hain.

Kisi Process ke saath interact karne ke liye humein us Process ko ek Signal send karna hota hai.

Hum neeche diye gaye command ka use karke sabhi Signals dekh sakte hain:

```shellsession
Toxicx01@htb[/htb]$ kill -l

 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL       5) SIGTRAP
 6) SIGABRT      7) SIGBUS       8) SIGFPE       9) SIGKILL     10) SIGUSR1
11) SIGSEGV     12) SIGUSR2     13) SIGPIPE     14) SIGALRM     15) SIGTERM
16) SIGSTKFLT   17) SIGCHLD     18) SIGCONT     19) SIGSTOP     20) SIGTSTP
21) SIGTTIN     22) SIGTTOU     23) SIGURG      24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGIO       30) SIGPWR
31) SIGSYS      34) SIGRTMIN    35) SIGRTMIN+1  36) SIGRTMIN+2  37) SIGRTMIN+3
38) SIGRTMIN+4  39) SIGRTMIN+5  40) SIGRTMIN+6  41) SIGRTMIN+7  42) SIGRTMIN+8
43) SIGRTMIN+9  44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
58) SIGRTMAX-6  59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2
63) SIGRTMAX-1  64) SIGRTMAX
```

Sabse zyada use kiye jaane wale Signals hain:

Signal	Description
1	**SIGHUP - Jab kisi Process ko control karne wala Terminal close ho jata hai, toh ye Signal us Process ko send hota hai.**

2	**SIGINT - Jab user controlling Terminal mein [Ctrl] + C press karta hai, toh Process ko interrupt karne ke liye ye Signal send hota hai.**
3	**SIGQUIT - Jab user [Ctrl] + D press karta hai, toh Process ko quit karne ke liye ye Signal send hota hai.**
9	**SIGKILL - Ye bina kisi Clean-up Operation ke Process ko turant kill kar deta hai.**
15	**SIGTERM - Program ko terminate karne ke liye use hota hai.**
19	**SIGSTOP - Program ko stop kar deta hai. Is Signal ko handle nahi kiya ja sakta.**
20	**SIGTSTP - Jab user [Ctrl] + Z press karta hai, toh Service ko suspend karne ke liye ye Signal send hota hai. User baad mein ise handle kar sakta hai.**

Example ke liye, agar koi Program Freeze (hang) ho jata hai, toh hum neeche diye gaye Command ka use karke use forcefully Kill kar sakte hain

---

Background a Process (Process ko Background mein chalana)

Kabhi-kabhi humein kisi scan ya process ko background mein bhejna padta hai, taaki hum current terminal session ka use karke system ke saath interact kar sakein ya doosre processes start kar sakein.

Jaise ki hum pehle dekh chuke hain, hum ye kaam **[Ctrl + Z]** shortcut se kar sakte hain.

Jaisa ki upar bataya gaya hai, **[Ctrl + Z]** dabane par hum kernel ko **SIGTSTP** signal bhejte hain, jo us process ko temporarily suspend (rok) deta hai.

```
Toxicx01@htb[/htb]$ ping -c 10 www.hackthebox.eu

PING www.hackthebox.eu (104.20.55.68) 56(84) bytes of data.
[Ctrl + Z]
[1]+  Stopped                 ping -c 10 www.hackthebox.eu

Toxicx01@htb[/htb]$ vim tmpfile
[Ctrl + Z]
[2]+  Stopped                 vim tmpfile
```

Ab hum following command ka use karke sabhi background processes ko dekh sakte hain.

```
Toxicx01@htb[/htb]$ jobs

[1]+  Stopped                 ping -c 10 www.hackthebox.eu
[2]+  Stopped                 vim tmpfile
```

[Ctrl] + Z shortcut process ko suspend (rok) deta hai, aur process aage execute nahi hota.

Agar humein process ko background mein chalu rakhna hai, toh humein **bg** command enter karni hogi.

```
Toxicx01@htb[/htb]$ bg

[1]+ ping -c 10 www.hackthebox.eu &

Toxicx01@htb[/htb]$
--- www.hackthebox.eu ping statistics ---
10 packets transmitted, 0 received, 100% packet loss, time 113482ms

[ENTER]
[1]+  Exit 1                  ping -c 10 www.hackthebox.eu
```

Ek aur option hai process ko automatically background mein run karne ka.

Iske liye command ke end mein AND sign **(&)** lagana hota hai.

Isse process start hote hi background mein run hone lagta hai, aur hum terminal mein doosre commands bhi chala sakte hain.

```
Toxicx01@htb[/htb]$ ping -c 10 www.hackthebox.eu &

[1] 10825
PING www.hackthebox.eu (172.67.1.1) 56(84) bytes of data.
```

Jab process complete (khatam) ho jayega, tab humein uske results dikhai denge.

```
Toxicx01@htb[/htb]$

--- www.hackthebox.eu ping statistics ---
10 packets transmitted, 0 received, 100% packet loss, time 9210ms

[ENTER]
[1]+  Exit 1                  ping -c 10 www.hackthebox.eu
```

Foreground a Process

Iske baad, hum jobs command ka use karke sabhi background processes ki list dekh sakte hain.

Background mein chal rahe processes ko user interaction ki zaroorat nahi hoti. Isliye hum process ke khatam hone ka wait kiye bina usi terminal session mein doosre commands chala sakte hain.

Jab scan ya process apna kaam complete kar lega, toh terminal humein notification dega ki process finish ho gaya hai.

```
Toxicx01@htb[/htb]$ jobs

[1]+  Running                 ping -c 10 www.hackthebox.eu &
```

Agar hum kisi background process ko wapas foreground mein lana chahte hain aur uske saath dobara interact karna chahte hain, toh hum **fg <ID>** command ka use kar sakte hain.

```
Toxicx01@htb[/htb]$ fg 1
ping -c 10 www.hackthebox.eu

--- www.hackthebox.eu ping statistics ---
10 packets transmitted, 0 received, 100% packet loss, time 9206ms
```

Execute Multiple Commands

Ek ke baad ek multiple commands chalane ke 3 tarike hain:

- Semicolon (;)
- Double Ampersand (&&)
- Pipe (|)

In teeno ke beech difference is baat par depend karta hai ki previous command ko kaise handle kiya jata hai aur kya previous command successfully complete hui thi ya usmein koi error aaya tha.

Semicolon (;) ek command separator hai. Ye previous command ke result aur errors ko ignore karke agali commands execute karta hai.

```
Toxicx01@htb[/htb]$ echo '1'; echo '2'; echo '3'

1
2
3
```

Example ke liye, agar hum wahi command execute karte hain, lekin doosri jagah ls command ko ek aisi file se replace kar dete hain jo exist (maujood) nahi karti, toh humein error milega.

Phir bhi, teesri command execute hogi.

```
Toxicx01@htb[/htb]$ echo '1'; ls MISSING_FILE; echo '3'

1
ls: cannot access 'MISSING_FILE': No such file or directory
3
```

Lekin agar hum commands ko ek ke baad ek execute karne ke liye Double AND (&&) ka use karte hain, toh result alag hota hai.

Agar kisi ek command mein error aata hai, toh uske baad wali commands execute nahi hongi aur poora process wahin ruk jayega.

```
Toxicx01@htb[/htb]$ echo '1' && ls MISSING_FILE && echo '3'

1
ls: cannot access 'MISSING_FILE': No such file or directory
```

Pipe (|) sirf previous commands ke successfully execute hone ya error-free hone par hi depend nahi karta, balki previous commands ke output (results) par bhi depend karta hai.

Q. Use the "systemctl" command to list all units of services and submit the unit name with the description "Load AppArmor profiles managed internally by snapd" as the answer.
Ans.

Task Scheduling

Linux systems mein Task Scheduling ek important feature hai, jo users aur administrators ko tasks automatically run karne ki permission deta hai. Ismein tasks ko kisi specific time par ya regular intervals par run karne ke liye set kiya ja sakta hai. Isse baar-baar manually tasks start karne ki zaroorat nahi padti.

Ubuntu, Red Hat Linux aur Solaris jaise operating systems mein ye functionality available hai. Iske through kai tarah ke tasks manage kiye ja sakte hain, jaise automatic software updates, scripts execute karna, database maintenance aur backup automatically lena.

Regular aur baar-baar hone wale tasks ko schedule karne se ye ensure hota hai ki woh consistently aur reliably perform hon. Iske alawa, kuch specific events hone par administrators ya users ko notify karne ke liye alerts bhi configure kiye ja sakte hain. Is tarah ke automation ke bahut saare uses hain, lekin ye examples sabse common hain.

Linux mein Task Scheduling ko samajhna ek cybersecurity specialist aur penetration tester ke liye zaroori hai, kyunki iska use legitimate administrative work ke saath-saath malicious activities ke liye bhi kiya ja sakta hai.

Automated tasks kaise kaam karte hain, ye jaankar hum potential security risks identify kar sakte hain. Jaise unauthorized cron jobs jo harmful scripts execute karte hain ya scheduled intervals par persistent backdoors ko maintain karte hain.

Task Scheduling ki working ko achhe se samajhkar hum hidden threats ko detect aur analyze kar sakte hain, system security audits ko improve kar sakte hain aur penetration testing ke dauran scheduled tasks ka use karke attack scenarios simulate bhi kar sakte hain.

Task Scheduling ka example:

Task Scheduling ko hum coffee ya tea maker ke example se samajh sakte hain.

Jaise hum coffee ya tea maker ko har subah ek fix time par automatically coffee banane ke liye set kar dete hain. Ek baar program karne ke baad, woh bina kisi manual help ke fix time par coffee ya tea bana deta hai. Isse jab humein zaroorat hoti hai, tab fresh coffee ready milti hai.

Systemd

Systemd Linux systems jaise Ubuntu, Red Hat Linux aur Solaris mein use hone wali ek service hai, jo processes aur scripts ko kisi specific time par start karne ke liye use hoti hai.

Iski help se hum processes aur scripts ko kisi fix time ya time interval par run karne ke liye set kar sakte hain. Iske alawa, hum specific events aur triggers bhi set kar sakte hain, jo kisi particular task ko start karenge.

Aisa karne ke liye, humein kuch steps follow karne hote hain aur kuch precautions lene hote hain, taaki system hamare scripts ya processes ko automatically execute kar sake.

1. Create a timer: Ye schedule karta hai ki mytimer.service kab run honi chahiye.
2. Create a service: Ye commands ya script ko execute karti hai.
3. Activate the timer: Timer ko activate karta hai.

Create a Timer

Systemd ke liye timer create karne ke liye, humein ek directory (folder) create karni hogi, jahan timer script store ki jayegi.

```
Toxicx01@htb[/htb]$ sudo mkdir /etc/systemd/system/mytimer.timer.d
Toxicx01@htb[/htb]$ sudo vim /etc/systemd/system/mytimer.timer
```

Iske baad, humein ek script banani hogi jo timer ko configure karti hai.

Is script mein ye options hone chahiye:

1. Unit: Ye timer ka description (vivaran) specify karta hai.

2. Timer: Ye specify karta hai ki timer kab start hoga aur kab activate hoga.

3. Install: Ye specify karta hai ki timer ko kahan install karna hai.

Mytimer.timer

```
[Unit]
Description=My Timer

[Timer]
OnBootSec=3min
OnUnitActiveSec=1hour

[Install]
WantedBy=timers.target
```

Yahan ye depend karta hai ki hum apni script ka use kis tarah karna chahte hain.

Example ke liye:

Agar hum chahte hain ki system boot (start) hone ke baad hamari script sirf ek baar run ho, toh humein **`Timer`** mein **`OnBootSec`** setting ka use karna chahiye.

Lekin agar hum chahte hain ki hamari script regularly run ho, toh humein **`OnUnitActiveSec`** ka use karna chahiye. Isse system hamari script ko regular intervals par run karega.

Iske baad, humein apni **`service`** create karni hogi.

Create a Service

```
Toxicx01@htb[/htb]$ sudo vim /etc/systemd/system/mytimer.service
```

Yahan hum ek description set karte hain aur us script ka complete path specify karte hain, jise hum run karna chahte hain.

`multi-user.target` ek unit system hai, jo tab activate hota hai jab system normal multi-user mode mein start hota hai.

Ye define karta hai ki system ke normal startup ke time kaun-kaun si services start honi chahiye.

```
[Unit]
Description=My Service

[Service]
ExecStart=/full/path/to/my/script.sh

[Install]
WantedBy=multi-user.target
```

Iske baad, humein **systemd** ko folders dobara read karne dena hoga, taaki woh kiye gaye changes ko include kar sake.

Reload Systemd

```
Toxicx01@htb[/htb]$ sudo systemctl daemon-reload
```

Iske baad, hum **systemctl** ka use karke service ko manually **start** kar sakte hain aur uske automatic startup ke liye **enable** kar sakte hain.

Start the Timer & Service

```
Toxicx01@htb[/htb]$ sudo systemctl start mytimer.timer
Toxicx01@htb[/htb]$ sudo systemctl enable mytimer.timer
```

Is tarah, **mytimer.service** automatically launch hogi, un intervals (time ke gap) ya delays ke according, jo humne **mytimer.timer** mein set kiye hain.

Cron

Cron ek aur tool hai jiska use Linux systems mein processes ko schedule aur automatically run karne ke liye kiya ja sakta hai.

Ye users aur administrators ko kisi task ko ek specific time ya fixed time intervals par execute karne ki permission deta hai.

Upar diye gaye examples ke liye bhi hum Cron ka use karke wahi tasks automatically run karwa sakte hain. Iske liye humein bas ek script banani hoti hai aur phir cron daemon ko batana hota hai ki us script ko kis specific time par run karna hai.

Cron ke through hum wahi tasks automate kar sakte hain, lekin Cron daemon ko set up karne ka process Systemd se thoda alag hota hai.

Cron daemon ko set up karne ke liye humein tasks ko crontab naam ki file mein store karna hota hai. Uske baad daemon ko batana hota hai ki un tasks ko kab run karna hai.
Phir hum cron daemon ko configure karke tasks ko schedule aur automatically run karwa sakte hain.

Cron ki structure mein neeche diye gaye components hote hain:

Time Frame	Description
Minutes (0-59)	Isse specify hota hai ki task kis minute par execute hoga.
Hours (0-23)	Isse specify hota hai ki task kis hour par execute hoga.
Days of month (1-31)	Isse specify hota hai ki month ki kis date ko task execute hoga.
Months (1-12)	Isse specify hota hai ki kis month mein task execute hoga.
Days of the week (0-7)	Isse specify hota hai ki week ke kis din task execute hoga.

Example ke liye, crontab aisa dikh sakta hai:-

```
# System Update
0 */6 * * * /path/to/update_software.sh

# Execute Scripts
0 0 1 * * /path/to/scripts/run_scripts.sh

# Cleanup DB
0 0 * * 0 /path/to/scripts/clean_database.sh

# Backups
0 0 * * 7 /path/to/scripts/backup.sh
```

Pehla task, **System Update**, har 6 ghante mein ek baar execute hona chahiye. Iske liye hour column mein **0 */6** diya gaya hai. Ye task **update_software.sh** script ke through execute hota hai, jiska path last column mein diya gaya hai.

Doosra task, **Execute Scripts**, har month ki pehli date ko raat 12 baje execute hona chahiye. Iske liye minute aur hour columns mein **0** aur **0** diya gaya hai aur days-of-the-month column mein 1 diya gaya hai. Ye task **run_scripts.sh** script ke through execute hota hai, jiska path last column mein diya gaya hai.

Teesra task, **Cleanup DB**, har Sunday ko raat 12 baje execute hona chahiye. Iske liye minute aur hour columns mein **0** aur **0** diya gaya hai aur days-of-the-week column mein 0 diya gaya hai. Ye task **clean_database.sh** script ke through execute hota hai, jiska path last column mein diya gaya hai.

Chautha task, **Backups**, har Sunday ko raat 12 baje execute hona chahiye. Iske liye minute aur hour columns mein **0** aur **0** diya gaya hai aur days-of-the-week column mein 7 diya gaya hai. Ye task **backup.sh** script ke through execute hota hai, jiska path last column mein diya gaya hai.

Hum task successfully ya unsuccessfully execute hone par notifications bhi receive kar sakte hain.

Iske alawa, tasks kab aur kaise execute ho rahe hain, isko monitor karne ke liye logs bhi create kar sakte hain.

Systemd vs. Cron

Systemd aur Cron dono Linux ke tools hain, jinka use tasks ko schedule aur automatically run karne ke liye kiya jata hai.

In dono tools ke beech main difference inke setup karne ke tareeke mein hai.

Systemd: Isme humein ek Timer aur Service script create karni hoti hai, jo operating system ko batati hai ki tasks kab run karne hain.

Cron: Isme humein ek `crontab` file create karni hoti hai, jo cron daemon ko batati hai ki tasks kab run karne hain.

Q. What is the Type of the service of the "dconf.service"?

Ans.

Network Services

Linux par kaam karte waqt alag-alag network services ko manage karna zaroori hota hai.

In services ko handle karna seekhna important hai, kyunki ye alag-alag kaam karti hain. Inmein se kai services humein door baithe kisi doosre computer par kaam karne ki suvidha deti hain.

Humein ye pata hona chahiye ki network ke through doosre computers se kaise communicate karna hai, connection kaise banana hai, files kaise transfer karni hain, network traffic kaise check karna hai aur services ko kaise configure karna hai.

Ye knowledge penetration testing ke dauran possible vulnerabilities (kamzoriyan) identify karne mein help karti hai.

Iske alawa, har service ki configuration settings ko samajhne se humein network security ko behtar tareeke se samajhne mein help milti hai.

Example:-

Maan lo hum penetration testing kar rahe hain aur humein ek Linux host milta hai, jismein hum vulnerabilities check kar rahe hain.

Network traffic ko monitor karte waqt hum dekhte hain ki Linux host ka ek user, bina encryption wale FTP server ke through doosre server se connect ho raha hai.

Is wajah se hum user ke login credentials ko plain text mein capture kar sakte hain.

Agar user ko pata hota ki FTP connection aur transfer hone wale data ko encrypt nahi karta, toh aisi situation hone ki possibility bahut kam hoti.

Linux administrator ke liye ye ek badi galti hai, kyunki isse network ki security weaknesses saamne aati hain. Saath hi, ye bhi pata chalta hai ki network security maintain karne wale administrators ne apni responsibility sahi se nahi nibhayi.

Har network service ko cover karna possible nahi hai, isliye hum sirf sabse important services par focus karenge.

Ye approach sirf administrators aur users ke liye hi nahi, balki penetration testers ke liye bhi useful hai. Unhe samajhna zaroori hai ki alag-alag hosts aur unke apne systems ek-doosre ke saath kaise communicate karte hain.

SSH

Secure Shell (**SSH**) ek network protocol hai, jo network ke through data aur commands ko safely bhejne ki permission deta hai.

Iska use door baithe doosre systems ko securely manage karne, un par commands chalane aur files transfer karne ke liye hota hai.

Apne ya kisi remote Linux system se SSH ke through connect hone ke liye, us system par SSH server available aur running hona chahiye.

Sabse zyada use hone wala SSH server OpenSSH server hai.

OpenSSH, Secure Shell (SSH) protocol ka ek free aur open-source version hai, jo network ke through data aur commands ko securely transfer karta hai.

Administrators OpenSSH ka use encrypted connection banakar remote systems ko securely manage karne ke liye karte hain.

OpenSSH ki help se administrators remote systems par commands chala sakte hain, files securely transfer kar sakte hain aur secure remote connection bana sakte hain.

Is tarah data aur commands ko third parties ke dwara beech mein pakde jaane (intercept) se bachaya ja sakta hai.

Install OpenSSH

```
Toxicx01@htb[/htb]$ sudo apt install openssh-server -y
```

Server run ho raha hai ya nahi, ye check karne ke liye hum neeche di gayi command ka use kar sakte hain:

Server Status

```
Toxicx01@htb[/htb]$ systemctl status ssh

● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/system/system/ssh.service; enabled; vendor preset:
enabled)
     Active: active (running) since Sun 2023-02-12 21:15:27 GMT; 1min 22s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 7740 (sshd)
      Tasks: 1 (limit: 9458)
     Memory: 2.5M
        CPU: 236ms
     CGroup: /system.slice/ssh.service
             └─7740 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups
```

Penetration testers ke roop mein, network audit karte waqt hum remote systems ko securely access karne ke liye OpenSSH ka use karte hain.

Iske liye hum neeche di gayi command ka use kar sakte hain:

SSH - Logging In

```
Toxicx01@htb[/htb]$ ssh cry0l1t3@10.129.17.122

The authenticity of host '10.129.17.122 (10.129.17.122)' can't be established.
ECDSA key fingerprint is SHA256:bKzhv+n2pYqr2r...Egf8LfqaHNxk.

Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

OpenSSH ko configure aur customize karne ke liye hum text editor se **`/etc/ssh/sshd_config`** file ko edit kar sakte hain.

Is file mein hum settings change kar sakte hain, jaise ek saath kitne connections allow honge, login ke liye password ya keys ka use hoga, host key checking aur doosri settings.

Lekin OpenSSH ki configuration file mein changes karte waqt humein saavdhani rakhni chahiye.

Example ke liye, hum SSH ka use karke kisi remote system mein securely login kar sakte hain aur wahan commands chala sakte hain.

Hum tunneling aur port forwarding ka use karke data ko encrypted connection ke through bhej sakte hain. Isse hum network aur system ki settings check kar sakte hain, bina is darr ke ki koi third party data ya commands ko beech mein pakad le.

NFS

Network File System (`NFS`) ek network protocol hai, jo humein remote systems par files store aur manage karne ki permission deta hai, bilkul waise hi jaise files hamare local system mein stored hoti hain.

Isse network ke through files ko aasani se manage kiya ja sakta hai.

Example ke liye, administrators NFS ka use files ko ek central location par store aur manage karne ke liye karte hain. Ye Linux aur Windows systems ke liye useful hai aur data ko easily share aur manage karne mein help karta hai.

Linux ke liye kai NFS servers available hain, jaise NFS-UTILS (`Ubuntu`), NFS-Ganesha (`Solaris`) aur OpenNFS (`Redhat Linux`).

NFS ka use resources ko efficiently share aur manage karne ke liye bhi kiya ja sakta hai. Jaise, ek server ke file system ko doosre server par replicate karna.

Iske alawa, NFS mein kuch features bhi hote hain:

- Access controls.
- Real-time file transfer.
- Ek saath multiple users ko data access karne ki permission.

Agar target system mein FTP client installed nahi hai ya FTP ki jagah NFS run ho raha hai, toh hum NFS ka use FTP ki tarah files access aur manage karne ke liye kar sakte hain.

Linux mein NFS install karne ke liye hum neeche di gayi command ka use kar sakte hain:-

Install NFS

```
Toxicx01@htb[/htb]$ sudo apt install nfs-kernel-server -y
```

Server run ho raha hai ya nahi, ye check karne ke liye hum neeche di gayi command ka use kar sakte hain:

Server Status

```shellsession
Toxicx01@htb[/htb]$ systemctl status nfs-kernel-server

● nfs-server.service - NFS server and services
     Loaded: loaded (/lib/system/system/nfs-server.service; enabled; vendor preset:
enabled)
     Active: active (exited) since Sun 2023-02-12 21:35:17 GMT; 13s ago
    Process: 9234 ExecStartPre=/usr/sbin/exportfs -r (code=exited,
status=0/SUCCESS)
    Process: 9235 ExecStart=/usr/sbin/rpc.nfsd $RPCNFSDARGS (code=exited,
status=0/SUCCESS)
   Main PID: 9235 (code=exited, status=0/SUCCESS)
        CPU: 10ms
```

Hum NFS ko configuration file **`/etc/exports`** ke through configure kar sakte hain.

Yeh file specify karti hai ki kaun-si directories share karni hain aur users aur systems ko kya access rights dene hain.

Isme transfer speed aur encryption ke use jaise settings ko configure karna bhi possible hai.

NFS ke access rights decide karte hain ki kaun-se users aur systems shared directories ko access kar sakte hain aur woh unke saath kya-kya actions perform kar sakte hain.

Yahan kuch important access rights diye gaye hain jinhe NFS mein configure kiya ja sakta hai:

Permissions	Description
`rw`	Users aur systems ko shared directory mein read (padhne) aur write (likhne ya change karne) ki permission deta hai.
`ro`	Users aur systems ko shared directory mein sirf read (padhne) ki permission deta hai.
`no_root_squash`	Client machine ke root user ko normal user ke rights tak restrict nahi karta. Matlab root user ke paas root wale permissions rehte hain.
`root_squash`	Client machine ke root user ke permissions ko restrict karke normal user ke permissions jaisa kar deta hai.
`sync`	Data transfer ko synchronize karta hai. Matlab changes tabhi transfer hote hain jab woh file system mein save ho chuke hote hain.
`async`	Data ko asynchronously transfer karta hai, jisse transfer fast hota hai. Lekin agar changes properly save nahi hue hain, toh file system mein inconsistency (data mismatch) ho sakti hai.

Example ke liye, hum ek naya folder create kar sakte hain aur use temporarily NFS mein share kar sakte hain.
Hum yeh kaam neeche diye gaye steps ke according karenge:

Create NFS Share

```shellsession
cry0l1t3@htb:~$ mkdir nfs_sharing
cry0l1t3@htb:~$ echo '/home/cry0l1t3/nfs_sharing hostname(rw,sync,no_root_squash)'
>> /etc/exports
cry0l1t3@htb:~$ cat /etc/exports | grep -v "#"

/home/cry0l1t3/nfs_sharing hostname(rw,sync,no_root_squash)
```

Agar humne NFS share create kar liya hai aur target system par uske saath kaam karna chahte hain, toh sabse pehle humein use mount karna hoga.

Hum yeh kaam neeche diye gaye command se kar sakte hain:

Mount NFS Share

```
cry0l1t3@htb:~$ mkdir ~/target_nfs
cry0l1t3@htb:~$ mount 10.129.12.17:/home/john/dev_scripts ~/target_nfs
cry0l1t3@htb:~$ tree ~/target_nfs

target_nfs/
├── css.css
├── html.html
├── javascript.js
├── php.php
└── xml.xml

0 directories, 5 files
```

Toh humne target (`10.129.12.17`) ke NFS share (`dev_scripts`) ko network ke through apne local system par `target_nfs` naam ke mount point par mount kar liya hai.

Ab hum iske contents ko bilkul waise hi dekh sakte hain, jaise hum target system par hi maujood hon.

Kuch specific situations mein NFS ka use karke remote system par apne privileges (permissions) ko bhi escalate (badhaya) ja sakta hai.

Web Server

Web servers kaise kaam karte hain, yeh samajhna penetration testers ke liye bahut zaroori hai, kyunki web servers web applications ka important part hote hain aur security testing ke dauraan aksar primary targets hote hain.

Web server ek software hota hai jo Internet ke through data, documents, applications aur alag-alag functions deliver karta hai.

Yeh data ko clients (jaise web browsers) tak bhejne aur unse requests receive karne ke liye Hypertext Transfer Protocol (`HTTP`) ka use karta hai.

Jo data receive hota hai, use client ke browser mein Hypertext Markup Language (`HTML`) ke form mein display kiya jata hai.

Isse dynamic web pages create karne mein help milti hai, jo users ki requests ke according interactively respond karte hain.

Isliye, secure aur efficient web applications develop karne aur overall system security maintain karne ke liye web server ke functions ko achhe se samajhna zaroori hai.

Linux platforms par sabse zyada use hone wale web servers hain:

- Apache
- Nginx
- Lighttpd
- Caddy

Inmein Apache kaafi popular hai, kyunki yeh Ubuntu, Solaris aur Red Hat Linux jaise alag-alag operating systems ke saath compatible hai.

Penetration testers ke liye web servers ke kai uses hote hain.

- File Transfer: Web servers ka use files transfer karne ke liye kiya ja sakta hai. Iski help se testers HTTP ya HTTPS ports ke through target systems mein login karke unke saath interact kar sakte hain.
- Phishing Attacks: Web servers par target website ke duplicate pages host kiye ja sakte hain, taaki users ki login credentials (username aur password) capture karne ki koshish ki ja sake.

Inke alawa bhi web servers network mein vulnerabilities ko test aur exploit karne ke kai opportunities provide karte hain.

Apache Web Server mein kai features hote hain, jo humein secure aur efficient web applications host karne ki facility dete hain.

Iske alawa, hum logging ko bhi configure kar sakte hain, jisse humein apne server par hone wale traffic ki information milti hai. Yeh information attacks ko analyze karne mein help karti hai.

Hum neeche diye gaye command ka use karke Apache install kar sakte hain:

Install Apache Web Server

```
Toxicx01@htb[/htb]$ sudo apt install apache2 -y
```

Apache2 mein yeh specify karne ke liye ki kaun-se folders ko access kiya ja sakta hai, hum kisi text editor ki help se **`/etc/apache2/apache2.conf`** file ko edit kar sakte hain.

Yeh file Apache ki global settings contain karti hai.

Hum is file ki settings change karke specify kar sakte hain:

- Kaun-si directories ko access kiya ja sakta hai.
- Un directories par kaun-se actions perform kiye ja sakte hain.

Apache Configuration

```
<Directory /var/www/html>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
</directory>
```

Yeh section specify karta hai ki default **`/var/www/html`** folder accessible hai, users **`Indexes`** aur **`FollowSymLinks`** options ka use kar sakte hain, **`AllowOverride All`** ke through is directory ki files ki settings ko override kiya ja sakta hai, aur **`Require all granted`** sabhi users ko is directory ka access deta hai.

Example ke liye, agar hum web server ka use karke apne target system mein files transfer karna chahte hain, toh hum required files ko **`/var/www/html`** folder mein rakh sakte hain. Uske baad target system par **`wget`**, **`curl`** ya doosre applications ka use karke in files ko download kar sakte hain.

`.htaccess` file ka use karke individual directories ki settings ko customize karna bhi possible hai. Hum yeh file us directory ke andar create kar sakte hain jiske settings hum change karna chahte hain.

Yeh file humein Apache ki configuration file ko edit kiye bina kuch directory-level settings configure karne ki permission deti hai, jaise access controls.

Hum additional modules bhi add kar sakte hain, jo humein extra features provide karte hain, jaise **`mod_rewrite`**, **`mod_security`** aur **`mod_ssl`**. Yeh modules hamari web application ki security improve karne mein help karte hain.

Python Web Server Apache ka ek simple aur fast alternative hai. Iska use ek single command ke through kisi ek folder ko host karne ke liye kiya ja sakta hai, taaki doosre system mein files transfer ki ja sakein.

Python Web Server install karne ke liye humein apne system mein Python3 install karna hoga aur uske baad neeche diya gaya command run karna hoga:

Install Python & Web Server

```
Toxicx01@htb[/htb]$ sudo apt install python3 -y
Toxicx01@htb[/htb]$ python3 -m http.server
```

Jab hum yeh command run karte hain, toh hamara Python Web Server `TCP/8000` port par start ho jayega aur hum jis folder mein currently hain, us folder ko access kar sakte hain.

Hum neeche diye gaye command ka use karke kisi doosre folder ko bhi host kar sakte hain:

```
shellsession
Toxicx01@htb[/htb]$ python3 -m http.server --directory /home/cry0l1t3/target_files
```

Yeh command `TCP/8000` port par Python Web Server start karega. Example ke liye, hum browser ke through `/home/cry0l1t3/target_files` folder ko access kar sakte hain.

Jab hum apne Python Web Server ko access karte hain, toh browser mein uska link type karke aur files download karke doosre system mein transfer kar sakte hain.

Hum Python Web Server ko default port ke bajaye kisi doosre port par bhi host kar sakte hain:

```
Toxicx01@htb[/htb]$ python3 -m http.server 443
```

Yeh command hamare Python Web Server ko default `TCP/8000` port ke bajaye port 443 par host karega.

Hum apne browser mein link type karke is web server ko access kar sakte hain.

VPN

Virtual Private Network (VPN) ek secure, invisible tunnel ki tarah kaam karta hai, jo humein kisi doosre network se connect karta hai. Isse hum us network ko aise access kar sakte hain, jaise hum physically usi network ke andar maujood hon.

Yeh client aur server ke beech ek encrypted tunnel establish karke possible hota hai. Isse connection ke through transfer hone wala saara data confidential rehta hai aur unauthorized access se protected rehta hai.

Organizations mainly VPN ka use apne employees ko internal network ka secure access dene ke liye karti hain, bina unhe office mein physically present hone ki zaroorat ke.

Is flexibility ki wajah se employees kisi bhi location se internal resources aur applications access kar sakte hain, jisse productivity aur mobility improve hoti hai.

Iske alawa, VPN internet traffic ko anonymous banane aur external intrusions ko block karne ke liye bhi use hota hai, jisse security aur strong hoti hai.

Linux servers ke liye sabse zyada use hone wale VPN solutions hain:

- OpenVPN
- L2TP/IPsec
- PPTP
- SSTP
- SoftEther

Inmein OpenVPN ek popular open-source option hai, jo Ubuntu, Solaris aur Red Hat Linux jaise alag-alag operating systems ke saath compatible hai.

Administrators OpenVPN ka use corporate networks ko secure remote access dene, network traffic encrypt karne aur users ki online anonymity maintain karne ke liye karte hain.

Penetration testers ke liye OpenVPN bahut useful capabilities provide karta hai.

Yeh testers ko internal networks se securely connect hone ki permission deta hai, khaaskar tab jab geographical restrictions ki wajah se direct access possible nahi hota.

OpenVPN ka use karke penetration testers internal systems ki complete security assessment kar sakte hain aur possible vulnerabilities ko identify karke unhe address kar sakte hain.

OpenVPN ki versatility aur iske features, jaise encryption, tunneling, traffic shaping, network routing aur dynamic network environments ke according adjust hone ki ability, ise network administrators aur security professionals dono ke liye ek important tool banati hai.

Hum neeche diye gaye command ka use karke OpenVPN server aur client install kar sakte hain:

Install OpenVPN

```
Toxicx01@htb[/htb]$ sudo apt install openvpn -y
```

OpenVPN ko customize aur configure karne ke liye hum configuration file **/etc/openvpn/server.conf** ko edit kar sakte hain.

Is file mein OpenVPN server ki settings hoti hain.

Hum in settings ko change karke kuch features configure kar sakte hain, jaise encryption, tunneling, traffic shaping etc.

Agar hum kisi OpenVPN server se connect hona chahte hain, toh hum server se mili **.ovpn** file ka use kar sakte hain aur use apne system mein save kar sakte hain.

Hum command line par neeche diya gaya command use karke aisa kar sakte hain:

Connect to VPN

```
Toxicx01@htb[/htb]$ sudo openvpn --config internal.ovpn
```

Connection establish hone ke baad, hum internal network par maujood hosts ke saath communicate kar sakte hain.

Working with Web Services

Web development ka ek aur important element browser aur web server ke beech communication hai.

Linux operating system par web server setup karne ke kai tareeke hain. Iske popular options mein Nginx, IIS aur Apache shamil hain. Inmein se Apache sabse zyada use hone wale web servers mein se ek hai.

Apache ko website ke engine ki tarah samjho, jo tumhari website aur visitors ke beech communication ko smoothly chalata hai.

Hum Apache ko ghar ki foundation aur framework ki tarah bhi samajh sakte hain. Jaise tum ghar mein alag-alag rooms add kar sakte ho ya apni zaroorat ke hisaab se features customize kar sakte ho, waise hi Apache mein bhi modules add kiye ja sakte hain.

Har module ek specific purpose ke liye banaya gaya hai, chahe woh communication ko secure karna ho, traffic ko doosri jagah bhejna ho, ya content ko dynamically change karna ho.

Apache ki asli strength iski modularity hai. Iska matlab hai ki Apache ko alag-alag modules ki help se customize aur extend kiya ja sakta hai, taaki woh specific tasks perform kar sake.

Examples:

- **mod_ssl**: Ye ek lockbox ki tarah kaam karta hai. Browser aur web server ke beech communication ko data encrypt karke secure banata hai.

- **mod_proxy**: Ye ek traffic controller ki tarah kaam karta hai. Requests ko correct destination tak pahunchata hai. Proxy server setup karne mein ye useful hota hai.

- **mod_headers**: Browser aur server ke beech travel karne wale HTTP headers ko modify karne ki permission deta hai.

- **mod_rewrite**: HTTP URLs ko turant modify ya rewrite karne ki facility deta hai.

Static web content handle karne ke alawa, Apache server-side scripting languages ki help se dynamic web pages banane ko bhi support karta hai.

Commonly used scripting languages mein PHP, Perl aur Ruby shamil hain. Lekin tum Python, JavaScript, Lua ya .NET jaisi doosri technologies ka bhi use kar sakte ho.

Ye scripting languages background mein creative tools ki tarah kaam karti hain. Ye dynamically content generate karti hain aur website ko interactive aur responsive banati hain.

Agar tumne abhi tak Apache install nahi kiya hai, toh chalo Apache install karte hain.

```shellsession
Toxicx01@htb[/htb]$ sudo apt install apache2 -y

Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2
0 upgraded, 1 newly installed, 0 to remove and 17 not upgraded.
Need to get 95,1 kB of archives.
After this operation, 535 kB of additional disk space will be used.
Get:1 http://de.archive.ubuntu.com/ubuntu bionic-updates/main amd64 apache2 amd64
2.4.29-1ubuntu4.13 [95,1 kB]
Fetched 95,1 kB in 0s (270 kB/s)
<SNIP>
```

Ab hum **apache2ctl**, **systemctl** ya **service** commands ka use karke server start kar sakte hain.

apache2 naam ka ek binary bhi available hota hai, lekin generally server ko directly start karne ke liye iska use nahi kiya jata.

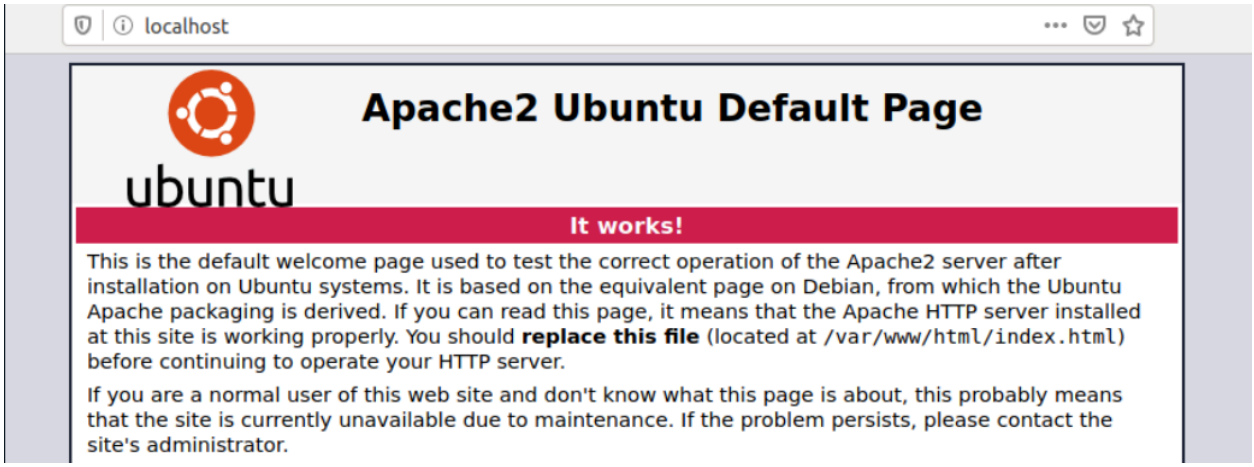
Aisa isliye hai kyunki default configuration mein environment variables ka use hota hai.

```
Toxicx01@htb[/htb]$ sudo systemctl start apache2
```

Apache start hone ke baad, hum apne browser mein default page open karne ke liye **http://localhost** par jaate hain.

Default settings ke according, Apache HTTP port 80 par run karta hai.

Jab bhi tum browser mein koi HTTP URI enter karte ho, toh browser automatically port 80 use karta hai, jab tak tum khud koi doosra port specify nahi karte.



Ye installation ke baad aane wala default page hai. Iska kaam confirm karna hai ki web server sahi tarah se kaam kar raha hai.

Agar tum Pwnbox use kar rahe ho, toh Apache start karte waqt error aa sakta hai. Aisa isliye hota hai kyunki port 80 pehle se hi kisi doosri service ke use mein hota hai.

Apne web server ke liye alternate port set karne ke liye hum `/etc/apache2/ports.conf` file ko edit kar sakte hain.

Yahan humne port ko 8080 set kiya hai.

```shellsession
  GNU nano 2.9.3
/etc/apache2/ports.conf

# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 8080

<IfModule ssl_module>
Listen 443
</IfModule>

<IfModule mod_gnutls.c>
Listen 443
</IfModule>

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur
Pos     M-U Undo
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell    ^_ Go To
Line  M-E Redo
```

Ab hum Apache ko restart kar sakte hain aur browser mein **`http://localhost:8080`** open kar sakte hain.

Ya phir verify karne ke liye **`curl`** jaise command-line tool ka bhi use kar sakte hain.

```
Toxicx01@htb[/htb]$ curl -I http://localhost:8080

HTTP/1.1 200 OK
Date: Mon, 04 Nov 2024 21:18:50 GMT
Server: Apache/2.4.62 (Debian)
Last-Modified: Mon, 07 Oct 2024 06:39:39 GMT
ETag: "29cd-623dd48f6dd5a"
Accept-Ranges: bytes
Content-Length: 10701
Vary: Accept-Encoding
Content-Type: text/html
```

Web servers ke saath kaam karne ka ek aur important aspect hai `curl` aur `wget` jaise command-line tools ka use karke unse communicate karna seekhna.

Jab hum web server par hosted kisi webpage ke content ko systematically analyze karna chahte hain, tab ye tools bahut useful hote hain.

In tools ko terminal ke andar tumhare personal web browsers ki tarah samjho. Ye tumhe command line se directly web content fetch (hasil) karne aur uske saath interact karne ki permission dete hain.

Example:

Hum ek simple Bash script likh sakte hain, jo kisi webpage ko download kare aur usmein maujood sabhi URLs ko extract (nikal) kare.

Ye bilkul aisa hai jaise information ke samundar mein jaal daalkar sirf zaroori links ko bahar nikalna.

Aisi scripts in tasks ke liye bahut useful hoti hain:

- Web scraping
- Automated testing
- Website mein hone wale changes ko monitor karna.

Lekin abhi hamara main goal Linux ko achhe se samajhna hai. Isliye tumhe doosre modules mein aisi scripts ko dekhne, banane aur use karne ka mauka milega.

Filhaal, hum is baat par focus karenge ki `curl` aur `wget` ka use karke web server ke saath kaise interact aur communicate kar sakte hain.

CURL

`cURL` ek aisa tool hai jo humein shell (terminal) se alag-alag protocols jaise `HTTP`, `HTTPS`, `FTP`, `SFTP`, `FTPS` aur `SCP` ke through files transfer karne ki permission deta hai.

Generally, ye humein command line ke through remotely websites ko control aur test karne ki facility bhi deta hai.

Remote servers ke content ke alawa, hum individual requests ko bhi dekh sakte hain, taaki samajh sakein ki client aur server ke beech communication kaise ho raha hai.

Usually, `cURL` pehle se hi Linux ke most systems mein installed hota hai.

Ye ek aur important reason hai ki humein is tool ko use karna seekhna chahiye, kyunki future mein ye kai processes ko bahut aasaan bana sakta hai.

```
Toxicx01@htb[/htb]$ curl http://localhost

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <!--
    Modified from the Debian original for Ubuntu
    Last updated: 2016-11-16
    See: https://launchpad.net/bugs/1288690
  -->
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Ubuntu Default Page: It works</title>
    <style type="text/css" media="screen">
...SNIP...
```

`title` tag mein hum dekh sakte hain ki wahi text hai jo humein browser mein dikhai deta hai.

Isse humein website ka source code inspect karne aur usse information nikalne ki permission milti hai.

Specifically, `curl` website ka page source STDOUT (standard output) ke roop mein return karta hai.

Browser mein website dekhne ke comparison mein ye alag hai. Browser HTML, CSS aur JavaScript ko render karta hai, jisse website visually achhi aur attractive dikhti hai.

Iske baare mein hum kisi doosre module mein detail mein padhenge.

Wget

`curl` ka ek alternative tool `wget` hai.

Is tool ki help se hum terminal se directly FTP ya HTTP servers se files download kar sakte hain. Ye ek achhe download manager ki tarah kaam karta hai.

Agar hum `wget` ka use isi tarah karte hain, toh `curl` se iska difference ye hai ki website ka content download hokar hamare system mein locally store ho jata hai, jaisa ki neeche diye gaye example mein dikhaya gaya hai.

```
Toxicx01@htb[/htb]$ wget http://localhost

--2020-05-15 17:43:52--  http://localhost/
Resolving localhost (localhost)... 127.0.0.1
Connecting to localhost (localhost)|127.0.0.1|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10918 (11K) [text/html]
Saving to: 'index.html'

index.html                100%[=======================================>]  10,66K
--.-KB/s    in 0s

2020-05-15 17:43:52 (33,0 MB/s) - ‘index.html’ saved [10918/10918]
```

Python 3

Data transfer ke liye ek aur option Python 3 ka use karna hai.

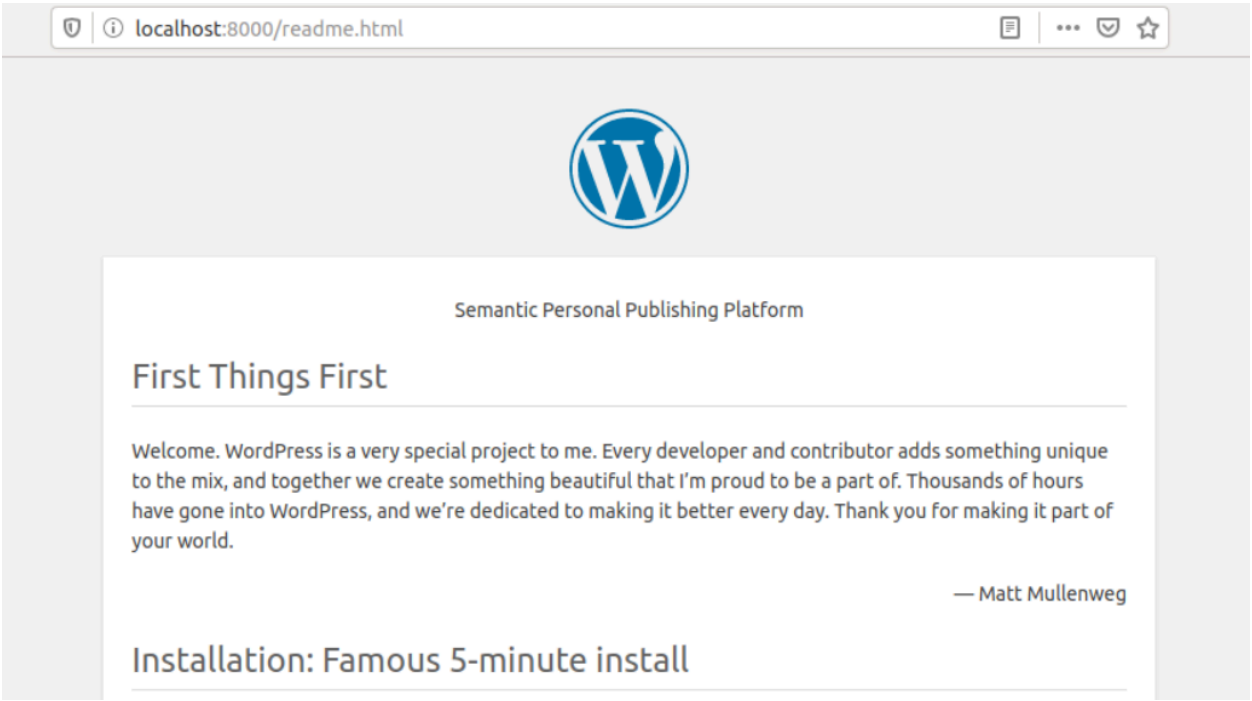
Is case mein, web server ki root directory mein jaakar command execute ki jaati hai, jisse server start hota hai.

Is example mein, hum ek aisi directory mein hain jahan WordPress install hai aur usmein `readme.html` file bhi maujood hai.

Ab hum Python 3 web server start karte hain aur dekhte hain ki kya hum browser ka use karke us file ko access kar sakte hain.

```
Toxicx01@htb[/htb]$ python3 -m http.server

Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```



Ab hum apne Python 3 web server ke events ko check karke dekh sakte hain ki kaun-kaun si requests bheji gayi thi.

```
Toxicx01@htb[/htb]$ python3 -m http.server

Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
127.0.0.1 - - [15/May/2020 17:56:29] "GET /readme.html HTTP/1.1" 200 -
127.0.0.1 - - [15/May/2020 17:56:29] "GET /wp-admin/css/install.css?ver=20100228
HTTP/1.1" 200 -
127.0.0.1 - - [15/May/2020 17:56:29] "GET /wp-admin/images/wordpress-logo.png
HTTP/1.1" 200 -
127.0.0.1 - - [15/May/2020 17:56:29] "GET /wp-admin/images/wordpress-logo.svg?
ver=20131107 HTTP/1.1" 200 -
```

Penetration testing mein, humein aksar aisi challenges ka saamna karna padta hai, jinhe solve karne ke liye creative problem solving aur out-of-the-box thinking ki zaroorat hoti hai.

Tumhe aisi situations milengi jinse tum pehle kabhi deal nahi kiye hoge. Iska matlab hai ki tumhe sirf kuch naya seekhna hi nahi hoga, balki research aur innovative thinking ki help se khud solutions bhi find karne honge.

Yaad rakho, ye ek learning process hai, exam nahi.

Apni skills ko improve karne ke liye khud research karna aur alag-alag approaches ko investigate karna bahut zaroori hai.

Asal mein, ye efforts tumhari expertise aur adaptability (alag-alag situations ke hisaab se khud ko adjust karne ki ability) build karne mein important role play karenge.

Ab se tumhe jo exercises milengi, woh jaanbujhkar tumhe tumhare comfort zone se bahar nikalne ke liye design ki gayi hain.

Aisa tumhari learning speed badhane aur tumhe jaldi improve karne mein help karne ke liye kiya gaya hai.

Jab tum in challenges ka saamna karoge, toh tum aisi skills develop karoge jo real-world situations ko handle karne ke liye zaroori hain.

Real-world situations mein aksar har problem ka ek fixed solution nahi hota.

Is learning process ko enjoy karo, naye tareeke explore karo aur khud solutions find karne ki koshish karo, kyunki isi tarah tumhari skills aur knowledge improve hongi.

Q. Find a way to start a simple HTTP server inside Pwnbox or your local VM using "npm". Submit the command that starts the web server on port 8080 (use the short argument to specify the port number).

Ans.

Q. Find a way to start a simple HTTP server inside Pwnbox or your local VM using "php". Submit the command that starts the web server on the localhost (127.0.0.1) on port 8080.

Ans.

Backup and Restore

Linux systems data ka backup lene aur use restore karne ke liye kai powerful tools provide karte hain. Ye tools efficient aur secure dono hote hain.

Ye tools ensure karte hain ki hamara data loss (data kho jana) ya corruption (data kharab hona) se safe rahe. Aur jab humein zaroorat ho, toh hum apne data ko aasani se wapas access kar sakein.

Ubuntu system mein data ka backup lene ke liye hamare paas kai options hain:

- **Rsync**
- **Deja Dup**
- **Duplicity**

Rsync

Rsync ek open-source tool hai jo fast aur secure backups lene ki permission deta hai, chahe backup locally liya jaaye ya kisi remote location par.

Iska ek important advantage ye hai ki ye sirf files ke un parts ko transfer karta hai jinmein changes hue hain.

Isse large amount of data ka backup lena bahut efficient ho jata hai.

Rsync khaas taur par network transfers ke liye useful hai, jaise:

- Ek server se doosre server mein files sync karna.
- Internet ke through incremental backups banana.

Duplicity

Duplicity ek aur powerful tool hai jo Rsync par based hai, lekin ismein data ko protect karne ke liye encryption features bhi diye gaye hain.

Ye tumhe backup copies encrypt karne ki permission deta hai.

Isse tumhara sensitive data secure rehta hai, chahe woh remote servers, FTP sites ya Amazon S3 jaisi cloud services par store ho.

Duplicity, Rsync ki efficient data transfer capabilities ko maintain karte hue security ki ek extra layer provide karta hai.

Example se samjho:

Apne data ko ek ghar mein rakhe hue valuable treasures ki tarah samjho.

Linux ke backup tools, jaise Rsync, Duplicity aur Deja Dup, alag-alag types ki tijoriyon ki tarah kaam karte hain.

- Rsync: Ye ek fast transport ki tarah hai, jo sirf naya ya changed data transfer karta hai. Isliye remote location par updates bhejne ke liye useful hai.

- Duplicity: Ye ek high-security tijori ki tarah hai, jo data ko store karne ke saath complex code (encryption) se lock bhi kar deti hai. Isse koi doosra insaan tumhara data access nahi kar sakta.

- Deja Dup: Ye ek simple aur easy-to-use tijori ki tarah hai, jise koi bhi aasani se operate kar sakta hai aur phir bhi protection provide karti hai.

Apne backups ko encrypt karna tijori par ek extra lock lagane jaisa hai. Isse agar kisi ko tumhari backup file mil bhi jaaye, toh bhi woh uske andar ka data access nahi kar sakta.

Deja Dup

Jo users simple aur user-friendly option prefer karte hain, unke liye Deja Dup ek graphical interface provide karta hai, jisse backup lena bahut aasaan ho jata hai.

Background mein ye bhi Rsync ka use karta hai aur Duplicity ki tarah encrypted backups ko support karta hai.

Deja Dup un users ke liye ideal hai jo command line mein jaaye bina aasani se backup aur restore options use karna chahte hain.

Backup ki Security

Apne backups ki security ensure karna utna hi important hai jitna backup create karna.

Backup data ko encrypt karne se woh unauthorized access (bina permission ke access) se safe rehta hai.

Ubuntu systems mein backups ko extra protection dene ke liye tum **GnuPG**, **eCryptfs** ya **LUKS** jaise additional encryption tools ka use kar sakte ho.

Ubuntu par Rsync install karne ke liye hum `apt` package manager ka use kar sakte hain.

Install Rsync

```
Toxicx01@htb[/htb]$ sudo apt install rsync -y
```

Isse hamare system mein Rsync ka latest version install ho jayega.

Installation complete hone ke baad, hum data ka backup lene aur use restore karne ke liye is tool ka use karna shuru kar sakte hain.

`rsync` ka use karke kisi poori directory ka backup lene ke liye hum neeche di gayi command ka use kar sakte hain:

Rsync - Backup a local Directory to our Backup-Server

```
Toxicx01@htb[/htb]$ rsync -av /path/to/mydirectory
user@backup_server:/path/to/backup/directory
```

Ye command poori directory (**`/path/to/mydirectory`**) ko remote host (**`backup_server`**) par copy karegi. Wahan ye data **`/path/to/backup/directory`** directory mein store hoga.

`archive` (**`-a`**) option original files ke attributes ko same rakhta hai, jaise permissions, timestamps aadi.

`verbose` (**`-v`**) option **`rsync`** operation ke progress ka detailed output dikhata hai.

Hum backup process ko customize karne ke liye additional options bhi add kar sakte hain, jaise compression aur incremental backups.

Hum ye kaam neeche di gayi command ki help se kar sakte hain:

```
Toxicx01@htb[/htb]$ rsync -avz --backup --backup-dir=/path/to/backup/folder --
delete /path/to/mydirectory user@backup_server:/path/to/backup/directory
```

Is command ki help se hum **`mydirectory`** ka backup remote **`backup_server`** par lete hain. Ismein original file attributes, timestamps aur permissions same rehte hain.

Saath hi, faster transfer ke liye compression (**`-z`**) enable kiya gaya hai.

- **`--backup`** option **`/path/to/backup/folder`** directory mein incremental backups create karta hai.

- `--delete` option remote host se un files ko delete kar deta hai, jo source directory mein ab present nahi hain.

Agar hum backup server se apni directory ko wapas local directory mein restore karna chahte hain, toh hum neeche di gayi command ka use kar sakte hain:

Rsync - Restore our Backup

```
Toxicx01@htb[/htb]$ rsync -av user@remote_host:/path/to/backup/directory
/path/to/mydirectory
```

Encrypted Rsync

Apne local host aur backup server ke beech `rsync` se file transfer ko secure banane ke liye, hum SSH aur doosre security measures ka use kar sakte hain.

SSH ka use karne se transfer hote waqt hamara data encrypt ho jata hai. Isse kisi unauthorized person (bina permission wale insaan) ke liye data access karna bahut mushkil ho jata hai.

Iske alawa, hum firewalls aur doosre security protocols ka bhi use kar sakte hain, taaki transfer ke dauraan hamara data safe aur secure rahe.

In steps ko follow karke hum apne data ko protect kar sakte hain aur file transfer ko secure bana sakte hain.

Isliye, hum `rsync` ko SSH use karne ke liye kehte hain, jaise neeche diya gaya hai:

Secure Transfer of our Backup

```
Toxicx01@htb[/htb]$ rsync -avz -e ssh /path/to/mydirectory
user@backup_server:/path/to/backup/directory
```

Hamare local host aur backup server ke beech data transfer encrypted SSH connection ke through hota hai. Ye connection transfer hone wale data ki confidentiality (gopniyata) aur integrity (data mein koi badlav na hona) ko protect karta hai.

Ye encryption process ensure karta hai ki data kisi bhi potential malicious person se protected rahe, jo bina permission ke data ko access ya modify kar sakta hai.

Encryption key ko bhi security protocols ke ek comprehensive set ke through protect kiya jata hai, jisse kisi unauthorized person ke liye data tak pahunchna aur bhi mushkil ho jata hai.

Iske alawa, encrypted connection ko is tarah design kiya gaya hai ki woh security todne ki koshishon ka strongly saamna kar sake.

Isse hum transfer hone wale data ki security ko lekar confident reh sakte hain.

Auto-Synchronization

rsync ka use karke auto-synchronization enable karne ke liye, hum **cron** aur **rsync** ko ek saath use kar sakte hain, taaki synchronization process automatically ho sake.

Cron job ko regular intervals (fix time gap) par run karne ke liye schedule karne se dono systems ka data ek-doosre ke saath synchronized rehta hai.

Ye un organizations ke liye khaas taur par beneficial hai, jinhe apna data multiple machines ke beech synchronized rakhna hota hai.

Iske alawa, **rsync** ke saath auto-synchronization setup karna time aur effort bachane ka ek achha tareeka hai, kyunki isse manually synchronization karne ki zaroorat nahi padti.

Ye ensure karne mein bhi help karta hai ki systems mein stored files aur data up-to-date aur consistent rahein. Isse errors kam hote hain aur efficiency improve hoti hai.

Isliye, hum **RSYNC_Backup.sh** naam ki ek nayi script banate hain, jo rsync command ko run karke hamari local directory ko remote directory ke saath sync karegi.

Lekin, kyunki hum **rsync** connection ke liye SSH perform karne wali script ka use kar rahe hain, isliye humein key-based authentication configure karna hoga.

Aisa isliye zaroori hai taaki SSH se connect karte waqt humein baar-baar password enter na karna pade.

Sabse pehle, hum apne user ke liye ek key pair generate karte hain.

```
Toxicx01@htb[/htb]$ ssh-keygen -t rsa -b 2048
```

Prompts ko follow karke key ki location specify karein (default location `~/.ssh/id_rsa` hai) aur chahein toh passphrase bhi set kar sakte hain. Agar passphrase nahi rakhna hai, toh ise khaali chhod dein.

Iske baad, humein apni public key remote server par copy karni hogi.

```
Toxicx01@htb[/htb]$ ssh-copy-id user@backup_server
```

Ab hum **rsync** backup process ko automatically karne ke liye apni script create kar sakte hain.

RSYNC_Backup.sh

```bash
#!/bin/bash

rsync -avz -e ssh /path/to/mydirectory user@backup_server:/path/to/backup/directory
```

Uske baad, script ko properly execute karne ke liye humein use zaroori permissions deni hongi.

Iske alawa, ye ensure karna bhi important hai ki script ka ownership (malikana haq) sahi user ke paas ho.

Isse ye ensure hoga ki sirf sahi user hi script ko access kar sake aur koi doosra user script ke saath chhedchhad (tampering) na kar sake.

```
Toxicx01@htb[/htb]$ chmod +x RSYNC_Backup.sh
```

Iske baad, hum ek **crontab** create kar sakte hain, jo **cron** ko instruction dega ki woh har ghante ke 0th minute par script ko run kare.

```
Toxicx01@htb[/htb]$ crontab -e
```

Hum apni zaroorat ke hisaab se timing change kar sakte hain.

Aisa karne ke liye, **crontab** mein neeche diya gaya content hona chahiye:

Auto-Sync - Crontab

```
0 * * * * /path/to/RSYNC_Backup.sh
```

Is setup ke saath, `cron` desired interval par script execute karne ki responsibility lega. Isse `rsync` command run hogi aur local directory ka data remote host ke saath synchronized rahega.

Hum tumhe encourage karte hain ki Pwnbox ka use karke `rsync` try karo.

Remote server ke saath files sync karne ke bajaye, tum Pwnbox ko hi source aur destination dono ke roop mein use kar sakte ho. Isse testing karna aasaan ho jayega.

Aisa karne ke liye, Pwnbox par do directories create karo:

1. `to_backup` – Jahan tumhara original data store hoga.
2. `synced_backup` – Jahan synchronized data copy kiya jayega.

Iske baad, `rsync` ka use karke `to_backup` directory ka data `synced_backup` directory mein transfer karo.

Is process ko automatically karne ke liye, ek `cron` job setup karo jo har minute run ho, taaki continuous synchronization hoti rahe.

Yaad rakho, kyunki hum locally testing kar rahe hain, isliye "remote" host ke address ke roop mein loopback IP address `127.0.0.1` ka use kar sakte hain.

File System Management

Linux mein File System ko Manage karna ek bahut important kaam hai. Ismein Disk ya kisi doosre Storage Device par Data ko Organize karna, Store karna aur Maintain karna shamil hota hai.

Linux ek versatile Operating System hai, jo bahut saare alag-alag File Systems ko support karta hai, jinmein ext2, ext3, ext4, XFS, Btrfs aur NTFS shamil hain.

In sabhi File Systems ke apne alag Features hote hain aur ye alag-alag kaamon ke liye suitable hote hain.

Sabse achha File System kaunsa hoga, ye Application ya User ki specific requirements par depend karta hai, jaise:

- - `ext2`: Ye ek purana File System hai, jismein Journaling ki facility nahi hoti. Is wajah se ye Modern Systems ke liye zyada suitable nahi hai, lekin kuch aise situations mein useful hai jahan kam overhead ki zarurat hoti hai, jaise USB Drives.

- - **`ext3`** aur **`ext4`**: Ye zyada Advanced File Systems hain, jinmein Journaling hoti hai. Journaling System Crash hone par Data Recover karne mein help karti hai. ext4 jyadatar Modern Linux Systems mein Default choice hota hai, kyunki ye Performance, Reliability aur Large File Support ke beech achha balance provide karta hai.
- - **`Btrfs`**: Ye apne Advanced Features ke liye jaana jaata hai, jaise Snapshot lena aur Built-in Data Integrity Checks. Ye Complex Storage Setups ke liye ideal hai.
- - **`XFS`**: Ye Large Files ko handle karne mein bahut achha hai aur iski Performance bhi high hoti hai. Ye un Environments ke liye sabse suitable hai jahan High I/O ki demand hoti hai.
- - **`NTFS`**: Ise originally Windows ke liye develop kiya gaya tha. Ye Compatibility ke liye useful hai, khaaskar Dual-Boot Systems ya External Drives ke case mein, jinhein Linux aur Windows dono par kaam karna hota hai.

File System select karte waqt Application ya User ki needs ko analyze karna zaroori hota hai. Performance, Data Integrity, Compatibility aur Storage Requirements jaise factors hamare decision ko influence karte hain.

Linux ka File System Architecture, Unix Model par based hai, jo ek Hierarchical Structure mein organized hota hai.

Is Structure mein kai components hote hain, jinmein sabse important `inodes` hote hain.

`Inodes` ek tarah ke Data Structures hote hain, jo har File aur Directory ka Metadata store karte hain. Ismein Permissions, Ownership, Size aur Timestamps jaisi information shamil hoti hai.

Inodes File ka actual Data ya uska Name store nahi karte, balki inmein Pointers hote hain, jo un Blocks ka address batate hain jahan Disk par File ka Data store hota hai.

`Inode Table` in sabhi Inodes ka ek Collection hota hai. Ye basically ek Database ki tarah kaam karta hai, jiska use Linux Kernel System ki har File aur Directory ko Track karne ke liye karta hai.

Is Structure ki wajah se Operating System Files ko efficiently Access aur Manage kar pata hai.

Linux mein Inodes ko samajhna aur Manage karna File System Management ka ek important part hai. Khaaskar aisi situation mein jab Disk ki actual Storage Capacity khatam hone se pehle hi Inode Space khatam ho jaati hai.

Ek Example se Samjhte Hain

Linux File System ko ek Library ki tarah samjho.

`Inodes` Library ke Catalog System (**`inode table`**) mein rakhe Index Cards ki tarah hote hain.

Har Card mein ek Book (File) ke baare mein detailed information hoti hai, jaise uska Title, Author, Location aur doosri Details.

Lekin Card mein actual Book nahi hoti.

Isi tarah Inode mein File ki detailed information hoti hai, lekin File ka actual Data nahi hota.

`Inode Table` poora Catalog hota hai, jo Library (Operating System) ko Books (Files) ko jaldi find aur Manage karne mein help karta hai.

Linux mein Files ke Main Types

Linux mein Files ko kuch important Types mein Store kiya ja sakta hai:

- Regular Files
- Directories
- Symbolic Links

Regular Files

Regular Files sabse common type ki files hoti hain. Inmein usually Text Data (jaise ASCII) aur/ya Binary Data (jaise images, audio ya executable files) hota hai.

Ye files poore File System ki alag-alag Directories mein hoti hain, sirf Root Directory mein nahi.

Root Directory (`/`) bas poore Hierarchical Directory Tree ka sabse upar wala part hoti hai. Is structure ke andar kisi bhi Directory mein files ho sakti hain.

Directories

Directories ek special type ki files hoti hain, jo doosri files (Regular Files aur doosri Directories) ko apne andar store karne ke liye containers ki tarah kaam karti hain.

Jab koi file kisi Directory ke andar store hoti hai, toh us Directory ko us file ki Parent Directory kaha jata hai.

Directories, Linux File System mein files ko organize karne mein help karti hain. Isse files ke collection ko efficiently manage karna aasan ho jata hai.

Symbolic Links

Regular Files aur Directories ke alawa, Linux Symbolic Links (`symlinks`) ko bhi support karta hai. Ye doosri files ya directories ke liye Shortcut ya Reference ki tarah kaam karte hain.

Symbolic Links ki help se hum File System ke alag-alag locations par rakhi files ko jaldi access kar sakte hain, bina original file ki duplicate copy banaye.

Symlinks ka use files ko access karna aasan banane ya complex Directory Structure ko organize karne ke liye kiya ja sakta hai. Ye alag-alag locations par important files ka path point karte hain.

Har user category ke liye alag-alag Permission Levels ho sakte hain.

Example ke liye, kisi file ke Owner ko us file ko Read, Write aur Execute karne ki permission ho sakti hai, jabki doosre users ko sirf Read karne ki permission ho sakti hai.

Ye permissions har category ke liye independent hoti hain. Iska matlab hai ki ek user ki permissions change karne se zaroori nahi ki doosre users ki permissions bhi change ho.

```
Toxicx01@htb[/htb]$ ls -il

total 0
10678872 -rw-r--r--  1 cry0l1t3  htb  234123 Feb 14 19:30 myscript.py
10678869 -rw-r--r--  1 cry0l1t3  htb   43230 Feb 14 11:52 notes.txt
```

Disks & Drives

Linux mein Disk Management ka matlab physical storage devices ko manage karna hota hai, jinmein Hard Drives, Solid-State Drives (SSD) aur Removable Storage Devices shamil hote hain.

Linux mein Disk Management ke liye main tool **`fdisk`** hai, jo humein kisi Drive par Partitions create, delete aur manage karne ki permission deta hai.

Ye Partition Table ki information bhi dikha sakta hai, jisme har Partition ka Size aur Type shamil hota hai.

Linux mein Drive ki Partitioning karne ka matlab physical storage space ko alag-alag logical sections mein divide karna hota hai.

Uske baad har Partition ko kisi specific File System, jaise **`ext4`**, **`NTFS`** ya **`FAT32`** ke saath format kiya ja sakta hai aur use ek separate File System ki tarah Mount kiya ja sakta hai.

Linux mein sabse commonly use hone wale Partitioning Tools **`fdisk`**, **`gpart`** aur **`GParted`** hain.

Fdisk

```
Toxicx01@htb[/htb]$ sudo fdisk -l

Disk /dev/vda: 160 GiB, 171798691840 bytes, 335544320 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x5223435f

Device     Boot     Start       End   Sectors  Size Id Type
/dev/vda1  *         2048 158974027 158971980 75.8G 83 Linux
/dev/vda2       158974028 167766794   8792767  4.2G 82 Linux swap / Solaris

Disk /dev/vdb: 452 KiB, 462848 bytes, 904 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

Mounting

Har Logical Partition ya Storage Drive ko File System ki kisi specific Directory ke saath assign karna zaroori hota hai. Is process ko **`Mounting`** kaha jata hai.

Mounting ka matlab kisi Drive ya Partition ko ek Directory ke saath link karna hota hai, jisse uske andar ke contents poore File System Hierarchy ke andar accessible ho jaate hain.

Jab kisi Drive ko kisi Directory ke saath Mount kar diya jata hai (jise Mount Point bhi kaha jata hai), toh us Drive ko system ki kisi bhi doosri Directory ki tarah access aur use kiya ja sakta hai.

Linux mein File Systems ko manually Mount karne ke liye commonly **`mount`** command ka use kiya jata hai.

Lekin agar aap chahte hain ki kuch specific File Systems ya Partitions system boot hone par automatically Mount ho jaayein, toh aap unhe **`/etc/fstab`** file mein define kar sakte hain.

Is file mein File Systems aur unke associated Mount Points ki list hoti hai.

Iske saath hi ismein kuch options bhi hote hain, jaise Read/Write Permissions aur File System Types.

Isse specific Drives ya Partitions system startup hote hi available ho jaate hain aur unhe manually Mount karne ki zaroorat nahi padti.

Mounted File systems at Boot

```shellsession
Toxicx01@htb[/htb]$ cat /etc/fstab

# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a device; this may
# be used with UUID= as a more robust way to name devices that works even if
# disks are added and removed. See fstab(5).
#
# <file system>                     <mount point>  <type>  <options>  <dump>
<pass>
UUID=3d6a020d-...SNIP...-9e085e9c927a /              btrfs
subvol=@,defaults,noatime,nodiratime,nodatacow,space_cache,autodefrag 0 1
UUID=3d6a020d-...SNIP...-9e085e9c927a /home          btrfs
subvol=@home,defaults,noatime,nodiratime,nodatacow,space_cache,autodefrag 0 2
UUID=21f7eb94-...SNIP...-d4f58f94e141 swap           swap    defaults,noatime 0 0
```

Currently mounted File Systems ko dekhne ke liye hum bina kisi argument ke `mount` command ka use kar sakte hain.

Iska output currently mounted sabhi File Systems ki list dikhayega, jisme Device Name, File System Type, Mount Point aur Options shamil honge.

List Mounted Drives

```shellsession
Toxicx01@htb[/htb]$ mount

sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs
(rw,nosuid,relatime,size=4035812k,nr_inodes=1008953,mode=755,inode64)
devpts on /dev/pts type devpts
(rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs
(rw,nosuid,nodev,noexec,relatime,size=814580k,mode=755,inode64)
/dev/vda1 on / type btrfs
(rw,noatime,nodiratime,nodatasum,nodatacow,space_cache,autodefrag,subvolid=257,subv
```

Kisi File System ko Mount karne ke liye hum `mount` command ke baad Device Name aur Mount Point dete hain.

Example ke liye, agar humein `/dev/sdb1` naam wali USB Drive ko `/mnt/usb` Directory mein Mount karna hai, toh hum neeche di gayi command ka use karenge:

Mount a USB drive

```
Toxicx01@htb[/htb]$ sudo mount /dev/sdb1 /mnt/usb
Toxicx01@htb[/htb]$ cd /mnt/usb && ls -l

total 32
drwxr-xr-x 1 root root   18 Oct 14  2021 'Account Takeover'
drwxr-xr-x 1 root root   18 Oct 14  2021 'API Key Leaks'
drwxr-xr-x 1 root root   18 Oct 14  2021 'AWS Amazon Bucket S3'
drwxr-xr-x 1 root root   34 Oct 14  2021 'Command Injection'
drwxr-xr-x 1 root root   18 Oct 14  2021 'CORS Misconfiguration'
drwxr-xr-x 1 root root   52 Oct 14  2021 'CRLF Injection'
drwxr-xr-x 1 root root   30 Oct 14  2021 'CSRF Injection'
drwxr-xr-x 1 root root   18 Oct 14  2021 'CSV Injection'
drwxr-xr-x 1 root root 1166 Oct 14  2021 'CVE Exploits'
...SNIP...
```

Linux mein kisi File System ko Unmount karne ke liye hum **`umount`** command ke baad us File System ka Mount Point dete hain, jise hum Unmount karna chahte hain.

Mount Point File System ke andar woh location hoti hai, jahan File System ko Mount kiya gaya hota hai aur jahan se hum use access kar sakte hain.

Example ke liye, agar humein USB Drive ko Unmount karna hai, jise pehle **`/mnt/usb`** Directory mein Mount kiya gaya tha, toh hum neeche di gayi command ka use karenge:

Unmount

```
Toxicx01@htb[/htb]$ sudo umount /mnt/usb
```

Ye dhyan rakhna zaroori hai ki kisi File System ko Unmount karne ke liye hamare paas sufficient permissions honi chahiye.

Hum kisi aise File System ko bhi Unmount nahi kar sakte, jiska use koi Running Process kar raha ho.

Ye check karne ke liye ki koi Running Process File System ka use toh nahi kar raha, hum `lsof` command ka use kar sakte hain.

Ye command File System par open files ki list dikhati hai.

```
cry0l1t3@htb:~$ lsof | grep cry0l1t3

vncserver 6006        cry0l1t3  mem       REG      0,24       402274 /usr/bin/perl
(path dev=0,26)
vncserver 6006        cry0l1t3  mem       REG      0,24      1554101
/usr/lib/locale/aa_DJ.utf8/LC_COLLATE (path dev=0,26)
vncserver 6006        cry0l1t3  mem       REG      0,24       402326
/usr/lib/x86_64-linux-gnu/perl-base/auto/POSIX/POSIX.so (path dev=0,26)
vncserver 6006        cry0l1t3  mem       REG      0,24       402059
/usr/lib/x86_64-linux-gnu/perl/5.32.1/auto/Time/HiRes/HiRes.so (path dev=0,26)
vncserver 6006        cry0l1t3  mem       REG      0,24      1444250
/usr/lib/x86_64-linux-gnu/libnss_files-2.31.so (path dev=0,26)
vncserver 6006        cry0l1t3  mem       REG      0,24       402327
/usr/lib/x86_64-linux-gnu/perl-base/auto/Socket/Socket.so (path dev=0,26)
vncserver 6006        cry0l1t3  mem       REG      0,24       402324
/usr/lib/x86_64-linux-gnu/perl-base/auto/IO/IO.so (path dev=0,26)
...SNIP...
```

Agar humein aise koi Processes milte hain jo File System ka use kar rahe hain, toh File System ko Unmount karne se pehle humein un Processes ko Stop karna hoga.

`/etc/fstab` file mein un File Systems ki information hoti hai jinhe System par Mount kiya jana chahiye. Ismein Boot Time par Automatic Mounting aur doosre Mount Options ki information bhi hoti hai.

Sabhi Mounted File Systems (chahe woh `/etc/fstab` mein listed hon ya manually Mount kiye gaye hon) System Shutdown ke dauran OS ke through automatically aur properly Unmount kar diye jaate hain.

Agar hum chahte hain ki koi File System Boot Time par automatically Mount na ho, toh humein `/etc/fstab` file mein uski Entry ke saath `noauto` option add karna hoga.

Example ke liye, ye kuch is tarah dikhega:

Fstab File

```
/dev/sda1 / ext4 defaults 0 0
/dev/sda2 /home ext4 defaults 0 0
/dev/sdb1 /mnt/usb ext4 rw,noauto,user 0 0
192.168.1.100:/nfs /mnt/nfs nfs defaults 0 0
```

SWAP

Swap Space Linux mein Memory Management ka ek important part hai. Ye System ki Performance ko smooth banaye rakhne mein important role play karta hai, especially jab available Physical Memory (RAM) poori tarah use ho jaati hai.

Jab System ki Physical Memory khatam ho jaati hai, toh Kernel inactive Memory Pages (aisa data jo turant use nahi ho raha hota) ko Swap Space mein move kar deta hai.

Isse Active Processes ke liye RAM free ho jaati hai. Is process ko Swapping kaha jata hai.

Creating Swap Space

Swap Space ko Operating System install karte waqt setup kiya ja sakta hai ya baad mein **`mkswap`** aur **`swapon`** commands ka use karke add kiya ja sakta hai.

- **`mkswap`** ka use kisi Device ya File ko Swap Space ke roop mein use karne ke liye prepare karne mein hota hai. Ye ek Linux Swap Area create karta hai.
- **`swapon`** Swap Space ko activate karta hai, jisse System iska use kar sakta hai.

Sizing and Managing Swap Space

`Swap Space` ka size fixed nahi hota. Ye aapke System ki Physical Memory (RAM) aur uske intended usage par depend karta hai.

Example ke liye, kam RAM wale System ya Memory-intensive Applications chalane wale System ko zyada Swap Space ki zaroorat ho sakti hai.

Lekin zyada RAM wale Modern Systems ko kam ya bilkul bhi Swap Space ki zaroorat nahi ho sakti, ye unke specific use cases par depend karta hai.

Swap Space setup karte waqt ye important hai ki ise ek Dedicated Partition ya File par allocate kiya jaye, jo baaki File System se alag ho.

Isse Fragmentation prevent hoti hai aur zaroorat padne par Swap Area ka efficiently use hota hai.

Iske alawa, kyunki Sensitive Data temporarily Swap Space mein store ho sakta hai, isliye Data Exposure se bachne ke liye Swap Space ko Encrypt karne ki recommendation di jaati hai.

Swap Space for Hibernation

Physical Memory ko extend karne ke alawa, Swap Space ka use `Hibernation` ke liye bhi kiya jata hai.

Hibernation ek Power-Saving Feature hai, jo System ki current state (open Applications aur Processes sahit) ko Swap Space mein save karta hai aur phir System ko completely Power Off kar deta hai.

Jab System ko dobara Power On kiya jata hai, toh ye Swap Space se apni previous state ko Restore kar leta hai.

Isse System bilkul wahi se Resume hota hai, jahan se usne kaam karna chhoda tha.

Q. How many partitions exist in our Pwnbox? (Format: 0)

Ans.

Containerization

Containerization ek process hai jisme Applications ko isolated environments mein package aur run kiya jata hai. In environments ko usually Containers kaha jata hai.

Ye Containers Applications ko lightweight aur consistent environments provide karte hain, jisse Applications same tarike se kaam karti hain, chahe unhe kahin bhi deploy kiya gaya ho.

Docker, Docker Compose aur Linux Containers (LXC) jaisi Technologies Containerization ko possible banati hain. Inka use mainly Linux-based Systems mein hota hai.

Containers aur Virtual Machines mein difference ye hai ki Containers Host System ke Kernel ko share karte hain. Is wajah se Containers, Virtual Machines ke comparison mein kaafi lightweight aur efficient hote hain.

In Technologies ki help se Users Applications ko quickly Create, Deploy aur Manage kar sakte hain, jisse Security, Portability aur Scalability improve hoti hai.

Containers ko bahut achhe se Configure kiya ja sakta hai, jisse Users unhe apni specific needs ke according customize kar sakte hain.

Inki Lightweight Nature ki wajah se ek hi Host System par multiple Containers ko ek saath run karna aasan hota hai.

Ye Feature Applications ko Scale karne aur Complex Microservice Architectures ko Manage karne mein kaafi useful hota hai.

Imagine karo ki aap ek bada Concert organize kar rahe ho aur har Band ko apne according customized Stage Setup chahiye.

Har Band ke liye ek bilkul naya Stage banane ke bajaye (jo Virtual Machine use karne jaisa hoga), aap portable aur self-contained "Stage Pods" banate ho.

In Stage Pods mein Band ki zaroorat ki har cheez hoti hai, jaise Lights, Instruments, Speakers etc.

Ye Pods lightweight aur reusable hote hain aur inhe aasani se ek Venue se doosre Venue par le jaya ja sakta hai.

Sabse important baat ye hai ki ye sabhi Pods ek hi Main Stage (Host System) par bina kisi problem ke kaam karte hain, lekin har Pod itna Isolated hota hai ki kisi ek Band ka Setup doosre Band ke Setup ko disturb nahi karta.

Isi tarah, Containers kisi Application ko uske sabhi required Tools aur Settings ke saath Package karte hain.

Isse Application alag-alag Systems par bina kisi Conflict ke consistently run kar sakti hai, aur saath hi underlying System ke Resources (Main Stage) ko share karti hai.

Security, Containerization ka ek bahut important aspect hai.

Containers Applications ko Host System aur ek-doosre se Isolate karte hain. Ye ek Security Barrier provide karte hain, jo kisi bhi Malicious Activity ke Host System ya doosre Containers ko affect karne ke Risk ko kam karta hai.

Ye Isolation, proper Configuration aur Hardening Techniques ke saath milkar Security ki ek additional layer provide karta hai.

Lekin ye dhyan rakhna important hai ki Containers, Traditional Virtual Machines jitna Isolation provide nahi karte.

Agar Containers ko properly Manage nahi kiya gaya, toh Privilege Escalation ya Container Escape jaisi Vulnerabilities ka fayda uthakar koi Attacker Host System ya doosre Containers mein Unauthorized Access gain kar sakta hai.

Enhanced Security aur Resource Efficiency ke alawa, Containers Applications ko Deploy, Manage aur Scale karna bhi aasan banate hain.

Containers Application ki zaroorat ki har cheez (jaise Libraries aur Dependencies) ko apne andar rakhte hain.

Isse Development, Testing aur Production Environments mein Consistency bani rehti hai.

Ye Portability ensure karti hai ki Applications alag-alag Environments mein reliably run karein.

Lekin ye samajhna bhi zaroori hai ki apne advantages ke bawajood, Containers Security Risks se completely safe nahi hote.

Aise methods exist karte hain jinka use karke hum Privileges Escalate kar sakte hain ya Containers dwara provide ki gayi Isolation se Escape kar sakte hain.

Docker

Docker ek open-source platform hai jo applications ko self-contained units ke roop mein deploy karne ke process ko automatically manage karta hai. In units ko Containers kaha jata hai.

Ye flexibility aur portability provide karne ke liye layered filesystem aur resource isolation features ka use karta hai.

Iske alawa, ye applications ko create, deploy aur manage karne ke liye tools ka ek powerful set provide karta hai, jo containerization ke process ko aasan banata hai.

Example:

Docker containers ko ek sealed lunchbox ki tarah samjho. Tum lunchbox ke andar ka khana kha sakte ho (applications run kar sakte ho), lekin jab tum lunchbox ko band kar dete ho (container stop kar dete ho), toh sab kuch reset ho jata hai.

Updated contents (modified configurations) ke saath ek naya lunchbox (new container) banane ke liye, tum original ke basis par ek nayi recipe (Dockerfile) create karte ho.

Jab ek restaurant mein multiple lunchboxes serve karne hote hain (production), toh sabhi orders ko smoothly manage karne ke liye ek kitchen system (Kubernetes/Docker Compose) ka use kiya jata hai.

Docker-Engine Install karna

Docker ko install karna kaafi aasan hai. Hum Ubuntu host par Docker install karne ke liye neeche di gayi script ka use kar sakte hain:

```
#!/bin/bash

# Preparation
sudo apt update -y
sudo apt install ca-certificates curl gnupg lsb-release -y
sudo mkdir -m 0755 -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/etc/apt/keyrings/docker.gpg
echo "deb [arch=$(dpkg --print-architecture) signed-
by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list >
/dev/null

# Install Docker Engine
sudo apt update -y
sudo apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-
compose-plugin -y

# Add user htb-student to the Docker group
sudo usermod -aG docker htb-student
echo '[!] You need to log out and log back in for the group changes to take
effect.'

# Test Docker installation
```

Container ko run karne ke liye Docker Engine aur specific Docker Images ki zaroorat hoti hai. Ye humein **Docker Hub** se mil sakti hain, jo pehle se bani hui images ka ek repository hai, ya phir user khud bhi images create kar sakta hai.

Docker Hub ek cloud-based registry hai, jahan software repositories ya Docker Images ki library hoti hai. Ye do parts mein divided hota hai: `public` aur `private`।

Public Area: Is area mein users images upload kar sakte hain aur unhe community ke saath share kar sakte hain. Ismein Docker development team aur established open-source projects ki official images bhi hoti hain.

Private Area: Registry ke private area mein upload ki gayi images publicly accessible nahi hoti hain. Inhe kisi company, teams ya jaan-pehchaan wale logon ke saath share kiya ja sakta hai.

Docker Image create karne ke liye **Dockerfile** banayi jaati hai. Ismein woh saari instructions hoti hain jinki Docker Engine ko container create karne ke liye zaroorat hoti hai.

Hum specific files ko apne target systems par transfer karte waqt Docker containers ko apne "file hosting" server ki tarah use kar sakte hain.

Isliye, humein Ubuntu 22.04 par based ek `Dockerfile` create karni hogi, jismein `Apache` aur `SSH` server running honge.

Iski madad se hum `scp` ka use karke Docker Image mein files transfer kar sakte hain. Aur Apache humein files host karne ki permission deta hai, jisse hum target system par `curl`, `wget` aur doosre tools ka use karke required files download kar sakte hain.

Aisi ek `Dockerfile` kuch is tarah dikh sakti hai:

Dockerfile

bash

```
# Use the latest Ubuntu 22.04 LTS as the base image
FROM ubuntu:22.04

# Update the package repository and install the required packages
RUN apt-get update && \
    apt-get install -y \
        apache2 \
        openssh-server \
        && \
    rm -rf /var/lib/apt/lists/*

# Create a new user called "docker-user"
RUN useradd -m docker-user && \
    echo "docker-user:password" | chpasswd

# Give the docker-user user full access to the Apache and SSH services
RUN chown -R docker-user:docker-user /var/www/html && \
    chown -R docker-user:docker-user /var/run/apache2 && \
    chown -R docker-user:docker-user /var/log/apache2 && \
    chown -R docker-user:docker-user /var/lock/apache2 && \
    usermod -aG sudo docker-user && \
    echo "docker-user ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers

# Expose the required ports
EXPOSE 22 80

# Start the SSH and Apache services
CMD service ssh start && /usr/sbin/apache2ctl -D FOREGROUND
```

Jab hum apni Dockerfile define kar lete hain, toh humein use ek image mein convert karna hota hai.

`build` command se hum us directory ko lete hain jismein Dockerfile hoti hai, Dockerfile ke steps execute karte hain aur image ko apne local Docker Engine mein store karte hain.

Agar kisi step mein error aata hai, toh container creation process rok diya jayega.

`-t` option se hum apne container ko ek tag dete hain, jisse baad mein use pehchanna aur uske saath kaam karna aasaan ho jaata hai.

Docker Build

```
Toxicx01@htb[/htb]$ docker build -t FS_docker .
```

Ek baar Docker image create ho jaati hai, toh use Docker Engine ke through execute kiya ja sakta hai. Isse container run karna bahut efficient aur easy ho jaata hai.

Yeh concept Virtual Machine jaisa hi hai, jo images par based hota hai.

Lekin ye images Read-only templates hoti hain aur ye runtime ke liye zaroori file system aur saare parameters provide karti hain.

Ek container ko kisi image ke running process ke roop mein samjha ja sakta hai.

Jab kisi system par container start karna hota hai, toh sabse pehle uski respective image ka package load kiya jaata hai, agar woh image system mein pehle se available nahi hai.

Hum neeche diye gaye command se container start kar sakte hain:

`docker run`

Docker Run - Syntax

```
Toxicx01@htb[/htb]$ docker run -p 8022:22 -p 8080:80 -d FS_docker
```

Docker Run

```
Toxicx01@htb[/htb]$ docker run -p 8022:22 -p 8080:80 -d FS_docker
```

Is case mein, hum FS_docker image se ek naya container start karte hain aur host ke ports 8022 aur 8080 ko respectively container ke ports 22 aur 80 ke saath map karte hain.

Container background mein run hota hai, jisse hum diye gaye host ports ka use karke container ke andar chal rahi SSH aur HTTP services ko access kar sakte hain.

Docker Management

Docker containers ko manage karte waqt, Docker humein tools ka ek complete set provide karta hai, jinki help se hum aasani se containers ko create, deploy aur manage kar sakte hain.

In powerful tools ki help se hum containers ki list dekh sakte hain, unhe start aur stop kar sakte hain aur effectively manage kar sakte hain.

Isse applications bina kisi problem ke smoothly execute hoti hain.

Docker ke kuch sabse commonly used management commands hain:

Command	Description
`docker ps`	Saare running containers ki list dekhna.
`docker stop`	Running container ko stop karna.

`docker start`	Stopped container ko dobara start karna.
`docker restart`	Running container ko restart karna.
`docker rm`	Container ko remove karna.
`docker rmi`	Docker image ko remove karna.
`docker logs`	Container ke logs dekhna.

Yeh dhyan rakhna important hai ki Docker commands ke saath extra functionality add karne ke liye alag-alag options use kiye ja sakte hain.

Example ke liye, aap specify kar sakte hain ki kaun-se ports expose karne hain, data ko save rakhne ke liye volumes mount kar sakte hain, ya apne containers ko configure karne ke liye environment variables set kar sakte hain.

Yeh flexibility aapko apni specific needs aur requirements ke according Docker containers ko customize karne ki permission deti hai.

Jab Docker images ke saath kaam karte hain, toh yeh samajhna bahut zaroori hai ki image par based running container mein kiye gaye changes automatically image mein save nahi hote.

In changes ko preserve karne ke liye, aapko ek nayi image create karni hoti hai, jismein ye changes included hon.

Yeh kaam ek nayi Dockerfile likhkar kiya jaata hai, jo `FROM` statement se start hoti hai (ismein base image specify ki jaati hai). Iske baad changes apply karne ke liye zaroori commands likhe jaate hain.

Jab Dockerfile ready ho jaati hai, toh aap `docker build` command ka use karke nayi image build kar sakte hain aur use identify karne ke liye ek unique tag de sakte hain.

Yeh process ensure karta hai ki original image unchanged rahe, jabki nayi image mein saare updates included hon.

Yeh yaad rakhna bhi important hai ki Docker containers design ke hisaab se stateless hote hain. Iska matlab hai ki running container ke andar kiye gaye changes (jaise files ko modify karna), container stop ya remove hone ke baad lost ho jaate hain.

Isi wajah se data ko permanently save rakhne ya application state store karne ke liye container ke bahar volumes use karna best practice hai.

Production environments mein, jab bahut saare containers ko manage karna hota hai, toh yeh process aur complex ho jaata hai.

Docker Compose ya Kubernetes jaise tools containers ko orchestrate karne mein help karte hain. Inki help se aap multiple containers ko efficiently manage, scale aur ek-doosre ke saath link kar sakte hain.

Linux Containers

Linux Containers (`LXC`) ek lightweight virtualization technology hai, jo ek hi host par multiple isolated Linux systems (jinhe containers kaha jaata hai) run karne ki permission deti hai.

LXC resource isolation ke important features ka use karta hai, jaise Control Groups (`cgroups`) aur `namespaces`, taaki har container independently kaam kar sake.

Traditional Virtual Machines ke opposite, jinmein har instance ke liye ek complete OS ki zaroorat hoti hai, containers host ka kernel share karte hain. Isse LXC resources ke use ke maamle mein zyada efficient hota hai.

LXC containers ko manage aur configure karne ke liye tools aur APIs ka ek complete set provide karta hai, jiski wajah se Linux systems mein containerization ke liye yeh ek popular choice hai.

Lekin LXC aur Docker dono containerization technologies hain, phir bhi inka purpose alag hai aur dono ke apne unique features hain.

Docker containerization ke concept ko aur aage badhata hai. Yeh use karne mein aasani aur portability provide karta hai, jiski wajah se DevOps ki duniya mein yeh bahut popular ho gaya hai.

Docker applications ko unki saari dependencies ke saath ek portable Image mein package karne par focus karta hai, taaki unhe alag-alag environments mein aasani se deploy kiya ja sake.

Lekin in dono ke beech kuch differences hain, jinhe neeche di gayi categories ke basis par samjha ja sakta hai:

Category	Description
Approach	LXC ko ek traditional, system-level containerization tool maana jaata hai. Yeh isolated Linux environments create karne par focus karta hai, jo lightweight Virtual Machines ki tarah behave karte hain. Dusri taraf, Docker application-focused hai, yani ise single applications ya microservices ko package aur deploy karne ke liye optimize kiya gaya hai.
Image building	Docker ek standardized image format (Docker Images) use karta hai, jismein application run karne ke liye zaroori sabhi cheezein (code, libraries, configurations) included hoti hain. LXC bhi isi tarah ki functionality provide kar sakta hai, lekin ismein environments ko build aur manage karne ke liye generally zyada manual setup karna padta hai.
Portability	Docker portability ke maamle mein bahut achha hai. Iski container images ko Docker Hub ya doosri registries ke through alag-alag systems par aasani se share kiya ja sakta hai. Is comparison mein LXC environments kam portable hote hain, kyunki ye host system ki configuration ke saath zyada closely integrated hote hain.
Ease of use	Docker ko simplicity ko dhyan mein rakhkar design kiya gaya hai. Yeh user-friendly CLI aur extensive community support provide karta hai. LXC powerful hai, lekin ise use karne ke liye Linux system administration ki zyada knowledge ki zaroorat ho sakti hai, jiski wajah se beginners ke liye ise samajhna thoda mushkil ho sakta hai.

Security	Docker containers generally shuru se hi zyada secure hote hain, kyunki inmein AppArmor aur SELinux jaise additional isolation layers ke saath read-only filesystem feature bhi hota hai. LXC containers bhi secure hote hain, lekin Docker ke default isolation level ke barabar security paane ke liye inmein additional configurations ki zaroorat ho sakti hai. Interesting baat yeh hai ki agar Docker aur LXC dono ko galat tarike se configure kiya jaaye, toh dono local privilege escalation ka raasta ban sakte hain. In techniques ko hamare **Linux Local Privilege Escalation module** mein detail mein cover kiya gaya hai.

LXC mein, images manually build ki jaati hain. Iske liye ek root filesystem create kiya jaata hai aur usmein zaroori packages aur configurations install ki jaati hain.

Ye containers host system se connected hote hain, isliye inhe ek system se doosre system par aasani se move nahi kiya ja sakta. Inhe configure aur manage karne ke liye zyada technical knowledge ki zaroorat ho sakti hai.

Dusri taraf, Docker ek application-centric platform hai, jo LXC ke concept par build kiya gaya hai aur containerization ke liye zyada user-friendly interface provide karta hai.

Docker ki images ek `Dockerfile` ka use karke build ki jaati hain. Is Dockerfile mein base image aur image build karne ke liye zaroori steps specify kiye jaate hain.

Ye images portable hone ke liye design ki gayi hain, taaki inhe aasani se ek environment se doosre environment mein move kiya ja sake.

Linux distribution mein LXC install karne ke liye hum us distribution ke package manager ka use kar sakte hain.

Example ke liye, Ubuntu mein hum `apt` package manager ka use karke neeche di gayi command se LXC install kar sakte hain:

Install LXC

```
Toxicx01@htb[/htb]$ sudo apt install lxc -y
```

LXC install hone ke baad, hum Linux host par containers create aur manage karna shuru kar sakte hain.

Yeh dhyan dene wali baat hai ki LXC ko containerization ke liye zaroori features support karne wale Linux kernel ki zaroorat hoti hai.

Aajkal ke most modern Linux kernels mein containerization ka support pehle se built-in hota hai, lekin kuch purane kernels mein LXC support enable karne ke liye additional configuration ya patching ki zaroorat ho sakti hai.

Creating an LXC Container

Ek naya LXC container create karne ke liye hum **`lxc-create`** command ka use kar sakte hain. Iske baad container ka naam aur use kiya jaane wala template specify karte hain.

Example ke liye, **`linuxcontainer`** naam ka ek naya Ubuntu container create karne ke liye hum neeche di gayi command ka use kar sakte hain:

```
Toxicx01@htb[/htb]$ sudo lxc-create -n linuxcontainer -t ubuntu
```

Managing LXC Containers

LXC containers ke saath kaam karte waqt, unhe manage karne ke liye kai tasks perform karne padte hain.

In tasks mein naye containers create karna, unki settings configure karna, zaroorat ke according unhe start aur stop karna aur unki performance monitor karna shamil hai.

Achhi baat yeh hai ki in tasks mein help karne ke liye bahut saare command-line tools aur configuration files available hain.

Ye tools humein apne containers ko jaldi aur aasani se manage karne ki permission dete hain, taaki unhe apni specific needs aur requirements ke according optimize kiya ja sake.

In tools ka effectively use karke hum ensure kar sakte hain ki hamare LXC containers efficiently aur effectively run karein. Isse hum apne system ki performance aur capabilities ka maximum use kar sakte hain.

Network, storage aur security settings ko **`lxc-config`** se manage nahi kiya jaata.

Yeh command sirf global LXC system parameters (jaise **`lxc.lxcpath`**) ko check karta hai.

Har container ki apni settings uski configuration file mein hoti hain. Privileged containers ke liye yeh file generally yahan hoti hai:

`/var/lib/lxc/<container>/config`

In settings ki documentation **`lxc.container.conf(5)`** mein di gayi hai.

Classic LXC tools (`lxc-start`, `lxc-attach`) aur LXD ke space-separated `lxc config` client ko ek jaisa nahi samajhna chahiye. Yeh dono alag hain.

Command	Description
`lxc-ls`	Saare existing containers ki list dikhata hai.
`lxc-info -n <container>`	Container ka status, PID, IP addresses aur resource statistics dikhata hai.
`lxc-stop -n <container>`	Running container ko stop karta hai.
`lxc-start -n <container>`	Stopped container ko start karta hai.
`lxc-stop -r -n <container>`	Running container ko reboot karta hai.
`lxc-cgroup -n <container> <key> [value]`	Running container ke live cgroup value ko check ya set karta hai.
`lxc-attach -n <container>`	Running container ke andar shell ya command execute karta hai.
`lxc-attach -n <container> -f /path/to/config`	Custom configuration file (`--rcfile`) ka use karke container se attach hota hai, shared host path ka use nahi karta.

`lxc-console -n <container>`	Container ke console se connect karta hai.

lxc-config can list or print global keys only:

```
Toxicx01@htb[/htb]$ lxc-config -l
Toxicx01@htb[/htb]$ lxc-config lxc.lxcpath
```

Kisi container ki network settings ko check ya change karne ke liye, us container ki `config` file mein `lxc.net.0.type`, `lxc.net.0.link` aur `lxc.net.0.flags` jaise keys ko edit karein. Iske baad container ko restart karein.

Aap kisi ek key ki value ya assigned IP addresses ko bhi display kar sakte hain:

```
Toxicx01@htb[/htb]$ sudo lxc-info -n linuxcontainer -i
Toxicx01@htb[/htb]$ sudo lxc-info -n linuxcontainer -c lxc.net.0.veth.pair
```

Host ki kisi directory ya file ko container ke saath share karne ke liye, lxc-attach mein path dene ke bajaye container ki configuration file mein bind mount add karein:

```
lxc.mount.entry = /path/on/host path/in/container none bind,create=dir 0 0
```

Entry add karne ke baad, mount ko apply karne ke liye container ko restart karein.

Penetration testers ke roop mein, humein aksar aisi situations ka saamna karna padta hai, jahan humein aise software ya systems test karne hote hain jinmein complex dependencies ya configurations hoti hain. In configurations ko apni local machines par dobara create karna mushkil hota hai.

Yahin par Linux Containers bahut useful hote hain.

Linux Container ek lightweight, standalone package hota hai, jismein kisi specific software ko run karne ke liye zaroori sabhi cheezein hoti hain, jaise code, libraries aur configuration files.

Containers ek isolated environment provide karte hain, jisse software kisi bhi Linux machine par consistently run kar sakta hai, chahe host system ki configuration kuch bhi ho.

Containers khaas taur par useful hote hain kyunki ye humein apni testing requirements ke according isolated environments jaldi create aur run karne ki permission dete hain.

Example ke liye, agar humein kisi web application ko test karna hai jo database ya web server ke kisi specific version par depend karti hai, toh hum apni zaroorat ke exact versions aur configurations ke saath ek container create kar sakte hain.

Isse humein apni machine par in components ko manually setup karne ki zaroorat nahi padti. Manual setup mein kaafi time lag sakta hai aur errors hone ki possibility bhi hoti hai.

Hum containers ka use exploits ya malware test karne ke liye bhi kar sakte hain. Iske liye hum ek controlled environment create karte hain, jismein ek vulnerable system ya network ko simulate kiya jaata hai.

Phir hum us container ka use karke safely exploits test kar sakte hain, bina apni machines ya networks ko damage karne ka risk liye.

Lekin unauthorized access ya container ke andar malicious activities ko rokne ke liye LXC container ki security configure karna important hai.

Yeh kaam kai security measures implement karke kiya ja sakta hai, jaise:

- Container tak access ko restrict karna.
- Resources ko limit karna.
- Container ko host system se isolate karna.
- Mandatory Access Control enforce karna.
- Container ko up to date rakhna.

LXC containers ko alag-alag methods se access kiya ja sakta hai, jaise SSH ya console.

Container tak access ko restrict karne ke liye unnecessary services disable karna, secure protocols ka use karna aur strong authentication mechanisms enforce karna recommended hai.

Example ke liye, hum `openssh-server` package ko remove karke ya SSH ko sirf trusted IP addresses se access allow karne ke liye configure karke container ka SSH access disable kar sakte hain.

Ye containers host system ke saath same kernel share karte hain. Iska matlab hai ki ye system par available resources ko access kar sakte hain.

Containers ko zaroorat se zyada resources consume karne se rokne ke liye hum resource limits ya quotas ka use kar sakte hain.

Example ke liye, hum `cgroups` ka use karke CPU, memory ya disk space ki quantity limit kar sakte hain, jiska use koi container kar sakta hai.

Securing LXC

Chaliye, container ke resources ko limit karte hain.

LXC ke liye **`cgroups`** configure karne aur CPU aur memory ko limit karne ke liye, container ki apni configuration file ko edit karein.

Privileged containers ki configuration file yahan hoti hai:

`/var/lib/lxc/<container name>/config`

`/usr/share/lxc/config/` directory mein sirf shared templates hoti hain, jaise **`common.conf`**. Yeh per-container resource limits add karne ki jagah nahi hai.

Example ke liye, agar humein **`linuxcontainer`** naam ke container ki configuration edit karni hai, toh hum neeche di gayi command ka use kar sakte hain:

```
Toxicx01@htb[/htb]$ sudo vim /var/lib/lxc/linuxcontainer/config
```

Is configuration file mein, hum neeche di gayi lines add kar sakte hain, taaki container jitna CPU aur memory use kar sakta hai, uski limit set ki ja sake.

```
lxc.cgroup.cpu.shares = 512
lxc.cgroup.memory.limit_in_bytes = 512M
```

Containers ke saath kaam karte waqt, **`lxc.cgroup.cpu.shares`** parameter ko samajhna zaroori hai.

Yeh parameter decide karta hai ki system ke doosre containers ke comparison mein ek container kitna CPU time use kar sakta hai.

Default roop se iski value **`1024`** hoti hai, jiska matlab hai ki container apne fair share ke according CPU time use kar sakta hai.

Lekin agar hum iski value **`512`** set kar dete hain, toh example ke liye, container system mein available CPU time ka sirf aadha hissa use kar sakta hai.

Yeh resources ko manage karne aur ensure karne ka ek useful tareeka hai, taaki sabhi containers ko zaroorat ke according CPU time mil sake.

Container ke resource allocation ko control karne ke liye ek important parameter **`lxc.cgroup.memory.limit_in_bytes`** hai.

Yeh parameter container ke liye maximum memory limit set karne ki permission deta hai, yani container maximum kitni memory use kar sakta hai.

Yeh dhyan rakhna zaroori hai ki is value ko alag-alag units mein specify kiya ja sakta hai, jaise bytes, kilobytes (**`K`**), megabytes (**`M`**), gigabytes (**`G`**) ya terabytes (**`T`**).

Isse container ke resource limits ko bahut accurately define kiya ja sakta hai.

In dono lines ko add karne ke baad, hum neeche diye gaye steps type karke file ko save aur close kar sakte hain:

- **`[Esc]`**
- **`:`**
- **`wq`**

In changes ko apply karne ke liye container ko restart karein, taaki woh apni configuration ko dobara load kar sake.

Sirf ek container ki **`config`** file mein changes karne ke liye **`lxc`** service ko restart karne ki zaroorat nahi hoti.

```
Toxicx01@htb[/htb]$ sudo lxc-stop -n linuxcontainer
Toxicx01@htb[/htb]$ sudo lxc-start -n linuxcontainer
```

Agar container pehle se running hai aur humein sirf temporary limit set karni hai, toh **`lxc-cgroup`** command ka use karke bina container restart kiye, live hi same values set kar sakte hain.

```
Toxicx01@htb[/htb]$ sudo lxc-cgroup -n linuxcontainer cpu.shares 512
Toxicx01@htb[/htb]$ sudo lxc-cgroup -n linuxcontainer memory.limit_in_bytes 512M
```

LXC, host system se processes, networks aur file systems ko alag rakhne ke liye **`namespaces`** ka use karta hai.

Namespaces, Linux kernel ka ek feature hai, jo system resources ka alag view provide karke isolated environments create karne ki permission deta hai.

Namespaces, containerization ka ek bahut important part hain, kyunki ye container ke processes, network interfaces, routing tables aur firewall rules ko high level ki isolation provide karte hain.

Har container ko ek unique Process ID (`pid`) number space diya jaata hai, jo host system ke process IDs se alag hota hai.

Isse container ke processes host system ke processes mein interfere nahi kar sakte, jisse system ki stability aur reliability badhti hai.

Iske alawa, har container ke apne network interfaces (`net`), routing tables aur firewall rules hote hain, jo host system ke network interfaces se completely alag hote hain.

Container ke andar hone wali koi bhi network-related activity host system ke network se alag rakhi jaati hai, jisse network security ki ek additional layer milti hai.

Iske alawa, containers ka apna root file system (`mnt`) hota hai, jo host system ke root file system se completely alag hota hai.

In dono ke beech yeh separation ensure karta hai ki container ke file system mein kiye gaye koi bhi changes ya modifications host system ke file system ko affect na karein.

Lekin yeh yaad rakhna important hai ki namespaces high level ki isolation provide karte hain, lekin complete security provide nahi karte.

Isliye container aur host system ko possible security breaches se aur zyada protect karne ke liye additional security measures implement karna hamesha recommended hai.

LXC ki practice karne ke liye yahan 9 optional exercises diye gaye hain:

No.	Exercise
1	Apni machine par LXC install karein aur apna pehla container create karein.
2	**`/var/lib/lxc/<container>/config`** mein **`lxc.net.0.*`** keys edit karke apne LXC container ki network settings configure karein.
3	Ek custom LXC image create karein aur uska use karke naya container launch karein.

4 Apne LXC containers ke liye resource limits (CPU, memory, disk space) configure karein.

5 Containers ko manage karne ke liye `lxc-*` commands explore karein.

6 LXC ka use karke kisi specific version ka web server (jaise Apache, Nginx) run karne wala container create karein.

7 Apne LXC containers mein SSH access configure karein aur unse remotely connect karein.

8 Ek aisa container create karein jismein persistence ho, taaki container mein kiye gaye changes save ho sakein aur dobara use kiye ja sakein.

9 LXC ka use karke controlled environment mein software test karein, jaise vulnerable web application ya malware.

Network Configuration

Penetration Tester ke liye Linux systems par network settings ko configure aur manage karna ek zaroori skill hai.

Is skill ko seekhne se hum testing environments ko aasani se set up kar sakte hain, network traffic ko manipulate kar sakte hain aur vulnerabilities ko identify ya exploit kar sakte hain.

Linux Network Configuration ki achhi understanding hone se hum apni zaroorat ke according testing approach ko customize kar sakte hain. Isse hamari testing process aur results dono better hote hain.

Network Configuration ke main tasks mein se ek hai Network Interfaces ko manage karna.

Ismein IP addresses assign karna, routers aur switches jaise network devices ko configure karna aur alag-alag network protocols ko set up karna shamil hai.

Humein TCP/IP (Internet communication ke liye main protocol suite), DNS (Domain Name Resolution), DHCP (automatically IP address assign karne ke liye) aur FTP (file transfer ke liye) jaise network protocols ki achhi understanding honi chahiye.

Humein alag-alag types ke network interfaces ke baare mein bhi pata hona chahiye, chahe wo wired ho ya wireless. Saath hi, network connectivity mein aane wali problems ko troubleshoot karna bhi aana chahiye.

Network Access Control (NAC)

Network Configuration ka ek aur important part hai Network Access Control (`NAC`).

Penetration Testers ke roop mein, humein achhe se pata hona chahiye ki NAC kis tarah network security ko improve kar sakta hai aur iske liye kaun-kaun si technologies available hain.

NAC ke kuch important models hain:

Type	Description
Discretionary Access Control (**`DAC`**)	Is model mein resource ka owner khud decide karta hai ki kaun us resource ko access kar sakta hai aur kise permission deni hai.
Mandatory Access Control (**`MAC`**)	Is model mein permissions resource ka owner nahi, balki Operating System control karta hai. Yeh zyada secure hota hai, lekin ismein flexibility kam hoti hai.
Role-Based Access Control (**`RBAC`**)	Is model mein users ko unke organization ke roles ke according permissions di jaati hain. Isse users ki permissions manage karna aasaan ho jata hai.

NAC ke liye Linux network devices ko configure karne mein security policies set karna shamil hai, jaise **SELinux** (Security-Enhanced Linux), applications ki security ke liye **AppArmor** profiles ka use karna aur IP addresses ke basis par services ka access control karne ke liye **TCP Wrappers** ka use karna.

Iske baare mein hum aage ke sections mein aur padhenge.

syslog, **rsyslog**, **ss** (socket statistics ke liye), `lsof` (open files ki list dekhne ke liye) aur **ELK Stack** (Elasticsearch, Logstash aur Kibana) jaise tools ka use network traffic ko monitor aur analyze karne ke liye kiya ja sakta hai.

Ye tools humein network mein hone wali unusual activities (anomalies), possible information disclosure/exposure, security breaches aur network se judi doosri important problems ko identify karne mein help karte hain.

Linux mein Network Configuration ko ek bade office building ko banane aur uski security karne jaisa samjho.

Network Interfaces ko configure karna building ki wiring aur infrastructure set up karne jaisa hai, jisse har room (network device) mein proper connection mil sake.

NAC building ki security manage karne jaisa hai, jahan kuch rooms sabhi ke liye open hote hain (**DAC**), jabki kuch rooms mein sirf selected log hi strict rules ke according enter kar sakte hain (**MAC** ya **RBAC**).

Network Traffic ko monitor karna building mein CCTV cameras aur alarms lagane jaisa hai. Isse hum nazar rakh sakte hain ki building mein kaun aa-ja raha hai.

Troubleshooting ka matlab hai apne paas aise tools rakhna jinse kisi bhi problem ko fix kiya ja sake, chahe woh broken connection (**ping**), kharab lock (**nslookup**) ya vulnerable entrance (**nmap**) ho.

Is section mein hum NAC aur uske tools ke baare mein thodi der baad aur detail mein padhenge.

Configuring Network Interfaces

Ubuntu par kaam karte waqt, hum **ifconfig** ya **ip** command ka use karke local network interfaces ko configure kar sakte hain.

Ye powerful commands humein apne system ke network interfaces ko dekhne aur configure karne ki permission deti hain.

Chahe hum apne existing network setup mein changes karna chahte hon ya apne network interfaces ka status check karna chahte hon, ye commands is process ko bahut aasaan bana sakti hain.

Iske alawa, Network Interfaces ki working ko achhe se samajhna aaj ki modern aur interconnected duniya mein ek zaroori skill hai.

Technology tezi se advance ho rahi hai aur digital communication par hamari dependency badhti ja rahi hai. Aise mein, Network Interfaces ke saath kaam karne ki achhi knowledge hone se hum aaj ke samay mein available alag-alag networks ko effectively samajh aur use kar sakte hain.

Network Interfaces ke baare mein information, jaise IP addresses, netmasks aur status pata karne ka ek tarika `ifconfig` command ka use karna hai.

Is command ko execute karne par hum available network interfaces aur unki respective details ko ek clear aur organized way mein dekh sakte hain.

Yeh information network connectivity ki problems ko troubleshoot karne ya naya network configuration set up karne mein bahut useful ho sakti hai.

Dhyan dene wali baat hai ki Linux ke naye versions mein `ifconfig` command ko deprecated kar diya gaya hai aur iski jagah `ip` command use hoti hai, jo isse zyada advanced features provide karti hai.

Phir bhi, `ifconfig` command abhi bhi bahut saare Linux distributions mein widely use hoti hai aur network management ke liye ek reliable tool bani hui hai.

Network Settings

```shellsession
cry0l1t3@htb:~$ ifconfig

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 178.62.32.126  netmask 255.255.192.0  broadcast 178.62.63.255
        inet6 fe80::88d9:faff:fecf:797a  prefixlen 64  scopeid 0x20<link>
        ether 8a:d9:fa:cf:79:7a  txqueuelen 1000  (Ethernet)
        RX packets 7910  bytes 717102 (700.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 7072  bytes 24215666 (23.0 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.106.0.66  netmask 255.255.240.0  broadcast 10.106.15.255
        inet6 fe80::b8ab:52ff:fe32:1f33  prefixlen 64  scopeid 0x20<link>
        ether ba:ab:52:32:1f:33  txqueuelen 1000  (Ethernet)
        RX packets 14  bytes 1574 (1.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 15  bytes 1700 (1.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
```

```
cry0l1t3@htb:~$ ip addr

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
default qlen 1000
    link/ether 8a:d9:fa:cf:79:7a brd ff:ff:ff:ff:ff:ff
    altname enp0s3
    altname ens3
    inet 178.62.32.126/18 brd 178.62.63.255 scope global dynamic eth0
       valid_lft 85274sec preferred_lft 85274sec
    inet6 fe80::88d9:faff:fecf:797a/64 scope link
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
default qlen 1000
    link/ether ba:ab:52:32:1f:33 brd ff:ff:ff:ff:ff:ff
    altname enp0s4
    altname ens4
    inet 10.106.0.66/20 brd 10.106.15.255 scope global dynamic eth1
```

Network Interfaces ko activate karne ke liye **ifconfig** aur **ip** commands do commonly used tools hain.

Ye commands users ko kisi specific network interface, jaise **eth0**, ki settings ko modify aur activate karne ki permission deti hain.

Sahi syntax ka use karke aur interface ka naam specify karke, hum apni zaroorat ke according network settings ko adjust kar sakte hain.

Activate Network Interface

```
Toxicx01@htb[/htb]$ sudo ifconfig eth0 up     # OR
Toxicx01@htb[/htb]$ sudo ip link set eth0 up
```

Network Interface ko IP address assign karne ka ek tarika **`ifconfig`** command ka use karna hai.

Iske liye humein command mein network interface ka naam aur IP address specify karna hota hai.

Network connection set up karne ke liye yeh ek bahut important step hai.

IP address network interface ki ek unique identity hoti hai, jo network par maujood alag-alag devices ko ek-doosre ke saath communicate karne mein help karti hai.

Assign IP Address to an Interface

```
Toxicx01@htb[/htb]$ sudo ifconfig eth0 192.168.1.2
```

Network Interface ka netmask set karne ke liye, hum neeche di gayi command ko execute kar sakte hain, jismein network interface ka naam aur netmask specify karna hota hai:

Assign a Netmask to an Interface

```
Toxicx01@htb[/htb]$ sudo ifconfig eth0 netmask 255.255.255.0
```

Jab humein kisi Network Interface ke liye Default Gateway set karna hota hai, toh hum `route` command ke saath `add` option ka use kar sakte hain.

Isse hum Gateway ka IP address aur woh Network Interface specify kar sakte hain jis par ye apply karna hai.

Default Gateway set karne ka matlab hai router ka IP address decide karna, jiska use local network ke bahar kisi destination par data bhejne ke liye kiya jayega.

Default Gateway ko sahi tarike se set karna bahut zaroori hai, kyunki agar configuration galat hui toh network connectivity mein problems aa sakti hain.

Assign the Route to an Interface

```
Toxicx01@htb[/htb]$ sudo route add default gw 192.168.1.1 eth0
```

Linux mein Network Interface configure karte waqt, proper network functionality ensure karne ke liye Domain Name System (**DNS**) servers set karna aksar zaroori hota hai.

DNS servers ka kaam domain names (jaise **example.com**) ko IP addresses mein convert karna hota hai. Isse devices internet par ek-doosre ko find aur connect kar paate hain.

Websites, online services aur doosre network resources ko access karne ke liye DNS configuration ka sahi hona bahut zaroori hai.

Agar DNS servers properly configure nahi hain, toh devices ko domain names resolve karne mein problems aa sakti hain, jiske karan network connectivity issues ho sakte hain.

Linux systems mein ye kaam **/etc/resolv.conf** file ko update karke kiya ja sakta hai. Ye ek simple text file hoti hai, jismein system ki DNS information hoti hai.

Is file mein appropriate DNS server addresses add karke (Google ke public DNS – **8.8.8.8** ya **8.8.4.4**), system domain names ko sahi tarike se IP addresses mein convert kar sakta hai, jisse network par communication smoothly ho sake.

Editing DNS Settings

```
Toxicx01@htb[/htb]$ sudo vim /etc/resolv.conf
```

/etc/resolv.conf

```
nameserver 8.8.8.8
nameserver 8.8.4.4
```

Network Configuration mein zaroori changes karne ke baad, ye ensure karna bahut zaroori hai ki changes save ho jaayein, taaki system reboot hone ke baad bhi ye changes bane rahein.

Ye kaam **/etc/network/interfaces** file ko edit karke kiya ja sakta hai. Ye file Linux-based operating systems mein Network Interfaces ko define karti hai.

Isliye, network connectivity se judi kisi bhi possible problem se bachne ke liye, is file mein kiye gaye changes ko save karna zaroori hai.

Ye baat dhyan rakhna important hai ki **/etc/resolv.conf** file mein directly kiye gaye changes reboot hone ya network configuration change hone ke baad permanent nahi rehte.

Aisa isliye hota hai kyunki **NetworkManager** ya **systemd-resolved** jaise network management services is file ko automatically overwrite kar sakte hain.

DNS changes ko permanent banane ke liye, humein appropriate network management tool ke through DNS settings configure karni chahiye. Jaise network configuration files ko edit karna ya aisi network management utilities ka use karna jo settings ko permanently save karti hain.

Editing Interfaces

```
Toxicx01@htb[/htb]$ sudo vim /etc/network/interfaces
```

Isse `interfaces` file Vim editor mein open ho jayegi.

Hum is file mein network configuration ki settings is tarah add kar sakte hain:

/etc/network/interfaces

```
auto eth0
iface eth0 inet static
  address 192.168.1.2
  netmask 255.255.255.0
  gateway 192.168.1.1
  dns-nameservers 8.8.8.8 8.8.4.4
```

eth0 network interface ko static IP address **192.168.1.2** par set karke, netmask **255.255.255.0** aur default gateway **192.168.1.1** configure karne se hum apne network connection ko stable aur reliable bana sakte hain.

Iske alawa, DNS servers **8.8.8.8** aur **8.8.4.4** specify karke hum ensure kar sakte hain ki hamara computer aasani se internet access kar sake aur domain names ko IP addresses mein convert kar sake.

Configuration file mein ye changes karne ke baad, file ko save karna aur editor se bahar nikalna zaroori hai.

Iske baad, changes apply karne ke liye humein Networking Service ko restart karna hoga.

Restart Networking Service

```
Toxicx01@htb[/htb]$ sudo systemctl restart networking
```

Network Access Control (NAC)

Network Access Control (NAC) network security ka ek bahut important part hai, khaaskar aaj ke samay mein jab cyber threats lagatar badh rahe hain.

Ek Penetration Tester ke roop mein, humein NAC ki importance samajhna zaroori hai ki ye network ko kaise protect karta hai aur security improve karne ke liye kaun-kaun si NAC technologies use ki ja sakti hain.

NAC ek security system hai jo ensure karta hai ki sirf authorized aur security rules ko follow karne wale devices ko hi network access mile.

Ye unauthorized access, data breaches aur doosre security threats ko rokne mein help karta hai.

NAC implement karne se organizations ko bharosa hota hai ki unke assets aur data cybercriminals se protected hain. Cybercriminals hamesha system ki vulnerabilities ka fayda uthane ki koshish karte hain.

Security ko improve karne ke liye alag-alag NAC technologies ka use kiya ja sakta hai:

- Discretionary Access Control (**DAC**)
- Mandatory Access Control (**MAC**)
- Role-Based Access Control (**RBAC**)\

Ye technologies access control aur security ke alag-alag levels provide karne ke liye design ki gayi hain.

Har technology ki apni alag characteristics hoti hain aur ye different situations mein use ki jaati hain.

Ek Penetration Tester ke roop mein, in technologies aur unke specific use cases ko samajhna zaroori hai, taaki hum network ki security ko effectively test aur evaluate kar sakein.

Discretionary Access Control (DAC)

DAC modern security systems ka ek important part hai, kyunki ye organizations ko apne resources ka access dene aur unauthorized access se jude risks ko manage karne mein help karta hai.

Ye ek widely used Access Control System hai, jo users ko apne resources ka access manage karne ki permission deta hai.

Is system mein resource ke owner ko ye responsibility di jaati hai ki woh decide kare ki uske resources ki access permissions kiske paas hongi.

Iska matlab hai ki kisi specific resource ke owner users ya groups khud decide kar sakte hain ki:

- Kaun unke resources ko access kar sakta hai.
- Unhe resources par kaun-se actions perform karne ki permission hai.

Ye permissions resource ko read, write, execute ya delete karne ke liye set ki ja sakti hain.

Mandatory Access Control (MAC)

MAC ka use aise infrastructure mein kiya jata hai jahan DAC systems ke mukable resources ke access par zyada detailed control hota hai.

Ye systems kuch rules define karte hain, jo decide karte hain ki kisi resource ko access diya jayega ya nahi. Ye rules resource ke security level aur access maangne wale user ya process ke security level ke basis par kaam karte hain.

Har resource ko ek Security Label diya jata hai, jo uske security level ko identify karta hai.

Isi tarah, har user ya process ko ek Security Clearance diya jata hai, jo uske security level ko identify karta hai.

Kisi resource ka access tabhi diya jata hai jab user ya process ka security level, us resource ke security level ke barabar ya usse zyada ho.

MAC ka use aksar un Operating Systems aur Applications mein kiya jata hai jahan high-level security ki zaroorat hoti hai, jaise Military ya Government Systems, Financial Systems aur Healthcare Systems.

MAC systems ko is tarah design kiya gaya hai ki ye resources ko unauthorized access se bachayein aur security breach hone par uske impact ko kam karein.

Role-Based Access Control (RBAC)

RBAC mein users ko organization ke andar unke roles ke basis par permissions di jaati hain.

Users ko unki job responsibilities ya doosre criteria ke according roles assign kiye jaate hain. Har role ko permissions ka ek set diya jata hai, jo decide karta hai ki user kaun-se actions perform kar sakta hai.

RBAC access permissions ko manage karna aasaan banata hai, galtiyon ke risk ko kam karta hai aur ensure karta hai ki users sirf unhi resources ko access kar sakein jo unke job ke liye zaroori hain.

Ye sensitive resources aur data ka access restrict kar sakta hai, security breaches ke impact ko kam kar sakta hai aur regulatory requirements ko follow karna ensure karta hai.

Discretionary Access Control (DAC) systems ke comparison mein, RBAC resource access ko manage karne ka zyada flexible aur scalable approach provide karta hai.

RBAC system mein har user ko ek ya ek se zyada roles assign kiye jaate hain. Har role ko permissions ka ek set diya jata hai, jo define karta hai ki user kaun-se actions perform kar sakta hai.

Resource ka access user ki identity ya resource par ownership ke basis par nahi, balki uske assigned role ke basis par diya jata hai.

RBAC systems ka use aam taur par un environments mein hota hai jahan bahut saare users aur resources hote hain, jaise:

- Large Organizations
- Government Agencies
- Financial Institutions

Monitoring

Network Monitoring mein network traffic ko capture, analyze aur samjha jata hai, taaki security threats, performance problems aur suspicious activities ko identify kiya ja sake.

Network Traffic ko analyze aur monitor karne ka main purpose security threats aur vulnerabilities ko identify karna hai.

Example ke liye, ek Penetration Tester ke roop mein, hum kisi user ke credentials capture kar sakte hain, jab koi user unencrypted connection ka use karke FTP server par login karne ki koshish karta hai.

Is tarah humein us user ke credentials mil sakte hain, jo network mein aur andar tak pahunchne ya apne privileges ko higher level tak badhane mein hamari help kar sakte hain.

Short mein, Network Traffic ko analyze karke hum network ke behaviour ko samajh sakte hain aur aise patterns identify kar sakte hain jo security threats ka signal dete hain.

Is analysis mein shamil hain:

- Suspicious network activities ko detect karna.
- Malicious traffic ko identify karna.
- Possible security risks ko identify karna.

Lekin hum is bade topic ko **Intro to Network Traffic Analysis** module mein cover karenge.

Is module mein hum Linux systems (jaise Ubuntu) aur Windows systems par Network Monitoring ke liye kai tools ka use karenge, jaise Wireshark, tshark aur Tcpdump.

Troubleshooting

Network Troubleshooting ek zaroori process hai, jismein network ki problems ka pata lagana aur unhe solve karna shamil hai. Ye problems network ki performance aur reliability par bura asar daal sakti hain.

Ye process ensure karne ke liye important hai ki network achhe se kaam kare aur aisi rukawaton se bacha ja sake jo hamare penetration tests ke dauran business operations ko affect kar sakti hain.

Is process mein problems ko identify karna, unka analysis karna aur unhe solve karne ke liye solutions implement karna bhi shamil hai.

In problems mein shamil hain:

- Network Connectivity ki problems.
- Network ki slow speed.
- Network Errors.

Linux systems mein Network Troubleshooting se judi problems ko identify aur solve karne ke liye alag-alag tools available hain.

Kuch sabse zyada use hone wale tools hain:

1. Ping
2. Traceroute
3. Netstat
4. Tcpdump
5. Wireshark
6. Nmap

In tools aur inke jaise doosre tools ka use karke, hum achhe se samajh sakte hain ki network kaise kaam karta hai aur aane wali kisi bhi problem ka jaldi pata laga sakte hain.

Example ke liye, **`ping`** ek command-line tool hai, jiska use do devices ke beech connectivity test karne ke liye kiya jata hai.

Ye remote host ko data packets bhejta hai aur measure karta hai ki un packets ko wapas aane mein kitna time lagta hai.

`ping` ka use karne ke liye hum neeche di gayi command enter kar sakte hain:

Ping

```
Toxicx01@htb[/htb]$ ping <remote_host>
```

Example ke liye, Google DNS server ko `ping` karne par Google DNS server ko ICMP packets bheje jayenge aur unka response aane mein kitna time laga, ye display hoga.

```
Toxicx01@htb[/htb]$ ping 8.8.8.8

PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=119 time=1.61 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=119 time=1.06 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=119 time=0.636 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=119 time=0.685 ms
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3017ms
rtt min/avg/max/mdev = 0.636/0.996/1.607/0.388 ms
```

Ek aur tool hai **`traceroute`**, jo remote host tak pahunchne ke liye data packets jis route se travel karte hain, us route ko trace karta hai.

Ye remote host ko increasing Time-to-Live (**`TTL`**) values ke saath packets bhejta hai aur un devices ke IP addresses display karta hai, jinse hokar ye packets guzarte hain.

Example ke liye, **www.inlanefreight.com** tak pahunchne wale route ko trace karne ke liye hum neeche di gayi command enter karenge:

Traceroute

```
Toxicx01@htb[/htb]$ traceroute www.inlanefreight.com

traceroute to www.inlanefreight.com (134.209.24.248), 30 hops max, 60 byte packets
 1  * * *
 2  10.80.71.5 (10.80.71.5)  2.716 ms  2.700 ms  2.730 ms
 3  * * *
 4  10.80.68.175 (10.80.68.175)  7.147 ms  7.132 ms 10.80.68.161 (10.80.68.161)
7.393 ms
```

Output mein packets ke us path ki detailed information di gayi hai, jise follow karke packets Inlanefreight web server tak pahunchte hain. Yeh server **`www.inlanefreight.com`** par located hai, jiska IP address **`134.209.24.248`** hai.

Yeh un sabhi beech ke devices ke IP addresses dikhata hai, jinse hokar packets guzarte hain.

`traceroute` ke output ki har line mein useful information hoti hai, jo humein network ke route aur uski performance ko samajhne mein help karti hai.

Second line traceroute ka pehla hop dikhati hai, jo local network gateway hai. Iska IP address **`10.80.71.5`** hai. Iske baad ke agle teen columns dikhate hain ki bheje gaye teen packets ko gateway tak pahunchne mein kitna time laga (milliseconds mein): 2.716 ms, 2.700 ms aur 2.730 ms.

Next, hum traceroute ka second hop dekhte hain. Lekin is hop par device se koi response nahi mila. Isko IP address ki jagah teen asterisks (`***`) se dikhaya gaya hai. Iska matlab ho sakta hai ki device down hai, ICMP traffic block kar raha hai, ya network problem ki wajah se packets drop ho gaye.

Fourth line mein hum traceroute ka third hop dekh sakte hain, jismein do devices hain, jinke IP addresses **`10.80.68.175`** aur **`10.80.68.161`** hain. Iske baad ke agle teen columns dikhate hain ki teen packets ko pehle device tak pahunchne mein kitna time laga: 7.147 ms, 7.132 ms aur 7.393 ms.

Netstat

Netstat ka use active network connections aur unse jude ports ko display karne ke liye kiya jata hai.

Iska use network traffic ko identify karne aur connectivity se judi problems ko troubleshoot karne ke liye kiya ja sakta hai.

netstat ka use karne ke liye, hum following command enter kar sakte hain:

```shellsession
Toxicx01@htb[/htb]$ netstat -a

Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:5901          0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:sunrpc          0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:http            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN
...SNIP...
```

Is tool ka use karne par hum har connection ke baare mein detailed information milne ki ummeed kar sakte hain. Ismein use kiya gaya protocol, receive aur send kiye gaye bytes ki sankhya, local aur remote devices ke IP addresses, port numbers aur current connection state shamil hoti hai.

Output system par hone wali network activity ke baare mein useful information provide karta hai. Yeh chaar specific connections ko highlight karta hai, jo currently active hain aur specific ports par listening kar rahe hain.

In connections mein shamil hain:

- VNC remote desktop software
- Sun Remote Procedure Call service
- Web traffic ke liye HTTP protocol
- Secure remote shell access ke liye SSH protocol

Kaunsi service kis port ka use karti hai, yeh jaankar users network ki problems ko jaldi identify kar sakte hain aur unhe troubleshoot kar sakte hain.

Hamare penetration tests ke dauran sabse common network problems ye hongi:

- Network connectivity issues
- DNS resolution issues (yeh hamesha DNS ki hi problem hoti hai)
- Data packets ka loss hona
- Network performance issues

In problems ke sabse common causes hain:

- Galat tarike se configure kiye gaye firewalls ya routers.
- Damaged network cables ya connections.
- Galat network settings.
- Hardware failures.
- Galat DNS server settings ya DNS server failures.
- Galat tarike se configure ki gayi DNS entries.
- Network congestion.
- Purane network hardware ya galat tarike se configure ki gayi network settings.
- Unpatched software ya firmware aur missing security controls.

In common network problems aur unke causes ko samajhna important hai, taaki hum testing ke dauran network systems mein vulnerabilities ko effectively identify aur exploit kar sakein.

Hardening

Linux systems ko secure rakhne aur apni aur doosri companies ke data ko safe rakhne ke liye kai security mechanisms bahut effective hain.

Aise teen mechanisms hain: SELinux, AppArmor aur TCP wrappers.

Ye tools Linux systems ko alag-alag security threats se protect karne ke liye design kiye gaye hain. Jaise unauthorized access se lekar malicious attacks tak.

Ye sirf penetration testing ke dauran hi important nahi hain, jahan vulnerabilities ko find karne ke liye systems ko intentionally test kiya jata hai, balki real-world situations mein bhi important hain, jahan actual security breach ke serious consequences ho sakte hain (real-life data breach jitni serious bahut kam situations hoti hain).

In security measures ko implement karke aur potential attackers ke against proper protection setup karke, hum data leaks ke risk ko kaafi kam kar sakte hain aur apne systems ko secure rakh sakte hain.

Halanki, ye tools kuch similarities share karte hain, lekin inke beech kuch important differences bhi hain.

Security-Enhanced Linux

Security-Enhanced Linux (`SELinux`) ek Mandatory Access Control (`MAC`) system hai, jo Linux kernel ke andar integrated hota hai.

Ye security policies ko enforce karke system resources aur applications par access ko bahut detail mein control karta hai.

Ye policies system ke har process aur file ke permissions define karti hain. Isse kisi compromised process ya service se hone wale damage ko kaafi had tak limit kiya ja sakta hai.

SELinux low level par kaam karta hai. Ye strong security provide karta hai, lekin iske detailed controls ki wajah se ise configure aur manage karna complex ho sakta hai.

AppArmor

SELinux ki tarah, `AppArmor` bhi ek MAC system hai, jo system resources aur applications ke access ko control karta hai. Lekin yeh zyada simple aur user-friendly tarike se kaam karta hai.

AppArmor ko Linux Security Module (`LSM`) ke roop mein implement kiya gaya hai aur yeh application profiles ka use karta hai, jo define karti hain ki koi application kin resources ko access kar sakti hai.

Halanki, yeh SELinux jitna detailed control provide nahi karta, lekin AppArmor ko configure karna aksar zyada easy hota hai aur daily use ke liye ise generally zyada straightforward maana jata hai.

TCP Wrappers

`TCP wrappers` ek host-based network access control tool hai, jo incoming connections ke IP address ke basis par network services ka access restrict karta hai.

Jab koi network request aati hai, toh TCP wrappers us request ko intercept karta hai aur check karta hai ki request allowed ya denied IP addresses ki list mein hai ya nahi.

Yeh network services ka access control karne ka ek simple lekin effective tarika hai. Khaaskar unauthorized systems ko network resources access karne se rokne ke liye.

Halanki, yeh SELinux ya AppArmor jitna detailed control provide nahi karta, lekin basic network-level protection ke liye TCP wrappers ek excellent tool hai.

In teenon security mechanisms ki similarities ki baat karein, toh inka common goal Linux systems ki safety aur security ensure karna hai.

Extra protection provide karne ke saath-saath, ye resources aur services ke access ko restrict kar sakte hain. Isse unauthorized access aur data breaches ka risk kam ho jata hai.

Yeh bhi dhyan dene wali baat hai ki ye mechanisms most Linux distributions mein pehle se available hote hain, jisse hum inka use karke apne systems ki security improve kar sakte hain.

Iske alawa, standard tools aur utilities ka use karke in mechanisms ko easily customize aur configure kiya ja sakta hai. Isliye ye Linux users ke liye ek convenient choice hain.

Halanki `SELinux` aur `AppArmor` dono MAC systems hain, jo detailed access control provide karte hain, lekin dono alag-alag tarike se kaam karte hain.

SELinux kernel ke saath deeply integrated hota hai aur zyada detailed security controls provide karta hai, lekin ise configure aur maintain karna thoda complex ho sakta hai.

Iske opposite, AppArmor kernel module ke roop mein kaam karta hai aur profile-based security ka use karta hai. Is wajah se ise manage karna easy hota hai, lekin yeh SELinux jitna detailed control provide nahi kar sakta.

Doosri taraf, `TCP wrappers` client ke IP address ke basis par network services ka access control karne par focus karta hai.

Is wajah se yeh simple hai, lekin iska control sirf network-level access tak limited hai.

Yeh SELinux aur AppArmor ki tarah poore system ke resources ko protect nahi karta, lekin unauthorized systems ko services access karne se rokne ke liye useful hai.

Setting Up

Jab hum Linux ki duniya mein aage badhte hain, toh humein alag-alag technologies, applications aur services ka saamna karna padta hai, jinse humein familiar hona zaroori hai.

Yeh ek bahut important skill hai, khaaskar agar hum cybersecurity mein kaam karte hain aur apni knowledge aur expertise ko continuously improve karna chahte hain.

Isi wajah se hum strongly recommend karte hain ki aap khud se `SELinux`, `AppArmor` aur `TCP wrappers` jaise important security measures ko configure karna seekhne ke liye time nikalein.

Is optional lekin bahut useful challenge ko complete karne se:

- In technologies ki aapki understanding aur deep hogi.
- Aapki problem-solving skills improve hongi.
- Aapko valuable practical experience milega, jo future mein aapke kaam aayega.

Hum strongly recommend karte hain ki changes karne se pehle apni personal VM ka use karein aur uska snapshot bana lein.

Cybersecurity measures implement karne ke liye koi ek fixed approach nahi hoti, jo har situation mein kaam kare.

Yeh consider karna important hai ki aap kis specific information ko protect karna chahte hain aur uske liye kaun-se tools use karenge.

Halanki, aap apni knowledge aur skills badhane ke liye Discord channel mein doosre logon ke saath milkar kai optional tasks ki practice aur implementation kar sakte hain.

Doosron ki help lekar aur apni knowledge share karke, aap cybersecurity ki apni understanding ko aur deep kar sakte hain aur doosron ko bhi seekhne mein help kar sakte hain.

Yaad rakhein, doosron ko concepts samjhana teaching aur learning ka ek important part hai.

SELinux

No.	Task
1.	Apni VM par SELinux install karein.
2.	SELinux ko is tarah configure karein ki koi specific user ek particular file ko access na kar sake.
3.	SELinux ko is tarah configure karein ki sirf ek particular user ko ek specific network service access karne ki permission mile aur baaki sabhi users ko access deny ho.
4.	SELinux ko is tarah configure karein ki kisi specific user ya group ko ek particular network service access karne ki permission na mile.

AppArmor

No.	Task
5.	AppArmor ko is tarah configure karein ki koi specific user ek particular file ko access na kar sake.
6.	AppArmor ko is tarah configure karein ki sirf ek particular user ko ek specific network service access karne ki permission mile aur baaki sabhi users ko access deny ho.
7.	AppArmor ko is tarah configure karein ki kisi specific user ya group ko ek particular network service access karne ki permission na mile.

TCP Wrappers

No.	Task
8.	TCP wrappers ko is tarah configure karein ki kisi specific IP address se ek particular network service ko access karne ki permission mile.
9.	TCP wrappers ko is tarah configure karein ki kisi specific IP address se ek particular network service ka access deny ho jaye.
10.	TCP wrappers ko is tarah configure karein ki IP addresses ki ek specific range se kisi particular network service ko access karne ki permission mile.

Remote Desktop Protocols in Linux

Remote desktop protocols ka use Windows, Linux aur macOS mein kisi system ko remotely graphical access provide karne ke liye kiya jata hai.

Ye protocols administrators ko remotely systems manage karne, troubleshoot karne aur update karne ki permission dete hain. Isliye aise situations mein ye bahut important tools hain.

Iske liye administrator, jis operating system ko manage kar raha hai, uske according appropriate protocol ka use karke remote system se connect karta hai.

Example ke liye, jab administrators ko kisi remote system par software install karna hota hai ya use manage karna hota hai, toh woh graphical session establish karne ke liye relevant protocol ka use karte hain.

Is tarah ke access ke liye do sabse common protocols hain:

- **`Remote Desktop Protocol`** (**`RDP`**): Iska use mainly Windows environments mein kiya jata hai. RDP administrators ko remotely Windows machine se connect karne aur uske desktop ke saath interact karne ki permission deta hai, bilkul aise jaise woh us computer ke saamne baithe hon.

- **`Virtual Network Computing`** (**`VNC`**): Yeh Linux environments mein ek popular protocol hai, halanki yeh multiple platforms par bhi kaam karta hai. VNC remote desktops ka graphical access provide karta hai, jisse administrators Linux systems par bhi tasks perform kar sakte hain, bilkul Windows mein RDP ki tarah.

Example se samjho:

Remote desktop protocols ko alag-alag buildings ki alag-alag keys ki tarah samjho.

RDP ek aisi key ki tarah hai jo specially Windows buildings ke liye banayi gayi hai. Iski help se aap remotely rooms (desktops) ko access aur manage kar sakte ho, bilkul aise jaise aap building ke andar hi ho.

Doosri taraf, VNC ek universal key ki tarah hai, jo kai tarah ki buildings mein kaam kar sakti hai, lekin iska use aksar Linux systems mein hota hai.

Jaise aap building ke according sahi key choose karte ho, waise hi administrators jis system ko access aur control karna chahte hain, uske according sahi protocol choose karte hain.

XServer

XServer, **X Window System network protocol** (**X11** / **X**) ka user-side part hota hai.

X11 ek fixed system hai, jismein protocols aur applications ka collection hota hai. Ye humein Graphical User Interface (GUI) mein displays par application windows open karne ki permission deta hai.

X11 ka use mainly Unix systems mein hota hai, lekin X servers doosre operating systems ke liye bhi available hain.

Aajkal XServer, Ubuntu aur uske derivatives ke almost har desktop installation ka part hota hai aur ise separately install karne ki zaroorat nahi hoti.

Jab Linux computer par desktop start hota hai, toh Graphical User Interface (GUI) aur operating system ke beech communication X server ke through hota hai.

Is process mein computer ka internal network use hota hai, chahe computer kisi network se connected na bhi ho.

X protocol ki ek useful feature Network Transparency hai.

Ye protocol mainly TCP/IP ko transport base ke roop mein use karta hai, lekin ise pure Unix sockets par bhi use kiya ja sakta hai.

X server ke liye use hone wale ports generally **TCP/6001-6009** range mein hote hain, jo client aur server ke beech communication allow karte hain.

Jab X server ke through ek naya desktop session start hota hai, toh pehle X display :0 ke liye **TCP port 6000** open hota hai.

Ports ki ye range server ko apne tasks perform karne mein help karti hai, jaise applications ko host karna aur clients ko services provide karna.

Inka use aksar kisi system ko remote access provide karne ke liye bhi hota hai, jisse users duniya mein kahin se bhi applications aur data access kar sakte hain.

Iske alawa, ye ports files aur data ko securely share karne ke liye bhi important hain, jisse ye Open X Server ka ek important part ban jaate hain.

Is tarah X server local computer par dependent nahi hota. Iska use doosre computers ko access karne ke liye bhi kiya ja sakta hai aur doosre computers bhi local X server ka use kar sakte hain.

Agar local aur remote dono computers mein Unix/Linux systems hain, toh additional protocols jaise VNC aur RDP ki zaroorat nahi hoti.

VNC aur RDP mein graphical output remote computer par generate hota hai aur network ke through transfer kiya jata hai.

Jabki X11 mein graphical output local computer par render hota hai.

Isse network traffic aur remote computer par load dono kam hote hain.

Lekin X11 ka ek bada disadvantage hai ki ismein data transmission encrypted nahi hota.

Halanki, SSH protocol ki tunneling karke is problem ko solve kiya ja sakta hai.

Iske liye humein application provide karne wale server ki SSH configuration file (**/etc/ssh/sshd_config**) mein X11 forwarding ko allow karna hoga.

Iske liye is option ki value **yes** karni hogi.

X11Forwarding

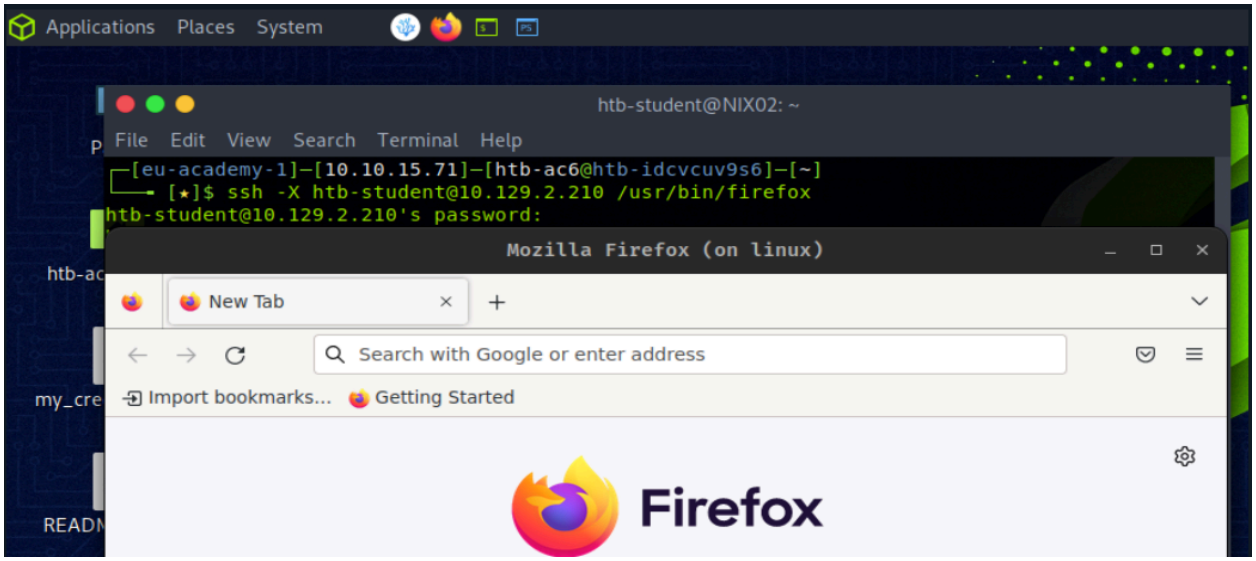
```shellsession
Toxicx01@htb[/htb]$ cat /etc/ssh/sshd_config | grep X11Forwarding

X11Forwarding yes
```

Iske baad hum following command ka use karke apne client se application start kar sakte hain:

```shellsession
Toxicx01@htb[/htb]$ ssh -X htb-student@10.129.23.11 /usr/bin/firefox

htb-student@10.129.14.130's password: ********
<SKIP>
```



X11 Security

Jaisa ki humne pehle bataya tha, X11 default roop se secure protocol nahi hai, kyunki iska communication encrypted nahi hota.

Isliye jab hum Linux-based targets ke saath kaam karte hain, toh humein TCP ports (`6000-6010`) par dhyan dena chahiye.

Proper security measures ke bina, ek open **`X server`** network par sensitive data expose kar sakta hai.

Example ke liye, same network par maujood attacker, user ki jaankari ke bina **`X server`** ki windows ka content read kar sakta hai. Iske liye attacker ko traditional network sniffing karne ki bhi zaroorat nahi hoti.

Yeh vulnerability serious security breaches ka reason ban sakti hai.

Attacker standard X11 tools jaise **`xwd`** (jo X windows ke screenshots capture karta hai) aur **`xgrabsc`** ka use karke passwords ya personal data jaisi sensitive information intercept kar sakta hai.

Iske alawa, pichhle kuch saalon mein XServer libraries aur software se related kai doosri security vulnerabilities bhi discover hui hain.

Example ke liye, 2017 mein XOrg Server mein vulnerabilities ka ek collection mila tha. XOrg Server, X Window System ka ek open-source implementation hai.

Ye vulnerabilities weak, predictable ya brute-force ki ja sakne wali session keys ki wajah se thi.

In vulnerabilities ka fayda uthakar attacker kisi doosre user ke Xorg session mein arbitrary code execute kar sakta tha.

Inse kai systems affected hue the, jaise:

- Unix
- Red Hat Enterprise Linux
- Ubuntu Linux
- SUSE Linux

In vulnerabilities ko **`CVE-2017-2624`**, **`CVE-2017-2625`** aur **`CVE-2017-2626`** ke naam se jaana gaya.

Is article mein in vulnerabilities ka ek achha summary diya gaya hai.

VNC

`Virtual Network Computing` (`VNC`) ek remote desktop sharing system hai, jo RFB protocol par based hai aur users ko kisi computer ko remotely control karne ki permission deta hai.

Yeh user ko network connection ke through remotely desktop environment dekhne aur uske saath interact karne ki permission deta hai.

User remote computer ko bilkul aise control kar sakta hai, jaise woh uske saamne baitha ho.

Yeh Linux hosts ke liye remote graphical connections ke sabse common protocols mein se ek hai.

VNC ko generally secure maana jata hai. Yeh data ko transfer ke dauran safe rakhne ke liye encryption ka use karta hai aur user ko access dene se pehle authentication ki requirement rakhta hai.

Administrators VNC ka use un computers ko access karne ke liye karte hain, jin tak physically pahunchna possible nahi hota.

Iska use servers ko troubleshoot aur maintain karne, doosre computers par applications access karne ya workstations ko remotely access karne ke liye kiya ja sakta hai.

VNC ka use screen sharing ke liye bhi kiya ja sakta hai, jisse multiple users kisi project par milkar kaam kar sakte hain ya kisi problem ko troubleshoot kar sakte hain.

VNC servers ke do alag concepts hote hain.

Pehla, normal server host computer ki actual screen users ko provide karta hai, jisse user support diya ja sake.

Kyunki remote computer par keyboard aur mouse ka use bhi kiya ja sakta hai, isliye ek proper arrangement recommend kiya jata hai.

Doosra group aise server programs ka hota hai, jo users ko virtual sessions mein login karne ki permission dete hain. Yeh concept terminal server jaisa hota hai.

VNC ke server aur viewer programs sabhi common operating systems ke liye available hain.

Isi wajah se kai IT services mein VNC ka use kiya jata hai.

Proprietary software TeamViewer aur RDP ka use bhi isi tarah ke kaamon ke liye hota hai.

Traditionally, VNC server `TCP port 5900` par listen karta hai. Isliye yeh apna `display 0` isi port par provide karta hai.

Doosre displays ko additional ports ke through provide kiya ja sakta hai, jo mostly **`590[x]`** hote hain. Yahan `x` display number ko represent karta hai.

Multiple connections add karne par unhe higher TCP ports assign kiye jaate hain, jaise:

- **`5901`**
- **`5902`**
- **`5903`**
- **etc.**

In VNC connections ke liye kai alag-alag tools ka use kiya jata hai. Unmein se kuch hain:

- **TigerVNC**
- **TightVNC**
- **RealVNC**
- **UltraVNC**

Is tarah ke connections ke liye sabse zyada use kiye jaane wale tools **`UltraVNC`** aur **`RealVNC`** hain, kyunki ye encryption aur higher security provide karte hain.

Is example mein, hum ek **`TigerVNC`** server setup karenge. Iske liye humein **`XFCE4`** desktop manager ki bhi zaroorat hogi, kyunki GNOME ke saath VNC connections thode unstable hote hain.

Isliye humein zaroori packages install karne honge aur VNC connection ke liye ek password create karna hoga.

TigerVNC Installation

```shellsession
htb-student@ubuntu:~$ sudo apt install xfce4 xfce4-goodies tigervnc-standalone-server -y
htb-student@ubuntu:~$ vncpasswd

Password: ******
Verify: ******
Would you like to enter a view-only password (y/n)? n
```

Installation ke dauran, home directory mein **`.vnc`** naam ka ek hidden folder create hota hai.

Iske baad humein do additional files create karni hoti hain: **`xstartup`** aur **`config`**.

`xstartup` decide karta hai ki display manager ke saath connection mein VNC session kaise create hoga.

Aur `config` file VNC session ki settings decide karti hai.

Configuration

```shellsession
htb-student@ubuntu:~$ touch ~/.vnc/xstartup ~/.vnc/config
htb-student@ubuntu:~$ cat <<EOT >> ~/.vnc/xstartup

#!/bin/bash
unset SESSION_MANAGER
unset DBUS_SESSION_BUS_ADDRESS
/usr/bin/startxfce4
[ -x /etc/vnc/xstartup ] && exec /etc/vnc/xstartup
[ -r $HOME/.Xresources ] && xrdb $HOME/.Xresources
x-window-manager &
EOT
```

```shellsession
htb-student@ubuntu:~$ cat <<EOT >> ~/.vnc/config

geometry=1920x1080
dpi=96
EOT
```

Iske alawa, `xstartup` executable file ko aise permissions deni hongi, jisse service ise start kar sake.

```
htb-student@ubuntu:~$ chmod +x ~/.vnc/xstartup
```

Ab hum VNC server ko start kar sakte hain.

Start VNC Server

```
htb-student@ubuntu:~$ vncserver

New 'linux:1 (htb-student)' desktop at :1 on machine linux

Starting applications specified in /home/htb-student/.vnc/xstartup
Log file is /home/htb-student/.vnc/linux:1.log

Use xtigervncviewer -SecurityTypes VncAuth -passwd /home/htb-student/.vnc/passwd :1
to connect to the VNC server.
```

Iske alawa, hum sabhi sessions ko unke associated ports aur Process ID (PID) ke saath bhi display kar sakte hain.

List Sessions

```
htb-student@ubuntu:~$ vncserver -list

TigerVNC server sessions:

X DISPLAY #     RFB PORT #      PROCESS ID
:1              5901            79746
```

Connection ko encrypt karne aur use zyada secure banane ke liye, hum ek SSH tunnel bana sakte hain, jiske through poora connection tunnel kiya jaata hai.

Tunneling kaam kaise karti hai, iske baare mein hum Pivoting, Tunneling, and Port Forwarding module mein detail mein seekhenge.

Setting Up an SSH Tunnel

```shellsession
Toxicx01@htb[/htb]$ ssh -L 5901:127.0.0.1:5901 -N -f -l htb-student 10.129.14.130

htb-student@10.129.14.130's password: *******
```

Aakhir mein, hum **`xtightvncviewer`** ka use karke SSH tunnel ke through server se connect kar sakte hain.

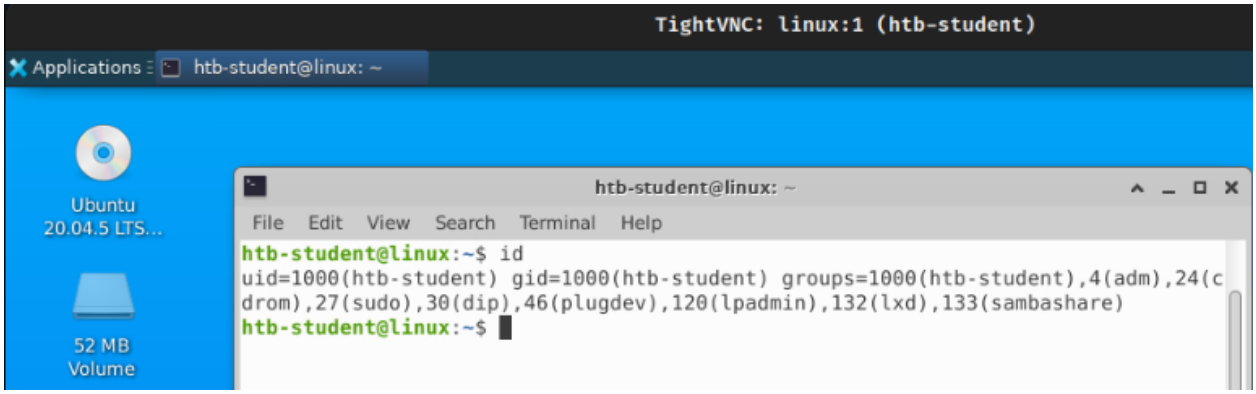
Connecting to the VNC Server

```shellsession
Toxicx01@htb[/htb]$ xtightvncviewer localhost:5901

Connected to RFB server, using protocol version 3.8
Performing standard VNC authentication

Password: ******

Authentication successful
Desktop name "linux:1 (htb-student)"
VNC server default format:
  32 bits per pixel.
```



Linux Security

Har computer system mein intrusion (bina permission ke system mein ghusne) ka risk hota hai.

Kuch computer systems mein doosron ke comparison mein zyada risk hota hai. Jaise, internet se connected ek web server, jo multiple complex web applications host karta hai.

Linux systems mein Windows operating systems ko affect karne wale viruses ka risk comparatively kam hota hai. Iske alawa, Linux systems ka attack surface bhi Active Directory domain-joined hosts ke comparison mein itna bada nahi hota.

Phir bhi, kisi bhi Linux system ko secure rakhne ke liye kuch basic security measures ka hona zaroori hai.

Linux operating system ke sabse important security measures mein se ek hai OS aur installed packages ko hamesha up to date rakhna.

Yeh kaam is tarah ki command ka use karke kiya ja sakta hai:

```
Toxicx01@htb[/htb]$ apt update && apt dist-upgrade
```

Agar network level par firewall rules sahi tarike se set nahi kiye gaye hain, toh hum Linux firewall aur/ya **`iptables`** ka use karke host ke andar aane wale aur bahar jaane wale traffic ko restrict kar sakte hain.

Agar server par SSH open hai, toh uski configuration is tarah set honi chahiye ki password login allow na ho aur root user ko SSH ke through login karne ki permission na ho.

Jab bhi possible ho, root user ke roop mein login karne aur system ko manage karne se bachna bhi important hai. Saath hi, access control ko sahi tarike se manage karna chahiye.

Users ko access dene ka decision Principle of Least Privilege ke according hona chahiye. Iska matlab hai ki user ko sirf utni hi permissions milni chahiye jitni uske kaam ke liye zaroori hain.

Example ke liye, agar kisi user ko root ke roop mein koi command run karni hai, toh us command ko **`sudoers`** configuration mein specify karna chahiye, na ki us user ko full sudo rights de dene chahiye.

Ek aur common protection mechanism **`fail2ban`** hai. Yeh tool failed login attempts ki counting karta hai. Agar koi user maximum allowed failed login attempts tak pahunch jata hai, toh connect karne ki koshish karne wale host ke saath configured settings ke according action liya jayega.

System ka time-time par audit karna bhi important hai, taaki aisi problems ka pata chal sake jo privilege escalation ko possible bana sakti hain.

Jaise:

- Out-of-date kernel
- User permission se judi problems
- World-writable files
- Galat tarike se configured cron jobs
- Galat tarike se configured services

Kai administrators yeh bhool jaate hain ki kuch kernel versions ko manually update karna padta hai.

Linux systems ko aur zyada secure banane ka ek option **`Security-Enhanced Linux`** (**`SELinux`**) ya **`AppArmor`** hai.

Yeh ek kernel security module hai, jiska use security access control policies ke liye kiya ja sakta hai.

SELinux mein har process, file, directory aur system object ko ek label diya jata hai.

In labeled processes aur objects ke beech access control karne ke liye policy rules banaye jaate hain, jinhe kernel enforce karta hai.

Iska matlab hai ki access ko is tarah configure kiya ja sakta hai ki kaunse users aur applications kin resources ko access kar sakte hain.

SELinux bahut detailed access controls provide karta hai. Jaise, yeh specify karna ki kaun kisi file mein data append kar sakta hai ya us file ko move kar sakta hai.

Iske alawa, alag-alag applications aur services bhi hain, jaise **Snort**, **chkrootkit**, **rkhunter**, **Lynis** aur doosre tools, jo Linux ki security ko improve karne mein help kar sakte hain.

Iske alawa, kuch security settings bhi configure karni chahiye, jaise:

- Sabhi unnecessary services aur software ko remove ya disable karna.
- Un sabhi services ko remove karna jo unencrypted authentication mechanisms par depend karti hain.
- Ensure karna ki NTP enabled ho aur Syslog running ho.
- Ensure karna ki har user ka apna alag account ho.
- Strong passwords ka use compulsory karna.
- Password aging set karna aur purane passwords ke dobara use ko restrict karna.
- Login failures ke baad user accounts ko lock karna.
- Sabhi unwanted SUID/SGID binaries ko disable karna.

Yeh list complete nahi hai, kyunki security koi product nahi, balki ek process hai.

Iska matlab hai ki systems ko aur zyada secure banane ke liye hamesha specific steps lene padte hain. Yeh administrators par depend karta hai ki unhe apne operating systems ki kitni achhi knowledge hai.

Administrators jitna achhe se apne system ko samjhenge aur jitni zyada unki training hogi, unki security precautions aur security measures utne hi achhe aur effective honge.

TCP Wrappers

TCP Wrapper ek security mechanism hai, jiska use Linux systems mein kiya jata hai. Yeh system administrator ko control karne ki permission deta hai ki kaun-si services ko system access karne ki permission hai.

Yeh access ko user ke hostname ya IP address ke basis par restrict karta hai, jo system ko access karne ki request kar raha hai.

Jab koi client kisi service se connect karne ki koshish karta hai, toh system sabse pehle TCP Wrappers ki configuration files mein define kiye gaye rules ko check karta hai, taaki client ka IP address pata chal sake.

Agar IP address configuration files mein diye gaye criteria se match karta hai, toh system client ko us service ka access de deta hai.

Lekin agar criteria match nahi hote hain, toh connection deny kar diya jata hai. Is tarah service ko security ki ek additional layer milti hai.

TCP Wrappers mein ye configuration files use hoti hain:

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

Short mein, **`/etc/hosts.allow`** file specify karti hai ki kaun-si services aur hosts ko system access karne ki permission hai.

Wahin, **`/etc/hosts.deny`** file specify karti hai ki kaun-si services aur hosts ko access karne ki permission nahi hai.

In files mein specific rules add karke inhe configure kiya ja sakta hai.

/etc/hosts.allow

```
Toxicx01@htb[/htb]$ cat /etc/hosts.allow

# Allow access to SSH from the local network
sshd : 10.129.14.0/24

# Allow access to FTP from a specific host
ftpd : 10.129.14.10

# Allow access to Telnet from any host in the inlanefreight.local domain
telnetd : .inlanefreight.local
```

/etc/hosts.deny

```
Toxicx01@htb[/htb]$ cat /etc/hosts.deny

# Deny access to all services from any host in the inlanefreight.com domain
ALL : .inlanefreight.com

# Deny access to SSH from a specific host
sshd : 10.129.22.22

# Deny access to FTP from hosts with IP addresses in the range of 10.129.22.0 to
10.129.22.255
ftpd : 10.129.22.0/24
```

Yeh yaad rakhna important hai ki in files mein rules ka order important hota hai.

Jo pehla rule requested service aur host se match karta hai, wahi apply kiya jata hai.

Yeh bhi dhyan rakhna important hai ki TCP Wrappers firewall ka replacement nahi hain, kyunki inki ek limitation hai: yeh sirf services ke access ko control kar sakte hain, ports ke access ko nahi.

Firewall Setup

Firewall ka main purpose ek security mechanism provide karna hai, jo different network segments ke beech network traffic ko control aur monitor karta hai. Jaise internal aur external networks ya alag-alag network zones ke beech.

Firewall computer networks ko unauthorized access, malicious traffic aur doosre security threats se protect karne mein important role play karta hai.

Linux ek popular operating system hai, jiska use servers aur doosre network devices mein kiya jata hai. Linux mein built-in firewall capabilities hoti hain, jinka use network traffic ko control karne ke liye kiya ja sakta hai.

Doosre words mein, firewall pehle se define kiye gaye rules, protocols, ports aur doosre criteria ke according incoming aur outgoing traffic ko filter kar sakta hai, taaki unauthorized access ko roka ja sake aur security threats ko kam kiya ja sake.

Firewall implement karne ka specific purpose organization ki zarooraton ke according alag ho sakta hai. Jaise network resources ki confidentiality (gopniyata), integrity (data ki sahi condition) aur availability (resources ki availability) ko ensure karna.

Linux firewall ki history ka ek example **`iptables`** tool ka development hai, jisne pehle ke **`ipchains`** aur **`ipfwadm`** tools ki jagah le li thi.

`iptables` utility ko pehli baar 2000 mein Linux 2.4 kernel mein introduce kiya gaya tha. Isne network traffic ko filter karne ke liye ek flexible aur efficient mechanism provide kiya tha.

`iptables` Linux systems ke liye ek standard firewall solution ban gaya tha aur ise bahut saari organizations aur users ne apnaya tha.

`iptables` utility firewall rules configure karne ke liye ek simple lekin powerful command-line interface provide karti thi, jiska use different criteria ke basis par network traffic ko filter karne ke liye kiya ja sakta tha. Jaise IP addresses, ports, protocols aur doosre criteria.

`iptables` ko bahut zyada customize karne ke liye design kiya gaya tha. Iska use complex firewall rulesets create karne ke liye kiya ja sakta tha, jo different security threats se protect kar sakte the.

Jaise:

- Denial-of-Service (DoS) attacks
- Port scans
- Network intrusion attempts

Linux mein firewall functionality generally Netfilter framework ka use karke implement ki jaati hai, jo kernel ka ek important part hai.

Netfilter hooks ka ek set provide karta hai, jinka use network traffic ko intercept (beech mein pakadna) aur modify (badalna) karne ke liye kiya ja sakta hai, jab network traffic system se hokar pass hota hai.

Linux systems mein firewall rules configure karne ke liye commonly **`iptables`** utility ka use kiya jata hai.

Iptables

`iptables` utility network traffic ko filter karne ke liye flexible rules provide karti hai. Ye rules different criteria ke basis par traffic ko filter kar sakte hain, jaise source aur destination IP addresses, port numbers, protocols aur doosre criteria.

Iske alawa, firewall ke liye kuch doosre solutions bhi available hain, jaise `nftables`, `ufw` aur **`firewalld`**.

`Nftables`, **`iptables`** ke comparison mein zyada modern syntax aur better performance provide karta hai. Lekin **`nftables`** ke rules ka syntax **`iptables`** ke saath compatible nahi hota. Isliye **`iptables`** se **`nftables`** par migrate karne ke liye kuch extra effort lagta hai.

`UFW` ka full form Uncomplicated Firewall hai. Ye firewall rules configure karne ke liye ek simple aur user-friendly interface provide karta hai.

UFW bhi **`nftables`** ki tarah **`iptables`** framework ke upar built hai aur firewall rules manage karne ka ek easier way provide karta hai.

Finally, **`FirewallD`** ek dynamic aur flexible firewall solution provide karta hai, jiska use complex firewall configurations manage karne ke liye kiya ja sakta hai.

Ye network traffic filter karne ke liye bahut saare rules support karta hai aur custom firewall zones aur services create karne ke liye bhi use kiya ja sakta hai.

Isme kai components hote hain, jo ek saath kaam karke ek flexible aur powerful firewall solution provide karte hain.

Iptables ke main components hain:

Component	Description
`Tables`	Tables ka use firewall rules ko organize aur categorize karne ke liye kiya jata hai.
`Chains`	Chains ka use firewall rules ke ek group ko banane ke liye kiya jata hai, jo ek specific type ke network traffic par apply hote hain.
`Rules`	Rules network traffic ko filter karne ke criteria define karte hain aur un packets ke liye action decide karte hain jo un criteria se match hote hain.
`Matches`	Matches ka use network traffic filter karne ke liye specific criteria match karne mein hota hai. Jaise source ya destination IP addresses, ports, protocols aur doosre criteria.
`Targets`	Targets un packets ke liye action specify karte hain jo kisi specific rule se match hote hain. Jaise packets ko accept, drop ya reject karna, ya packets mein kisi doosre tarike se modification karna.

Tables

Jab hum Linux systems mein firewalls ke saath kaam karte hain, toh ye samajhna important hai ki `iptables` mein tables kaise kaam karti hain.

`iptables` mein Tables ka use firewall rules ko traffic ke type ke basis par categorize aur organize karne ke liye kiya jata hai, jiske liye un rules ko design kiya gaya hai.

In tables ka use firewall rules ko organize aur categorize karne ke liye hota hai.

Har table ek specific set of tasks perform karne ke liye responsible hoti hai.

Table Name	Description	Built-in Chains
`filter`	IP addresses, ports aur protocols ke basis par network traffic ko filter karne ke liye use hoti hai.	INPUT, OUTPUT, FORWARD
`nat`	Network packets ke source ya destination IP addresses ko modify (change) karne ke liye use hoti hai.	PREROUTING, POSTROUTING
`mangle`	Network packets ke header fields ko modify (change) karne ke liye use hoti hai.	PREROUTING, OUTPUT, INPUT, FORWARD, POSTROUTING

Built-in tables ke alawa, `iptables` ek fourth table bhi provide karta hai, jise raw table kaha jata hai. Iska use special packet processing options configure karne ke liye kiya jata hai.

`raw` table mein do built-in chains hoti hain:

- `PREROUTING`
- `OUTPUT`

Chains

`iptables` mein Chains, rules ko organize karti hain. Ye rules define karte hain ki network traffic ko kaise filter ya modify kiya jana chahiye.

`iptables` mein Chains do types ki hoti hain:

- Built-in chains
- User-defined chains

Built-in Chains

Built-in chains pehle se defined hoti hain aur jab koi table create hoti hai, toh ye automatically create ho jaati hain.

Har table mein built-in chains ka alag set hota hai.

Example ke liye, `filter` table mein 3 built-in chains hoti hain:

- `INPUT`
- `OUTPUT`
- `FORWARD`

In chains ka use incoming aur outgoing network traffic ko filter karne ke liye kiya jata hai. Saath hi, different network interfaces ke beech forward hone wale traffic ko bhi filter kiya jata hai.

nat table mein 2 built-in chains hoti hain:

- `PREROUTING`
- `POSTROUTING`

`PREROUTING` chain ka use incoming packets ke destination IP address ko modify karne ke liye kiya jata hai, routing table dwara un packets ko process karne se pehle.

`POSTROUTING` chain ka use outgoing packets ke source IP address ko modify karne ke liye kiya jata hai, routing table dwara un packets ko process karne ke baad.

mangle table mein 5 built-in chains hoti hain:

- `PREROUTING`
- `OUTPUT`
- `INPUT`
- `FORWARD`
- `POSTROUTING`

In chains ka use incoming aur outgoing packets ke header fields ko modify karne ke liye kiya jata hai. Saath hi, un packets ke header fields ko bhi modify kiya jata hai jo corresponding chains ke through process ho rahe hote hain.

`User-defined chains` firewall rules ko manage karna easy bana sakti hain. Ye firewall rules ko specific criteria ke basis par group karti hain, jaise source IP address, destination port ya protocol.

Inhe kisi bhi three main tables mein add kiya ja sakta hai.

Example ke liye, agar kisi organization ke paas multiple web servers hain aur un sabhi ke liye similar firewall rules ki zaroorat hai, toh har server ke rules ko ek user-defined chain mein group kiya ja sakta hai.

Ek doosra example hai, jab user-defined chain ka use kisi specific port par jaane wale traffic ko filter karne ke liye kiya jata hai. Jaise port 80 (HTTP).

Iske baad user is chain mein aise rules add kar sakta hai, jo specifically port 80 par jaane wale traffic ko filter karte hain.

Rules and Targets

`iptables` rules ka use network traffic ko filter karne ke criteria define karne aur un packets ke liye action decide karne ke liye kiya jata hai, jo un criteria se match hote hain.

Rules ko chains mein add karne ke liye `-A` option ka use kiya jata hai, jiske baad chain ka naam diya jata hai. Rules ko different options ka use karke modify ya delete bhi kiya ja sakta hai.

Har rule mein kuch criteria ya matches hote hain aur ek target hota hai, jo un packets ke liye action specify karta hai jo diye gaye criteria se match hote hain.

Criteria ya matches, IP header ke specific fields ko match karte hain. Jaise source ya destination IP address, protocol, source port number, destination port number aur doosre fields.

Target ye specify karta hai ki jo packets kisi specific rule ke criteria se match hote hain, unke saath kya action lena hai.

Example ke liye, targets packets ko accept, drop, reject ya modify kar sakte hain.

`iptables` rules mein use hone wale kuch common targets mein ye shamil hain:

Target Name	Description
ACCEPT	Packet ko firewall se pass hone aur apni destination tak pahunchne ki permission deta hai.
DROP	Packet ko drop kar deta hai, jisse woh firewall se pass nahi ho pata.
REJECT	Packet ko drop kar deta hai aur source address par error message bhejta hai, jisse source ko pata chalta hai ki packet block kar diya gaya hai.
LOG	Packet ki information ko system log mein record karta hai.
SNAT	Packet ke source IP address ko modify karta hai. Generally NAT mein private IP address ko public IP address mein translate karne ke liye use hota hai.
DNAT	Packet ke destination IP address ko modify karta hai. Generally NAT mein traffic ko ek IP address se doosre IP address par forward karne ke liye use hota hai.
MASQUERADE	Ye `SNAT` ke similar hai, lekin iska use tab hota hai jab source IP address fixed nahi hota, jaise dynamic IP address ke case mein.

`REDIRECT`	Packets ko kisi doosre port ya IP address par redirect karta hai.
`MARK`	Packet ki Netfilter mark value ko add ya modify karta hai, jiska use advanced routing ya doosre purposes ke liye kiya ja sakta hai.

Chaliye ek rule ka example samajhte hain. Maan lijiye ki hum `INPUT` chain mein ek nayi entry add karna chahte hain, jo port 22 (SSH) par aane wale incoming TCP traffic ko accept karne ki permission de.

Iske liye command kuch is tarah dikhegi:

```
Toxicx01@htb[/htb]$ sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

Match Name	Description
`-p` or `--protocol`	Match karne ke liye protocol specify karta hai. Jaise TCP, UDP, ICMP.
`--dport`	Match karne ke liye destination port specify karta hai.
`--sport`	Match karne ke liye source port specify karta hai.

`-s` or `--source`	Match karne ke liye source IP address specify karta hai.
`-d` or `--destination`	Match karne ke liye destination IP address specify karta hai.
`-m state`	Connection ki state ko match karta hai. Jaise NEW, ESTABLISHED, RELATED.
`-m multiport`	Multiple ports ya port ranges ko match karta hai.
`-m tcp`	TCP packets ko match karta hai aur ismein TCP-specific additional options bhi include hote hain.
`-m udp`	UDP packets ko match karta hai aur ismein UDP-specific additional options bhi include hote hain.
`-m string`	Un packets ko match karta hai jinmein koi specific string hoti hai.
`-m limit`	Ek specified rate limit ke according packets ko match karta hai.
`-m conntrack`	Connection tracking information ke basis par packets ko match karta hai.
`-m mark`	Packets ko unki Netfilter mark value ke basis par match karta hai.
`-m mac`	Packets ko unke MAC address ke basis par match karta hai.

`-m iprange`	IP addresses ki ek range ke basis par packets ko match karta hai.

Generally, **`iptables`** mein matches ko **`-m`** option ka use karke specify kiya jata hai.

Example ke liye, neeche di gayi command **`filter`** table ki **`INPUT`** chain mein ek rule add karti hai, jo port 80 par aane wale incoming TCP traffic ko match karta hai:

```shellsession
Toxicx01@htb[/htb]$ sudo iptables -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
```

Ye example rule port 80 (**`--dport 80`**) par aane wale incoming TCP traffic (**`-p tcp`**) ko match karta hai.

Agar traffic diye gaye criteria se successfully match ho jata hai, toh ye `ACCEPT` target (**`-j ACCEPT`**) par jump karta hai, jiska matlab hai ki packet ko accept kar liya jata hai.

No.	Description
1.	Apne target par TCP/8080 port par ek web server start karein aur `iptables` ka use karke us port par aane wale incoming traffic ko block karein.
2.	TCP/8080 port par incoming traffic ko allow karne ke liye `iptables` rules change karein.
3.	Kisi specific IP address se aane wale traffic ko block karein.

4. Kisi specific IP address se aane wale traffic ko allow karein.

5. Protocol ke basis par traffic ko block karein.

6. Protocol ke basis par traffic ko allow karein.

7. Ek nayi chain create karein.

8. Traffic ko kisi specific chain par forward karein.

9. Kisi specific rule ko delete karein.

10. Saare existing rules ki list dekhein.

System Logs

Linux mein System Logs files ka ek collection hoti hain, jinmein system aur usmein hone wali activities ki information hoti hai.

Ye logs system ko monitor karne aur problems ko solve karne (troubleshooting) mein important hoti hain. Inse humein system ke behaviour, applications ki activities aur security events ke baare mein jaankari mil sakti hai.

Ye System Logs Linux system ke andar possible security weaknesses aur vulnerabilities ko identify karne ke liye bhi information ka ek important source hain.

Apne target system ke logs ko analyze karke hum system ke behaviour, network activity aur user activity ke baare mein jaankari hasil kar sakte hain. Is information ka use karke hum kisi bhi abnormal activity ko identify kar sakte hain, jaise:

- Unauthorized logins – Bina permission ke login karna.
- Attempted attacks – Attack karne ki koshish karna.
- Clear text credentials – Bina encryption ke credentials hona.
- Unusual file access – Files ko unusual tarike se access karna.

Ye activities kisi possible security breach (system ki security tootne) ka sanket ho sakti hain.

Hum penetration testers bhi apni security testing activities ki effectiveness ko monitor karne ke liye System Logs ka use kar sakte hain.

Security testing perform karne ke baad logs ko check karke hum pata laga sakte hain ki hamari activities ki wajah se koi security event trigger hua hai ya nahi, jaise Intrusion Detection Alerts (attack detect hone par alert) ya System Warnings (system ki taraf se warnings).

Ye information hamari testing strategies ko improve karne aur overall system ki security ko behtar banane mein help kar sakti hai.

Linux system ki suraksha banaye rakhne ke liye System Logs ko sahi tarike se set karna zaroori hai.

Ismein sahi Log Levels set karna, Log Rotation configure karna shamil hai, taaki log files bahut zyada badi na ho jaayein. Saath hi, logs ko surakshit jagah par store karna aur bina permission ke access se bachana bhi zaroori hai.

Iske alawa, possible security risks ka pata lagane aur kisi bhi security event par samay par action lene ke liye logs ko niyamit roop se check aur analyze karna chahiye.

Linux mein kai tarah ke System Logs hote hain, jinmein shamil hain:

- Kernel Logs
- System Logs
- Authentication Logs
- Application Logs
- Security Logs

Kernel Logs

In Logs mein system ke Kernel se related information hoti hai, jismein Hardware Drivers, System Calls aur Kernel Events shamil hote hain.

Ye Logs `/var/log/kern.log` file mein store hoti hain.

Example ke liye, Kernel Logs humein vulnerable ya outdated drivers ke baare mein bata sakti hain, jinhe attackers target karke system mein access hasil kar sakte hain.

Ye Logs humein System Crashes, Resource Limitations aur doosri events ke baare mein bhi information deti hain, jo Denial of Service (DoS) ya doosri security problems ka reason ban sakti hain.

Iske alawa, Kernel Logs ki help se hum suspicious System Calls aur doosri activities ko identify kar sakte hain, jo system mein Malware ya kisi doosre malicious software ki presence ka indication ho sakti hain.

`/var/log/kern.log` file ko monitor karke hum kisi bhi unusual behaviour ka pata laga sakte hain aur system ko aage hone wale damage se bachane ke liye appropriate action le sakte hain.

System Logs

In Logs mein system-level events ki jaankari hoti hai, jaise services ka start aur stop hona, login karne ki koshish aur system ka dobara start (reboot) hona.

Ye Logs `/var/log/syslog` file mein store hoti hain.

Login attempts, services ke start aur stop hone aur doosri system-level activities ko analyze karke hum system par hone wale possible access aur activities ka pata laga sakte hain.

Isse hum un vulnerabilities ko identify kar sakte hain, jinhe attackers exploit kar sakte hain. Saath hi, hum in risks ko kam karne ke liye security measures suggest kar sakte hain.

Iske alawa, hum `syslog` ki help se aisi possible problems ka pata laga sakte hain, jo system ki availability ya performance ko affect kar sakti hain.

Jaise:

- Service ka start hone mein fail hona.
- System ka reboot hona.

Neeche ek example diya gaya hai ki aisi `syslog` file kaisi dikh sakti hai:

Syslog

```shellsession
Feb 28 2023 15:00:01 server CRON[2715]: (root) CMD (/usr/local/bin/backup.sh)
Feb 28 2023 15:04:22 server sshd[3010]: Failed password for htb-student from
10.14.15.2 port 50223 ssh2
Feb 28 2023 15:05:02 server kernel: [  138.303596] ata3.00: exception Emask 0x0
SAct 0x0 SErr 0x0 action 0x6 frozen
Feb 28 2023 15:06:43 server apache2[2904]: 127.0.0.1 - - [28/Feb/2023:15:06:43
+0000] "GET /index.html HTTP/1.1" 200 13484 "-" "Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.149
Safari/537.36"
Feb 28 2023 15:07:19 server sshd[3010]: Accepted password for htb-student from
10.14.15.2 port 50223 ssh2
Feb 28 2023 15:09:54 server kernel: [  367.543975] EXT4-fs (sda1): re-mounted.
Opts: errors=remount-ro
Feb 28 2023 15:12:07 server systemd[1]: Started Clean PHP session files.
```

Authentication Logs

In Logs mein user authentication attempts ki information hoti hai, jismein successful aur failed dono tarah ke login attempts shamil hote hain.

Ye Logs **`/var/log/auth.log`** file mein store hoti hain.

Dhyan dene wali baat ye hai ki **`/var/log/syslog`** file mein bhi login se related similar information ho sakti hai, lekin **`/var/log/auth.log`** file specifically user authentication attempts par focus karti hai.

Is wajah se ye file possible security threats ko identify karne ke liye zyada useful resource hai.

Isliye penetration testers ke liye zaroori hai ki woh **`/var/log/auth.log`** file mein stored logs ko check karein, taaki ensure kiya ja sake ki system secure hai aur kisi attacker ne system ko compromise nahi kiya hai.

Auth.log

```
Feb 28 2023 18:15:01 sshd[5678]: Accepted publickey for admin from 10.14.15.2 port
43210 ssh2: RSA SHA256:+KjEzN2cVhIW/5uJpVX9n5OB5zVJ92FtCZxVzzcKjw
Feb 28 2023 18:15:03 sudo:   admin : TTY=pts/1 ; PWD=/home/admin ; USER=root ;
COMMAND=/bin/bash
Feb 28 2023 18:15:05 sudo:   admin : TTY=pts/1 ; PWD=/home/admin ; USER=root ;
COMMAND=/usr/bin/apt-get install netcat-traditional
Feb 28 2023 18:15:08 sshd[5678]: Disconnected from 10.14.15.2 port 43210 [preauth]
Feb 28 2023 18:15:12 kernel: [  778.941871] firewall: unexpected traffic allowed on
port 22
Feb 28 2023 18:15:15 auditd[9876]: Audit daemon started successfully
Feb 28 2023 18:15:18 systemd-logind[1234]: New session 4321 of user admin.
Feb 28 2023 18:15:21 CRON[2345]: pam_unix(cron:session): session opened for user
root by (uid=0)
Feb 28 2023 18:15:24 CRON[2345]: pam_unix(cron:session): session closed for user
root
```

Is example mein hum dekh sakte hain ki first line mein user **`admin`** ke authentication ke liye successful Public Key ka use kiya gaya hai.

Iske alawa, hum dekh sakte hain ki ye user **`sudoers`** group mein hai, kyunki ye **`sudo`** ka use karke commands execute kar sakta hai.

Kernel message indicate karta hai ki Port 22 par unexpected traffic ko allow kiya gaya tha, jo kisi possible security breach ka indication ho sakta hai.

Iske baad, hum dekhte hain ki **`systemd-logind`** ne user **`admin`** ke liye ek naya session create kiya hai.

Aur **`cron`** ka ek session user **`root`** ke liye open aur close hua

Application Logs

In Logs mein system par run hone wali specific applications ki activities ki information hoti hai.

Ye aksar apni alag files mein store hoti hain, jaise Apache web server ke liye **`/var/log/apache2/error.log`** aur MySQL database server ke liye **`/var/log/mysql/error.log`**.

Jab hum specific applications, jaise Web Servers ya Databases ko target karte hain, tab ye Logs bahut important hoti hain, kyunki inse humein pata chalta hai ki applications data ko kaise process aur handle kar rahi hain.

In Logs ko examine karke hum possible vulnerabilities ya misconfigurations (galat configurations) identify kar sakte hain.

Example ke liye, Access Logs ka use Web Server par ki gayi requests ko track karne ke liye kiya ja sakta hai, jabki Audit Logs ka use system ya specific files mein kiye gaye changes ko track karne ke liye kiya ja sakta hai.

In Logs ka use unauthorized access attempts, data exfiltration (bina permission ke data ko system se bahar bhejna) ya doosri suspicious activities ko identify karne ke liye kiya ja sakta hai.

Iske alawa, Access Logs aur Audit Logs important logs hoti hain, jo system par users aur processes dwara kiye gaye actions ki information record karti hain.

Ye security aur compliance (rules aur standards ka palan) ke liye bahut zaroori hoti hain. Hum inka use possible security problems aur attack vectors (attack karne ke possible raaste) ko identify karne ke liye kar sakte hain.

Example ke liye, **`Access Logs`** system par users aur processes ki activities ka record rakhti hain, jismein login attempts, files ko access karna aur network connections shamil hote hain.

`Audit Logs` system par hone wale security-related events ki information record karti hain, jaise system configuration files mein changes karna ya system files aur settings ko modify karne ki koshish karna.

Ye Logs possible attacks aur activities ko track karne aur security breaches ya doosri problems ko identify karne mein help karti hain.

Access Log file mein ek example entry kuch is tarah dikh sakti hai:

Access Log Entry

```shellsession
2023-03-07T10:15:23+00:00 servername privileged.sh: htb-student accessed
/root/hidden/api-keys.txt
```

Is Log Entry mein hum dekh sakte hain ki user **`htb-student`** ne **`/root/hidden/`** directory mein maujood **`api-keys.txt`** file ko access karne ke liye **`privileged.sh`** script ka use kiya.

Linux systems mein, most common services ke Access Logs ke liye default locations hoti hain:

Service	Description
`Apache`	Access Logs `/var/log/apache2/access.log` file mein store hoti hain (ya distribution ke according kisi similar location par).
`Nginx`	Access Logs `/var/log/nginx/access.log` file mein store hoti hain (ya kisi similar location par).
`OpenSSH`	Ubuntu mein Access Logs `/var/log/auth.log` file mein aur CentOS/RHEL mein `/var/log/secure` file mein store hoti hain.
`MySQL`	Access Logs `/var/log/mysql/mysql.log` file mein store hoti hain.
`PostgreSQL`	Access Logs `/var/log/postgresql/postgresql-version-main.log` file mein store hoti hain.
`Systemd`	Access Logs `/var/log/journal/` directory mein store hoti hain.

Security Logs

Ye Security Logs aur inmein hone wale events alag-alag log files mein record kiye jaate hain. Ye depend karta hai ki kaunsi specific security application ya tool use kiya ja raha hai.

Example ke liye, Fail2ban application failed login attempts ko `/var/log/fail2ban.log` file mein record karti hai, jabki UFW Firewall apni activities ko `/var/log/ufw.log` file mein record karta hai.

Doosre security-related events, jaise system files ya settings mein changes, general system logs mein record ho sakte hain, jaise `/var/log/syslog` ya `/var/log/auth.log`.

Penetration testers ke roop mein, hum Log Analysis Tools aur Techniques ka use karke specific events ya activities ke patterns search kar sakte hain, jo kisi security issue ka indication de sakte hain.

Is information ka use karke hum system ko aur test kar sakte hain, taaki vulnerabilities ya possible attack vectors identify kiye ja sakein.

In sabhi Logs ko alag-alag tools ki help se access aur analyze kiya ja sakta hai.

In tools mein most Linux desktop environments ke andar built-in Log File Viewers aur command-line tools shamil hain, jaise **`tail`**, **`grep`** aur **`sed`** commands.

System Logs ka sahi tarike se analysis karne se humein system ki problems ko identify aur troubleshoot karne mein help milti hai. Iske saath hi, hum security breaches aur doosre important events ka bhi pata laga sakte hain.

Solaris

Solaris ek Unix-based Operating System hai, jise 1990s mein Sun Microsystems ne develop kiya tha (baad mein Oracle Corporation ne Sun Microsystems ko acquire kar liya).

Solaris apni robustness (majbooti), scalability (zaroorat ke according expand hone ki ability) aur high-end hardware aur software systems ke support ke liye jaana jaata hai.

Solaris ka use Enterprise Environments mein widely kiya jaata hai, khaaskar mission-critical applications ke liye, jaise Database Management, Cloud Computing aur Virtualization.

Example ke liye, ismein ek built-in hypervisor hota hai, jiska naam `Oracle VM Server for SPARC` hai. Ye ek single physical server par multiple Virtual Machines ko run karne ki permission deta hai.

Overall, Solaris ko large amount of data handle karne aur users ko reliable aur secure services provide karne ke liye design kiya gaya hai.

Iska use aksar Enterprise Environments mein hota hai, jahan security, performance aur stability important requirements hoti hain.

Solaris ka main goal Enterprise Computing ke liye ek highly stable, secure aur scalable platform provide karna hai.

Ismein High Availability, Fault Tolerance aur System Management ke liye built-in features hote hain, jo ise Mission-Critical Applications ke liye suitable banate hain.

Solaris ka use Banking, Finance aur Government sectors mein widely kiya jaata hai, jahan security, reliability aur performance bahut important hoti hain.

Iska use Large-Scale Data Centers, Cloud Computing Environments aur Virtualization Platforms mein bhi kiya jaata hai.

Amazon, IBM aur Dell jaisi companies apne products aur services mein Solaris ka use karti hain, jo industry mein iski importance ko highlight karta hai.

Linux Distributions vs Solaris

Solaris aur Linux Distributions do alag tarah ke Operating Systems hain, jinmein kaafi differences hote hain.

Sabse pehle, Solaris ek Proprietary Operating System hai, jiska ownership aur development Oracle Corporation ke paas hai. Iska Source Code general public ke liye available nahi hai.

Iske opposite, most Linux Distributions Open-Source hoti hain. Iska matlab hai ki inka Source Code kisi ke liye bhi available hota hai, jise koi bhi modify aur use kar sakta hai.

Iske alawa, Linux Distributions commonly Zettabyte File System (`ZFS`) ka use karti hain. Ye ek advanced File System hai, jo Data Compression, Snapshots aur High Scalability jaise features provide karta hai.

Dusri taraf, Solaris Service Management Facility (`SMF`) ka use karta hai. Ye ek advanced Service Management Framework hai, jo system services ke liye better Reliability aur Availability provide karta hai.

Directory	Description
`/`	Root directory mein file system ki baaki sabhi directories aur files hoti hain.
`/bin`	Ismein essential system binaries hoti hain, jo system ko boot karne aur basic operations perform karne ke liye zaroori hoti hain.

`/boot`	Is directory mein boot se related files hoti hain, jaise Boot Loader aur Kernel Images.
`/dev`	Ismein device files hoti hain, jo system se connected physical aur logical devices ko represent karti hain.
`/etc`	Ismein system configuration files hoti hain, jaise system startup scripts aur user authentication data.
`/home`	Is directory mein users ki home directories hoti hain.
`/kernel`	Is directory mein Kernel Modules aur Kernel se related doosri files hoti hain.
`/lib`	Is directory mein `/bin` aur `/sbin` directories ki binaries ke liye zaroori libraries hoti hain.
`/lost+found`	Is directory ka use File System Consistency Check aur Repair Tool dwara recover ki gayi files ko store karne ke liye kiya jaata hai.
`/mnt`	Is directory ka use File Systems ko temporarily mount karne ke liye kiya jaata hai.
`/opt`	Is directory mein system par install kiye gaye optional software packages hote hain.
`/proc`	Ye directory files ke roop mein system ke processes aur Kernel Status ki information provide karti hai.

`/sbin`	Is directory mein system administration tasks ke liye zaroori system binaries hoti hain.
`/tmp`	System aur applications dwara create ki gayi temporary files is directory mein store hoti hain.
`/usr`	Is directory mein system-wide read-only data aur programs hote hain, jaise documentation, libraries aur executable files.
`/var`	Is directory mein variable data files hoti hain, jaise system logs, mail spools aur printer spools.

Solaris mein kai unique features hain, jo ise doosre operating systems se alag banate hain.

Iski ek sabse badi khaasiyat hai ki yeh high-end hardware aur software systems ko support karta hai.

Ise bade data centers aur complex network infrastructures ke saath kaam karne ke liye design kiya gaya hai. Yeh bina kisi performance problem ke bahut saara data handle kar sakta hai.

Package management ki baat karein, toh Solaris `Image Packaging System (IPS)` package manager ka use karta hai, jo packages aur updates ko manage karne ka powerful aur flexible tareeka provide karta hai.

Solaris advanced security features bhi provide karta hai, jaise Role-Based Access Control (`RBAC`) aur Mandatory Access Controls. Ye features har Linux distribution mein available nahi hote.

Differences

Aaiye, Solaris aur Linux distributions ke beech ke differences ko detail mein samajhte hain.

Sabse important differences mein se ek yeh hai ki Solaris ka source code open source nahi hai aur ise sirf limited log hi dekh sakte hain.

Iska matlab hai ki Ubuntu ya kai doosre Linux distributions ke unlike, Solaris ke source code ko public log dekh aur analyze nahi kar sakte.

Short mein, inke main differences ko in categories mein divide kiya ja sakta hai:

- Filesystem
- Process Management
- Package Management
- Kernel aur Hardware Support
- System Monitoring
- Security

System Information

Ubuntu mein hum **`uname`** command ka use system ki information dekhne ke liye karte hain, jaise kernel ka naam, hostname aur operating system.

Yeh kuch is tarah dikh sakta hai:

```shellsession
Toxicx01@htb[/htb]$ uname -a

Linux ubuntu 5.4.0-1045 #48-Ubuntu SMP Fri Jan 15 10:47:29 UTC 2021 x86_64 x86_64
x86_64 GNU/Linux
```

Doosri taraf, Solaris mein system ki information dekhne ke liye **`showrev`** command ka use kiya ja sakta hai.

Ismein Solaris ka version, hardware ka type aur patch level ki information shamil hoti hai.

Yeh iska ek example output hai:

```shellsession
$ showrev -a

Hostname: solaris
Kernel architecture: sun4u
OS version: Solaris 10 8/07 s10s_u4wos_12b SPARC
Application architecture: sparc
Hardware provider: Sun_Microsystems
Domain: sun.com
Kernel version: SunOS 5.10 Generic_139555-08
```

Dono commands ke beech main difference yeh hai ki **`showrev`**, Solaris system ke baare mein zyada detailed information deta hai, jaise patch level aur hardware provider.

Jabki **uname** Linux system ke baare mein sirf basic information deta hai.

Installing Packages

Ubuntu mein packages install karne ke liye **apt-get** command ka use kiya jata hai.

Yeh kuch is tarah dikh sakta hai:

```
Toxicx01@htb[/htb]$ sudo apt-get install apache2
```

Lekin Solaris mein **SUNWapchr** jaise packages install karne ke liye humein **pkgadd** command ka use karna padta hai.

```
$ pkgadd -d SUNWapchr
```

Dono commands ke beech main difference unka syntax aur package manager hai.

Ubuntu packages ko manage karne ke liye Advanced Packaging Tool (APT) ka use karta hai, jabki Solaris Solaris Package Manager (SPM) ka use karta hai.

Yeh bhi dhyan dein ki is case mein hum **sudo** ka use nahi karte.

Aisa isliye hai kyunki Solaris **RBAC** privilege management tool ka use karta tha, jo users ko alag-alag aur limited permissions dene ki permission deta tha.

Lekin Solaris 11 se **sudo** ko bhi support kiya gaya hai.

Permission Management

Ubuntu jaise Linux systems aur Solaris dono mein files aur directories ki permissions change karne ke liye **chmod** command ka use kiya jata hai.

Yahan ek example command hai, jo file ke owner ko read, write aur execute karne ki permission deti hai:

```
Toxicx01@htb[/htb]$ chmod 700 filename
```

Ubuntu mein specific permissions wali files ko find karne ke liye hum **`find`** command ka use karte hain.

Aaiye, ek aisi file ka example dekhte hain jismein SUID bit set hai:

```
Toxicx01@htb[/htb]$ find / -perm 4000
```

Solaris mein bhi specific permissions wali files, jaise jinmein SUID bit set hai, unhe find karne ke liye hum **`find`** command ka use kar sakte hain, lekin ismein thoda sa change karna padta hai.

```
$ find / -perm -4000
```

In dono commands ke beech main difference Solaris command mein permission value se pehle **`-`** ka use karna hai.

Aisa isliye hai kyunki Solaris ka permission system Linux se alag hota hai.

NFS in Solaris

Solaris ka apna NFS implementation hai, jo Ubuntu jaise Linux distributions se thoda alag hai.

Solaris mein NFS server ko configure karne ke liye **`share`** command ka use kiya ja sakta hai. Is command ka use network par kisi directory ko share karne ke liye hota hai.

Ismein hum alag-alag options bhi specify kar sakte hain, jaise read/write permissions, access restrictions aur bhi bahut kuch.

Solaris mein NFS ke through kisi directory ko share karne ke liye hum neeche di gayi command ka use kar sakte hain:

```
$ share -F nfs -o rw /export/home
```

Yeh command **`/export/home`** directory ko NFS ke through read aur write permissions ke saath share karti hai.

NFS client, Ubuntu ki tarah hi **`mount`** command ka use karke NFS file system ko mount kar sakta hai.

Solaris mein NFS file system ko mount karne ke liye humein server ka naam aur shared directory ka path specify karna hota hai.

Example ke liye, agar kisi server ka IP address `10.129.15.122` hai aur shared directory `/nfs_share` hai, toh NFS share ko mount karne ke liye hum neeche di gayi command ka use karte hain:

```
Toxicx01@htb[/htb]$ mount -F nfs 10.129.15.122:/nfs_share /mnt/local
```

Solaris mein NFS ki configuration `/etc/dfs/dfstab` file mein store hoti hai.

Is file mein har shared directory ki entry hoti hai, saath hi NFS sharing ke liye alag-alag options bhi diye hote hain.

```
# cat /etc/dfs/dfstab

share -F nfs -o rw /export/home
```

Process Mapping

Process mapping, system administration aur troubleshooting ka ek important part hai.

`lsof` command ek powerful tool hai, jo kisi process ke dwara open ki gayi sabhi files ki list dikhata hai. Ismein network sockets aur doosre file descriptors bhi shamil hote hain.

Hum Ubuntu jaise Debian distributions mein `lsof` command ka use kar sakte hain.

Kisi process ke dwara open ki gayi sabhi files ki list dekhne ke liye hum `lsof` ka use kar sakte hain.

Example ke liye, Apache web server process ke dwara open ki gayi sabhi files ki list dekhne ke liye hum neeche di gayi command ka use kar sakte hain:

```
Toxicx01@htb[/htb]$ sudo lsof -c apache2
```

Solaris mein kisi process ke dwara open ki gayi sabhi files ki list dekhne ke liye **`pfiles`** command ka use kiya ja sakta hai.

Example ke liye, Apache web server process ke dwara open ki gayi sabhi files ki list dekhne ke liye hum neeche di gayi command ka use kar sakte hain:

```
$ pfiles `pgrep httpd`
```

Yeh command Apache web server process ke dwara open ki gayi sabhi files ki list dikhati hai.

`pfiles` command ka output **`lsof`** command ke output jaisa hi hota hai.

Yeh file descriptor ka type, file descriptor number aur file name ke baare mein information provide karta hai.

Executable Access

Solaris mein **`truss`** ka use kiya jata hai. Yeh developers aur system administrators ke liye ek bahut useful tool hai, jinhe Solaris operating system par complex software problems ko debug karna hota hai.

Kisi process ke dwara kiye gaye system calls ko trace karke, **`truss`** errors, performance issues aur doosri problems ka source identify karne mein help karta hai.

Lekin application development ya system maintenance ke dauran yeh kuch sensitive information bhi dikha sakta hai.

Yeh tool system calls ke baare mein detailed information bhi provide karta hai, jismein unhe diye gaye arguments aur unke return values shamil hote hain.

Isse users apni applications aur underlying operating system ke behaviour ko achhe se samajh sakte hain.

`Strace`, **`truss`** ka alternative hai, lekin Ubuntu ke liye. Yeh system administrators aur developers dono ke liye ek important tool hai, jo unhe real-time mein problems ko diagnose aur troubleshoot karne mein help karta hai.

Yeh users ko operating system aur us par chalne wali applications ke beech hone wale interactions ko analyze karne ki permission deta hai.

Yeh especially un environments mein useful hai, jahan systems bahut complex aur mission-critical hote hain.

truss ki help se users application performance, network connectivity aur system resources ke use se judi problems ko jaldi identify aur alag kar sakte hain.

Example ke liye, Apache web server process ke dwara kiye gaye system calls ko trace karne ke liye hum neeche di gayi command ka use kar sakte hain:

```
Toxicx01@htb[/htb]$ sudo strace -p `pgrep apache2`
```

Yahan ek example diya gaya hai ki Solaris mein `ls` command ke dwara kiye gaye system calls ko trace karne ke liye `truss` ka use kaise karte hain:

```
$ truss ls

execve("/usr/bin/ls", 0xFFBFFDC4, 0xFFBFFDC8)  argc = 1
...SNIP...
```

Iska output **strace** jaisa hi hota hai, lekin iska format thoda alag hota hai.

strace aur **truss** ke beech ek difference yeh hai ki **truss**, kisi process ko bheje gaye signals ko bhi trace kar sakta hai, jabki **strace** aisa nahi kar sakta.

Ek aur difference yeh hai ki ***truss***, child processes ke dwara kiye gaye system calls ko bhi trace kar sakta hai.

Jabki **strace** sirf usi process ke system calls ko trace kar sakta hai, jise command line par specify kiya gaya ho.

Shortcuts

Linux mein bahut saare shortcuts hain, jinka use karke hum apna kaam aasaan aur fast kar sakte hain.

Jab hum important shortcuts ko achhe se samajh lete hain aur unhe apni aadat bana lete hain, toh humein bahut kam typing karni padti hai.

Inmein se kuch shortcuts humein terminal mein mouse ka use karne se bhi bachate hain.

Auto-Complete

[TAB] - Auto-complete shuru karta hai.

Yeh humare diye gaye **STDIN** ke aadhar par alag-alag options suggest karta hai.

Ye suggestions ho sakte hain, jaise:

- Humari current working directory mein maujood directories.
- Aise commands jo humare pehle type kiye gaye characters se shuru hote hain.
- Alag-alag options.

Cursor Movement

[CTRL] + A - Cursor ko current line ke shuru mein le jata hai.

[CTRL] + E - Cursor ko current line ke end mein le jata hai.

[CTRL] + [←] / [→] - Cursor ko current/previous word ke shuru mein le jata hai.

[ALT] + B / F - Cursor ko ek word peeche ya aage le jata hai.

Erase The Current Line

[CTRL] + U - Cursor ki current position se lekar line ke shuru tak ka sab kuch delete kar deta hai.

[CTRL] + K - Cursor ki current position se lekar line ke end tak ka sab kuch delete kar deta hai.

[CTRL] + W - Cursor ki position se pehle wale word ko delete kar deta hai.

Paste Erased Contents

[CTRL] + Y - Delete kiye gaye text ya word ko wapas paste karta hai.

Ends Task

[CTRL] + C - *SIGINT* signal bhejkar current task/process ko stop karta hai.

Example ke liye, agar kisi tool se scan chal raha hai aur hum us scan ko dekh rahe hain, toh is shortcut ka use karke hum us process ko stop ya kill kar sakte hain.

Yeh shortcut humare use kiye ja rahe tool ke configuration aur development par depend nahi karta.

Yeh process ko bina humse confirmation maange hi kill kar deta hai.

End-of-File (EOF)

[CTRL] + D - *STDIN* pipe ko close karta hai, jise End-of-File (EOF) ya End-of-Transmission bhi kaha jata hai.

Clear Terminal

[CTRL] + L - Terminal ko clear karta hai.

Is shortcut ka alternative **`clear`** command hai, jise type karke hum apne terminal ko clear kar sakte hain.

Search Through Command History

[CTRL] + R - Command history mein pehle type kiye gaye commands ko humare search pattern ke according search karta hai.

[↑] / **[↓]** - Command history mein previous/next command par jaane ke liye use hota hai.

Switch Between Applications

[ALT] + [TAB] - Open kiye gaye applications ke beech switch karta hai.

Zoom

[CTRL] + [+] - Zoom in karta hai (screen ko bada karta hai).

[CTRL] + [-] - Zoom out karta hai (screen ko chhota karta hai).