INTRODUCTION

The aim of this study is improve knowledge about Linux systems and improve Hacking capability.

DETAILS

Find

find -name password.txt : Bulunulan konumda adı password.txt olan dosyaların veya dosyanın konumunu getirir

find -name \*.txt : Bulunulan konumda sonu .txt ile biten tüm dosyaların konumlarını getirir

NOT : find komutu bulunulan konumda arama yapar. O yüzden find ile hangi konumda ve directory’nin altında arama yapılacağı bilinmediğinde cd .. denilerek en üst dizine çıkılmalı ve sonrasında arama yapılması daha faydalıdır.

Grep

Cat WebAppPentest.txt | grep XSS : WebAppPentest dosyasından XSS yazısının bulunduğu cümleleri getirir. CASE SENSITIVEDIR. Dikkat edilmeli

OPERATORS

& (Background Operator) : Eğer sistemde uzun sürecek bir command yazmışsak (Örneğin büyük bir dosyayı başka bir yere kopyalamak gibi) bu uzun bir vakit alacağından ve terminali kitleyeceğinden dolayı & operatörü ile yazdığımız kodun arka planda çalışmasını sağlayabiliriz. Böylece command arka planda çalışmaya devam ederken terminali kullanmaya devam edebiliyor oluruz.

sh myscript.sh > output.log & : myscript.sh scriptinin çıktısını output.log dosyasına arkaplanda yazdırır.

&& (Double Command Operator) : Command1 && Command2 şeklinde sistemde çalıştırılacak commandler listesi hazıralayabiliriz. Dikkatli olunması gereken konu “Command2” ancak “Command1” başarılı bir şekilde run edilebildiyse çalıştırılır.

> (Output Write Operator) : Sistemde output oluşturacak herhangi bir komut yazdığımızda o outputun kaydedileceği bir dosyayı belirtmek için kullanılır.

Echo hey > welcome.txt : Hey outputunu welcome.txt dosyasına kaydeder.

\*\*\*ÖNEMLİ NOT : Welcome.txt dosyası eğer sistemde bulunmuyor ise > operatörü welcome.txt şeklinde bir dosya oluşturur ve outputu onun içerisine kaydeder. Eğer sistemde zaten içerisinde datalar olan bir welcome.txt dosyası var ise TÜM İÇERİKLERİ SİLER ve dosyaya > operatöründen önceki command’in çıktısını yazdırır.

>> (Output Redirector Operator) : > operatörünün yaptığı iş ile aynı işi gerçekleştirir. Tek farkı “Echo hey >> welcome.txt” : welcome.txt dosyasının içeriklerini olduğu gibi bırakarak sonuna >> operatöründen önceki command’den gelen çıktıyı da ekler.

File System Interactions

Creating Files And Folders (touch, mkdir) : Touch komutu sistemde boş bir file oluşturmak için kullanılırken, mkdir komutu sistemde boş bir directory folder oluşturmak için kullanılır. Touch ile oluşturulan boş file’lara vi,nano gibi komutlar ile girdi ekleyebiliriz.

touch note : Note isminde boş bir file oluşturur.

mkdir notefolder : Notefolder isminde boş bir klasör oluşturur.

Removing Files And Folders (rm) : Bir dosyayı sistemden silmek için kullanılır

rm -rf notefolder : Notefolder dosyasını ve içerisinde bulunan tüm içerikleri siler.

Copying and Moving Files And Folders (cp, mv) : cp ve mv iki komut da iki parametre alır. Cp ilk parametrenin içeriklerini ikincisine kopyalarken Mv ilk file içeriğini ikinci file’a replace eder.

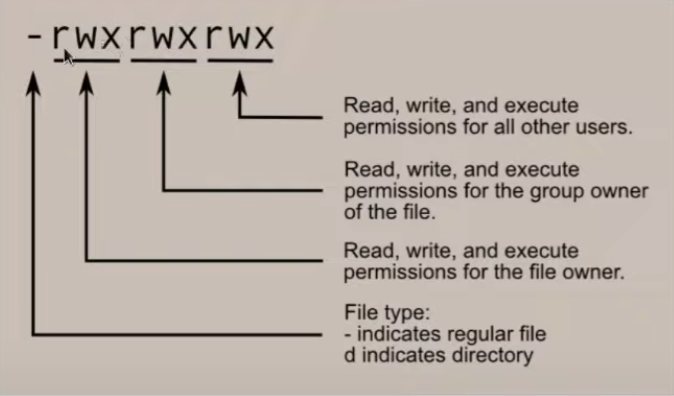
cp /root/tasinacakdosya/tasi.txt /var/tasinacakkonum/tasıburaya.txt : Şeklinde bir kullanımı vardır. Mv da aynı şekilde kullanılır.

Determining File Type (file) : Özellikle CTF’lerde bir dosyanın ne tür bir dosya olduğunu anlayabilmek için “file” commandini kullanırız. Bu command bize dosyanın türünü söyler.

file ngrok : ngrok dosyasının file type’ını output olarak getirir.

Output : ngrok: ELF 64-bit LSB executable

Permissions



1. 1.Kısım dosyanın bir Directory mi yoksa regular bir file mı olduğunu belirtiyor. (“d” ise directory “–“ ise regular bir file)
2. 2. Kısım dosyanın owner’ının izinlerini gösteriyor
3. 3. Kısım dosyanın owner’ının sistem içerisinde bulunduğu grubun dosya üzerindeki izinlerini gösteriyor.
4. 4. Kısım sistemdeki tüm diğer kullanıcıların bu dosya üzerindeki izinlerini gösteriyor.

To change directory permissions for everyone, use “u” for users, “g” for group, “o” for others, and “ugo” or “a” (for all).

chmod ugo+rwx foldername : This command gives read, write, execute permission to everyone

chmod o-rwx foldername : This command take away all the permission of read, write, execute from other users in the system.

chmod a+r foldername : This command gives only read permission for everyone.

Changing Ownership

chown user filelist(testfile e.g.) : Give ownership of a file to a specified user.

Changing Group Ownership

chgrp group filelist : Give ownership of a file to a specified group of users.

Permission Types

Read: This permission give you the authority to open and read a file. Read permission on a directory gives you the ability to lists its content.

Write: The write permission gives you the authority to modify the contents of a file. The write permission on a directory gives you the authority to add, remove and rename files stored in the directory. Consider a scenario where you have to write permission on file but do not have write permission on the directory where the file is stored. You will be able to modify the file contents. But you will not be able to rename, move or remove the file from the directory.

Execute: In Windows, an executable program usually has an extension “.exe” and which you can easily run. In Unix/Linux, you cannot run a program unless the execute permission is set. If the execute permission is not set, you might still be able to see/modify the program code(provided read & write permissions are set), but not run it.

Directories

/etc : Sistemimizdeki en önemli root directory’si. OS tarafından kullanılan system file’larının saklandığı konumdur. passwd, shadow gibi dosyaların da saklandığı konum olarak geçer. Shadow dosyasında sistem kullanıcılarının hashlenmiş şekilde şifreleri yer alır.

/var : Bu directory, sistemde sık sık erişilen veya sistemdeki servisler ve uygulamalar tarafından oluşturulan dosyaları barındırır. Sistemdeki LOG file’ları ya da specific bir user’a ait olmayan dosyalar (DATABASELER) genellikle bu klasör altında olur.

/tmp : Bu bölme sistemde sadece bir veya iki kez erişilecek dosyaları tutar. Makine restart edildiği anda bu klasördeki tüm dosyalar silinir, temizlenir.

\*\*\*ÖNEMLİ NOT : PENTEST işlemlerinde aklımızda olmalı ki, /tmp dosyası default olarak tüm user’lara Write permissionu sağlar. Yani sistemdeki her kullanıcı default olarak bu klasöre bir dosya ekleyebilir. Bu da demektir ki Pentestlerde enumeration(listelendirme) scriptlerimiz gibi şeyleri saklamak için çok uygun bir konumdur.

\*\*\*Enumeration Scripts : Enumeration is defined as the process of extracting user names, machine names, network resources, shares and services from a system. In this phase, the attacker creates an active connection to the system and performs directed queries to gain more information about the target. The gathered information is used to identify the vulnerabilities or weak points in system security and tries to exploit in the System gaining phase.

Types of information enumerated by intruders:

* Network Resource and shares
* Users and Groups
* Routing tables
* Auditing and Service settings
* Machine names
* Applications and banners
* SNMP and DNS details

General Utilities

Download Files

wget <https://assets.tryhackme.com/additional/myfile.txt> : Downloads myfile.txt from tryhackme website.

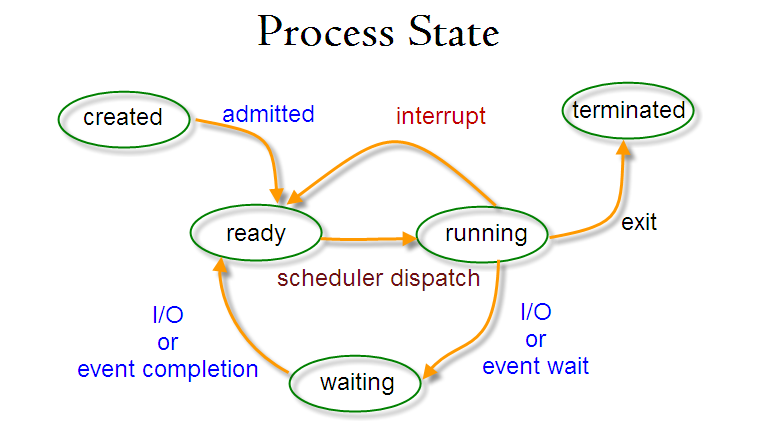
Transferring Files From Host – SCP (SSH)

This command allow us to copy file contents securely via SSH protocol to a remote computer.

scp textfrommycomputer.txt [ubuntu@192.168.1.30:/home/ubuntu/transferred.txt](mailto:ubuntu@192.168.1.30:/home/ubuntu/transferred.txt) : This commands copy the content of the textfrommycomputer.txt to a file which is stored in a remote machine via SSH.

scp [ubuntu@192.168.1.30:/home/ubuntu/documents.txt notes.txt](mailto:ubuntu@192.168.1.30:/home/ubuntu/documents.txt%20notes.txt) : Copy the content of the documents.txt which stored in remote device, to the notes.txt which stored in our device.

\*\*\*Processes



Process Listing

ps : O anda sistemde userların çalıştırdığı processleri gösterir

ps aux : Sistemde sadece user tarafından değil OS tarafından da çalıştırılmış olan tüm processleri getirir.

top : O anda sistemde çalışan tüm processleri ekrana real time statistic olarak verir. Navigate işlemi arrow keyler ile yapılır. (ps den farkı : ps anlık processleri bir ss alarak bize getirir top ise real time statistic şeklinde getirir)

top -u username : Displayss specific user’s process

A picture containing text

Description automatically generated

Gösterge tablosundaki ilk sayı satırı, süreyi, bilgisayarınızın ne kadar süredir çalıştığını, oturum açan kişi sayısını ve son bir, beş ve 15 dakika için yük ortalamasının ne olduğunu içerir. İkinci satır, görevlerin sayısını ve durumlarını gösterir: running, stopped, sleeping, veya zombie. Processdeki sütun başlıkları:

* **PID:** Process ID.
* **USER:** The owner of the process.
* **PR:** Process priority (process’s actual priority which is used by the Linux kernel to schedule a task) **Priority\_value = Nice\_value + 20**
* **NI:** The nice value of the process(Nice values are user-space values that we can use to control the priority of a process)
* **VIRT:** Amount of virtual memory used by the process.
* **RES:** Amount of resident memory used by the process.
* **SHR:** Amount of shared memory used by the process.
* **S:** Status of the process. (Runnable (R), Interruptible sleep (S), Uninterruptible sleep (D), Stopped (T), Zombie (Z), IDLE (I)) (Zombie Process : If a parent process isn’t programmed properly and never calls wait(), its zombie children will stick around in memory until they’re cleaned up.) (Uninterruptible sleep : This is a state that a process can enter when doing certain system calls. The process cannot be interrupted (or killed) until the system call completes.)
* For BSD formats and when the stat keyword is used, additional characters may be displayed:

**<**  high-priority (not nice to other users)   
**N** low-priority (nice to other users)   
**L** has pages locked into memory (for real-time and custom IO)  
**s**  is a session leader   
**l**  is multi-threaded (using CLONE\_THREAD, like NPTL pthreads do)   
**+**  is in the foreground process group

* **%CPU:** The share of CPU time used by the process since the last update.
* **%MEM:** The share of physical memory used.
* **TIME+:** Total CPU time used by the task in hundredths of a second.
* **COMMAND:** The command name or command line (name + options)

Managing Processes

kill -SIGKILL 1337 : 1337 Pid’li process’i sona erdirir.

kill -SIGTERM 1337 : 1337 Pid’li process’i son bir cleanup işlemlerini yapmasına izin vererek sona erdirir.

kill -SIGSTOP 32560 : Bir processi öldürmek yerine processi durdurur

kill -SIGCONT 32560 : Durmuş olan bir processi yeniden başlatır

kill -all ssh : Tüm ssh process’lerini öldürür(Aynı türdeki tüm işlemleri silmek istiyorsak, kill-all kullanabiliriz)

Getting Processes/Services to Start on Boot (systemctl)

Bir processin startup processler arasına alınması için yazılan komuttur

systemctl start apache2 : Bu komut ile Apache2 server start olacaktır.

systemctl stop apache2 : Bu komut ile Apache2 server durdurulacaktır

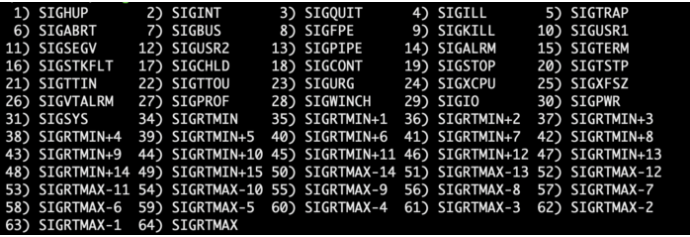
systemctl enable apache2 : Bu komut ile Apache2 server artık system start olurken çalıştırılacak komutlardan birisi olarak kabul edilir.

systemctl disable apache2 : Belirtilen process artık system startup’ta çalıştırılacak bir process olmayacaktır.

jobs -l : O anda askıya alınmış veya arka planda çalışan tüm processleri görüntüler.

fg %1(durdurulmuş process sırası): Background edilmiş bir process’i tekrar terminal ekranında çalışmaya başlatır (Yani processi foreground eder)

List Of All Available Signals On Lınux



**All of these signals will be used with kill command**

SIGHUP (1): If a process is running from a terminal and that terminal is closed accidentally, then the process receives this signal. “HUP” is short for “hang up” and terminates the process. If you don’t want to hang up the process when the terminal is disconnected, start the process with nohup.

SIGINT (2): Manual interruption of a process from a keyboard. When you hit **Ctrl-C**, it sends a SIGINT signal to the running process and terminates it.

SIGQUIT(3): When a user types <ctrl>+\ in the terminal. It will force the process to produce a **core dump**(A core dump is a file containing a process's address space (memory) when the process terminates unexpectedly.) and will terminate the process.

SIGKILL(9): Forcefully terminate a process. This signal can’t be blocked or handled. The receiving process cannot perform clean up when it is terminated with SIGKILL.

SIGPIPE (13): Broken pipe. If a process is piping in its output to another process (producer | consumer) and if the consumer dies, the producer gets the SIGPIPE signal from the consumer, which terminates the producer process

SIGTERM (15): It is the software termination signal. This is the signal that is sent by the kill command by default.

SIGCHLD(17): This signal is sent by a child process to its parent to intimate that the child is stopped or terminated. If a child process terminates before the parent calls the wait syscall on it, the kernel tries to keep the information about the process so that the parent can later call wait to get to know the exit status. This state of the child process, where the process terminates but the entry for the child still exists on the process table of the kernel, is referred to as zombie state and the process is called the zombie process.

SIGUSR1, SIGUSR2: Signals reserved for developers. These signals can have a custom disposition as configured by the developer.

Trapping Signals

This is used to do something when the process receives a specific signal is the trap command. The trap command can be used in shell scripts for following reasons;

* Cleaning up files
* Ignoring Signals

**trap** **command signals** : This is the way that how we use trap

trap "rm -f ${BASH\_SOURCE%/\*}/tmp\_data; exit" 2 : This command removes the tmp\_data file that gets created in the source directory of the script. The exit at the end is important for the process to terminate when it encounters signal 2, otherwise, the process deletes the file tmp\_data and continues to execute from the point in the program where the SIGINT (2) was invoked.

Set-User Identification (SUID)

To give an explanatory example for this sitation, when you change your password with the passwd command, your new password is stored in the file /etc/shadow.   
As a regular user, you do not have read or write access to this file for security reasons, but when you change your password, you need to have the write permission to this file. This means that the passwd program has to give you additional permissions so that you can write to the file /etc/shadow.

A file with **SUID** always executes as the user who owns the file, regardless of the user passing the command. If the file owner doesn't have execute permissions, then use an uppercase **S** here.

root@kali:/usr/bin# ls -lah | grep passwd

-rwsr-xr-x 1 root root 71K Jan 31 04:33 passwd

chmod 4555 [path\_to\_file] : Sets SUID on a file.

### Example—Finding Files With setuid Permissions

find / -user root (sahibi root olan dosyalar) -perm -4000 (s bitinin olduğu dosyalar) -exec ls -ldb {} \; > /tmp/fileswithsuidbit : Suidbit’e sahip root kullanıcısının owner’ı olduğu dosyaları ls -ldb formatında listele.

Set-Group Identification (SGID)

This special permission has a couple of functions:

* If set on a file, it allows the file to be executed as the group that owns the file (similar to SUID)
* If set on a directory, any files created in the directory will have their group ownership set to that of the directory owner

NOTE : If the owning group does not have execute permissions, then an uppercase S is used.

chmod g+s [path\_to\_directory] : Sets SGID on a directory.

chmod 2555 [path\_to\_file] : Sets GUID on a file.

root@kali:~/Desktop ls -l

total 0

drwxrws---. 2 root root 69k Jun 20 11:31 my\_articles

Sticky Bit

This permission does not affect individual files. However, at the directory level, it restricts file deletion. Only the owner (and root) of a file can remove the file within that directory. A common example of this is the /tmp directory.

chmod +t [path\_to\_directory] : This command sets sticky bit permission to the specified directory

/tmp ls -lah

drwxrwxrwt 2 root root 4.0K Jun 20 02:55 .ICE-unix

Partition Management (fdisk)

fdisk is used for creating and manipulating disk partition table. It is used for the view, create, delete, change, resize, copy and move partitions on a hard drive

sudo fdisk -l : Views all disk partitions

sudo fdisk -l /dev/sda : Views partition on a specific disk

fdisk /dev/sda : Views all fdisk commands

Text

Description automatically generated

Create a Hard Disk Partition

fdisk /dev/sda -> n (to create new partition) -> p (for making a primary partition) -> e (for making an extended partition) -> w (to write the changes and reboot your system)

Delete a Hard Disk Partition

fdisk /dev/sda -> d -> Partition number we want to delete -> w (For saving)

View The Size Of Your Partition

fdisk -s /dev/sda : Shows the size of our partition

REFERENCES

<https://tryhackme.com/room/linuxfundamentalspart1>

<https://tryhackme.com/room/linuxfundamentalspart2>

<https://tryhackme.com/room/linuxfundamentalspart3>

<https://www.howtogeek.com/668986/how-to-use-the-linux-top-command-and-understand-its-output/>

<https://www.redhat.com/sysadmin/suid-sgid-sticky-bit>

<https://www.thegeekdiary.com/what-is-suid-sgid-and-sticky-bit/>

[https://www.tutorialspoint.com/unix/unix-file-permission.htm#](https://www.tutorialspoint.com/unix/unix-file-permission.htm)

<https://www.geeksforgeeks.org/fdisk-command-in-linux-with-examples/#:~:text=fdisk%20also%20known%20as%20format,using%20the%20dialog%2Ddriven%20interface>.

<https://www.bogotobogo.com/Linux/linux_process_and_signals.php>