**Linux Privilege Escalation**

It's rare when performing a real-world penetration test to be able to gain a foothold (initial access) that gives you direct administrative access. Privilege escalation is crucial because it lets you gain system administrator levels of access, which allows you to perform actions such as:

* Resetting passwords
* Bypassing access controls to compromise protected data
* Editing software configurations
* Enabling persistence
* Changing the privilege of existing (or new) users
* Execute any administrative command

1. **Enumeration**

* Hostname: return the hostname of the target machine
* Uname -a: Sẽ in thông tin hệ thống cung cấp cho chúng tôi chi tiết bổ sung về kernel được hệ thống sử dụng.
* /proc/version: file hệ thống, may give you information on the kernel version and additional data such as whether a compiler (e.g. GCC) is installed.
* /proc: thư mục lưu trữ các file hệ thống về các tiến trình của hệ thống
* /etc/issue: file chứa thông tin về hệ điều hành
* /etc: thưc mục chứa các file cấu hình
* Ps: câu lệnh liệt kê các tiến trình, bao gồm 3 câu lệnh thường dùng khác nhau:
* Ps -A
* Ps axjf
* Ps aux
* Env: câu lệnh mà show các biến môi trường
* Sudo -l: hiển thị danh sách các lệnh hoặc quyền mà người dùng được phép thực hiện thông qua sudo.
* Ls: liệt kê
* Ls: liệt kê
* Ls -l: liệt kê chi tiết
* Ls -la: liệt kê chi tiết kể cả những file ẩn
* Ls -a: liệt kê kể cả những file ẩn
* Id: command will provide a general overview of the user’s privilege level and group memberships.
* /etc/passwd: file chứa các thông tin về user trên hệ thống
* /etc/shadow: file chứa thông tin về mật khẩu của người dùng hệ thống.
* History: liệt kê các câu lệnh đã sử dụng trong hệ thống
* Ifconfig: give us information about the network interfaces of the system
* Ip route:  see which network routes exist
* Python: kiểm tra phiên bản python
* **Netstat: thu thập thông tin về các kết nối hiện có.**
* NetStat -a: Hiển thị tất cả các cổng nghe và kết nối được thiết lập.
* Netstat -at hoặc Netstat -au cũng có thể được sử dụng để liệt kê các giao thức TCP hoặc UDP tương ứng.
* NetStat -L: Liệt kê các cổng trong chế độ nghe. Các cổng này được mở và sẵn sàng chấp nhận các kết nối đến. Điều này có thể được sử dụng với tùy chọn Tiên để chỉ liệt kê các cổng đang nghe bằng giao thức TCP (bên dưới)

A black background with white text

Description automatically generated

* netstat -s: liệt kê số liệu thống kê sử dụng mạng theo giao thức (bên dưới) Điều này cũng có thể được sử dụng với các tùy chọn -t hoặc -u để giới hạn đầu ra ở một giao thức cụ thể.

A screenshot of a computer

Description automatically generated

* netstat -tp: liệt kê các kết nối với tên dịch vụ và thông tin PID.

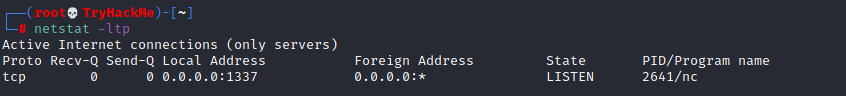
This can also be used with the -l option to list listening ports (below)

A screen shot of a computer

Description automatically generated

We can see the “PID/Program name” column is empty as this process is owned by another user.

Below is the same command run with root privileges and reveals this information as 2641/nc (netcat)



* The netstat usage you will probably see most often in blog posts, write-ups, and courses is netstat -ano which could be broken down as follows;
* **-a**: Display all sockets
* **-n**: Do not resolve names
* **-o**: Display timers
* **Lệnh find:**
* **Find files:**
* **find . -name flag1.txt**: find the file named “flag1.txt” in the current directory
* **find /home -name flag1.txt**: find the file names “flag1.txt” in the /home directory
* **find / -type d -name config**: find the directory named config under “/”
* **find / -type f -perm 0777**: find files with the 777 permissions (files readable, writable, and executable by all users)
* **find / -perm a=x**: find executable files
* **find /home -user frank**: find all files for user “frank” under “/home”
* **find / -mtime 10**: find files that were modified in the last 10 days
* **find / -atime 10**: find files that were accessed in the last 10 day
* **find / -cmin -60**: find files changed within the last hour (60 minutes)
* **find / -amin -60**: find files accesses within the last hour (60 minutes)
* **find / -size 50M**: find files with a 50 MB size

This command can also be used with (+) and (-) signs to specify a file that is larger or smaller than the given size.

A screen shot of a computer program

Description automatically generated

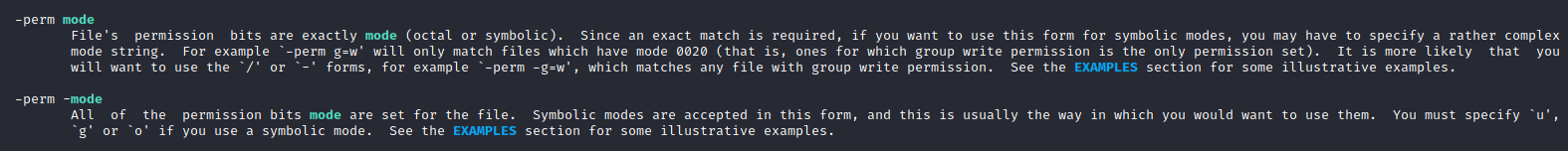
The example above returns files that are larger than 100 MB. It is important to note that the “find” command tends to generate errors which sometimes makes the output hard to read. This is why it would be wise to use the “find” command with “-type f 2>/dev/null” to redirect errors to “/dev/null” and have a cleaner output (below).

A screen shot of a computer code

Description automatically generated

* **Folders and files that can be written to or executed from:**
* **find / -writable -type d 2>/dev/null** : Find world-writeable folders
* **find / -perm -222 -type d 2>/dev/null**: Find world-writeable folders
* **find / -perm -o w -type d 2>/dev/null**: Find world-writeable folders

The reason we see three different “find” commands that could potentially lead to the same result can be seen in the manual document. As you can see below, the perm parameter affects the way “find” works.



* **Find development tools and supported languages:**
* **find / -name perl\***
* **find / -name python\***
* **find / -name gcc\***
* **Find specific file permissions:**

Below is a short example used to find files that have the SUID bit set. The SUID bit allows the file to run with the privilege level of the account that owns it, rather than the account which runs it. This allows for an interesting privilege escalation path,we will see in more details on task 6. The example below is given to complete the subject on the “find” command.

* **find / -perm -u=s -type f 2>/dev/null**: Find files with the SUID bit, which allows us to run the file with a higher privilege level than the current user.

General Linux Commands

As we are in the Linux realm, familiarity with Linux commands, in general, will be very useful. Please spend some time getting comfortable with commands such as **find**, **locate**, **grep**, **cut**, **sort**, etc.

1. Automated Enumeration Tools

Một số công cụ có thể giúp bạn tiết kiệm thời gian trong quá trình liệt kê. Những công cụ này chỉ nên được sử dụng để tiết kiệm thời gian khi biết rằng chúng có thể bỏ lỡ một số vectơ leo thang đặc quyền. Dưới đây là danh sách các công cụ liệt kê Linux phổ biến có liên kết đến kho lưu trữ Github tương ứng.

Môi trường của hệ thống mục tiêu sẽ ảnh hưởng đến công cụ bạn có thể sử dụng. Ví dụ, bạn sẽ không thể chạy một công cụ được viết bằng Python nếu nó không được cài đặt trên hệ thống mục tiêu. Đây là lý do tại sao tốt hơn là nên quen thuộc với một vài công cụ hơn là chỉ có một công cụ duy nhất.

* **LinPeas** : <https://github.com/carlospolop/privilege-escalation-awesome-scripts-suite/tree/master/linPEAS>
* **LinEnum:**<https://github.com/rebootuser/LinEnum>
* **LES ( Trình đề xuất khai thác Linux ):**<https://github.com/mzet-/linux-exploit-suggester>
* **Liệt kê thông minh Linux :**<https://github.com/diego-treitos/linux-smart-enumeration>
* **Trình kiểm tra quyền riêng tư của Linux :**<https://github.com/linted/linuxprivchecker>

1. Privilege escalation: Kernel Exploits

Privilege escalation ideally leads to root privileges. This can sometimes be achieved simply by exploiting an existing vulnerability, or in some cases by accessing another user account that has more privileges, information, or access.

Unless a single vulnerability leads to a root shell, the privilege escalation process will rely on misconfigurations and lax permissions (quyền lỏng lẻo).

The kernel on Linux systems manages the communication between components such as the memory on the system and applications. This critical function requires the kernel to have specific privileges; thus, a successful exploit will potentially (có khả năng) lead to root privileges.

The Kernel exploit methodology is simple. Theo các bước dưới đây

* Identify the kernel version
* Search and find an exploit code for the kernel version of the target system
* Run the exploit

**Research sources:**

1. Based on your findings, you can use Google to search for an existing exploit code.
2. Sources such as <https://www.linuxkernelcves.com/cves> can also be useful.
3. Another alternative would be to use a script like LES (Linux Exploit Suggester) but remember that these tools can generate false positives – dương tính sai (report a kernel vulnerability that does not affect the target system) or false negatives – âm tính sai (not report any kernel vulnerabilities although the kernel is vulnerable).

**Hints/Notes:**

1. Being too specific about the kernel version when searching for exploits on Google, Exploit-db, or searchsploit
2. Be sure you understand how the exploit code works BEFORE you launch it. Some exploit codes can make changes on the operating system that would make them unsecured in further use or make irreversible changes to the system, creating problems later. Of course, these may not be great concerns within a lab or CTF environment, but these are absolute no-nos during a real penetration testing engagement.
3. Some exploits may require further interaction once they are run. Read all comments and instructions provided with the exploit code.
4. You can transfer the exploit code from your machine to the target system using the **SimpleHTTPServer** Python module and **wget** respectively.

**X. Privilege Escalation: PATH**

**PATH là gì?**

PATH là một **biến môi trường** trong hệ điều hành (Linux, macOS, Windows), chứa danh sách các thư mục mà hệ thống sẽ tìm kiếm khi thực hiện các lệnh trong dòng lệnh (terminal hoặc command prompt).

**Tác dụng của PATH?**

Tác dụng chính của biến PATH là giúp hệ thống biết nơi tìm kiếm các chương trình thực thi khi bạn nhập một lệnh. Thay vì phải chỉ rõ đường dẫn đầy đủ của một chương trình, bạn chỉ cần gõ tên chương trình, và hệ thống sẽ tự động kiểm tra các thư mục được liệt kê trong biến PATH.

Ví dụ:

Chẳng hạn bạn có PATH như sau:

/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin

Khi bạn nhập lệnh python trong terminal, hệ thống sẽ tìm file thực thi python trong các thư mục được liệt kê trong PATH. Nếu tìm thấy, nó sẽ chạy chương trình đó. Cụ thể:

* Đầu tiên, nó kiểm tra thư mục /usr/local/bin xem có file thực thi nào tên là python không.
* Nếu không tìm thấy, nó tiếp tục kiểm tra thư mục /usr/bin, rồi /bin, và cứ thế cho đến khi duyệt hết tất cả các thư mục trong PATH.