HackSheet MASTER

Author: BERKE1337
Web: https://github.com/berke1337/hacksheet

Terminology

Each command contains a list of flags that indicate the OS requirement: Linux (L), BSD (B), FreeBSD (F), Mac OS (M), UNIX (U), and Windows (W).

Reconnaissance

Scanning

⋆ Ping sweep of subnet and host range

```
U # nmap -sP 10.0.0.0/24 192.168.0.128-254
```

⋆ List all computers in network

```
W # net view
```

⋆ Scan specific TCP and UDP ports

```
U # nmap -pT:21-25,80,U:5000-6000 target
```

⋆ TCP SYN scan without connecting

```
U # nmap -P0 -sS target
```

⋆ Detect OS

```
U # nmap -O target
U # p0f -s trace.pcap
```

⋆ Grab application banners

```
U # nmap -sV target
U # echo QUIT | nc target 1-1024
```

Wireless

Vulnerability Scanning

Web

⋆ Look for web server vulnerabilities

```
U # nikto -host 10.0.0.1
```

Hardening

Physical

⋆ Check devices

- Hardware keylogger (e.g., USB dongles)
- Rogue WiFi cards

OS & Software

⋆ Check for suspicious package repositories

```
L # vi /etc/apt/sources.list (Ubuntu)
L # vi /etc/yum.repos.d/* (RHEL/Fedora)
```

⋆ Run package updates

```
L # yum upgrade package
L # apt-get upgrade package
```

⋆ Update Kernel

```
L # yum update kernel (RHEL/Fedora)
L # apt-cache search linux-image; apt-get
  install linux-image-x.x.x-xx (Debian)
```

⋆ Harden SSHD

```
U fail2ban
U # vi /etc/ssh/sshd_config
  Protocol 2
  AllowUsers root admin webmaster
  AllowGroup sshusers
  PasswordAuthentication no
  HostbasedAuthentication no
  RSAAuthentication yes
  PubkeyAuthentication yes
  PermitEmptyPasswords no
  PermitRootLogin no
  ServerKeyBits 2048
  IgnoreRhosts yes
  RhostsAuthentication no
  RhostsRSAAuthentication no
```

User Management

⋆ Inspect logged in and past users

```
U # w
U # last | head
U # ps -ef | awk '$6 != "?"' (interactive procs)
W PsLoggedOn
W Task Manager → Users Tab
W # wmic computersystem get username
W # wmic /node:remotecomputer computersystem
  get username
```

⋆ Show account security settings

```
U # passwd -l user
L # chage -l user
W # net accounts
W # net accounts /domain
```

⋆ View Users

```
W # wmic useraccount list brief
```

⋆ Look for users with root privileges

```
U # awk -F: '$3 == 0 {print $1}' /etc/passwd
W # net localgroup administratos
```

⋆ Look for users with empty passwords

```
U # awk -F: '$2 == "" {print $1}' /etc/shadow
```

⋆ Make passwords expire

```
W # wmic path Win32_UserAccount Set
  PasswordExpires=True
W # wmic path Win32_UserAccount where
  name="username" Set PasswordExpires=True
W # wmic path /Node:remotecomputer
  Win32_UserAccount where name="username"
  Set PasswordExpires=True
L # chage -d 0 username
```

⋆ Set maximum number of login failures

```
L # faillog -M maxNumber -u username
L # faillog -r -u username
W # net accounts /lockoutthreshold:maxNumber
W # net accounts /lockoutduration:numberOfMinutes
```

⋆ Verify group memberships

```
U # vi /etc/group (admin, sudo, wheel)
```

⋆ Check sudo users

```
U # visudo
```

⋆ Check crontab users

```
U # for u in $(cut -f1 -d:  /etc/passwd); do
  crontab -u $u -l; done
```

⋆ Check remote authentication

```
U # vi ~/.rhosts
U # vi ~/.ssh/*
```

⋆ Change passwords

`U # pwgen -sy` (generate strong passwords)

`U # passwd` *`user`*

`W # net user` *`user`* `*`

File System

⋆ Secure mount points

```
U # mount -o nodev,noexec,nosuid /dev..  /tmp
```

⋆ List file attributes

```
L # lsattr /var/log/foo
B # ls -ol /var/log/foo
W # cacls.exe file.txt
```

⋆ File creation date

```
W # dir /tc /od
U # ls -li /etc | sort -n
```

⋆ System file checker

```
W # sfc /scannow
```

⋆ File signature serification

`W # sigverif`

`W` SIGCHECK

`W # sigcheck -e -u -s c:\`

⋆ Make files append-only

```
L # chattr +a /var/log/foo
```

Network

⋆ Show firewall rules

```
L # for t in nat mangle filter raw; do
  iptables -t $t -nL; done
W # netsh firewall show portopening
W # netsh firewall show allowedprogram
W # netsh firewall show config
```

⋆ Close ports

```
W # netsh advfirewall firewall add rule
  name="BlockAIM"
  protocol=TCP
  dir=out remoteport=4099 action=block
```

⋆ Shut down SMB vulnerable services

`W` Seconfig XP ☑Disable NetBIOS over TCP/IP (all interfaces) ☑Disable SMB over TCP/IP ☑Disable RPC over TCP/IP → Apply → Yes

⋆ Check DNS resolver

```
U # vi /etc/resolv.conf
```

⋆ Disable IPv6

`L # ipv6.disable=1` (add to kernel line)

`L # vi /etc/sysctl.conf`
`net.ipv6.conf.all.disable_ipv6 = 1`
`net.ipv6.conf.<interface0>.disable_ipv6 = 1`
`net.ipv6.conf.<interfaceN>.disable_ipv6 = 1`
`vi /etc/hosts` (comment IPv6 hosts)

```
L # vi /etc/sysconfig/network
  NETWORKING_IPV6=no
  IPV6INIT=no
  service network restart
```

`L # vi /etc/modprobe.conf`
`install ipv6 /bin/true` (append to file)

`L # vi /etc/modprobe.conf` (RHEL/CentOS)
`alias net-pf-10 off`

`L # vi /etc/modprobe.conf` (Debian/Ubuntu)
`alias net-pf-10 off`
`alias ipv6 off`

```
W # reg add hklm\system\currentcontrolset\services\
  tcpip6\parameters /v DisabledComponents /t
  REG_DWORD /d 255
```

⋆ Check network configuration

`L # vi /etc/network/interfaces` (Ubuntu)

`L # vi /etc/sysconfig/network-scripts/ifcfg-eth*` (RHEL)

Forensics

Processes

⋆ Inspect startup items

`L # initctl show-config` (upstart, Ubuntu)

`F # less /etc/rc.local` (deprecated)

`F # grep local_start /etc/defaults/rc.conf`

`W` Autoruns → Options → Filter Options ☑Verify code signatures ☑Hide Microsoft entries

⋆ Find SETUID and SETGID files and types

```
U # find / \( -perm -4000 -o -perm -2000 \)
  -exec file \{\} \;
U # crontab -e
  0 4 * * * find / \( -perm -4000 -o -perm
  -2000 \) -type f > /var/log/sidlog.new &&
  diff /var/log/sidlog.new /var/log/sidlog &&
  mv /var/log/sidlog.new /var/log/sidlog
```

⋆ Find world/group writeable directories

```
U # find / \( -perm -g+w -o -perm -o+w \)
  -type d -exec ls -ald \{\} \;
```

⋆ Find all unsigned processes

`W` ProcessExplorer Options → Verify Image Signatures

⋆ View Process File Location

`W` ProcessExplorer View → Select Columns... → Image Path

⋆ Currently Running Tasks/Processes

```
 W # tasklist -svc
LU # ps aux | less
LU # top
LU # ps -u user
```

⋆ Kill Tasks/Processes

```
 W # taskkill -pid pid
LU # kill pid
```

Network

⋆ Display listening TCP/UDP ports

```
LU # netstat -plunt
 W # netstat -abon | select-string -Context 1,
     0 LISTENING(PowerShell Only)
 W # netstat -aon | findstr LISTENING(cmd.exe)
 W TCPVIEW
 B # netstat -p tcp -an | egrep
     'Proto|LISTEN|udp'
 U # lsof -nPi | awk '/LISTEN/'
 F # sockstat -4 -l
```

⋆ Check active connections to find backdoors

```
 L # netstat -punt
 U # lsof -nPi | awk '/ESTABLISHED/'
```

Cleanup

⋆ Kill all processes accessing a mount point

```
 U # fuser -k -c /mnt/secret
```

Miscellaneous

Date and Time

⋆ Set date and time

```
 U # date MMddhhmm[[cc]yy]
 W # date
 W # time
```

Network

⋆ Forward a TCP/UDP port

```
 U # mkfifo f ;
     nc -l 80 < f | nc 127.0.0.1 6666 > f &
 L # iptables -t nat -A OUTPUT|POSTROUTING \
     -p tcp -s x.x.x.x -sport 80 -j SNAT \
     -to-destination 6666
 L # iptables -t nat -A INPUT|PREROUTING \
     -p tcp -d x.x.x.x -dport 80 -j DNAT \
     -to-destination :6666
```

Databases

⋆ Export/Restore

```
 mysql # mysqldump -u username -p database_name >
         dump.sql
 mysql # mysql -u username -p database_name <
         dump.sql
  psql # pg_dump database_name > dump.sql
  psql # psql -d database_name -f dump.sql
```

⋆ Change user password

```
 mysql # SET PASSWORD FOR 'root' =
         PASSWORD('new-pass'); FLUSH PRIVILEGES;
  psql # ALTER USER root WITH PASSWORD 'new-pass';
sqlcmd # ALTER LOGIN user WITH PASSWORD = 'pass';
         GO;
```

⋆ Add/Delete user

```
 mysql # CREATE USER 'user'@'localhost' IDENTIFIED
         BY 'pass';
 mysql # DROP USER user;
  psql # CREATE USER user-name WITH PASSWORD
         'pass' VALID UNTIL 'Jan 1 2014';
  psql # DROP USER user-name;
```

⋆ Permissions

```
 mysql # GRANT ALL ON db1.* TO 'foo'@'localhost';
         FLUSH PRIVILEGES;
 mysql # GRANT SELECT ON db2.invoice TO
         'bar'@'localhost'; FLUSH PRIVILEGES;
 mysql # REVOKE ALL ON *.* TO 'bar'@'localhost';
         FLUSH PRIVILEGES;
  psql # GRANT ALL PRIVILEGES ON *.* TO user;
  psql # REVOKE ALL PRIVILEGES ON *.* FROM user;
sqlcmd # GRANT ALL PRIVILEGES ON *.* TO
         windows-db-user [WITH GRANT OPTION]; GO;
sqlcmd # GRANT SELECT ON *.* TO user; GO;
sqlcmd # USE db-name; REVOKE ALL PRIVILGES FROM
         user; GO;
sqlcmd # USE db-name; REVOKE [GRANT OPTION FOR]
         ALTER FROM user; GO;
```

Windows Tasks

⋆ Download file from Internet

```
 W #  Powershell
   $source = "http:www.download.com/file.txt"
   $destination = "c:\temp\file.txt"
   $wc = New-Object System.Net.WebClient
   $wc.DownloadFile($source, $destination)
```

⋆ List device drivers and their properties

```
 W # driverquery (-v)
```

OpenSSL Certificate Manipulation

⋆ Generate a new private key and Certificate Signing Request

```
 L # openssl req -out CSR.csr -new -newkey
     rsa:2048 -nodes -keyout privateKey.key
```

⋆ Generate a self-signed certificate

```
 L # openssl req -x509 -nodes -days 365
     -newkey rsa:2048 -keyout privateKey.key
     -out certificate.crt
```

⋆ Generate a certificate signing request (CSR) for an existing private key

```
 L # openssl req -out CSR.csr -key
     privateKey.key -new
```

⋆ Generate a certificate signing request based on an existing certificate

```
 L # openssl x509 -x509toreq -in
     certificate.crt -out CSR.csr -signkey
     privateKey.key
```

⋆ Remove a passphrase from a private key

```
 L # openssl rsa -in privateKey.pem -out
     newPrivateKey.pem
```

⋆ Check a Certificate Signing Request (CSR)

```
 L # openssl req -text -noout -verify -in
     CSR.csr
```

⋆ Check a private key

```
 L # openssl rsa -in privateKey.key -check
```

⋆ Check a certificate

```
 L # openssl x509 -in certificate.crt -text
     -noout
```

⋆ Check a PKCS#12 file (.pfx or .p12)

```
 L # openssl pkcs12 -info -in keyStore.p12
```

References

- http://bit.ly/cmd-line-kung-fu
- http://bit.ly/useful-windows-one-liners
- http://bit.ly/vmware-esxi-reference
- http://bit.ly/ssl-commands

Tool Downloads

- Sys Internals: http://bit.ly/sys-internals
- Seconfig XP: http://seconfig.sytes.net/