|  |  |  |
| --- | --- | --- |
| Suggested action | Commands (via CLI) | Learning point |
| Open WireShark on a linked in Pi (lecturers) | - | Traffic monitoring can be used to highlight / show different actions and protocols – Especially if actions are carried out under instruction |
| Find IP and MAC address | ifconfig | Device network information |
| Scan network to identify connected devices | nmap -sT 192.168.99.0/24 | Show how we can scan a network – Can also point out that only devices on that network are being shown. |
| Ping the target device on the network (lecturer pi) | ping <ip.address> | Showcase ping – Use wireshark to show ICMP protocol |
| Block a specific IP address | iptables -A INPUT -s <ip.address> -j DROP | Showcase use of iptables / simulation of firewalls (sort of) |
| Re-run the previous ping | ping <ip.address> | We should now notice that one of the pi’s is unable to ping the target – This is due to the iptables rule |
| Remove current iptables rule and add in a new one | iptables INPUT -D 1 iptables -A INPUT -p ICMP -j DROP | We delete the previous rule and now we are blocking all ICMP traffic |
| Re-run the previous ping | ping <ip.address> | This time ALL pi’s should be unable to ping the target device |
| Connect to HTTP on the target pi (lecturer pi) | (via web browser) uri = <ip.address> | We should notice that the devices can all connect to the splash page – This highlights the different protocols used |
| Remove current iptables rule and add in a new one | iptables INPUT -D 1 iptables -A INPUT -p TCP -j DROP | We delete the previous rule and now we are blocking all TCP traffic – This will stop HTTP connection(s) |
| Connect to HTTP on the target pi (lecturer pi) | (via web browser) uri = <ip.address> | This time the devices should be unable to access the splash page / it should hang on a refresh |
| Re-run the previous ping | ping <ip.address> | This time all pi’s should be able to ping the target device – Again we are highlighting the difference in protocols |
| Scan the target device (lecturer pi) | nmap -sT <ip.address> | We should find that our scan results come back with limited to no info – This is because we are blocking TCP |
| Scan the target device (lecturer pi) using UDP | nmap -sU <ip.address> | We will not get much in the way of results from this either – But it will show UDP usage as well as highlight the different protocols |

Basics

Intermediate

|  |  |  |
| --- | --- | --- |
| Suggested action | Commands (via CLI) | Learning point |
| Open WireShark on a linked in Pi (lecturers) | - | Traffic monitoring can be used to highlight / show different actions and protocols – Especially if actions are carried out under instruction |
| Block a specific IP address | iptables INPUT -D 1 iptables -A INPUT -s <ip.address> -j DROP | Showcase use of iptables / simulation of firewalls (sort of) – Also make sure we delete any pre-existing rules, if for example we are following on from the “Basics” |
| Ping + browser connect to target device | ping ip.address (via web browser) uri = <ip.address> | All devices except the blocked device should be able to access the target device with no issues |
| Change IP address | (via GUI) – Screenshots to be added | This will show how a device can change it’s IP address and use that to evade / workaround things like a block.  NOTE: The Access point seems to become flaky when IPs are set manually – If a PI changes IPs and finds the network drops just reconnect (see screenshots) |
| Ping + browser connect to target device | ping ip.address (via web browser) uri = <ip.address> | All devices should now be able to access the target device with no issues |
| Check WireShark – Look at traffic for the blocked and then changed IP address | (WireShark filters) ip.src == <blocked.ip.address> ip.src == <changed.ip.address> | Make note of the MAC address – This should be the same in both occurrences. Discuss IP address v MAC (DHCP v Manual) |
| Delete iptables rule and revert to DHCP for all PIs | iptables INPUT -D 1 (via GUI) – Screenshots to be added | Point demonstrated we don’t need to risk the flake really |

Misconfigurations / probing

|  |  |  |
| --- | --- | --- |
| Suggested action | Commands (via CLI) | Learning point |
| Open WireShark on a linked in Pi (lecturers) | - | Traffic monitoring can be used to highlight / show different actions and protocols – Especially if actions are carried out under instruction |
| SSH onto target device (lecturer pi) | ssh kali@<ip.address> | Get them to think about / consider what the password is likely to be…  HINT: They’re all the same! |
| Create file on desktop | mkdir /home/kali/Desktop/<dir\_name> | If the lecturer PI is being shown on a larger screen visible proof of the action |
| Change the default port | sudo nano /etc/ssh/sshd\_config  change the Port from 22 to whatever (999) | Changes the port from a non-default  HIDE / DISCONNECT YOUR SCREEN! |
| Terminate SSH connections | systemctl disable sshd systemctl stop sshd  lsof -w -t +d /dev/pts/ | sort | uniq | \  xargs ps -oppid= -p | \  xargs ps -ocomm=,pid=,user= -p | \  awk '($1 == "sshd") {print $2}' | \  xargs kill | NEED TO TEST THIS  This will stop all active SSH connections |
| Restart SSH (with new port) | systemctl start sshd systemctl enable sshd | SSH will now be back up and running on a new port |
| SSH onto target device (lecturer pi) | ssh kali@<ip.address> | This should fail because SSH will assume that the default port is being used – They will need to find the new port |
| Scan target device (lecturer pi) | Nmap -sT ip.address   nmap -sT -p 0-1000 <ip.address>   nmap -sTV -p <target port> <ip.address> | Should now show that port 22 was previously open – Now missing / closed  They will need to re-run the scan and if the new port wasn’t shown then add in some port ranges to check for (provided by lecturer). For example if the port picked was 999  This will then show the new SSH port (but likely won’t have it listed as SSH). To have the correct version info run with the -sTV and target port |
| SSH onto target device (lecturer pi) | ssh kali@<ip.address> -p <target port> | Login as before |
| Terminate SSH connections | systemctl disable sshd systemctl stop sshd  lsof -w -t +d /dev/pts/ | sort | uniq | \  xargs ps -oppid= -p | \  xargs ps -ocomm=,pid=,user= -p | \  awk '($1 == "sshd") {print $2}' | \  xargs kill | NEED TO TEST THIS  This will stop all active SSH connections |
| Change kali user password | passwd | Where prompted enter the current password and then select a new one from:  /usr/share/wordlists/fasttrack.txt |
| Restart SSH (with new port + password) | systemctl start sshd systemctl enable sshd | SSH will now be back up and running on a new port and with the new user creds |
| SSH onto target device (lecturer pi) | ssh kali@<ip.address> -p <target port> | Login as before – This should now fail due to the cred change |
| Bruteforce the login using fasttrack | hydra -l kali -P /usr/share/wordlists/fasttrack.txt <ip.address> ssh -s <target port> -V | This will run Hydra against the target PI and will bruteforce the login – They should eventually get the correct username and password and can then… |
| SSH onto target device (lecturer pi) | ssh kali@<ip.address> -p <target port> | Login as before (using new creds) |
| Discuss the above | - | Show the hydra traffic using WireShark – Discuss the added difficulties that non-default ports and creds added. Discuss the importance / merit of non-default ports and strong creds (not taken from fasttrack) etc |