Exercise 4 – Nmap challenge

Stewart Johnston
CIS 127 – Intro to Information Security
NCMC
johnstons1@student.ncmich.edu

1 Capture 1

1. *Most likely scan*: Host discovery scan
2. *Exact syntax of the command*: `nmap -PE -PS443 -PA80 -PP 10.10.10.0/24`
3. *Result of scan*: Shows three other hosts active on that subnet, but with filtered ports.
4. *Machine vendors*: Cisco, AMD, Raspberry-Pi
5. *MAC address of the machine running nmap*: AmdPcnet_af:14:cb
6. *IPv4 address of machine running nmap*: 10.10.10.5
7. *IPv6 address of the machine running nmap*: fe80::c14:ca44:6c33:171
8. *MAC addresses of devices on network*:
   (a) 10.10.10.5 – AmdPcnet_af:14:cb
   (b) 10.10.10.1 – Cisco_2f:b1:dc
   (c) 10.10.10.4 – AmdPcnet_94:98:0b
   (d) 10.10.10.6 – Raspberr_1a:17:61

2 Capture 2 – OS scan

1. *Exact syntax of the command*: nmap -sV -O 10.10.10.6
2. *Why are there packets from hosts other than the scanner and scanee*? 10.10.10.4 is sending multicast UDP signals, about every second, which may be a keepalive signal to a phone.
3. *What services are responding*? 10.10.10.6 responds on SSH, for port 22
4. *What packet numbers are checking telnet*? Packet number 23 sends a SYN on port 23, and packet number 24.

3 Capture 3

1. *Type of scan*: Full-connect TCP scan along top 973 ports with some retries and no ping. What's notable about this is that it isn't a stealthy scan, because it completes the three-way-handshake before resetting, which is an event deemed worthy of logging by most systems. It's also the only kind of scan available if you don't have elevated permissions, because it uses the OS' underlying systemcalls to operate, and therefore behaves the legal way that OSs are expected to behave.

2. *Major clue to type of scan*: All rst packets sent by the attacker were first preceeded by a full three-way-handshake, of syn, syn/ack, ack. There are a number of retransmissions, about half a second after the prior attempt to send the packet. I can only assume that some attenuation occured on the line, or for some other reason the attacker was not receiving the responses before sending a retransmission. By a little scripting I've determined that the maximum number of retries seen in the packet capture file is 2, so I will also assume that is the case for the actual command being used. As for no ping, well, there are no ICMP from the attacker.

3. *Exact Syntax*: `nmap -sT -Pn --max-retries 2 --top-ports 973 10.10.10.4`

4. *What ports are listening on the victim*:

   (a) Port 139: netbios-ssn

   (b) Port 135: epmap

   (c) Port 25: SMTP

   (d) Port 443: HTTPS

   (e) Port 445: microsoft-ds

   (f) Port 1119: bnetgame

   (g) Port 99: metagram

   (h) Port 990: ftps

4 Capture 4

1. *Most likely type of scan*: No ping, along the top 945 ports, looks like a SYN scan, but no rst packets are sent resulting in several retransmissions from the victims.

2. *Major clue to type of scan*: The only packets sent by the attacker were SYN packets (excepting the ARP packets sent by the hardware addresss by default), which were at no point retransmitted. The only way I can see for such a result is with an iptable or other filter which prevents the host OS of the attacker from responding with rst packets.

3. *Exact syntax*: `nmap -sS -Pn --max-retries 0 --top-ports 945 10.10.10.1,4,6`

4. *What common service and port number is active on 2 of the devices, and which are the two*? HTTPS on port 443 is active/open at 10.10.10.1 and 10.10.10.4; SSH on port 22 is active/opent at 10.10.10.1 and 10.10.10.2

5 Capture 5 – version scan

1. *Exact syntax*: `nmap -sV -Pn 10.10.10.4`

2. *What services are running on the internet facing target*? It's pretty clear that the target is a frankenstein of a server, hosting web, files, mail, and potentially Directory Controller duties. Per wireshark:

    (a) Port 135: epmap (end-point mapper)

    (b) Port 25: smtp (mail transfer protocol)

    (c) Port 445: microsoft-ds (directory services)

    (d) Port 139: netbios-ssn (provides netbios api over TCP)

    (e) Port 443: https (hypertext – secure)

    (f) Port 990: ftps (file transfer – secure)

    (g) Port 1119: bnetgame (battlenet, used for Blizzard entertainment titles)

    (h) Port 99: metagram ?? Shockingly difficult to find what this really is.

3. *OS of the target*: Windows NT of some variety, I think.

4. *Protocol which shows the OS of the target*: microsoft-ds and netbios-ssn kind of give it away. epmap carried some packets which made clear that it was Microsoft and Samba. Similarly metagram outright stated that it was Microsoft-IIS/5.1 during an HTTP 400 err. SMTP carried a conversation to the effect that it was Microsoft ESMTP MAIL Service, and so on. I don't doubt that if you run them in an emulation layer, several of these things could be run on a *nix OS, but... why bother with that?

5. *Target hostname*: DHSCTE.local

6. *Protocol which shows the hostname of the target*: At least one call to Multicast-DNS shows the hostname, along with netbios, and some of the various other packets reply with the hostname.

7. *Hostname of the attacker*: CTE-tablet

8. *Service which shows the hostname of the attacker*: netbios-dgm, a datagram service running on UDP. This is apparently on behalf of the Browser service, which runs on MailSlot/SMB. Announcing MailSlot/SMB workgroup etc was a part of the packet containing the attacker's name.