NMAP Assignment

Jim Kinter

CIDM6340

 1. What did you do – I started the exercise by asking my manager for permission to execute the assignment. He declined to grant permission and referred me to the network security team. I spoke to the network security supervisor and he replied that, per company policy, both the installation of the NMAP software and the probing/scanning of the network were restricted and that failure to comply would result in termination. Subsequently, because I was traveling for work, I tried to run an NMAP scan on the hotel network where I was staying. The Marriott network configuration limited the results to the computer I was using and the wireless router I was connecting to. Upon returning home, the only other network that was available to scan was my home network. I attempted to run an Xmas scan using the -sX flag however I mistakenly added a /16 mask to the query. After 24 hours of running, I aborted the process. Next, I read the documentation and realized my mistake. I ascertained that the best initial query to run was “nmap -sV -T4 -O -F --version-light 192.168.1.0/24”. This quickly (--version-light & -T4 [timing=4] & -F [fast]) scans the 254 potential hosts on my network (/24 mask) and scans for the Operating System (-O). Finally, after reviewing the output of the light-scan, I ran an intense scan and changed the timing variable to 2 as “nmap -sV -T4 -O –top-ports 100 –version-intensity 4 192.168.1.0/24”.

2. What are the results – The initial scan identified 32 active devices on the network. Eleven of those were identified as having a Linux operating system. The Netgear Orbi Mesh network components and all of the Amazon Echo devices are counted among these. Thirteen other devices were identified as having a specialized operating system. Multiple Ring cameras, a Ring Doorbell, a Smart Lightbulb, and several Apple devices are counted among these. Seven devices were found but NMAP was unable to ascertain any information about the device. Finally one device was found with a Windows operating system. There were twelve services identified in total. Subsequently I performed an intense scan “nmap -T4 -A -v 192.168.1.XX” against the seven devices that did not provide enough information in the original scan to classify the device.

* Services
  + abyss – TCP Port 9999 - Proprietary Web Server
  + domain – TCP Port 53 – DNS Server
  + http – TCP Port 80 / TCP Port 8009 / TCP Port 8080 / TCP Port 5357 / TCP Port 631 / TCP Port 443 – Web Server
  + https – TCP Port 443 – Secure Web Server
  + jetdirect – TCP Port 9100 – Proprietary Printer Service
  + microsoft-ds – TCP Port 445 – Microsoft Directory Service
  + msrpc – TCP Port 135 – Microsoft Remote Procedure Call
  + netbios-ssn – TCP Port 139 – Microsoft Windows Netbios
  + rtsp – TCP Port 5000 – Apple AirTunes Server
  + tcpwrapped – TCP Port 8888 / TCP Port 49152 / TCP Port 8009 – wrapper protocol used by Amazon Echo devices and Apple iOS Devices
  + unknown – TCP Port 49154
  + upnp – TCP Port 5000 – UpnP [also matched for rtsp and sip]
* Hosts
  + 192.168.1.1 - Netgear Orbi router. I was not surprised to find TCP port 53 (domain), TCP port 80 (http), TCP port 443 (https), and TCP port 5000 (upnp) open because the router provides DNS service to the network and the router’s configuration is performed by the WebUI (http and https). The Orbi router is a mesh router that can connect to other mesh satellites through UPNP or through a physical ethernet cable.
  + 192.168.1.2 - Netgear Orbi satellite unit. It also has TCP port 53, TCP port 80, and TCP port 443 open.
  + 192.168.1.4 – Amazon Echo Dot– TCP Port 8888 (tcpwrapped) is open
  + 192.168.1.6 – Amazon Echo Show – TCP Port 8009 (tcpwrapped - Amazon Whisperplay DIAL REST Service) and TCP Port 8888 (tcpwrapped) are open
  + 192.168.1.7 – Amazon Echo Show – TCP Port 8009 (tcpwrapped - Amazon Whisperplay DIAL REST service)
  + 192.168.1.8 – Amazon Echo Dot – TCP Port 8888 (tcpwrapped) is open
  + 192.168.1.9 – Meross Smart Light Bulb – No open ports
  + 192.168.1.11 – GE Smart Washer – No open ports
  + 192.168.1.12 – Amazon Echo Dot – TCP Port 8888 (tcpwrapped)
  + 192.168.1.13 – Ring Doorbell – No open ports
  + 192.168.1.14 – Amazon SmartPlug – No open ports
  + 192.168.1.15 – Hubspace Smart Plug – No open ports
  + 192.168.1.16 – Apple iPhone 15 – TCP Port 49152 (unknown) and TCP Port 62078 (unknown)
  + 192.168.1.17 – Amazon Fire TV – TCP Port 8009 (tcpwrapped)
  + 192.168.1.18 – Wyze Camera – No open ports
  + 192.168.1.19 – HP OfficeJet Printer – TCP Port 80 (http) / TCP Port 443 (https) / TCP Port 631 (http) / TCP Port 8080 (http) / TCP Port 9100 (jetdirect)
  + 192.168.1.21 – TPLink Range Extender – TCP Port 9999 (abyss) is open
  + 192.168.1.22 – GE Smart Dryer – No open ports
  + 192.168.1.23 – Windows PC – TCP Port 80 (http)
  + 192.168.1.26 – Amazon Echo Show – TCP Port 1080 (socks5) / TCP Port 8009 (http) / TCP Port 8888 (tcpwrapped)
  + 192.168.1.27 – Apple iPhone 15 – TCP Port 49152 (unknown)
  + 192.168.1.29 – Amazon Echo Show - TCP Port 1080 (socks5) / TCP Port 8009 (http) / TCP Port 8888 (tcpwrapped)
  + 192.168.1.31 – Apple iPhone 14 – TCP Port 49152 (unknown) is open
  + 192.168.1.35 - AppleTV – TCP Port 7000 (rtsp) / TCP Port 8009 (tcpwrapped) / TCP Port 9080 (glrpc)
  + 192.168.1.37 – Amazon Echo Dot – TCP Port 8009 (http) is open
  + 192.168.1.38 – AppleTV – TCP Port 5000 (rtsp) / TCP Port 7000 (rtsp) / TCP Port 7100 (rtsp) / TCP Port 49152 (tcpwrapped) / TCP Port 49153 (unknown) / TCP Port 49154 (unknown) / TCP Port 62078 (tcpwrapped)
  + 192.168.1.40 – Samsung Smart TV – TCP Port 8080 (http)
  + 192.168.1.45 – Nintendo Switch – No open ports
  + 192.168.1.46 – Microsoft Windows PC (this is my work PC and I shut it off due to concerns previously mentioned)
  + 192.168.1.52 – Microsoft Windows PC – TCP Port 135 (msrpc) / TCP Port 139 (netbios-ssn) / TCP Port 445 (Microsoft-ds) / TCP Port 5357 (http)
  + 192.168.1.55 – Samsung SmartTV – No open ports
  + 192.168.1.66 – Apple iPhone – TCP Port 49152 (tcpwrapped)

3. What did you learn – I learned several key items from this exercise. First, I learned that NMAP is a great tool for interrogating a network and, because it allows the user to silently probe a network and identify attack surfaces for a specific device. By understanding what protocols/ports a particular device is actively listening/communicating on, an administrator can understand where they might be vulnerable, or a bad actor can focus their efforts on exploring how they can use that information to attack the device. I’ve learned that, as an administrator, I can’t depend on the device manufacturer to protect me and that I am responsible for making sure that all devices in my control are patched and secured, running the latest version of the operating system and configured in a way that disables any unnecessary services or interfaces. This is a time-consuming process that requires a high level of attention to detail. By extrapolating this experience to an organization as large and complex as ExxonMobil, I can appreciate their rigid stance and have a newfound appreciation for all the administrative overhead borne by the company to secure such a vast and complex environment, especially when you consider that the company is a prime target for any cyber-attacker, including state sponsored actors with virtually unlimited resources.

Some of the most valuable learnings from this exercise came from reading the documentation and understanding the significant level of configurability that a skilled cyber-practitioner can exert through command-line arguments. For a network or security administrator there is a certain level of network awareness already in hand and the tool can be used to specifically and explicitly address a specific potential vulnerability or a specific concern. For a black/white-hat attacker the tool’s ability to silently probe the network, including the ability to control the resulting DNS resolution, spoof a firewall or IDS, and control the sequence and intensity of probes in order to appear random, control the timing and level of parallelism to employ in an attack, and ultimately the ability to make a system’s interrogation seem random, disorganized, and without any logical pattern, makes for a very powerful toolkit indeed. One of my favorite configuration features is the ability to pipe the output of the query to an XML or other iterable format for consumption by another script.