2024

DATA SCIENCE NETWORK SCAN

Table Of Contents

Table Of Figures

Network Description

The Data Science Network (the network on which the scans were run) holds great importance within the COLSA domain, which in turn, means the security and assessment of this network is of a very high importance. This network is comprised of many machines and resources used to display proof-of-concepts for vulnerabilities, tool testing (Data packet tool sets, Metasploit, other applications), along with housing much of the Data Science testing as a whole. Data Science itself is essential to COLSA as a whole because it provides the ability to extract essential data, insights, patterns, and trends, which are needed to help with decision making and solving real-world problems which may pertain to the industry. Understanding the fundamental attributes of our network was the first step into structuring and maintaining our networking report.

Introduction

Cybersecurity is a rapidly growing problem for company networks, due to the large numbers of data breaches and identity thefts. One thing that companies can do to test their security is called a vulnerability assessment. These are used to assess any potential vulnerabilities and the frequency at which they might occur. In particular, network scanners are a popular tool used to carry out these assessments, which is covered in this report. Networking tools, such as Nmap, are able to identify vulnerable parts within these networks. Nmap can give insight on running services, ports open in each device, and active devices in the network which can allow for identifying solutions to security risks.

The objective of this scan is to provide system administrators and security experts a non-intrusive overview of COLSA's networks. This scan would include domains such as the lab.local, dacs.test, and corps.local subnets, which are then inspected using the Linux Nmap tool set. This report will cover vulnerabilities, hosts, and services found in these networks.

The rest were identified to provide a comprehensive report to enhance cyber security detection and response efforts within the Data Science domain.

Procedure

The Nmap (network mapping) tool set is a Linux CLI (command line interface) designed to scan IP addresses and ports within a network. Through the nature of Nmap, the command has the potential to intrude surrounding networks from utilizing pings, creating false positives for a network intrusion. In order to prevent these false positives around the surrounding networks in COLSA the Data Science network subnet was used to bound the network scan. Subnets define the beginning and end of an IP (10.0.10.0 - 10.0.10.255) which helps to represent the inner and outer bounds of a local network. From utilizing the local subnet and operating within its bounds, this ensured that the work done would not bypass preventative measures and cause suspicion from surrounding networks.

In terms of networking, ports are the virtual start and end points of network connections within local machines. Ports are a vital part of creating an effective and efficient networking solution. Some ports allow forwarding outbound connections between separate services, others manage the inbound connections within different parts of the network. While ports are an incredibly useful tool for attaching systems and networks together, they come with prevalent vulnerabilities that could compromise systems. By scanning for what ports are open and closed on a system, it allows for a coherent understanding of susceptible ports within a system.

The first scan command that was used for performing the vulnerability assessment used the most popular ports on Windows and Linux machines:

```
nmap -oX /home/kali/Desktop/Windows_scan.xml -vv -sV -p
23,80,443,445,139,3389,8000,8080,8443,20,21,25,53 --open -Pn -max-retries 3
10.0.10.0/24
```

Figure 1 First Nmap Scan

Common Ports Scan

Common Ports	*Services*
20	FTP (File Transfer Protocol) data transfer
21	FTP control
22	SSH (Secure Shell)
23	Telnet
25	SMTP (Simple Mail Transfer Protocol)
53	*DNS (Domain Name System)*
80, 8000, 8080	HTTP (Hypertext Transfer Protocol)
139	NetBIOS Session Service
443, 8443	HTTPS (Hypertext Transfer Protocol Secure)
445	SMB (Server Message Block)
3389	RDP (Remote Desktop Protocol)

Figure 2 Common Ports Scanned

As a secondary scan to capture any ports that may have been missed, this command was run:

nmap -oX /home/kali/Desktop/Windows_scan_two.xml -vv -sV –p1,10000 --open -Pn -max-retries 3 10.0.10.0/24

Figure 3 Broad Nmap Scan

Other open ports found on the network with the broader scan

Port	Services	Port	Services
81	Http OpenResty Web app server Nginx proxy login	*111*	Rpcbind
135	Msrpc	*464*	Kpasswd5 open on dns server
514	Tcp shell	*515*	Printer port open
593	Ncacn_http Microsoft Windows RPC over HTTP 1.0	*631*	Soap gSOAP 2.7
873	Rsync	*3261*	Iscsi Synology DSM Snapshot Replication iSCSI LUN
3268 3269	Active Directory for DNS server	*4045*	Nlockmgr 1-4 RPC #100021
500	Http nginx	*5002*	Http Docker Registry API 2.0
504	Blank but open	*5357*	Twsdapi
543	Postgresql	*9090*	Ubuntu login page .176

909	Xmltec-xmlmail	*9100*	Jetdirect open
920	SSL/RTSP	*9999*	Abyss Dozzle logger running wide open

Figure 4 Open ports on the network with respective services

Adjustments used for the Nmap scan

Attributes	Definitions
-oX	Takes output from the command and puts into an XML file that is defined as a file path after this attribute
-vv	Increases verbosity (Extra information displayed)
-sV	Checks ports for version numbers and services running on that port
-p	Defines ports to be scanned
--open	Checks only for open ports
--max-retries	Sets the max number of pings that the command can use before moving on to the next IP.
10.0.10.0/24	Defines the subnet for the Cyber Lab network

Figure 4 Nmap Adjustments

Procedure For Mounting NFS Shares

Network scans found eight (8) unique NFS shares hosted on three (3) NFS servers.

```
nmap 10.0.10.0/24 -p 2019 111 --open
```

Figure 5 NFS Nmap Command

Using Metasploit's *auxiliary/scanner/nfs/nfsmount* module we were able to enumerate the names of the shares as well as the access control list (ACL) protecting those shares. The screenshots below highlight the following vulnerabilities:

- ACLs allow all hosts on default network (10.0.10.0/24)
- ACLs allow hosts which no longer exist, this gives attackers the ability to simply change their IP address and/or host name to access the share

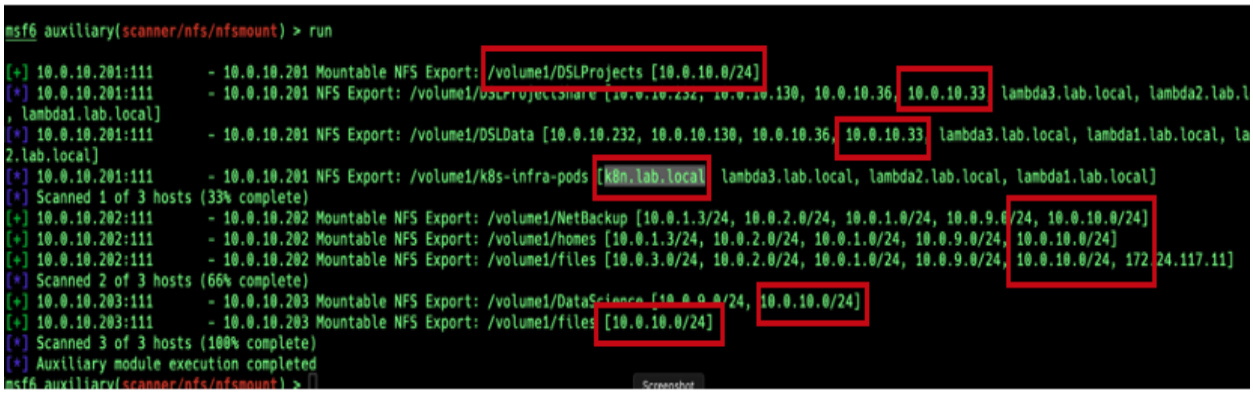

Figure 6 nfsmount scanner

The Following screenshot shows the method used to mount and access a share that was limited to systems named k8n.lab.local. The machine k8n.lab.local was no longer on the network which rendered the ACLs out of date and less effective. ACLs could be bypassed by changing the systems hostname to match requirements.

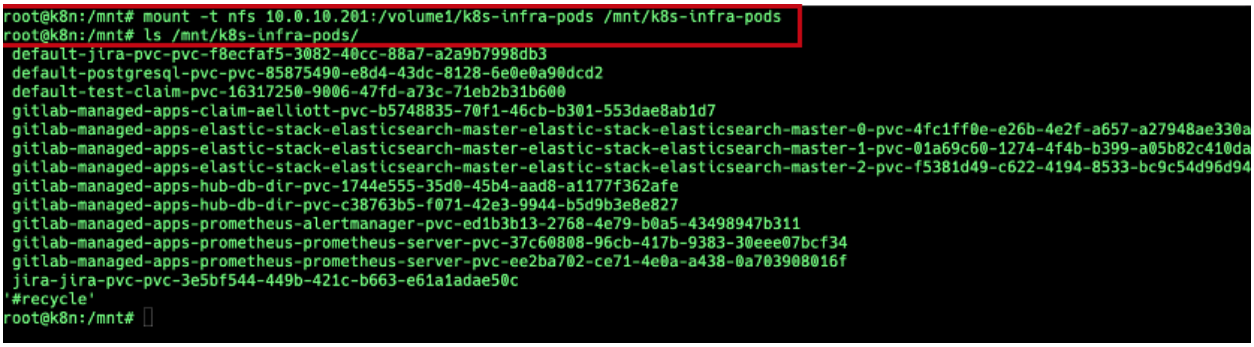

Figure 7 Mounting NFS shares

The team extensively searched through the NFS shares and was not able to find any data that could be used for further access or arbitrary code execution or sensitive in any way. If the data stored on these servers does not require further protections, then more strict ACLs may not be required. However, it is worth mentioning that even application data or development data left exposed can be vulnerable to ransom-ware attacks, accidental deletion or unauthorized modifications.

Vulnerabilities

Outdated software/Potential Exploits for services running on the network.

Outdated Software
OpenSSH 7.7
GSOAP 2.7
Samba 4.6.2

Figure 8 Outdated Software

Listed above are the software running on our network that is not updated to the newest release. Some reasons to keep software like this up to date, is to help prevent potential vulnerabilities, compatibility issues may arise with software that is not on the latest versions, and keeps the software performing at the highest level to ensure it is efficient as possible.

Webpages that are insecure

10.0.10.15:80	This Altona site is connected to an 8x8 HDMI Matrix Switcher. It is using default credentials set by the manufacture of the device.
10.0.10.63	This site on our network has an open page for a HP printer that is currently on the network.
10.0.10.9:9090	Prometheus Webpage that does not require a login to view and move throughout the webpage.
10.0.10.244:9999	Dozzle Logger (Loggin GitLab information from 10.0.10.244:80)

Figure 9 Insecure Webpages

Other Webpages on the network

10.0.10.12	CyberLab Updog, CyberLab CTF
10.0.10.22	Redirects to ZenWifi router login page
10.0.10.39	CyberLab CTF
10.0.10.101:80	Dc DNS info services page
10.0.10.199	ZenWIFI
10.0.10.201-203	Data Science NAS Login pages

10.0.10.253	DDwrt HomePage
10.0.10.49:9090	Ubuntu 22.04.4 Login page

Figure 10 More Insecure Webpages

Solutions and Preventative Measures

A preventative measure that should be taken for the webpages on this network include removing any pages that are no longer in use. This is especially important if the services running on these pages become outdated, hence prone to more exploits. Along with that, making sure to change default credentials on all logins, this is a step that might be overlooked when setting up a system; however, it can be very vulnerable to the network. Any sites on our network that do not require a login to see information being displayed is a heavily prominent security risk. Any web-based site that is not for public viewing should be locked with credentials that satisfy company policy. For example, the Prometheus instance that is running on 10.0.10.9:9090 has no authentication tied to it. However, there is a way to secure these instances, by creating a hashed password and inputting it into a .yml file on your Prometheus instance. More information about this certain case can be found at https://prometheus.io/docs/guides/basic-auth/. Another instance of a webpage with no login authorization is the Dozzle Logger found at IP 10.0.10.244:9999. This webpage can also be secured by configuring certain files on the system. For more information about this certain case visit https://dozzle.dev/guide/authentication

There are multiple solutions for cleaning up outdated software. One of which is keeping a close eye on new updates for all software. This is a key element on keeping a network safe. Setting a schedule to scan through and check for outdated software is a good start to automate this process. Along with updating old software, any software that is out of date and obsolete should be looked at for potential termination and/or swapped out for newer software.

When it comes to securing NFS shares we would recommend using DNS hostnames rather than IP addresses for more restrictive access/mount to NFS shares (based on the NFS

ACLs). Also look into different NFS versions that may offer more security features as well as their benefits. As well as keeping the ACLs up to date

CONCLUSION

Within the vulnerability scan there are a couple of big takeaways. From the utilization of network scanning, active common services running could be identified, many of which may contain vulnerabilities in addition to effective solutions towards providing network security on the CyberLab local network. The scan brought attention towards numerous problems around the network including, default credentials being used, possible remote access vulnerabilities, unused services, and source code access. Most problems found on the network can be fixed with a simple update of the software or change of credentials. With unused services checking whether this service is mandatory or not can allow for a process of removing the service. Collectively, we have determined that Cyberlab's network has many vulnerabilities each with individual capabilities. Our data science network is comprised of a myriad of systems and resources which all hold data from personal to corporate statuses. The management of this network is essential towards progressing effectiveness and efficiency, allowing all COLSA employees alike to withhold and maintain esteemed policy standards.

Network Diagram