Cybersecurity

Penetration Test Report Template

MegaCorpOne

Penetration Test Report

CyberTrain, 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

Contact Information

Company Name	CyberTrain, LLC
Contact Name	Wayne Hunt
Contact Title	Penetration Tester
Contact Phone	555.224.2411
Contact Email	wayneh@cybertrain.com

Document History

Version	Date	Author(s)	Comments
001	20.06.2024	Wayne H	created template, added company details.
002	27.06.2024	Wayne H	added content to summary strengths, weaknesses, executive summary, and vulnerability summary.
003	28.06.2024	Wayne H	added rest of content up to Mitre Att&ck table.
004	29.06.2024	Wayne H	completed Mitre Att&ck table, did final proof, and submitted.

Introduction

In accordance with MegaCorpOne’s policies, CyberTrain, LLC (henceforth known as CbrTr) 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 CbrTr during June of 2024.

For the testing, CybTr 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.

CybTr 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

CybTr 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

CybTr 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

CybTr'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
172.22.117.0/24 MCO.local *.Megacorpone.com	MegaCorpOne internal domain, range and public website

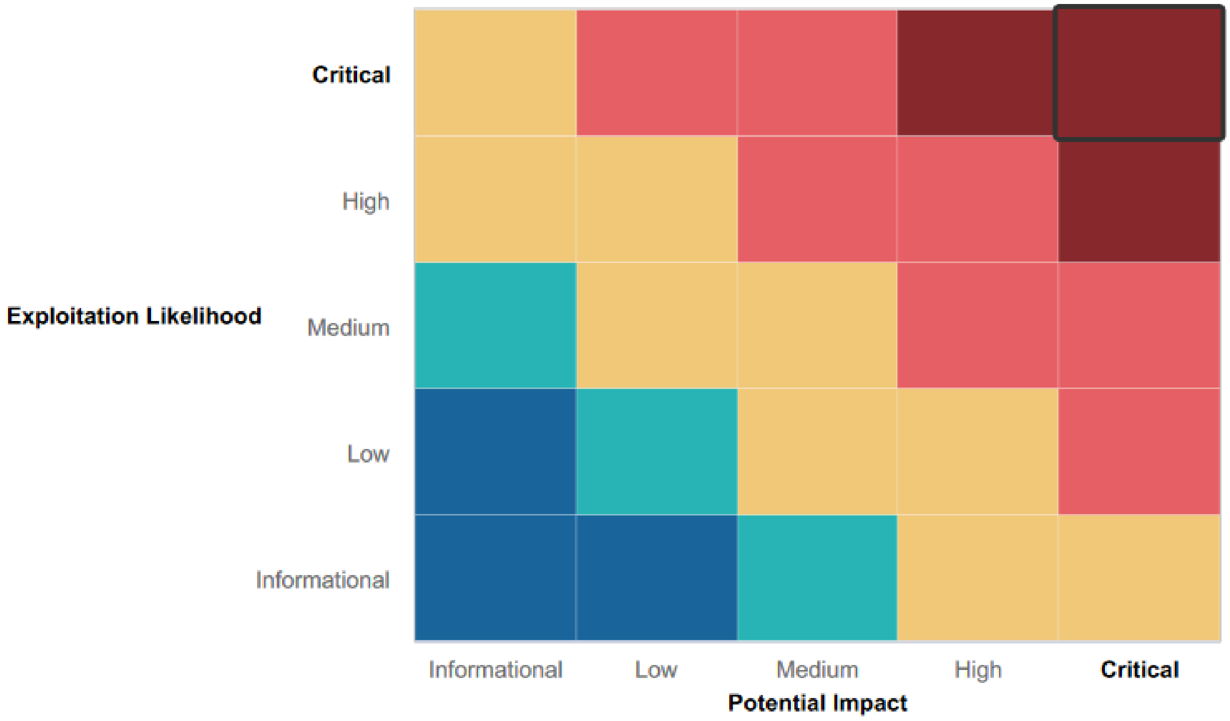
Executive Summary of Findings

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:



Summary of Strengths

While the assessment team was successful in finding several vulnerabilities, the team also recognized several strengths within MegaCorpOne's environment. These positives highlight the effective countermeasures and defenses that successfully prevented, detected, or denied an attack technique or tactic from occurring.

- Some metasploit tools, known for having great success accessing systems, were unable to connect to any Megacorpone machines.
- Effective segmentation of network resources to limit the impact of potential breaches.

Summary of Weaknesses

CybTr 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.

- Weak passwords and poor password hygiene are allowed.
- CVE vulnerabilities on apache servers.
- Megacorpone's domain server ip address is publicly available.
- Administrator credentials were located on the system in plain text.
- Successful elevation of privileges to Domain Administrator status.
- Compromised machines across both Linux and Windows 10 environments.
- Establishment of persistent backdoor access at high privilege levels, including on the Domain Controller.
- Vulnerabilities related to open ports potentially exposing systems to unauthorized access.
- Susceptibility to LLMNR attacks due to identified DNS server vulnerabilities.
- Potential vulnerabilities in Apache servers highlighted by Shodan report, though not directly tested.
- Critical need for immediate attention to open ports and DNS server vulnerabilities to prevent security breaches.

Executive Summary

During the engagement, our CyberTrain team successfully achieved all outlined objectives specified in the scope of work. We identified and extracted sensitive information, elevated our privileges to Domain Administrator status, and compromised two machines within Megacorpone's network. Our assessment uncovered seven vulnerabilities, primarily stemming from weak password management practices. Exploiting these weaknesses allowed us to gain unauthorized access to both Linux and Windows 10 machines, exfiltrate additional user credentials, and establish persistent backdoor access at the highest privilege levels. This included compromising the Domain Controller on the Windows network, enabling lateral movement across machines.

Furthermore, our investigation revealed vulnerabilities related to open ports, potentially exposing Megacorpone's systems to backdoor entry. Through open-source intelligence (OSINT) gathering, we pinpointed vulnerabilities associated with Megacorpone's DNS servers and identified susceptibility to LLMNR attacks. Additionally, our Shodan report highlighted potential vulnerabilities in Megacorpone's Apache servers, although these were not directly tested during our engagement. These findings underscore the critical need for immediate attention to these entry points to mitigate potential security breaches.

Our report's Vulnerability Findings section offers detailed insights into each identified vulnerability and provides comprehensive mitigation strategies. While some vulnerabilities necessitate urgent remediation due to their critical nature, the majority of recommended measures are straightforward to implement and cost-effective. Addressing these vulnerabilities promptly will significantly enhance Megacorpone's overall security posture, reducing the risk of unauthorized access and potential data breaches.

Below is outline of steps taken to achieve these results:

- Collect domain info by using OSINT techniques and tools (Shodan.io, Google hacking, and certificate transparency).
- Use Shodan.io and Recon-ng to discover domain server info.
- Performed advanced nmap scans with nse scripts.
- Exploit a machine with a python script.
- Used Metasploit to automate exploitation activities.
- Performed post-exploitation tasks (collecting password hashes).
- Performed poisoning attack on Windows network (msfvenom).
- Generated payloads using msfvenom.
- Operated meterpreter shells.
- Using Mimikatz (kiwi) to gather Windows credentials.
- Performed lateral movement to other machines on the network.
- Used dcsync attack to request the identified users NTLM passwords.
- Used john the ripper to crack NTLM and mscash2 hashes.

Summary Vulnerability Overview

Vulnerability	Severity
Weak password on public web application	**Critical**
Admin credentials in plain text on system	**Critical**
Port 21 (FTP) is open	**Critical**
Privilege Escalation	**High**
LLMNR	**High**
CVE Vulnerabilities	**Medium**
DNS IP addresses publicly exposed	**Medium**

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

Scan Type	Total
Hosts	194.56.244.87 – www.megacorpone.com 172.22.117.100 – host machine 172.22.117.150 – Linux machine 172.22.117.20 – Windows10 machine 172.22.117.10 – WinDC01 – Domain Controller
Ports	21 FTP 22 SSH 80 HTTP 88 Kerberos 139 RPC/SMB 443 HTTPS 445 SMB 3389 RDP

Exploitation Risk	Total
Critical	3
High	2
Medium	2
Low	0

Vulnerability Findings

Weak Password on Public Web Application

Risk Rating: Critical

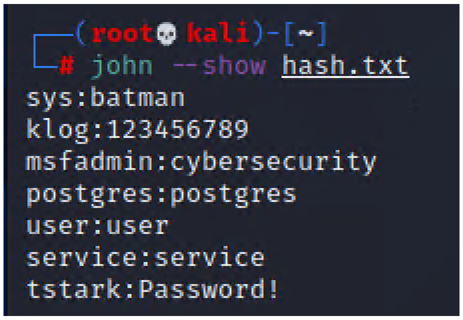
Description:
The site **vpn.megacorpone.com** is used to host the Cisco AnyConnect configuration file for MegaCorpOne. This site is secured with basic authentication but is susceptible to a dictionary attack. CybTr was able to use a username gathered from OSINT in combination with a wordlist in order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

Affected Hosts:

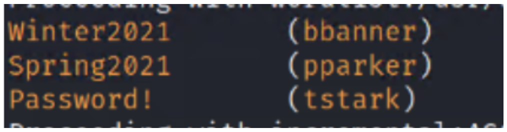
- vpn.megacorpone.com

Remediation:

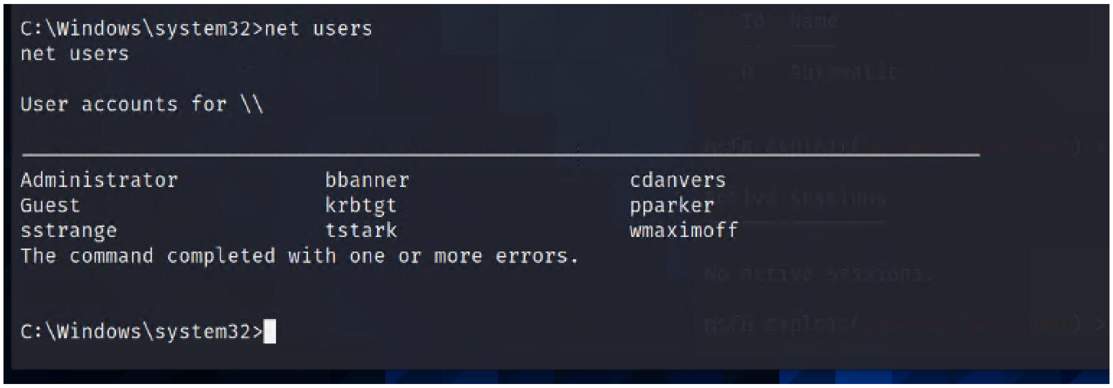
- Set up two-factor authentication instead of basic authentication to prevent dictionary attacks from being successful.
- Require a strong password complexity that requires passwords to be over 12 characters long, upper+lower case, & include a special character.
- Reset the user **thudson**'s password.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



```
┌──(root💀kali)-[~]
└─# john --show hash.txt
sys:batman
klog:123456789
msfadmin:cybersecurity
postgres:postgres
user:user
service:service
tstark:Password!
```



```
Winter2021       (bbanner)
Spring2021       (pparker)
Password!        (tstark)
```

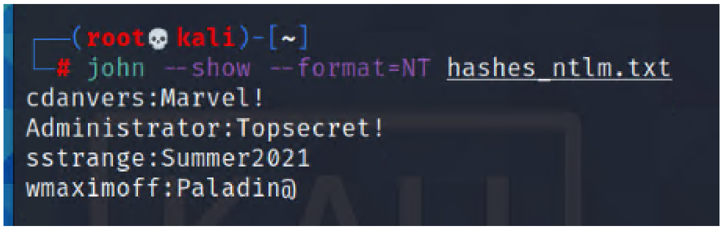


```
C:\Windows\system32>net users
net users

User accounts for \\

-------------------------------------------------------------------------------
Administrator            bbanner                  cdanvers
Guest                    krbtgt                   pparker
sstrange                 tstark                   wmaximoff
The command completed with one or more errors.


C:\Windows\system32>
```



```
┌──(root💀kali)-[~]
└─# john --show --format=NT hashes_ntlm.txt
cdanvers:Marvel!
Administrator:Topsecret!
sstrange:Summer2021
wmaximoff:Paladin@
```

Admin credentials in plain text on system

Risk Rating: Critical

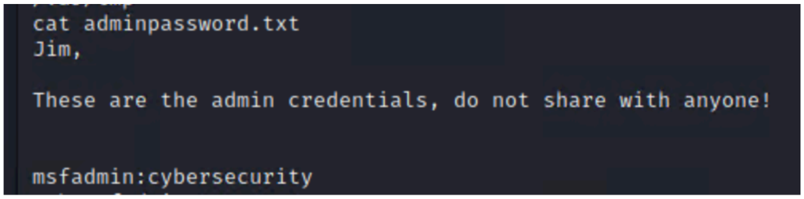
Description:
A Linux machine (host IP 172.22.117.150) was compromised using a malicious script from Metasploit due to a weak password. Once access was gained, administrative credentials were found in plaintext, enabling further privilege escalation. Password information from user files was extracted to create an SSH backdoor on port 22, ensuring persistent access.

Affected Hosts:

- 172.22.117.150 - Linux Machine

Remediation:

- Address the weak password vulnerability by implementing stronger passwords.
- Utilize secure password management software instead of manual storage, especially for users with administrative privileges.
- Maintain up-to-date software versions to prevent exploitation of vulnerabilities.
- Deploy and regularly update advanced antivirus software.
- Implement and maintain a robust firewall configuration to enhance network security.



```
cat adminpassword.txt
Jim,

These are the admin credentials, do not share with anyone!


msfadmin:cybersecurity
```

Port 21 (FTP) is open

Risk Rating: Critical

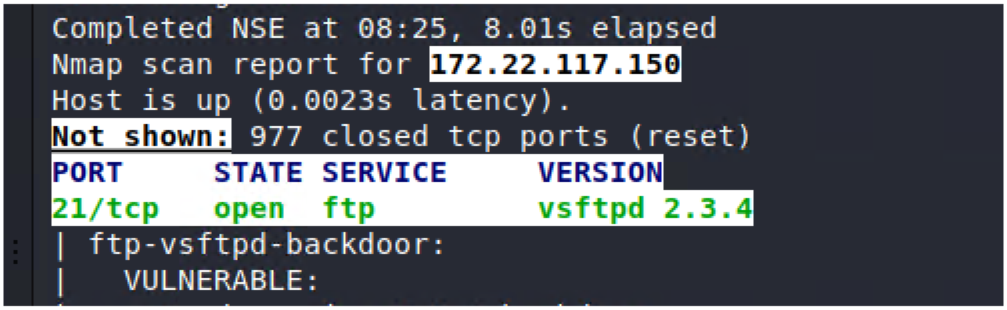
Description:
A Zenmap scan identified an open port 21 on Linux machine 172.22.117.150, known for vulnerabilities that can lead to backdoor attacks. These attacks enable persistent connections and potential data exploitation.

Affected Hosts:

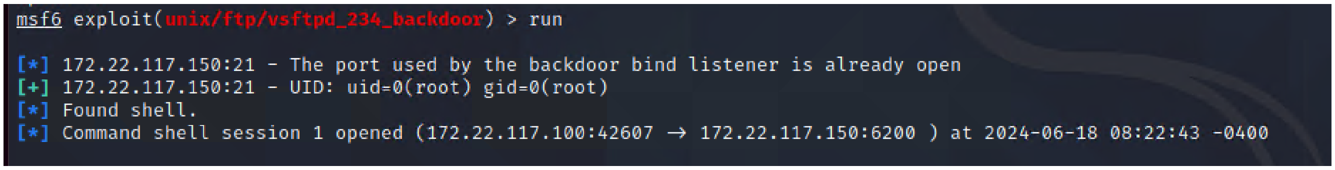
- 172.22.117.150 - Linux Machine

Remediation:

- Close port 21 to prevent unauthorized access.
- Address weak password concerns to enhance security.
- Maintain up-to-date software to mitigate vulnerabilities.
- Deploy and regularly update advanced antivirus/antimalware software.
- Utilize a firewall for enhanced network protection.
- The below screenshots show port 21 being open and then creating backdoor access through it.



```
Completed NSE at 08:25, 8.01s elapsed
Nmap scan report for 172.22.117.150
Host is up (0.0023s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE     VERSION
21/tcp    open  ftp         vsftpd 2.3.4
| ftp-vsftpd-backdoor:
|   VULNERABLE:
```



```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run

[*] 172.22.117.150:21 - The port used by the backdoor bind listener is already open
[+] 172.22.117.150:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (172.22.117.100:42607 → 172.22.117.150:6200 ) at 2024-06-18 08:22:43 -0400
```

Privilege Escalation

Risk Rating: High

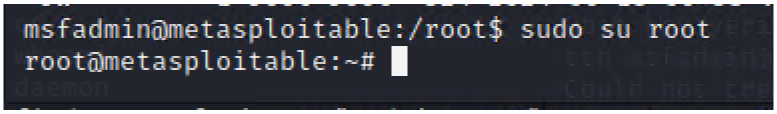
Description:
Privilege escalation is another issue related to poor password hygiene. Need to take action to prevent escalation of privileges if the system is compromised, as resolving weak passwords will reduce the risk, but not totally eliminate the risk.

Affected Hosts:

- 172.22.117.20 - Windows10 Machine
- 172.22.117.150 - Linux Machine

Remediation:

- Ensure the use of the principle of least privilege. This gives the employees the minimum level of access to do their roles.
- Follow password security best practices.
- Use MFA tools.
- Use vulnerability scanning tools.
- Monitor network traffic.
- Regular penetration testing.
- Patch and update systems regularly to protect against known malicious content.



```
msfadmin@metasploitable:/root$ sudo su root
root@metasploitable:~#
```

LLMNR

Risk Rating: High

Description:
LLMNR, also known as Local Link Multicast Name Resolution, is an outdated broadcast protocol designed as a fallback for DNS in local networks. However, it poses security risks as attackers can eavesdrop on LLMNR requests and send falsified responses. This can trick users into disclosing credentials, granting unauthorized access to the network. In a recent simulation, we successfully executed an LLMNR attack and obtained additional credentials previously unknown to us.

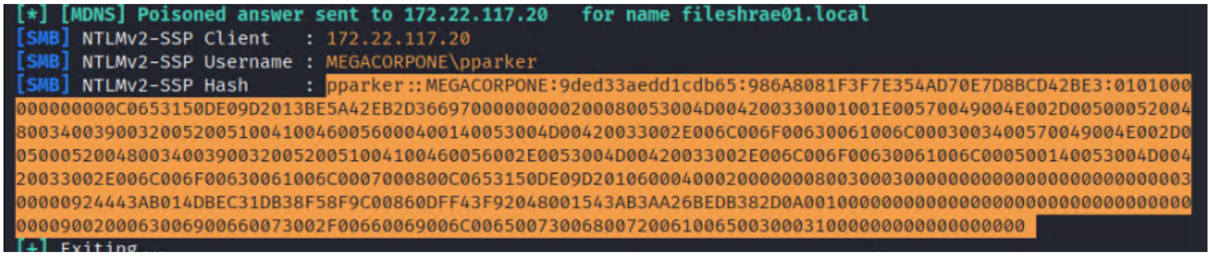
Affected Hosts:

- 172.22.117.20 - Windows10 Machine

Remediation:

- Turn off LLMNR protocol across all devices in the network to eliminate the risk of eavesdropping and spoofing attacks.
- Implement DNSSEC (Domain Name System Security Extensions) to ensure DNS responses are authenticated and tamper-proof.
- Encourage the use of HTTPS for all internal communications to prevent interception and credential theft.
- Segment the network to limit the scope of LLMNR traffic. This will reduce exposure.
- Conduct regular security awareness training to educate the team.
- Monitor and detect tools to detect traffic and spoofing attempts in real-time.

Below screenshot shows an intercept of traffic listening for LLMNR and retrieving credentials.



```
[*] [MDNS] Poisoned answer sent to 172.22.117.20   for name fileshrae01.local
[SMB] NTLMv2-SSP Client   : 172.22.117.20
[SMB] NTLMv2-SSP Username : MEGACORPONE\pparker
[SMB] NTLMv2-SSP Hash     : pparker::MEGACORPONE:9ded33aedd1cdb65:986A8081F3F7E354AD70E7D8BCD42BE3:0101000
000000000C0653150DE09D2013BE5A42EB2D36697000000000200080053004D004200330001001E00570049004E002D00500052004
800340039003200520051004100460056000400140053004D00420033002E006C006F00630061006C0003003400570049004E002D0
0500052004800340039003200520051004100460056002E0053004D00420033002E006C006F00630061006C00050014005300 4D004
20033002E006C006F00630061006C0007000800C0653150DE09D20106000400020000000800300030000000000000000000000003
00000924443AB014DBEC31DB38F58F9C00860DFF43F92048001543AB3AA26BEDB382D0A001000000000000000000000000000000
0000900200063006900660073002F00660069006C006500730068007200610065003000310000000000000000000
[+] Exiting...
```

CVE Vulnerabilities

Risk Rating: **Medium**

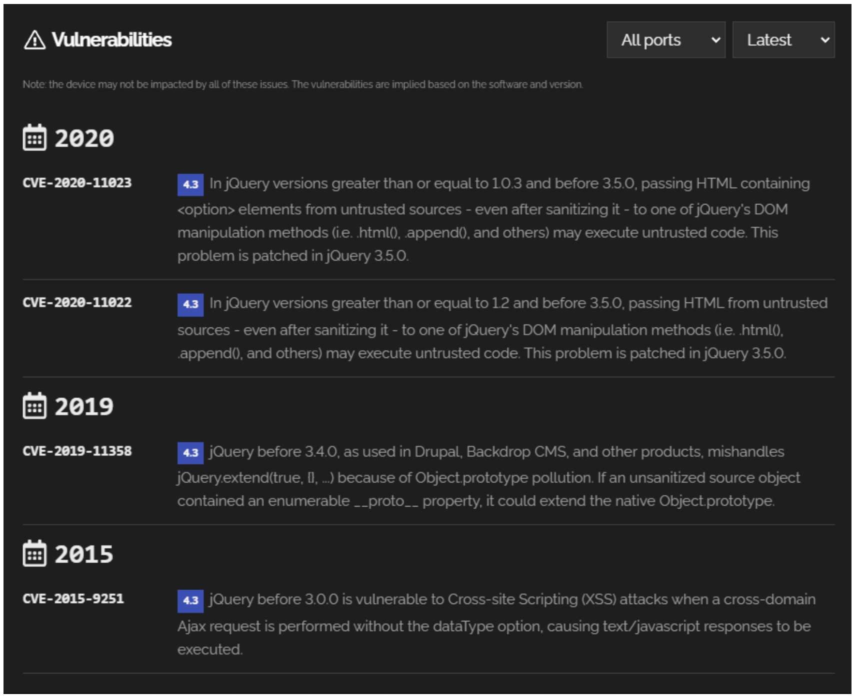
Description:
CVE vulnerabilities refer to entries in the Common Vulnerabilities and Exposures (CVE) system, which is a publicly accessible dictionary or catalog of known cybersecurity vulnerabilities. Each CVE entry provides a unique identifier, a brief description, and relevant references regarding specific security vulnerabilities found in software or hardware products. These vulnerabilities can range from flaws in software code, configuration errors, to design weaknesses that could be exploited by attackers to compromise the confidentiality, integrity, or availability of systems or data.

Affected Hosts:

- Apache servers

Remediation:

- Implement a robust patch management process to promptly apply security updates.
- Conduct regular vulnerability scans and assessments using automated tools.
- Segment networks to contain the impact of CVE vulnerabilities. This will limit the spread of the attacks and reduce exposure to critical systems and data.
- Following the principle of least privilege will reduce the attack surface.



⚠ Vulnerabilities

Note: the device may not be impacted by all of these issues. The vulnerabilities are implied based on the software and version.

2020

CVE-2020-11023 4.3 In jQuery versions greater than or equal to 1.0.3 and before 3.5.0, passing HTML containing <option> elements from untrusted sources - even after sanitizing it - to one of jQuery's DOM manipulation methods (i.e. .html(), .append(), and others) may execute untrusted code. This problem is patched in jQuery 3.5.0.

CVE-2020-11022 4.3 In jQuery versions greater than or equal to 1.2 and before 3.5.0, passing HTML from untrusted sources - even after sanitizing it - to one of jQuery's DOM manipulation methods (i.e. .html(), .append(), and others) may execute untrusted code. This problem is patched in jQuery 3.5.0.

2019

CVE-2019-11358 4.3 jQuery before 3.4.0, as used in Drupal, Backdrop CMS, and other products, mishandles jQuery.extend(true, {}, ...) because of Object.prototype pollution. If an unsanitized source object contained an enumerable __proto__ property, it could extend the native Object.prototype.

2015

CVE-2015-9251 4.3 jQuery before 3.0.0 is vulnerable to Cross-site Scripting (XSS) attacks when a cross-domain Ajax request is performed without the dataType option, causing text/javascript responses to be executed.

DNS IP addresses publicly exposed

Risk Rating: **Medium**

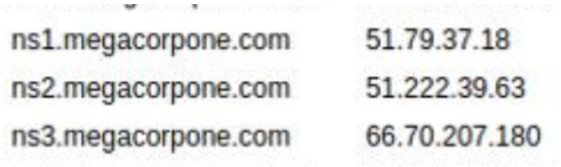
Description:
Using Recon-ng searches we were able to discover the IP addresses to three of Megacorpone's Named Servers (NS). Recon-ng is a publicly available tool, so threat actors would also be able to discover this information. This can potentially leave Megacorpone vulnerable to DNS poisoning, or spoofing, where users are redirected away from your site and to a malicious site of the threat actors choosing.

Affected Hosts:

- 51.79.37.18 - ns1.megacorpone.com
- 51.222.39.63 - ns2.megacorpone.com
- 66.70.207.180 - ns3.megacorpone.com

Remediation:

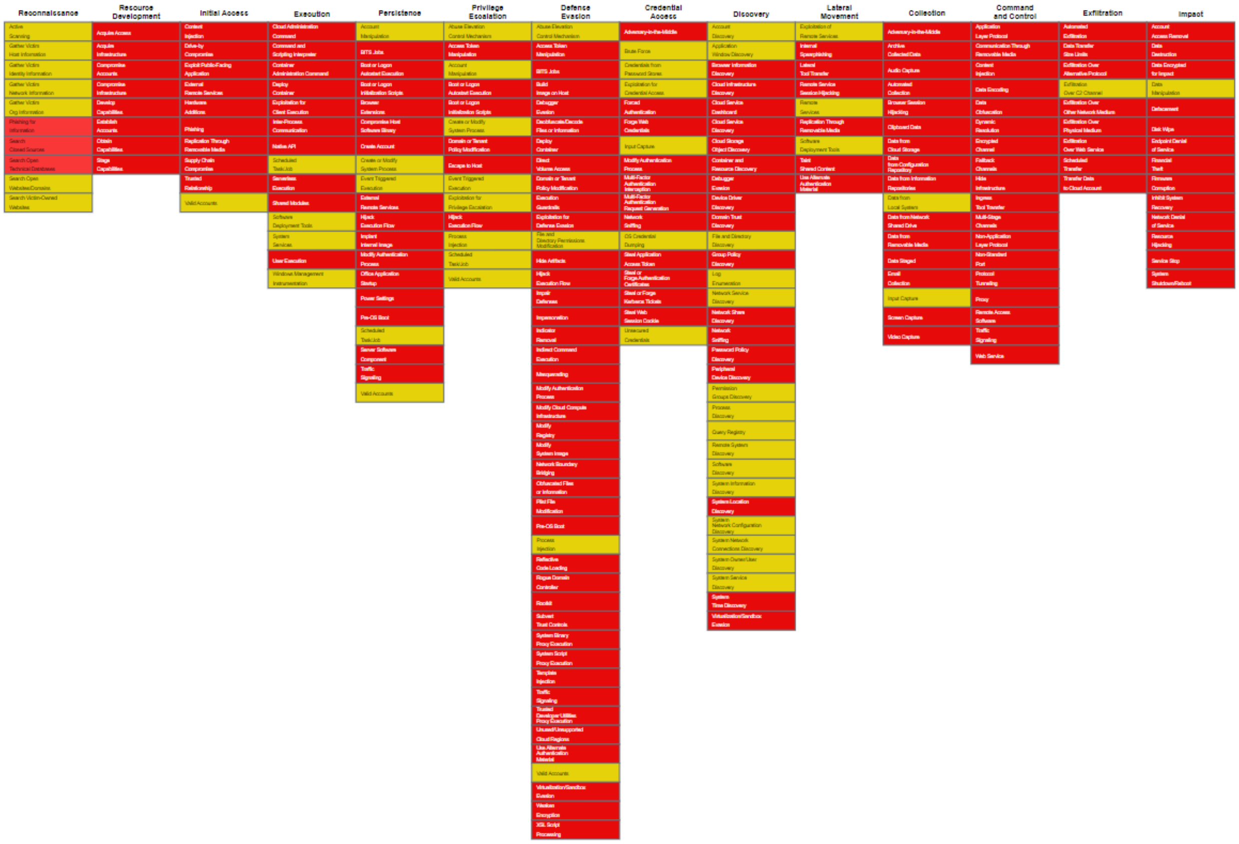
- Utilize IP whitelisting to allow only authorized IP addresses (such as internal networks or known entities) to communicate with DNS servers.
- Make the IP addresses for the servers private.
- Implement firewall rules and access controls to restrict access to DNS servers from unauthorized networks or IP addresses.



```
ns1.megacorpone.com     51.79.37.18
ns2.megacorpone.com     51.222.39.63
ns3.megacorpone.com     66.70.207.180
```

MITRE ATT&CK Navigator Map

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



Reconnaissance

- 🟨 Active Scanning
- 🟨 Gather Victim Host Information
- 🟨 Gather Victim Identity Information
- 🟨 Gather Victim Network Information
- 🟨 Gather Victim Org Information
- 🟥 Phishing for Information
- 🟥 Search Closed Sources
- 🟥 Search Open Technical Databases
- 🟨 Search Open Websites/Domains
- 🟨 Search Victim-Owned Websites

Resource Development

- 🟥 Acquire Access
- 🟥 Acquire Infrastructure
- 🟥 Compromise Accounts
- 🟥 Compromise Infrastructure
- 🟥 Develop Capabilities
- 🟥 Establish Accounts
- 🟥 Obtain Capabilities
- 🟥 Stage Capabilities

Initial Access

- 🟥 Content Injection
- 🟥 Drive-by Compromise
- 🟥 Exploit Public-Facing Application
- 🟥 External Remote Services
- 🟥 Hardware Additions
- 🟥 Phishing
- 🟥 Replication Through Removable Media
- 🟥 Supply Chain Compromise
- 🟥 Trusted Relationship
- 🟨 Valid Accounts

Execution

- 🟥 Cloud Administration Command
- 🟥 Command and Scripting Interpreter
- 🟥 Container Administration Command
- 🟥 Deploy Container
- 🟥 Exploitation for Client Execution
- 🟥 Inter-Process Communication
- 🟥 Native API
- 🟨 Scheduled Task/Job
- 🟥 Serverless Execution
- 🟥 Shared Modules
- 🟨 Software Deployment Tools
- 🟨 System Services
- 🟥 User Execution
- 🟨 Windows Management Instrumentation

Persistence

- 🟨 Account Manipulation
- 🟥 BITS Jobs
- 🟥 Boot or Logon Autostart Execution
- 🟥 Boot or Logon Initialization Scripts
- 🟥 Browser Extensions
- 🟥 Compromise Host Software Binary
- 🟥 Create Account
- 🟨 Create or Modify System Process
- 🟨 Event Triggered Execution
- 🟥 External Remote Services
- 🟥 Hijack Execution Flow
- 🟥 Implant Internal Image
- 🟥 Modify Authentication Process
- 🟥 Office Application Startup
- 🟥 Power Settings
- 🟥 Pre-OS Boot
- 🟨 Scheduled Task/Job
- 🟥 Server Software Component
- 🟥 Traffic Signaling
- 🟨 Valid Accounts

Privilege Escalation

- 🟨 Abuse Elevation Control Mechanism
- 🟥 Access Token Manipulation
- 🟨 Account Manipulation
- 🟥 Boot or Logon Autostart Execution
- 🟥 Boot or Logon Initialization Scripts
- 🟨 Create or Modify System Process
- 🟥 Domain or Tenant Policy Modification
- 🟥 Escape to Host
- 🟨 Event Triggered Execution
- 🟨 Exploitation for Privilege Escalation
- 🟥 Hijack Execution Flow
- 🟨 Process Injection
- 🟨 Scheduled Task/Job
- 🟨 Valid Accounts

Defense Evasion

- 🟨 Abuse Elevation Control Mechanism
- 🟥 Access Token Manipulation
- 🟥 BITS Jobs
- 🟥 Build Image on Host
- 🟥 Debugger Evasion
- 🟥 Deobfuscate/Decode Files or Information
- 🟥 Deploy Container
- 🟥 Direct Volume Access
- 🟥 Domain or Tenant Policy Modification
- 🟥 Execution Guardrails
- 🟥 Exploitation for Defense Evasion
- 🟨 File and Directory Permissions Modification
- 🟥 Hide Artifacts
- 🟥 Hijack Execution Flow
- 🟥 Impair Defenses
- 🟥 Impersonation
- 🟥 Indicator Removal
- 🟥 Indirect Command Execution
- 🟥 Masquerading
- 🟥 Modify Authentication Process
- 🟥 Modify Cloud Compute Infrastructure
- 🟥 Modify Registry
- 🟥 Modify System Image
- 🟥 Network Boundary Bridging
- 🟥 Obfuscated Files or Information
- 🟥 Plist File Modification
- 🟥 Pre-OS Boot
- 🟨 Process Injection
- 🟥 Reflective Code Loading
- 🟥 Rogue Domain Controller
- 🟥 Rootkit
- 🟥 Subvert Trust Controls
- 🟥 System Binary Proxy Execution
- 🟥 System Script Proxy Execution
- 🟥 Template Injection
- 🟥 Traffic Signaling
- 🟥 Trusted Developer Utilities Proxy Execution
- 🟥 Unused/Unsupported Cloud Regions
- 🟥 Use Alternate Authentication Material
- 🟨 Valid Accounts
- 🟥 Virtualization/Sandbox Evasion
- 🟥 Weaken Encryption
- 🟥 XSL Script Processing

Credential Access

- 🟥 Adversary-in-the-Middle
- 🟨 Brute Force
- 🟨 Credentials from Password Stores
- 🟨 Exploitation for Credential Access
- 🟥 Forced Authentication
- 🟥 Forge Web Credentials
- 🟨 Input Capture
- 🟥 Modify Authentication Process
- 🟥 Multi-Factor Authentication Interception
- 🟥 Multi-Factor Authentication Request Generation
- 🟥 Network Sniffing
- 🟨 OS Credential Dumping
- 🟥 Steal Application Access Token
- 🟥 Steal or Forge Authentication Certificates
- 🟥 Steal or Forge Kerberos Tickets
- 🟥 Steal Web Session Cookie
- 🟨 Unsecured Credentials

Discovery

- 🟨 Account Discovery
- 🟨 Application Window Discovery
- 🟥 Browser Information Discovery
- 🟥 Cloud Infrastructure Discovery
- 🟥 Cloud Service Dashboard
- 🟥 Cloud Service Discovery
- 🟥 Cloud Storage Object Discovery
- 🟥 Container and Resource Discovery
- 🟥 Debugger Evasion
- 🟥 Device Driver Discovery
- 🟥 Domain Trust Discovery
- 🟨 File and Directory Discovery
- 🟥 Group Policy Discovery
- 🟨 Log Enumeration
- 🟨 Network Service Discovery
- 🟥 Network Share Discovery
- 🟥 Network Sniffing
- 🟥 Password Policy Discovery
- 🟥 Peripheral Device Discovery
- 🟨 Permission Groups Discovery
- 🟨 Process Discovery
- 🟨 Query Registry
- 🟨 Remote System Discovery
- 🟨 Software Discovery
- 🟨 System Information Discovery
- 🟥 System Location Discovery
- 🟨 System Network Configuration Discovery
- 🟨 System Network Connections Discovery
- 🟨 System Owner/User Discovery
- 🟨 System Service Discovery
- 🟥 System Time Discovery
- 🟥 Virtualization/Sandbox Evasion

Lateral Movement

- 🟨 Exploitation of Remote Services
- 🟥 Internal Spearphishing
- 🟥 Lateral Tool Transfer
- 🟥 Remote Service Session Hijacking
- 🟨 Remote Services
- 🟥 Replication Through Removable Media
- 🟨 Software Deployment Tools
- 🟥 Taint Shared Content
- 🟥 Use Alternate Authentication Material

Collection

- 🟥 Adversary-in-the-Middle
- 🟥 Archive Collected Data
- 🟥 Audio Capture
- 🟥 Automated Collection
- 🟥 Browser Session Hijacking
- 🟥 Clipboard Data
- 🟥 Data from Cloud Storage
- 🟥 Data from Configuration Repository
- 🟥 Data from Information Repositories
- 🟨 Data from Local System
- 🟥 Data from Network Shared Drive
- 🟥 Data from Removable Media
- 🟥 Data Staged
- 🟥 Email Collection
- 🟨 Input Capture
- 🟥 Screen Capture
- 🟥 Video Capture

Command and Control

- 🟥 Application Layer Protocol
- 🟥 Communication Through Removable Media
- 🟥 Content Injection
- 🟥 Data Encoding
- 🟥 Data Obfuscation
- 🟥 Dynamic Resolution
- 🟥 Encrypted Channel
- 🟥 Fallback Channels
- 🟥 Hide Infrastructure
- 🟥 Ingress Tool Transfer
- 🟥 Multi-Stage Channels
- 🟥 Non-Application Layer Protocol
- 🟥 Non-Standard Port
- 🟥 Protocol Tunneling
- 🟥 Proxy
- 🟥 Remote Access Software
- 🟥 Traffic Signaling
- 🟥 Web Service

Exfiltration

- 🟥 Automated Exfiltration
- 🟥 Data Transfer Size Limits
- 🟥 Exfiltration Over Alternative Protocol
- 🟨 Exfiltration Over C2 Channel
- 🟥 Exfiltration Over Other Network Medium
- 🟥 Exfiltration Over Physical Medium
- 🟥 Exfiltration Over Web Service
- 🟥 Scheduled Transfer
- 🟥 Transfer Data to Cloud Account

Impact

- 🟥 Account Access Removal
- 🟥 Data Destruction
- 🟥 Data Encrypted for Impact
- 🟨 Data Manipulation
- 🟥 Defacement
- 🟥 Disk Wipe
- 🟥 Endpoint Denial of Service
- 🟥 Financial Theft
- 🟥 Firmware Corruption
- 🟥 Inhibit System Recovery
- 🟥 Network Denial of Service
- 🟥 Resource Hijacking
- 🟥 Service Stop
- 🟥 System Shutdown/Reboot

Legend:

🟨 Performed successfully
🟥 Failure to perform