MegaCorpOne

Penetration Test Report

IronCurtain Testing, 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	IronCurtain Testing, LLC
Contact Name	Collin Janecka
Contact Title	Penetration Tester
Contact Phone	888.736.8378 *(888.PEN.TESTS)*
Contact Email	collinj@ictesting.com

Document History

Version	**Date**	**Author(s)**	**Comments**
001	09/04/2023	Collin Janecka	Initial Draft v1
002	09/05/2023	Collin Janecka	Grammar revisions
003	09/06/2023	Collin Janecka	Final Draft v1
004	09/09/2023	Collin Janecka	Final Draft v2
005	10/07/2023	Collin Janecka	Final Draft v3

Introduction

In accordance with MegaCorpOne's policies**,** IronCurtain Testing LLC (henceforth known as IC Testing) 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 IC Testing during August of 2023.

For the testing, IC Testing 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.

IC Testing 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

IC Testing 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

IC Testing 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

IC Testing’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

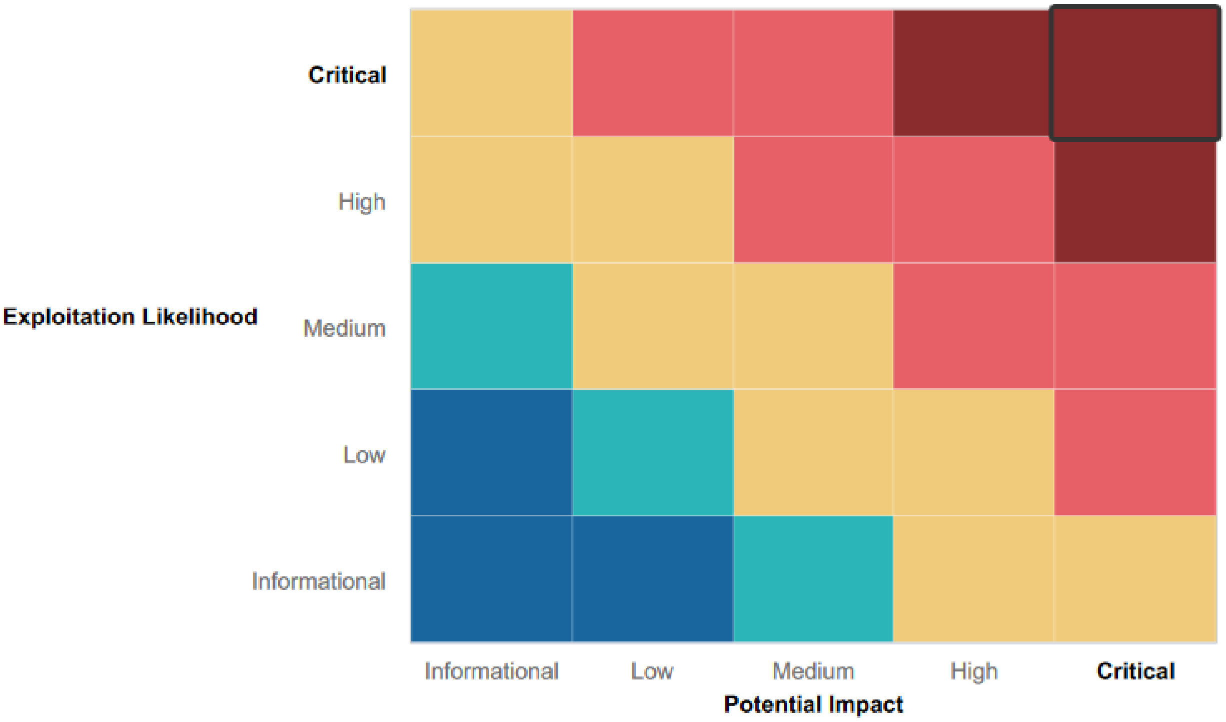
Executive Summary of Findings

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:



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.

- **Resilience to Exploitation Attempts**: MegaCorpOne has exhibited remarkable resilience against exploitation attempts employing Metasploit tools, recognized for their effectiveness in vulnerability assessment. Despite numerous efforts, unauthorized access was achieved only on a limited subset of MegaCorpOne's machines, and this required a substantial investment of time and resources. This underscores the organization's formidable defense posture and the efficacy of its security measures in countering well-documented attack vectors.

- **Committed to Security**: A commitment to security was demonstrated, by conducting regular internal security audits and assessments, proactively identifying and addressing vulnerabilities.

- **Network Architecture**: The network architecture was designed with security in mind, incorporating segmentation and network isolation to contain potential breaches.

- **Encrypted Data**: Data encryption methodologies have been instituted to fortify the safeguarding of sensitive information. Although certain instances of data compromise were encountered, the presence of encryption measures is a commendable security practice.

- **Isolated Network**: By making the internal domain largely inaccessible from the external network, the organization has effectively reduced its attack surface. This makes it more challenging for attackers to gain a foothold inside the network.

Summary of Weaknesses

IC Testing 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**: The organization permits the use of weak passwords, posing a significant security risk by making it easier for attackers to gain unauthorized access to systems and sensitive data.

- **Open Port 21**: Port 21, associated with FTP, is left open, potentially exposing the network to exploitation and backdoor attacks if not properly secured.

- **Administrative Credentials in Plain Text**: Administrative-level credentials are stored in plain text on the system, creating a serious vulnerability that could lead to unauthorized access and data compromise.

- **LLMNR Vulnerability**: LLMNR (Local Link Multicast Name Resolution) is susceptible to exploitation, allowing attackers to intercept credentials and potentially redirect users to malicious sites, compromising the integrity of the network.

- **Privilege Escalation**: Privilege escalation vulnerabilities exist, which could enable attackers to elevate their access privileges within the network, increasing the potential damage they can inflict.

- **Publicly Available Domain Server IP Addresses**: IP addresses for Megacorpone's domain servers are publicly accessible, making the organization susceptible to DNS attacks, including DNS poisoning and spoofing.

- **CVE Vulnerabilities on Apache Servers**: The presence of CVE (Common Vulnerabilities and Exposures) vulnerabilities on Apache servers raises concerns about the security of these servers and their potential exploitation by malicious actors.

Executive Summary

IronCurtain Testing has successfully completed a comprehensive security assessment in alignment with the specified scope of work for this engagement. The evaluation aimed to identify vulnerabilities and potential security weaknesses within Megacorpone's network infrastructure. Our assessment included attempts to locate and exfiltrate sensitive information, privilege escalation, and compromise of networked machines. The procedural steps were as follows:

Phase 1 – Reconnaissance

During this phase, IC Testing gathered information about Megacorpone using Google Dorking techniques. Specific search operators were used, within the google search bar, to focus on the website and find sensitive information, including employee contact data and hidden files. This revealed a security vulnerability and the potential exposure of confidential information.

The subsequent step encompassed a comprehensive enumeration of Megacorpone's domain, employing a suite of sophisticated tools including Shodan.io, Nmap, Zenmap, and Recon-ng. The results of this revelaed the accessibility of ports 21, 22, 80, and 443 on Megacorpone's domain, providing valuable insights into the server's geographic location and a compilation of Common Vulnerabilities and Exposures (CVEs).

Phase 2 – Implementation/Exploitation

IC Testing obtained usernames and passwords for five (5) users through deduction, revealing vulnerable password selections. These compromised users included: Tom Hudson (Web Designer), Tanya Rivera (Senior Developer), Matt Smith (Marketing Director), Mike Carlow (VP of Legal), and Alan Grofield (IT & Security Director).

The presence of a shell script named "vpn.sh" was discovered, which was specifically crafted to enable direct access to "www.megacorpone.com". Subsequently, this script was subjected to modification using the text editor "nano", facilitating the incorporation of the compromised login credentials. This customized script was employed to gain access to Megacorpone's webpage.

After gaining access to the webpage, a Python file was identified that exhibited a vulnerability, enabling remote access through an exploit. Upon penetrating the system; a plaintext document was located, addressed to Jim, which contained administrator-level credentials. These credentials were used to gain access to Megacorpone's Linux server with an administrator level account. From here, privileges were elevated from administrator to a root level user.

This elevated status granted unrestricted access to all files, commands, and system settings, thereby affording the ability to manipulate files, install or uninstall software, and modify system configurations. Using the root level privileges acquired, a directory containing hashed system and user passwords was found. These hashes were copied into a new text file and then successfully cracked using the tool “John the Ripper”.

In order to establish persistence an additional port was added to the “sshd_config” file, specifically port 10022. This facilitated a hidden alternative access point that was previously unavailable, nor should be. Additionally, a novel user with administrative privileges, was created under the guise of a system service to establish a backdoor for covert access.

In light of these findings, it is imperative that Megacorpone takes immediate and comprehensive action to address these vulnerabilities and strengthen its cybersecurity posture. Failure to do so could result in severe consequences for the security and integrity of Megacorpone's network infrastructure and sensitive data. Remediation strategies can be found within the “Vulnerability Findings” section of this report below.

Summary Vulnerability Overview

Vulnerability	Severity
Weak password(s) on public web application	**Critical**
Port 21 (FTP) is open	**Critical**
Admin credentials in plain text	**Critical**
Privilege Escalation	**High**
LLMNR	**High**
Exposed IP addresses for three of Megacorpone’s domain servers	**Medium**
CVE Vulnerabilities	**Medium**

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

Scan Type	Total
Hosts	www.megacorpone.com – 149.56.244.87 WinDC01 Domain Controller – 172.22.117.10 Windows10 machine – 172.22.117.20 Host machine – 172.22.117.100 Linux machine – 172.22.117.150
Ports	21 FTP, 22 SSH, 80 HTTP, 88 Kerberos, 443 HTTPS, 139 RPC/SMB, 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:

Megacorpone is currently exposed to a critical security risk due to the lax enforcement of strong password policies. This vulnerability was exploited during our assessment, and it poses a significant threat to the organization's information security. We were able to identify one of these weak passwords and leverage it to gain unauthorized access to a Linux machine, where we executed a script file. If this script had been malicious in nature, it could have inflicted severe damage to both the system and its data.

Furthermore, once we established this initial unauthorized connection, we were able to escalate our access privileges to the highest level, known as root access, effectively gaining complete control over the Linux machine. This elevated access allowed us to implant a backdoor, which provides an ongoing entry point for malicious activities in the future, thereby compromising the long-term security and integrity of the machine.

In addition to the Linux machine, we exploited the same weak password to conduct a password spray attack using Metasploit. This attack vector enabled us to breach two critical targets within the network: a Windows10 machine and the Windows Domain Controller (WinDC01). By escalating our privileges to the System level, we gained extensive control over these systems and also obtained additional user credentials.

Affected Host(s): vpn.megacorpone.com, 172.22.117.20 – Windows10 machine, 172.22.117.10 – WinDC01 – Domain Controller

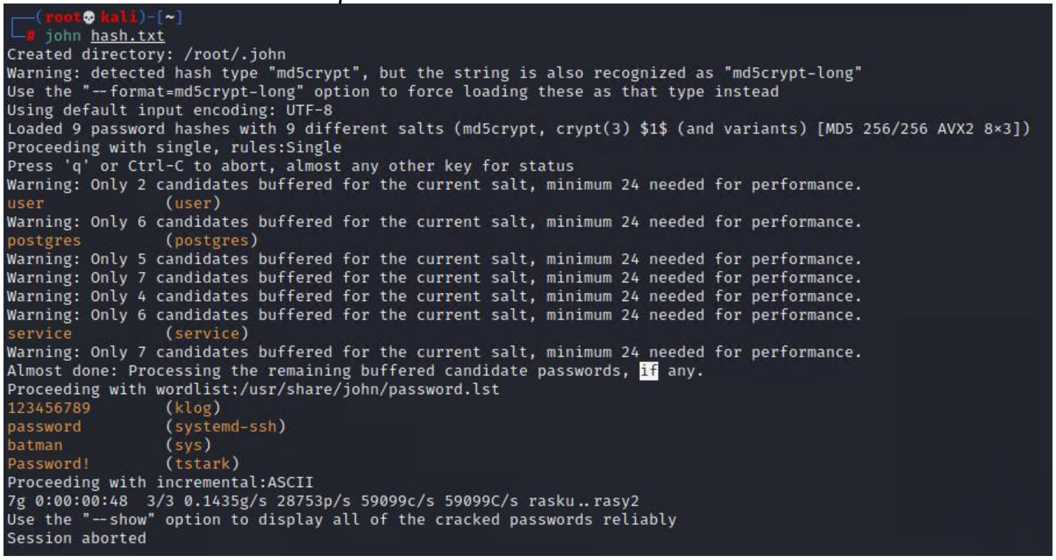
Remediation:

a) **Require Complex Passwords**: Enforce a strict password policy that mandates the use of complex passwords. Passwords should contain a minimum of eight characters and include a combination of upper- and lower-case letters, numbers, and symbols. This complexity makes it significantly harder for attackers to guess or crack passwords through brute force or dictionary attacks.

b) **Password Expiry**: Implement a password change policy that requires users to change their passwords every 60-90 days. Regular password changes help prevent long-term exposure to the same weak credentials.

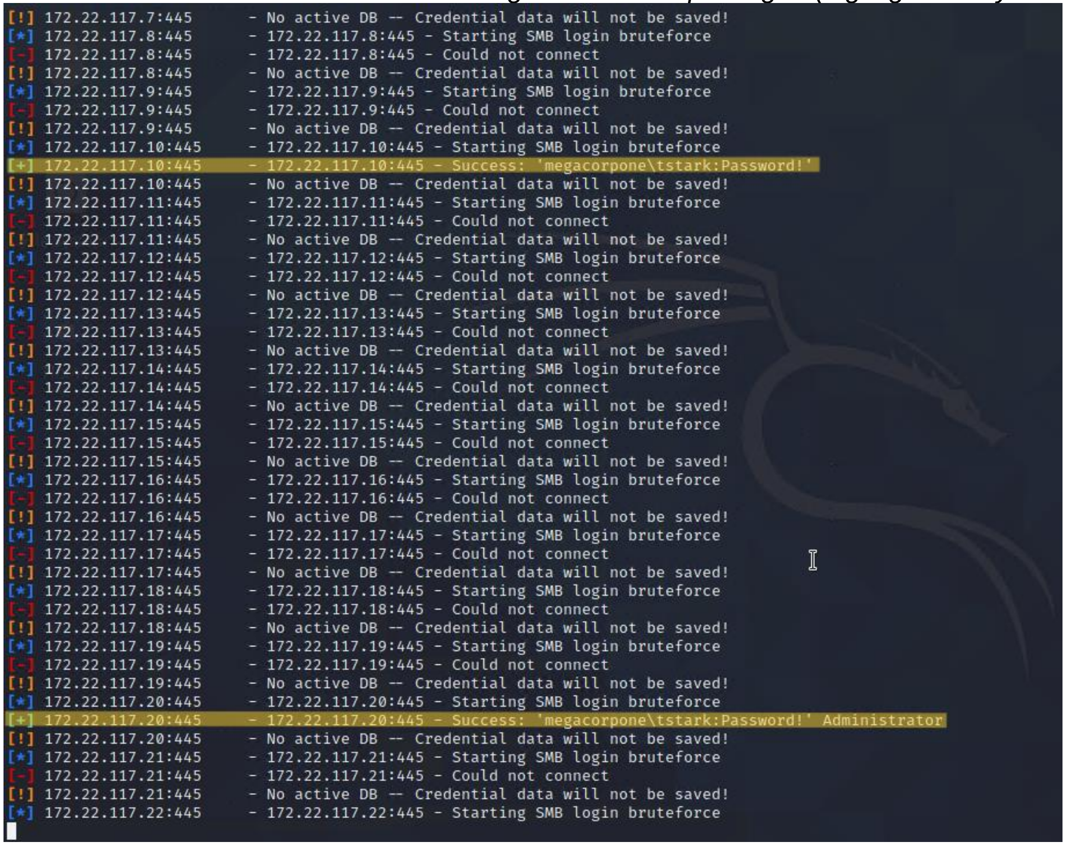
c) **Multi-Factor Authentication (MFA)**: If possible, enable multi-factor authentication (MFA) across all relevant systems and services. MFA adds an additional layer of security by requiring users to provide multiple forms of authentication before gaining access, making it much more challenging for unauthorized individuals to breach accounts.

Additional user names and passwords accessed



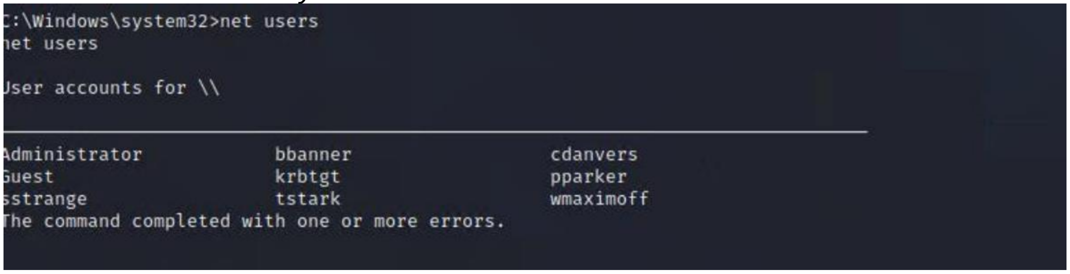
```
┌──(root💀 kali)-[~]
└─# john hash.txt
Created directory: /root/.john
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 9 password hashes with 9 different salts (md5crypt, crypt(3) $1$ (and variants) [MD5 256/256 AVX2 8×3])
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 2 candidates buffered for the current salt, minimum 24 needed for performance.
user             (user)
Warning: Only 6 candidates buffered for the current salt, minimum 24 needed for performance.
postgres         (postgres)
Warning: Only 5 candidates buffered for the current salt, minimum 24 needed for performance.
Warning: Only 7 candidates buffered for the current salt, minimum 24 needed for performance.
Warning: Only 4 candidates buffered for the current salt, minimum 24 needed for performance.
Warning: Only 6 candidates buffered for the current salt, minimum 24 needed for performance.
service          (service)
Warning: Only 7 candidates buffered for the current salt, minimum 24 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
123456789        (klog)
password         (systemd-ssh)
batman           (sys)
Password!        (tstark)
Proceeding with incremental:ASCII
7g 0:00:00:48  3/3 0.1435g/s 28753p/s 59099c/s 59099C/s rasku..rasy2
Use the "--show" option to display all of the cracked passwords reliably
Session aborted
```

Access to Domain Controller & Escalating user access privileges (highlighted in yellow)



```
[!] 172.22.117.7:445      - No active DB -- Credential data will not be saved!
[*] 172.22.117.8:445      - 172.22.117.8:445 - Starting SMB login bruteforce
[-] 172.22.117.8:445      - 172.22.117.8:445 - Could not connect
[!] 172.22.117.8:445      - No active DB -- Credential data will not be saved!
[*] 172.22.117.9:445      - 172.22.117.9:445 - Starting SMB login bruteforce
[-] 172.22.117.9:445      - 172.22.117.9:445 - Could not connect
[!] 172.22.117.9:445      - No active DB -- Credential data will not be saved!
[*] 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!'
[!] 172.22.117.10:445     - No active DB -- Credential data will not be saved!
[*] 172.22.117.11:445     - 172.22.117.11:445 - Starting SMB login bruteforce
[-] 172.22.117.11:445     - 172.22.117.11:445 - Could not connect
[!] 172.22.117.11:445     - No active DB -- Credential data will not be saved!
[*] 172.22.117.12:445     - 172.22.117.12:445 - Starting SMB login bruteforce
[-] 172.22.117.12:445     - 172.22.117.12:445 - Could not connect
[!] 172.22.117.12:445     - No active DB -- Credential data will not be saved!
[*] 172.22.117.13:445     - 172.22.117.13:445 - Starting SMB login bruteforce
[-] 172.22.117.13:445     - 172.22.117.13:445 - Could not connect
[!] 172.22.117.13:445     - No active DB -- Credential data will not be saved!
[*] 172.22.117.14:445     - 172.22.117.14:445 - Starting SMB login bruteforce
[-] 172.22.117.14:445     - 172.22.117.14:445 - Could not connect
[!] 172.22.117.14:445     - No active DB -- Credential data will not be saved!
[*] 172.22.117.15:445     - 172.22.117.15:445 - Starting SMB login bruteforce
[-] 172.22.117.15:445     - 172.22.117.15:445 - Could not connect
[!] 172.22.117.15:445     - No active DB -- Credential data will not be saved!
[*] 172.22.117.16:445     - 172.22.117.16:445 - Starting SMB login bruteforce
[-] 172.22.117.16:445     - 172.22.117.16:445 - Could not connect
[!] 172.22.117.16:445     - No active DB -- Credential data will not be saved!
[*] 172.22.117.17:445     - 172.22.117.17:445 - Starting SMB login bruteforce
[-] 172.22.117.17:445     - 172.22.117.17:445 - Could not connect
[!] 172.22.117.17:445     - No active DB -- Credential data will not be saved!
[*] 172.22.117.18:445     - 172.22.117.18:445 - Starting SMB login bruteforce
[-] 172.22.117.18:445     - 172.22.117.18:445 - Could not connect
[!] 172.22.117.18:445     - No active DB -- Credential data will not be saved!
[*] 172.22.117.19:445     - 172.22.117.19:445 - Starting SMB login bruteforce
[-] 172.22.117.19:445     - 172.22.117.19:445 - Could not connect
[!] 172.22.117.19:445     - No active DB -- Credential data will not be saved!
[*] 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
[!] 172.22.117.20:445     - No active DB -- Credential data will not be saved!
[*] 172.22.117.21:445     - 172.22.117.21:445 - Starting SMB login bruteforce
[-] 172.22.117.21:445     - 172.22.117.21:445 - Could not connect
[!] 172.22.117.21:445     - No active DB -- Credential data will not be saved!
[*] 172.22.117.22:445     - 172.22.117.22:445 - Starting SMB login bruteforce
```

Used SYSTEM-level access privileges on the Domain Controller to extract supplementary credentials via the DCSync mechanism



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

ser accounts for \\

-------------------------------------------------------------------------------
dministrator             bbanner                  cdanvers
uest                     krbtgt                   pparker
sstrange                 tstark                   wmaximoff
he command completed with one or more errors.
```

Port 21 (FTP) is open

Risk Rating: Critical

Description:

A Zenmap scan has identified a critical security vulnerability on the Windows machine with the IP address 172.22.117.20. Specifically, Port 21, which is commonly associated with FTP (File Transfer Protocol), has been found to be open. This represents a significant security risk to the system. Port 21, when left open, exposes the system to known vulnerabilities that can be exploited by attackers, potentially leading to backdoor attacks.

Backdoor attacks are particularly concerning as they allow malicious actors to establish a persistent and covert connection with the compromised machine. Once this connection is established, attackers can gain unauthorized access to the system, exfiltrate sensitive data, manipulate files, and potentially install malware or other malicious software without detection. This presents a grave threat to the confidentiality, integrity, and availability of the affected system's data and functionality.

Affected Host(s): 172.22.117.20 – Windows10 machine

Remediation:

a) **Close Port 21**: The immediate action to mitigate this vulnerability is to close Port 21. This can be achieved by configuring the firewall settings on the Windows machine to block incoming and outgoing traffic on this port. This step will effectively prevent unauthorized access through this specific entry point.

b) **Correct Weak Password Issue**: In addition to closing the port, it is crucial to address any weak password issues associated with FTP accounts on the system. Enforce strong password policies and consider implementing multi-factor authentication (MFA) for FTP access to enhance security.

c) **Keep Software Updated**: Ensure that all software, including the FTP server software and the operating system, is regularly updated with the latest security patches and updates. Vulnerabilities often arise due to outdated software, and keeping everything up-to-date is a fundamental defense measure.

d) **Use Advanced Antivirus/Antimalware**: Employ advanced antivirus and antimalware solutions on the Windows machine. These security tools can help detect and mitigate threats in real-time. Ensure that the antivirus and antimalware definitions are kept up-to-date to recognize the latest threats.

e) **Use a Firewall**: Implement a robust firewall solution to control and monitor network traffic both inbound and outbound. Configure the firewall to restrict access to only necessary ports and services, blocking any unauthorized access attempts.

File(s) with administrative credentials in plain text

Risk Rating: Critical

Description:

A severe security vulnerability has been identified within the network, stemming from the presence of administrative credentials stored in plain text format. During our assessment, we exploited this vulnerability using a malicious script within Metasploit, in combination with the previously discovered weak password. This security lapse allowed us to gain unauthorized access to a Linux machine within the network.

Once inside the compromised machine, a file was discovered containing administrative-level credentials that were stored in plain view, without any encryption or protection. This critical lapse in security enabled us to escalate our privileges within the Linux system, granting us extensive access to user files and system resources. Subsequently, we extracted password information from these files, which further exacerbated the security breach.

Affected Host(s): 172.22.117.150 - Linux machine

Remediation:

a) **Correct Weak Password Issue**: The immediate action is to rectify the weak password issue, as it was the initial point of exploitation. Enforce strong password policies throughout the organization, mandating complex passwords that are resistant to brute-force attacks.

b) **Secure Password Management**: Implement secure password management practices. Instead of storing passwords in plain text, encourage the use of password manager tools that securely store and encrypt sensitive credentials. This is especially crucial for staff members with high-level administrative privileges.

c) **Keep Software Updated**: Maintain a rigorous schedule for software updates and patches. Timely updates are essential to addressing vulnerabilities and maintaining the security of systems and applications.

d) **Use Advanced Antivirus Software**: Deploy advanced antivirus and endpoint protection software on all systems within the network. Regularly update antivirus definitions to ensure that the software can effectively detect and mitigate emerging threats.

e) **Implement a Firewall**: A firewall should be in place to monitor and control network traffic, both incoming and outgoing. Configure the firewall to block unauthorized access attempts and only allow traffic that is necessary for business operations.

Privilege Escalation

Risk Rating: **High**

Description:

Privilege escalation presents a critical security concern within our environment and is closely linked to the identified issue of weak passwords. While addressing weak password vulnerabilities is a crucial step in reducing risk, it is equally important to proactively prevent the escalation of privileges in the event unauthorized access is gained. Privilege escalation allows attackers to gain higher levels of access within a system, potentially leading to unauthorized control over critical resources and sensitive data.

Affected Host(s): 172.22.117.150 – Linux machine, 172.22.117.20 – Windows10 machine

Remediation:

a) **Enforce the Principle of Least Privilege**: Implement and strictly enforce the principle of least privilege (PoLP). This principle ensures that staff members are granted the minimum level of access and permissions necessary to perform their job functions. By limiting access rights, the impact of potential privilege escalation is significantly reduced.

b) **Patch and Update Systems**: Regularly apply security patches and updates to all systems and software within the network. These updates are essential to protect against known vulnerabilities and malicious content that could be exploited for privilege escalation.

c) **Vulnerability Scanning Tools**: Utilize robust vulnerability scanning tools to proactively identify and address potential security weaknesses. These tools can help identify vulnerabilities before they can be exploited by attackers, allowing for timely remediation.

d) **Monitoring and Logging**: Implement comprehensive monitoring and logging solutions to track user activities and system changes. This helps in the early detection of unauthorized access or suspicious behavior, enabling a rapid response to potential privilege escalation attempts.

e) **User Training and Awareness**: Provide ongoing cybersecurity training and awareness programs for staff members. Educate users about the risks associated with privilege escalation and the importance of adhering to security best practices.

f) **Access Control and Authentication**: Strengthen access control mechanisms and authentication procedures. Implement multi-factor authentication (MFA) wherever possible to add an extra layer of security when accessing sensitive systems and resources.

g) **Incident Response Plan**: Develop and regularly test an incident response plan that outlines procedures for detecting, containing, and mitigating privilege escalation incidents. Having a well-defined plan in place is crucial for swift and effective responses.

LLMNR

Risk Rating: High

Description:

LLMNR, which stands for Local Link Multicast Name Resolution, is an older broadcast protocol used as a local backup for DNS (Domain Name System). However, it presents a significant security risk within the network. Attackers can exploit LLMNR by intercepting and responding to LLMNR requests, effectively spoofing responses. This malicious activity can trick users into unknowingly providing their credentials to the attacker. Subsequently, these stolen credentials can be leveraged by the attacker to gain unauthorized access to the network.

During our assessment, we successfully simulated an LLMNR attack, highlighting the real-world danger this vulnerability poses. This simulation allowed us to obtain a new set of credentials that we did not previously possess, underscoring the potential for unauthorized access and data compromise.

Affected Host(s): 172.22.117.20 - Windows10 machine

Remediation:

a) **Turn Off LLMNR in Group Policy Editor**: Disable LLMNR at the network level by configuring Group Policy settings. This action prevents the use of LLMNR within the network, thereby eliminating the attack vector. Specific steps to disable LLMNR can be found in the Group Policy Editor settings.

b) **Monitor Network Traffic**: Implement continuous network traffic monitoring and intrusion detection systems to detect and respond to suspicious LLMNR-related activities. Anomalies in LLMNR traffic patterns should be investigated promptly to identify potential attacks or credential theft attempts.

c) **Enable DNSSEC**: DNS Security Extensions (DNSSEC) can be employed to enhance the security of DNS resolution. DNSSEC adds a layer of authentication and integrity verification to DNS queries and responses, making it more challenging for attackers to spoof DNS responses.

d) **User Awareness and Training**: Educate users about the risks associated with LLMNR attacks and the importance of vigilance when entering credentials. Training can help users recognize suspicious behavior and reduce the likelihood of falling victim to such attacks.

e) **Segmented Network Design**: Implement network segmentation to isolate critical assets and sensitive data from potentially compromised segments. This reduces the impact of an LLMNR attack by limiting an attacker's lateral movement within the network.

f) **Regular Vulnerability Scanning**: Conduct regular vulnerability scanning and penetration testing to identify and address potential weaknesses, including those related to LLMNR. Proactively identifying and mitigating vulnerabilities is essential to maintaining network security.

CVE Vulnerabilities

Risk Rating: Medium

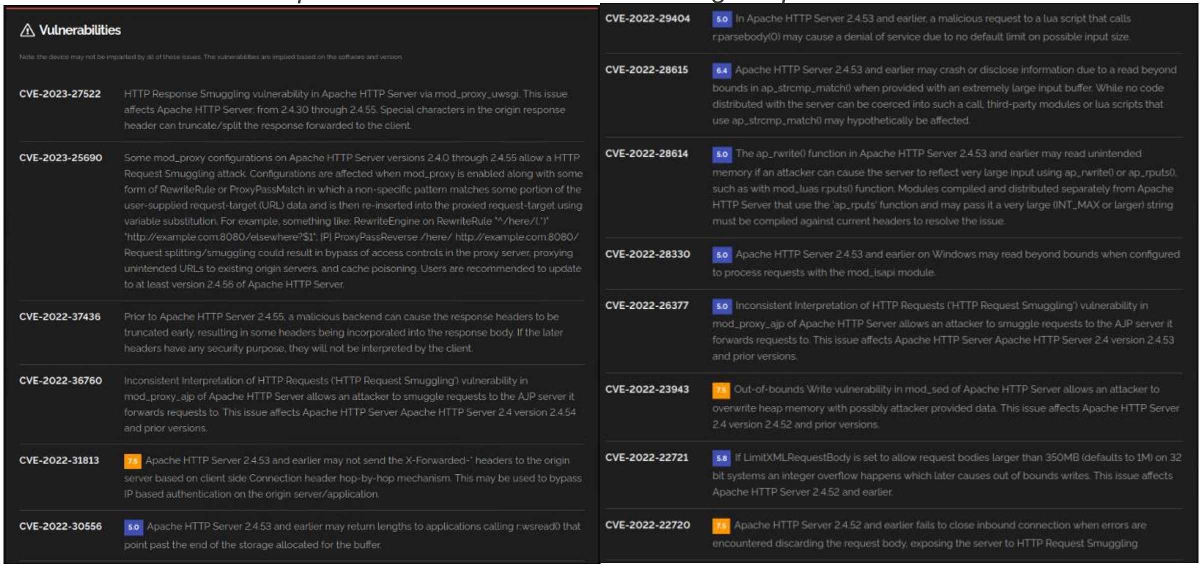
Description: A report was ran using Shodan, which identified the following potential vulnerabilities on Megacorpone’s apache servers: CVE-2023-27522, CVE-2023-25690, CVE-2022-37436, CVE-2022-36760, CVE-2022-31813, CVE-2022-30556, CVE-2022-29404, CVE-2022-28615, CVE-2022-28614, CVE-2022-28330, CVE-2022-26377, CVE-2022-23943, CVE-2022-22721, CVE-2022-22720, CVE-2022-22719, CVE-2021-44790, CVE-2021-44224, CVE-2021-40438, CVE-2021-39275

Affected Host(s): apache servers

Remediation:

- CVEs are publicly known security flaws. These potential vulnerabilities were not specifically tested for on your system, but it is recommended that you inquire about them further.
    - Details about these vulnerabilities can be found at: https://cve.mitre.org/cve/search_cve_list.html

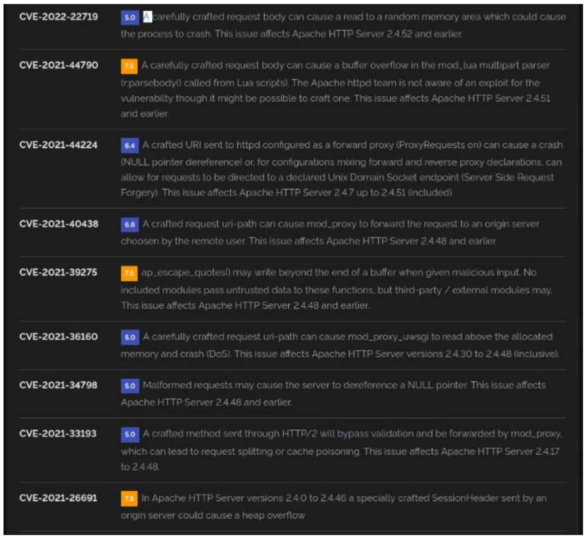
Shodan.io CVE report for 149.56.244.87 – www.megacorpone.com



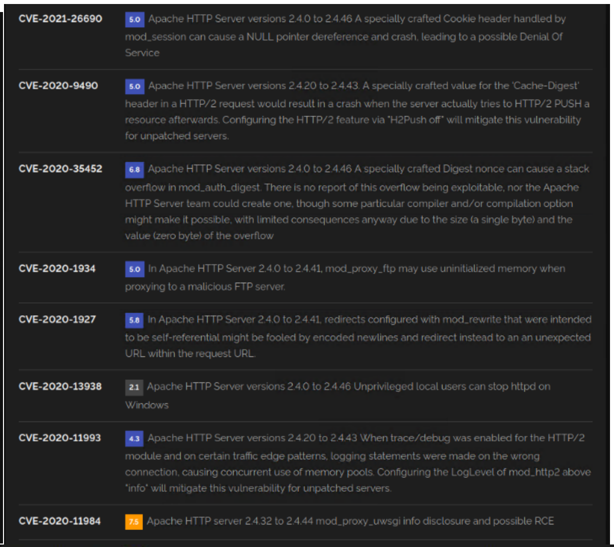
⚠ Vulnerabilities

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

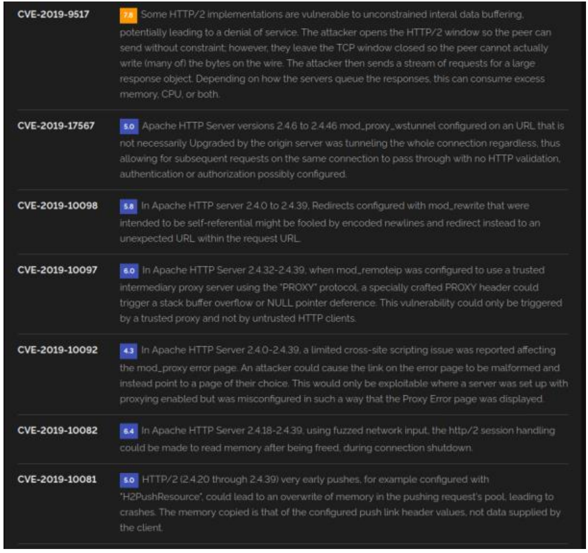
CVE-2023-27522	HTTP Response Smuggling vulnerability in Apache HTTP Server via mod_proxy_uwsgi. This issue affects Apache HTTP Server: from 2.4.30 through 2.4.55. Special characters in the origin response header can truncate/split the response forwarded to the client.
CVE-2023-25690	Some mod_proxy configurations on Apache HTTP Server versions 2.4.0 through 2.4.55 allow a HTTP Request Smuggling attack. Configurations are affected when mod_proxy is enabled along with some form of RewriteRule or ProxyPassMatch in which a non-specific pattern matches some portion of the user-supplied request-target (URL) data and is then re-inserted into the proxied request-target using variable substitution. For example, something like: RewriteEngine on RewriteRule "^/here/(.*)" "http://example.com:8080/elsewhere?$1"; [P] ProxyPassReverse /here/ http://example.com:8080/ Request splitting/smuggling could result in bypass of access controls in the proxy server, proxying unintended URLs to existing origin servers, and cache poisoning. Users are recommended to update to at least version 2.4.56 of Apache HTTP Server.
CVE-2022-37436	Prior to Apache HTTP Server 2.4.55, a malicious backend can cause the response headers to be truncated early, resulting in some headers being incorporated into the response body. If the later headers have any security purpose, they will not be interpreted by the client.
CVE-2022-36760	Inconsistent Interpretation of HTTP Requests ('HTTP Request Smuggling') vulnerability in mod_proxy_ajp of Apache HTTP Server allows an attacker to smuggle requests to the AJP server it forwards requests to. This issue affects Apache HTTP Server Apache HTTP Server 2.4 version 2.4.54 and prior versions.
CVE-2022-31813	7.5 Apache HTTP Server 2.4.53 and earlier may not send the X-Forwarded-* headers to the origin server based on client side Connection header hop-by-hop mechanism. This may be used to bypass IP based authentication on the origin server/application.
CVE-2022-30556	5.0 Apache HTTP Server 2.4.53 and earlier may return lengths to applications calling r:wsread() that point past the end of the storage allocated for the buffer.
CVE-2022-29404	5.0 In Apache HTTP Server 2.4.53 and earlier, a malicious request to a lua script that calls r:parsebody(0) may cause a denial of service due to no default limit on possible input size.
CVE-2022-28615	6.4 Apache HTTP Server 2.4.53 and earlier may crash or disclose information due to a read beyond bounds in ap_strcmp_match() when provided with an extremely large input buffer. While no code distributed with the server can be coerced into such a call, third-party modules or lua scripts that use ap_strcmp_match() may hypothetically be affected.
CVE-2022-28614	5.0 The ap_rwrite() function in Apache HTTP Server 2.4.53 and earlier may read unintended memory if an attacker can cause the server to reflect very large input using ap_rwrite() or ap_rputs(), such as with mod_luas r:puts() function. Modules compiled and distributed separately from Apache HTTP Server that use the 'ap_rputs' function and may pass it a very large (INT_MAX or larger) string must be compiled against current headers to resolve the issue.
CVE-2022-28330	5.0 Apache HTTP Server 2.4.53 and earlier on Windows may read beyond bounds when configured to process requests with the mod_isapi module.
CVE-2022-26377	5.0 Inconsistent Interpretation of HTTP Requests ('HTTP Request Smuggling') vulnerability in mod_proxy_ajp of Apache HTTP Server allows an attacker to smuggle requests to the AJP server it forwards requests to. This issue affects Apache HTTP Server Apache HTTP Server 2.4 version 2.4.53 and prior versions.
CVE-2022-23943	7.5 Out-of-bounds Write vulnerability in mod_sed of Apache HTTP Server allows an attacker to overwrite heap memory with possibly attacker provided data. This issue affects Apache HTTP Server 2.4 version 2.4.52 and prior versions.
CVE-2022-22721	5.8 If LimitXMLRequestBody is set to allow request bodies larger than 350MB (defaults to 1M) on 32 bit systems an integer overflow happens which later causes out of bounds writes. This issue affects Apache HTTP Server 2.4.52 and earlier.
CVE-2022-22720	7.5 Apache HTTP Server 2.4.52 and earlier fails to close inbound connection when errors are encountered discarding the request body, exposing the server to HTTP Request Smuggling



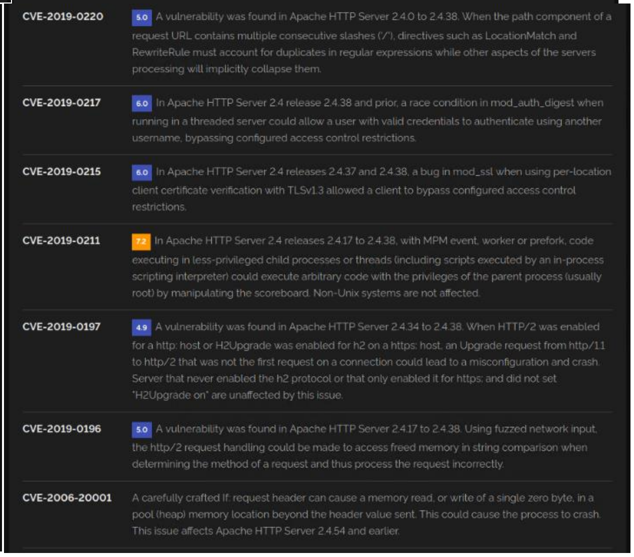
CVE-2022-22719	5.0	A carefully crafted request body can cause a read to a random memory area which could cause the process to crash. This issue affects Apache HTTP Server 2.4.52 and earlier.
CVE-2021-44790	7.5	A carefully crafted request body can cause a buffer overflow in the mod_lua multipart parser (r:parsebody() called from Lua scripts). The Apache httpd team is not aware of an exploit for the vulnerabilty though it might be possible to craft one. This issue affects Apache HTTP Server 2.4.51 and earlier.
CVE-2021-44224	6.4	A crafted URI sent to httpd configured as a forward proxy (ProxyRequests on) can cause a crash (NULL pointer dereference) or, for configurations mixing forward and reverse proxy declarations, can allow for requests to be directed to a declared Unix Domain Socket endpoint (Server Side Request Forgery). This issue affects Apache HTTP Server 2.4.7 up to 2.4.51 (included).
CVE-2021-40438	6.8	A crafted request uri-path can cause mod_proxy to forward the request to an origin server choosen by the remote user. This issue affects Apache HTTP Server 2.4.48 and earlier.
CVE-2021-39275	7.5	ap_escape_quotes() may write beyond the end of a buffer when given malicious input. No included modules pass untrusted data to these functions, but third-party / external modules may. This issue affects Apache HTTP Server 2.4.48 and earlier.
CVE-2021-36160	5.0	A carefully crafted request uri-path can cause mod_proxy_uwsgi to read above the allocated memory and crash (DoS). This issue affects Apache HTTP Server versions 2.4.30 to 2.4.48 (inclusive).
CVE-2021-34798	5.0	Malformed requests may cause the server to dereference a NULL pointer. This issue affects Apache HTTP Server 2.4.48 and earlier.
CVE-2021-33193	5.0	A crafted method sent through HTTP/2 will bypass validation and be forwarded by mod_proxy, which can lead to request splitting or cache poisoning. This issue affects Apache HTTP Server 2.4.17 to 2.4.48.
CVE-2021-26691	7.5	In Apache HTTP Server versions 2.4.0 to 2.4.46 a specially crafted SessionHeader sent by an origin server could cause a heap overflow



CVE-2021-26690	5.0	Apache HTTP Server versions 2.4.0 to 2.4.46 A specially crafted Cookie header handled by mod_session can cause a NULL pointer dereference and crash, leading to a possible Denial Of Service
CVE-2020-9490	5.0	Apache HTTP Server versions 2.4.20 to 2.4.43. A specially crafted value for the 'Cache-Digest' header in a HTTP/2 request would result in a crash when the server actually tries to HTTP/2 PUSH a resource afterwards. Configuring the HTTP/2 feature via "H2Push off" will mitigate this vulnerability for unpatched servers.
CVE-2020-35452	6.8	Apache HTTP Server versions 2.4.0 to 2.4.46 A specially crafted Digest nonce can cause a stack overflow in mod_auth_digest. There is no report of this overflow being exploitable, nor the Apache HTTP Server team could create one, though some particular compiler and/or compilation option might make it possible, with limited consequences anyway due to the size (a single byte) and the value (zero byte) of the overflow
CVE-2020-1934	5.0	In Apache HTTP Server 2.4.0 to 2.4.41, mod_proxy_ftp may use uninitialized memory when proxying to a malicious FTP server.
CVE-2020-1927	5.8	In Apache HTTP Server 2.4.0 to 2.4.41, redirects configured with mod_rewrite that were intended to be self-referential might be fooled by encoded newlines and redirect instead to an an unexpected URL within the request URL.
CVE-2020-13938	2.1	Apache HTTP Server versions 2.4.0 to 2.4.46 Unprivileged local users can stop httpd on Windows
CVE-2020-11993	4.3	Apache HTTP Server versions 2.4.20 to 2.4.43 When trace/debug was enabled for the HTTP/2 module and on certain traffic edge patterns, logging statements were made on the wrong connection, causing concurrent use of memory pools. Configuring the LogLevel of mod_http2 above "info" will mitigate this vulnerability for unpatched servers.
CVE-2020-11984	7.5	Apache HTTP server 2.4.32 to 2.4.44 mod_proxy_uwsgi info disclosure and possible RCE



CVE-2019-9517	7.8	Some HTTP/2 implementations are vulnerable to unconstrained interal data buffering, potentially leading to a denial of service. The attacker opens the HTTP/2 window so the peer can send without constraint; however, they leave the TCP window closed so the peer cannot actually write (many of) the bytes on the wire. The attacker then sends a stream of requests for a large response object. Depending on how the servers queue the responses, this can consume excess memory, CPU, or both.
CVE-2019-17567	5.0	Apache HTTP Server versions 2.4.6 to 2.4.46 mod_proxy_wstunnel configured on an URL that is not necessarily Upgraded by the origin server was tunneling the whole connection regardless, thus allowing for subsequent requests on the same connection to pass through with no HTTP validation, authentication or authorization possibly configured.
CVE-2019-10098	5.8	In Apache HTTP server 2.4.0 to 2.4.39, Redirects configured with mod_rewrite that were intended to be self-referential might be fooled by encoded newlines and redirect instead to an unexpected URL within the request URL.
CVE-2019-10097	6.0	In Apache HTTP Server 2.4.32-2.4.39, when mod_remoteip was configured to use a trusted intermediary proxy server using the "PROXY" protocol, a specially crafted PROXY header could trigger a stack buffer overflow or NULL pointer deference. This vulnerability could only be triggered by a trusted proxy and not by untrusted HTTP clients.
CVE-2019-10092	4.3	In Apache HTTP Server 2.4.0-2.4.39, a limited cross-site scripting issue was reported affecting the mod_proxy error page. An attacker could cause the link on the error page to be malformed and instead point to a page of their choice. This would only be exploitable where a server was set up with proxying enabled but was misconfigured in such a way that the Proxy Error page was displayed.
CVE-2019-10082	6.4	In Apache HTTP Server 2.4.18-2.4.39, using fuzzed network input, the http/2 session handling could be made to read memory after being freed, during connection shutdown.
CVE-2019-10081	5.0	HTTP/2 (2.4.20 through 2.4.39) very early pushes, for example configured with "H2PushResource", could lead to an overwrite of memory in the pushing request's pool, leading to crashes. The memory copied is that of the configured push link header values, not data supplied by the client.



CVE-2019-0220	5.0	A vulnerability was found in Apache HTTP Server 2.4.0 to 2.4.38. When the path component of a request URL contains multiple consecutive slashes ('/'), directives such as LocationMatch and RewriteRule must account for duplicates in regular expressions while other aspects of the servers processing will implicitly collapse them.
CVE-2019-0217	6.0	In Apache HTTP Server 2.4 release 2.4.38 and prior, a race condition in mod_auth_digest when running in a threaded server could allow a user with valid credentials to authenticate using another username, bypassing configured access control restrictions.
CVE-2019-0215	6.0	In Apache HTTP Server 2.4 releases 2.4.37 and 2.4.38, a bug in mod_ssl when using per-location client certificate verification with TLSv1.3 allowed a client to bypass configured access control restrictions.
CVE-2019-0211	7.2	In Apache HTTP Server 2.4 releases 2.4.17 to 2.4.38, with MPM event, worker or prefork, code executing in less-privileged child processes or threads (including scripts executed by an in-process scripting interpreter) could execute arbitrary code with the privileges of the parent process (usually root) by manipulating the scoreboard. Non-Unix systems are not affected.
CVE-2019-0197	4.9	A vulnerability was found in Apache HTTP Server 2.4.34 to 2.4.38. When HTTP/2 was enabled for a http: host or H2Upgrade was enabled for h2 on a https: host, an Upgrade request from http/1.1 to http/2 that was not the first request on a connection could lead to a misconfiguration and crash. Server that never enabled the h2 protocol or that only enabled it for https: and did not set "H2Upgrade on" are unaffected by this issue.
CVE-2019-0196	5.0	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.
CVE-2006-20001		A carefully crafted If: request header can cause a memory read, or write of a single zero byte, in a pool (heap) memory location beyond the header value sent. This could cause the process to crash. This issue affects Apache HTTP Server 2.4.54 and earlier.

Exposed IP addresses for three of Megacorpone's domain servers

Risk Rating: Medium

Description:

An investigation using Recon-ng, a publicly available reconnaissance tool, has revealed that the IP addresses of Megacorpone's three named servers (NS) are exposed to the public domain. This exposure poses a potential security risk for the organization. Given the accessibility of Recon-ng to both legitimate users and malicious actors, there is a concern that bad actors could exploit this information to engage in DNS (Domain Name System) poisoning or spoofing attacks.

DNS poisoning or spoofing attacks involve manipulating the DNS resolution process, redirecting users to malicious websites instead of the intended legitimate sites. In the context of Megacorpone, this could result in users being directed away from the company's services and toward potentially harmful or fraudulent sites, posing a threat to user trust and information security.

Affected Host(s): ns1.megacorpone.com, ns2.megacorpone.com, ns3.megacorpone.com

Remediation:

a) **Make IP Addresses Private**: The most effective and recommended approach is to make the IP addresses of the domain servers private. By restricting access to these IP addresses, you limit the exposure of sensitive network information to potential attackers. Ensure that access controls are properly configured to prevent unauthorized access to the servers.

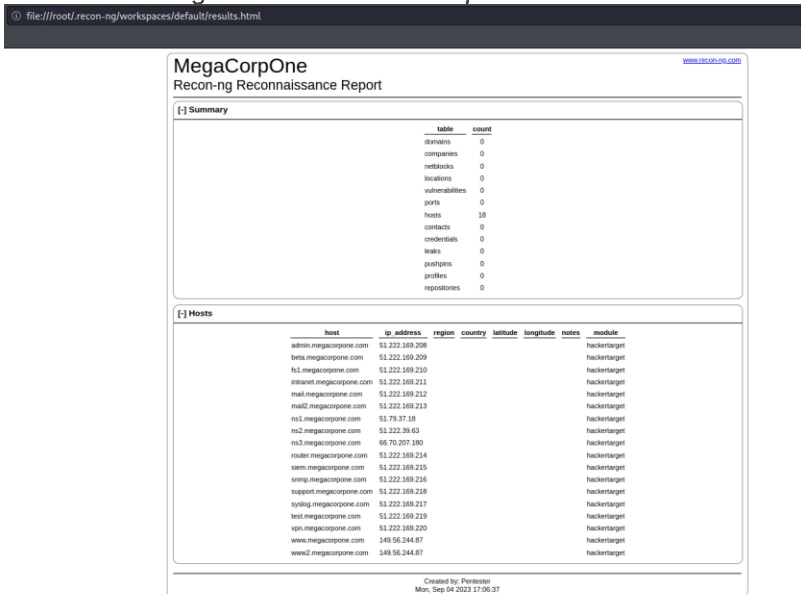
b) **Regularly Update and Secure Servers**: If you choose to keep the IP addresses public for legitimate reasons, it is imperative to maintain these servers with up-to-date security patches and strong firewall protections. Regularly monitoring and enhancing the security posture of these servers is essential to reduce the risk of exploitation.

c) **Implement DNS Security Practices**: Employ DNS security best practices, such as DNSSEC (Domain Name System Security Extensions), to enhance the security of your DNS infrastructure. DNSSEC adds a layer of authentication and data integrity verification to DNS responses, making it more challenging for attackers to manipulate DNS records.

d) **Regular Security Audits**: Conduct regular security audits and vulnerability assessments to identify and address potential weaknesses in your DNS infrastructure. This proactive approach helps ensure the ongoing security of your domain servers.

e) **User Education**: Educate users and IT staff about the risks associated with exposed IP addresses and the potential consequences of DNS attacks. Encourage vigilance and prompt reporting of any suspicious DNS-related activities.

MegaCorpOne’s Recon-ng Reconnaissance Report



MegaCorpOne

Recon-ng Reconnaissance Report

[-] Summary

table	count
domains	0
companies	0
netblocks	0
locations	0
vulnerabilities	0
ports	0
hosts	18
contacts	0
credentials	0
leaks	0
pushpins	0
profiles	0
repositories	0

[-] Hosts

host	ip_address	region	country	latitude	longitude	notes	module
admin.megacorpone.com	51.222.169.208						hackertarget
beta.megacorpone.com	51.222.169.209						hackertarget
fs1.megacorpone.com	51.222.169.210						hackertarget
intranet.megacorpone.com	51.222.169.211						hackertarget
mail.megacorpone.com	51.222.169.212						hackertarget
mail2.megacorpone.com	51.222.169.213						hackertarget
ns1.megacorpone.com	51.79.37.18						hackertarget
ns2.megacorpone.com	51.222.39.63						hackertarget
ns3.megacorpone.com	66.70.207.180						hackertarget
router.megacorpone.com	51.222.169.214						hackertarget
siem.megacorpone.com	51.222.169.215						hackertarget
snmp.megacorpone.com	51.222.169.216						hackertarget
support.megacorpone.com	51.222.169.218						hackertarget
syslog.megacorpone.com	51.222.169.217						hackertarget
test.megacorpone.com	51.222.169.219						hackertarget
vpn.megacorpone.com	51.222.169.220						hackertarget
www.megacorpone.com	149.56.244.87						hackertarget
www2.megacorpone.com	149.56.244.87						hackertarget

Created by: Pentester
Mon, Sep 04 2023 17:06:37

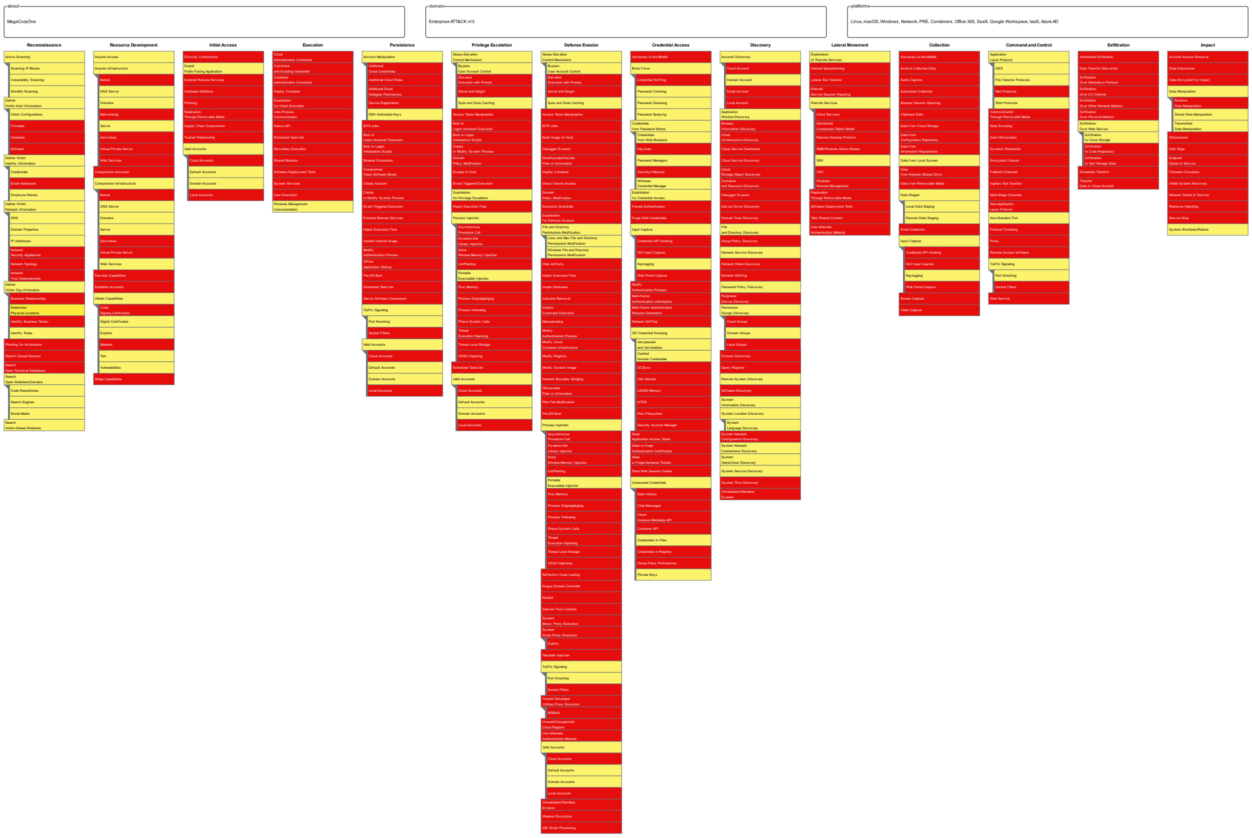
MITRE ATT&CK Navigator Map

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

Legend:

🟨 Performed successfully
🟥 Failure to perform

Reference image of the Navigator map, for optimal viewing use the proceeding link(s)



This image of the MITRE ATT&CK navigator map is for general reference only. Due to the size of this map the image shown does not always display at full resolution. For optimal viewing, the files have been attached separately, as .json and .svg format.

Use the following link(s) to access the file(s) directly:

- .json:
  - https://drive.google.com/file/d/1yYP0XbVqQKtEZJJ9yuJtxVm2QwtCqboP/view?usp=drive_link
- .svg:
  - https://drive.google.com/file/d/1l7fvR2jph9G6270CuXJWZkp9HF6u0OnG/view?usp=drive_link