{ no logo }

coming soon

Penetration Test Report

Conducted by:

New Haven Hacking Inc.

Team Members	*Email*
Samuel Zurowski	szuro1@unh.newhaven.edu
Charles Barone	cbaro4@unh.newhaven.edu

March 3, 2022

Table of Contents

1 Report Overview **2**
1.1 Executive Summary . . . 2
1.2 Engagement Overview . . . 3
1.3 Scope of Engagement . . . 3

2 Observations **4**
2.1 Summary of Recommendations . . . 5
2.2 Positive Security Measures . . . 5
2.3 Compliance . . . 6
2.3.1 PCI DSS Violations . . . 6

3 Testing Methodology **7**
3.1 Penetration Testing Execution Standard . . . 7
3.2 MITRE ATT&CK Framework . . . 7
3.3 OWASP Top 10 . . . 7
3.4 PCI DSS Auditing . . . 7
3.5 NIST SP 800-53 . . . 8

4 Technical Findings **9**
4.1 Critical Risk . . . 10
4.1.1 Lack Of PostgreSQL Authentication . . . 10
4.2 High Risk . . . 12
4.2.1 Lack of MariaDB Authentication . . . 12
4.3 Moderate Risk . . . 15
4.3.1 Payment Transaction Enumeration . . . 15
4.4 Low Risk . . . 17
4.4.1 ScadaBR Reflected XSS (Username) . . . 17
4.5 Informational . . . 19
4.5.1 Insufficient Firewalls . . . 19

5 Conclusion **20**

Appendices **22**

A Network Topology **22**

B Tools **23**

1 Report Overview

1.1 Executive Summary

New Haven Hacking Inc. was contacted by An Example Company (AEC) for a penetration test in order to identify security issues within their infrastructure. This report was written initially on January 7th and submitted on January 9th at 1:00AM. This penetration test is in the interest of AEC, as part of a restrained scope penetration test and risk assessment. The Report Overview section contains an outlined summary of New Haven Hacking Inc.'s findings, including recommendations for improving AEC's security, mitigating potential business risk, and reducing attack surface. The Technical Findings section expands upon the report overview by including each discovered vulnerability's evaluated risk, exploitation details, and recommended remediation steps.

Based upon the results of the assessment, AEC is at risk to be fined by payment providers due to severe PCI DSS violations. These fines could range from $5,000 to $100,000 per month depending on factors such as size of business [1]. Based on these issues, we suggest spending resources to become complaint. Details of all PCI DSS violations can be found in Section 2.3.1.

New Haven Hacking Inc. was able to gain full access to a SCADA system using the default credentials. This device is extremely important as it is critical to industrial systems which operate the storage, delivery, and packaging warehouse facility. Similarly, the industrial control device is not isolated in any way, could be manipulated by any user on the network. It is important to take this with extreme caution, as malicious tampering with these devices could result in loss of life. This would result in unwanted attention, could harm the companies reputation, and could cost AECvast sums of money from both lawsuits and long term loss of business.

During this engagement, a total of **11** vulnerabilities were found in AEC's network. In terms of severity, **3** vulnerabilities are critical, **3** vulnerabilities are high, **1** vulnerability is moderate, and **4** vulnerabilities posed low risk. Many of the vulnerabilities in the environment are due to improper authentication, default credentials, or not applying principles of least privileges. More information about these vulnerabilities can be found in Section 4.

1.2 Engagement Overview

New Haven Hacking Inc. conducted a penetration test starting on January 7th, 2022 based upon the Request for Proposal (RFP) document obtained by New Haven Hacking Inc.. Focus was placed on the following goals during the engagement:

- Assessing internally developed and customized software packages.
- Assessment of Industrial controls within a storage, delivery, and packing warehouse facility.
- Assessment of APIs related to payment, transaction, billing, and inventory processing systems.
- 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.
- Evaluate AEC's security posture against the Payment Card Industry - Digital Security Standard (PCI-DSS).

1.3 Scope of Engagement

The full scope of this penetration test was limited to the following CIDR range. At the request of AEC , the first day of the engagement (Jan 7 2022) excluded hosts 10.0.17.50 and 10.0.17.51 in order to ensure availability of critical infrastructure. These hosts were included on the second day (Jan 8th starting at 9:20AM) to ensure proper testing. Care was taken by New Haven Hacking Inc. to ensure penetration test activity did not reduce the availability of industrial systems.

- 10.0.17.0/24

The penetration test was conducted with extreme care to ensure actions were contained within the defined scope. Additionally, because the engagement was within a production environment, the team ensured that no services were disrupted. New Haven Hacking Inc. did not exfiltrate, modify, or delete any data not included in this report.

New Haven Hacking Inc. is available upon request to improve the security, protect the employees, and customers of AEC. This includes verifying and validating implemented mitigation techniques as well as deploying security strategies to ensure AEC has several layers of defense. The team is happy to continue a partnership with AEC and excited to work along side them in securing their operations.

2 Observations

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 4. It is important to note that this list is by no means exhaustive and that there are most likely vulnerabilities that New Haven Hacking Inc. did not find.

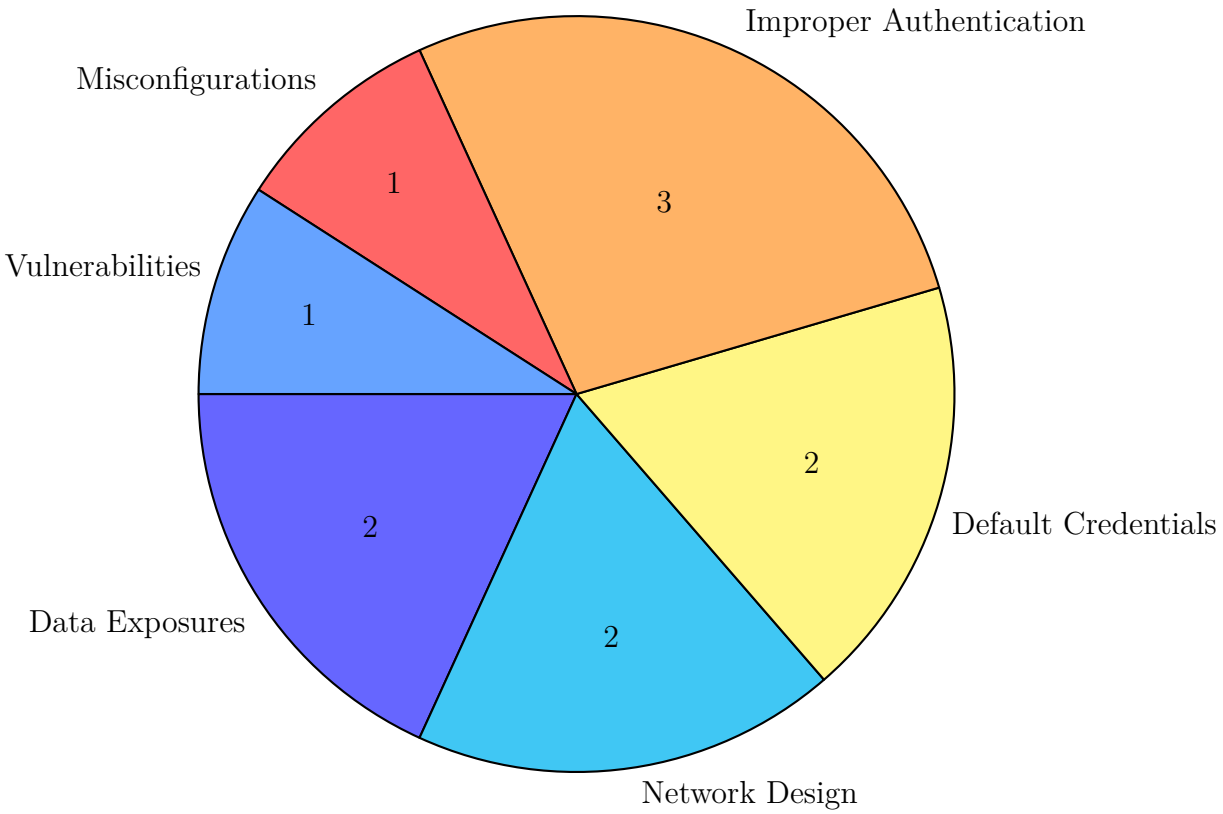

Figure 1: Summary of Issues within the Network

Default Credentials & Lack of authentication

The most immediate observation about AEC's security posture is that default, null, and passwordless authentication was discovered on multiple systems. New Haven Hacking Inc. was able to gain access into a SCADA system using the same credentials (which were default) as the last engagement. The system in question is critical to the storage, delivery, and packaging processes within the warehouse facility. Moreover, several databases contained this same issue and were found to not be requiring password authentication. It is important to remember these credentials are for critical services and PII. These weaknesses can have an enormous impact on AEC's ability to operate if discovered by threat actors. These vulnerabilities can be remediated with low cost and have an outsized impact on the security of AEC. More details on mitigation for vulnerabilities such as these can be found in each vulnerability's remediation suggestions in Section 4.

2.1 Summary of Recommendations

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

- Resolve any PCI DSS compliance violations in Section 2.3.1 and ensure that AEC is currently meeting all documentation requirements set by PCI DSS.
- Implement both ingress & egress filtering to reduce attack surface on hosts.
- Ensure all hosts use least privilege principals to reduce attack surface.
- Ensure proper encryption is used for confidential data (e.g. passwords and card holder data).
- Implementation of a strong password policy.
- Implement Multi-Factor authentication to provide defense in depth in addition to passwords.
- Uses centralized logging to be able to respond to potential incidents faster.
- Ensure null or password-less authentication is not allowed.
- Ensure only necessary services are running within the subnet.

2.2 Positive Security Measures

As the engagement progressed, New Haven Hacking Inc. was impeded by the security safeguards AEC had in place. A number of basic security best practices were observed that limited New Haven Hacking Inc.'s ability to move through the network. Some instances of aforementioned security practices implemented by AEC include:

- The usage of TLS on websites to protect information.
- The usage of Cross-Origin Resource Sharing (CORS) headers prevented specific attacks.
- The marketplace & music player required authentication.
- Some APIs required authentication in order to query sensitive information (although not all).

These controls should be continuously monitored and regulated to maintain the company's security posture.

2.3 Compliance

2.3.1 PCI DSS Violations

The following table details violations of the PCI Data Security Standard discovered by New Haven Hacking Inc. over the scope of this engagement.

Table 1: PCI DSS Compliance Violations.

Regulation	Reason	Reference
PCI DSS 1.1.4	A firewall is not implemented at every internet connection.	Section 4.5.1
PCI DSS 1.2	The PostgreSQL server that stores cardholder data does not have a firewall restricting connections from untrusted networks.	Section 4.1.1
PCI DSS 1.3	The PostgreSQL server is not segmented through the use of a DMZ.	Section 4.1.1
PCI DSS 2.1	Default accounts were not removed from all systems on the network.	Sections 4.1.1
PCI DSS 2.2	No evidence found to indicate that AEC has configuration standards.	N/A
PCI DSS 2.2.1	The server "charley" had more than one primary function implemented: MariaDB and PostgreSQL.	Sections 4.1.1 & 4.2.1
PCI DSS 2.2.2	No evidence of documentation for enabled insecure services, daemons, or protocols.	N/A
PCI DSS 2.2.4	There were insecure security parameters present on system configurations. No found evidence of documented common system security parameters and no evidence of system configuration standards.	Sections 4.1.1
PCI DSS 2.4	An inventory of what was considered to be in scope for PCI DSS was not provided to penetration testers nor found during the engagement.	N/A
PCI DSS 2.5	No evidence of security policies and operational procedures was found during the engagement	N/A

3 Testing Methodology

3.1 Penetration Testing Execution Standard

Throughout the engagement New Haven Hacking Inc., references the Penetration Testing Execution Standard (PTES) when conducting security assessments [2].

Figure 2: PTES Methodology

3.2 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. New Haven Hacking Inc. 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 [3].

3.3 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 [4]. OWASP Top 10 focuses vulnerabilities focus on common vulnerabilities that pose security risks to web applications:

Table 2: 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

3.4 PCI DSS Auditing

One of the requests of AEC was for experience in PCI DSS from the RFP. Throughout the engagement, New Haven Hacking Inc. audited PCI DSS compliance in accordance with the proper Self-Assessment Questionnaire (SAQ). By doing this, New Haven Hacking Inc. can ensure which PCI DSS security requirements are met in accordance with the proper SAQ for

the type(s) of transactions being performed by AEC. Each PCI DSS compliance failure can be found in Section 2.3.1.

3.5 NIST SP 800-53

NIST SP 800-53 is the National Institute of Standards and Technology Special Publication 800-53, Security and Privacy Controls for Federal Information Systems and Organization. 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 3 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 [5].

Table 3: 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

4 Technical Findings

This table shows the total number of vulnerabilities found during the penetration test engagement. The vulnerabilities are categorized based on the risk level. The risk levels were calculated using the Common Vulnerability Scoring System (CVSS) [6].

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	1	1	1	1

The following table breaks down the discovered vulnerabilities by overall risk score, impact, and exploitability. The scores were calculated using NIST's CVSS v3.1 calculator [7].

Summary of Vulnerabilities by Base Score

Risk Summary	Overall Risk Score	Impact	Exploitability
Lack of PostgreSQL Authentication	9	8	10
Lack of MariaDB Authentication	7.5	5	10
Payment Transaction Enumeration	4.5	4	5
ScadaBR Reflected XSS (Username)	1.25	1	1.5

4.1 Critical Risk

4.1.1 Lack Of PostgreSQL Authentication

Threat Level: Critical (9.5)

Description:

The host Charley on the network did not require password authentication for the postgres user in PostgreSQL. As a result, attackers can access all databases on charley and enumerate data found. The postgres user has full control over the database within the host.

```
┌──(root㉿kali01)-[/tmp]
└─# psql -U postgres -p 5432 -h 10.0.17.14
psql (14.1 (Debian 14.1-1), server 12.9 (Ubuntu 12.9-0ubuntu0.20.04.1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.

postgres=# \l
                                  List of databases
    Name    |  Owner   | Encoding |   Collate   |  Ctype  |   Access privileges
------------+----------+----------+-------------+---------+-----------------------
 jawbreaker | postgres | UTF8     | en_US.UTF-8 | C.UTF-8 |
 postgres   | postgres | UTF8     | C.UTF-8     | C.UTF-8 |
 template0  | postgres | UTF8     | C.UTF-8     | C.UTF-8 | =c/postgres          +
            |          |          |             |         | postgres=CTc/postgres
 template1  | postgres | UTF8     | C.UTF-8     | C.UTF-8 | =c/postgres          +
            |          |          |             |         | postgres=CTc/postgres
(4 rows)
```

Figure 3: User postgres does not require a password to authenticate.

Potential Business Impact:

The data stored within this database contained unencrypted database information which is a direction violation of PCI DSS more information information can be found in Section 2.3.1. Failures of PCI DSS can result in fines and other punishments. Each security incidents and breaches can result of a $500,000 fine [8]. Figure 4 shows that credit card information was stored encrypted.

Affected Host:

Eggdicator (10.0.17.10)

Scrumdiddlyumptious (10.0.17.12)

Charley (10.0.17.14)

Exploitation Details:

A user who can connect to 10.0.17.14 can connect to the postgresql server by running the following command:

```
psql –U postgres –p 5432 –h 10.0.17.14
```

Recommended Remediation:

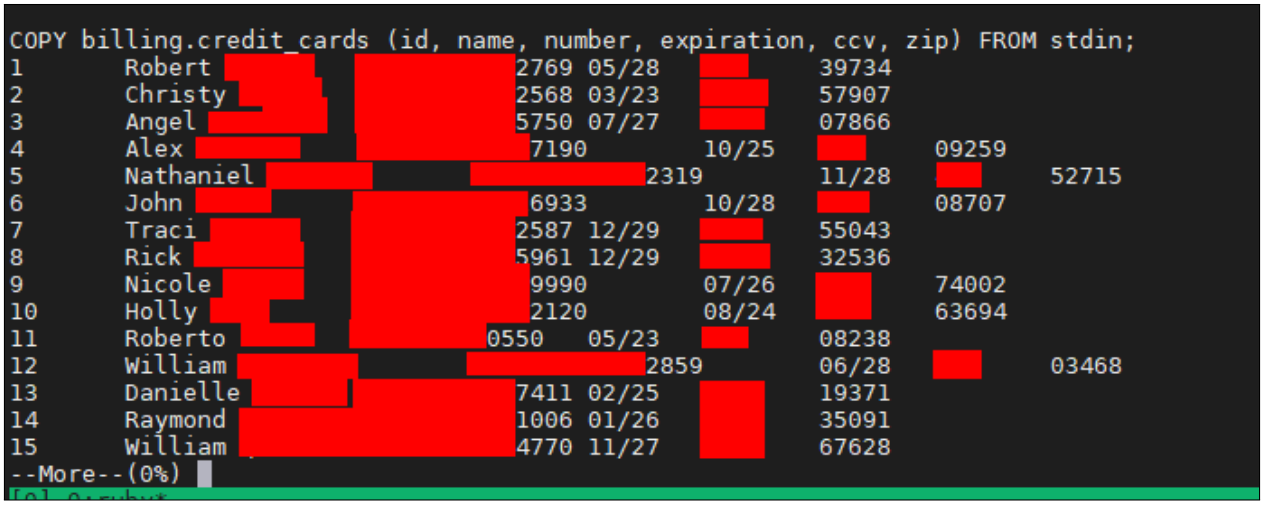

Figure 4: PostgreSQL Billing Table stored in the clear.

Harden the PostgreSQL server to require password authentication. Additionally, having firewall access controls to restrict what respective IP addresses can access the database would provide an additional layer of security.

```
ALTER USER postgres PASSWORD 'B3tt3rP@ssw0rd';
```

Additionally, the PostgreSQL instance could be further hardened by making rules in the *pg_hba.conf* file to only allow for authentication from certain hosts. More information about this configuration file can be found in the references for this section.

References:

`https://www.postgresql.org/docs/13/auth-password.html`

`https://www.postgresql.org/docs/9.2/auth-pg-hba-conf.html`

4.2 High Risk

4.2.1 Lack of MariaDB Authentication

Threat Level: High (7.5)

Description:

Unauthenticated access to a MySQL database permits access/modification to sensitive datasets, including the following:

- Customer Accounts and passwords (Base64 encoded).
- Customer PII - includes phone numbers, address, and payments (including amounts).
- Invoices and payments.
- Creation of administrator accounts for AEC's croissant marketplace.
- Insertion, deletion, and modification of all data within the database.

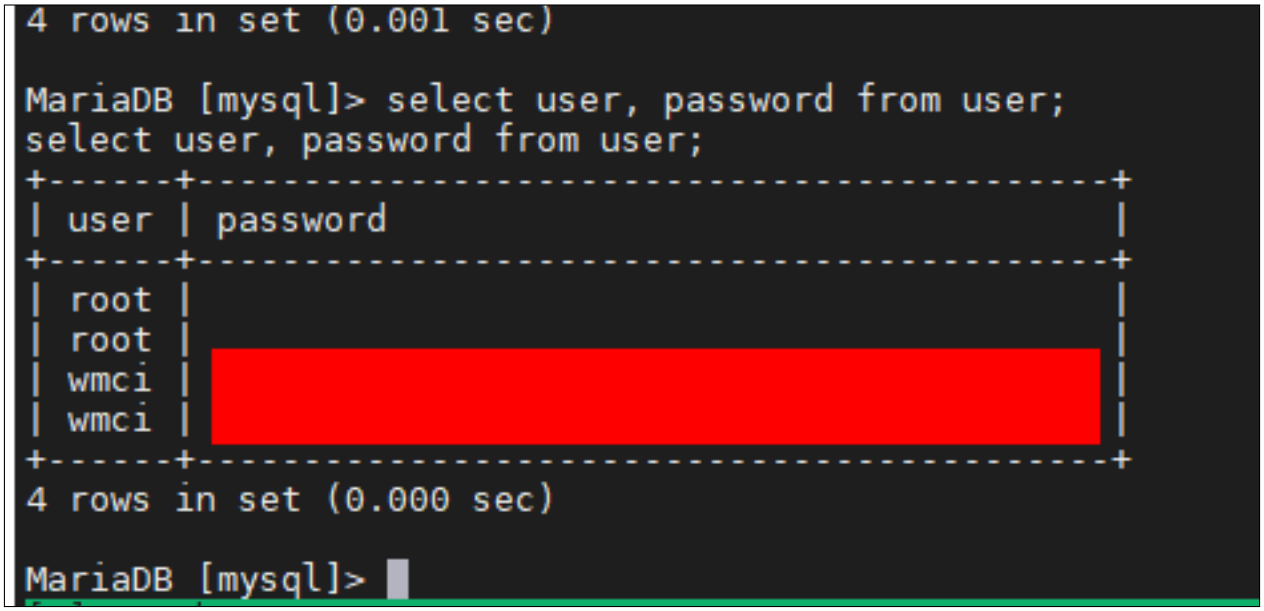

Figure 5: Passwordless Root MariaDB Access

Potential Business Impact:

This host (Charley) can severely impact the Confidentiality, Integrity, and Availability (CIA) of transactions within the warehouse management systems. Improper encoding schemes result in an environment in which all of the AEC store website (Scrumdiddlyumptious) users' passwords can be decoded from base64.

All accounts can also be modified to granted administrator level access on the AEC store. As data is parsed through the root user, transactions, account details, items (for sale), and other information which is integral to AEC's ability to sell its products online may be

subject to unauthorized modification. This data is not directly accessed by the web store host (Scrumdiddlyumptious), but rather through an API endpoint host (Whatchamacallit).

Affected Hosts:

Charley (10.0.17.14)

Scrumdiddlyumptious (10.0.17.12)

Whatchamacallit (10.0.17.13)

Exploitation Details:

New Haven Hacking Inc. used the MySQL command line application to query the unauthenticated database as the root user. All customer information could be obtained without modifying the database schema or contents. This user also had write permissions, meaning a malicious actor could arbitrarily modify database contents.

Figure 6 shows the connection from the PostgreSQL server and initial shell commands.

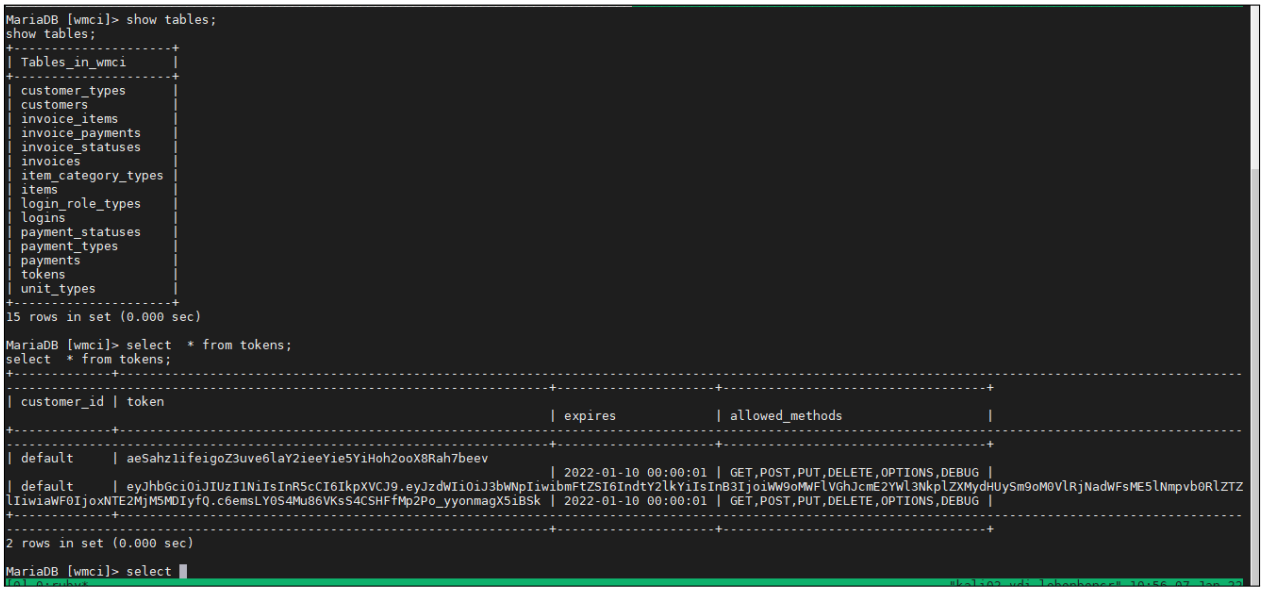

Figure 6: MariaDB tokens Table Dump

Recommended Remediation:

A minimal number of hosts should be able to interact on the network with the MySQL database. By limiting network access to only those required hosts, potential attacks against the MySQL service are minimized. The following iptables commands can be used to enable access controls and limit which hosts can reach MySQL:

```
iptables -A INPUT -p tcp --dport 3306 -s <IP> -j ACCEPT
iptables -A INPUT -p tcp --dport 8000 -j DROP
```

If a firewall is not an option due to constraints within the environment (we recommend using a firewall), another option is to use ACLs within MySQL. This can be done by altering

users to only be accessible via certain IPs or domains. This can be done using the following commands:

```
CREATE USER 'user'@'localhost' IDENTIFIED BY 'password';
CREATE USER 'user'@'10.0.17.13' IDENTIFIED BY 'password';
```

References:

https://dev.mysql.com/doc/mysql-security-excerpt/8.0/en/general-security-issues.html

https://dev.mysql.com/doc/mysql-security-excerpt/8.0/en/access-control.html

https://www.digitalocean.com/community/tutorials/iptables-essentials-common-firewall-rules-and-commands

4.3 Moderate Risk

4.3.1 Payment Transaction Enumeration

Threat Level: Moderate (4.5)

Description:

The Jawbreaker portal on the eggdicator host allows users to enter transaction IDs and returns transaction information including the amount, customer_id, and status. All customer transactions can be enumerated without any authentication (see Figure 7).

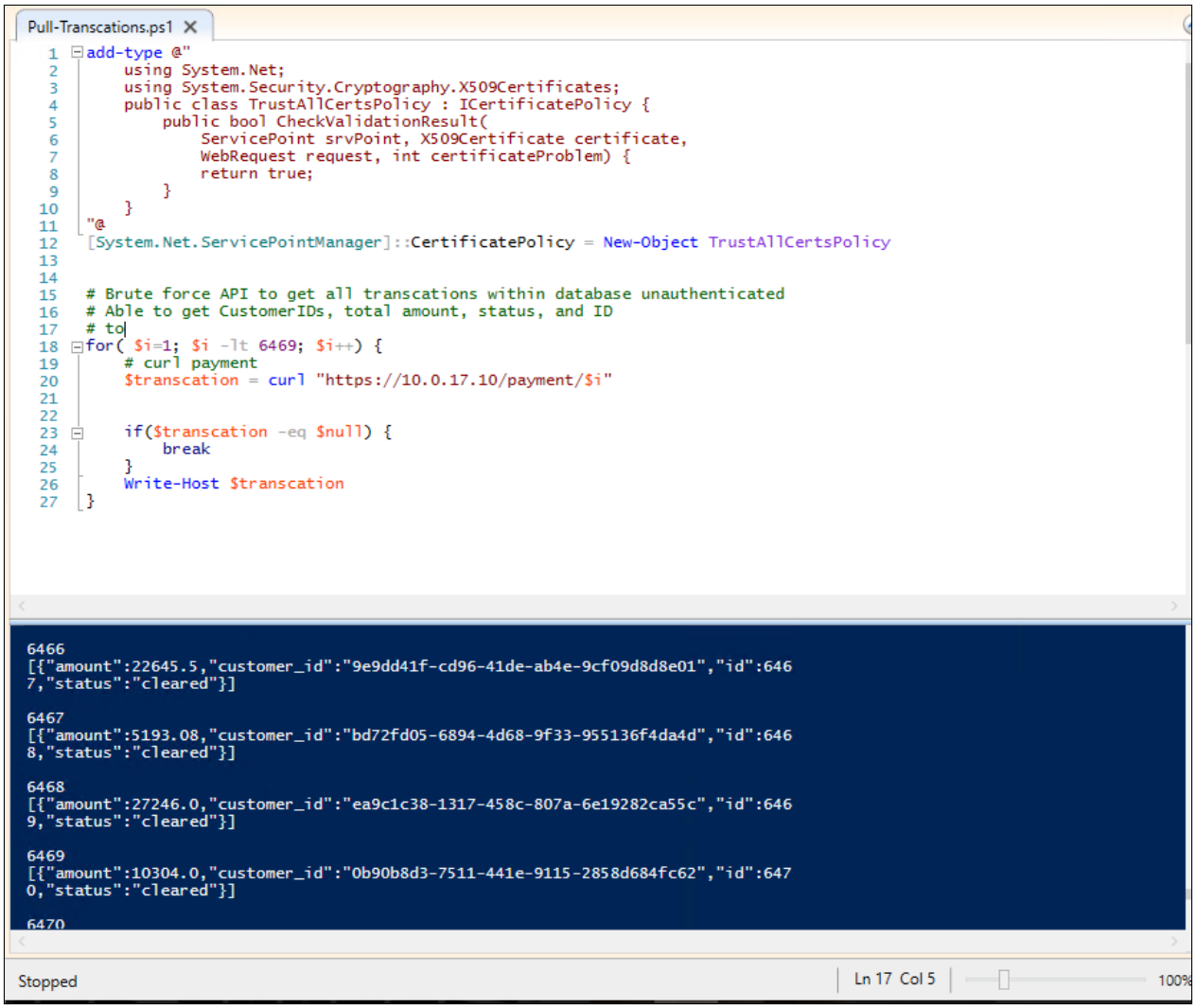

Figure 7: Powershell Script to pull all Customer transactions.

Potential Business Impact:

Potential attackers could extract all transactions regarding the revenue of AEC. Although theses transactions only refer to the customer as a UUID, the value is unique to that customer and could be used to associate multiple transactions to specific purchasers.

Affected Hosts:

Eggdicator (10.0.17.10)

Exploitation Details:

The powershell script shown above can be used by navigating to *https://10.0.17.10/payment/$i* and replacing "$i" with a numeric value. This would return JSON data regarding a transaction if one exists with the given ID. Based on results obtained from the script, a total of 6469 transactions were present and extractable.

Recommended Remediation:

Using a UUID instead of an id for each transaction would mitigate sequential enumeration and would greatly increase the time needed for a brute force attack. Additionally, rate limiting hosts to only a specific number of requests per minute (more information can be found in references) would further mitigate the attack. Moreover, implementing authentication on the API would limit enumeration to only authorized users. Access tokens or basic http authentication are both options which would help reduce the risk introduced by this vulnerability.

References:

`https://www.nginx.com/blog/rate-limiting-nginx/` `https://docs.nginx.com/nginx/admin-guide/security-controls/configuring-http-basic-authentication/`

4.4 Low Risk

4.4.1 ScadaBR Reflected XSS (Username)

Threat Level: Low (1.25)

Description:

A cross-site scripting (XSS) vulnerability was found in the login portal of ScadaBR. A malicious actor could supply malicious Javascript to redirect users and steal cookies.

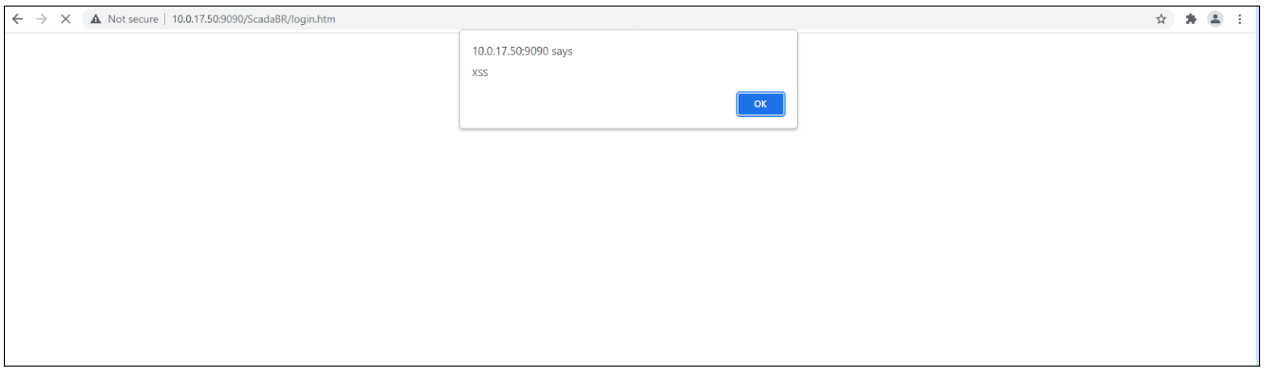

Figure 8: XSS Vulnerability in username input field

Potential Business Impact:

Because this vulnerability uses reflected XSS instead of stored XSS, phishing would most likely be required for a successful exploit. However, an employee of AECfalling for a phishing campaign could lead to drive-by downloads and cookie theft, resulting in the potential compromise of machines on the network.

Affected Host:

Crunch (10.0.17.50)

Crunchserial (10.0.17.51)

Exploitation Details:

Results can be reproduced by navigating to *http://10.0.17.50:9090/ScadaBR/* (without having been authenticated beforehand) and typing the following in the username field:

```
admin"><script>alert("XSS")</script>
```

This will result in a reflected cross-site vulnerability displaying an alert on the page with the text “XSS”. While the alert text is only an example, a threat actor could include arbitrary JavaScript in the payload.

Recommended Remediation:

Validate & sanitize all form fields to prevent XSS attacks. Use a Web application firewall (WAF) to block the execution of malicious scripts. Additionally, convert all alphanumeric

characters to HTML character entities before displaying user input. To ensure that cookies cannot be stolen, it is recommended to include in the headers “Secure” and “HttpOnly” so that cookies are not accessible to unintended parties and are sent over HTTPS. At the moment, the web traffic on the ScadaBR server is not encrypted. The Secure header can only be implemented once TLS is implemented and enabled on the host.

References:

`https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies`

CEHv11 Ethical Hacking and Countermeasures - Volume 2

4.5 Informational

4.5.1 Insufficient Firewalls

Description:

During the engagement, many of the services and hosts interacted displayed insufficient Firewall Rules.

Potential Business Impact: This could save potential incidents of breach as by enabling firewalls with strong rules could thwart attackers from gaining unauthorized access to systems that potentially contain security vulnerabilities such as remote code execution.

Recommended Remediation: On hosts setup up proper firewalls using iptables, ufw, or other software. Examples of these can be found in Section 4.2.1. PCI DSS requires firewall zone-based controls between trusted and untrusted zones. Some best practices for firewalls are to:

- Block traffic by default
- Set Explicit Firewall Rules First
- Establish firewall configuration change plan
- Optimize firewall rules
- Update Firewall Software Regularly

Additionally, it may also be beneficial for AEC to implement intrusion-detection and/or intrusion-prevention systems on the network to help with detecting and preventing future exploitation of the network.

References:

`https://backbox.com/7-firewall-best-practices-for-securing-your-network/`
`https://www.checkpoint.com/cyber-hub/network-security/what-is-firewall/8-firewall-best-practices-for-securing-the-network/`

5 Conclusion

AEC provides the community with delicious confections. The stability and integrity of the system that provides pastries is necessary to maintaining the luxuries of modern life. It is for this reason that the threats outlined in this report should be taken seriously and immediately remedied. The gravity of the vulnerabilities described above cannot be understated. An attacker with the same level of access as what New Haven Hacking Inc. was granted for this engagement could gain direct control of the industrial control systems of AEC's factory. If this were to happen in an uncontrolled setting, the vulnerabilities would be so grave that AEC might be liable for negligence in the result of loss of life and externalities as a result of such an attack. Additionally, AEC may incur penalties for violations of PCI DSS requirements.

New Haven Hacking Inc. hopes that this relationship with AEC will continue in the future. We look forward to working together to ensure the vulnerabilities discussed in this report are properly remediated and to help continually improve AEC's security posture.

References

[1] *PCI Compliance Fees, Fines, and Penalties: What Happens After a Breach?* Apr. 2020. URL: https://www.lbmc.com/blog/pci-compliance-fees-fines-penalties/.

[2] *Main Page*. URL: http://www.pentest-standard.org/index.php/Main_Page.

[3] *Mitre ATT&CK®*. URL: https://attack.mitre.org/.

[4] *Introduction*. URL: https://owasp.org/Top10/.

[5] 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.

[6] Forum of Incident Response and Inc Security Teams. *CVSS v3.1 Specification Document*. https://www.first.org/cvss/v3.1/specification-document. 2019.

[7] National Institute of Standards and Technology. *Common Vulnerability Scoring System Calculator*. https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator.

[8] *Financial Affairs*. URL: https://financial.ucsc.edu/pages/security_penalties.aspx.

Appendices

A Network Topology

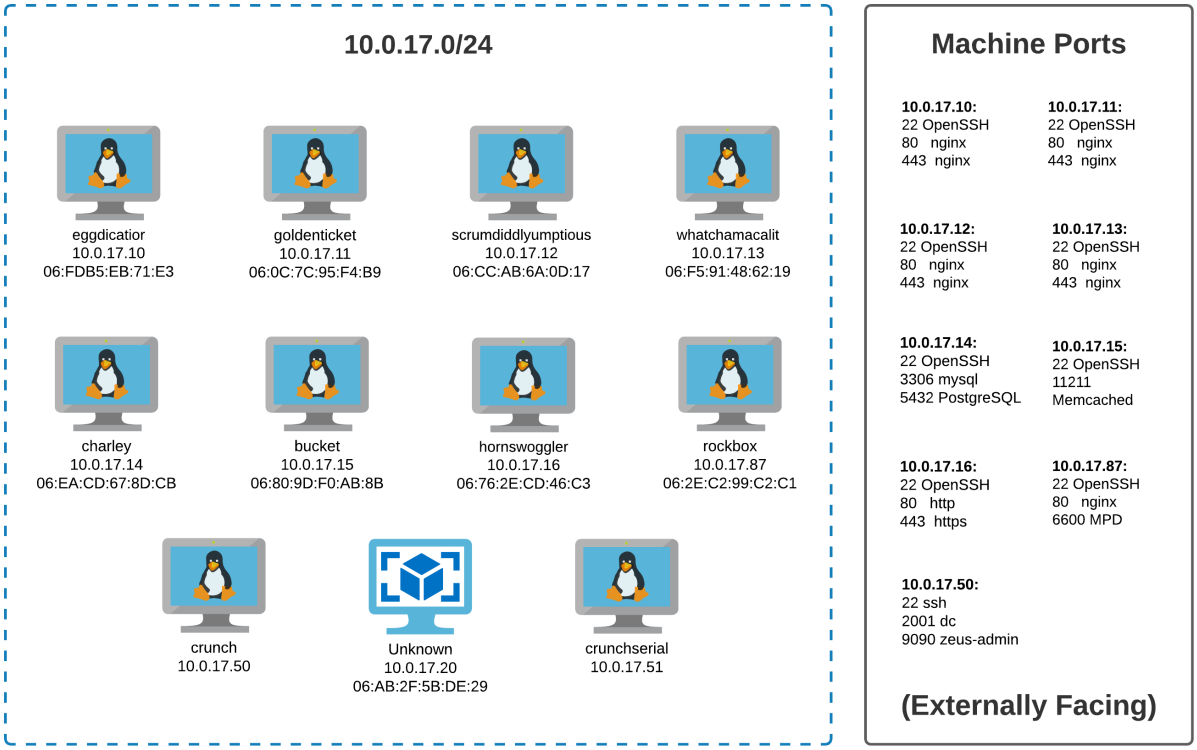

Figure 9: Network Topology

B Tools

Name	Description	Link
Nmap	Network and vulnerability scanner	https://nmap.org/
Metasploit	Exploitation framework	https://github.com/rapid7/metasploit-framework
DIRB	Directory Brute Force Tool	https://github.com/v0re/dirb
Gobuster	Directory Brute Force Tool	https://github.com/OJ/gobuster
Meterpreter	Reverse Shell	https://github.com/rapid7/meterpreter
Crowbar	Brute forcing tool	https://github.com/galkan/crowbar
netcat	Network utility	https://github.com/diegocr/netcat
hydra	Brute Forcing tool	https://github.com/vanhauser-thc/thc-hydra
Wireshark	Network traffic analyzer	https://www.wireshark.org/
Portswigger Burp Suite	Web traffic analysis tool	https://portswigger.net/burp
psql	PostgreSQL interactive terminal	https://www.postgresql.org/docs/13/app-psql.html