

{ no logo }

coming soon

Security Assessment Report

Conducted by:

Team XX

October 10, 2025

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

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
3	Testing Methodology	6
3.1	Penetration Testing Execution Standard	6
3.2	MITRE ATT&CK Framework	6
3.3	OWASP Top 10	6
3.4	NIST SP 800-53	6
4	Technical Findings	8
4.1	Critical Risk	9
4.1.1	Lack Of PostgreSQL Authentication	9
4.2	High Risk	11
4.2.1	Lack of MariaDB Authentication	11
4.3	Moderate Risk	14
4.3.1	Payment Transaction Enumeration	14
4.4	Low Risk	16
4.4.1	ScadaBR Reflected XSS (Username)	16
4.5	Informational	18
4.5.1	Insufficient Firewalls	18
5	Conclusion	19
	Appendices	20
A	Network Topology	20
B	Tools	21

1 Report Overview

1.1 Executive Summary

Team XX was contacted by All Ports Tour (APT) for a penetration test in order to identify security issues within their infrastructure. This report was written initially on October 17th, 2025 and submitted same day at <insert submission time>. This penetration test is in the interest of APT, as part of a restrained scope penetration test and risk assessment. The Report Overview section contains an outlined summary of Team XX's findings, including recommendations for improving APT'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, APT 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. Based on these issues, we suggest spending resources to become complaint. Details of all PCI DSS violations can be found in Section ??.

Team XX 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 APT vast sums of money from both lawsuits and long term loss of business.

During this engagement, a total of **X** vulnerabilities were found in APT's network. In terms of severity, **X** vulnerabilities are critical, **X** vulnerabilities are high, **X** vulnerability is moderate, and **X** vulnerabilities posed low risk. Many of the vulnerabilities in the environment are due to **reason1**, **reason2**, and **reason3**. More information about these vulnerabilities can be found in Section 4.

1.2 Engagement Overview

Team XX conducted a penetration test starting on October 18th based upon the Request for Proposal (RFP) document obtained by Team XX. 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 APT's information systems.
- Assisting APT in improving their security posture.

1.3 Scope of Engagement

The full scope of this penetration test was limited to the following CIDR range. <Continue talking about the scope>

- X.X.X.X/X

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. Team XX did not exfiltrate, modify, or delete any data not included in this report.

Team XX is available upon request to improve the security, protect the employees, and customers of APT. This includes verifying and validating implemented mitigation techniques as well as deploying security strategies to ensure APT has several layers of defense. The team is happy to continue a partnership with APT 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 APT. 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 Team XX did not find.

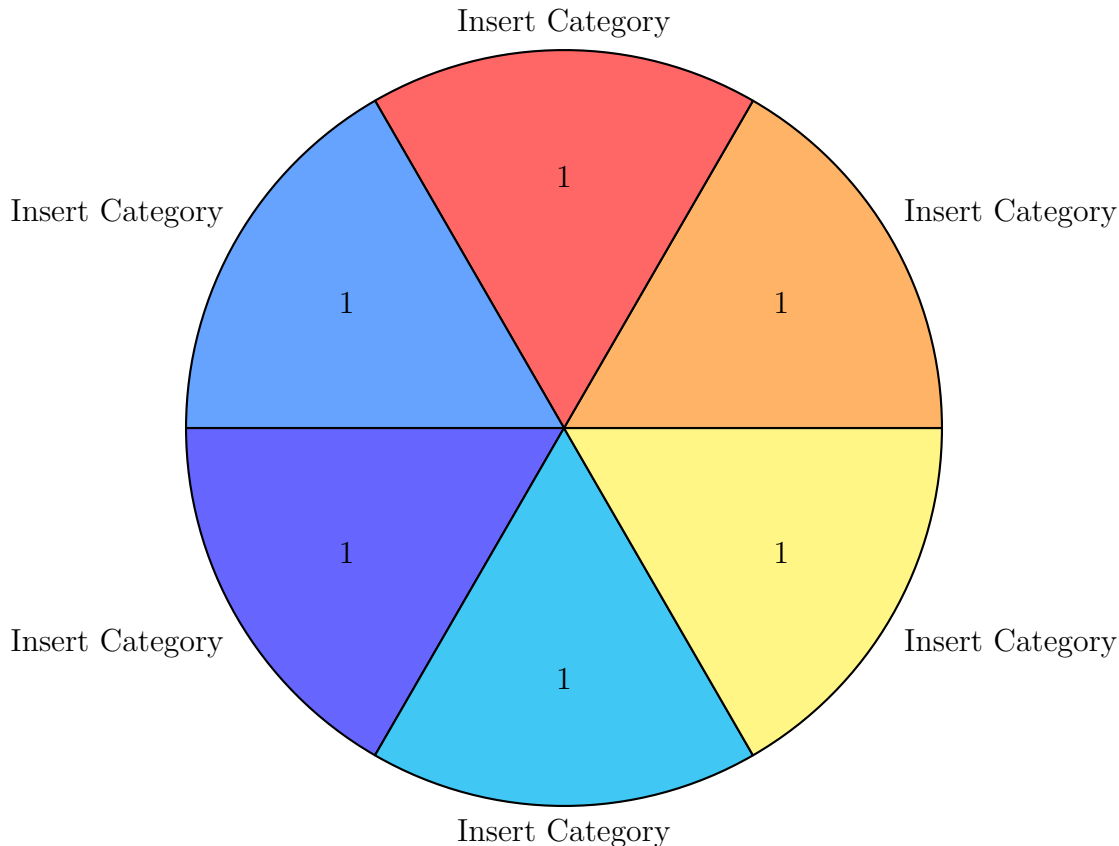


Figure 1: Summary of Issues within the Network

Default Credentials & Lack of authentication

The most immediate observation about APT's security posture is that default, null, and passwordless authentication was discovered on multiple systems. Team XX 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 APT'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 APT. 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:

- 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, Team XX was impeded by the security safeguards APT had in place. A number of basic security best practices were observed that limited Team XX's ability to move through the network. Some instances of aforementioned security practices implemented by APT 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.

3 Testing Methodology

3.1 Penetration Testing Execution Standard

Throughout the engagement Team XX, references the Penetration Testing Execution Standard (PTES) when conducting security assessments.

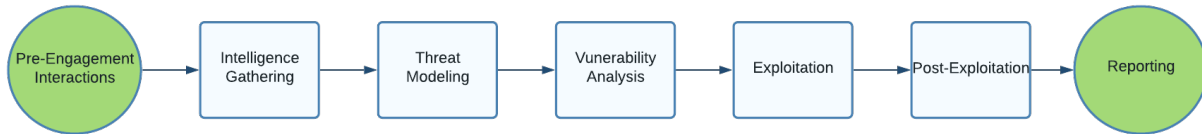


Figure 2: PTES Methodology

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. Team XX uses ATT&CK to aide in understanding TTPs that can be used to conduct an attack against APT that could be conduct by real world adversaries.

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. 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

3.4 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 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 APT's information systems.

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

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) [cvssdocs].

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	X	X	X	X

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 [nvdcvss].

Summary of Vulnerabilities by Base Score

Risk Summary	Overall Risk Score	Impact	Exploitability
<Insert Vulnerability>	0	0	0
<Insert Vulnerability>	0	0	0
<Insert Vulnerability>	0	0	0
<Insert Vulnerability>	0	0	0

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 ??.

Failures of PCI DSS can result in fines and other punishments. Each security incidents and breaches can result of a \$500,000 fine [pcidssfineexample]. 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:

```

COPY billing.credit_cards (id, name, number, expiration, ccv, zip) FROM stdin;
1  Robert ██████████ 2769 05/28 ████████ 39734
2  Christy ██████████ 2568 03/23 ████████ 57907
3  Angel ██████████ 5750 07/27 ████████ 07866
4  Alex ██████████ 7190 10/25 ████████ 09259
5  Nathaniel ██████████ 2319 11/28 ████████ 52715
6  John ██████████ 6933 10/28 ████████ 08707
7  Traci ██████████ 2587 12/29 ████████ 55043
8  Rick ██████████ 5961 12/29 ████████ 32536
9  Nicole ██████████ 9990 07/26 ████████ 74002
10 Holly ██████████ 2120 08/24 ████████ 63694
11 Roberto ██████████ 0550 05/23 ████████ 08238
12 William ██████████ 2859 06/28 ████████ 03468
13 Danielle ██████████ 7411 02/25 ████████ 19371
14 Raymond ██████████ 1006 01/26 ████████ 35091
15 William ██████████ 4770 11/27 ████████ 67628
--More-- (0%) █

```

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

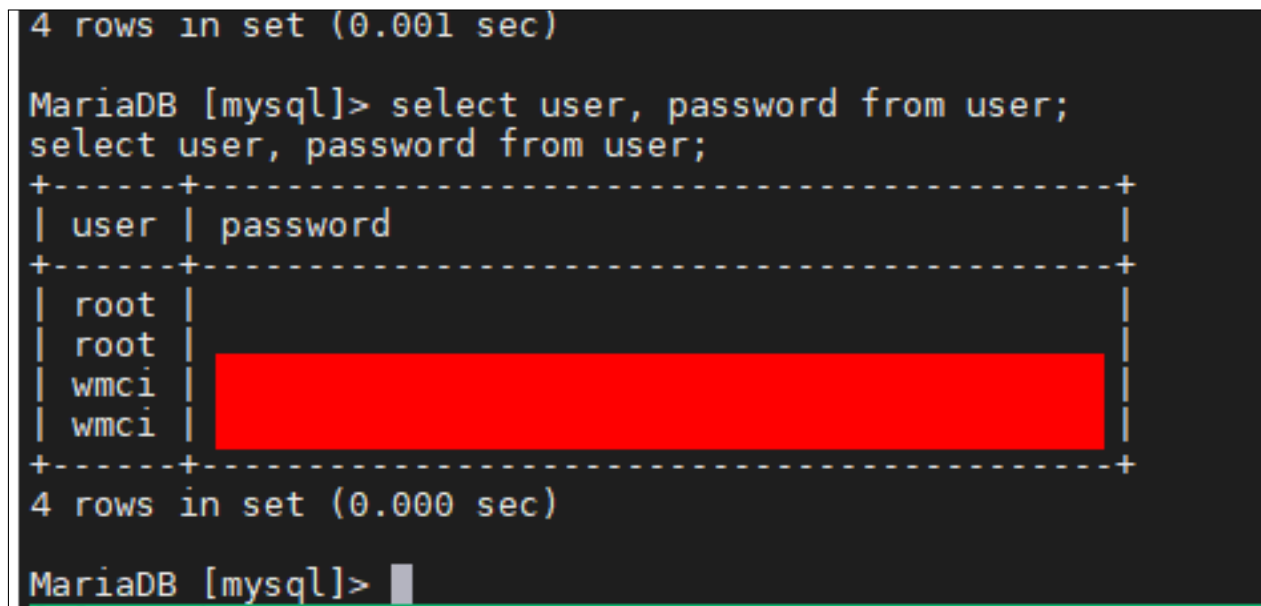
4.2.1 Lack of MariaDB Authentication

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 APT's croissant marketplace.
- Insertion, deletion, and modification of all data within the database.



```
4 rows in set (0.001 sec)

MariaDB [mysql]> select user, password from user;
select user, password from user;
+-----+-----+
| user | password |
+-----+-----+
| root |          |
| root |          |
| wmc |          |
| wmc |          |
+-----+-----+
4 rows in set (0.000 sec)

MariaDB [mysql]>
```

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 APT store website (Scrumdiddlyumptious) users' passwords can be decoded from base64.

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

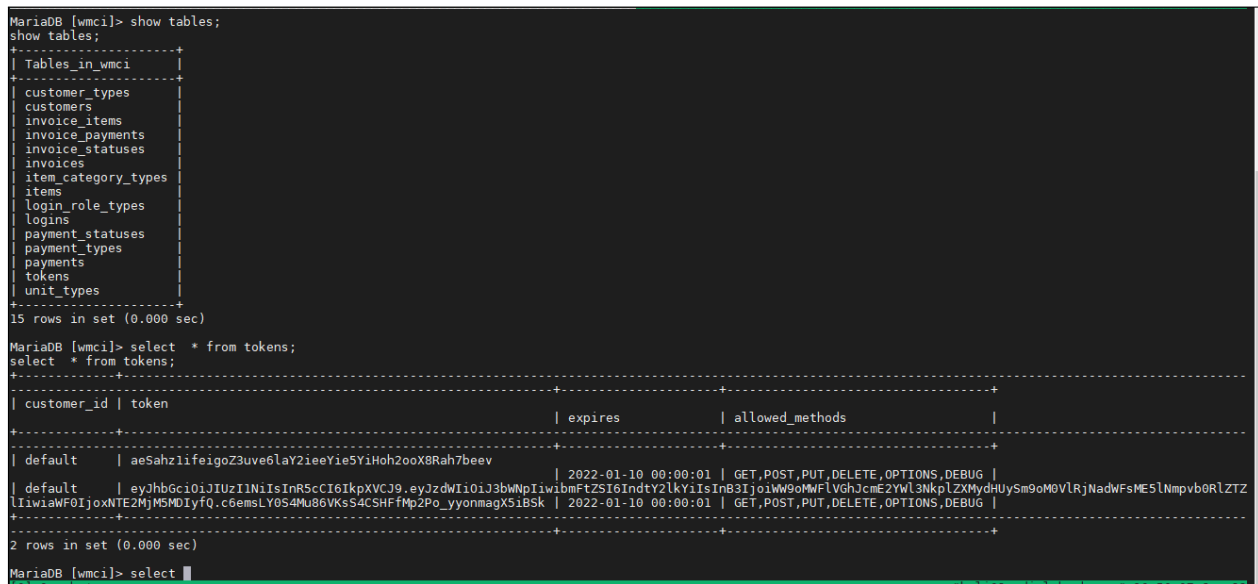
Scrumdiddlyumptious (10.0.17.12)

Whatchamacallit (10.0.17.13)

Exploitation Details:

Team XX 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.



```

MariaDB [wmcil]> show tables;
show tables;
+-----+
| Tables_in_wmcil |
+-----+
| customer_types  |
| customers       |
| invoice_items   |
| invoice_payments|
| invoice_statuses|
| invoices        |
| item_category_types|
| items          |
| login_role_types|
| logins          |
| payment_statuses|
| payment_types   |
| payments        |
| tokens          |
| unit_types      |
+-----+
15 rows in set (0.000 sec)

MariaDB [wmcil]> select * from tokens;
select * from tokens;
+-----+-----+-----+-----+
| customer_id | token | expires | allowed_methods |
+-----+-----+-----+-----+
| default    | aeSahz1ifeigoZ3uve6laY2ieYie5YiHoh2ooX8Rah7beev | 2022-01-10 00:00:01 | GET,POST,PUT,DELETE,OPTIONS,DEBUG |
| default    | eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJ3bWpIiwibmFtZSI6ImV2bW90M0VlRjNadWFsME5lNmpvb0RlZTZlIiwiaWF0IjoxNTE2MjM5MDIyfQ.c6emsLY054Mu86Vks54CSHFFMp2Po_yyonmagXS1BSK | 2022-01-10 00:00:01 | GET,POST,PUT,DELETE,OPTIONS,DEBUG |
+-----+-----+-----+-----+
2 rows in set (0.000 sec)

MariaDB [wmcil]> select

```

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>

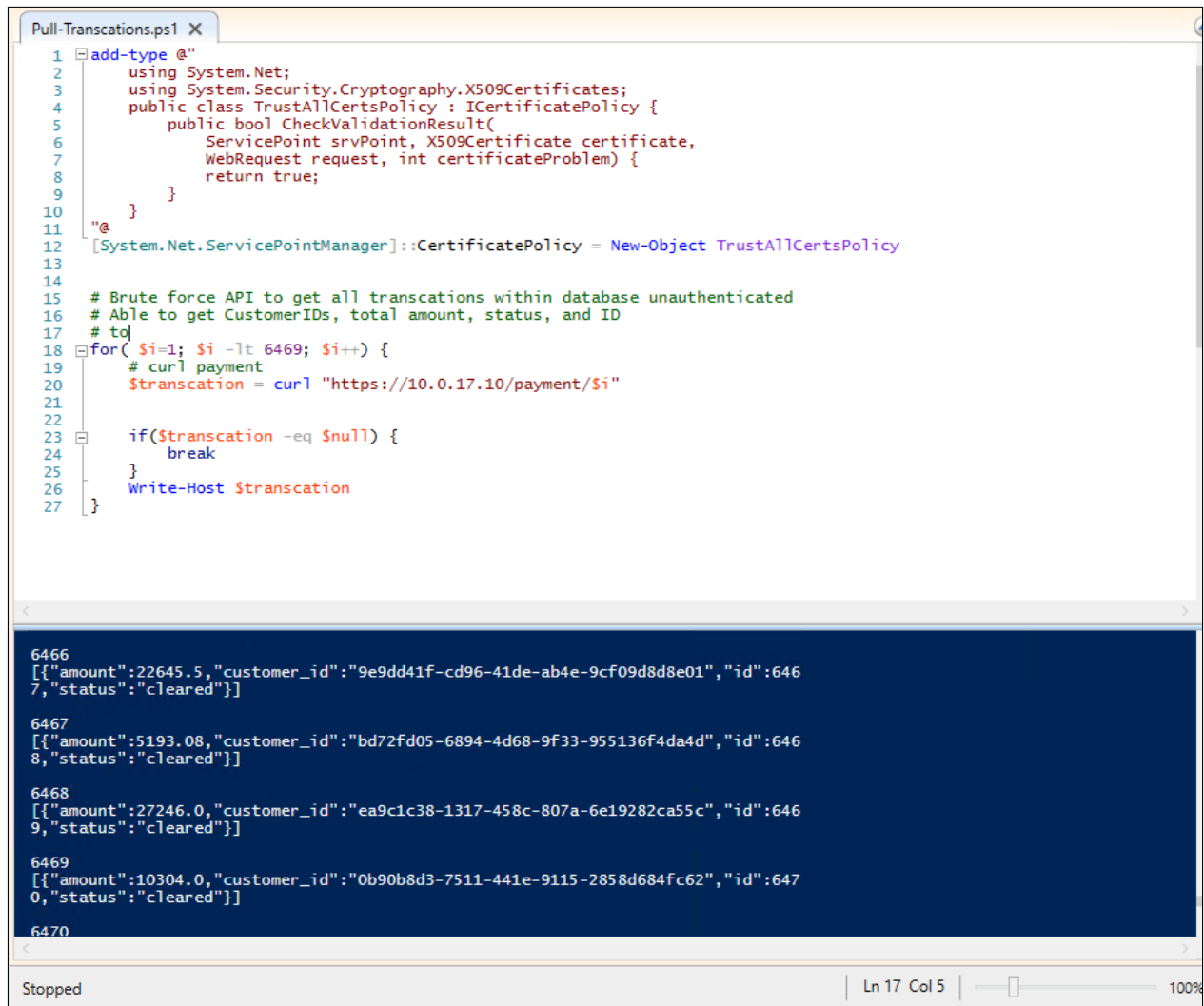
4.3 Moderate Risk

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).



```
Pull-Transactions.ps1 X
1 add-type @"
2     using System.Net;
3     using System.Security.Cryptography.X509Certificates;
4     public class TrustAllCertsPolicy : ICertificatePolicy {
5         public bool CheckValidationResult(
6             ServicePoint srvPoint, X509Certificate certificate,
7             WebRequest request, int certificateProblem) {
8             return true;
9         }
10    }
11  @"
12  [System.Net.ServicePointManager]::CertificatePolicy = New-Object TrustAllCertsPolicy
13
14  # Brute force API to get all transactions within database unauthenticated
15  # Able to get CustomerIDs, total amount, status, and ID
16  # to
17  for( $i=1; $i -lt 6469; $i++) {
18      # curl payment
19      $transaction = curl "https://10.0.17.10/payment/$i"
20
21      if($transaction -eq $null) {
22          break
23      }
24      Write-Host $transaction
25  }
26
27  }
```

```
6466 [{"amount":22645.5,"customer_id":"9e9dd41f-cd96-41de-ab4e-9cf09d8d8e01","id":646
7,"status":"cleared"}]
6467 [{"amount":5193.08,"customer_id":"bd72fd05-6894-4d68-9f33-955136f4da4d","id":646
8,"status":"cleared"}]
6468 [{"amount":27246.0,"customer_id":"ea9c1c38-1317-458c-807a-6e19282ca55c","id":646
9,"status":"cleared"}]
6469 [{"amount":10304.0,"customer_id":"0b90b8d3-7511-441e-9115-2858d684fc62","id":647
0,"status":"cleared"}]
6470
```

Stopped | Ln 17 Col 5 | 100%

Figure 7: Powershell Script to pull all Customer transactions.

Potential Business Impact:

Potential attackers could extract all transactions regarding the revenue of APT. Although these 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.

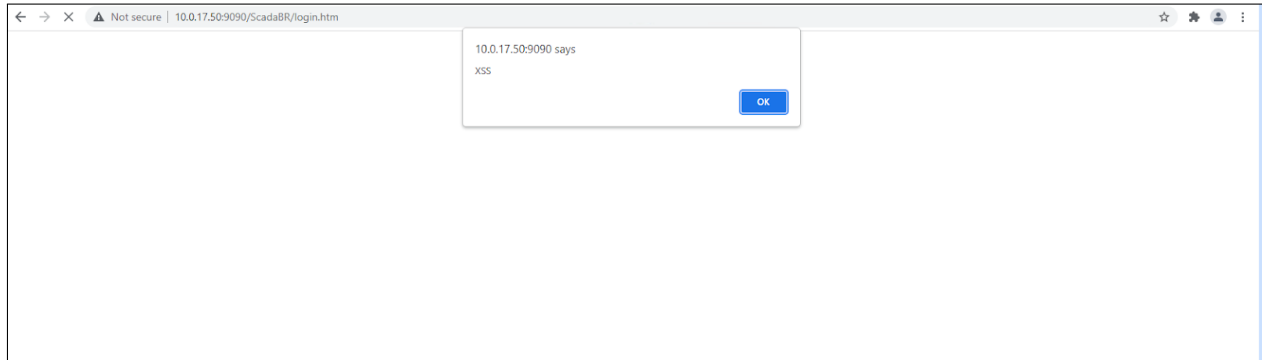


Figure 8: XSS Vulnerability in username input field

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 APTfalling 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 APT 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

APT 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 Team XX was granted for this engagement could gain direct control of the industrial control systems of APT's factory. If this were to happen in an uncontrolled setting, the vulnerabilities would be so grave that APT might be liable for negligence in the result of loss of life and externalities as a result of such an attack. Additionally, APT may incur penalties for violations of PCI DSS requirements.

Team XX hopes that this relationship with APT 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 APT's security posture.

Appendices

A Network Topology

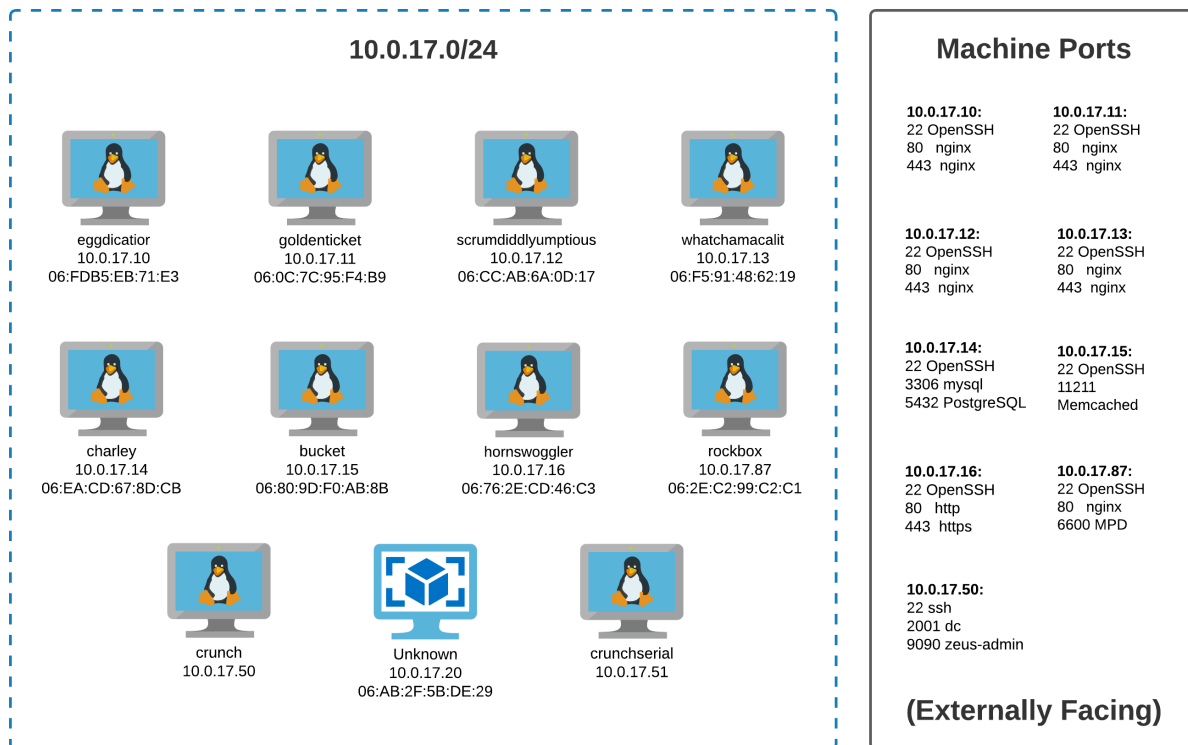


Figure 9: Network Topology

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