Robert A. Kalka Metropolitan Skyport

SECURITY ASSESSMENT

1/13/24

Performed by: Finals-XX

DISCLOSURE

This report contains confidential and sensitive information. It is intended solely for the information and use of Robert A. Kalka Metropolitan Skyport and its authorized personnel.

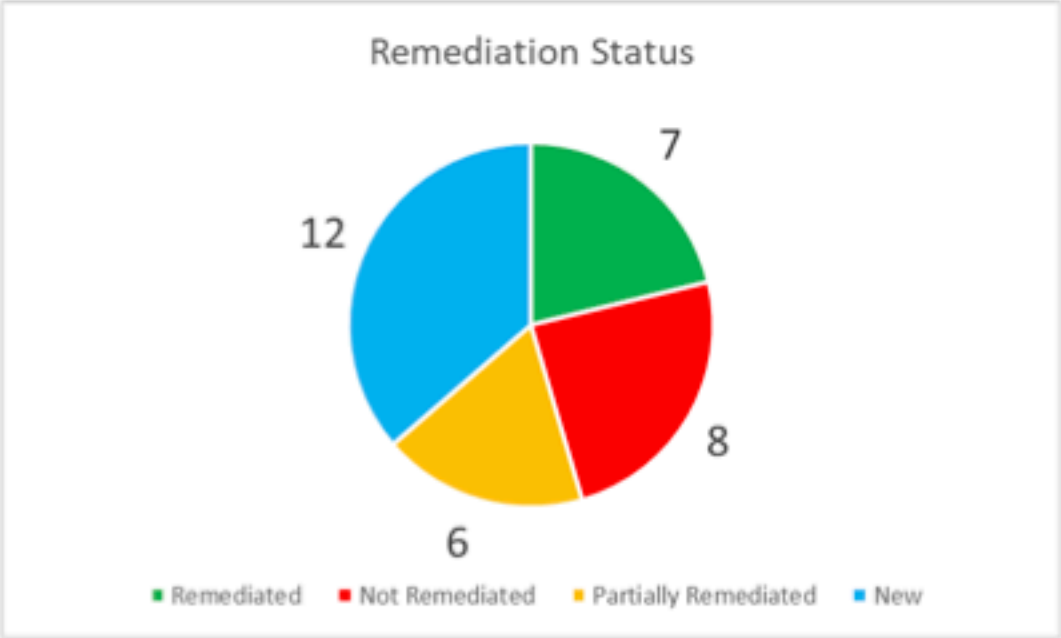
TABLE OF CONTENTS

EXECUTIVE SUMMARY

RAKMS requested a comprehensive retest against the guest, user, corp, and train networks. This assessment was executed in two stages: the pre-engagement phase, consisting of open-source intelligence gathering and the assessment phase, consisting of active reconnaissance and leveraging exploits to prove the existence of system vulnerabilities. The following report outlines the findings from these engagement phases and mitigations for pressing issues.

Overall, Finals-XX found 34 vulnerabilities, 9 of which were critical or high. Some reoccurring finds were weak passwords and known-vulnerable software still in use. The RAKMS security team remediated many previously seen vulnerabilities. However, there was a theme of partial/incomplete patches. Multiple applications were patched in ways that introduced new vulnerabilities or patched in ways where the vulnerability could still be exploited.

Testing also revealed the storage of customer data was greatly improved. However serious vulnerabilities with the people movers still exist. These vulnerabilities could possibly result in bodily harm or other serious accidents and need to be handled as soon as possible.



KEY FINDINGS AND REMEDIATION

Finals-XX observed multiple overarching issues with RAKMS network security. These are namely weak password policies, lack of a patch management program, and insecure coding practices. These security practices are what lead to most vulnerabilities within the RAKMS network. Although these have been greatly improved upon by the RAKMS security team since the previous test, many symptoms of these underlying issues still plague the network.

Weak passwords were observed across the RAKMS network during the previous assessment and this assessment. Many critical accounts including domain administrators, database administrators, and service accounts were observed to be using weak and easily guessable passwords. Although a password policy was enabled, multiple accounts were observed using passwords that do not comply with this policy. Implementing a stronger password policy and ensuring its enforcement will remediate this issue across the network.

The lack of a patch of a patch management policy was felt throughout the RAKMS network. [redacted] observed multiple outdated, unsupported, and vulnerable versions of software being used on critical servers. It should also be noted that updates and patches were applied inconsistently across the network. Since the previous test, some systems and software were completely patched while more critical servers and software were left un-updated. Implementing a patch management program will ensure systems are consistently kept up to date across the network.

Insecure coding practices were observed in multiple places across the RAKMS network. Many of the previous vulnerabilities stemming from insecure coding practices were patched. However, it was observed

that many of these patches either did not fix the vulnerability or introduced new vulnerabilities. Educating RAKMS developers on secure coding practices and proper handling of sensitive information will ensure patches are properly implemented and these same vulnerabilities will not be introduced again.

COMPLIANCE

Considering the TSA's continuous commitment to cybersecurity, ensuring compliance is a key priority for airport management. One major idea is to separate the ICS network from the rest of the networks. For the most part, RAKMS does this very well by having a separate train network. Additionally, network ACLs were observed during the assessment, initially limiting Finals-XX's access to these networks. However, during the assessment, these ACLs were seemingly non-functional.

While there has been progress in password compliance through the adoption of password policies, various hosts and web applications continue to breach TSA and FAA password regulations.

During their prior review, Finals-XX identified a major violation of PCI DSS. It appears Finals-XX's urgent recommendation to address this issue was taken seriously, as no further breaches, leaks, or plaintext occurrences of PCI DSS data were observed by Finals-XX in the current assessment.

ASSESSMENT SUMMARY

SCOPE

Finals-XX was authorized to perform a network penetration test by Robert A. Kalka Metropolitan Skyport. Testing was conducted from the perspective of an attacker with a local connection to the guest network of Robert A. Kalka Metropolitan Skyport.

The following scope was given by the client to test on:

- User Network - 10.0.1.0/24
- Domain Network - 10.0.0.0/24
- Train Network - 10.0.20.0/24
- Guest Network - 10.0.200.0/24

The following hosts were explicitly out of scope

- VDI Network - 10.0.254.0/24
- VPN Network - 10.0.255.0/24

ASSESSMENT NARRATIVE

Reconnaissance:

In the initial phase of the assessment, meticulous reconnaissance was conducted as open-source intelligence (OSINT) to gather valuable information for later testing and the phishing exercises. This involved identifying company size, employee names, and uncovering any underlying technologies through means such as job postings, including the presence of AWS servers.

Active Reconnaissance:

With an entry point secured on the guest network in the form of provided virtual desktop infrastructure (VDI), Finals-XX delved into active reconnaissance. The team systematically enumerated all active hosts within the guest network and quickly tested the ability to reach the other networks within the scope. From the guest network, Finals-XX scanned the entire Corporate and Train networks. However, the devices on the User network were unresponsive aside from replying to pings.

Initial Access:

Having identified active machines in the previous step, Finals-XX targeted outdated software with known vulnerabilities. This strategic approach resulted in initial access to the Windows domain controller by allowing Finals-XX to extract all hashed passwords. While enumerating the authentication mechanisms on the Train subnet, Finals-XX determined the trains were using an improper session handling. The sessions were saved via serialized data in cookies; however, the application was not signing the cookies. This allows the end user to modify their current user. The trains were also relying on an insecure

serialization module called “Pickle.” This module allows for arbitrary code execution while deserializing the data. With this, Finals-XX was able to establish a connection to the trains as root.

Privilege Escalation:

Finals-XX circumvented the need for further privilege escalation on this described hosts as the vulnerabilities allowed for root/administrator access. Efforts to discover other privilege escalation vectors were still being made; this involved running enumeration tools to discover any access control misconfigurations, accessible passwords, internal services, and many other techniques. Finals-XX identified only one privilege escalation technique noted in the findings section.

Internal Reconnaissance and Pivot:

Returning to internal reconnaissance, Finals-XX continued the exploration of network hosts. Leveraging domain accounts, the team pivoted to other domain-connected Windows machines, notably the mail server. This server became a pivotal point for executing the phishing portion of the test. However, when trying to send the phishing email, Exchange was unresponsive and after support rebooted the system the email failed to send.

Data Enumeration and Vulnerability Assessment:

In the final phase, Finals-XX systematically enumerated all accessible machines. The focus shifted to identifying personally identifiable information (PII) and uncovering any vulnerabilities overlooked earlier in the assessment. The team was unable to locate any PII, but several additional vulnerabilities were discovered and can be found in the findings section of the report.

FINDINGS SUMMARY

The following vulnerabilities were found.

Vulnerability	Risk
Zerologon	Critical – 9.8
Python Pickle Deserialization	High - 8.8
Tram Insecure Authentication	High - 8.6
All Domain Users Have Local Admin Permissions	High - 8.0
PetitPotam	High - 7.5
Insecure Sudo Configuration	High - 7.3
Weak Password Policy	High - 7.3
SQL Injection in Timesheet Site	High - 7.1
Service Account SPN Usage	High - 7.1
AWS Root User Missing MFA	Medium - 6.6
Stored XSS in Timesheets Site	Medium - 6.3
Unauthenticated API Permits Addition of Trams	Medium - 5.4
Insecure Kerberos Attribute	Medium - 5.4
Real Time Protection Disabled	Medium - 5.3
IDOR on Barcode System	Medium - 5.3
SMB Null Session	Medium - 4.3
PHP Info Exposed	Medium - 4.3
Hardcoded Auth Token	Medium - 4.3
Sensitive Information in S3 Bucket dev-s3-role	Medium - 4.3
Sensitive Information in S3 Bucket dev-barcode-role	Medium - 4.3

Insecure Bag Check-In	Medium - 4.2
AWS Assume Role Privilege Escalation	Medium - 4.1
Improper Input Validation	Low - 3.5
AWS Secrets Found	Low - 3.5
Directory Listing Enabled	Informational
EOL Nginx Version	Informational
SSH PermitRootLogin	Informational
Publicly Accessible S3 Bucket	Informational
Unencrypted EBS Volumes in AWS	Informational
Upcoming ACM Certificate Expiration	Informational
Dev Variable Allows Unlimited Session Limit	Informational
EOL Ruby Version	Informational
Unusual Discoveries on AWS Environment	Informational

TECHNICAL FINDINGS

ZEROLOGON

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H	9.8

AFFECTED SYSTEMS

SKYCONTROL01.CORP.KKMS.LOCAL - 10.0.0.5

DESCRIPTION

Zerologon allows an attacker full domain access by overwriting the domain controller's machine account hash.

BUSINESS IMPACT

Any attacker could take complete control over KKMS's domain leading to complete loss of airport functionality.

OBSERVATION

During testing Finals-XX identified the SKYCONTROL01 domain controller was vulnerable to Zerologon using the Zerologon module from CrackMapExec.

Zerologon validation via CrackMapExec

STEPS TO REPRODUCE

1. Execute "crackmapexec smb 10.0.0.5 -M zerologon"

REMEDIATION

To mitigate these critical vulnerabilities and enhance the security of the domain controller and the entire network, Finals-XX recommends the following actions:

- **Patch and Update**: Immediately apply security patches and updates provided by Microsoft to address the vulnerabilities.

- **Segment the Network**: Isolate the domain controller from the rest of the network by implementing proper network segmentation and firewall rules to minimize lateral movement.
- **Monitor and Audit**: Continuously monitor and audit the network for suspicious activities and implement robust logging mechanisms to detect potential security breaches.

REFERENCES

https://www.crowdstrike.com/blog/cve-2020-1472-zerologon-security-advisory/

PYTHON PICKLE DESERIALIZATION

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H	8.8

AFFECTED SYSTEMS

TRAM1.TRAIN.KKMS.LOCAL - 10.0.20.101
TRAM2.TRAIN.KKMS.LOCAL - 10.0.20.102
TRAM3.TRAIN.KKMS.LOCAL - 10.0.20.103

DESCRIPTION

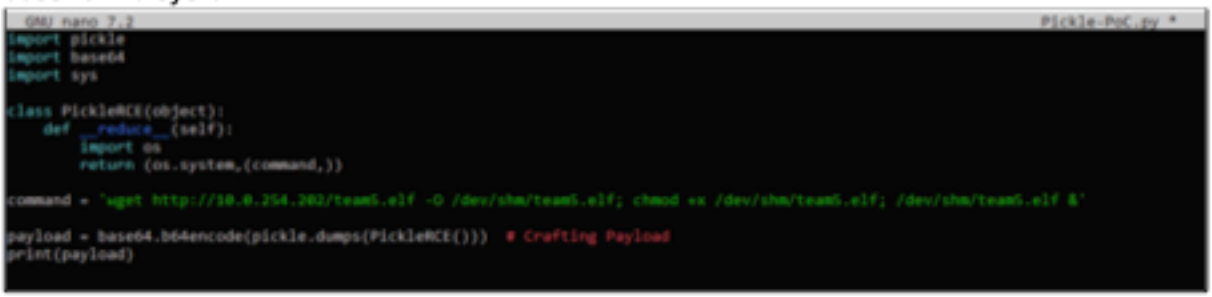
When serialized objects accept user input, it can lead to deserialization attacks and RCE.

BUSINESS IMPACT

An unauthenticated attacker can send a pickle payload in the cookie and receive a shell or other command execution. This could result in an attacker removing functionality to stop and start trams which could have profound consequences.

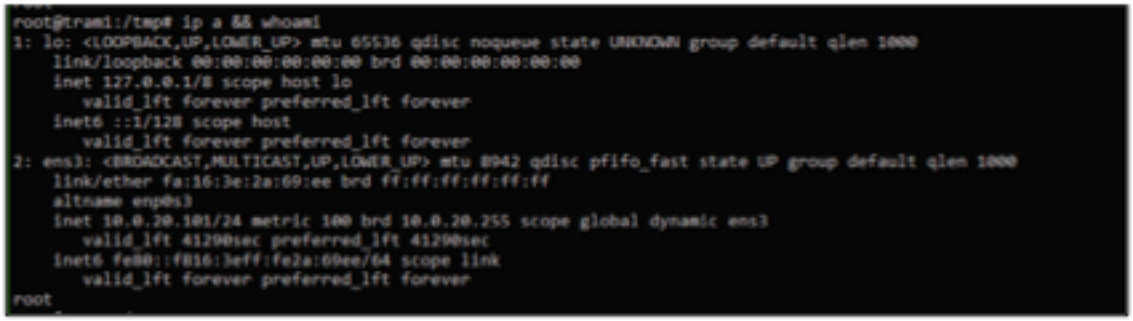
OBSERVATION

Finals-XX used python to create a python program that would download an executable that gave remote shell access to the system.



Pickle Code

And the reverse shell was successful.



Reverse Shell;

STEPS TO REPRODUCE

1. Use a pickle payload that is serialized
2. Put it in the x-auth cookie to be handled by the application
3. Verify the command was run locally

REMEDIATION

- Do not use serialized objects where the user can edit them
- Use a different authentication token that is cryptographically secure

REFERENCES

https://owasp.org/www-community/vulnerabilities/Deserialization_of_untrusted_data

TRAM INSECURE AUTHENTICATION

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:N/I:N/A:H	8.6

AFFECTED SYSTEMS

TRAM1.TRAIN.KKMS.LOCAL - 10.0.20.101
TRAM2.TRAIN.KKMS.LOCAL - 10.0.20.102
TRAM3.TRAIN.KKMS.LOCAL - 10.0.20.103

DESCRIPTION

An unauthenticated attacker on the guest network can gain access to the Tram Location web application, enabling them to start and stop the airport's trams.

BUSINESS IMPACT

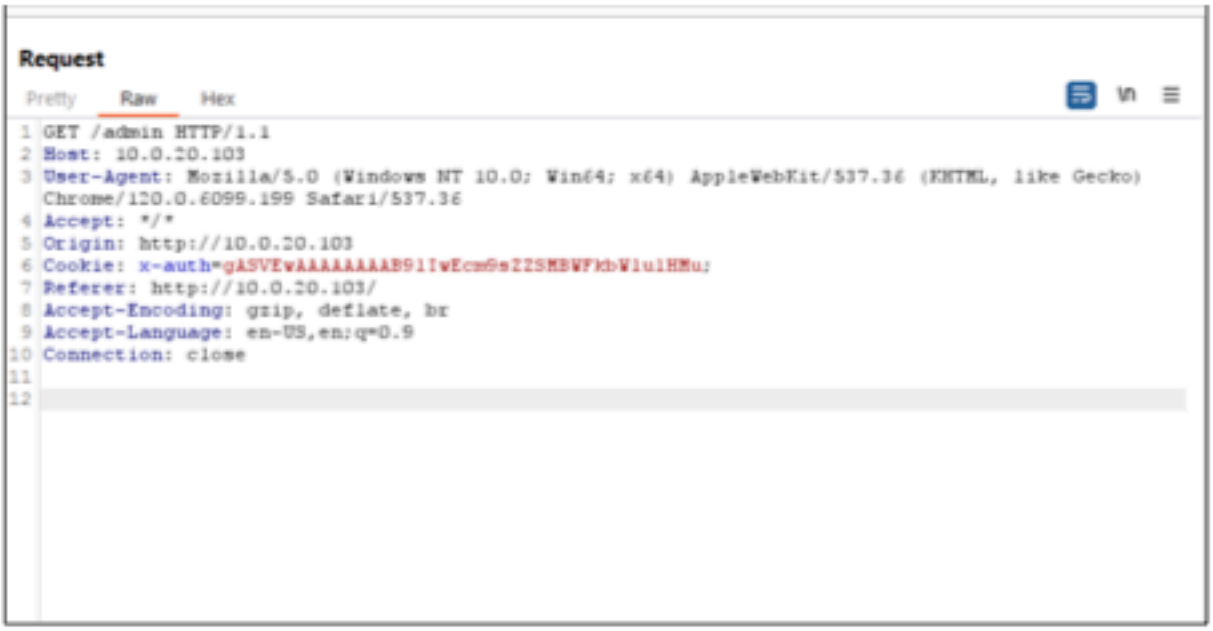
If an attacker gains access to the Tram Admin Interface, they can stop the airport's trams on the tracks, causing service disruptions, possible crashes, and potential loss of life.

OBSERVATION

Finals-XX discovered one of the web applications has a vulnerable authorization cookie which is a serialized pickle object. The object contains the text for the user role which is 'guest' by default.

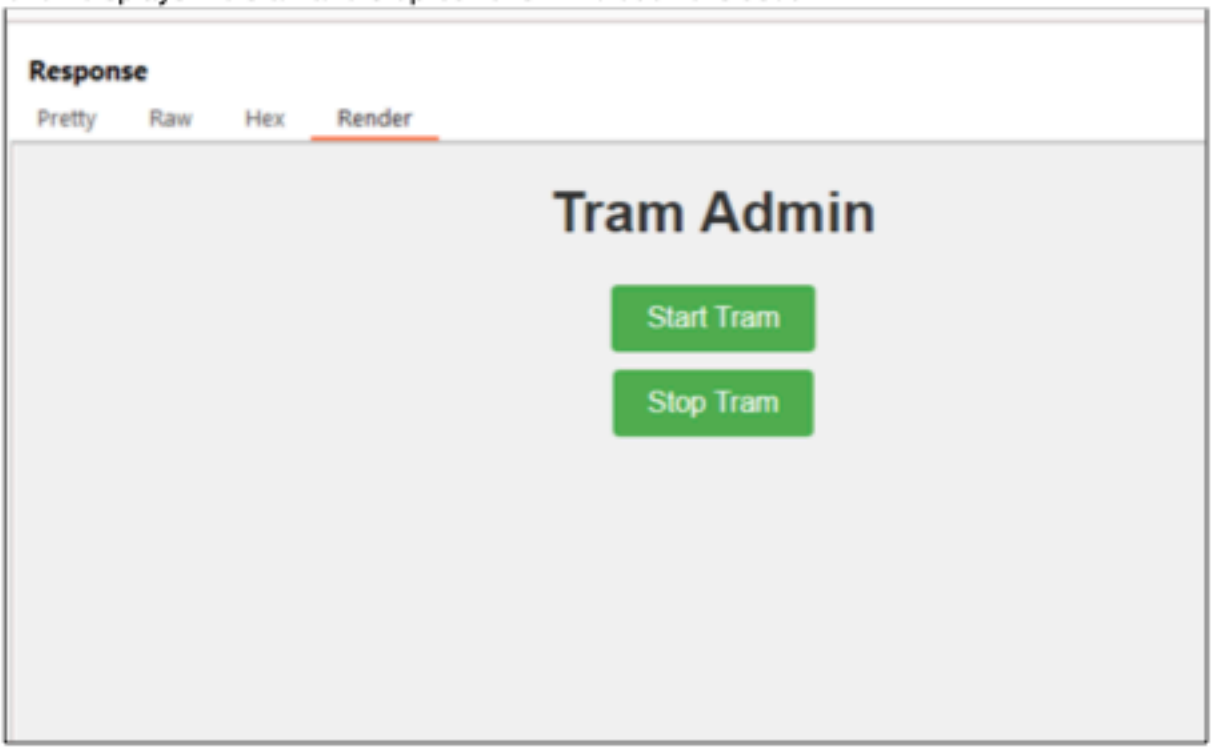
decoded cookie

An attacker can rewrite this to 'admin' which allows any unauthenticated user to gain administrative access.



Request with the new cookie

This function displays the start and stop buttons if the cookie is used.



The tram Start & Stop buttons were revealed on the screen.

STEPS TO REPRODUCE

1. Create a new pickle object with the attributes "role:admin"
2. Pass the base64 object in the *x-auth* cookie
3. Verify the browser now shows admin functionality

REMEDIATION

Finals-XX recommends adding these hosts to the train subnet instead of the guest network. This keeps the separation of ICSs from the rest of the infrastructure. Additionally, Finals-XX recommends implementing a cryptographically secure authentication system with a username or password and/or only allowing whitelisted IP addresses.

REFERENCES

https://owasp.org/Top10/A01_2021-Broken_Access_Control/

ALL DOMAIN USERS HAVE LOCAL ADMIN PERMISSIONS

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:A/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H	8.0

AFFECTED SYSTEMS

SKYDESKTOP01.CORP.KKMS.LOCAL 10.0.0.201
SKYDESKTOP02.CORP.KKMS.LOCAL 10.0.0.202
SKYDESKTOP03.CORP.KKMS.LOCAL 10.0.0.203

DESCRIPTION

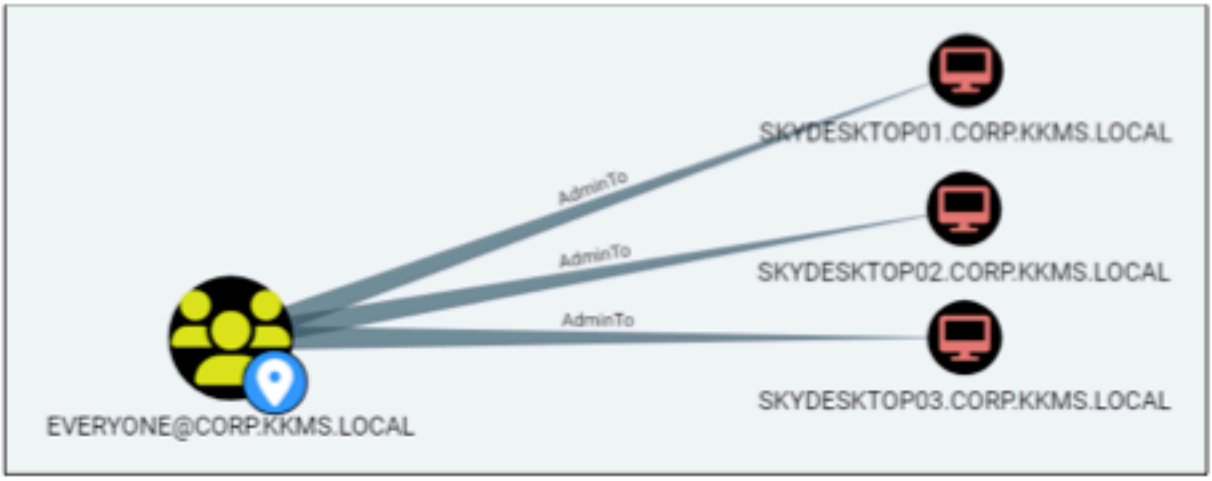
Any domain user can remotely control the affected servers with Administrator/SYSTEM level access.

BUSINESS IMPACT

Any domain user can be granted administrative privileges which could disrupt all employees' computers leading to downtime for RAKMS employees.

OBSERVATION

[redacted] discovered this vulnerability while performing enumeration of the CORP.KKMS.LOCAL domain with Bloodhound.



Bloodhound Output

STEPS TO REPRODUCE

1. Log on to the affected machine with any user

2. View Users and which group they are part of
3. Notice that all users are part of the 'Administrator' group

REMEDIATION

Remove the Everyone group from the Administrators group on the domain machines. Only add accounts/groups to remote access groups when strictly necessary for a business function.

REFERENCES

https://learn.microsoft.com/en-us/windows/security/identity-protection/access-control/local-accounts

PETITPOTAM

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:A/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H	7.5

AFFECTED SYSTEMS

SKYCONTROL01.CORP.KKMS.LOCAL - 10.0.0.5

DESCRIPTION

PetitPotam allows an attacker to force an authentication and intercepts the resulting certificate to achieve high-level access in the domain without the need of credentials.

BUSINESS IMPACT

Any attacker could take complete control over KKMS's domain leading to complete loss of airport functionality.

OBSERVATION

During testing, Finals-XX identified the SKYCONTROL01 domain controller was vulnerable to PetitPotam using the PetitPotam module from CrackMapExec.

PetitPotam validation via CrackMapExec

STEPS TO REPRODUCE

Execute “crackmapexec smb 10.0.0.5 -m petitpotam”

REMEDIATION

To mitigate these critical vulnerabilities and enhance the security of the domain controller and the entire network, Finals-XX recommends the following actions:

1. **Patch and Update**: Immediately apply security patches and updates provided by Microsoft to address the vulnerabilities.
2. **Segment the Network**: Isolate the domain controller from the rest of the network by implementing proper network segmentation and firewall rules to minimize lateral movement.
3. **Monitor and Audit**: Continuously monitor and audit the network for suspicious activities and implement robust logging mechanisms to detect potential security breaches.

REFERENCES

https://www.rapid7.com/blog/post/2021/08/03/petitpotam-novel-attack-chain-can-fully-compromise-windows-domains-running-ad-cs/

INSECURE SUDO CONFIGURATION

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:L/AC:L/PR:L/UI:R/S:U/C:H/I:H/A:H	7.3

AFFECTED SYSTEMS

TRAM1.TRAIN.KKMS.LOCAL - 10.0.20.101
TRAM2.TRAIN.KKMS.LOCAL - 10.0.20.102
TRAM3.TRAIN.KKMS.LOCAL - 10.0.20.103

DESCRIPTION

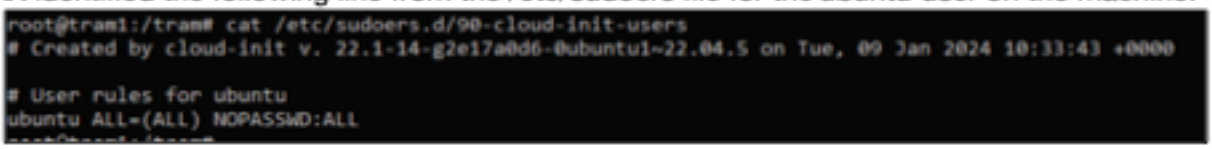
Sudo access without a password has been granted to the *ubuntu* user which allows it to run commands as root.

BUSINESS IMPACT

The Ubuntu user can fully remove functionality from the TRAM machines, which can result in disruptions to the tram service.

OBSERVATION

Finals-XX identified the following line from the /etc/sudoers file for the *ubuntu* user on the machine.



Sudo File

This means that the user can run any command as root, which allows for escalation of privileges in a variety of ways (e.g., run *sudo /bin/bash*). This allows an attacker to maintain elevated privileges.

STEPS TO REPRODUCE

1. Log onto one of the affected machines with the ubuntu user.
2. View the */etc/sudoers* file.
3. Notice that the 'Ubuntu' user can run all commands as root

REMEDIATION

To mitigate this vulnerability, Finals-XX recommends the following actions:

1. **Limit sudo permissions**: Limit sudo permissions exclusively to users whose job functions strictly require them.
2. **Implement least privilege**: Adhering to the principle of least privilege will reduce the likelihood of similar scenarios.

REFERENCES

https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-123.pdf
https://github.com/carlospolop/PEASS-ng

WEAK PASSWORD POLICY

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:L	7.3

AFFECTED SYSTEMS

10.0.0.5 - SKYCONTROL01.CORP.KKMS.LOCAL - svc_atc
10.0.0.5 - SKYCONTROL01.CORP.KKMS.LOCAL - edr_test
10.0.0.33 - BAGGAGECHECKIN.CORP.KKMS.LOCAL - admin
10.0.0.43 - EmployeeTimeDB.CORP.KKMS.LOCAL - admin
10.0.0.100 - AFWS.CORP.KKMS.LOCAL - Plaintext API Key
10.0.20.101 - Tram1.train.kkms.local - admin
10.0.20.102 - Tram2.train.kkms.local - admin
10.0.20.103 - Tram3.train.kkms.local - admin

DESCRIPTION

Weak password policies allow attackers to gain access to systems through brute-force attacks and can increase the attack surface when password hashes are cracked.

BUSINESS IMPACT

Weak passwords in a corporation lead to unauthorized access to machines on the network. This allows for data breaches and jeopardizes sensitive information.

OBSERVATION

Finals-XX found weak passwords used across the network. This includes domain credentials, web application passwords, and Linux passwords. Default Credentials were also found in the environment which violates PCI 3.2.1 regulations.

A full list of hosts cannot be verified with the time constraints of testing, but those confirmed to be vulnerable are listed as so.

STEPS TO REPRODUCE

Go through a list of user hashes and identify any domain users in common attacker databases. These are extremely weak and can be reversed quickly.

REMEDIATION

- Implement a strong, company-wide password policy that mandates higher-complexity passwords (i.e., include special characters, numbers, uppercase, and lowercase characters).
- Routinely check these password hashes to verify they cannot be cracked.
- Continue to provide training to individuals on why password strength is important.

REFERENCES

https://github.com/carlospolop/PEASS-ng
https://listings.pcisecuritystandards.org/documents/PCI_DSS-QRG-v3_2_1.pdf

SQL INJECTION IN TIMESHEET SITE

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:L/I:H/A:N	7.1

AFFECTED SYSTEMS

EmployeeTimeDB.corp.kkms.local - 10.0.0.43

DESCRIPTION

SQL Injection allows an attacker full access to databases. This also allows access to restricted file read and write on the host system.

BUSINESS IMPACT

An unauthenticated attacker can read and write to the SQL database, meaning all timesheets can be viewed or edited. This would affect how wages are paid. Confidential information, such as source code and passwords, could also be read.

OBSERVATION

Finals-XX discovered that the timesheet web application was vulnerable to a full SQL injection database leak. This was discovered with the tool sqlmap.

```
[10:26:21] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: employee (GET)
    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: employee=admin' AND (SELECT 2444 FROM (SELECT(SLEEP(5)))ypCd) AND 'fRgH'='fRgH&page=admin
---
```

The SQL statement that sqlmap found

This injection allowed Finals-XX access to all databases. In this case, it was observed that this injection also leads to restricted file read and write.

```
available databases [4]:
[*] employeedb
[*] information_schema
[*] mysql
[*] performance_schema
```

The Databases Found

STEPS TO REPRODUCE

1. Log in at 10.0.0.43
2. Replace the ‘employee’ field in the URL with the SQL statement found with sqlmap or SQL characters like an apostrophe to generate output
3. Verify that the variable is handled as SQL

REMEDIATION

Wherever SQL queries are used, prepared statements should be in place to separate user input from the query itself.
Implement input sanitization on both the server and client side. For the most part, usernames do not need to include special characters, such as apostrophes, dashes, and quotes.

REFERENCES

https://owasp.org/www-community/attacks/SQL_Injection

SERVICE ACCOUNT SPN USAGE

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:L	7.1

AFFECTED SYSTEMS

SKYCONTROL01.CORP.KKMS.LOCAL - 10.0.0.5

DESCRIPTION

A service account given a Service Principal Name (SPN) is vulnerable to its hash being extracted by an authenticated domain user.

BUSINESS IMPACT

Any attacker could take complete control over the svc_ATC account, leading to complete loss of airport functionality.

OBSERVATION

When looking for service accounts that are vulnerable to this attack, Finals-XX found the *svc_ATC* account had an SPN assigned which provided a hash.

Dumped hash

STEPS TO REPRODUCE

1. Request TGS for svc_ATC using any valid pair of domain credentials

REMEDIATION

- Do not grant SPNs to service accounts if they are not strictly needed.
- If SPNs are needed, give the account a strong enough password so the hash cannot be cracked

REFERENCES

https://owasp.org/www-pdf-archive/OWASP_Frankfurt_-44_Kerberoasting.pdf
https://attack.mitre.org/techniques/T1558/003/

AWS ROOT USER MISSING MFA

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:N/AC:H/PR:H/UI:N/S:U/C:H/I:H/A:H	6.6

AFFECTED SYSTEMS

RAKMS AWS Environment

DESCRIPTION

Multi-Factor Authentication is not enforced on the *root* AWS account.

BUSINESS IMPACT

If the password is somehow compromised, an attacker could log in without verifying their identity and make any changes they see fit, including removing any functionality as the root user.

OBSERVATION

An authenticated prowler scan by user *ctf-starting-user-27* found that the *root* AWS account does not enforce MFA. This is a large risk within the RAKMS infrastructure. MFA greatly reduces the risk of breaches and notifies the company when a log-in attempt has succeeded.

STEPS TO REPRODUCE

1. The root user, sign into the management console
2. Go to the Security Credentials section
3. Notice that MFA is not set up

REMEDIATION

Ensure all accounts are secured by implementing Multi-Factor Authentication in line with Amazon's best practices and the specific needs of the business.

REFERENCES

https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63-3.pdf
https://docs.aws.amazon.com/wellarchitected/latest/securitypillar/sec_identities_enforce_mechanisms.html

STORED XSS IN TIMESHEETS SITE

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:L	6.3

AFFECTED SYSTEMS

EMPLOYEETIME-DB.CORP.KKMS.LOCAL - 10.0.0.43

DESCRIPTION

Improper sanitization in the username variable allows an attacker to store potentially malicious JavaScript elements that run when a user accesses a page

BUSINESS IMPACT

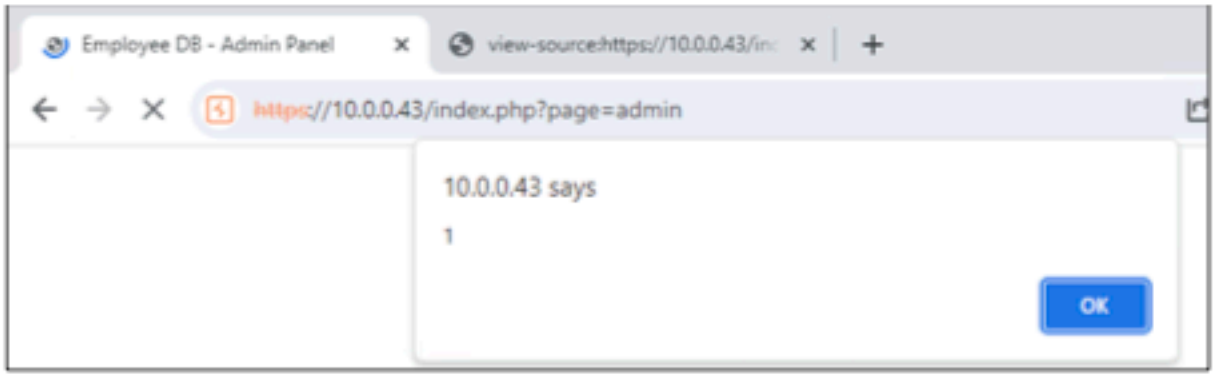
This vulnerability could be leveraged in multiple ways including defacing the timesheets site or stealing personal browser data resulting in loss of data confidentiality.

OBSERVATION

When adding a new user in the timesheets site, an XSS payload was used in the username variable and sent to the server.

Request with XSS payload

When accessing the site again, the payload ran.



Successful Exploitation

STEPS TO REPRODUCE

1. Send a post request to 10.0.0.43/index.php
2. Include "<script>alert(1);</script>" in the POST data being sent with the username variable
3. Go to 10.0.0.43/index.php to verify the payload worked

REMEDIATION

To mitigate this vulnerability, and preserve the integrity of the Timesheets Dashboard, Finals-XX recommends the following action:

1. **Input Sanitization**: Implement client-side and server-side sanitization on inputs provided by users for all inputs

REFERENCES

https://owasp.org/www-community/attacks/xss/

UNAUTHENTICATED API PERMITS ADDITION OF TRAMS

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:L	5.4

AFFECTED SYSTEMS

TRAM-OPS.TRAIN.KKMS.LOCAL - 10.0.20.100

DESCRIPTION

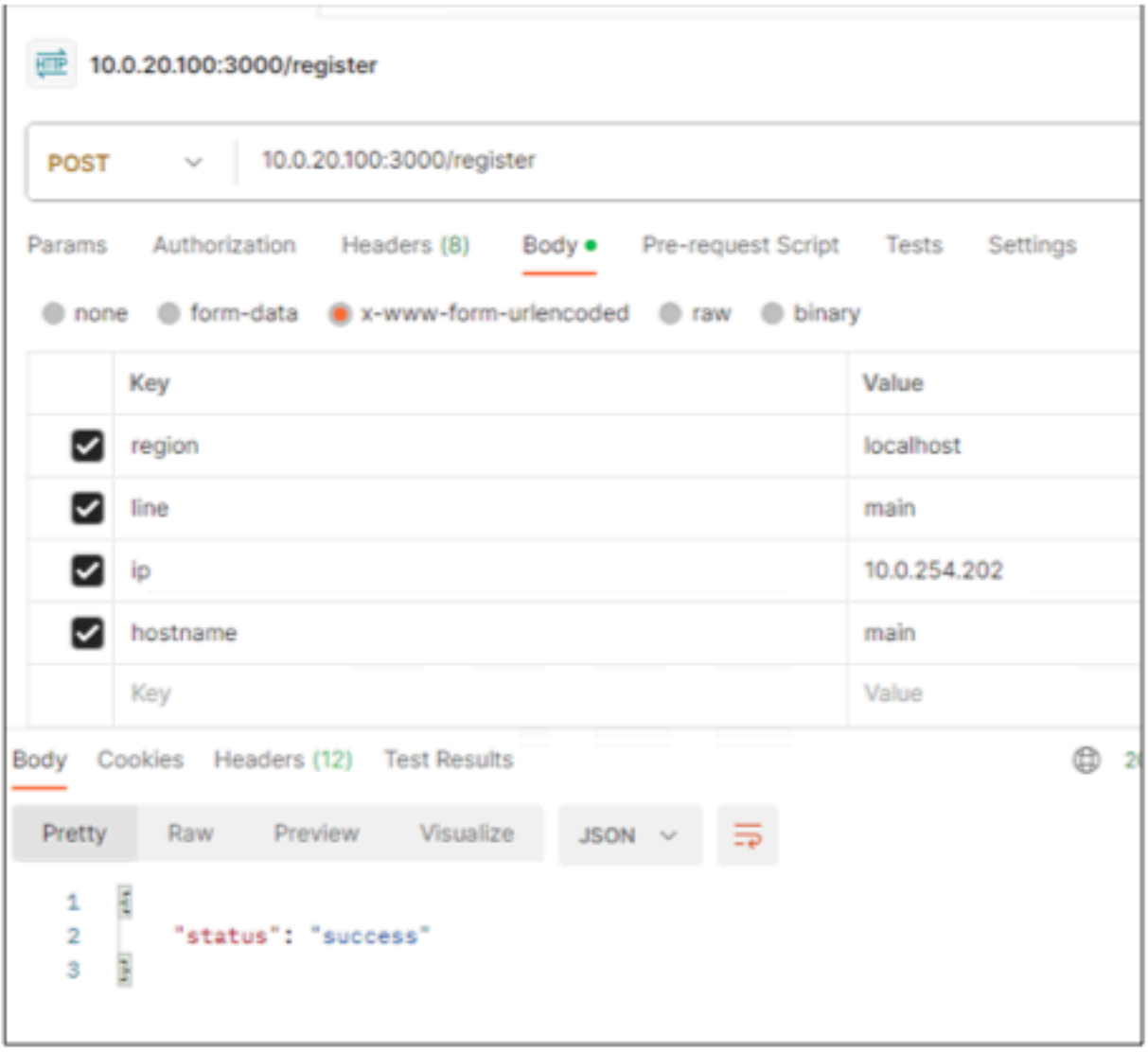
An unauthenticated API on the tram API allows an attacker to add trams.

BUSINESS IMPACT

Attackers can deface the website and stop valid employees from viewing the correct tram locations.

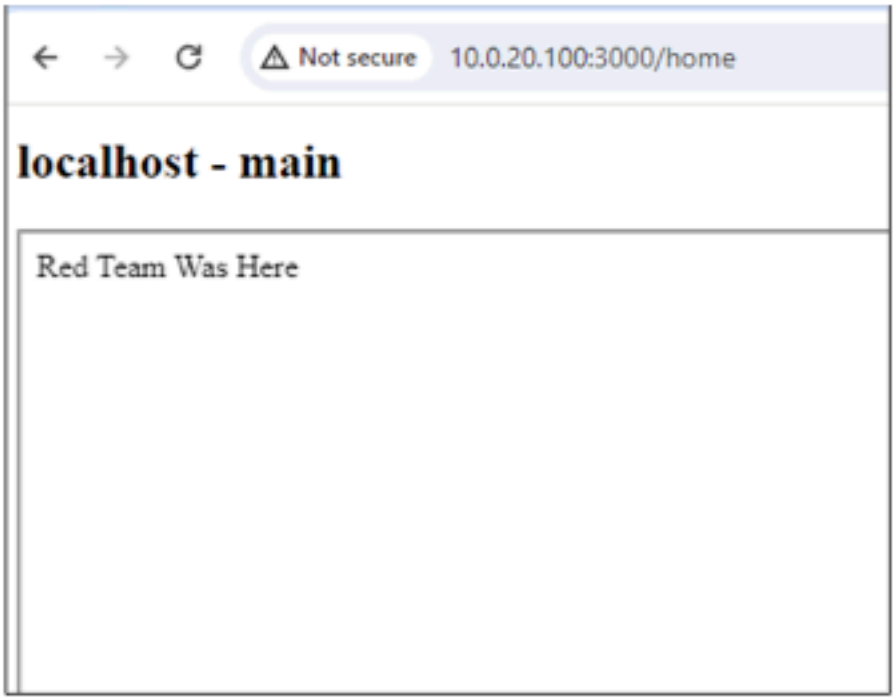
OBSERVATION

Finals-XX created a web request to the register endpoint, allowing a new tram to be registered by an unauthenticated attacker.



Initial Request

Finals-XX was able to add a malicious iframe.



Exploitation

STEPS TO REPRODUCE

1. Create a POST request to the *register* endpoint with any region, line, IP, and hostname value
2. Verify the request was successful on the *home* page

REMEDIATION

Implement authentication on the *register* endpoint to only allow authorized administrators.

REFERENCES

https://owasp.org/www-community/Broken_Access_Control

INSECURE KERBEROS ATTRIBUTE

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N	5.4

AFFECTED SYSTEMS

CORP.KKMS.LOCAL

DESCRIPTION

Accounts without Kerberos pre-authentication required attribute (DONT_REQ_PREAUTH) can have their hashes leaked by an unauthenticated user.

BUSINESS IMPACT

An unauthenticated hacker can retrieve an account hash to crack or pass to gain access to the domain as a relatively privileged user. This could result in long downtime for RAKMS.

OBSERVATION

When looking for service accounts that are vulnerable to this attack, Finals-XX found the *EDR_TEST* account was vulnerable

Dumped Hash

STEPS TO REPRODUCE

1. View accounts that have the Kerberos attribute DONT_REQ_PREAUTH
2. Verify the affected account has the listed attribute

REMEDIATION

Do not grant the DONT_REQ_PREAUTH if it is not strictly needed.

REFERENCES

https://learn.microsoft.com/en-us/defender-for-identity/security-assessment-unconstrained-kerberos

REAL TIME PROTECTION DISABLED

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:L/I:L/A:L	5.3

AFFECTED SYSTEMS

CORP.KKMS.LOCAL

DESCRIPTION

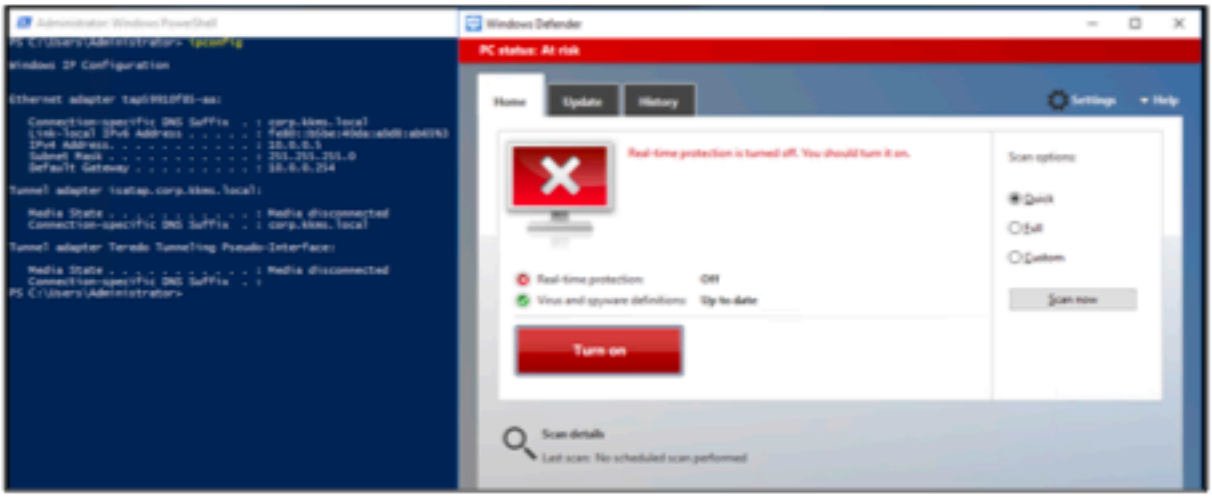
Antivirus (AV) is important for making sure any malicious files downloaded or brought onto a machine through other means do not execute. Not using AV brings unwanted risk onto an organization.

BUSINESS IMPACT

Attackers can execute malicious code without needing to bypass any AV which greatly increases the likelihood of exploitation.

OBSERVATION

Finals-XX verified that Defender was inactive on the domain.



Defender Disabled

STEPS TO REPRODUCE

1. Log into any domain joined machine
2. Open Windows Defender from the Windows search bar

REMEDIATION

Enable Windows Defender Real Time Protection by clicking the red "turn on" button.

REFERENCES

https://learn.microsoft.com/en-us/mem/intune/user-help/turn-on-defender-windows

IDOR ON BARCODE SYSTEM

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N	5.3

AFFECTED SYSTEMS

http://rakmsbarcode202401110348007218000000004.s3-website-us-east-1.amazonaws.com/
https://v6yqfrnhvs4dilwcdw2jaqsevi0vghev.lambda-url.us-east-1.on.aws/

DESCRIPTION

Any user can access barcodes from flight passes using IDOR.

BUSINESS IMPACT

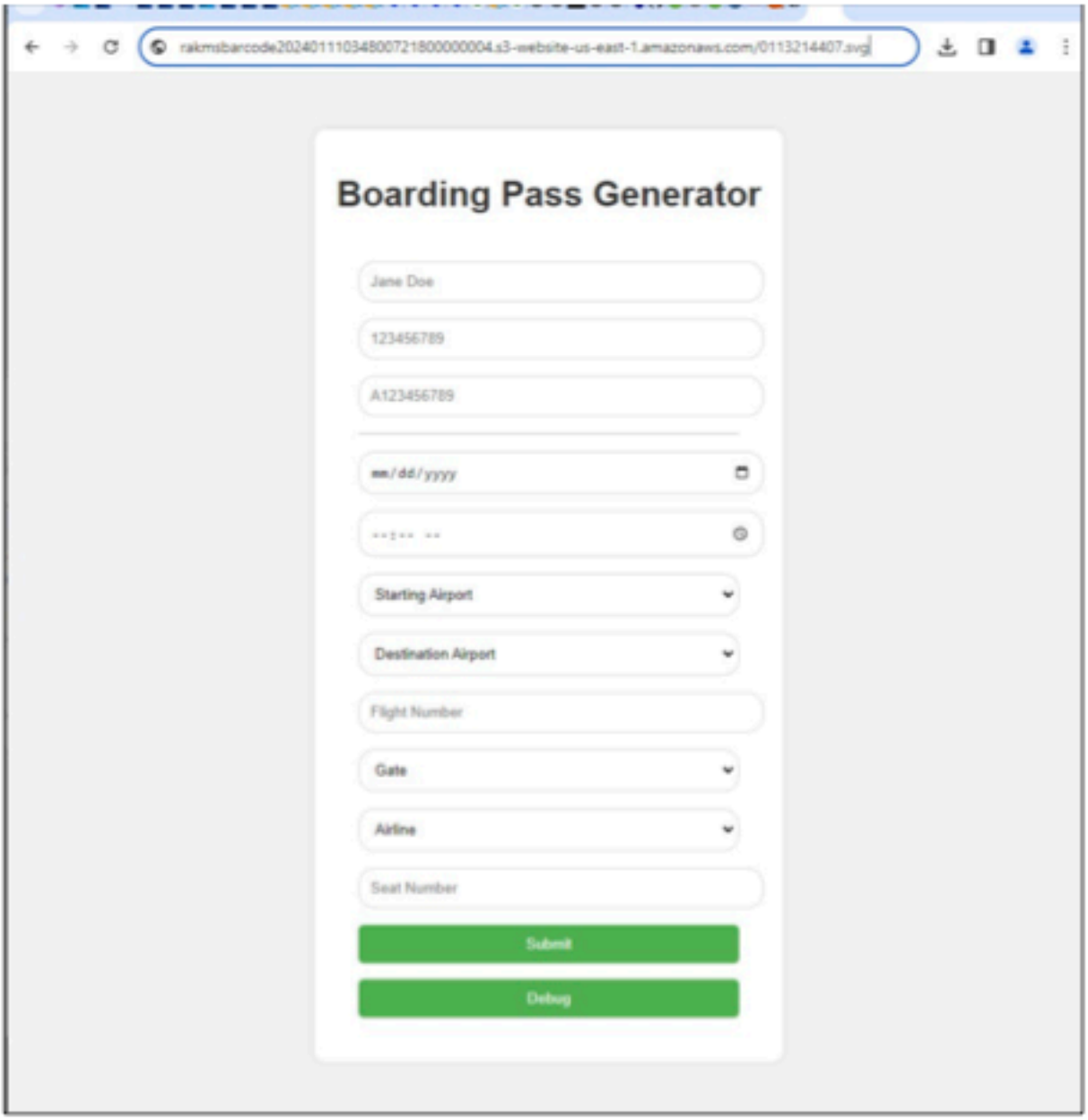
1. **Reputational Damage**: An IDOR vulnerability breach can severely damage a company's reputation, leading to lost customer trust and diminished brand value.

2. **Financial Losses**: Such a security breach can potentially result in substantial financial losses due to regulatory fines, costs of incident management, and compensatory measures for affected customers.

OBSERVATION

During the assessment, Finals-XX discovered that anyone may access the barcodes of flight passes.

Creating a boarding pass via a lambda function URL



Accessing the .SVG barcode object

Downloading, and viewing retrieved barcode

STEPS TO REPRODUCE

1. Navigate to https://v6yqfrnhvs4dilwcdw2jaqsevi0vghev.lambda-url.us-east-1.on.aws/
2. Manipulate the *?message=* parameter with any value to generate a barcode
3. Send the request, and note the JSON values returned from the web app.
4. Navigate to *http://rakmsbarcode202401110348007218000000004.s3-website-us-east-1.amazonaws.com/,* and add the value of the "path" key to the end of the URL. (ex: *http://rakmsbarcode202401110348007218000000004.s3-website-us-east-1.amazonaws.com/12345.svg)*
5. Send the request to download the respective *.SVG* file containing the barcode

REMEDIATION

To mitigate unauthorized access of sensitive customer data, Finals-XX recommends the following actions:

1. **Access Controls:** Upgrade Nginx to the most recent stable version available.
2. **Implement custom error pages**: Implementing custom error pages will help avoid the disclosure of version information.

REFERENCES

https://portswigger.net/web-security/access-control/idor

SMB NULL SESSION

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N	4.3

AFFECTED SYSTEMS

SKYCONTROL01.CORP.KKMS.LOCAL - 10.0.0.5

DESCRIPTION

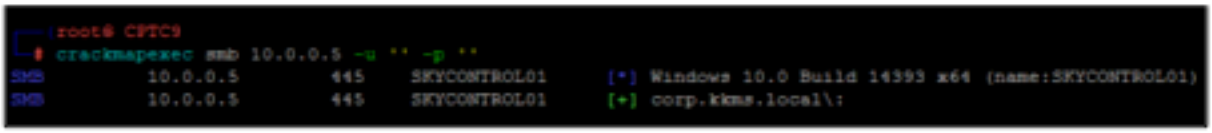
SMB null sessions allow attackers to authenticate to the domain without any credentials.

BUSINESS IMPACT

This configuration allows an attacker to gain access to company resources without any credentials, potentially leading to a loss in data confidentiality.

OBSERVATION

While performing enumeration with CrackMapExec, Finals-XX discovered one host using SMB null sessions.



CrackMapExec Output

STEPS TO REPRODUCE

1. Attempt to authenticate to any of the affected systems using null credentials.

REMEDIATION

Remove the possibility for NULL sessions by editing the relevant configurations via Group Policy.

REFERENCES

https://www.blumira.com/integration/how-to-disable-null-session-in-windows/

PHP INFO EXPOSED

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N	4.3

AFFECTED SYSTEMS

EmployeeTimeDB.corp.kkms.local - 10.0.0.43
TSA.guest.corp.kkms.local - 10.0.200.43

DESCRIPTION

A publicly accessible */info.php* page was identified during Finals-XX's assessment. The presence of a publicly accessible */info.php* page poses a potential security risk by divulging an extensive amount of information that could provide attackers with valuable insights, potentially aiding in identifying and exploiting vulnerabilities.

BUSINESS IMPACT

If left exposed, further updates to the web application could cause confidential information to be leaked in addition to the information already exposed which can be used to aid attackers in exploitation.

OBSERVATION

The */info.php* page hosts the output of the phpinfo() function, which generates a comprehensive report containing an abundance of data regarding the server's PHP environment, including PHP version, extensions, configurations, and more. This wealth of information can potentially expose vulnerabilities in components and configurations, making it easier for malicious actors to target and exploit weaknesses in the system.

PHP Version 7.4.3-4ubuntu2.19

System	Linux TSA.guest.kkms.local 5.4.0-113-generic #127-Ubuntu SMP Wed May 18 14:30:56 UTC 2022 x86_64
Build Date	Jun 27 2023 15:49:59
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.4/fpm
Loaded Configuration File	/etc/php/7.4/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/7.4/fpm/conf.d
Additional .ini files parsed	/etc/php/7.4/fpm/conf.d/10-mysqlnd.ini, /etc/php/7.4/fpm/conf.d/10-opcache.ini, /etc/php/7.4/fpm/conf.d/10-pdo.ini, /etc/php/7.4/fpm/conf.d/15-psr.ini, /etc/php/7.4/fpm/conf.d/15-xml.ini, /etc/php/7.4/fpm/conf.d/20-bcmath.ini, /etc/php/7.4/fpm/conf.d/20-calendar.ini, /etc/php/7.4/fpm/conf.d/20-ctype.ini, /etc/php/7.4/fpm/conf.d/20-curl.ini, /etc/php/7.4/fpm/conf.d/20-dom.ini, /etc/php/7.4/fpm/conf.d/20-exif.ini, /etc/php/7.4/fpm/conf.d/20-ffi.ini, /etc/php/7.4/fpm/conf.d/20-fileinfo.ini, /etc/php/7.4/fpm/conf.d/20-ftp.ini, /etc/php/7.4/fpm/conf.d/20-gd.ini, /etc/php/7.4/fpm/conf.d/20-gettext.ini, /etc/php/7.4/fpm/conf.d/20-iconv.ini, /etc/php/7.4/fpm/conf.d/20-intl.ini, /etc/php/7.4/fpm/conf.d/20-json.ini, /etc/php/7.4/fpm/conf.d/20-ldap.ini, /etc/php/7.4/fpm/conf.d/20-mbstring.ini, /etc/php/7.4/fpm/conf.d/20-mysqli.ini, /etc/php/7.4/fpm/conf.d/20-pdo_mysql.ini, /etc/php/7.4/fpm/conf.d/20-phar.ini, /etc/php/7.4/fpm/conf.d/20-posix.ini, /etc/php/7.4/fpm/conf.d/20-readline.ini, /etc/php/7.4/fpm/conf.d/20-shmop.ini, /etc/php/7.4/fpm/conf.d/20-simplexml.ini, /etc/php/7.4/fpm/conf.d/20-sockets.ini, /etc/php/7.4/fpm/conf.d/20-sysvmsg.ini, /etc/php/7.4/fpm/conf.d/20-sysvsem.ini, /etc/php/7.4/fpm/conf.d/20-sysvshm.ini, /etc/php/7.4/fpm/conf.d/20-tokenizer.ini, /etc/php/7.4/fpm/conf.d/20-xdebug.ini, /etc/php/7.4/fpm/conf.d/20-xmlreader.ini, /etc/php/7.4/fpm/conf.d/20-xmlwriter.ini, /etc/php/7.4/fpm/conf.d/20-xsl.ini
PHP API	20190902
PHP Extension	20190902
Zend Extension	320190902
Zend Extension Build	API320190902,NTS
PHP Extension Build	API20190902,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, convert.iconv.*

This program makes use of the Zend Scripting Language Engine:
Zend Engine v3.4.0, Copyright (c) Zend Technologies
with Zend OPcache v7.4.3-4ubuntu2.19, Copyright (c), by Zend Technologies
with Xdebug v2.9.2, Copyright (c) 2002-2020, by Derick Rethans

Screenshot depicting PHP Info

STEPS TO REPRODUCE

1. Browse to the /info.php endpoint on either of the affected systems.

REMEDIATION

To mitigate this information disclosure [redacted] recommends the following actions:

1. **Disable phpinfo()**: Disable the phpinfo() functionality

REFERENCES

https://docs.backdropcms.org/documentation/enabling-and-disabling-phpinfo

HARDCODED AUTH TOKEN

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:L	4.3

AFFECTED SYSTEMS

AFWS.CORP.KKMS.LOCAL - 10.0.0.100

DESCRIPTION

An attacker can create new Flights by exploiting a poor authentication implementation.

BUSINESS IMPACT

The impacts of an attacker being able to create new flights without prior authorization can be significant and pose several risks such as:

1. **Data Integrity and Accuracy:**
   - Unauthorized creation of flights can lead to inaccurate and unreliable data within the system.
2. **Financial Loss:**
   - If the system is used for managing bookings and ticket sales, unauthorized creation of flights could lead to financial losses. Fraudulent flights may be created, and tickets could be sold without proper payment, impacting revenue.
3. **Customer Experience:**
   - Customers may be affected by incorrect flight information or by the system being compromised.

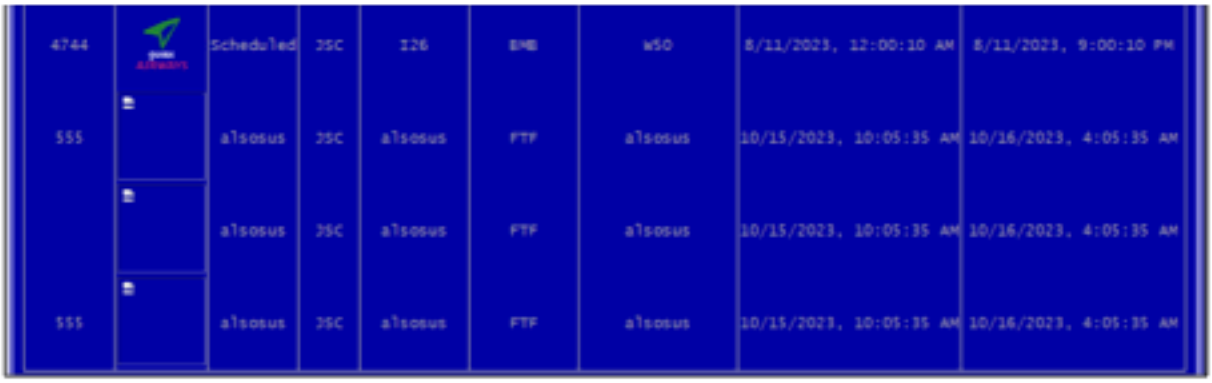
OBSERVATION

Finals-XX noticed a section of JavaScript that looks for an auth header which is hard-coded.

```
const xhr = new XMLHttpRequest();
xhr.open('GET', full_url, true);
xhr.setRequestHeader("Auth","FCKGW-RHQQ2-YXRKT-8TG6W-2B7Q8");
xhr.onreadystatechange = function (e) {
  if (xhr.readyState === 4 && xhr.status !== 200) {
    reject(xhr.status + " " + xhr.responseText);
  }
}
xhr.ontimeout = function () {
  reject('timeout');
}
xhr.onloadend = function (result) {
```

Hard-coded Auth Header in JavaScript

When the auth token is added, any user can add a new flight to the dashboard.



Added Flights

STEPS TO REPRODUCE

1. Make a post request to the *10.0.0.100/flight* endpoint
2. Include a JSON body with the following fields: *OriginIATA, DestinationIATA, AircraftType, FlightNumber, Status, OriginGate, DestinationGate*
3. Include the hardcoded auth header
4. Notice the response contains a newly created flight

REMEDIATION

To mitigate this vulnerability, Finals-XX recommends the following actions:
1. **Authentication**: Implement cryptographically secure authentication for the API endpoint to ensure that access is restricted to authenticated users only.

REFERENCES

https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-204.pdf

SENSITIVE INFORMATION IN S3 BUCKET DEV-S3-ROLE

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:N/A:N	4.3

AFFECTED SYSTEMS

Role:
ctf-starting-user accounts assuming the role of the dev-s3-role
Bucket:
This affects the kalka-passes20240111034800610800000003 bucket

dev-s3-policy

arn:aws:iam::677302527522:policy/dev-s3-policy


```
{
  "Statement": [
    {
      "Action": [
        "s3:Get*",
        "s3:List*"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::kalka-passes*"
      ]
    }
  ],
  "Version": "2012-10-17"
}
```


Attached Entities

- Roles
  - dev-s3-role

DESCRIPTION

Starting with the ctf-starting-user-4, Finals-XX was able to assume the role of dev-s3-role which had download access of the "kalka-passes*" bucket. This bucket contained many customer boarding tickets containing sensitive information.

BUSINESS IMPACT

This exposes sensitive customer passenger tickets stored in AWS and allows the attacker to exfiltrate the sensitive customer tickets.

OBSERVATION

Finals-XX used a ScoutSuite scan to find permissions in RAKMS AWS environment. Finals-XX then assumed the role of dev-s3-role which was able to dump the kalka-passes bucket.

STEPS TO REPRODUCE

1. use aws cli to configure access for the ctf-starting-user-4
2. Assume the dev-s3-role
3. download the s3 bucket by using aws cli sync
4. This bucket contained ~50 instances of customer airplane tickets

REMEDIATION

Either remove the Assume-Role permission for ctf-starting-user accounts on dev-s3-role or remove dev-s3-role's permission to dump the kalka-passes* database depending on which solution makes more sense in the environment. Ensure the principle of least privileges is followed when configuring access in AWS.

REFERENCES

https://rhinosecuritylabs.com/aws/aws-privilege-escalation-methods-mitigation/

SENSITIVE INFORMATION IN S3 BUCKET DEV-BARCODE-ROLE

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:N/A:N	4.3

AFFECTED SYSTEMS

Roles:
ctf-starting-user-4 assuming the roles of dev-barcode-role
S3 Bucket
arn:aws:s3:::rakmsbarcode20240111034800721800000004

dev-barcode-policy

arn:aws:iam::677302527522:policy/dev-barcode-policy


```
{
  "Statement": [
    {
      "Action": [
        "s3:List*",
        "s3:Get*"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::rakmsbarcode20240111034800721800000004",
        "arn:aws:s3:::rakmsbarcode20240111034800721800000004/*"
      ]
    }
  ],
  "Version": "2012-10-17"
}
```


Attached Entities

- Roles
  - dev-barcode-role

DESCRIPTION

Finals-XX was able to assume the roles of a higher privilege user and download a S3 bucket that contained sensitive passenger ticket barcodes that contained customer information.

BUSINESS IMPACT

This exposes the barcodes that are stored on the passenger tickets, as well as an AWS url that enables the AWS IDOR vulnerability listed in this report. These barcodes contain customer information such as names, time, gate, and flight.

OBSERVATION

Finals-XX used ScoutSuite to scan the AWS environment and find ways to escalate permissions. The dev-barcode-role was found to have access to download over the

STEPS TO REPRODUCE

1. Configure ctf-starting-user-4 for aws cli
2. using Pacu, assume_role of dev-barcode-role
3. use aws s3 sync to download this bucket: rakmsbarcode20240111034800721800000004

REMEDIATION

Follow the principle of least privilege when configuring access in AWS. Either remove the ability for the ctf-starting-user accounts to assume the dev-barcode-role account or remove dev-barcode-roles permission to download the bucket.

REFERENCES

https://docs.aws.amazon.com/prescriptive-guidance/latest/defining-bucket-names-data-lakes/handling-sensitive-data.html

INSECURE BAG CHECK-IN

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:A/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:L	4.2

AFFECTED SYSTEMS

BAGGAGECHECKIN.CORP.KKMS.LOCAL - 10.0.0.33

DESCRIPTION

An attacker can print out another person's baggage label if they know their name and flight.

BUSINESS IMPACT

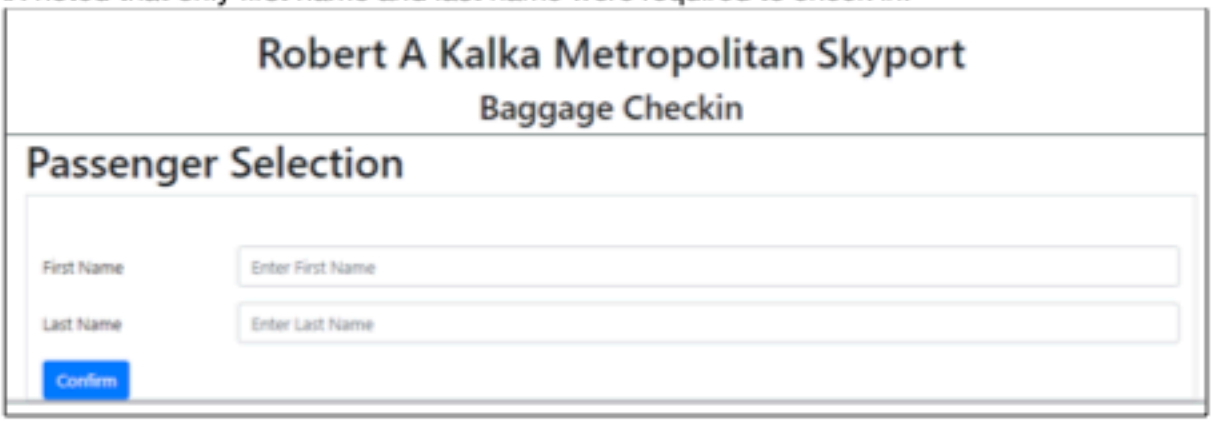
The disruption or alteration of baggage services could result in several consequences:

1. **Customer Dissatisfaction:** Inefficiencies in the baggage check-in process may frustrate customers, potentially leading to a decline in customer satisfaction.

2. **Need for Extra Personnel:** The demand for more staff to manage the heightened workload could strain resources, possibly affecting other airport operations and leading to further operational challenges.

OBSERVATION

Finals-XX noted that only first name and last name were required to check in.



Check-In Page

An attacker can print out the baggage tags if they know a person's name on a flight.

STEPS TO REPRODUCE

1. Access the baggage claim system, and enter a valid name, and flight time.
2. View the Customer's baggage check-in data.

REMEDIATION

To prevent unauthorized access to the baggage check-in system, Finals-XX recommends the following:

1. **Unique Confirmation Number:** Implement a system where each user is assigned a unique confirmation number. This number is sent to the user's registered email, enhancing security by requiring more than just the user's name for authentication.
2. **Email-Based Authentication:** Utilize the user's email as a key part of the authentication process. The unique confirmation number sent to their email ensures more secure and personalized access to the web portal.

REFERENCES

https://owasp.org/Top10/A01_2021-Broken_Access_Control/

AWS ASSUME ROLE PRIVILEGE ESCALATION

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:N/AC:L/PR:L/UI:R/S:C/C:L/I:N/A:N	4.1

AFFECTED SYSTEMS

Starting Users:

- ctf-starting-user-* (all of them)

Roles that can be assumed:

- dev-lambda-role
- secret_viewer
- secrets_viewer
- dev-s3-role
- dev-barcode-role
- dev2-lambda-role
- dev-lambda-bar-role
- dev1-role
- dev2-role

DESCRIPTION

Starting with the ctf-starting-user-4 credentials given to us during this assessment, Finals-XX was able to assume the roles and gain new privileges to 9 different roles in the RAKMS AWS environment.

BUSINESS IMPACT

This allows any of the low-level ctf-starting-user IAM users to elevate permissions and gain further access into AWS systems. A new employee or intern given one of these accounts or could be compromised and lead to a breach of sensitive data and systems in AWS.

OBSERVATION

Finals-XX ran a ScoutSuite scan using the original AWS credentials of ctf-starting-user-4 given to us for this engagement. Finals-XX was able to see all the roles in the RAKMS environment and which ones allowed all the ctf-starting-user accounts to assume them. From here, Finals-XX enumerated the 9 roles assumed and gained further sensitive data and access to the AWS environment.

STEPS TO REPRODUCE

1. Configure AWS credentials for ctf-starting-user-4 in the aws cli
2. Run the Pacu tool and then import_keys of the account credentials stored in the aws cli
3. Now, the user has assumed the new roles and permission of one of the 9 roles vulnerable

```
Pacu (new:imported-finals) > set_regions us-east-1
  Session regions changed: ['us-east-1']
Pacu (new:imported-finals) > assume_role arn:aws:iam::677302527522:role/dev1-role
AWS key is now new/arn:aws:sts::677302527522:assumed-role/dev1-role/assume-role.
Pacu (new:new/arn:aws:sts::677302527522:assumed-role/dev1-role/assume-role) >
```

Example assume_role escalation for the dev1-role

REMEDIATION

Ensure that the least privileges principle is followed when configuring user permissions within AWS. Verify if all the ctf-starting-user accounts need the ability to assume 9 other roles.

REFERENCES

https://docs.aws.amazon.com/STS/latest/APIReference/API_AssumeRole.html
https://rhinosecuritylabs.com/aws/assume-worst-aws-assume-role-enumeration/

IMPROPER INPUT VALIDATION

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:A/AC:L/PR:N/UI:R/S:U/C:L/I:N/A:N	3.5

AFFECTED SYSTEMS

TSA.GUEST.CORP.KKMS.LOCAL - 10.0.200.43

DESCRIPTION

Improper input validation allows servers to accept data that does not follow business logic, which can result in consequences during data manipulation steps later.

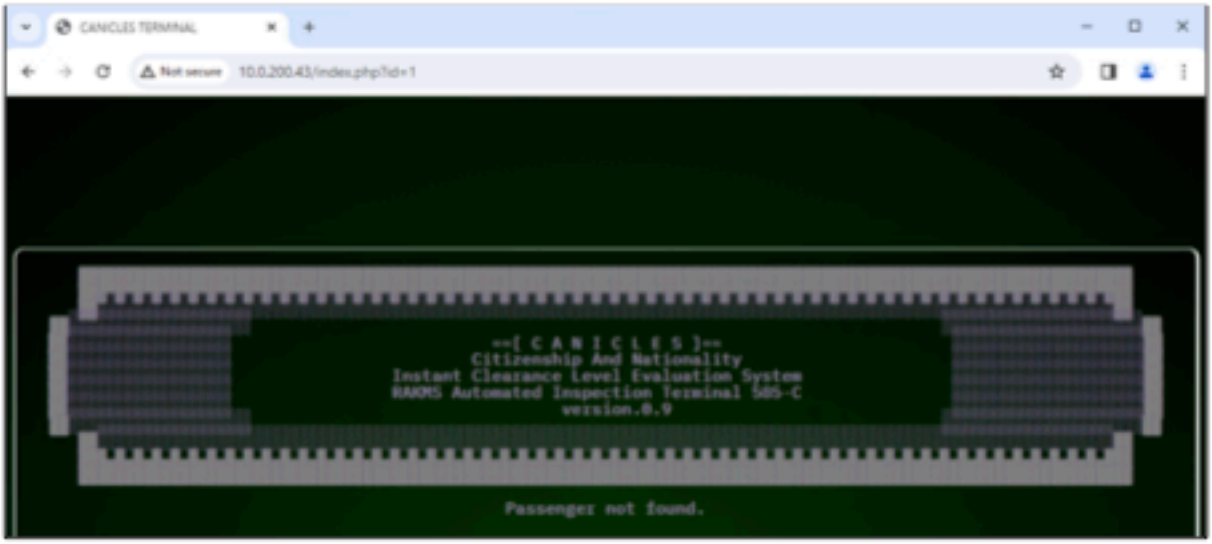
BUSINESS IMPACT

If the input is handled incorrectly, downstream components could behave in erratic ways, resulting in downtime and revenue loss.

OBSERVATION

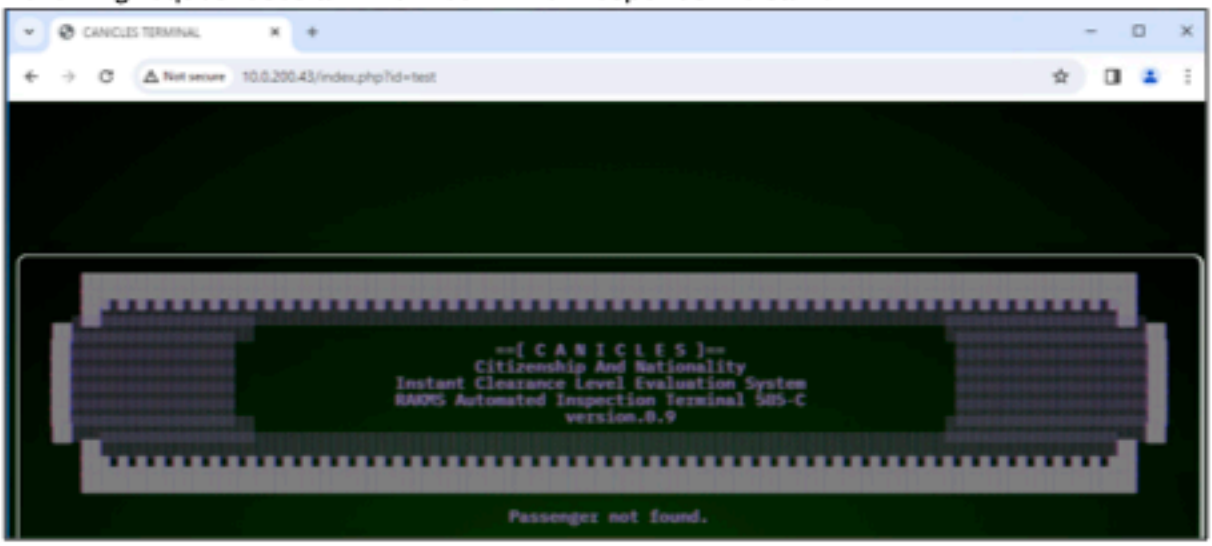
Input validation is an important part of web application security to make sure user input is not used maliciously such as in injection attacks.

Finals-XX found one web application that implements client-side validation to verify that only integers are entered in an ID field. This is done by refreshing the page whenever a non-numeric key is pressed.



Request with Numeric ID

And the following request uses an ID of ‘test’ which responds the same.



Request with Non-Numeric ID

While it cannot be confirmed, Finals-XX asserts that it is likely this data was accepted by the back-end server. The team cannot confirm how the input is handled, so the finding is ranked accordingly.

STEPS TO REPRODUCE

1. Intercept a request sent to the endpoint with a numeric ID
2. Change the ID to be non-numeric
3. Send the packet
4. Verify the web application accepts the request similarly to when it accepts a numeric ID.

REMEDIATION

A better method to validate user input such as regular expression or if-statements in the front-end JavaScript should be used. Ensure that the backend is also validating input as well using proper logic control when handling the user input.

REFERENCES

https://cwe.mitre.org/data/definitions/20.html
https://cheatsheetseries.owasp.org/cheatsheets/Input_Validation_Cheat_Sheet.html

AWS SECRETS FOUND

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:N/AC:L/PR:L/UI:R/S:U/C:N/I:L/A:N	3.5

AFFECTED SYSTEMS

Roles:

- dev1-role
- dev2-role
- secret_viewer
- secrets_viewer

SSM Keys

- arn:aws:ssm:us-east-1:677302527522:parameter/target/dev/thingy1
- arn:aws:ssm:us-east-1:677302527522:parameter/target/dev/thingy2
- arn:aws:ssm:us-east-1:677302527522:parameter/target/password/another-secret
- arn:aws:ssm:us-east-1:677302527522:parameter/testdeploy/password/secrets

DESCRIPTION

Starting with the ctf-starting-user-4, Finals-XX was able to assume the roles of secret_viewer, secrets_viewer, dev2-role, and dev1-role. Each of these roles had access to a single aws ssm secure string which was decrypted with the aws cli. Finals-XX was unable to use these secrets to gain further access in the AWS environment.

BUSINESS IMPACT

This could lead to further access into the AWS environment, and potentially lead to sensitive customer data being breached. An attacker could also be able to use the keys to potentially interact with sensitive RAKMS systems hosted in AWS depending on what they are for.

OBSERVATION

Finals-XX used the Pacu tool for our AWS testing. Finals-XX imported the keys for ctf-starting-user-4 which were given to us, then Finals-XX assumed each of the 4 roles previously specified. Our team also ran a ScoutSuite scan to enumerate the permissions each role had. Using the scan information, Finals-XX was able to use the aws cli to see and decrypt a Secure-String for each role.

STEPS TO REPRODUCE

1. Use the Pacu tool and import-Keys of ctf-starting-user-4
2. Assume the role of one of the 4 roles such as secret_viewer
3. Use the respective ssm key and obtain it using this command

Encrypted SecureString

4. Use the –decrypt-key
5. Repeat for the other 3 vulnerable roles

REMEDIATION

If these Secure-Strings are supposed to be private or allow further access into the AWS environment, ensure that the ctf-starting-users are not able to assume any of these 4 roles or remove the permission for each of these roles to have access to an ssm secret.

REFERENCES

https://cloud.hacktricks.xyz/pentesting-cloud/aws-security/aws-privilege-escalation/aws-ssm-privesc

DIRECTORY LISTING ENABLED

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:N	0.0

AFFECTED SYSTEMS

AFWS.CORP.KKMS.LOCAL - 10.0.0.100:7000/assets

DESCRIPTION

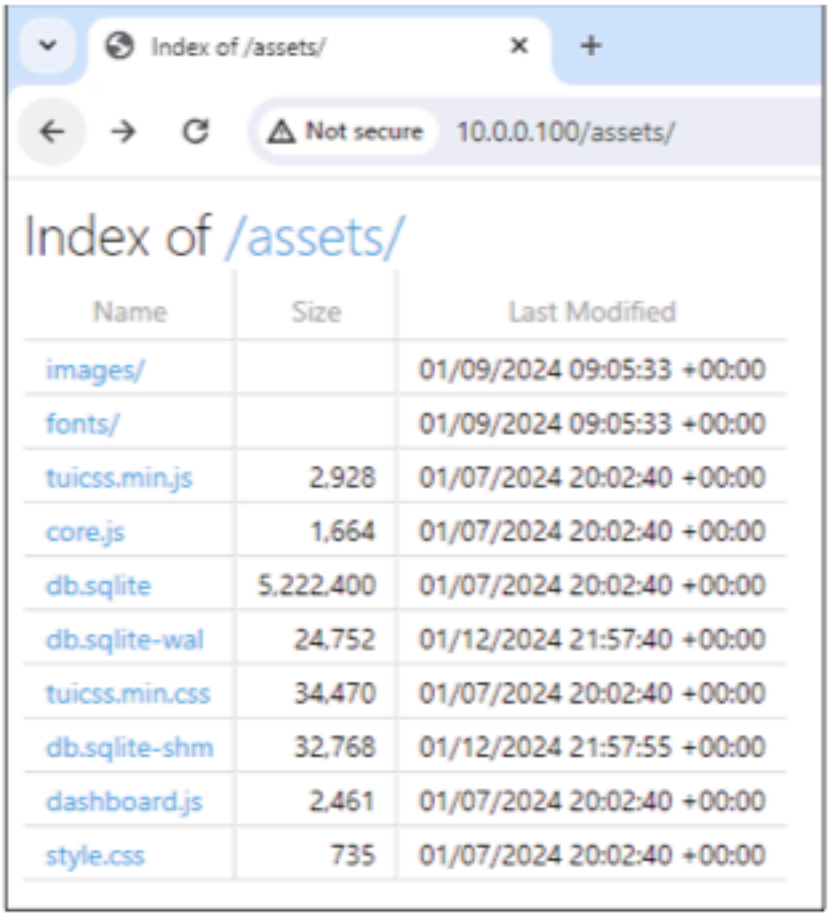
Directory listing allows an attacker to view files or filenames on the local machine in a web browser.

BUSINESS IMPACT

An attacker has more information on the frameworks and technologies in use, making subsequent attacks more likely.

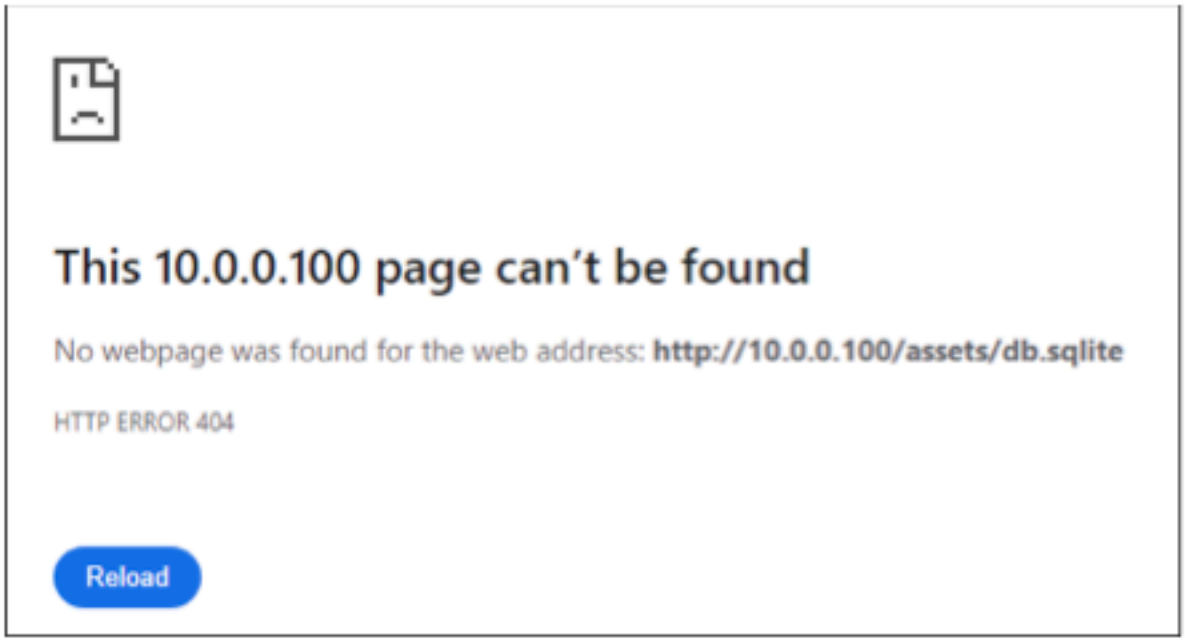
OBSERVATION

Finals-XX discovered a host that had directory listing enabled for the *asset's* directory. This disclosed the following filenames.



Directory Listing

The JavaScript and CSS files could be viewed in the browser while the SQLite data files could not. The severity has been lowered accordingly.



Database File

STEPS TO REPRODUCE

1. Go to the URL at 10.0.0.100/assets
2. Notice the list of files/directories in the assets folder being displayed

REMEDIATION

Disable directory listing in the web application and move the database files to a different more secure folder if possible.

REFERENCES

https://cwe.mitre.org/data/definitions/548.html

EOL NGINX VERSION

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:N	0.0

AFFECTED SYSTEMS

TRAM1.GUEST.KKMS.LOCAL - 10.0.200.101
TRAM2.GUEST.KKMS.LOCAL - 10.0.200.102
TRAM3.GUEST.KKMS.LOCAL - 10.0.200.103

DESCRIPTION

An End-of-life version of Nginx (1.18.0) was discovered during the assessment. End-of-life software is no longer supported and will not receive any patches. As time goes on the likelihood this software is exploited greatly increases.

BUSINESS IMPACT

In the event of an exploit being found for this NGINX version, attackers may severely disrupt tram operations by manipulating control systems, such as altering tram speeds, posing serious safety risks and potential financial losses.

OBSERVATION

Finals-XX discovered the End-of-life version of Nginx (1.18.0) via a 404 Not Found error page.

Error Page Depicting Nginx Version

STEPS TO REPRODUCE

1. Navigate to the IP address of the affected systems
2. Notice the *404 not found* error page, revealing the EOL Nginx version (1.18.0)

REMEDIATION

To mitigate potential vulnerabilities and enhance the security of the Nginx server, Finals-XX recommends the following actions:

1. **Patch and Update**: Upgrade Nginx to the most recent stable version available.
2. **Implement custom error pages**: Implementing custom error pages will help avoid the disclosure of version information.

REFERENCES

https://vuldb.com/?ctiid.155282

SSH PERMITROOTLOGIN

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:N	0.0

AFFECTED SYSTEMS

EMPLOYEETIME-DB.CORP.KKMS.LOCAL - 10.0.0.43

DESCRIPTION

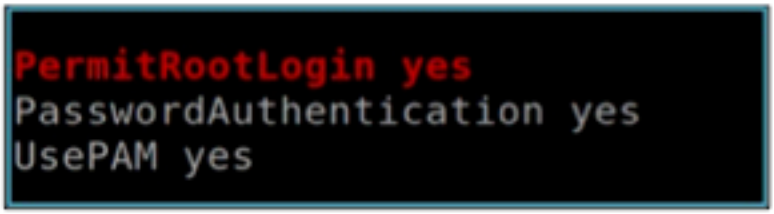
In the event that root credentials are disclosed, an attacker may log into this system remotely via SSH.

BUSINESS IMPACT

If an attacker were to gain root credentials, they would be able to access the system with the highest level of permissions remotely leading to severe disruption of business operations and leaking of personal information.

OBSERVATION

During a vulnerability assessment, a specific configuration setting was detected within the /etc/ssh/sshd_config file. This configuration, which permits root user login via SSH, poses a security concern.



SSH Config, permit root login

STEPS TO REPRODUCE

1. View PermitRootLogin value within /etc/ssh/sshd_config on affected system.

REMEDIATION

To mitigate this vulnerability, Finals-XX recommends the following actions:
1. **Disable Root Login**: To restrict root user login, disable *PermitRootLogin* in the */etc/ssh/sshd_config* file. Finals-XX strongly recommends only employing this configuration when strictly necessary for critical business functions.

REFERENCES

https://www.tenable.com/audits/items/CIS_Red_Hat_EL7_STIG_v2.0.0_L1_Server.audit:464e688c7f15c8990e6308faf2ab79c1

PUBLICLY ACCESSIBLE S3 BUCKET

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:A/AC:L/PR:N/UI:R/S:U/C:N/I:N/A:N	0.0

AFFECTED SYSTEMS

http://rakmstoolrequisition20231009225108646300000003.s3-website-us-west-2.amazonaws.com/

DESCRIPTION

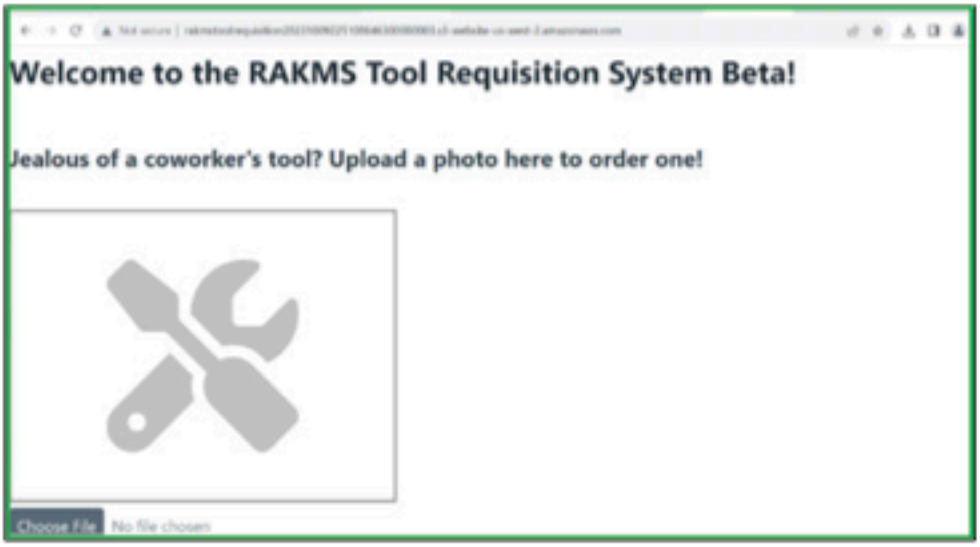
It is considered best practice to host beta applications in a pre-production environment and not be accessible to anyone on the internet due to the increased risk of vulnerabilities in unfinished applications.

BUSINESS IMPACT

This site being open to the internet with no authentication allows anyone to purchase and order a tool from RAKMS. An attacker could maliciously place orders which could cause financial losses if left unchecked.

OBSERVATION

This S3 bucket is hosted in the RAKMS AWS environment and can be visited and interacted with by anyone. This could potentially lead to exploitation and is an unnecessary risk in the environment.



AWS hosted webserver.

STEPS TO REPRODUCE

1. Visit the S3 AWS website from an external connection. You are still able to access this site and make orders without authentication.

REMEDIATION

RAKMS should restrict public access to this s3 bucket as it is only for internal use. The AWS administrator would need to change the block public access settings in the AWS console to restrict access to employees only. An IAM policy that only traffic from whitelisted IP ranges can access this application could also be implemented.

REFERENCES

https://docs.aws.amazon.com/AmazonS3/latest/userguide/access-control-block-public-access.html

UNENCRYPTED EBS VOLUMES IN AWS

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:N	0.0

AFFECTED SYSTEMS

AWS EBS Volumes

DESCRIPTION

It is considered best practice to encrypt databases, to reduce the risk of unauthorized access to information. An unencrypted database is a much higher risk to have in the environment than an encrypted one.

BUSINESS IMPACT

This could open up the potential for privilege escalation attacks or a breach of sensitive data depending on what is stored on the EBS volumes. Widens the scope of attack against RAKMS if a low-level AWS user is compromised.

OBSERVATION

As the ctf-starting-user-27 which was given to [redacted] by Ted Striker, Finals-XX was able to see and download 4 unencrypted EBS volume snapshots. Unfortunately, due to network and time limitations, Finals-XX was unable to verify the specific contents of these volumes. If there is sensitive information housed inside of these drives, a low-level malicious user in the AWS environment could be able to read and exfiltrate sensitive information. Finals-XX used the Pacu tool to obtain these snapshots.

Unencrypted EBS snapshots

	A	B	C	D	E
1	Volume Name	Volume ID	Region		
2		vol-0cdf0c903ca1c15aa	us-east-1		
3		vol-0ea9809833dd8c43a	us-east-1		
4		vol-0997e2787dfe697a3	us-east-1		
5		vol-04b03c3b69792c4e1	us-east-1		
6					
7					

EBS Volume ID's

STEPS TO REPRODUCE

1. add ctf-starting-user-27 credentials into Pacu configure
2. run ebs__download_snapshots volume in Pacu

REMEDIATION

It is recommended to enable encryption on EBS volumes by default. This can be set up within the AWS console as an administrator (See References).

REFERENCES

https://aws.amazon.com/blogs/compute/must-know-best-practices-for-amazon-ebs-encryption/
https://rules.sonarsource.com/cloudformation/RSPEC-6275/
https://github.com/RhinoSecurityLabs/pacu

UPCOMING ACM CERTIFICATE EXPIRATION

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:N	0.0

AFFECTED SYSTEMS

cptc-aws-vpn.cyberrange.rit.edu

DESCRIPTION

When an ACM (Amazon Certificate Manager) digital certificate expires, it can lead to a loss of secure HTTPS connections and potentially expose sensitive data to interception or compromise due to the lack of valid encryption on the associated websites or services.

BUSINESS IMPACT

There are various impacts expired ACM certificates can have on the business:

1. **Operational Disruption and Downtime**: Expired ACM certificates can cause critical systems to fail, leading to service outages and impacting customer access.
2. **Security Concerns**: With expired certificates, communication between servers and clients becomes insecure, increasing the risk of data breaches and breaking user trust.

OBSERVATION

ACM Certificate 3c979ef1-6285-477b-83f1-f51e81fc1c4d for cptc-aws-vpn.cyberrange.rit.edu is about to expire in 7 days. Ensure that this does not expire as this could lead to regulation issues (see references). The Prowler tool was used with ctf-starting-user-27 permission to find this.

STEPS TO REPRODUCE

1. Run Prowler on the AWS instance, with the provided *ctf-starting-user-27* user
2. Notice the finding stating the upcoming expired Certificate

REMEDIATION

Follow the AWS documentation in the references to manage ACM certificate renewal.

REFERENCES

https://docs.aws.amazon.com/acm/latest/userguide/managed-renewal.html

DEV VARIABLE ALLOWS UNLIMITED SESSION LIMIT

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:N	0.0

AFFECTED SYSTEMS

BAGGAGECLAIM.CORP.KKMS.LOCAL - 10.0.0.33

DESCRIPTION

Due to insecure JavaScript, an attacker can go through the baggage check-in process without creating a session.

BUSINESS IMPACT

An attacker will not need to create a new session, meaning they are more able to perform other exploits more easily.

OBSERVATION

When analyzing the homepage, Finals-XX identified a segment of code that allows an attacker to skip the need for a session by setting the JavaScript variable 'dev' to true which will no longer invalidate the session token in use.

```
<script>
    dev = false
    function fetchData() {
        if (dev == true) {
            return
        }

        currentURL = window.location.href
        if (currentURL.endsWith('/kiosk/go/')) {
            return
        }


        fetch('/api/v3/session/heartbeat')
            .then(response => {
                if (!response.ok) {
                    window.location.href = '/api/v3/session/destroy';
                }
            })
    }
```

Dev Code

It appears this is a remnant from the site in development.

STEPS TO REPRODUCE

1. Go to the site at BAGGAGECLAIM.CORP.KKMS.LOCAL
2. View the source of the site and locate the JavaScript section
3. Notice a check for a variable called 'dev'
4. Set the variable to true
5. Notice the session no longer expires

REMEDIATION

Remove this development segment of code from the program for production.

REFERENCES

https://owasp.org/Top10/A01_2021-Broken_Access_Control/

EOL RUBY VERSION

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:N	0.0

AFFECTED SYSTEMS

TRAM-OPS.TRAIN.KKMS.LOCAL - 10.0.20.100

DESCRIPTION

An end-of-life version of Ruby (2.7.8) was discovered during the assessment. End-of-life software is no longer supported and will not receive any patches. As time goes on the likelihood this software is exploited greatly increases.

BUSINESS IMPACT

In the event of an exploit being found for this Ruby version, attackers may severely disrupt tram operations by manipulating control systems, such as altering tram speeds, posing serious safety risks and potential financial losses.

OBSERVATION

[redacted] discovered the End-of-life version of Ruby via the homepage of the web application.

Homepage Version Numbers

STEPS TO REPRODUCE

1. Go to the affected system's homepage
2. Verify that the version of Ruby is EOL

REMEDIATION

To mitigate potential vulnerabilities and enhance the security of the Ruby on Rails server, Finals-XX recommends the following actions:

1. **Patch and Update**: Upgrade Ruby to the most recent stable version available.
2. **Implement custom home page**: Implementing custom home pages will help avoid the disclosure of version information.

REFERENCES

https://www.ruby-lang.org/en/downloads/branches/

UNUSUAL DISCOVERIES ON AWS ENVIRONMENT

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:N	0.0

AFFECTED SYSTEMS

AWS Environment

DESCRIPTION

RAKMS staff mentioned that their AWS environment had been inherited from another company. The following is a list of informational, but notable findings that may be worth looking into.

BUSINESS IMPACT

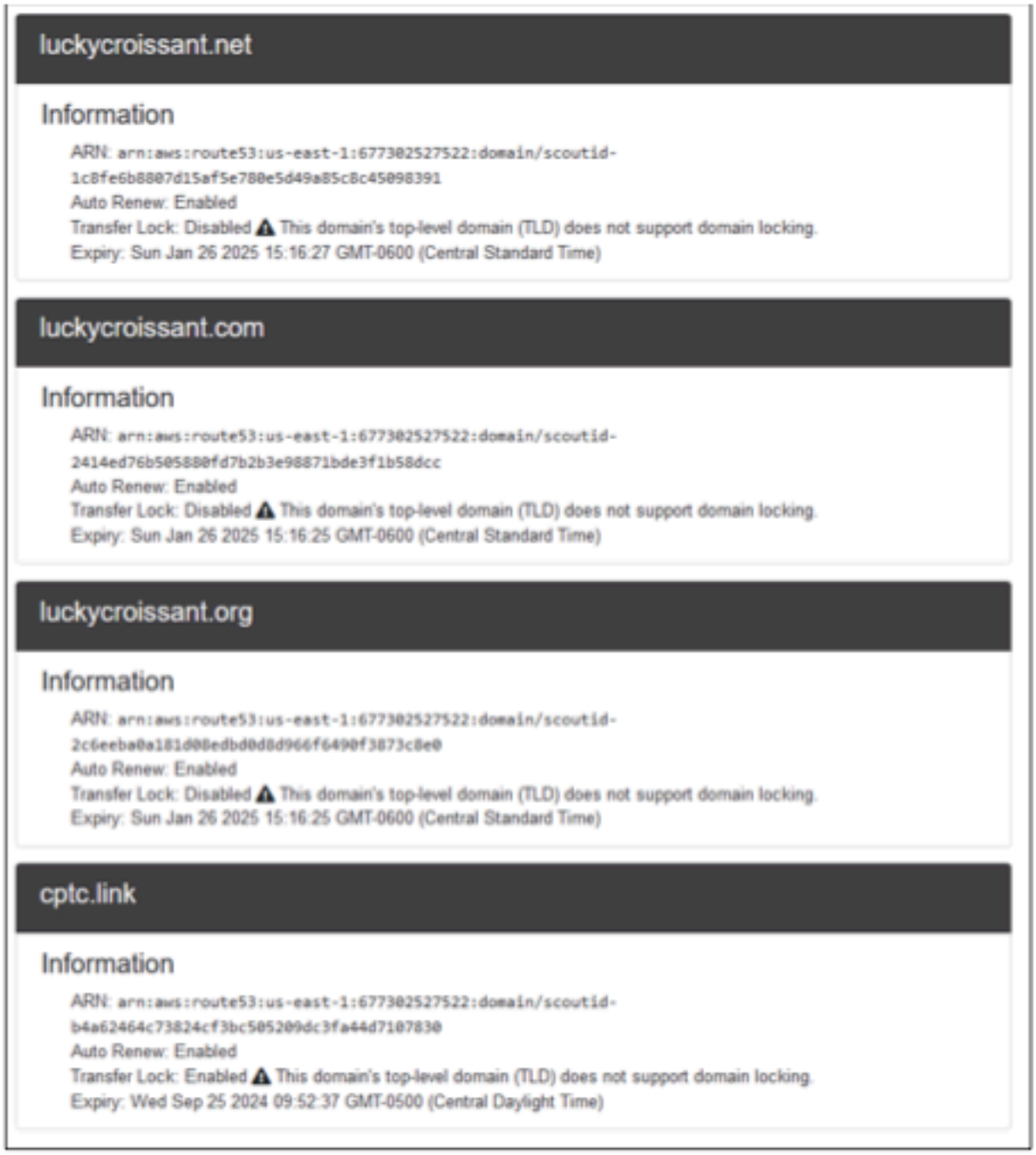
1. **Increased Security Risks**: Neglecting to remove unused resources and outdated ACLs/groups can lead to vulnerabilities, making the system susceptible to breaches.
2. **Operational Inefficiencies**: Overlooking these aspects can result in a cluttered and inefficient system, hindering performance and increasing maintenance complexity.

OBSERVATION

1. EC2 Security Groups
    - Various EC2 security groups (launch-wizard-*) allow access to various ports. Additionally, these security groups were in an unused state during the assessment.

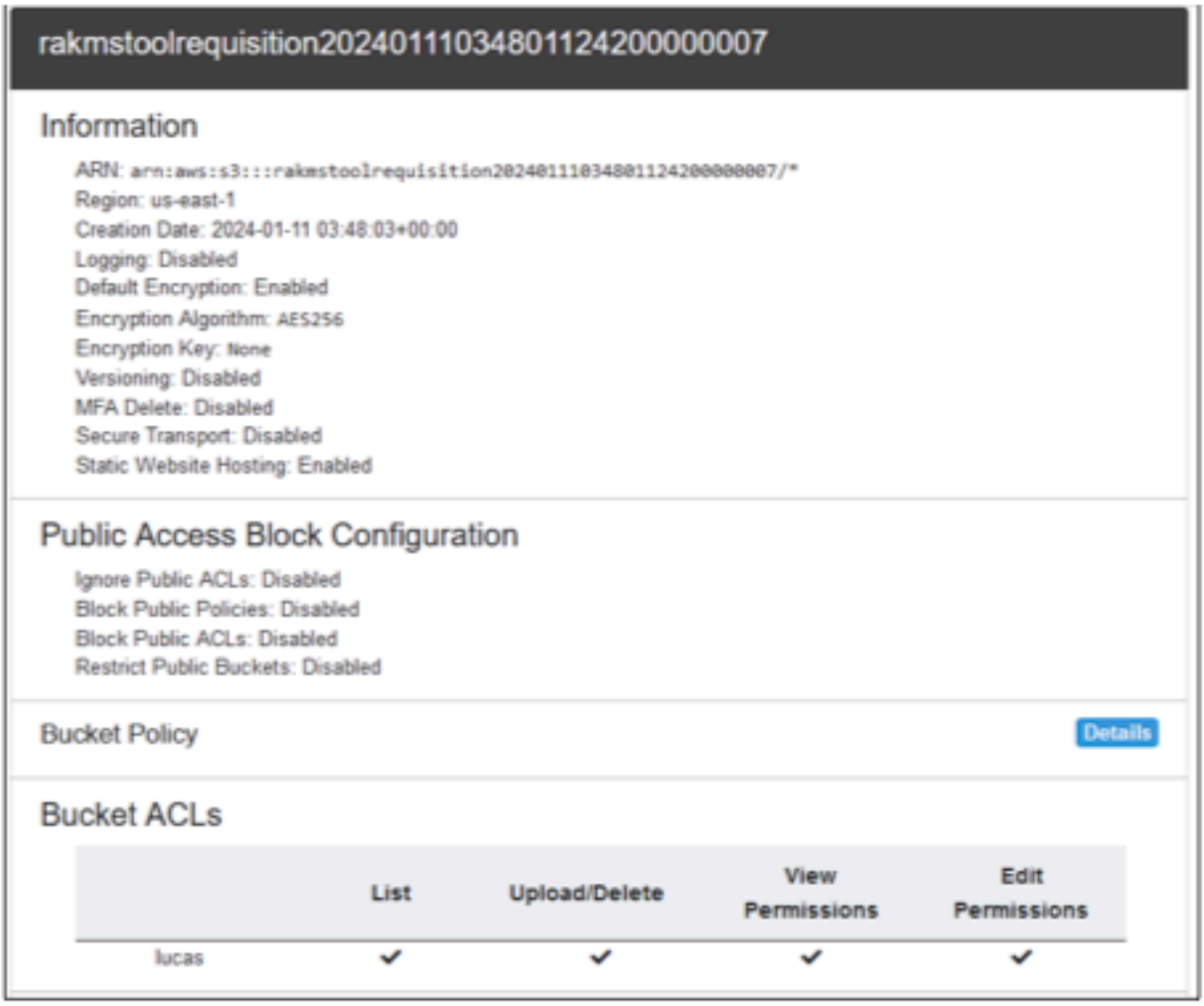
AWS Security Group example

2. DNS
    - Seemingly random DNS records were found via the Route53 service.

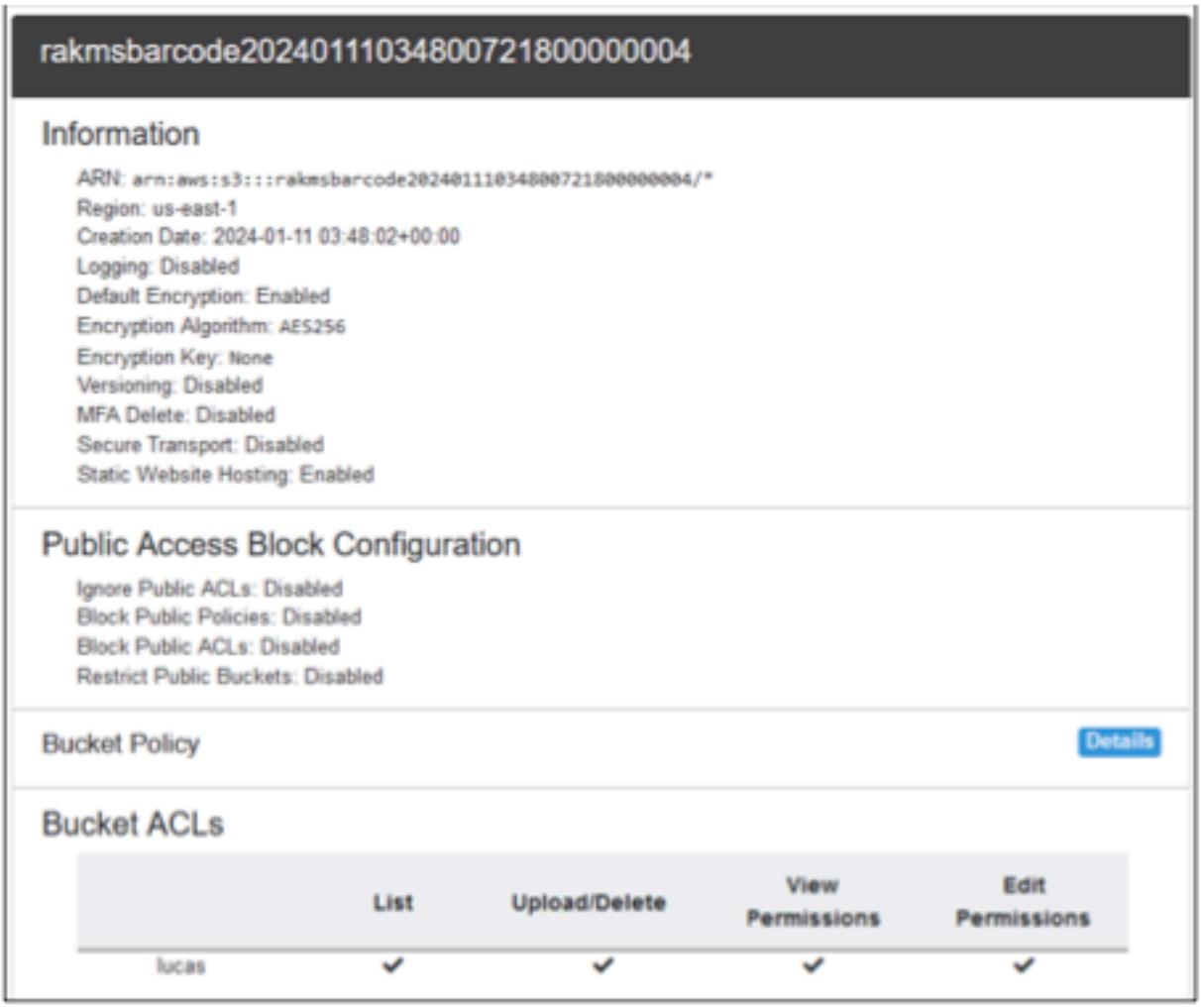


DNS Domains

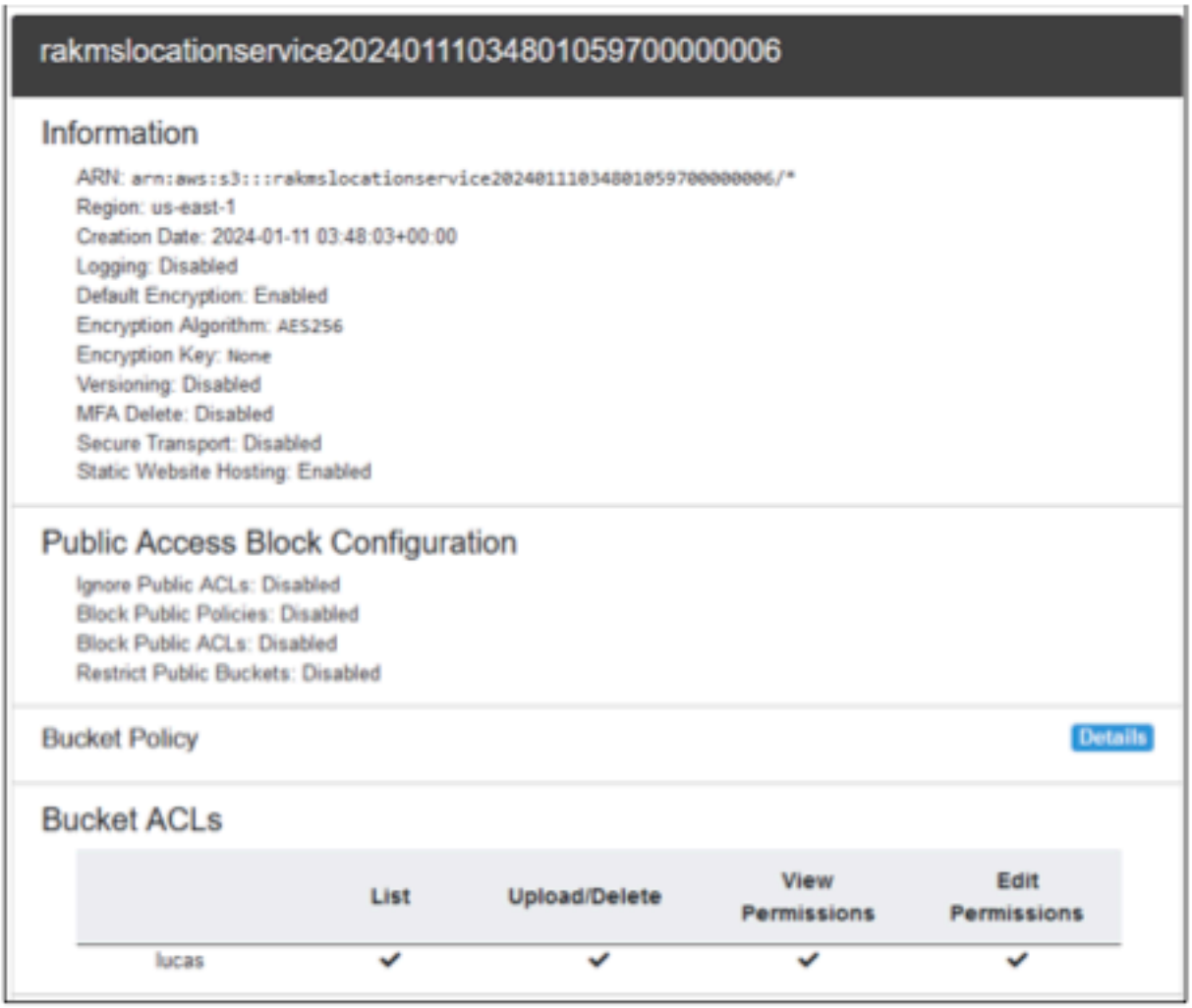
- S3 Bucket ACL's & Ownership
- All of the S3 Buckets observed were owned, and full access was given to the *lucas* user. This username does not seem to fit in with other users.



S3 Bucket ACL's



S3 Bucket ACL's



S3 Bucket ACL's

STEPS TO REPRODUCE

1. Run an authenticated Scout Suite scan or similar vulnerability scan on the AWS environment.
2. Review & observe findings

REMEDIATION

Finals-XX recommends the following actions:

1. **Delete Unused Resources:** Regularly deleting unused resources helps prevent accidental usage or misconfiguration, maintaining a more efficient and manageable system environment.

2. **Review Access Controls:** Regularly review and update access controls to ensure ongoing security and compliance with policies.

REFERENCES

https://docs.aws.amazon.com/vpc/latest/userguide/vpc-security-groups.html

RAKMS BAGGAGE CLAIM ENVIRONMENT

CVSS Vector String	CVSSv3 Score
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:N	0.0

AFFECTED SYSTEMS

Baggage Claim Symbols

DESCRIPTION

During the assessment, Finals-XX was tasked with evaluating the security of the RAKMS baggage claim system. The testing methodology encompassed capturing, decoding, analyzing, and transmitting data within these systems. Within the allocated timeframe, Finals-XX successfully captured signals from the system, but was unsuccessful in detecting or executing any illicit actions. Nonetheless, the absence of detected vulnerabilities does not equate to a guarantee of security, and potential vulnerabilities may still exist.

APPENDIX A: METHODOLOGY

Finals-XX utilizes a custom version of the Penetration Testing Execution Standard, which is both intended for business and security specialist organizations, with a standardized language and set of operations for performing evaluations.