Finals-XX

Internal Penetration Testing Report

Saturday, January 13, 2024

Table of Contents

Engagement Overview

Finals-XX was contracted by Robert A. Kalka Metropolitan Skyport to perform a reassessment of their security posture across all systems from January 12th to 13th, 2024. The purpose of this audit was to evaluate whether the company has been able to remediate the security flaws we identified in our previous engagement on November 11th, 2023 and to further test for vulnerabilities. All activities were conducted to simulate a threat actor in a targeted attempt to gain unauthorized access to assets. The goals of this assessment were:

1. Identify remediated, unremediated, and newly discovered vulnerabilities
2. Assess their risk and influence on data integrity, confidentiality, and availability
3. Assess impact on business operation and overall infrastructure
4. Outline findings and suggested remediations to secure RAKMS's network

Finals-XX identified a total of 23 findings during the engagement.

Informational	Low	Moderate	High	Critical
5	2	5	9	2

Executive Summary

Overall, the system was notably more secure than our previous engagement, and it was clear that some of our recommendations were implemented to remediate critical vulnerabilities. Despite this, there remain overt, highly impactful security flaws with the capability to significantly impact business operations. Possible impacts such as threat actor tram control, employee impersonation, and leaks of highly sensitive customer information can lead to loss of life, revenue, and reputation, requiring urgent consideration. We also noticed that RAKMS improved their alignment with the TSA regulations by making efforts to secure independent networks. In this report, we outline key remediations that would greatly secure the business environment.

Scope

Finals-XX was authorized for the following internal subnets:

Corp - 10.0.0.0/24
Guest - 10.0.200.0/24
Train - 10.0.20.0/24
User - 10.0.1.0/24
AWS Environment

The systems explicitly out of scope were VDI - 10.0.254.0/24 and VPN - 10.0.255.0/24.

Network Topology

Corporate Network 10.0.0.0/24	Tram Network 10.0.20.0/24	Guest Network 10.0.200.0/24	User Network 10.0.1.0/24
Domain Controller - 10.0.0.5 Exchange Email Server - 10.0.0.6 Baggage Checkin - 10.0.0.33 Employee DB - 10.0.0.43 MySQL Compatible Server - 10.0.0.99 Flight Dashboard - 10.0.0.100 Oracle DB - 10.0.0.101 Workstation #1 - 10.0.0.201 Workstation #2 - 10.0.0.202 Workstation #3 - 10.0.0.203	Trams Controller - 10.0.20.100 Tram #1 - 10.0.20.101 Tram #2 - 10.0.20.102 Tram #3 - 10.0.20.103	WiFi Captive Portal - 10.0.200.5 Kanicles - 10.0.200.43	(Unknown) - 10.0.1.51

METHODOLOGY

Finals-XX conducted the penetration test using the MITRE ATT&CK and Open Web Application Security Project frameworks.

The assessment commenced with a reconnaissance phase, including information gathering, scanning, and vulnerability analysis. Information procurement encompassed detailed port and service enumeration, host identification, IP address extraction, and the collation of personally identifiable and comprehensive application/service information. Techniques such as OSINT (open-source intelligence) and tools like Nmap were instrumental in this phase. The team leveraged publicly available vulnerability databases, notably ExploitDB, to identify potential exploits within the targeted services, applications, and operating systems.

Upon aggregating substantial data, the attack phase was initiated, leveraging the collected information. Potential vulnerabilities underwent scrutiny for exploitability using industry-standard tools such as Metasploit and BurpSuite. The primary aim was to secure initial system access, with a strategic pivot towards privilege escalation in cases where only user-level access was achieved. Attention was given to ensure that RAKMS complied with industry standards. Furthermore, the team re-assessed previously found vulnerabilities.

Evidence of vulnerabilities were gathered through screenshots with sensitive information appropriately redacted.

REGULATIONS AND COMPLIANCE

TSA

The United States Transportation Security Administration (TSA) issued cybersecurity requirements for airport and aircraft operators on an emergency basis. In addition to develop an approved implementation plan with measures to improve resilience against disruption and degradation, operators must proactively assess the effectiveness of these measures as follows:

1. Develop network segmentation policies to ensure that operational technology systems can continue to safely operate if an information technology system has been compromised, and vice versa.
2. Create access control measures.
3. Implement monitoring and detection policies and procedures.
4. Reduce risk of exploitation of unpatched systems by applying security patches and updating critical cyber systems.

VIOLATIONS

1. Network Segmentation	Lack of Firewall
2. Access Control	Guest account enabled without password
3. Monitoring and Detection	Lack of strict antivirus, service account login
4. Apply Security Patches and Updates	EternalBlue

METRICS

RISK SCALE

		Impact				
		Informational	Low	Moderate	High	Critical
Likelihood	Certain	Informational	Moderate	High	Critical	Critical
	Expected	Informational	Low	Moderate	High	Critical
	Common	Informational	Low	Moderate	High	High
	Rare	Informational	Low	Low	Moderate	High
	Undetermined	Informational	Informational	Informational	Informational	Informational

OVERALL RISK

Rating	Description
Critical	An immediate, easily accessible threat of compromise.
High	An immediate threat or an easily accessible threat of a large breach.
Moderate	An exploit that may be difficult to execute but may pose a large threat, or an easy compromise of a small portion.
Low	A minor threat.
Informational	No immediate threat, but provides context, suggestions for improvement, or conditions that later may lead to an exploitable finding.

Impact

The threat impact was determined with the considerations of operations, assets, individuals, organizations, and the nation in mind. It reflects the effects that an exploit may have upon the system(s) and regards damage to confidentiality, integrity, availability, and reputation.

Likelihood

The likelihood reflects the probability of a threat occurring and the chance for a threat event that occurred to trigger an adverse impact. It assesses the potential ease with which an attacker could exploit a discovery by weighing the level of access required, availability of exploitation information, and other impediments to exploitation.

Remediation Difficulty Scale

During our technical evaluation, we assessed the remediation difficulty to aid in prioritizing tasks for remediation. The difficulty indicates the amount of time it may take to resolve a vulnerability. There are three levels: low, medium, high.

Assessment Summary

Positive Measures

We noticed a few effective security measures to highlight from our evaluation:

- Remote access to the domain controller was blocked for low privilege accounts (e.g. guest)
- Strict password policy for users including password complexity and enforcing account lockout after 3 failed password attempts
- Including guest, tram, user, and corp networks are on separate subnetworks
- Multiple web applications remediated vulnerabilities highlighted in our previous engagement

Key Findings & Recommendations

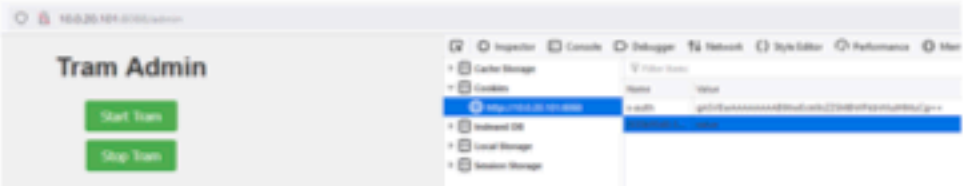
- Guest Account Enabled Without Password in Active Directory
    - We recommend disabling the guest account or only enabling it when needed by certain users
- Exposed Credential in Active Directory Description Attribute
    - Audit accounts to confirm that they do not expose confidential data and ensure that employees are trained on how to handle confidential information
- Weak Tram Control Authorization
    - Implement stricter authorization without easily modifiable cookies

VULNERABILITIES

Risk	Vulnerability	Affected Scope
Critical	Guest Account Enabled Without Password in Active Directory	10.0.20.101:80 10.0.20.102:80 10.0.20.103:80
Critical	Exposed Credential in Active Directory Description Attribute	10.0.0.6, 10.0.0.201, 10.0.0.202, 10.0.0.203
High	AWS Assumable Dev Roles	AWS IAM – S3 and SSM
High	Arbitrary Users can Register Trams on the Tram Control Site	10.0.20.100:3000
High	Hard-coded Authorization Secret in Flight Dashboard	10.0.0.100:80
High	Weak Administrator Credentials on Employee DB Portal	10.0.0.43:80
High	SQL Injection for Employee DB Portal	10.0.0.43:80
High	Insecure Direct Object Reference	https://v6yqfrnhvs4dilwcdw2jaqsevi0vghev.lambda-url.us-east-1.on.aws https://rakmsbarcode202401110348007218000000 04.s3-website-us-east-1.amazonaws.com
High	Vulnerability in AWS Boarding Pass Generator	https://v6yqfrnhvs4dilwcdw2jaqsevi0vghev.lambda-url.us-east-1.on.aws https://rakmsbarcode202401110348007218000000 04.s3-website-us-east-1.amazonaws.com
High	Interactive Logon for Service Account	10.0.0.201, 10.0.0.202, 10.0.0.203
High	Weak/Simple Password for Service Accounts	10.0.0.5
Moderate	Anonymous LDAP Bind on Corporate Domain Controller	10.0.0.5:389,636,3268,3269
Moderate	Stored XSS on Flight Dashboard	10.0.0.100:80
Moderate	Improper Network Segmentation	10.0.0.200/24, 10.0.0.20/24, 10.0.1.0/24
Moderate	Administrator Access to Corporate Workstations	10.0.0.201, 10.0.0.202, 10.0.0.203

Moderate	Stored XSS in Tram Registration IFrame	10.0.20.100:3000
Low	User Personal Information Not Requiring Privileged Access in Active Directory	10.0.0.5
Low	Reflected XSS on Employee DB Portal	10.0.0.43:80
Informational	MS17-010 (EternalBlue) on Mail Server	10.0.0.6
Informational	Verbose Error Messages on Tram Operations 404	10.0.20.100:3000
Informational	Self XSS on Tram Operations Webpage 404	10.0.20.100:3000
Informational	Social Security Number Used as ID in Boarding Pass	https://rakmsbarcode20240111034800721800000004.s3-website-us-east-1.amazonaws.com https://v6yqfrnhvs4dilwcdw2jaqsevi0vghev.lambda-url.us-east-1.on.aws
Informational	AWS CPTC 2022 Regionals Artifacts	AWS - (Various Services)

VULNERABILITY DETAILS

Critical	Weak Authentication on Tram Control Server
Risk Criteria	Likelihood: **High** Impact: **Critical**
Affected Scope	10.0.20.101:80 10.0.20.102:80 10.0.20.103:80
Description	We noticed that upon using /login.html on 10.0.20.101-103:80, if we provided an incorrect "admin code" we were assigned an authorization header. This header, when decoded from base64, revealed that we were authorized as a "guest." ┌──(root@[redacted])-[~] └─# echo "gASVEwAAAAAAAAB9lIwEcm9sZZSMBWdlZXN0lHMu" | base64 -d [redacted]role[redacted]guest[redacted]s. Upon changing "guest" to "admin" and reencoding the authorization header into base64, we were given full admin privileges over the service. This allowed us to start and stop the tram as an admin. ┌──(root@[redacted])-[~] └─# cat modified.txt [redacted]role[redacted]admin[redacted]s. ┌──(root@[redacted])-[~] └─# cat modified.txt | base64 gASVEwAAAAAAAAB9lIwEcm9sZZSMBWFkbWlulHMuCg== 
Impact	An attacker would be able to completely control all three trams, stopping or starting them at will. This could cause massive damage to trams and cause the loss of life of passengers using the trams, should the trams start and stop in an unsafe manner.
Steps to Reproduce	1. Provide an arbitrary code to 10.0.20.101-103/login.html and capture the Authorization header

	2. Decode the header from base64 3. Change "guest" to "admin" 4. Encode the new payload back into base64 5. Change the Authorization header to the new base64 payload, and visit 10.0.20.101-103/admin
Remediation	Authorization token should not be encoded with base64, as these are easily reversible and cryptographically insecure. Instead, authentication can be carried out using various types of tokens like OAuth and JSON Web Tokens. Additionally, users with elevated privileges should be stored server-side, as opposed to client-side, as attacker can control all client-side input.
References	https://frontegg.com/blog/token-based-authentication

Critical	Guest Account Enabled Without Password in Active Directory
Risk Criteria	Likelihood: **High** Impact: **Critical**
Affected Scope	10.0.0.6, 10.0.0.201, 10.0.0.202, 10.0.0.203
Description	There is an active guest account that can be used to log in to most Windows authentication applications and machines. This account does not have a password and is thus easily accessible to attackers. The Guest account is disabled by default and is not intended to be enabled without specifically restricted privileges for a limited period of time.
Impact	Attackers can use default guest credentials to access many company resources such as workstations and email services. Additionally, attackers can read active directory information that does not require privileged access.
Steps to Reproduce	Connect to Windows machines via services like RDP by providing "guest" as the username and supplying a blank password. It may be required to provide a domain within the username (e.g. "corp.kkms.local\guest").
Remediation	Difficulty: Easy Disable the guest account. If the guest account is needed, temporarily enable it with reduced permissions and immediately disable the account when finished.
References	https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/understand-default-user-accounts#guest-account

Critical	Exposed Credential in Active Directory Description Attribute
Risk Criteria	Likelihood: **Certain** Impact: **High**
Affected Scope	10.0.0.201 10.0.0.202 10.0.0.203
Description	A user's password was found in plaintext on the Active Directory description attribute associated with the user's account. | cn: Mark[redacted] | sn:[redacted] | title: Manager | description: Password: [redacted]

Impact	Since usernames can also be publicly seen by anybody in the network, this can lead to an account compromise. With this account, the following machines could be accessed through the Remote Desktop Protocol:
	• 10.0.0.201
	• 10.0.0.202
	• 10.0.0.203
	Additionally, the compromised account can also be used to acquire a service-related Kerberos ticket, which is subject to password cracking.
	`# Impacket-GetUserSPNs -request -dc-ip 10.0.0.5 corp.kkms.local/mmag[redacted] -outputfile hashes.kerberoast`
	`mpacket v0.10.0 - Copyright 2022 SecureAuth Corporation`
	`ervicePrincipalName  Name  MemberOf  PasswordLastSet  LastLogon  Deleg`
	`on`
	`---------- ------- ---------- ---------- --------- -----`
	`---`
	`TC-Sync/SkyControl01  svc_ATC  CN=all,CN=Users,DC=corp,DC=kkms,DC=local  2024-01-09 02:44:47.281858  <never>  const`
	`ned`

	
Steps to Reproduce	1. Open a command prompt 2. Run *nmap –script=safe 10.0.0.5* 3. Parse through the results. The exposed password will be present in the description attribute for the user *mmagnolia*
Remediation	Difficulty: High Review the attributes of current users to make sure no sensitive data is being exposed. Additionally, add a policy to prevent this from occurring or enforce the policy if it exists. This can be done by performing regular audits and training employees on best practices.

High	AWS Assumable Dev Roles
Risk Criteria	Likelihood: **Expected** Impact: **High**
Affected Scope	AWS IAM – S3 and SSM
Description	By assuming roles allowed us to dump 2 of the buckets: - rakmsbarcode20240111034800721800000004 - kalka-passes20240111034800610800000003 as well as to gain some plaintext AWS System Manager secrets: - aws ssm get-parameter --name /target/dev/thingy1 --with-decryption --query "Parameter.Value" - aws ssm get-parameter --name /target/dev/thingy2 --with-decryption --query "Parameter.Value" - aws ssm get-parameter --name /testdeploy/password/secrets --with-decryption --query "Parameter.Value" - aws ssm get-parameter --name /target/password/another-secret --with-decryption --query "Parameter.Value"
Impact	
Steps to Reproduce	
Remediation	These roles should not allow all users to assume into them but rather only the devs that require them. Furthermore, if the dev1 and dev2 are of specific developers, the policy can be attached directly to their users removing the need of a seperated role. This does increase the complexity of managing permissions, but this practice is already being used for the “ctf” users.

High	Arbitrary Users can Register Trams on the Tram Control Site
Risk Criteria	Likelihood: **High** Impact: **Moderate**
Affected Scope	10.0.20.100:3000
Description	Arbitrary, unauthenticated users can register trams that will appear on the trams homepage. The methods to register a tram are documented on 10.0.20.100:3000/docs.
Impact	An attacker could register multiple false trams with false schedules to cause havoc and could also use the text fields to deface the homepage.
Steps to Reproduce	1. Visit 10.0.20.100:3000/register 2. Make a POST request with the parameters contained in 10.0.20.100:3000/docs 3. Visit 10.0.20.100:3000/home and view the new tram that was registered
Remediation	Require some form of authentication to register trams. This should only be accessible to users with high privileges, so ensure that only the proper admins can register new trams.
References	N/A

High	Hard-coded Authorization Secret in Flight Dashboard
Risk Criteria	Likelihood: **High** Impact: **High**
Affected Scope	10.0.0.100:80
Description	In the HTML of http://10.0.0.100, we identified a hard-coded authorization secret that can be used by any user to view and edit the flight data present on the dashboard.
Impact	An unauthenticated attacker can use this authorization token to easily view and add to all flight data present on the dashboard. This includes creating arbitrary and incorrect flights, which could create havoc within an airport. Additionally, an attacker or web scanner could easily find this secret, as it's visible in a non-authenticated web-facing page, making it a very likely attack to occur.
Steps to Reproduce	1. View the source of the page at 10.0.0.100 2. Search for the string "Auth=" 3. View the authorization token after the string 4. If desired, use the string as a header when accessing and adding to /Flight
Remediation	Difficulty: Medium Each user with permission to edit the dashboard's flight data should be assigned their own unique authorization token. All secrets should never be stored in plaintext in internet-facing applications, as they are very easy for unauthenticated users to find.
References	N/A

High	Weak Administrator Credentials on Employee DB Portal
Risk Criteria	Likelihood: **High** Impact: **High**
Affected Scope	10.0.0.43:80
Description	We were able to easily brute force the Administrator's username and password for the Employee DB portal webpage. While we won't disclose the username and password on this document, please ensure all credentials for users with elevated privileges follow proper password hygiene. This vulnerability was present in our previous engagement with RAKMS, and the password was not rotated/improved.
Impact	With these credentials, an attacker was able to create, view, and modify time sheets for themselves and users. Additionally, they were able to create additional admin users, enabling persistence.
Steps to Reproduce	1. Visit 10.0.0.43:80/login 2. Login with the weak Administrator credentials 3. Observe the elevated privileges of an admin, including access to the /admin page
Remediation	Difficult: Low Passwords (especially of elevated individuals) should conform to a strong, rotating password policy. Ensure that there is a high minimum password length, and that the password is not a common password.
References	N/A

High	SQL Injection for Employee DB Portal
Risk Criteria	Likelihood: **Medium** Impact: **Critical**
Affected Scope	10.0.0.43:80
Description	There is a SQL injection vulnerability in all four parameters when making a POST request to http://10.0.0.43/index.php?employee=admin&page=admin. Injecting an apostrophe results in a verbose MariaDB SQL error for displaying injection output, allowing attackers to directly interface with the SQL database for this server. In this image, you can see that we added an apostrophe to the "employee" category, resulting in the error in the 2nd image. POST /index.php?employee=admin&page=admin HTTP/1.1 Host: 10.0.0.43 Cookie: PHPSESSID=7a5o689ufit4pitke2gb9cnn4f Content-Length: 74 Cache-Control: max-age=0 Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120" Sec-Ch-Ua-Mobile: ?0 Sec-Ch-Ua-Platform: "Windows" Upgrade-Insecure-Requests: 1 Origin: https://10.0.0.43 Content-Type: application/x-www-form-urlencoded User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.199 Safari/537.36 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7 Sec-Fetch-Site: same-origin Sec-Fetch-Mode: navigate Sec-Fetch-User: ?1 Sec-Fetch-Dest: document Referer: https://10.0.0.43/index.php?employee=admin&page=admin Accept-Encoding: gzip, deflate, br Accept-Language: en-US,en;q=0.9 Priority: u=0, i Connection: close clockIn=00%3A00%3A00&clockOut=00%3A00%3A00&date=2024-01-01&employee=a'dmin

	 This vulnerability was present in our previous engagement and was not addressed.
Impact	SQL injection attacks allow attackers to cause repudiation issues (i.e., adding/modifying/deleting invalid data), to exfiltrate and disclose of all data (including permissions) on the database, or to make it otherwise unavailable for proper use. We were able to access
Steps to Reproduce	1. Visit 10.0.0.43:80/index.php?employee=admin&page=admin. 2. Make an edit to one of the time sheets, capturing the POST request in some kind of packet editor. Edit any field in the POST parameters (not the URL) by adding an apostrophe. 3. Observe the SQL error on the resultant page, indicative of SQL injection.
Remediation	Difficulty: Low SQL injection can be avoided through the use of parametrized queries, as opposed to string concatenation. Refer to the second reference link for more details.
References	https://owasp.org/www-community/attacks/SQL_Injection https://cheatsheetseries.owasp.org/cheatsheets/Query_Parameterization_Cheat_Sheet.html

High	Insecure Direct Object Reference Vulnerability in AWS Boarding Pass Generator
Risk Criteria	Likelihood: **High** Impact: **High**
Affected Scope	https://v6yqfrnhvs4dilwcdw2jaqsevi0vghev.lambda-url.us-east-1.on.aws https://rakmsbarcode20240111034800721800000004.s3-website-us-east-1.amazonaws.com
Description	We were able to identify an insecure direct object reference (IDOR) vulnerability with the AWS boarding pass generator. An IDOR exists when a user can access objects that are similar to theirs that they shouldn't be able to access. When a boarding pass is generated, the SVG image of the boarding pass is assigned an identifier consisting of the time it was created (Month, Day, Hour, Minute, Second). There is no authentication present when accessing the boarding pass SVGs, and thus any user can access any other user's boarding pass
Impact	A boarding pass is not only a ticket onto a plane, but it reveals a lot of personally identifiable information about a passenger. For example, criminals will often monitor when passengers go on vacation in order to stage robberies at the optimal times, and these boarding passes would provide those exact times. Any threat actor would be able to easily view any boarding pass for any passenger who has used the application before them.
Steps to Reproduce	1. Register a boarding pass at https://rakmsbarcode20240111034800721800000004.s3-website-us-east-1.amazonaws.com, ignoring any errors that pop up 2. Upon receiving the boarding pass SVG location from https://v6yqfrnhvs4dilwcdw2jaqsevi0vghev.lambda-url.us-east-1.on.aws, note that the SVG "path" returned is represented by the time it was registered. 3. Access the SVG registered by appending the path onto https://rakmsbarcode20240111034800721800000004.s3-website-us-east-1.amazonaws.com after a "/".

	4. Access any previously registered boarding pass SVG without the need for authentication by simply appending the time and ".svg" onto the same url.
Remediation	Ensure that, after a user has generated a boarding pass, they are given some form of authentication to use to access the boarding pass. While the generation of sequential identifiers isn't a strong vulnerability on its own, the access of the boarding passes should only be allowed to the user that generates it, no matter the name.
References	N/A

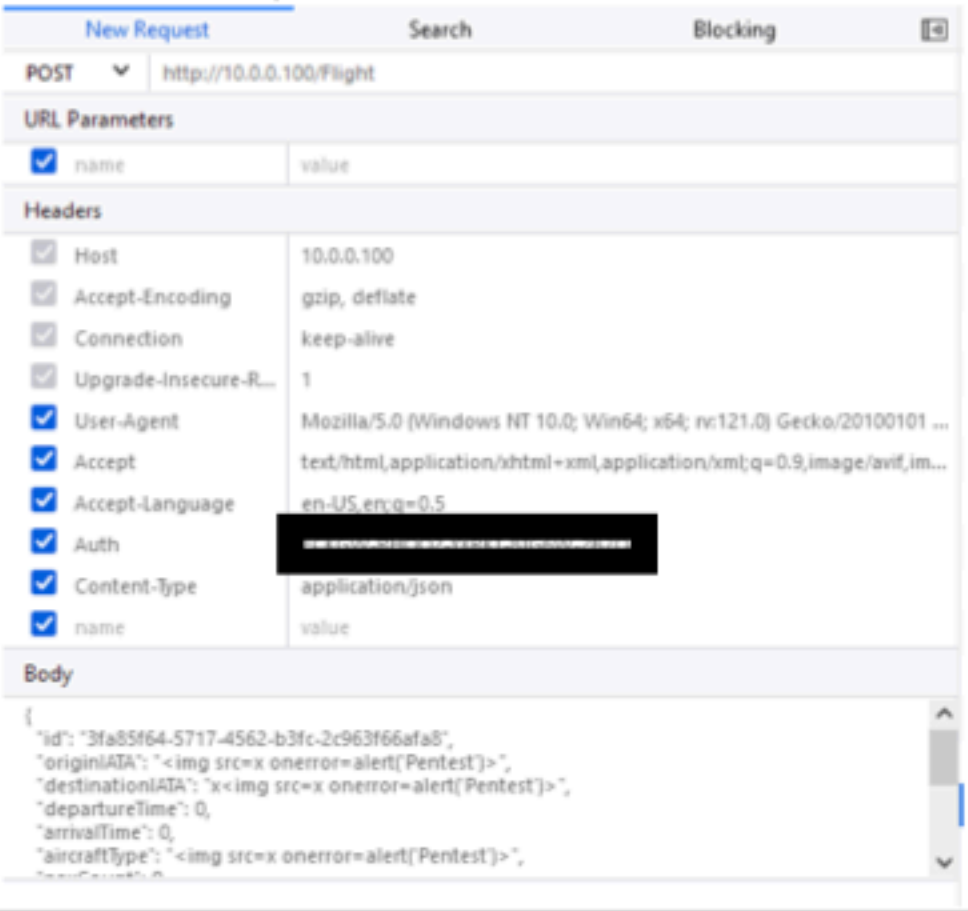
High	Interactive Logon for Service Account
Risk Criteria	Likelihood: **Common** Impact: **High**
Affected Scope	10.0.0.201, 10.0.0.202, 10.0.0.203
Description	We were able to interactively log on to a service account, svc_ATC. Service accounts are designed for services or applications to login to interact with the operating system. Due to their nature, no human should be permitted to log on. This requires credentials to the service account.
Impact	Service accounts have higher privileges due to their nature. Interactive logins present a way for a person to exploit privileges not granted to the user. Furthermore, any logon to a service account is not directly tied to an end-user account therefore bypassing logging mechanisms. This can lead to insider threat.
Steps to Reproduce	Login to svc_ATC.
Remediation	Configure service accounts to deny interactive logons, practice the principle of least privilege, and implement a change management process.
References	https://serverfault.com/q/771820

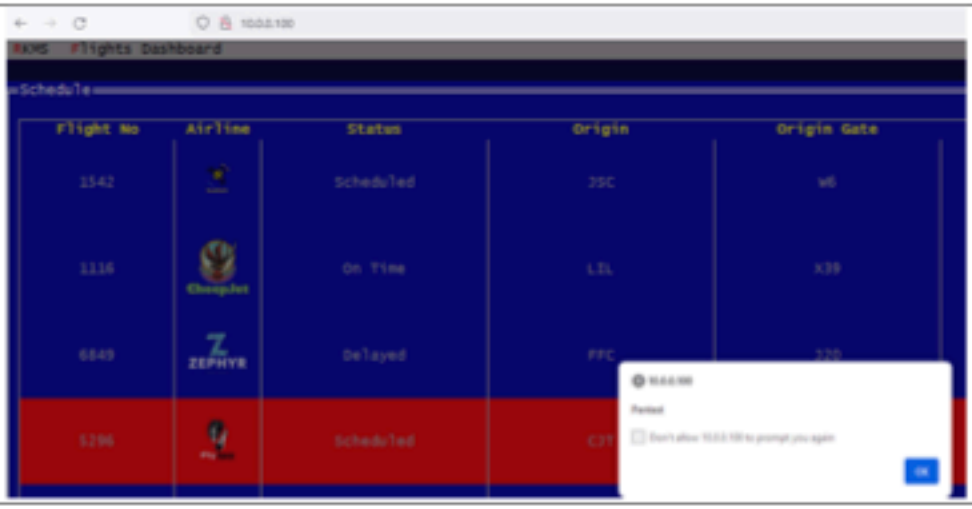
High	Weak/Simple Password for Service Accounts
Risk Criteria	Likelihood: **Common** Impact: **High**
Affected Scope	10.0.0.5
Description	We were able to crack the hash for the svc_ATC service using the associated Ticket Granting Service (TGS) ticket. This was due to the service having a weak/simple password.
Impact	Access to credentials can lead to impersonation and incorrect authorization. Service accounts typically have higher privileges due to the nature of their task, which can lead to a higher chance of privilege escalation.
Steps to Reproduce	1. Open a command line and run the following command: *hashcat -m 13100 --force -a 0 <hash file> /usr/share/wordlists/rockyou.txt --show* 
Remediation	Difficulty: Medium Services accounts are non-human privileged accounts that are used to execute applications and run automated services and other processes. As they are used by applications, not people, they are not constrained to human tendencies. Good password hygiene includes using long passwords (at least twenty five characters) and regularly rotating passwords every 30 days.
References	https://www.beyondtrust.com/blog/entry/how-to-manage-and-secure-service-accounts-best-practices

Moderate	Anonymous LDAP Bind on Corporate Domain Controller
Risk Criteria	Likelihood: **High** Impact: **Moderate**
Affected Scope	10.0.0.5:389,636,3268,3269
Description	This is an un-remediated vulnerability from our previous assessment. To summarize: the active directory server supports anonymous binding, allowing attackers to enumerate and display most LDAP information of the employees.
Impact	The exposed information from the corporate LDAP server includes an employee's name, email, street address, and title amongst other metadata. This exposes personal information and provides data that is valuable for potential phishing opportunities.

Steps to Reproduce	1. Run the nmap script for LDAP search to obtain information from anonymous LDAP bind if enabled.
	[redacted]
	``root@[redacted]:~/sqlmap
	└─# nmap -p389 --script ldap-search 10.0.0.5
	Starting Nmap 7.93 (https://nmap.org) at 2024-01-13 16:56 EST
	Nmap scan report for 10.0.0.5
	Host is up (0.0038s latency).
	
	PORT STATE SERVICE
	389/tcp open ldap
	| ldap-search:
	| Context: DC=corp,DC=kkms,DC=local
	| dn: DC=corp,DC=kkms,DC=local
	| dn: OU=HR,OU=Departments,DC=corp,DC=kkms,DC=local
	| objectClass: top
	| objectClass: organizationalUnit
	| ou: HR
	| distinguishedName: OU=HR,OU=Departments,DC=corp,DC=kkms,DC=local
	| instanceType: 4
	| whenCreated: 2024/01/09 07:39:28 UTC
	| whenChanged: 2024/01/09 07:42:43 UTC
	| uSNCreated: 16458
	| uSNChanged: 18628
	| name: HR
	| objectGUID: 84189ef5-a98-4b4a-81c3-cec37852708f8
	| objectCategory: CN=Organizational-Unit,CN=Schema,CN=Configuration,DC=corp,DC=kkms,DC=local
	| dSCorePropagationData: 2024/01/09 15:20:15 UTC
	| dSCorePropagationData: 2024/01/09 15:20:15 UTC
	| dSCorePropagationData: 2024/01/09 15:20:15 UTC
	| dSCorePropagationData: 2024/01/09 15:20:15 UTC
	| dSCorePropagationData: 1601/07/14 22:36:49 UTC
	| dn: CN=Scott Smith,OU=HR,OU=Departments,DC=corp,DC=kkms,DC=local
	| objectClass: top
	| objectClass: person
	| objectClass: organizationalPerson
	``
	[redacted]

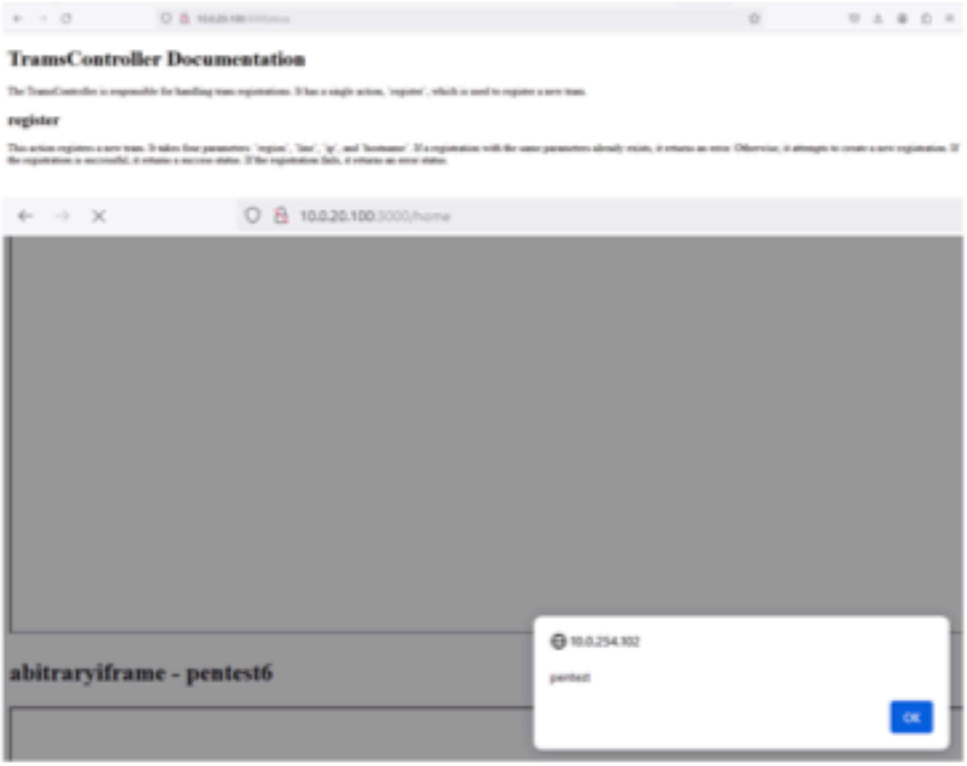
Remediation	Difficulty: Easy We recommend disabling anonymous bind, as described in the second reference link, or if it is necessary for some applications, utilizing a bind account.
References	https://nmap.org/nsedoc/scripts/ldap-search.html https://blog.lithnet.io/2018/12/disabling-unauthenticated-binds-in.html

Moderate	Stored XSS on Flight Dashboard
Risk Criteria	Likelihood: **Low** Impact: **Moderate**
Affected Scope	10.0.0.100:80
Description	We were able to identify a stored XSS vulnerability in the flight dashboard, which provides information about the flights at the time. The vulnerability is present in the "flightNumber" parameter of the flight, allowing attackers with the ability to create flight data to put malicious JavaScript in the flight number of a newly-created flight. In order to attain the proper authorization to register flight data, we used the hard-coded authorization present in the HTML of 10.0.0.100:80 titled "Auth=". 

	
Impact	With stored XSS, an attacker could inject malicious JavaScript on any user that visits the flight dashboard. Through this JavaScript, the attacker could perform any authenticated action as the victim whose browser loads the site. That said, since new flight data can only be registered by authenticated users, there will be very few users who should be able to execute this attack, resulting in a low likelihood of the attack happening.
Steps to Reproduce	1. Navigate to http://10.0.0.100/swagger/index.html 2. Observe the JSON data required to make a POST request to /Flight 3. Make a GET request to /Flight and capture the request 4. Change the GET method to POST 5. Add the JSON data from Swagger to the body of the request 6. Add an XSS payload in the "flightNumber" parameter, such as <img src=x onerror=alert('Pentest')> 7. Add the header: "Content-Type": "application/json" 8. Add the header: "Auth": "{AUTH SECRET}" 9. Send the request and navigate back to <http://10.0.0.100> 10. Observe the XSS payload
Remediation	Difficulty: Low To remediate this vulnerability, ensure that HTML special characters are either filtered or escaped, using a library. For example, in PHP, this function is called htmlspecialchars(), and will prevent HTML and JavaScript injections in the text.
References	https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html

Moderate	Improper Network Segmentation
Risk Criteria	Likelihood: **Certain** Impact: **Moderate**
Affected Scope	10.0.0.200/24, 10.0.0.20/24, 10.0.1.0/24
Description	Machines in the guest, tram, and user network are visible from the corporate network during network scans.
Impact	Due to the lack of network segmentation, any employees or consultants given access to the corporate network could discover infrastructure and probe operations in all other networks. This leaves open the opportunity for internal malicious actors to exceed their intended access and exploit found vulnerabilities.
Steps to Reproduce	1. Run nmap or similar network scans from the corporate network on the other networks (guest, tram, or user) 2. Observe the discovered machines/infrastructure from other networks
Remediation	We recommend moving all infrastructure to a separate network that can only be accessed by authorized users
References	N/A

Moderate	Administrator Access to Corporate Workstations
Risk Criteria	Likelihood: **Moderate** Impact: **Moderate**
Affected Scope	10.0.0.201, 10.0.0.202, 10.0.0.203
Description	Non privileged user accounts on the domain have administrator access to corporate workstations which should be reserved for domain admins.
Impact	Users can read/write configuration settings, other users' files, and environment variables. An attacker can leverage local admin access to infect a machine and further attack future users.
Steps to Reproduce	1. Use remote desktop protocol to access workstation machines via non privileged account 2. Observe that account has admin access
Remediation	Only domain administrators should have local administrator access to workstations. If users need to have admin access to configure certain applications, they should make a request to domain admins.
References	N/A

Moderate	Stored XSS in Tram Registration IFrame
Risk Criteria	Likelihood: **Moderate** Impact: **Moderate**
Affected Scope	10.0.20.100:3000
Description	Upon registering a new tram, a user can specify the IP of the tram. The service will then reach out to the IP and Iframe the content of port 80 on the IP. If the resultant webserver on the IP contains malicious JavaScript, the JavaScript will execute on the tram homepage, resulting in XSS. 
Impact	Any user who visits the trams homepage will execute the malicious JavaScript included in the Iframe of the page. The attacker who wrote the JavaScript could execute the malicious JavaScript on any user who views the home page viewing the tram listing
Steps to Reproduce	1. Register a new tram at 10.0.20.100:3000/register 2. Specify the IP as a controlled webserver hosting a JavaScript alert

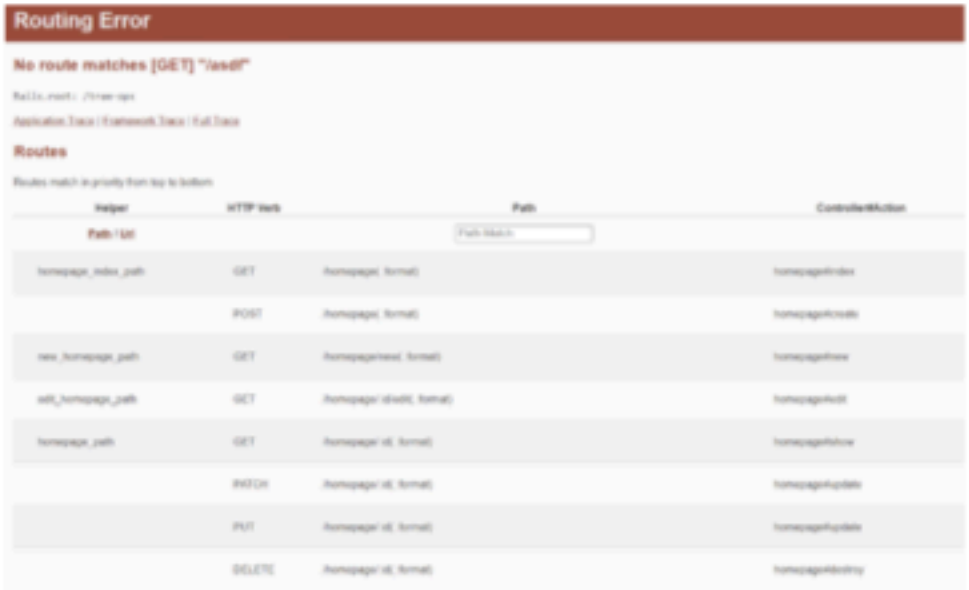
	3. Observe the XSS payload upon visiting 10.0.20.100:3000/home
Remediation	Allowing users to control the Iframe of the site can be made a lot safer if "allow-same-origin" is turned on for Iframes. This will prevent the site from Iframe-ing sites that aren't of the same origin, preventing XSS coming from other sites.
References	https://developer.mozilla.org/en-US/docs/Web/Security/Same-origin_policy

Low	User Personal Information Not Requiring Privileged Access in Active Directory
Risk Criteria	Likelihood: **Medium** Impact: **Low**
Affected Scope	10.0.0.5
Description	Users on active directory can read personal information of other users such as email, full name, and address. There is no requirement of privileged access to view individual user information; any user authorized on Active Directory can read it.
Impact	Employees lack privacy of personal information within the organization. Any fellow employees have access to all stored information and any attacker that infiltrates a single account would be able to read large amounts of PII from all company employees.
Steps to Reproduce	1. Display all available ldap data using a command like "ldapdomaindump -u <USERNAME> -p <PASSWORD> <IP>"
Remediation	Difficulty: Medium Enforce access rules that require privileged access to view other users' personal information. For users that require this access, assign them a specific role, group, or permission.
References	https://github.com/dirkjanm/ldapdomaindump

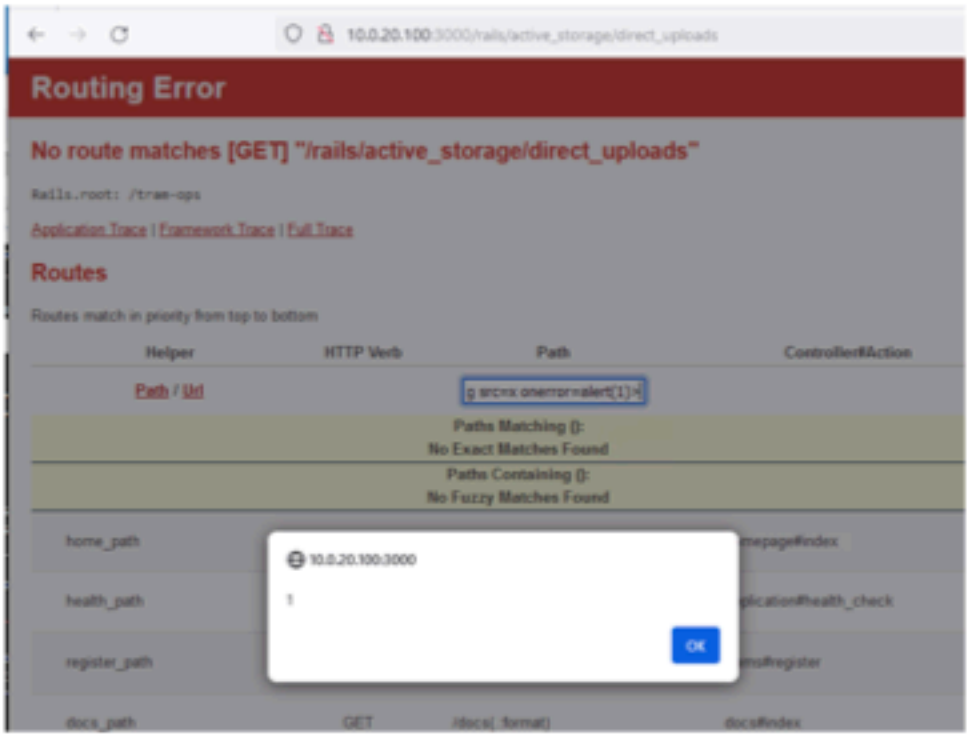
Low	Reflected XSS on Employee DB Portal
Risk Criteria	Likelihood: **Low** Impact: **Moderate**
Affected Scope	10.0.0.43:80
Description	A user is able to inject JavaScript into the "employee" parameter of the /index.php URL on the Employee DB portal when viewing as an admin user. This means that, if an attacker was able to figure out the existence of this parameter in this URL, they would be able to inject JavaScript onto any admin who clicks the page.  This vulnerability was present in our previous engagement with RAKMS, and no security features have been added.
Impact	An attacker who can inject malicious JavaScript onto the page of an admin can steal the admin's credentials, steal plaintext data on the page, and perform authenticated actions as that admin on the server. This, however, is quite unlikely, as it would be difficult for an employee with regular permissions to discover this injection.
Steps to Reproduce	1. http://10.0.0.43/index.php?employee=%3Cscript%3Ealert('Pentest')%3C/script%3E&page=admin as an administrator.

	2. Observe the JavaScript alert of "Pentest".
Remediation	Difficulty: Low All user input that is reflected onto the page must be sanitized properly, especially HTML special characters. This can be done by escaping or sanitizing user input with libraries like htmlspecialchars() in PHP.
References	https://owasp.org/www-community/attacks/xss/

Informational	MS17-010 (EternalBlue) on Mail Server
Risk Criteria	Likelihood: **Expected** Impact: **Informational**
Affected Scope	10.0.0.6
Description	EternalBlue is an exploit developed by the NSA and leaked via ShadowBrokers in 2017. Recent similar "Eternal" exploits have been developed to attack systems from Windows Server 2000 up to certain versions of Windows 10. EternalBlue gives the attacker complete root access to the target system via remote code execution through specially crafted SMB packets sent to the target.
Impact	Once a remote shell is opened using EternalBlue, the attacker has control of the system, allowing a complete system takeover. The SMBv1 vulnerability opens the system up to the possibility of Ransomware attacks such as WannaCry, which are delivered as payloads via EternalBlue-type attacks.
Steps to Reproduce	1. Run an nmap or Metasploit command to scan for ms17-010 (nmap -p445 --script smb vuln-ms17-010 <IP>) 2. Observe result of script to determine if machine is vulnerable
Remediation	Difficulty: Low Apply Microsoft Updates: Patch devices with Microsoft Windows OS with the security update for Microsoft Windows SMBv1. The Microsoft Security Bulletin, MS17-010, includes the list of affected Windows OS. Disable SMBv1: Where appropriate and after thorough testing, utilize SMBv2 or SMBv3 instead of SMBv1.
References	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143 https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/ https://technet.microsoft.com/en-us/library/security/ms17-010.aspx

Informational	Verbose Error Messages on Tram Operations 404
Risk Criteria	Likelihood: **Low** Impact: **Medium**
Affected Scope	10.0.20.100:3000
Description	Getting a 404 Not Found error on the Tram Operations page resulted in a sitemap of the entire site being revealed. Many functions revealed appeared to have dangerous functionality, such as uploading dangerous files or adding new homepages for the server.  These verbose error messages existed in our last engagement, and in this engagement enabled us to find a further vulnerability.
Impact	An attacker who could view the entire sitemap may be able to abuse functionality given to unauthenticated users. We were unable to fully test the functionality due to time constraints, but should they exploit functionality such as the file upload, they may have been able to upload ransomware, cryptocurrency miners, or other dangerous files to the server.

Steps to Reproduce	1. Visit 10.0.20.100:3000/asdf (or any other nonexistent page) and observe the sitemap and error message present.
Remediation	We recommend setting a custom 404 error that reveals nothing about the back-end. This error should only say something along the lines of "Error 404: Page not found", which wouldn't reveal any hidden functionality.

Informational	Self XSS on Tram Operations Webpage 404
Risk Criteria	Likelihood: **Low** Impact: **Informational**
Affected Scope	10.0.200.100:3000
Description	We were able to achieve self-XSS on the Tram Operations page through the path search in a 404 page.  This self-XSS existed during the previous engagement.
Impact	While a self-XSS is not immediately exploitable, if an attacker were able to somehow include the XSS payload in a URL or permanent page, they would be able to execute a full stored or reflected XSS attack. With this, they could perform any authenticated action as the victim and steal credentials/plaintext data from the victim.
Steps to Reproduce	1. Visit 10.0.20.100:3000/asdf (or any other 404 page on the server)

	2. Inject <img src=x onerror=alert("Pentest")> into the "Path search" functionality 3. Search for the path 4. Receive the JavaScript alert of "Pentest", indicating XSS
Remediation	All user input that is reflected onto the page must be sanitized properly. This can be done by escaping, encoding, or sanitizing the input reflected onto the page. In Ruby, HTML characters can be encoded with CGI.escapeHTML().
References	https://stackoverflow.com/questions/1600526/how-do-i-encode-decode-html-entities-in-ruby

Informational	Social Security Number Used as ID in Boarding Pass
Risk Criteria	Likelihood: **Undetermined** Impact: **Moderate**
Affected Scope	https://rakmsbarcode20240111034800721800000004.s3-website-us-east-1.amazonaws.com https://v6yqfrnhvs4dilwcdw2jaqsevi0vghev.lambda-url.us-east-1.on.aws
Description	We noticed that, in order to properly register a boarding pass, a passenger had to provide their SSN as their ID for their boarding pass. We additionally noticed that this ID was included in a URL passed to https://v6yqfrnhvs4dilwcdw2jaqsevi0vghev.lambda-url.us-east-1.on.aws, which could be easily leaked as including sensitive data in URL parameters is dangerous (as previously discussed).
Impact	Boarding pass barcodes can be reversed and scanned, which we tested on some of our own generated barcodes. Thus, anyone with access to someone else's boarding pass would be able to immediately know their SSN, which is a big PII leak.
Steps to Reproduce	1. Generate a boarding pass at https://v6yqfrnhvs4dilwcdw2jaqsevi0vghev.lambda-url.us-east-1.on.aws 2. Upon receiving the boarding pass SVG, use any kind of boarding pass bar code scanner to scan the boarding pass 3. Observe that the SSN entered as the ID can be extracted from the boarding pass barcode.
Remediation	We recommend using a different form of identification as the ID on a boarding pass. This could be some kind of unique identifier assigned to each passenger upon purchasing a flight, but using Social Security Numbers is dangerous given how tied they are to a person's identity.

Informational	AWS CPTC 2022 Regionals Artifacts
Risk Criteria	Likelihood: **Certain** Impact: **Informational**
Affected Scope	AWS - (Various Services)
Description	Artifacts from previous work remains on the AWS infrastrucutre. References to a "CPTC 2022 Regionals" exists as well as empty DNS entries for a "luckycrossaint". Furthermore, old EC2 disks are present which may be used to extract valuable information.
Impact	Artifacts can be exploited by users who find themselves in the environment and since they are typically forgotten, they may come out of compliancy with the business security model.

Steps to Reproduce	`PS C:\Users\Administrator\Downloads\lll> aws cloudformation list-stacks --query "StackSummaries[*].[StackName]"`
	`[`
	`  [`
	`  "CPTC-Regional-VPC-Flow-Testing"`
	`  ],`
	`  [`
	`  "Splunk-Logging-Test"`
	`  ]`
	`]`
	`PS C:\Users\Administrator\Downloads\lll> aws route53 list-hosted-zones --query "HostedZones[*].Name"`
	`[`
	`  "cptc.link.",`
	`  "luckycroissant.net.",`
	`  "luckycroissant.com.",`
	`  "luckycroissant.org."`
	`]`

```
PS C:\Users\Administrator\Downloads\lll> aws ec2 describe-tags --query "Tags[*].[ResourceId, ResourceType, Value]"
[
    [
        "nat-002596f3e61944828",
        "natgateway",
        "regionals-2022-aws-team-50-public_subnet-50ad878e"
    ],
    [
        "nat-00ea6d84a51251b2e",
        "natgateway",
        "regionals-2022-aws-team-20-public_subnet-c9efbef8"
    ],
    [
        "nat-00f9941abe7e9e6cf",
        "natgateway",
        "regionals-2022-aws-team-18-public_subnet-50ad878e"
    ],
    [
        "nat-036e0db56a433fc31",
        "natgateway",
        "regionals-2022-aws-team-12-public_subnet-ff7ffdf3"
    ],
    [
        "nat-0473577e133696986",
        "natgateway",
        "regionals-2022-aws-team-01-public_subnet-579b0e41"
    ],
    [
        "nat-05deb9db1265314fc",
        "natgateway",
        "regionals-2022-aws-team-16-public_subnet-50ad878e"
    ],
    [
        "nat-06b9a7b6c670b9ff3",
        "natgateway",
        "regionals-2022-aws-team-15-public_subnet-50ad878e"
    ],
    [
        "nat-0abf336149ec5831d",
        "natgateway",
        "regionals-2022-aws-team-31-public_subnet-50ad878e"
    ],
    [
        "nat-0b9c6abfe00d00cdc",
        "natgateway",
        "regionals-2022-aws-team-26-public_subnet-50ad878e"
    ],
    [
        "nat-0bd89cc53b729f62a",
        "natgateway",
        "regionals-2022-aws-team-24-public_subnet-50ad878e"
    ],
    [
        "nat-0e97f92480d42f353",
        "natgateway",
        "regionals-2022-aws-team-40-public_subnet-50ad878e"
    ],
    [
        "nat-0f9a3ff172a9e8ff1",
        "natgateway",
        "cptc2022-Team-01-Public_Subnet-248f1a2c-d1d8-4c88-9203-83d2432902fb"
    ],

PS C:\Users\Administrator\Downloads\lll> aws ec2 describe-volumes --query "Volumes[*].[Attachments, VolumeId]"
[
    [
        [],
        "vol-0cdf0c903ca1c15aa"
    ],
    [
        [],
        "vol-0ea9809833dd8c43a"
    ],
    [
        [],
        "vol-0997e2787dfe697a3"
    ],
    [
        [],
        "vol-04b03c3b69792c4e1"
    ]
]

PS C:\Users\Administrator\Downloads\lll> aws firehose list-delivery-streams --query "DeliveryStreamNames[*]"
[
    "CPTC-Regional-VPC-Flow-Testing-VPCFirehoseDeliveryS-GZNEhYx9UO2f"
]
```

```
PS C:\Users\Administrator\Downloads\III> aws ec2 describe-internet-gateways --query "InternetGateways[*].[InternetGatewayId, Tags[*].Value]"
[
    [
        "igw-02dce2d471f16596a",
        [
            "cptc2022-Team-01-abccb68b-d860-4d0e-b9d7-b6b5a366e343"
        ]
    ],
    [
        "igw-0655d5dc8b8b3e06e",
        []
    ],
    [
        "igw-eb260590",
        []
    ]
]
```

```
PS C:\Users\Administrator\Downloads\III> aws resourcegroupstaggingapi get-resources --query "ResourceTagMappingList[*].[ResourceARN, Tags[*].Value]"
[
    [
        "arn:aws:ec2:us-east-1:677302527522:natgateway/nat-0b06a766c679b69ff3",
        [
            "regionals-2022-aws-team-15-public_subnet-50ad878e"
        ]
    ],
    [
        "arn:aws:ec2:us-east-1:677302527522:natgateway/nat-0fa0b4d8c8efdcee5",
        [
            "cptc2022-Team-06-Public_Subnet-248f1a2c-d1d8-4c88-9283-83d2432982fb"
        ]
    ],
    [
        "arn:aws:ec2:us-east-1:677302527522:natgateway/nat-05de0b0b1265314fc",
        [
            "regionals-2022-aws-team-16-public_subnet-50ad878e"
        ]
    ],
    [
        "arn:aws:ec2:us-east-1:677302527522:natgateway/nat-0f9a3ff172a9e8ff1",
        [
            "cptc2022-Team-01-Public_Subnet-248f1a2c-d1d8-4c88-9283-83d2432982fb"
        ]
    ],
    [
        "arn:aws:ec2:us-east-1:677302527522:vpc/vpc-06e915eb",
        [
            "MANAGEMENT"
        ]
    ],
    [
        "arn:aws:ec2:us-east-1:677302527522:natgateway/nat-0473577e1336969b6",
        [
            "regionals-2022-aws-team-01-public_subnet-579b0e41"
        ]
    ],
    [
        "arn:aws:ec2:us-east-1:677302527522:natgateway/nat-0025960f3e61944828",
        [
            "regionals-2022-aws-team-50-public_subnet-50ad878e"
        ]
    ],
    [
        "arn:aws:ec2:us-east-1:677302527522:natgateway/nat-036e0db56a433fc31",
        [
            "regionals-2022-aws-team-12-public_subnet-ff7ffdf3"
        ]
    ],
    [
        "arn:aws:ec2:us-east-1:677302527522:network-insights-path/nip-0f4c1cf8d5a72121b",
        [
            "t8-vdi-win02"
        ]
    ],
    [
        "arn:aws:acm:us-east-1:677302527522:certificate/3c979ef1-6285-477b-83f1-f51e81fc1c4d",
        [
            "aws-vmware-vpn-cert"
        ]
    ],
```

Remediation	Delete old artifacts and use different AWS Accounts for different purposes to allow for better resource management.

APPENDIX

Social Engineering

As requested by the client, we were tasked to perform social engineering attacks against the employees in two controlled attempts. The first was performed as a vishing attack (phishing via phone) against helpdesk, while the second took place as an email.
For the vishing attack, our objective was to get information on a user to use later in an email. Adopting the premise that we were a part of HR having payroll issues, we began a casual conversation in which we were successful in retrieving several personal pieces of information about an employee: their first name, the department that they worked in, the hours they were regularly in the office, etc. Additionally, helpdesk disclosed vital information by sharing that they frequented the airport swag website. This information was used to help frame the phishing email that we sent later in the engagement.
During the phishing portion, we emailed the user a malicious executable file and told them it contained a new merchandising application. This was in line was the information that the target frequented swag websites. In the email, we specified that they had to change immediately as it was better supported. Furthermore, they were also told to ignore all notifications and virus warnings since the application was still in development.
After conducting these social attacks, we recommend that RAKMS becomes more vigilant about recognizing vishing attacks. Despite taking a significant amount of time to answer basic questions during the vishing portion as well as including an executable attachment in an email instead of sending a link, two red flags, we were given ample of information including that unrequested. We recommend regular security training so that employees become more familiar with social engineering attacks and are careful about the information they give out in the future.

AWS Methodology

As a part of the initial scope given during this engagement, we were also granted access to an additional AWS environment. Initial entry was granted by RAKMS as an AWS CLI access key id and secret pair.

```
PS C:\> aws sts get-caller-identity
{
    "UserId": "AIDAZ3MTAMYRICUZIRDMB",
    "Account": "677302527522",
    "Arn": "arn:aws:iam::677302527522:user/ctf-starting-user-6"
}
```

Once we gained access, we began enumerating the various AWS services and concluded that the AWS solution consisted of a combination of Lambda, S3, and DynamoDB.

```
PS C:\Users\Administrator\Downloads\lll> aws dynamodb list-tables --query "TableNames[*]"
[
    "requisitions",
    "toolinfo"
]
```

```
PS C:\Users\Administrator\Downloads\lll> aws lambda list-functions --query "Functions[*].[FunctionName, FunctionArn, Role, Environment]"
[
    [
        "lambda-barcode-function",
        "arn:aws:lambda:us-east-1:677302527522:function:lambda-barcode-function",
        "arn:aws:iam::677302527522:role/lambda-barcode-role",
        {
            "Variables": {
                "rakms_barcode_endpoint": "https://rakmsbarcode202401110348007218000000004.s3-website-us-east-1.amazonaws.com/",
                "rakms_barcode_bucket": "rakmsbarcode202401110348007218000000004"
            }
        }
    ],
    [
        "lambda-map-function",
        "arn:aws:lambda:us-east-1:677302527522:function:lambda-map-function",
        "arn:aws:iam::677302527522:role/lambda-map-role",
        {
            "Variables": {
                "rakms_endpoint": "http://rakmslocationservice202401110348010597000000006.s3-website-us-east-1.amazonaws.com/",
                "logging_bucket": "rakmslocationservice-logging202401110348003406000000001"
            }
        }
    ],
    [
        "tool-requisition-function",
        "arn:aws:lambda:us-east-1:677302527522:function:tool-requisition-function",
        "arn:aws:iam::677302527522:role/tool-requisition-role",
        {
            "Variables": {
                "rakms_endpoint": "http://rakmstoolrequisition202401110348011242000000007.s3-website-us-east-1.amazonaws.com/",
                "logging_bucket": "rakmstoolrequisition-logging202401110348009749000000005",
                "rakms_bucket": "rakmstoolrequisition202401110348011242000000007"
            }
        }
    ]
]
```

```
PS C:\> aws s3 ls
2024-01-10 22:48:02 devlog202401110348003539000000002
2024-01-10 22:48:02 kalka-passes202401110348006108000000003
2024-01-10 22:48:02 rakmsbarcode202401110348007218000000004
2024-01-10 22:48:02 rakmslocationservice-logging202401110348003406000000001
2024-01-10 22:48:03 rakmslocationservice202401110348010597000000006
2024-01-10 22:48:03 rakmstoolrequisition-logging202401110348009749000000005
2024-01-10 22:48:03 rakmstoolrequisition202401110348011242000000007
```

```
PS C:\> aws ssm describe-parameters --query "Parameters[*].[Name, Type]"
[
    [
        "/production/database/password",
        "SecureString"
    ],
    [
        "/production/database/username",
        "String"
    ],
    [
        "/staging/database/password",
        "String"
    ],
    [
        "/staging/database/user",
        "String"
    ],
    [
        "/target/dev/thingy1",
        "SecureString"
    ],
    [
        "/target/dev/thingy2",
        "SecureString"
    ],
    [
        "/target/password/another-secret",
        "SecureString"
    ],
    [
        "/testdeploy/password/secrets",
        "SecureString"
    ]
]
```

After we enumerated the services, we began to look through the IAM roles in order to check for any ways to gain better privileges in the environment. Our current user's privileges were mainly limited to only list operations, so we wanted to get specific "get" rights to some of the services.

```
PS C:\> aws iam list-attached-role-policies --role-name dev-barcode-role
{
    "AttachedPolicies": [
        {
            "PolicyName": "dev-barcode-policy",
            "PolicyArn": "arn:aws:iam::677302527522:policy/dev-barcode-policy"
        }
    ]
}
```

After conducting enumeration with the current role, we moved on to assuming other roles we have access to as mentioned in the "".

Boarding Passes Leak

During our engagement, we were notified of a threat actor who had acquired access to passenger boarding passes. More specifically, we were tasked with finding out how they accomplished such a task and whether we would need to pay to reclaim control. Thankfully, we have figured out how the threat actor may have gotten access, and more details are in the "AWS Assumable Dev Roles".

Radio Behavior

During this second engagement, the RAKMS team requested aid with two different rogue radio encounters.
In the first encounter, we were tasked with triangulating the source of an unknown radio emission source which had been bothering the airport for over a week. One of our group members was successful in finding this unknown radio emission source along with the help of two other assistants provided. This involved using an omnidirectional antenna, and once the source was found, it was turned in to the accompanying staff member.
The second encounter consisted of improper manipulation of the baggage claim systems. We were able to capture these packets and crack the decoding procedure

Tools

Enumeration

- nmap
- enum4linux
- Bloodhound
- smbmap
- Powerview
- ldapdomaindump

Vulnerbility Exploiting

- sqlmap
- Nessus
- rpcdump
- Metasploit
- Impacket
- Mimikatz
- Phishing
- MSFvenom

Privledge Escalation

- WinPEAS
- Certipy

Utilities/Lists

- hashcat
- smbclient
- mariadb
- Burpsuite
- Dirsearch/dirbuster/gobuster/fuff/RustScan
- Netcat
- rockyou/SecLists