CTF Write-Up: Secure SFTP Flag Protection

Overview

This document explains the location of the CTF flag within the SFTP jail and the layered security controls implemented to protect it. It also describes possible attack attempts and why they fail under the enforced DAC, MAC, RBAC, and jail mechanisms.

Flag Location

The flag is stored inside the SFTP jail at:

/secret/flag.txt

This path is intentionally protected by multiple access control models and is only accessible by the admin user.

Protection Mechanisms

1. Jail Enforcement

All SFTP paths are resolved using a safe_join() function that ensures:

• No path traversal (e.g., '/../secret' → blocked)

• All files accessed remain strictly under the sftp_root directory

• Attempts to escape the jail are denied at the filesystem level

2. DAC (Discretionary Access Control)

The DAC configuration in data/dac_owners.csv specifies:

• The /secret directory and flag.txt are owned by 'admin'

• Permissions restrict all non-admin users from reading or writing

This ensures only the owner (admin) can access the flag.

3. MAC (Mandatory Access Control)

Using the Bell-LaPadula model from data/mac_labels.json:

• /secret is labeled “secret”

• Users have clearance levels: unclassified < internal < confidential < secret

Only the admin user has “secret” clearance.

Therefore:

• No Read Up: lower-clearance users cannot read the flag

• No Write Down: prevents downgrading sensitive data

4. RBAC (Role-Based Access Control)

Role permissions in data/role_perms.csv ensure:

• Only admin role has read/write privileges for sensitive directories

• Reader, editor, and guest roles cannot perform read on /secret or its contents

Combined Protection

Authorization requires success under:

DAC AND MAC AND RBAC

If any one model denies access, the entire request is denied.

Attack Attempts and Why They Fail

1. Traversal Attacks

Example: read "/../secret/flag.txt"

Outcome: Blocked by safe_join() → PermissionError

2. Low-Clearance Read Attempts

Example: user “test” or “guest” attempts read "/secret/flag.txt"

Outcome: MAC denies due to insufficient clearance

3. Permission Abuse

Example: editors attempt write or forced-open

Outcome: RBAC denies (no read/write permissions)

4. Direct File System Probing

Example: stat, realpath, or readdir on /secret

Outcome: Authorization denies at the SFTP operation level

5. Brute Force Authentication

Authentication is protected by scrypt-secured passwords, so guessing is impractical.

Conclusion

Only the admin user, with matching DAC ownership, MAC clearance, and RBAC permissions, can successfully retrieve /secret/flag.txt. All other users are blocked at multiple layers of defense.