GRC TECHNICAL CASE STUDY — CONTROLLED LAB ENVIRONMENT

Access Control Failure Leading to System Compromise

A GRC Risk Assessment & Compliance Report

Prepared by: Majed Al-Shammari

Date: May 29, 2026
Classification: Confidential
Version: 2.0
Framework: NCA ECC 2018
Environment: Controlled Laboratory

Assessment performed in a controlled laboratory environment.
All findings are based on a simulated scenario for GRC training purposes.

Contents

1 Executive Summary

A security assessment was conducted against a controlled Linux environment running a web application with network capture functionality. A critical chain of vulnerabilities was identified, culminating in full system compromise. The attack path originated from an **Insecure Direct Object Reference (IDOR)** weakness, which exposed sensitive network captures belonging to other users. Analysis of the captured traffic revealed **plaintext credentials**, subsequently reused for unauthorized SSH access. A **misconfigured Linux capability** then enabled privilege escalation to root.

This scenario demonstrates the compounding effect of weak access control, inadequate data protection, and improper privilege management. Each weakness combined to create a complete and severe compromise path.

Overall Risk Rating: CRITICAL **Total Findings:** 4

Assessment performed in a controlled laboratory environment based on the Cap scenario.

2 Incident Overview

2.1 Attack Path

Step 1. Attacker accessed the web application running on the target system.

Step 2. An IDOR vulnerability allowed enumeration and access to other users' Security Snapshots.

Step 3. A PCAP file belonging to a privileged user was accessed without authorization.

Step 4. Sensitive FTP credentials were discovered in plaintext within the packet capture.

Step 5. Credentials were reused to authenticate via SSH (*Password Reuse*).

Step 6. A misconfigured Linux capability on the Python3 interpreter enabled privilege escalation to root.

2.2 Business Impact

- Unauthorized access to sensitive network data and user credentials.
- Full compromise of a privileged user account.
- Root-level system takeover — complete loss of confidentiality, integrity, and availability.
- Potential violation of data protection obligations under PDPL and ISO 27001.
- Reputational and regulatory exposure if the system processed personal or sensitive organizational data.

3 Evidence Reference

The following evidence items were collected during the assessment and are referenced throughout this report.

Evidence ID-01 — IDOR: Unauthorized Resource Access

Unauthorized access to Security Snapshot ID 0 via URL parameter manipulation (`/data/0`).
Type: Application Log Review / Manual Testing

Evidence ID-02 — PCAP Analysis: Cleartext Credential Exposure

PCAP file review revealed an FTP authentication session containing plaintext username and password.
Type: Packet Capture Analysis (Wireshark)

Evidence ID-03 — SSH Authentication via Reused Credential

Successful SSH login achieved using the FTP password recovered from Evidence ID-02.
Type: Authentication Log / Live Testing

Evidence ID-04 — Linux Capability Misconfiguration

Command `getcap -r / 2>/dev/null` revealed `cap_setuid` assigned to `/usr/bin/python3.8`.
Type: System Configuration Review

4 Root Cause Analysis

Finding 1: Insecure Direct Object Reference (IDOR)

Evidence: ID-01

Description: The web application exposed sequential numeric IDs in the URL for Security Snapshot files (e.g., `/data/1`, `/data/2`). No server-side authorization check validated whether the requesting user owned the resource.

Root Cause: Insufficient access control logic. The application relied solely on the URL parameter to determine which resource to serve, without verifying session ownership.

Risk: Any authenticated user can access any other user's network captures by manipulating the ID parameter.

Finding 2: Exposure of Sensitive Credentials in PCAP

Evidence: ID-02

Description: The PCAP file contained an FTP session in which a privileged user transmitted credentials in plaintext. The password was directly visible within the captured packet data.

Root Cause: Use of an unencrypted protocol (FTP) for authenticated sessions. No policy existed prohibiting the transmission of credentials over cleartext channels.

Risk: Credential theft enabling unauthorized access to multiple services.

Finding 3: Password Reuse Across Services

Evidence: ID-03

Description: The FTP password obtained from the PCAP was identical to the user's SSH password, enabling lateral movement from a single credential exposure to full shell access.

Root Cause: Absence of a password uniqueness policy and no enforcement of service-specific credentials.

Risk: A single credential disclosure leads to multi-service compromise.

Finding 4: Misconfigured Linux Capability High

Evidence: ID-04

Description: The binary `/usr/bin/python3.8` was assigned the `cap_setuid` Linux capability, allowing it to set arbitrary user IDs. This was leveraged to escalate privileges to root.

Root Cause: Improper privilege assignment during system configuration with no periodic review of assigned capabilities.

Risk: Any user with access to the binary can escalate to root, resulting in full system compromise.

5 Risk Assessment

5.1 Risk Register

Risk	Likelihood	Impact	Risk Rating
IDOR Vulnerability	High	High	Critical
Credential Exposure (PCAP)	High	High	Critical
Password Reuse	High	High	Critical
Privilege Escalation	Medium	High	High

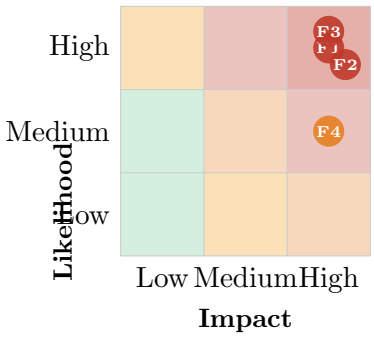
5.2 Risk Ownership

Risk	Risk Owner
IDOR Vulnerability	Application Security Team
Credential Exposure (PCAP)	Infrastructure Team
Password Reuse	Identity & Access Management (IAM) Team
Privilege Escalation	Linux Administration Team

5.3 Residual Risk After Remediation

Finding	Current Risk	Residual Risk	Key Control
IDOR Vulnerability	Critical	Low	Server-side authorization
Credential Exposure (PCAP)	Critical	Medium	SFTP / TLS enforcement
Password Reuse	Critical	Medium	Password uniqueness policy
Privilege Escalation	High	Low	Capability audit & removal

5.4 Risk Heatmap



F1=IDOR F2=Credential Exposure F3=Password Reuse F4=Privilege Escalation

6 Compliance Mapping — NCA ECC 2018

ECC Ref.	Control Domain	Gap Identified	Finding
ECC-2-3	Access Control Management	No server-side ownership validation; IDOR access permitted.	F1
ECC-2-6	Identity & Access Management	Cleartext credentials over FTP; same password reused across SSH and FTP.	F2, F3
ECC-2-10	Secure Configuration Management	`cap_setuid` assigned to Python3 without documented business justification.	F4
ECC-2-12	Cryptography & Key Management	No encryption enforced for authentication (FTP used instead of SFTP/FTPS).	F2
ECC-3-2	Vulnerability Management	No periodic review of application logic or system capabilities to detect misconfig.	F1, F4

7 Recommendations

7.1 Immediate Actions (0–30 Days)

- [**F1**] Implement server-side authorization checks for all resource requests. Validate that the authenticated session owns the requested object before serving it.
- [**F2**] Disable FTP immediately. Replace with SFTP or FTPS. Enforce encrypted protocols for all authenticated sessions across all services.
- [**F3**] Rotate all credentials associated with the compromised account. Enforce unique passwords per service through policy and technical controls.
- [**F4**] Audit and remove unnecessary Linux capabilities from all system binaries: `getcap -r / 2>/dev/null`.

7.2 Long-Term Actions (30–90 Days)

- Conduct a full access control review of all web application endpoints.
- Implement a Password Policy mandating uniqueness and minimum complexity across all systems.
- Integrate automated capability auditing into the hardening baseline (e.g., CIS Benchmark Level 1).
- Perform periodic vulnerability assessments and penetration tests (minimum annually).
- Deploy a SIEM rule to alert on access to capture files outside of authorized administrator sessions.
- Establish a formal Secure Development Lifecycle (SDLC) policy requiring authorization testing before production deployment.

8 Policy Clause — The Golden Policy

Policy: Access Control & Credential Management Policy

Version: 1.0 **Owner:** Information Security Team **Review Cycle:** Annual

4.1 All application resources must be protected by server-side authorization logic. Access to any user-owned object shall be denied unless the requesting session is the verified owner or holds explicit administrative privilege.

4.2 Transmission of authentication credentials over unencrypted protocols (including FTP, Telnet, and plain HTTP) is strictly prohibited. All credential exchange must occur over encrypted channels (TLS 1.2 minimum).

4.3 Users and service accounts must not reuse passwords across different systems or services. Passwords shall be unique per service and managed in accordance with the organization's Credential Management Standard.

4.4 Linux capabilities assigned to system binaries must be reviewed quarterly. Any capability without a documented business justification shall be removed and recorded in the Configuration Baseline Register.

9 Lessons Learned

This scenario illustrates how individual misconfigurations compound into a complete compromise chain. Three key lessons emerge:

1. **Defense-in-Depth is not optional.** A single missing authorization check was the entry point for full root compromise. No single control is sufficient on its own.
2. **Cleartext protocols are a critical organizational risk.** The use of FTP in an environment that produces network captures is a direct path to credential disclosure.
3. **Privilege reviews must be systematic.** The `cap_setuid` capability on a Python interpreter had no legitimate use case. A periodic capability audit would have identified and removed it before exploitation.

10 Conclusion

This assessment demonstrates that the convergence of weak access control, inadequate data protection, and improper privilege management creates a severe and exploitable risk profile. Each finding was directly mapped to **NCA ECC 2018** controls, revealing compliance gaps that carry regulatory and operational significance for any organization.

Proper implementation of server-side authorization, enforced use of encrypted protocols, mandatory password uniqueness, and a rigorous privilege review program would collectively eliminate this attack path and substantially reduce organizational risk.

— End of Report —