

# Cyber Security Projekt 7 Patch Report

## Cybersecurity Project 7 – Secure Update Mechanism & Vulnerability Remediation

**Repository:** <https://github.com/preich21/cybersecurity-project-7>

**Affected versions:** main (legacy / vulnerable)

**Patched version:** fix/cli

**Date:** 09.01.2026

**Authors:** Axel Herrmann, Luca Ilchen, Patrick Reich

---

## 1. Executive Summary

This patch release remediates multiple **critical and high-severity security vulnerabilities** in the legacy update and command execution mechanisms.

The updated implementation introduces a **secure-by-design update pipeline**, hardened input handling, cryptographic verification, and operational safeguards aligned with **CRA, NIS2, and ISO/IEC 27001** requirements.

The fixes eliminate attack vectors including **remote code execution, man-in-the-middle attacks, rollback attacks, data tampering, credential exposure, and denial-of-service conditions**.

---

## 2. Scope of Changes

Area	Legacy State ( main )	Patched State ( fix/cli )
Update transport	HTTP allowed	HTTPS enforced
Integrity verification	None	SHA-256 checksum
Authenticity verification	None	Ed25519 digital signature
Rollback protection	None	Version pinning / anti-rollback
File handling	Direct writes	Atomic file replacement
Input validation	Missing / unsafe	Strict validation & sanitization
Resource limits	Unbounded	Size limits & timeouts
Logging	Sensitive data logged	Redacted / safe logging
Supply-chain visibility	None	SBOM generated
Test coverage	None	Security tests for all vulnerabilities
Password handling	Hardcoded & plain	Via env variables & bcrypt hashed
Authentication	Not enforced	Enforced for all internal functionalities
Supply-chain security	None	UV pyproject.toml & uv.lock
Command execution	Insecure shell based execution	Direct program call

---

### **3. Vulnerabilities Addressed & Patches Applied**

#### **3.1 Command Injection (Critical – CVSS 9.8)**

**CWE:** 78, 77, 88, 94, 20

**OWASP:** Top 10 – Injection

**Issue:**

User-controlled input was concatenated into shell commands, enabling arbitrary command execution.

**Patch:**

- Replaced shell invocation with `subprocess.run(..., shell=False)`
- Enforced strict hostname/IP validation
- Added negative test cases for injection payloads

**Security Impact:**

Prevents remote code execution and system compromise.

---

#### **3.2 Hardcoded Credentials (Critical – CVSS 9.8)**

**CWE:** 798

**OWASP:** Top 10 #2 – Security Misconfiguration

**Issue:**

Secrets and credentials were embedded directly in source code.

**Patch:**

- Removed all hardcoded secrets
- Enforced environment-variable based configuration
- Fail-fast behavior if secrets are missing

**Security Impact:**

Eliminates credential leakage and unauthorized access risk.

---

#### **3.3 Plaintext Logging of Sensitive Data (Medium – CVSS 6.5)**

**CWE:** 532

**OWASP:** Top 10 #9 – Logging & Monitoring Failures

**Issue:**

Authentication-related information was logged in plaintext.

**Patch:**

- Redacted sensitive fields
- Restricted log verbosity
- Logging limited to operational metadata

**Security Impact:**

Prevents credential disclosure via log access.

---

### **3.4 Insecure Hashing (Medium – CVSS 5.3)**

**CWE:** 327, 256

**OWASP:** Top 10 #4 – Cryptographic Failures

**Issue:**

Weak or inappropriate hashing mechanisms were used.

**Patch:**

- Migrated to SHA-256
- Clear separation between hashing (integrity) and signing (authenticity)

**Security Impact:**

Reduces risk of hash collisions and offline attacks.

---

### **3.5 Cleartext Transmission of Sensitive Information (Critical – CVSS 9.8)**

**CWE:** 319

**OWASP:** Top 10 #4 – Cryptographic Failures

**Issue:**

Updates could be downloaded over unsecured HTTP.

**Patch:**

- Mandatory HTTPS enforcement
- Optional override only for demo/testing purposes

**Security Impact:**

Prevents MITM attacks and content tampering.

---

### **3.6 Insufficient Verification of Data Authenticity (Critical – CVSS 9.8)**

**CWE:** 345

**OWASP:** Top 10 #8 – Software & Data Integrity Failures

**Issue:**

Downloaded updates were trusted without proof of origin.

**Patch:**

- Introduced signed update manifests
- Ed25519 public key embedded in client

- Signature verification before installation

**Security Impact:**

Ensures updates originate exclusively from the legitimate publisher.

---

### 3.7 Missing Integrity Check (Critical – CVSS 9.1)

**CWE:** 353

**OWASP:** Top 10 #8

**Issue:**

No checksum verification of update payloads.

**Patch:**

- Mandatory SHA-256 checksum verification
- Abort on mismatch

**Security Impact:**

Detects tampering even if transport is compromised.

---

### 3.8 Uncontrolled Resource Consumption (High – CVSS 7.5)

**CWE:** 400

**Issue:**

Unlimited payload sizes and missing timeouts allowed DoS scenarios.

**Patch:**

- Maximum update size enforced
- Network request timeouts configured

**Security Impact:**

Prevents disk exhaustion and hanging connections.

---

### 3.9 TOCTOU Race Condition (Medium – CVSS 6.3)

**CWE:** 367

**Issue:**

Update files could be modified between validation and use.

**Patch:**

- Atomic write strategy using temporary files and `os.replace`

#### **Security Impact:**

Prevents partial or inconsistent update states.

---

### **3.10 Rollback / Downgrade Attacks (High – CVSS 7.4)**

**CWE:** 1328

#### **Issue:**

Older, vulnerable versions could be re-installed.

#### **Patch:**

- Version comparison enforced
- Downgrades blocked by default

#### **Security Impact:**

Prevents reintroduction of known vulnerabilities.

---

## **4. Secure Update Architecture (Post-Patch)**

The updated system implements **defense-in-depth**:

1. HTTPS-only transport
2. Redirect blocking
3. Size & timeout enforcement
4. Manifest-based update metadata
5. SHA-256 integrity verification
6. Ed25519 signature verification (authenticity)
7. Anti-rollback version checks
8. Atomic installation

Each control can be toggled **only for demonstration purposes** via CLI flags, never by default.

---

## **5. Testing & Verification**

- **55 unit and integration tests**
- Dedicated negative tests for:
  - Injection payloads
  - Invalid signatures
  - Checksum mismatches
  - Oversized updates
  - Downgrade attempts
- All tests passing on fix/cli

## 6. Supply-Chain Security & SBOM

- SBOM generated according to **ECMA-424**
  - Dependency versions pinned via **uv**
  - Enables:
    - Post-release vulnerability identification
    - Third-party component tracking
    - Reproducible builds
- 

## 7. Compliance Alignment

Framework	Alignment Achieved
Cyber Resilience Act (CRA)	Secure updates, integrity & authenticity, vulnerability handling
NIS2	Art. 21(2)(d–f): supply-chain security, vulnerability management
ISO/IEC 27001	A.5.9, A.5.23, A.8.8 – asset inventory, supplier security, vulnerability management

---

## 8. Residual Risk & Recommendations

**Residual risk:** Low

### Recommendations:

- Rotate signing keys periodically
  - Store private keys in an HSM or CI secret store
  - Integrate automated dependency scanning (e.g. Dependabot)
  - Log signature verification failures to a SIEM
- 

## 9. Conclusion

This patch transitions the project from an **insecure proof-of-concept** to a **production-grade, compliance-ready implementation**.

All identified vulnerabilities have been systematically mitigated, verified by tests, and mapped to regulatory requirements.

**Status:**  Approved for secure deployment