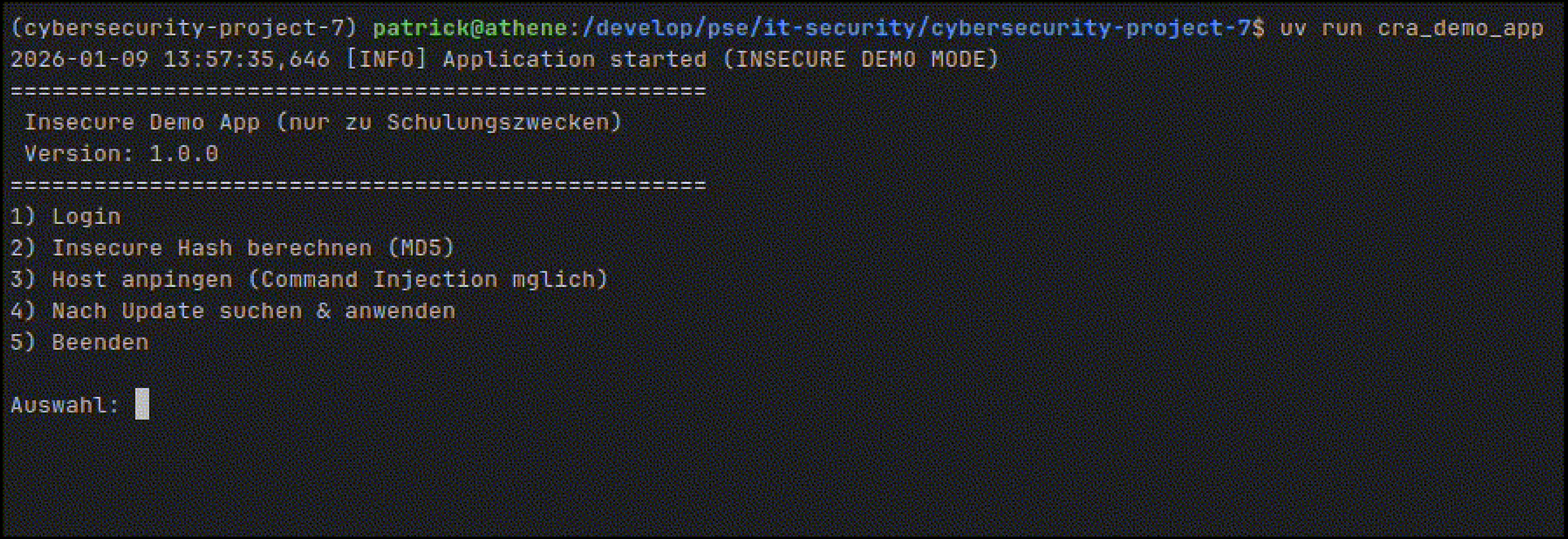
Cybersecurity Project 7 – CRA-compliant patch & vulnerability handling

Axel Herrmann, Luca Ilchen & Patrick Reich – 09.01.2026

Insecure CLI Application

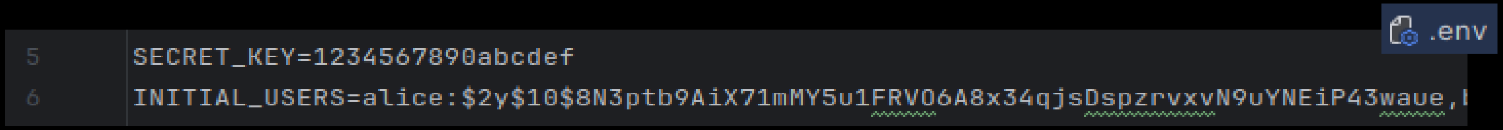


```
(cybersecurity-project-7) patrick@athene:/develop/pse/it-security/cybersecurity-project-7$ uv run cra_demo_app
2026-01-09 13:57:35,646 [INFO] Application started (INSECURE DEMO MODE)
==================================================
 Insecure Demo App (nur zu Schulungszwecken)
 Version: 1.0.0
==================================================
1) Login
2) Insecure Hash berechnen (MD5)
3) Host anpingen (Command Injection mglich)
4) Nach Update suchen & anwenden
5) Beenden

Auswahl: 
```

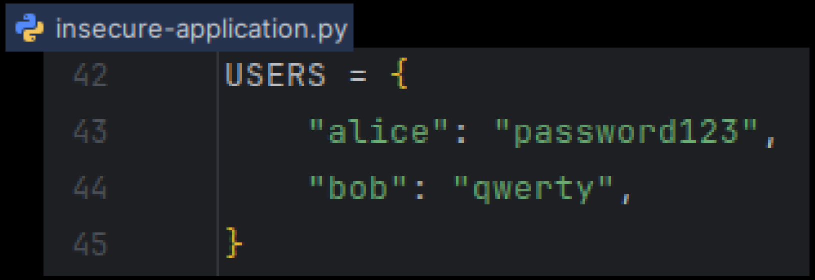
Hardcoded credentials

CWE-798[2]
OWASP Top 10 #2[3]

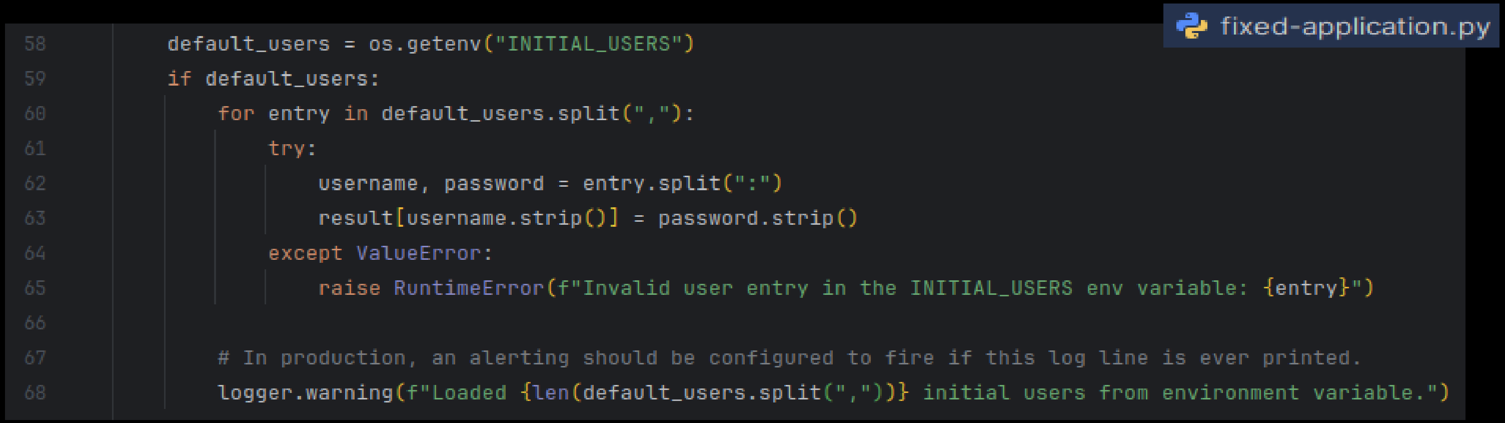
CVSS 9.8
Critical[1]



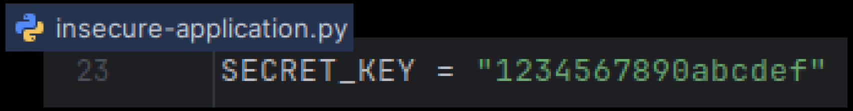
```
SECRET_KEY=1234567890abcdef
INITIAL_USERS=alice:$2y$10$8N3ptb9AiX71mMY5u1FRVO6A8x34qjsDspzrvxvN9uYNEiP43waue,
```



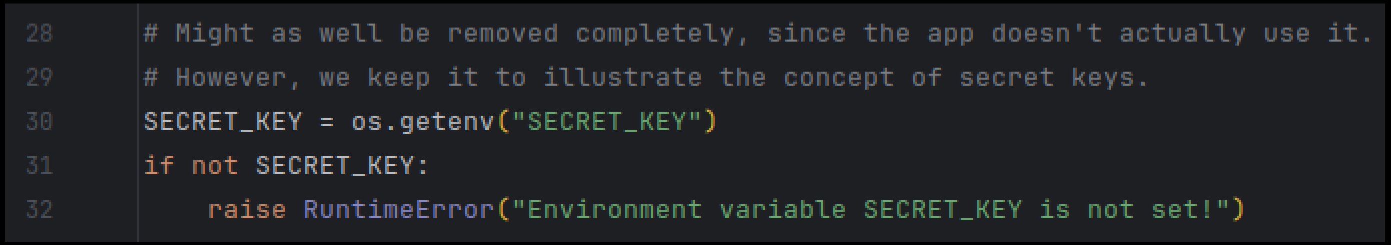
```
USERS = {
    "alice": "password123",
    "bob": "qwerty",
}
```



```
    default_users = os.getenv("INITIAL_USERS")
    if default_users:
        for entry in default_users.split(","):
            try:
                username, password = entry.split(":")
                result[username.strip()] = password.strip()
            except ValueError:
                raise RuntimeError(f"Invalid user entry in the INITIAL_USERS env variable: {entry}")

        # In production, an alerting should be configured to fire if this log line is ever printed.
        logger.warning(f"Loaded {len(default_users.split(","))} initial users from environment variable.")
```



```
SECRET_KEY = "1234567890abcdef"
```



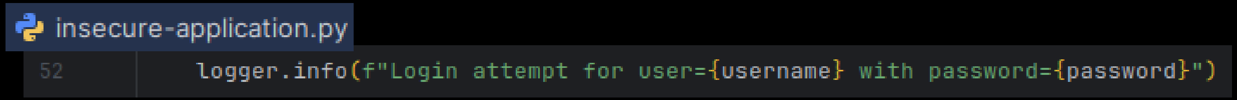
```
# Might as well be removed completely, since the app doesn't actually use it.
# However, we keep it to illustrate the concept of secret keys.
SECRET_KEY = os.getenv("SECRET_KEY")
if not SECRET_KEY:
    raise RuntimeError("Environment variable SECRET_KEY is not set!")
```

[1] https://www.first.org/cvss/calculator/3-1#CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
[2] https://cwe.mitre.org/data/definitions/798.html
[3] https://owasp.org/Top10/2025/A02_2025-Security_Misconfiguration/

Plaintext Logging

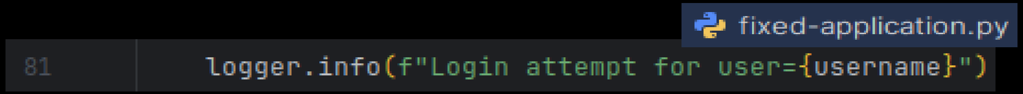
CWE-532[2]
OWASP Top 10 #9[3]

CVSS 6.5
Medium[1]



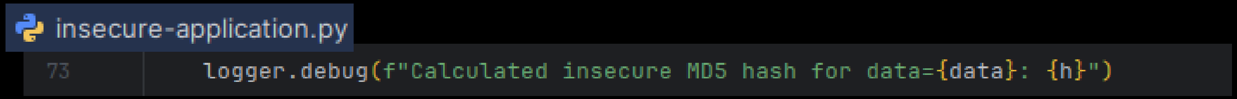
insecure-application.py

```
        logger.info(f"Login attempt for user={username} with password={password}")
```



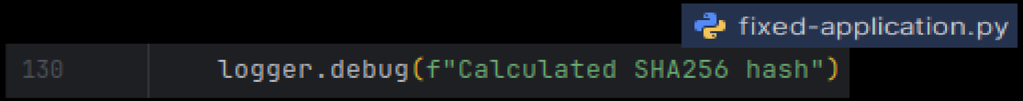
fixed-application.py

```
        logger.info(f"Login attempt for user={username}")
```



insecure-application.py

```
        logger.debug(f"Calculated insecure MD5 hash for data={data}: {h}")
```



fixed-application.py

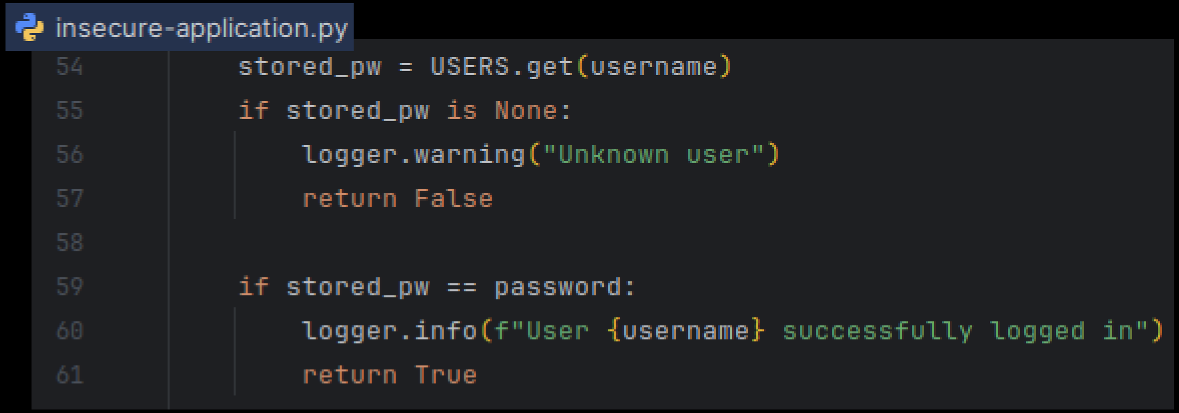
```
        logger.debug(f"Calculated SHA256 hash")
```

[1] https://www.first.org/cvss/calculator/3-1#CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N
[2] https://cwe.mitre.org/data/definitions/532.html
[3] https://owasp.org/Top10/2025/A09_2025-Security_Logging_and_Alerting_Failures/

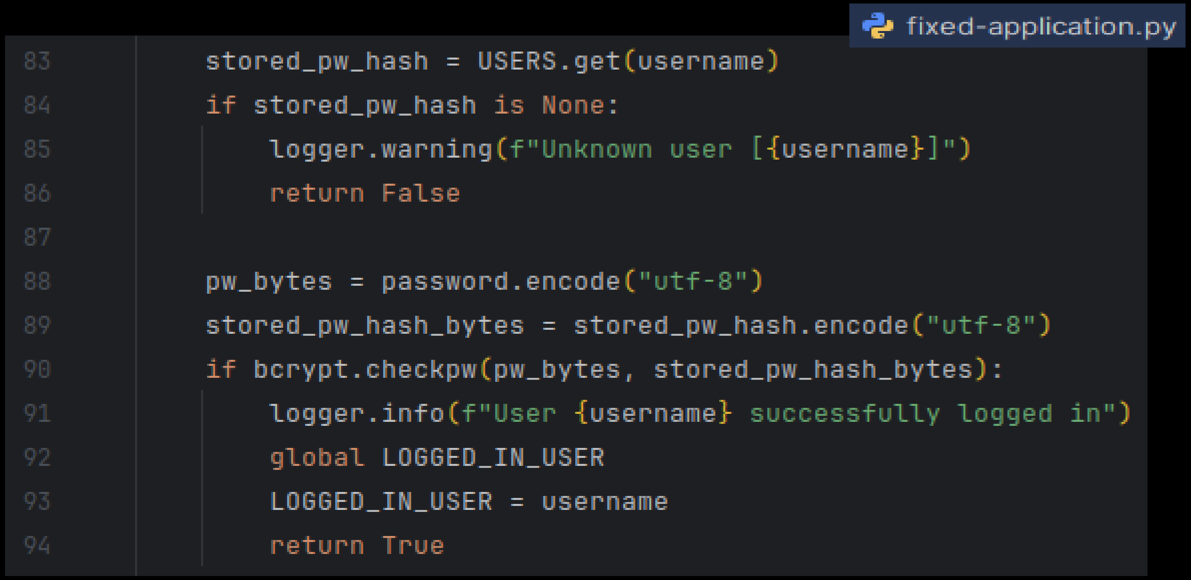
Insecure Hashing

CWE-327[2]
CWE-256[3]
OWASP Top 10 #4[4]

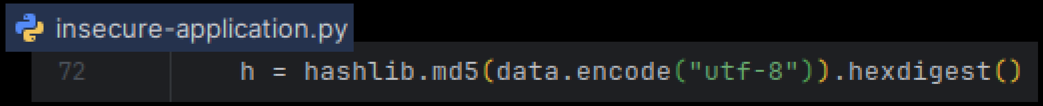
CVSS 5.3
Medium[1]



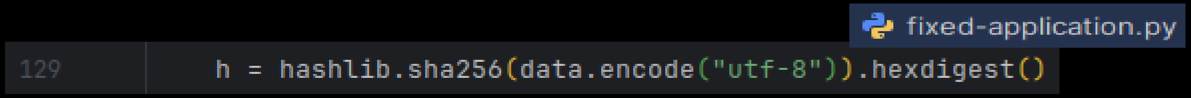
```
    stored_pw = USERS.get(username)
    if stored_pw is None:
        logger.warning("Unknown user")
        return False

    if stored_pw == password:
        logger.info(f"User {username} successfully logged in")
        return True
```



```
    stored_pw_hash = USERS.get(username)
    if stored_pw_hash is None:
        logger.warning(f"Unknown user [{username}]")
        return False

    pw_bytes = password.encode("utf-8")
    stored_pw_hash_bytes = stored_pw_hash.encode("utf-8")
    if bcrypt.checkpw(pw_bytes, stored_pw_hash_bytes):
        logger.info(f"User {username} successfully logged in")
        global LOGGED_IN_USER
        LOGGED_IN_USER = username
        return True
```



```
    h = hashlib.md5(data.encode("utf-8")).hexdigest()
```



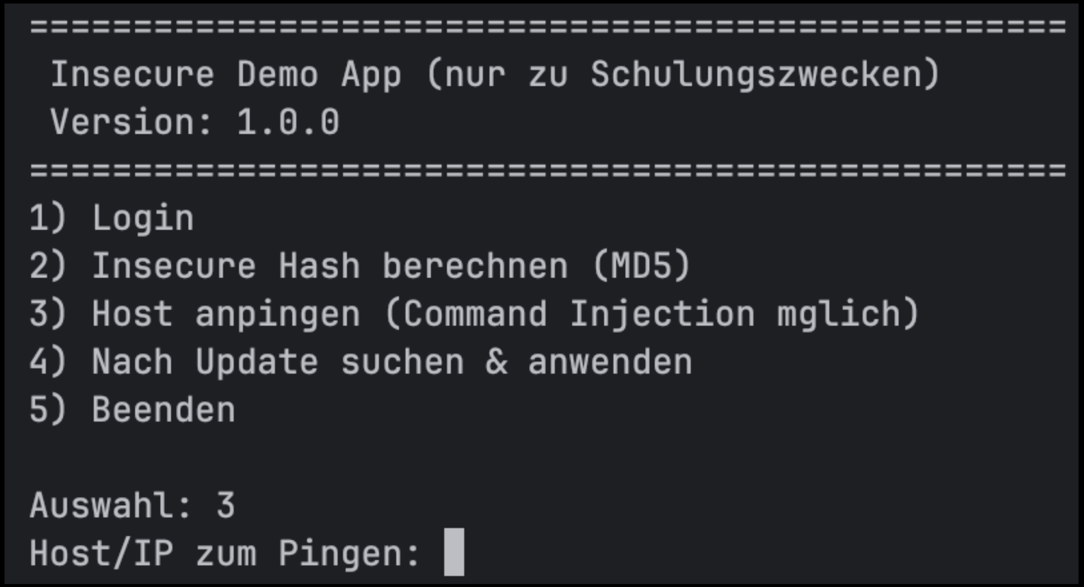
```
    h = hashlib.sha256(data.encode("utf-8")).hexdigest()
```

[1] https://www.first.org/cvss/calculator/3-1#CVSS:3.1/AV:N/AC:H/PR:L/UI:N/S:U/C:H/I:N/A:N
[2] https://cwe.mitre.org/data/definitions/327.html
[3] https://cwe.mitre.org/data/definitions/256.html
[4] https://owasp.org/Top10/2025/A04_2025-Cryptographic_Failures/

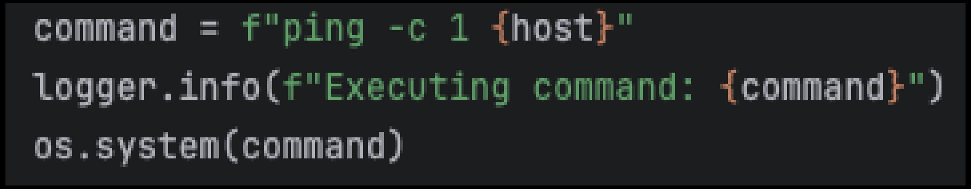
Command Injection

CWE-78[2]
CWE-77[3]
CWE-88[4]
CWE-94[5]
CWE-20[6]

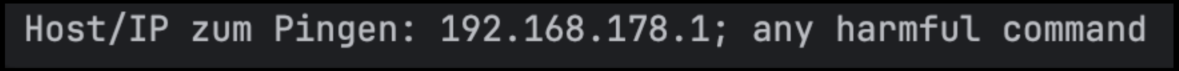
CVSS 9.8 Critical[1]



```
==================================================
 Insecure Demo App (nur zu Schulungszwecken)
 Version: 1.0.0
==================================================
1) Login
2) Insecure Hash berechnen (MD5)
3) Host anpingen (Command Injection mglich)
4) Nach Update suchen & anwenden
5) Beenden

Auswahl: 3
Host/IP zum Pingen: █
```



```
command = f"ping -c 1 {host}"
logger.info(f"Executing command: {command}")
os.system(command)
```



```
Host/IP zum Pingen: 192.168.178.1; any harmful command
```

[1] https://www.first.org/cvss/calculator/3-1#CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

[2] https://cwe.mitre.org/data/definitions/78.html

[3] https://cwe.mitre.org/data/definitions/77.html

[4] https://cwe.mitre.org/data/definitions/88.html

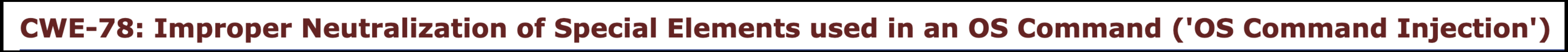
[5] https://cwe.mitre.org/data/definitions/94.html

[6] https://cwe.mitre.org/data/definitions/20.html

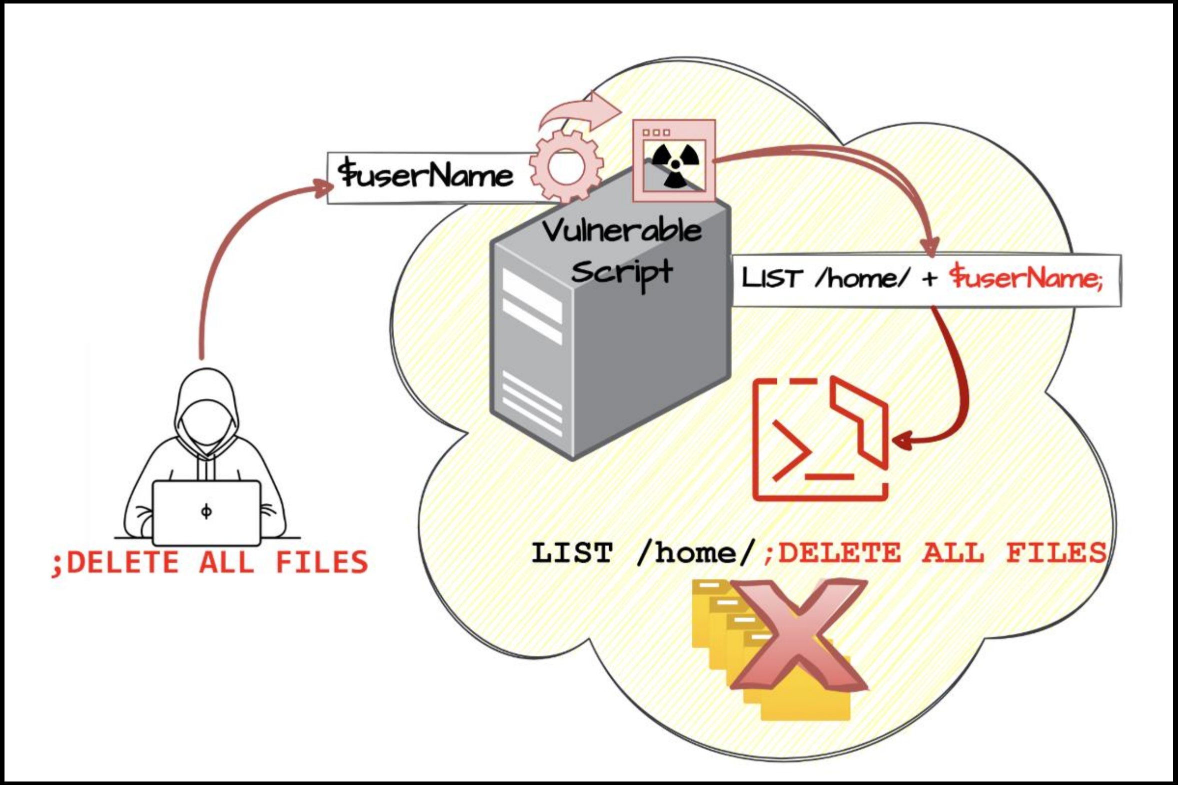
Command Injection

CWE-78[2]
CWE-77[3]
CWE-88[4]
CWE-94[5]
CWE-20[6]

CVSS 9.8 Critical[1]



CWE-78: Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')



[1] https://www.first.org/cvss/calculator/3-1#CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

[2] https://cwe.mitre.org/data/definitions/78.html

[3] https://cwe.mitre.org/data/definitions/77.html

[4] https://cwe.mitre.org/data/definitions/88.html

[5] https://cwe.mitre.org/data/definitions/94.html

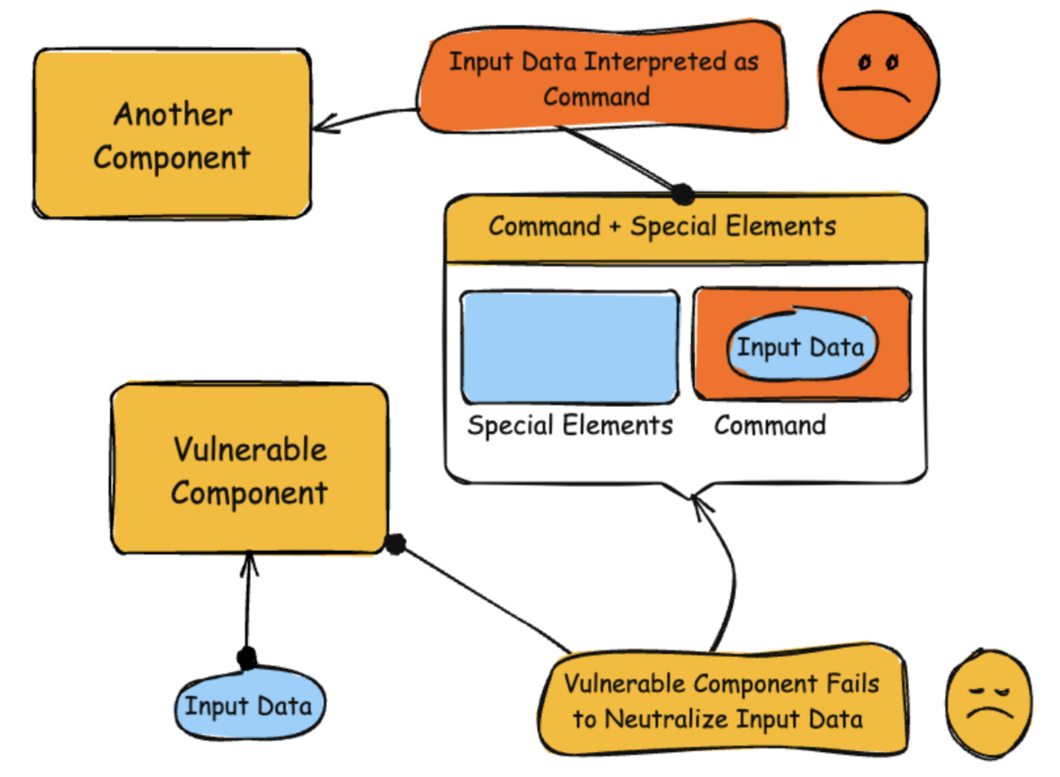
[6] https://cwe.mitre.org/data/definitions/20.html

Command Injection

CWE-78[2]
CWE-77[3]
CWE-88[4]
CWE-94[5]
CWE-20[6]

CVSS 9.8
Critical[1]

CWE-77: Improper Neutralization of Special Elements used in a Command ('Command Injection')


Another Component
Input Data Interpreted as Command
Command + Special Elements
Input Data
Special Elements
Command
Vulnerable Component
Input Data
Vulnerable Component Fails to Neutralize Input Data

[1] https://www.first.org/cvss/calculator/3-1#CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

[2] https://cwe.mitre.org/data/definitions/78.html

[3] https://cwe.mitre.org/data/definitions/77.html

[4] https://cwe.mitre.org/data/definitions/88.html

[5] https://cwe.mitre.org/data/definitions/94.html

[6] https://cwe.mitre.org/data/definitions/20.html

Command Injection

CWE-78[2]
CWE-77[3]
CWE-88[4]
CWE-94[5]
CWE-20[6]

CVSS 9.8
Critical[1]

CWE-88: Improper Neutralization of Argument Delimiters in a Command ('Argument Injection')

CWE-94: Improper Control of Generation of Code ('Code Injection')

CWE-20: Improper Input Validation

[1] https://www.first.org/cvss/calculator/3-1#CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

[2] https://cwe.mitre.org/data/definitions/78.html

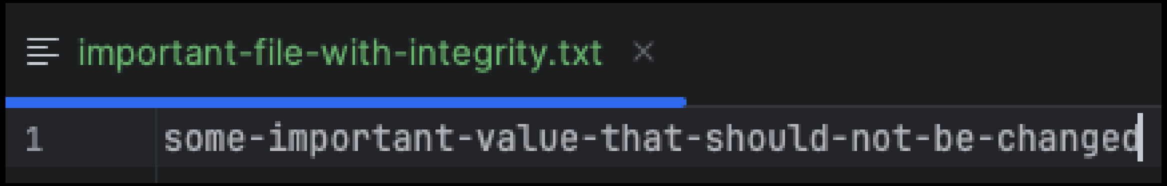
[3] https://cwe.mitre.org/data/definitions/77.html

[4] https://cwe.mitre.org/data/definitions/88.html

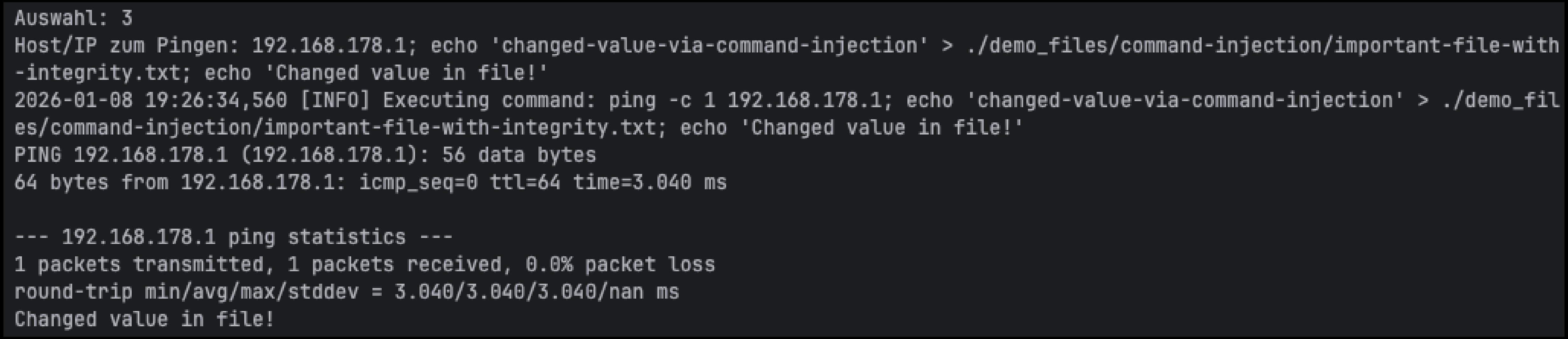
[5] https://cwe.mitre.org/data/definitions/94.html

[6] https://cwe.mitre.org/data/definitions/20.html

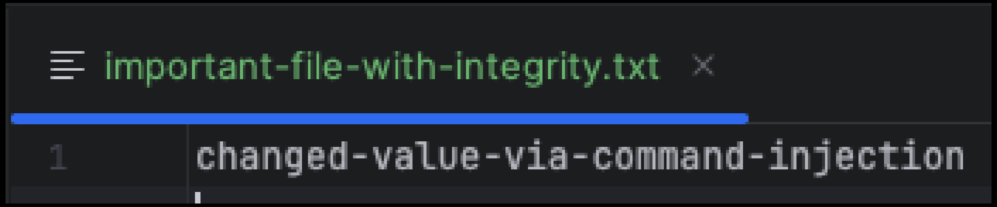
Command Injection - Exploit



```
some-important-value-that-should-not-be-changed
```

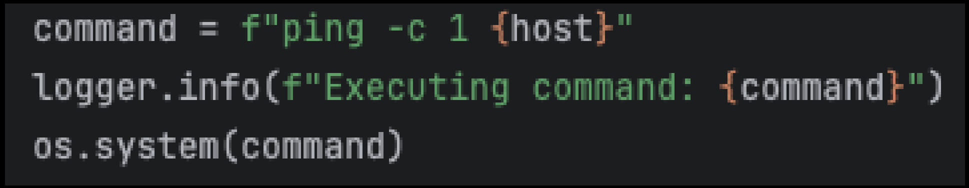


```
Auswahl: 3
Host/IP zum Pingen: 192.168.178.1; echo 'changed-value-via-command-injection' > ./demo_files/command-injection/important-file-with
-integrity.txt; echo 'Changed value in file!'
2026-01-08 19:26:34,560 [INFO] Executing command: ping -c 1 192.168.178.1; echo 'changed-value-via-command-injection' > ./demo_fil
es/command-injection/important-file-with-integrity.txt; echo 'Changed value in file!'
PING 192.168.178.1 (192.168.178.1): 56 data bytes
64 bytes from 192.168.178.1: icmp_seq=0 ttl=64 time=3.040 ms

--- 192.168.178.1 ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 3.040/3.040/3.040/nan ms
Changed value in file!
```

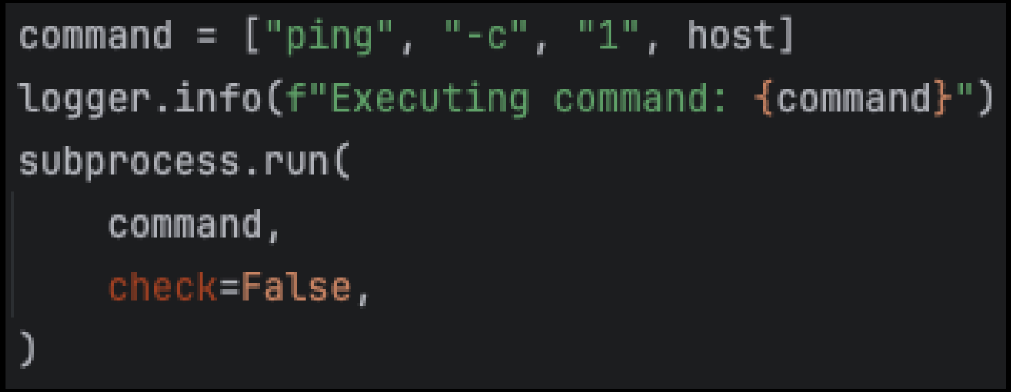


```
changed-value-via-command-injection
```

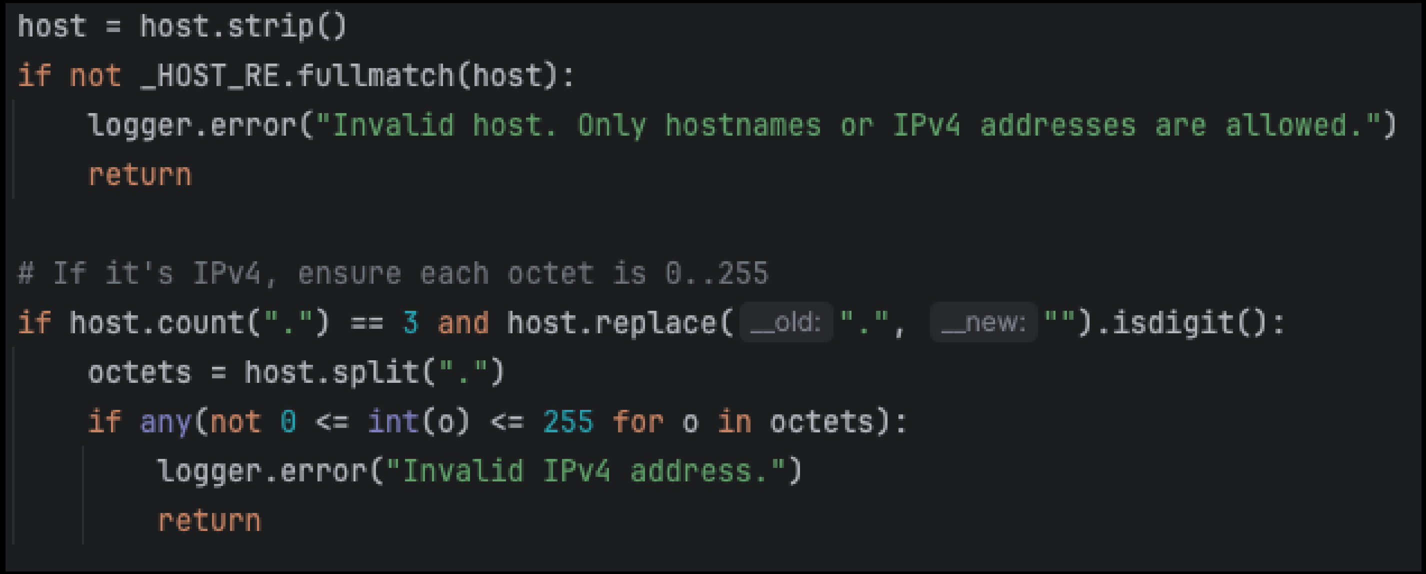
Command Injection - Fix



```python
command = f"ping -c 1 {host}"
logger.info(f"Executing command: {command}")
os.system(command)
```



```python
command = ["ping", "-c", "1", host]
logger.info(f"Executing command: {command}")
subprocess.run(
    command,
    check=False,
)
```

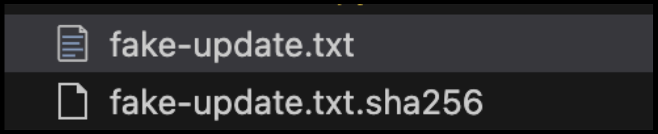


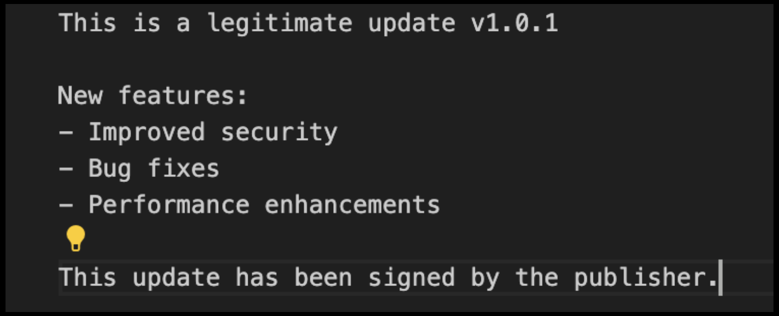
```python
host = host.strip()
if not _HOST_RE.fullmatch(host):
    logger.error("Invalid host. Only hostnames or IPv4 addresses are allowed.")
    return

# If it's IPv4, ensure each octet is 0..255
if host.count(".") == 3 and host.replace( __old: ".",  __new: "").isdigit():
    octets = host.split(".")
    if any(not 0 <= int(o) <= 255 for o in octets):
        logger.error("Invalid IPv4 address.")
        return
```

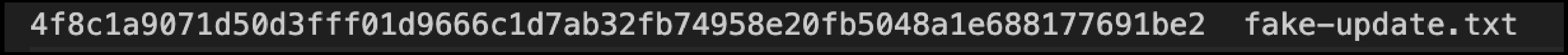
UPDATE MECHANISM

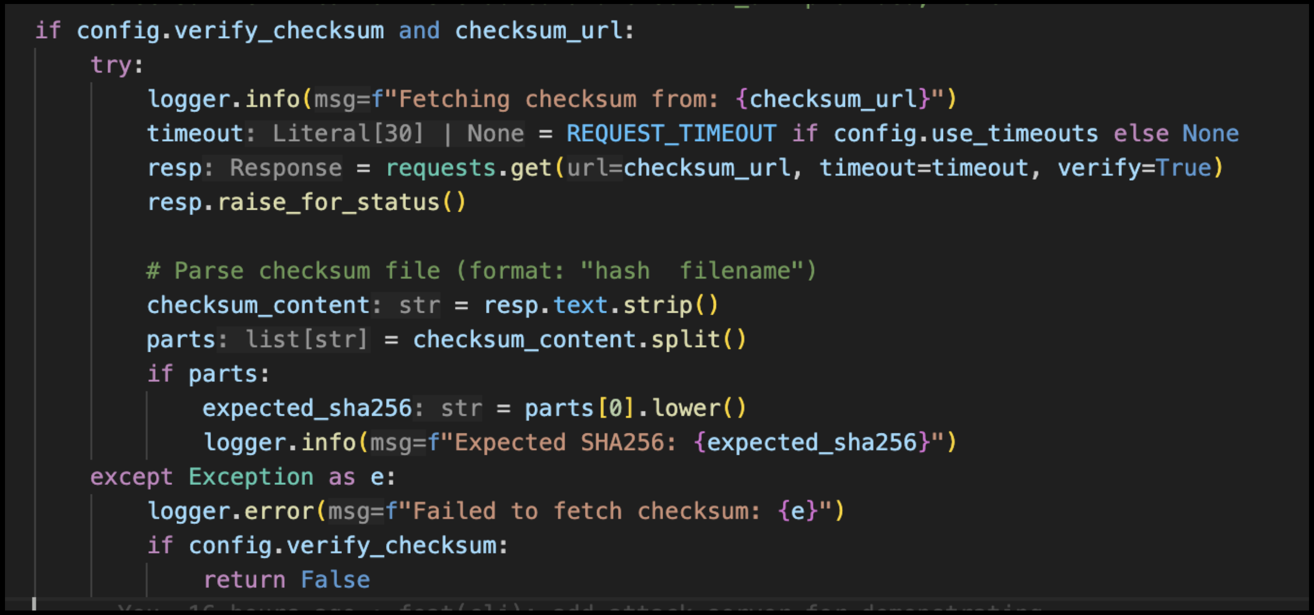
Added Security Features:	**Can be toggled for demo reasons with:**
- Require HTTPS instead of HTTP	--no-https
- Verify SHA256 hash of downloaded content	--no-checksum
- Cryptographically verify update authenticity	--no-signature
- Enforce maximum update size	--no-size-limit
- Block downgrades to older versions	--no-rollback
- Set request timeouts	--no-timeouts
- Use atomic file operations	--no-atomic
- Block HTTP redirects	--allow-redirects

Adding a checksum


```
fake-update.txt
fake-update.txt.sha256
```


```
This is a legitimate update v1.0.1

New features:
– Improved security
– Bug fixes
– Performance enhancements

This update has been signed by the publisher.
```

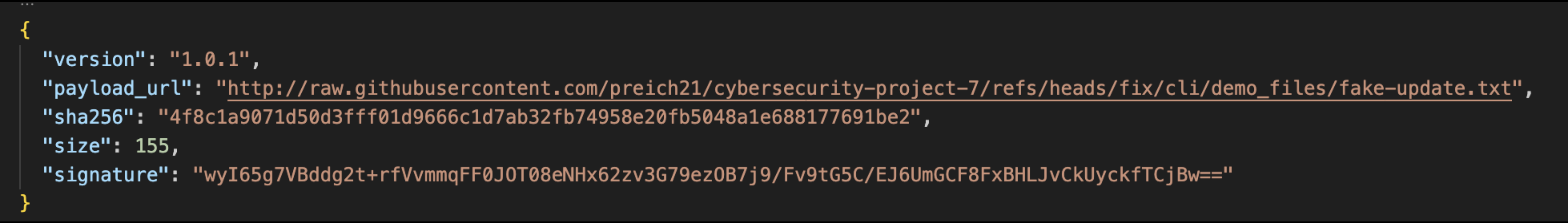

```
4f8c1a9071d50d3fff01d9666c1d7ab32fb74958e20fb5048a1e688177691be2  fake–update.txt
```


```python
if config.verify_checksum and checksum_url:
    try:
        logger.info(msg=f"Fetching checksum from: {checksum_url}")
        timeout: Literal[30] | None = REQUEST_TIMEOUT if config.use_timeouts else None
        resp: Response = requests.get(url=checksum_url, timeout=timeout, verify=True)
        resp.raise_for_status()

        # Parse checksum file (format: "hash  filename")
        checksum_content: str = resp.text.strip()
        parts: list[str] = checksum_content.split()
        if parts:
            expected_sha256: str = parts[0].lower()
            logger.info(msg=f"Expected SHA256: {expected_sha256}")
    except Exception as e:
        logger.error(msg=f"Failed to fetch checksum: {e}")
        if config.verify_checksum:
            return False
```

Provides only INTEGRITY

-> Detects tampering (MITM)
-> Does not prove its coming from the correct the origin

The update manifest



```
...
{
  "version": "1.0.1",
  "payload_url": "http://raw.githubusercontent.com/preich21/cybersecurity-project-7/refs/heads/fix/cli/demo_files/fake-update.txt",
  "sha256": "4f8c1a9071d50d3fff01d9666c1d7ab32fb74958e20fb5048a1e688177691be2",
  "size": 155,
  "signature": "wyI65g7VBddg2t+rfVvmmqFF0JOT08eNHx62zv3G79ezOB7j9/Fv9tG5C/EJ6UmGCF8FxBHLJvCkUyckfTCjBw=="
}
```

Including size check, sha256 checksum and signature with a keypair (private key on the publisher side, public key in the software code)

Provides INTEGRITY and AUTHENTICITY
-> If signature matches, we know its coming from the original publisher (only one with private key)
-> If the checksum matches, we know its wasn't tampered with on the way from the publisher to us

Other improvements

Require HTTPS instead of HTTP

Prevents man-in-the-middle (MITM) attacks where an attacker intercepts or modifies the update in transit

Enforce maximum update size

Mitigates denial-of-service (DoS) and disk exhaustion attacks using oversized or malformed update files.

Block downgrades to older versions

Prevents rollback attacks where an attacker forces installation of a vulnerable but validly signed older version.

Set request timeouts

Protects against resource exhaustion / hanging connections (e.g. slow-loris–style attacks or stalled servers).

Use atomic file operations

Prevents partial or inconsistent updates caused by crashes or interruptions, which could lead to corruption or code execution issues.

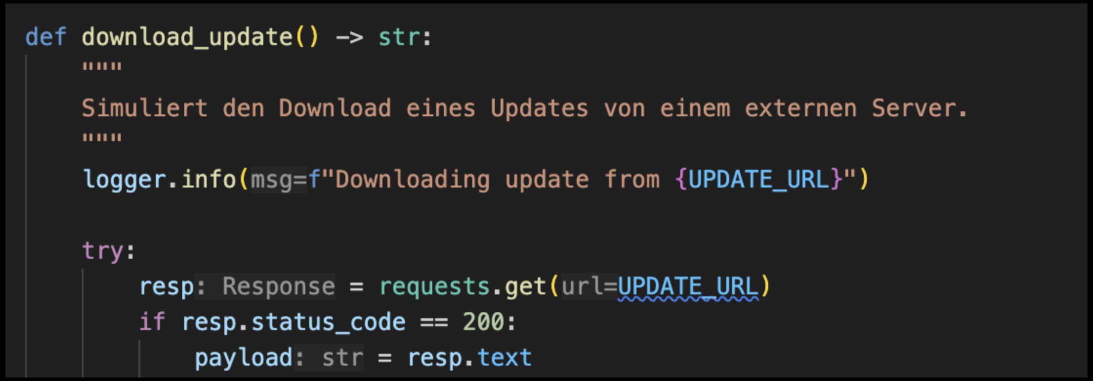
Block HTTP redirects

Prevents redirect-based attacks where a legitimate URL forwards to a malicious update server.

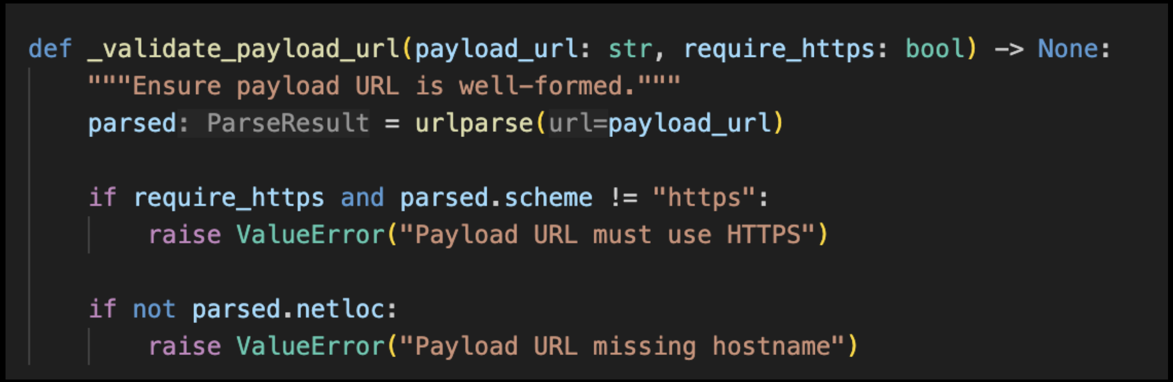
Cleartext Transmission of Sensitive Information

CWE-319[2]
OWASP Top 10 #4[3]

CVSS 9.8
Critical[1]



```
def download_update() -> str:
    """
    Simuliert den Download eines Updates von einem externen Server.
    """
    logger.info(msg=f"Downloading update from {UPDATE_URL}")

    try:
        resp: Response = requests.get(url=UPDATE_URL)
        if resp.status_code == 200:
            payload: str = resp.text
```



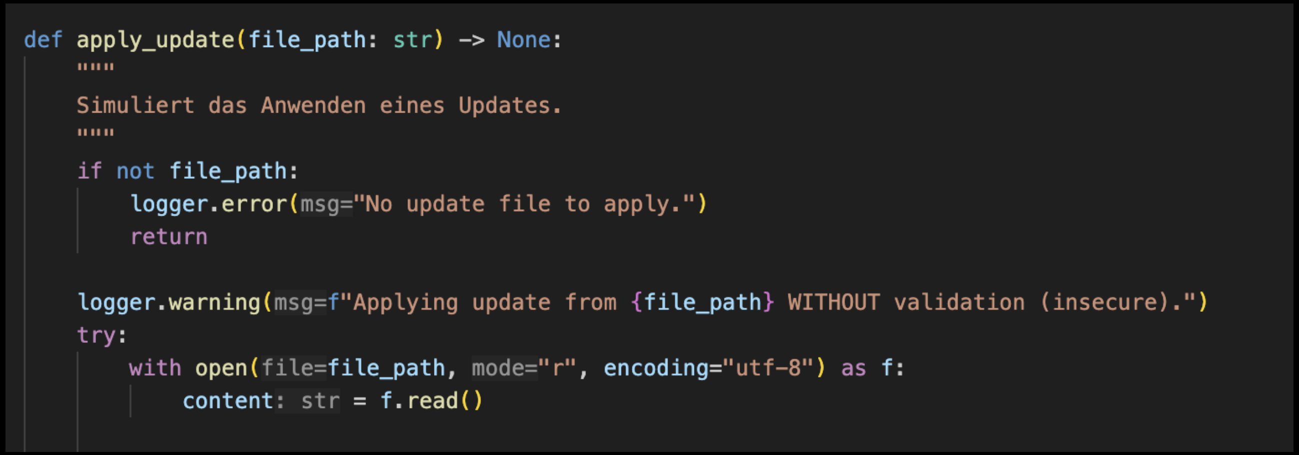
```
def _validate_payload_url(payload_url: str, require_https: bool) -> None:
    """Ensure payload URL is well-formed."""
    parsed: ParseResult = urlparse(url=payload_url)

    if require_https and parsed.scheme != "https":
        raise ValueError("Payload URL must use HTTPS")

    if not parsed.netloc:
        raise ValueError("Payload URL missing hostname")
```

[1] https://www.first.org/cvss/calculator/3-1#CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
[2] https://cwe.mitre.org/data/definitions/319.html
[3] https://owasp.org/Top10/2025/A04_2025-Cryptographic_Failures/

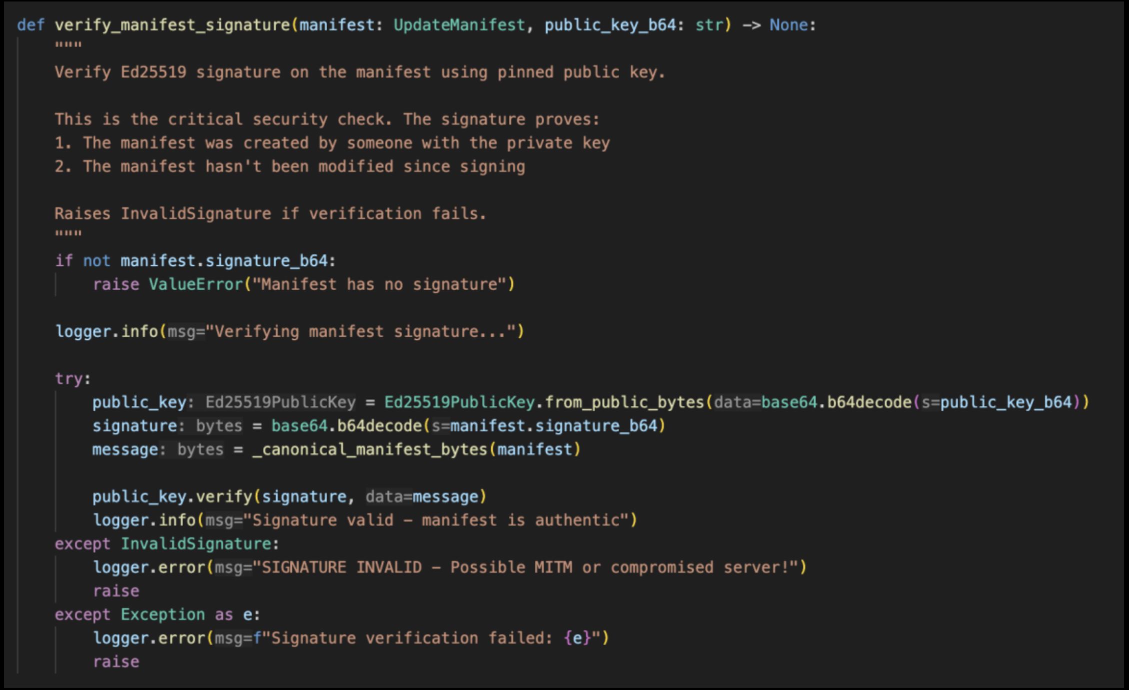
Insufficient Verification of Data Authenticity

CWE-345[2]
OWASP Top 10 #8[3]

CVSS 9.8 Critical[1]



```python
def apply_update(file_path: str) -> None:
    """
    Simuliert das Anwenden eines Updates.
    """
    if not file_path:
        logger.error(msg="No update file to apply.")
        return

    logger.warning(msg=f"Applying update from {file_path} WITHOUT validation (insecure).")
    try:
        with open(file=file_path, mode="r", encoding="utf-8") as f:
            content: str = f.read()
```



```python
def verify_manifest_signature(manifest: UpdateManifest, public_key_b64: str) -> None:
    """
    Verify Ed25519 signature on the manifest using pinned public key.

    This is the critical security check. The signature proves:
    1. The manifest was created by someone with the private key
    2. The manifest hasn't been modified since signing

    Raises InvalidSignature if verification fails.
    """
    if not manifest.signature_b64:
        raise ValueError("Manifest has no signature")

    logger.info(msg="Verifying manifest signature...")

    try:
        public_key: Ed25519PublicKey = Ed25519PublicKey.from_public_bytes(data=base64.b64decode(s=public_key_b64))
        signature: bytes = base64.b64decode(s=manifest.signature_b64)
        message: bytes = _canonical_manifest_bytes(manifest)

        public_key.verify(signature, data=message)
        logger.info(msg="Signature valid - manifest is authentic")
    except InvalidSignature:
        logger.error(msg="SIGNATURE INVALID - Possible MITM or compromised server!")
        raise
    except Exception as e:
        logger.error(msg=f"Signature verification failed: {e}")
        raise
```

[1] https://www.first.org/cvss/calculator/3-1#CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

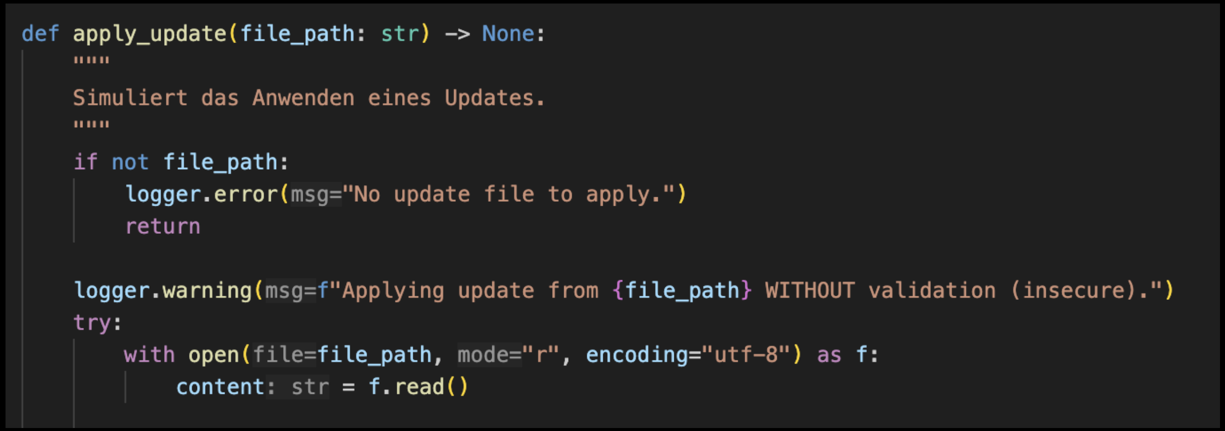
[2] https://cwe.mitre.org/data/definitions/345.html

[3] https://owasp.org/Top10/2025/A08_2025-Software_or_Data_Integrity_Failures/

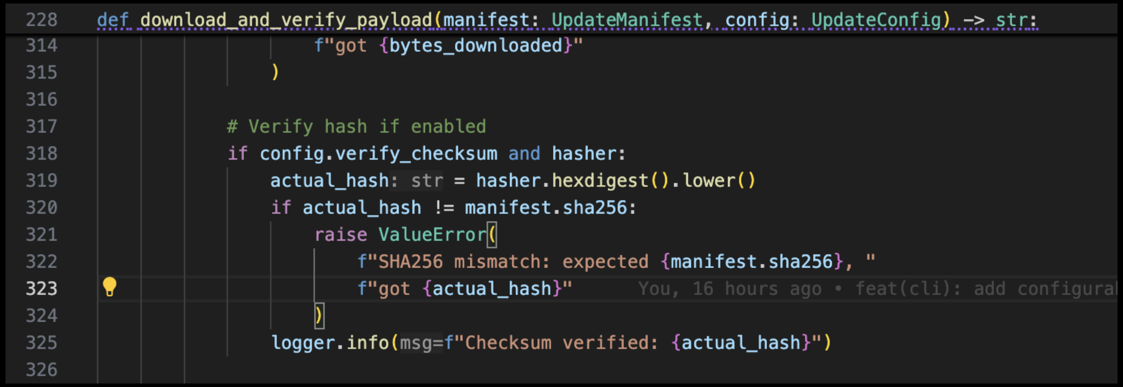
Missing Support for Integrity Check

CWE-353[2]
OWASP Top 10 #8[3]

CVSS 9.1 Critical[1]



```python
def apply_update(file_path: str) -> None:
    """
    Simuliert das Anwenden eines Updates.
    """
    if not file_path:
        logger.error(msg="No update file to apply.")
        return

    logger.warning(msg=f"Applying update from {file_path} WITHOUT validation (insecure).")
    try:
        with open(file=file_path, mode="r", encoding="utf-8") as f:
            content: str = f.read()
```



```python
def download_and_verify_payload(manifest: UpdateManifest, config: UpdateConfig) -> str:
                    f"got {bytes_downloaded}"
                )

            # Verify hash if enabled
            if config.verify_checksum and hasher:
                actual_hash: str = hasher.hexdigest().lower()
                if actual_hash != manifest.sha256:
                    raise ValueError(
                        f"SHA256 mismatch: expected {manifest.sha256}, "
                        f"got {actual_hash}"      You, 16 hours ago • feat(cli): add configura
                    )
                logger.info(msg=f"Checksum verified: {actual_hash}")

```

[1] https://www.first.org/cvss/calculator/3-1#CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N

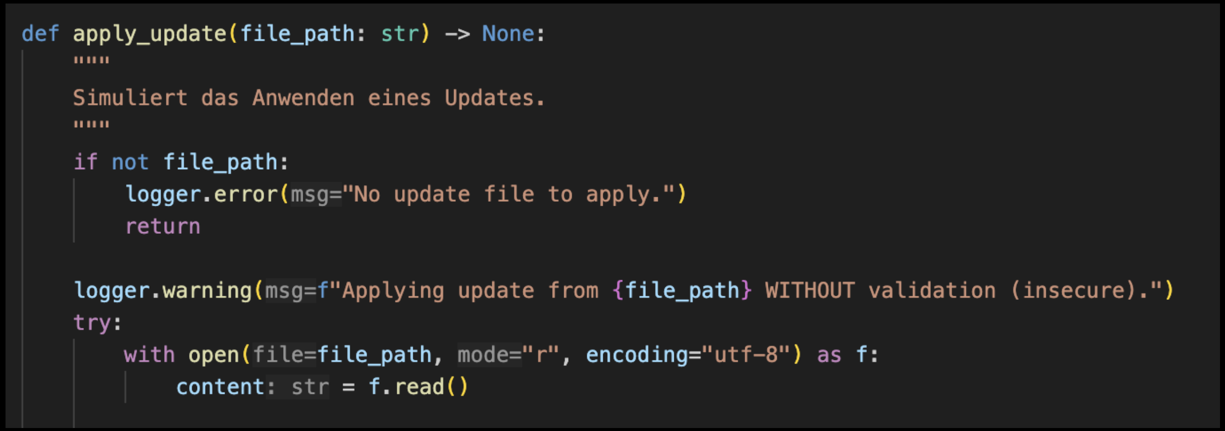
[2] https://cwe.mitre.org/data/definitions/353.html

[3] https://owasp.org/Top10/2025/A08_2025-Software_or_Data_Integrity_Failures/

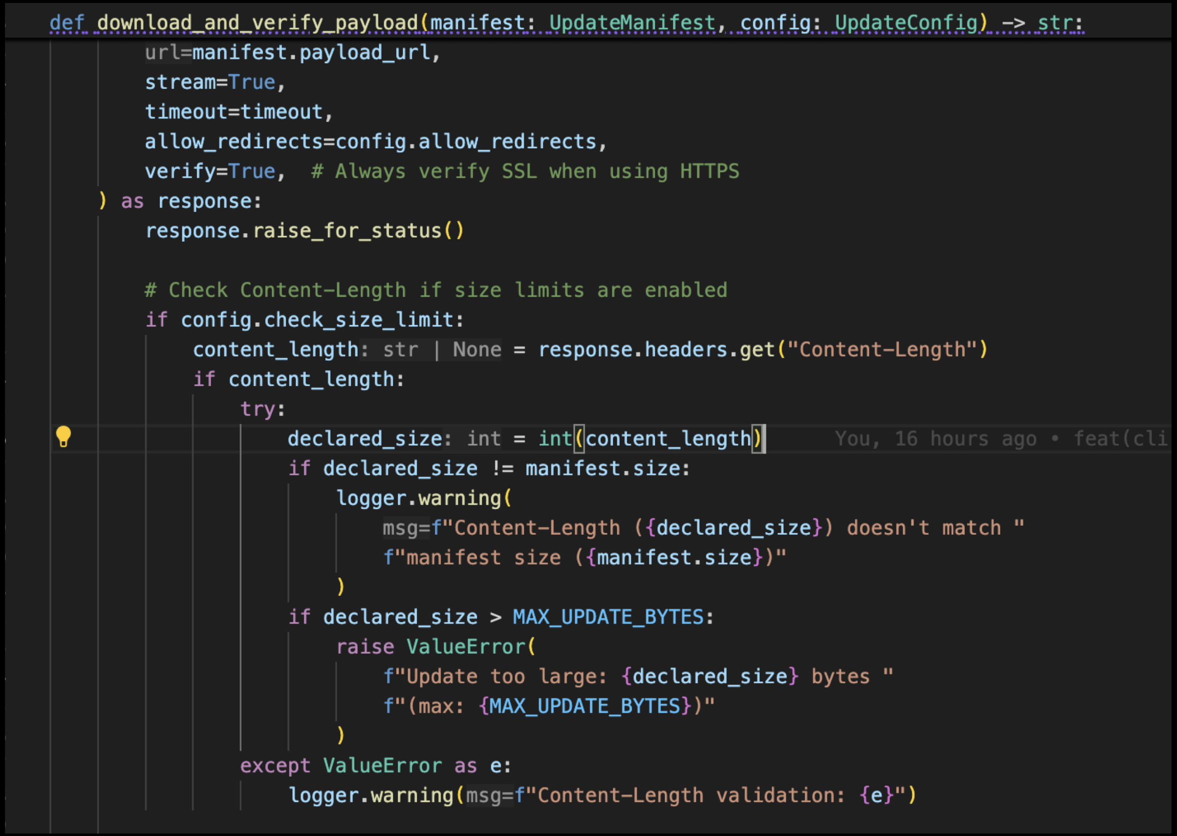
Uncontrolled Resource Consumption

CWE-400[2]

CVSS 7.5
High [1]



```
def apply_update(file_path: str) -> None:
    """
    Simuliert das Anwenden eines Updates.
    """
    if not file_path:
        logger.error(msg="No update file to apply.")
        return

    logger.warning(msg=f"Applying update from {file_path} WITHOUT validation (insecure).")
    try:
        with open(file=file_path, mode="r", encoding="utf-8") as f:
            content: str = f.read()
```



```
def download_and_verify_payload(manifest: UpdateManifest, config: UpdateConfig) -> str:
        url=manifest.payload_url,
        stream=True,
        timeout=timeout,
        allow_redirects=config.allow_redirects,
        verify=True,  # Always verify SSL when using HTTPS
    ) as response:
        response.raise_for_status()

        # Check Content-Length if size limits are enabled
        if config.check_size_limit:
            content_length: str | None = response.headers.get("Content-Length")
            if content_length:
                try:
                    declared_size: int = int(content_length)      You, 16 hours ago • feat(cli
                    if declared_size != manifest.size:
                        logger.warning(
                            msg=f"Content-Length ({declared_size}) doesn't match "
                            f"manifest size ({manifest.size})"
                        )
                    if declared_size > MAX_UPDATE_BYTES:
                        raise ValueError(
                            f"Update too large: {declared_size} bytes "
                            f"(max: {MAX_UPDATE_BYTES})"
                        )
                except ValueError as e:
                    logger.warning(msg=f"Content-Length validation: {e}")
```

[1] https://www.first.org/cvss/calculator/3-1#CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H

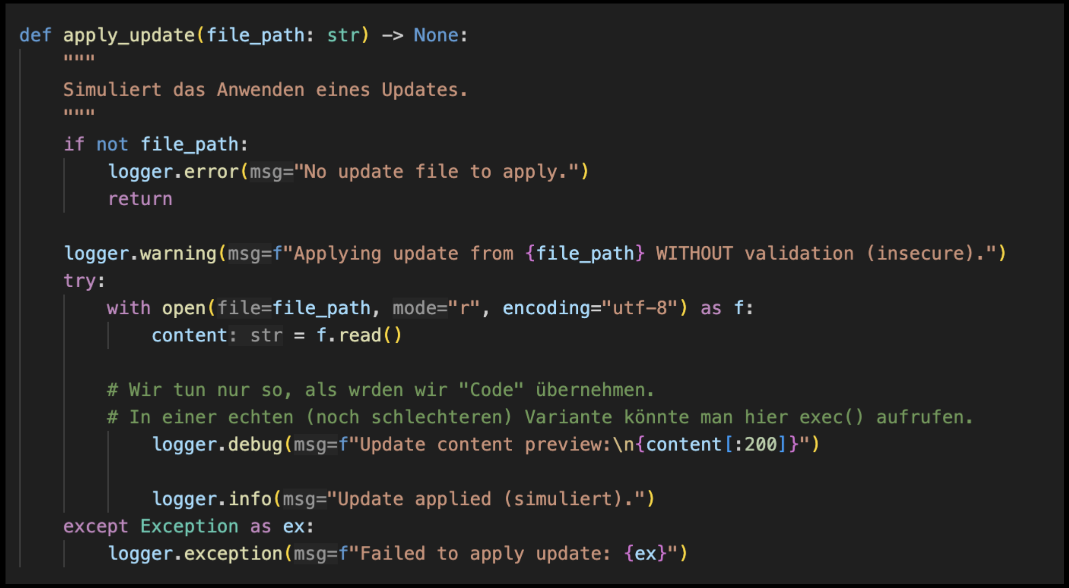
[2] https://cwe.mitre.org/data/definitions/400.html

Time-of-check Time-of-Use Race Condition

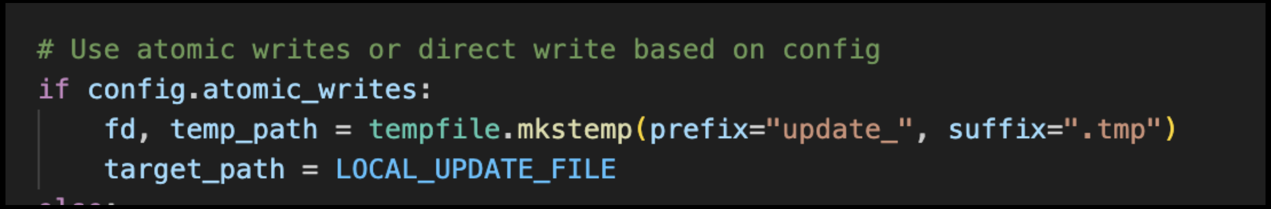
CWE-367[2]

CVSS 6.3 Medium[1]

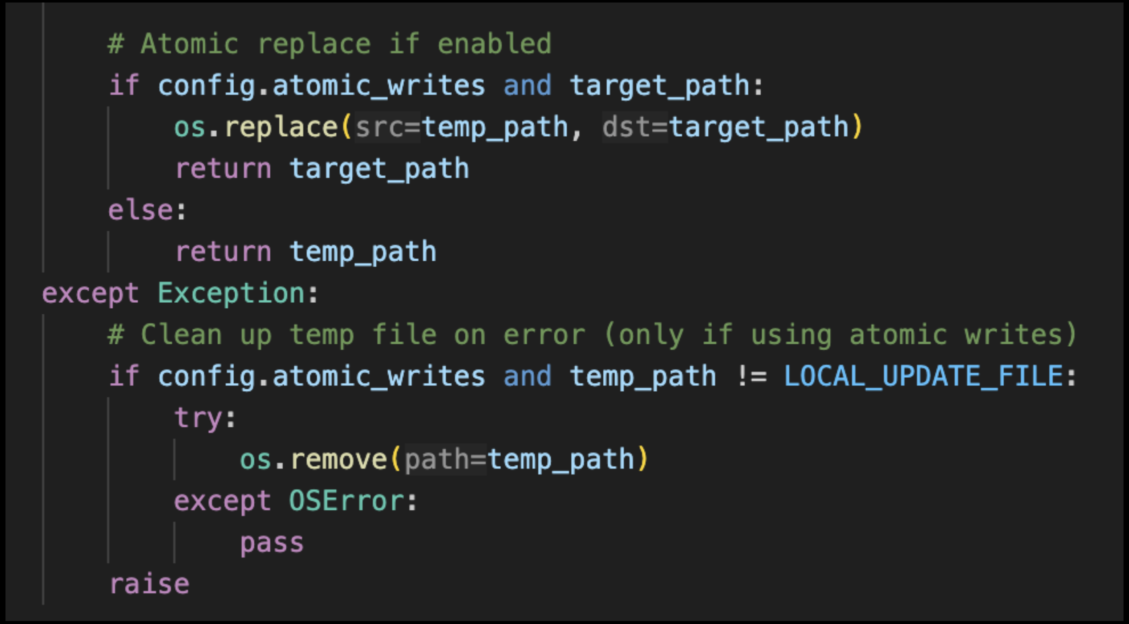
prevents inconsistent/partial state from being used



```python
def apply_update(file_path: str) -> None:
    """
    Simuliert das Anwenden eines Updates.
    """
    if not file_path:
        logger.error(msg="No update file to apply.")
        return

    logger.warning(msg=f"Applying update from {file_path} WITHOUT validation (insecure).")
    try:
        with open(file=file_path, mode="r", encoding="utf-8") as f:
            content: str = f.read()

        # Wir tun nur so, als wrden wir "Code" übernehmen.
        # In einer echten (noch schlechteren) Variante könnte man hier exec() aufrufen.
            logger.debug(msg=f"Update content preview:\n{content[:200]}")

            logger.info(msg="Update applied (simuliert).")
    except Exception as ex:
        logger.exception(msg=f"Failed to apply update: {ex}")
```



```python
# Use atomic writes or direct write based on config
if config.atomic_writes:
    fd, temp_path = tempfile.mkstemp(prefix="update_", suffix=".tmp")
    target_path = LOCAL_UPDATE_FILE
```



```python
    # Atomic replace if enabled
    if config.atomic_writes and target_path:
        os.replace(src=temp_path, dst=target_path)
        return target_path
    else:
        return temp_path
except Exception:
    # Clean up temp file on error (only if using atomic writes)
    if config.atomic_writes and temp_path != LOCAL_UPDATE_FILE:
        try:
            os.remove(path=temp_path)
        except OSError:
            pass
    raise
```

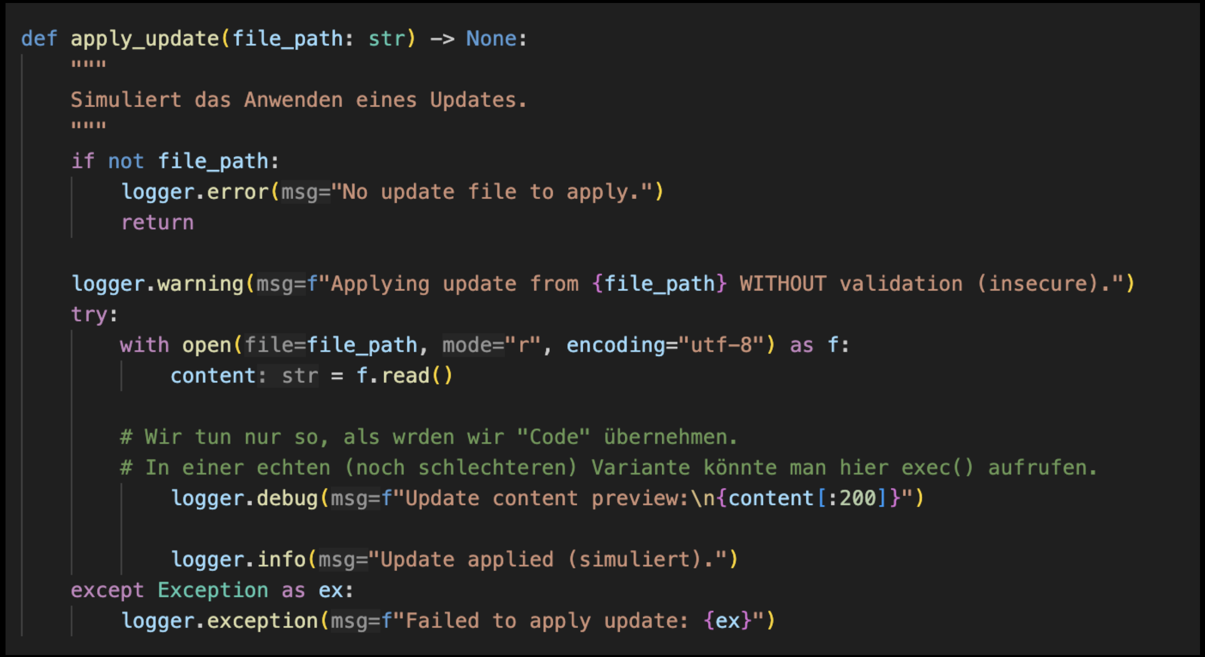
[1] https://www.first.org/cvss/calculator/3-1#CVSS:3.1/AV:L/AC:H/PR:L/UI:N/S:U/C:N/I:H/A:H

[2] https://cwe.mitre.org/data/definitions/367.html

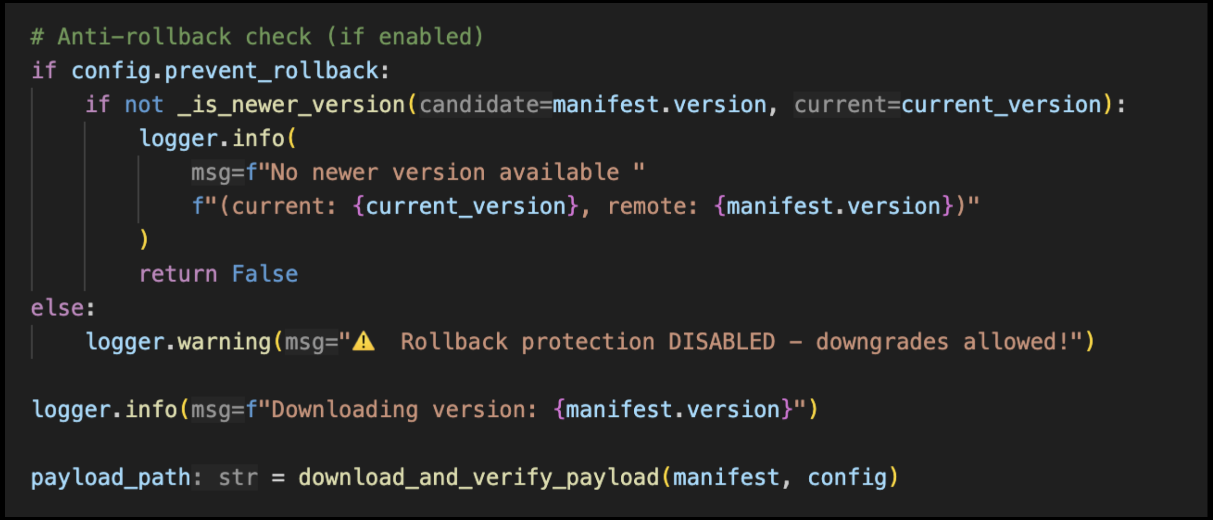
Security Version Number Mutable to Older Version

CWE-1328[2]

CVSS 7.4 High[1]



```python
def apply_update(file_path: str) -> None:
    """
    Simuliert das Anwenden eines Updates.
    """
    if not file_path:
        logger.error(msg="No update file to apply.")
        return

    logger.warning(msg=f"Applying update from {file_path} WITHOUT validation (insecure).")
    try:
        with open(file=file_path, mode="r", encoding="utf-8") as f:
            content: str = f.read()

        # Wir tun nur so, als wrden wir "Code" übernehmen.
        # In einer echten (noch schlechteren) Variante könnte man hier exec() aufrufen.
            logger.debug(msg=f"Update content preview:\n{content[:200]}")

            logger.info(msg="Update applied (simuliert).")
    except Exception as ex:
        logger.exception(msg=f"Failed to apply update: {ex}")
```



```python
# Anti-rollback check (if enabled)
if config.prevent_rollback:
    if not _is_newer_version(candidate=manifest.version, current=current_version):
        logger.info(
            msg=f"No newer version available "
            f"(current: {current_version}, remote: {manifest.version})"
        )
        return False
else:
    logger.warning(msg="⚠️  Rollback protection DISABLED - downgrades allowed!")

logger.info(msg=f"Downloading version: {manifest.version}")

payload_path: str = download_and_verify_payload(manifest, config)
```

[1] https://www.first.org/cvss/calculator/3-1#CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:N

[2] https://cwe.mitre.org/data/definitions/1328.html

Live Demo: Insecure File

Framework Impact

Cyber Resilience Act Violations

Article 6

Products with digital elements shall be designed, developed and produced in accordance with the essential cybersecurity requirements set out in Annex I.

Cyber Resilience Act Violations

Annex I

Part 1

- (a) - Minimise attack surface
  Removal of command injection vectors, Strict input validation (hostnames / IPs), Blocking HTTP redirects, Disabling shell execution, Enforcing HTTPS only
- (b) - Prevent unauthorised access
  Digital signature verification (Ed25519), Public-key pinned in client, No unsigned updates accepted, Environment-based secret handling
- (c) - Protect integrity of data and software
  SHA-256 checksum verification, Signed update manifest, Atomic file replacement, TOCTOU race-condition mitigation
- (d) - Secure default configuration
  HTTPS enabled by default, Rollback protection enabled by default, Size limits enabled by default, Timeouts enabled by default, Security features can only be disabled explicitly (CLI flags)
- (e) - Protection against known attacks
  Well-known classes (OWASP Top 10, CWE Top 25) must be addressed -> see previous slides

Part 2

- (a) - Vulnerability handling process
  Structured update mechanism, Versioned update manifests, Explicit vulnerability fixes, Test coverage proving mitigations
- (b) - Secure updates
  Digital Secure delivery, Integrity & authenticity, Protection against downgrade attacks
- (c) - Protection against supply-chain attacks
  SBOM compliant with ECMA-424, Dependency pinning via uv, Reproducible environments, Visibility into third-party components

NIS2 Violations

Article 21, Cybersecurity risk-management measures

(e) security in network and information systems acquisition, development and maintenance, including vulnerability handling and disclosure;

ϟ Insecure command execution, insecure update mechanism etc.

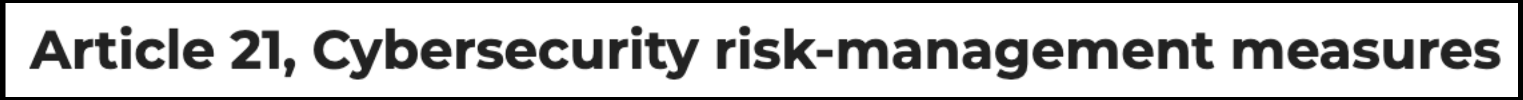
(a) policies on risk analysis and information system security;

(f) policies and procedures to assess the effectiveness of cybersecurity risk-management measures;

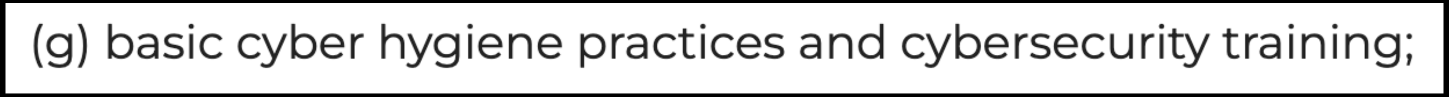
ϟ With proper risk analysis process & policies, basic risks from well-known vulnerability classes (injection, weak crypto, broken auth) should have been detected

[1] https://www.nis-2-directive.com/NIS_2_Directive_Articles.html

NIS2 Violations



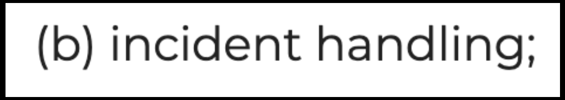
Article 21, Cybersecurity risk-management measures



(g) basic cyber hygiene practices and cybersecurity training;

↯ Hard-coding secrets & using trivial passwords are part of basic cyber hygiene

↯ Existence of those issues suggests that training was not suffucient



(b) incident handling;

→Process for this is very important for the case that vulnerabilites are detected to mitigate incidents‘ impact

[1] https://www.nis-2-directive.com/NIS_2_Directive_Articles.html

NIS2 Violations

Article 20, Governance

→Management body must approve the cybersecurity risk-management measures taken to comply with Article 21, oversee their implementation, and can be held liable for infringements.

Article 23, Reporting obligations

→Management body must approve the cybersecurity risk-management measures taken to comply with Article 21, oversee their implementation, and can be held liable for infringements.

[1] https://www.nis-2-directive.com/NIS_2_Directive_Articles.html

ISO-27001 Violations

Violated Clauses

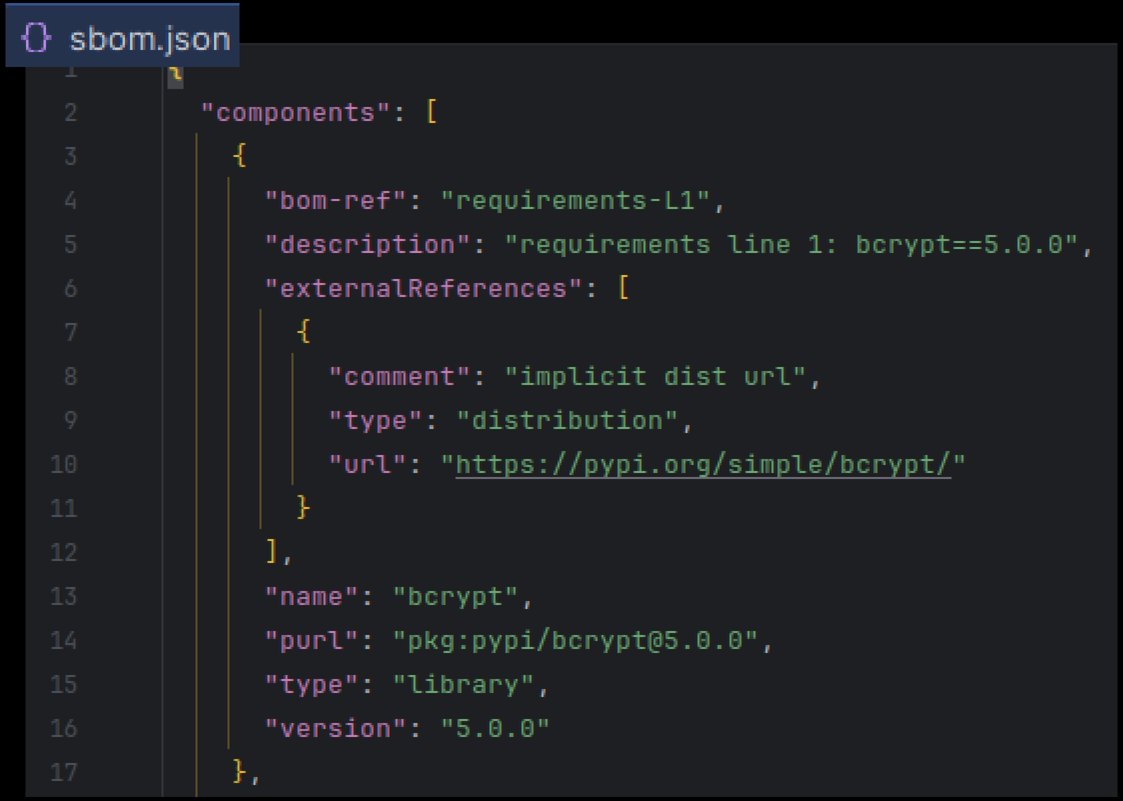
- 6.1.2 - Information security risk assessment
- 6.1.3 - Information security risk treatment
- 7.2 - Competence
- 8.1 - Operational planning and control
- 9.2 - Internal Audits

Violated Annex A controls

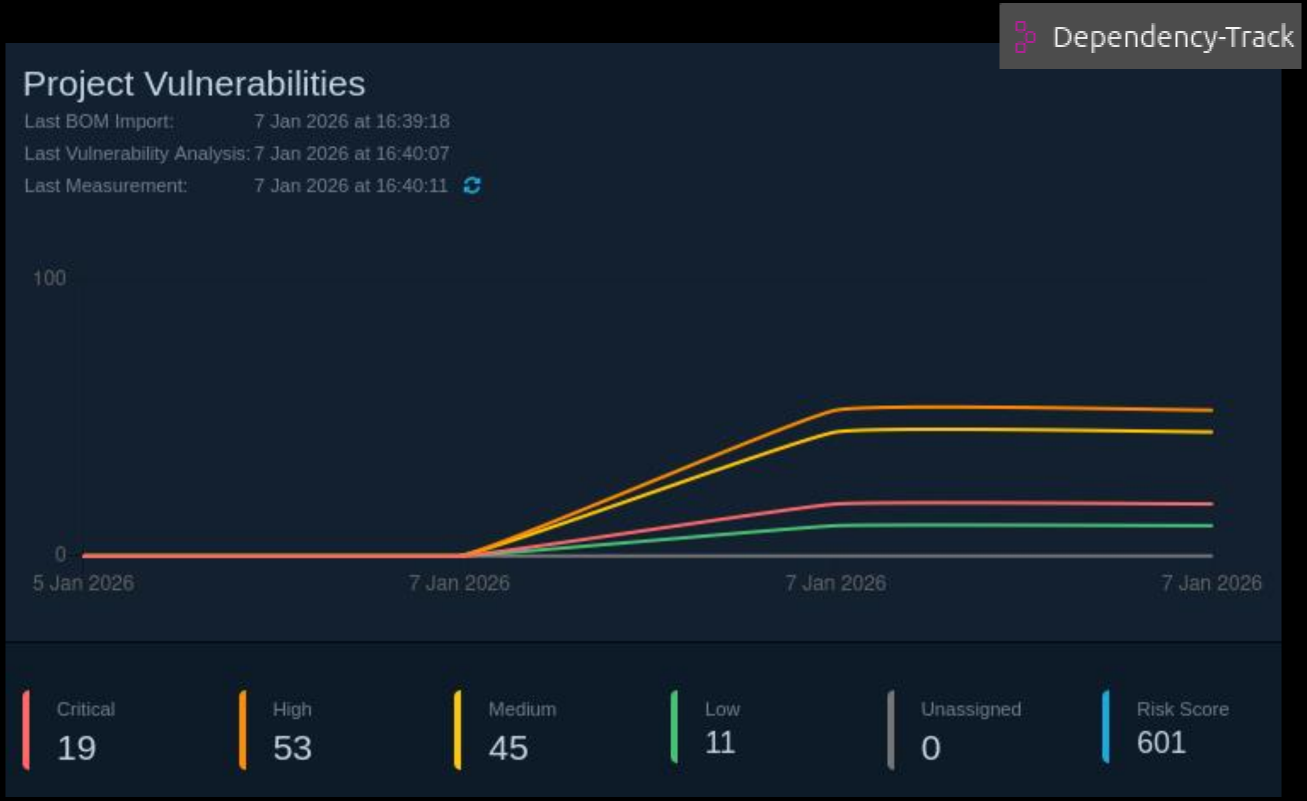
- A.5.17 - Authentication information
- A.8.2 - Privileged access rights
- A.8.3 - Information access restriction
- A.8.7 - Protection against malware
- A.8.9 - Configuration management
- A.8.11 - Data masking
- A.8.15 - Logging
- A.8.16 - Monitoring activities
- A.8.24 - Use of cryptography
- A.8.28 - Secure coding

Additional Tooling

SBOM



```
{
  "components": [
    {
      "bom-ref": "requirements-L1",
      "description": "requirements line 1: bcrypt==5.0.0",
      "externalReferences": [
        {
          "comment": "implicit dist url",
          "type": "distribution",
          "url": "https://pypi.org/simple/bcrypt/"
        }
      ],
      "name": "bcrypt",
      "purl": "pkg:pypi/bcrypt@5.0.0",
      "type": "library",
      "version": "5.0.0"
    },
```



Project Vulnerabilities

Last BOM Import: 7 Jan 2026 at 16:39:18

Last Vulnerability Analysis: 7 Jan 2026 at 16:40:07

Last Measurement: 7 Jan 2026 at 16:40:11

Critical 19

High 53

Medium 45

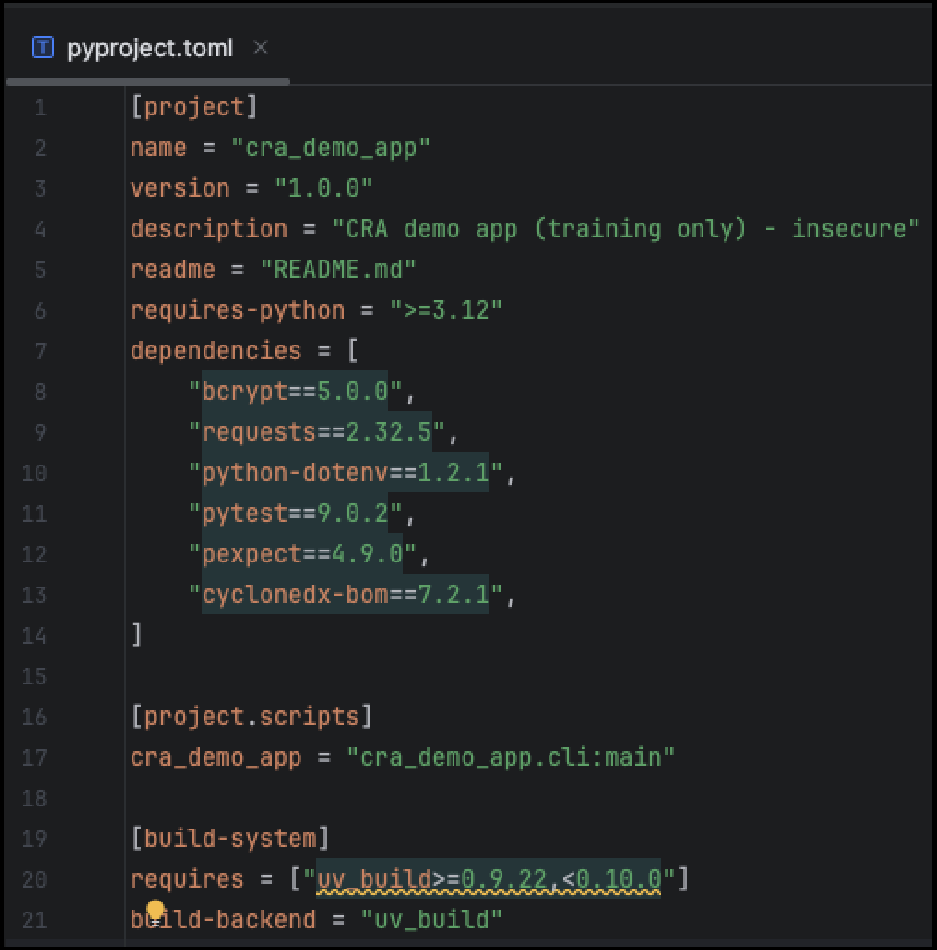
Low 11

Unassigned 0

Risk Score 601

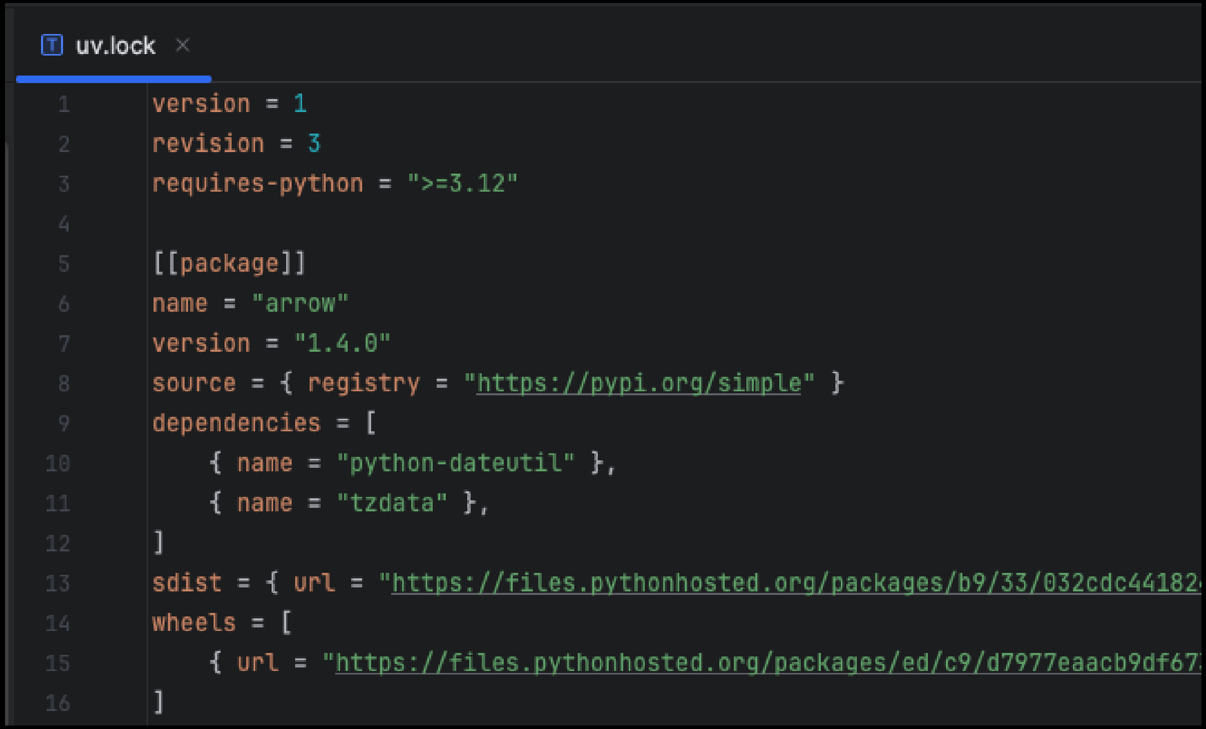
SBOM standard: https://ecma-international.org/publications-and-standards/standards/ecma-424/

Package & Version Manager - UV



pyproject.toml

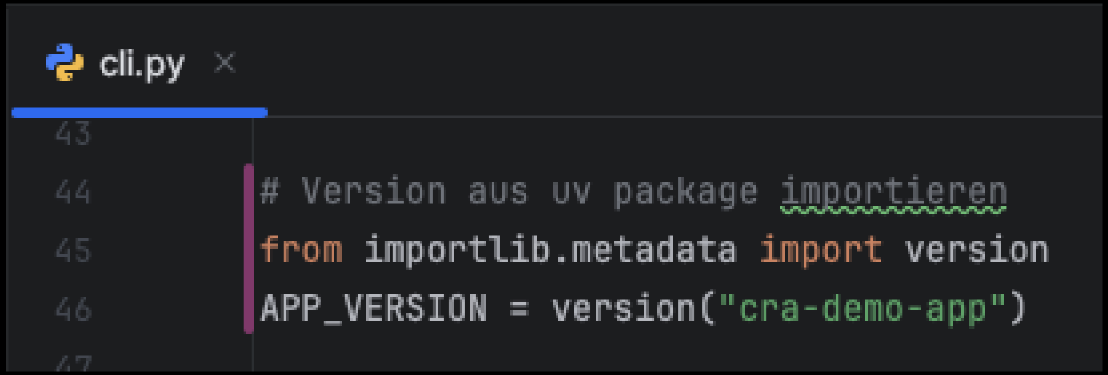
```toml
[project]
name = "cra_demo_app"
version = "1.0.0"
description = "CRA demo app (training only) - insecure"
readme = "README.md"
requires-python = ">=3.12"
dependencies = [
    "bcrypt==5.0.0",
    "requests==2.32.5",
    "python-dotenv==1.2.1",
    "pytest==9.0.2",
    "pexpect==4.9.0",
    "cyclonedx-bom==7.2.1",
]

[project.scripts]
cra_demo_app = "cra_demo_app.cli:main"

[build-system]
requires = ["uv_build>=0.9.22,<0.10.0"]
build-backend = "uv_build"
```



uv.lock

```toml
version = 1
revision = 3
requires-python = ">=3.12"

[[package]]
name = "arrow"
version = "1.4.0"
source = { registry = "https://pypi.org/simple" }
dependencies = [
    { name = "python-dateutil" },
    { name = "tzdata" },
]
sdist = { url = "https://files.pythonhosted.org/packages/b9/33/032cdc44182
wheels = [
    { url = "https://files.pythonhosted.org/packages/ed/c9/d7977eaacb9df67
]
```

Package & Version Manager - UV



```
# Version aus uv package importieren
from importlib.metadata import version
APP_VERSION = version("cra-demo-app")
```

Package & Version Manager - UV

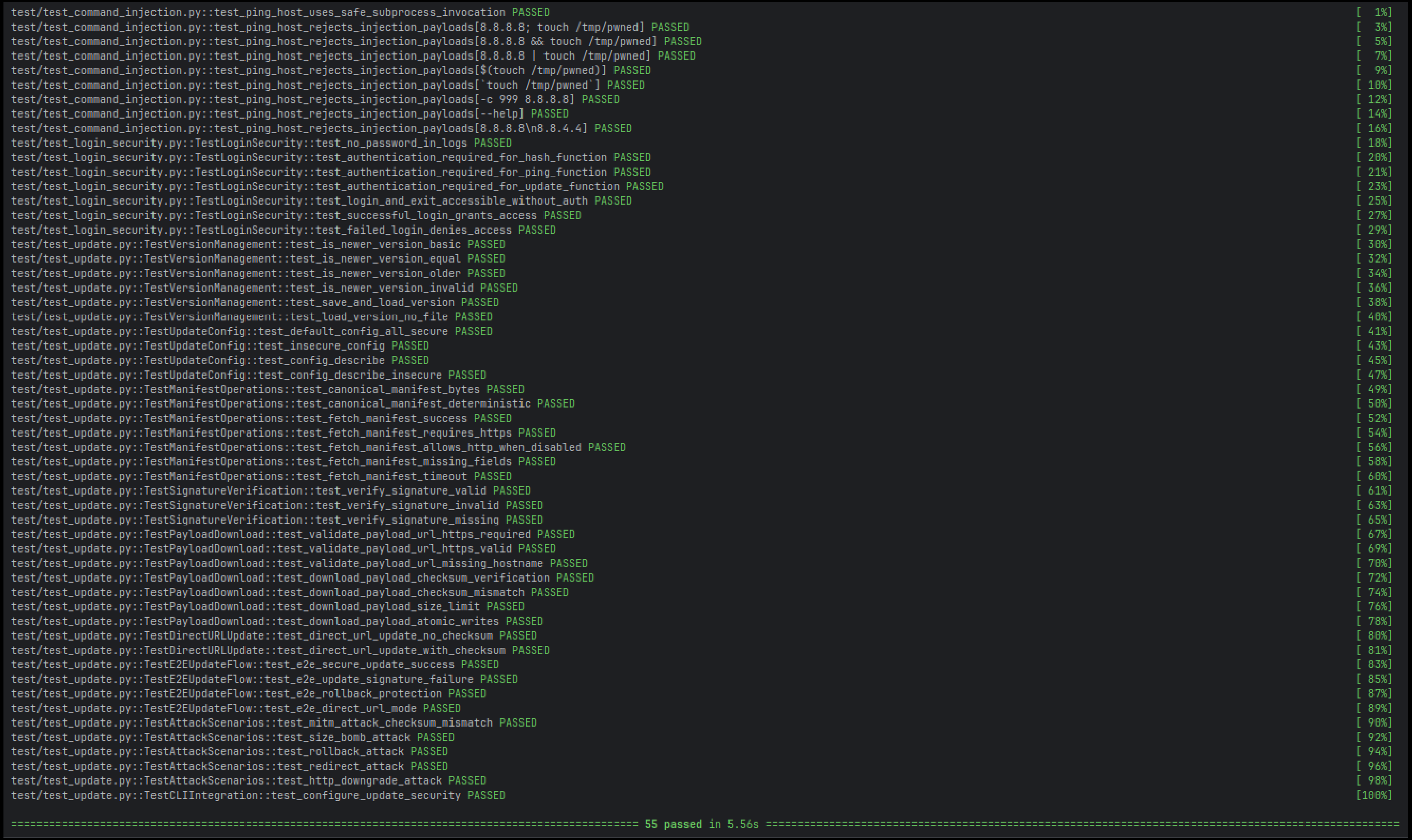
- Reproducible environments and drift prevention
- Enforcement of dependency policy, not just documentation
- Enables SBOM generation

SBOM & Package Manager - Impact

Framework	Requirements (at least partially) fulfilled by SBOM
CRA	• Software composition transparency • Post-release vulnerability identification • Third-party component tracking
NIS2	• Art. 21(2)(d) – Supply chain security • Art. 21(2)(e) – Vulnerability handling • Art. 21(2)(f) – Security testing & monitoring
ISO-27001	• A.5.9 – Inventory of information assets • A.5.23 – Information security for supplier relationships • A.8.8 – Management of technical vulnerabilities

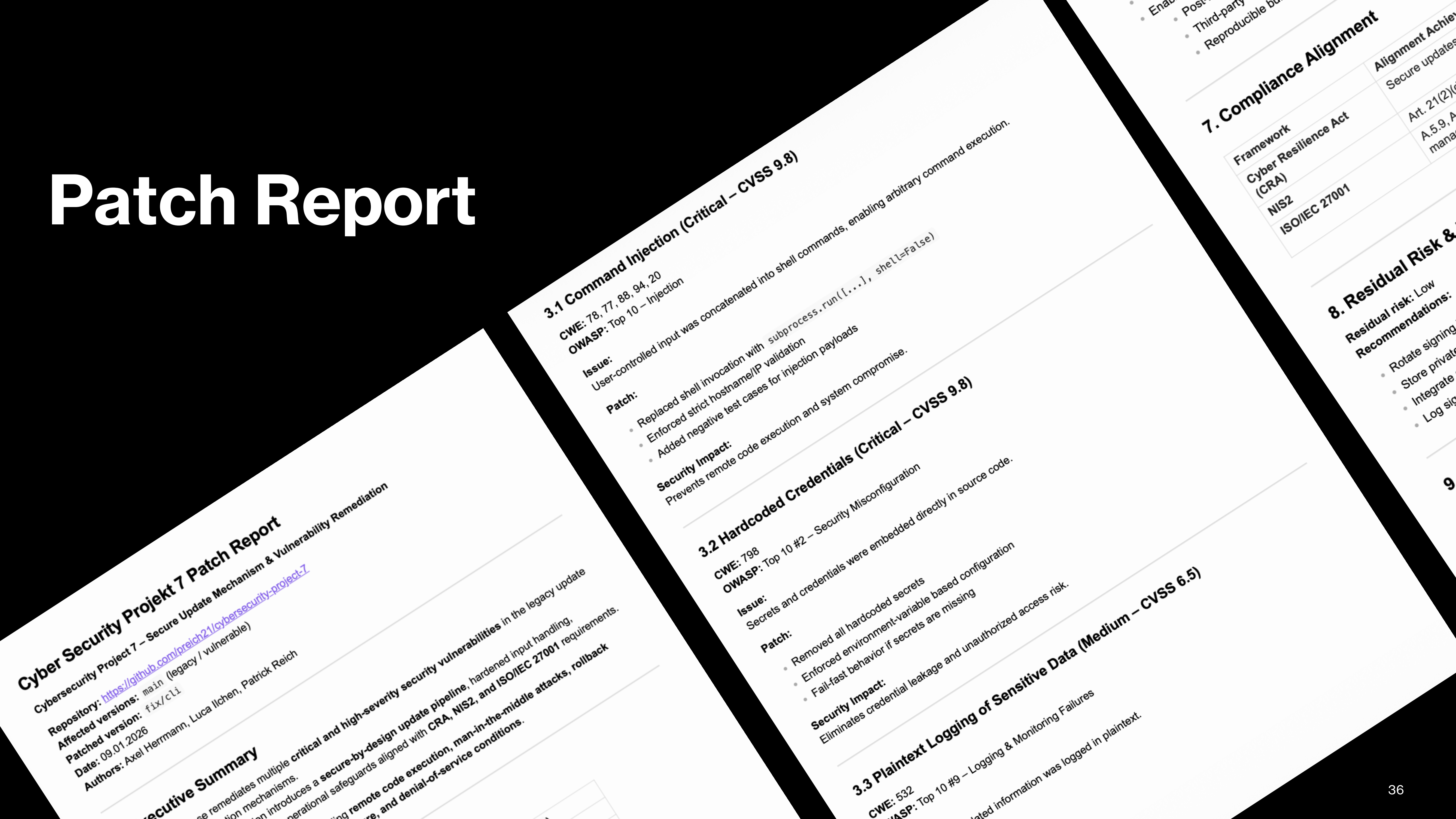
Tests

Unit and Integration tests included
55 tests overall



```
test/test_command_injection.py::test_ping_host_uses_safe_subprocess_invocation PASSED                                      [  1%]
test/test_command_injection.py::test_ping_host_rejects_injection_payloads[8.8.8.8; touch /tmp/pwned] PASSED                [  3%]
test/test_command_injection.py::test_ping_host_rejects_injection_payloads[8.8.8.8 && touch /tmp/pwned] PASSED              [  5%]
test/test_command_injection.py::test_ping_host_rejects_injection_payloads[8.8.8.8 | touch /tmp/pwned] PASSED               [  7%]
test/test_command_injection.py::test_ping_host_rejects_injection_payloads[$(touch /tmp/pwned)] PASSED                      [  9%]
test/test_command_injection.py::test_ping_host_rejects_injection_payloads[`touch /tmp/pwned`] PASSED                       [ 10%]
test/test_command_injection.py::test_ping_host_rejects_injection_payloads[-c 999 8.8.8.8] PASSED                           [ 12%]
test/test_command_injection.py::test_ping_host_rejects_injection_payloads[--help] PASSED                                   [ 14%]
test/test_command_injection.py::test_ping_host_rejects_injection_payloads[8.8.8.8\n8.8.4.4] PASSED                         [ 16%]
test/test_login_security.py::TestLoginSecurity::test_no_password_in_logs PASSED                                            [ 18%]
test/test_login_security.py::TestLoginSecurity::test_authentication_required_for_hash_function PASSED                      [ 20%]
test/test_login_security.py::TestLoginSecurity::test_authentication_required_for_ping_function PASSED                      [ 21%]
test/test_login_security.py::TestLoginSecurity::test_authentication_required_for_update_function PASSED                    [ 23%]
test/test_login_security.py::TestLoginSecurity::test_login_and_exit_accessible_without_auth PASSED                         [ 25%]
test/test_login_security.py::TestLoginSecurity::test_successful_login_grants_access PASSED                                 [ 27%]
test/test_login_security.py::TestLoginSecurity::test_failed_login_denies_access PASSED                                     [ 29%]
test/test_update.py::TestVersionManagement::test_is_newer_version_basic PASSED                                             [ 30%]
test/test_update.py::TestVersionManagement::test_is_newer_version_equal PASSED                                             [ 32%]
test/test_update.py::TestVersionManagement::test_is_newer_version_older PASSED                                             [ 34%]
test/test_update.py::TestVersionManagement::test_is_newer_version_invalid PASSED                                           [ 36%]
test/test_update.py::TestVersionManagement::test_save_and_load_version PASSED                                              [ 38%]
test/test_update.py::TestVersionManagement::test_load_version_no_file PASSED                                               [ 40%]
test/test_update.py::TestUpdateConfig::test_default_config_all_secure PASSED                                               [ 41%]
test/test_update.py::TestUpdateConfig::test_insecure_config PASSED                                                         [ 43%]
test/test_update.py::TestUpdateConfig::test_config_describe PASSED                                                         [ 45%]
test/test_update.py::TestUpdateConfig::test_config_describe_insecure PASSED                                                [ 47%]
test/test_update.py::TestManifestOperations::test_canonical_manifest_bytes PASSED                                          [ 49%]
test/test_update.py::TestManifestOperations::test_canonical_manifest_deterministic PASSED                                  [ 50%]
test/test_update.py::TestManifestOperations::test_fetch_manifest_success PASSED                                            [ 52%]
test/test_update.py::TestManifestOperations::test_fetch_manifest_requires_https PASSED                                     [ 54%]
test/test_update.py::TestManifestOperations::test_fetch_manifest_allows_http_when_disabled PASSED                          [ 56%]
test/test_update.py::TestManifestOperations::test_fetch_manifest_missing_fields PASSED                                     [ 58%]
test/test_update.py::TestManifestOperations::test_fetch_manifest_timeout PASSED                                            [ 60%]
test/test_update.py::TestSignatureVerification::test_verify_signature_valid PASSED                                         [ 61%]
test/test_update.py::TestSignatureVerification::test_verify_signature_invalid PASSED                                       [ 63%]
test/test_update.py::TestSignatureVerification::test_verify_signature_missing PASSED                                       [ 65%]
test/test_update.py::TestPayloadDownload::test_validate_payload_url_https_required PASSED                                  [ 67%]
test/test_update.py::TestPayloadDownload::test_validate_payload_url_https_valid PASSED                                     [ 69%]
test/test_update.py::TestPayloadDownload::test_validate_payload_url_missing_hostname PASSED                                [ 70%]
test/test_update.py::TestPayloadDownload::test_download_payload_checksum_verification PASSED                               [ 72%]
test/test_update.py::TestPayloadDownload::test_download_payload_checksum_mismatch PASSED                                   [ 74%]
test/test_update.py::TestPayloadDownload::test_download_payload_size_limit PASSED                                          [ 76%]
test/test_update.py::TestPayloadDownload::test_download_payload_atomic_writes PASSED                                       [ 78%]
test/test_update.py::TestDirectURLUpdate::test_direct_url_update_no_checksum PASSED                                        [ 80%]
test/test_update.py::TestDirectURLUpdate::test_direct_url_update_with_checksum PASSED                                      [ 81%]
test/test_update.py::TestE2EUpdateFlow::test_e2e_secure_update_success PASSED                                              [ 83%]
test/test_update.py::TestE2EUpdateFlow::test_e2e_update_signature_failure PASSED                                           [ 85%]
test/test_update.py::TestE2EUpdateFlow::test_e2e_rollback_protection PASSED                                                [ 87%]
test/test_update.py::TestE2EUpdateFlow::test_e2e_direct_url_mode PASSED                                                    [ 89%]
test/test_update.py::TestAttackScenarios::test_mitm_attack_checksum_mismatch PASSED                                        [ 90%]
test/test_update.py::TestAttackScenarios::test_size_bomb_attack PASSED                                                     [ 92%]
test/test_update.py::TestAttackScenarios::test_rollback_attack PASSED                                                      [ 94%]
test/test_update.py::TestAttackScenarios::test_redirect_attack PASSED                                                      [ 96%]
test/test_update.py::TestAttackScenarios::test_http_downgrade_attack PASSED                                                [ 98%]
test/test_update.py::TestCLIIntegration::test_configure_update_security PASSED                                             [100%]

===================================================== 55 passed in 5.56s =====================================================
```

Patch Report



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

ecutive Summary

se remediates multiple **critical and high-severity security vulnerabilities** in the legacy update
tion mechanisms.
n introduces a **secure-by-design update pipeline**, hardened input handling,
perational safeguards aligned with **CRA, NIS2, and ISO/IEC 27001** requirements.
ing **remote code execution, man-in-the-middle attacks, rollback**
re, and denial-of-service conditions.

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
ASP: Top 10 #9 – Logging & Monitoring Failures

ated information was logged in plaintext.

- Enab
- Post-
- Third-party
- Reproducible bu

7. Compliance Alignment

Framework	Alignment Achie
Cyber Resilience Act (CRA)	Secure updates
NIS2	Art. 21(2)(
ISO/IEC 27001	A.5.9, A mana

8. Residual Risk &

Residual risk: Low
Recommendations:

- Rotate signing
- Store privat
- Integrate
- Log sig

9

Repository

For reference, you can find the project, including all source code of the insecure application as well as the fixed version, and the SBOM file on GitHub*:

https://github.com/preich21/cybersecurity-project-7

*the main branch contains only the insecure application. For the fixed version, please see the fix/cli branch