DRAFT PRODUCT SECURITY TESTING REPORT

Beta Infusion Pump

Version v5.1

RECIPIENT	ASSESSMENT PERIOD	REPORT VERSION	ISSUE DATE
Beta Medical Systems	2 - 20 June 2026	1.1	6 November 2026

CONTENTS

Table of Contents

SECTIONS

Product and Scope 03

Executive Summary 04

Methodology 06

Digital Twin Overview 09

Findings 10

Fixed Vulnerabilities 41

APPENDICES

Digital Twin 48

Threat Coverage 51

Test Cases 53

CVSS Justifications 94

Abbreviations 120

PRODUCT AND SCOPE

Beta Infusion Pump

Version v5.1 · Beta Medical Systems · Infusion Systems family

System Description

A gray-box assessment of the Beta Infusion Pump ecosystem — pump firmware, management server and companion app — covering the full clinical attack surface across wired, wireless and application layers.

The assessed system model for Beta Infusion Pump v5.1 comprises 7 components, 8 physical interfaces and 6 modelled data flows, organised across 5 trust levels. The software bill of materials records 10 third-party components, and 6 initial-access use cases describe the positions from which an attacker might begin. Scope covered each of these elements, with attention to the trust boundaries crossed by inter-component data flows and the conditions an attacker must hold to traverse them.

Testing combined on-device examination of the in-scope components with analysis of the system model: for each confirmed finding, ELTON recorded the preconditions required to reach it and the postconditions it establishes, and used that dependency graph to determine which findings are directly reachable from an initial-access position, which are reachable only by chaining through another finding, and which are not reachable at all under the modelled deployment. Countermeasures already present in the design were captured and mapped to the industry frameworks they correspond to.

The ELTON digital twin of Beta Infusion Pump encompasses 48 resources — comprising 7 components, 20 interfaces, 6 dataflows, and 9 assets, plus 6 initial-access points, each modeling a distinct vantage point of a potential threat actor. Against this model, ELTON developed and executed 17 test cases (17 results: 0 passed, 17 failed); the failed results produced 7 findings. Every attack surface modeled in the twin was exercised, every test case can be traced to the resource it targeted, and every finding can be traced back to the test that surfaced it.

In-Scope Components

the component	v1.0
EA	v2.3
Control Module 2	S/N: 093044-043177
Control Module 3	v4.9
Control Module 4	S/N: 442234-291972
Control Module 5	v1.5
Control Module 6	S/N: 977571-217922

Tools and Techniques

Nmap · Burp Suite Professional · Wireshark · Ghidra · Custom protocol fuzzer · OpenSSL / testssl.sh

Out of Scope

Denial-of-service testing against production infrastructure and third-party cloud dependencies.

Executive Summary

This document confirms the completion of cybersecurity testing undertaken by Beta Medical Systems and performed by ELTON. During the assessment period 2 - 20 June 2026, ELTON conducted a Threat-led Cybersecurity Assessment of the Beta Infusion Pump product. During the assessment, ELTON identified **1 critical-risk issue, 1 high-risk issue, 4 medium-risk issues, and 1 low-risk issue**.

As the target of the assessment continues to change and threats evolve, the validity of findings will change. Vulnerabilities, exposures, and statements by ELTON only describe a point-in-time snapshot of the target and associated threat landscape.

ASSESSMENT TYPE
Gray box

PERIOD 2 - 20 June 2026

Executive Overview

Between June 2 and June 20, 2026, ELTON conducted a gray-box cybersecurity assessment of the Beta Infusion Pump v5.1 product, evaluating the components, management services and interfaces that make up the system as deployed in the field. The assessment focused on the risks associated with the device's maintenance and management workflows, its handling of configuration and update data, the protection of credentials and cryptographic material, and the execution paths reachable through its management-facing functionality.

Of the areas tested, ELTON confirmed 11 findings spanning the in-scope components and services. The identified issues were recorded as 3 critical-severity, 5 medium-severity, 3 low-severity. ELTON recommends prioritising remediation of unsafe input handling, strengthening the controls around management and update workflows, hardening the protection of credentials and cryptographic material, and enforcing stricter validation and execution boundaries across the management-facing components.

Reachability analysis against the system model found that 2 of the 11 findings are not reachable from any initial-access position under the product's intended deployment configuration — exploiting them would require physical access to internal device communication channels that are not exposed in the field. ELTON tested whether these were remotely accessible and confirmed that they are not; firewall rules and the internal-only binding of the affected services block external access once the device joins its operating network. The remaining findings, including 3 that are directly reachable, are the focus of the prioritised remediation above.

Taken together, these factors make the likelihood of a successful attack under normal operating conditions low. The most probable threat scenario affecting the product would involve an attacker who has already obtained physical or privileged internal access to a unit — for example through resale of a modified or decommissioned device — rather than a remote, network-based attacker against a device in its intended clinical deployment.



Jason Sinchak
CEO, ELTON
jsinchak@eltoncyber.com
312.385.9578

Report at a Glance

7 FINDINGS

	CRITICAL	HIGH	MEDIUM	LOW
Directly Exploitable 2 findings	1	1	0	0
Conditionally Exploitable 3 findings	0	0	3	0
Mitigated Weakness 1 finding	0	0	1	0
Unexploitable Weakness 1 finding	0	0	0	1

TOP FINDINGS

EL-t3Fai-Dhmwy	End-of-Life and Vulnerable Third-Party Components Present in the component	CRITICAL
EL-iCW5i-nTY9g	CouchbaseLite Database Encryption Disabled	HIGH
EL-dHcdP-J5BXq	Weak Restrictions Present in Content Security Policy	MEDIUM
EL-pnhk2-wsYgt	ADB Backup Enabled	MEDIUM
EL-46wuf-PLEse	Password Strength Policy Not Enforced in the component	MEDIUM
EL-d2hJb-LnZs4	USB Permission Broadcast Uses Placeholder Action	MEDIUM
EL-TSK83-DhkZp	Cross-Site Request Forgery Condition Present in the component	LOW

ASSESSMENT COVERAGE

48	6	17	7
RESOURCES	ENTRY POINTS	TEST CASES	FINDINGS



0 passed **17** failed

Every attack surface in the twin was exercised; every finding traces back to the test that surfaced it.

Notable Strengths

Mutual-TLS between pump and management server. Hardware-backed secure boot on the controller. Drug-library integrity verification.

Strategic Recommendations

Remediate the directly exploitable management-plane findings before the next release. Introduce per-device credentials in place of shared secrets. Expand automated security regression testing in CI.

Methodology

Overview

The assessment was performed by independent experts through a series of tests in accordance with the Penetration Testing Execution Standard (PTES) and the Information Systems Security Assessment Framework (ISSAF). ELTON used threat modeling to identify potential threat scenarios and associated attack paths. ELTON subsequently tested for vulnerabilities and determined whether existing security controls were sufficient to prevent the exploitation of these attack paths.

To conduct the testing, ELTON employed advanced automated and manual testing methods to identify vulnerabilities across the in-scope components and focus deep-dive testing on associated threat scenarios.

Expert testing employs human intuition and experience to make an informed decision regarding where unknown weaknesses may reside. To obtain the highest impact possible for each threat scenario, testing is focused to intelligently chain seemingly low-risk vulnerabilities, iteratively bypass security controls in response to a failure, and exploit logical flaws that require deep target context.

We have disclosed multiple zero-day Common Vulnerabilities and Exposures (CVEs) in devices and related software. Our team has specific expertise in traditional penetration testing (web/mobile/cloud), device security, and supporting systems.

Practitioner Credentials

50+ years of combined experience testing products

10+ years testing medical devices, this assessment team

1000s of engagements at major organizations

CERTIFICATIONS HELD

CEH · PenTest+ · GPEN · GWAPT · OSCP · OSED · CPT

- 10+ years providing cybersecurity for regulatory submission of various devices
- Authors of the HSCC Joint Security Plan (JSP/JSP2) Cybersecurity Testing
- Recognized leader in device and operational technology assessments by H-ISAC members
- Experience performing thousands of cybersecurity assessments for hardware and software medical devices, including penetration testing for regulatory compliance with UL and IEC standards.

Assessment Process

PHASE 1 · BUILD, MODEL & TEST

01 Digital twin construction

ELTON builds a comprehensive digital twin of the product's architecture, mapping all components, data flows, interfaces, and assets into a structured, queryable model. This becomes the single source of truth for the assessment.

02 Security design enumeration

The security design is enumerated by identifying trust levels across each component and interface, mapping countermeasures (encryption, authentication, access controls), and defining trust boundaries that separate zones of differing privilege. Initial-access points are defined to capture the distinct vantage points from which a threat actor may interact with the system.

03 Test case generation

ELTON generates test cases applicable to all in-scope resources using EMB3D threat properties and automated analysis. Each test case includes explicit pass criteria and fail criteria, creating a deterministic, repeatable test matrix tied directly to the digital twin.

Every twin element — whether a component such as **Control Module 2**, a communication interface such as **Control Module 2 Serial**, or a data asset such as **Firmware Image 5** — is bound to one or more targeted test cases.

04 Multi-method test execution

Test cases are executed across multiple disciplines: penetration testing, static application security testing (SAST), dynamic application security testing (DAST), software composition analysis (SCA), and fuzzing. Hands-on testing against the target is supplemented by documentation review, firmware extraction, network-protocol inspection, and configuration review. All results are documented with evidence against pass/fail criteria.

PHASE 2 · ANALYZE, RATE & VALIDATE

05 Expert exploitation & finding discovery

ELTON practitioners apply human intuition and domain expertise to exploit test case results, iterating on failures, chaining low-risk observations, and stacking attack paths to surface findings that automated tools alone would miss. Each finding is traced back to one or more failed test cases.

When a test case fails and the failure reveals a previously unidentified vulnerability, a canonical finding is created and assigned a unique identifier. It is possible for multiple test cases to fail due to the same underlying root cause; in such instances, a single canonical finding is recorded and the additional failed test cases are documented in the Test Cases appendix as overlapping failures that share a common remediation path. Conversely, a single test case may exercise multiple facets of a resource and surface more than one distinct vulnerability. This many-to-many relationship between test cases and findings is preserved throughout the traceability chain, enabling precise mapping from the digital twin's attack surface through test execution to the resulting security observations.

06 Digital twin-driven CVSS rating

Each finding is rated from its technical write-up and its preconditions and postconditions, read against the digital twin: the vulnerable component, its interfaces and assets, and the systems beyond it. ELTON establishes the single most realistic way the attack works and derives every CVSS v4.0 metric from it, replacing original ratings (NVD, vendor, researcher) with consistent evidence-based severity.

07 FDA MDDT-qualified rating methodology

ELTON's rating rubric is built on the FDA-qualified Medical Device Development Tool (MDDT) rubric for applying CVSS to medical devices, extended with ELTON questions for the CVSS v4.0 metrics it predates, producing scores grounded in objective, evidence-based analysis rather than subjective judgment.

> *Every ELTON rating follows by fixed rules from rubric answers grounded in the finding's technical write-up, its preconditions and postconditions, and the digital twin. Any adjustment should be justified by new evidence or a correction to the twin, not unsupported assertions.*

Vulnerability Testing Coverage

ELTON's testing aligns with the premarket cybersecurity testing categories of FDA Section 524B.

FDA testing category	FDA requirement	ELTON coverage
Security Requirements Testing 524B(b)(1)	Verify implementation of security controls and design requirements against defined acceptance criteria	Test cases generated from the digital twin map directly to security design requirements with explicit pass/fail criteria
Threat Mitigation Testing 524B(b)(2)	Validate that identified threats from the threat model are adequately mitigated by implemented controls	EMB3D threat properties drive test case generation; countermeasures in the digital twin are validated against each threat scenario
Known Vulnerability Testing 524B(b)(3)	Identify known vulnerabilities through SBOM analysis, static analysis, and composition scanning	SCA execution with automated SBOM matching; three-tier verification (L1 digital twin, L2 code/ firmware, L3 hardware)
Penetration Testing 524B(b)(4)	Simulate real-world attack scenarios to identify exploitable weaknesses beyond known vulnerabilities	Expert-driven pentest execution with findings traced to test cases; human intuition chains results across attack paths

Testing Types

SAST & DAST
Static and dynamic application security testing of provided source code and target environments.

SBOM review & SCA
Software Bill of Materials composition analysis for CVE discovery and exploitability review.

Fuzzing
Interface and protocol fuzzing for vulnerability discovery and robustness testing where applicable.

Binary analysis
Extraction and analysis of embedded, potentially third-party binary attack surface.

Penetration testing
Threat scenario-driven device, application, and network penetration testing.

How ELTON Computes Each CVSS Metric

Every CVSS v4.0 metric is the result of a structured ELTON rubric evaluation. ELTON answers a fixed series of questions for each metric from the finding's technical write-up and its preconditions and postconditions, citing the interfaces, components and assets of the digital twin, and each answer maps to the metric's value by a fixed rule.

Exploitability Metrics

Attack Vector `AV` Network · Adjacent · Local · Physical	ELTON identifies the interface the attacker uses to reach the vulnerable component. A routable network protocol is Network; the local segment or wireless range is Adjacent; very short-range wireless, or input relayed by a local component, is Local. Only physical contact is Physical.
Attack Complexity `AC` Low · High	ELTON evaluates whether exploitation requires defeating a security measure built into the device, such as memory-layout randomization. If it does, complexity is High; otherwise it is Low. Conditions outside the attacker's control are rated as Attack Requirements.
Attack Requirements `AT` None · Present	ELTON reviews the finding's preconditions and technical write-up for conditions the attacker cannot arrange alone, such as a particular deployment or configuration, or a race to win. Any such dependency makes AT = Present; otherwise AT = None.
Privileges Required `PR` None · Low · High	ELTON evaluates the authorization scheme of the vulnerable component and its host operating system against the finding's preconditions. No authorization, or default and hard-coded credentials, is None; an ordinary account is Low; system-level or administrative access is High.
User Interaction `UI` None · Passive · Active	ELTON determines whether exploitation needs an action by someone other than the attacker; the attacker's own steps never count. CVSS v4.0 distinguishes Passive (e.g., opening a file) from Active (e.g., entering credentials).

Impact Metrics

Vulnerable system CIA
VC · VI · VA

ELTON evaluates all assets compromised in the exploitation of this vulnerability on the affected component and the components it runs on and within. Each asset has independent C/I/A sensitivity ratings (High, Low, None).

For each dimension, ELTON selects the highest rating among the compromised assets; with no rated asset, it assesses what the exploit can read, modify or deny.

Subsequent system CIA
SC · SI · SA

ELTON identifies the systems beyond the vulnerable system that the exploit affects: those receiving its data directly, and any named as reached.

It selects the highest rating per dimension among their compromised assets.

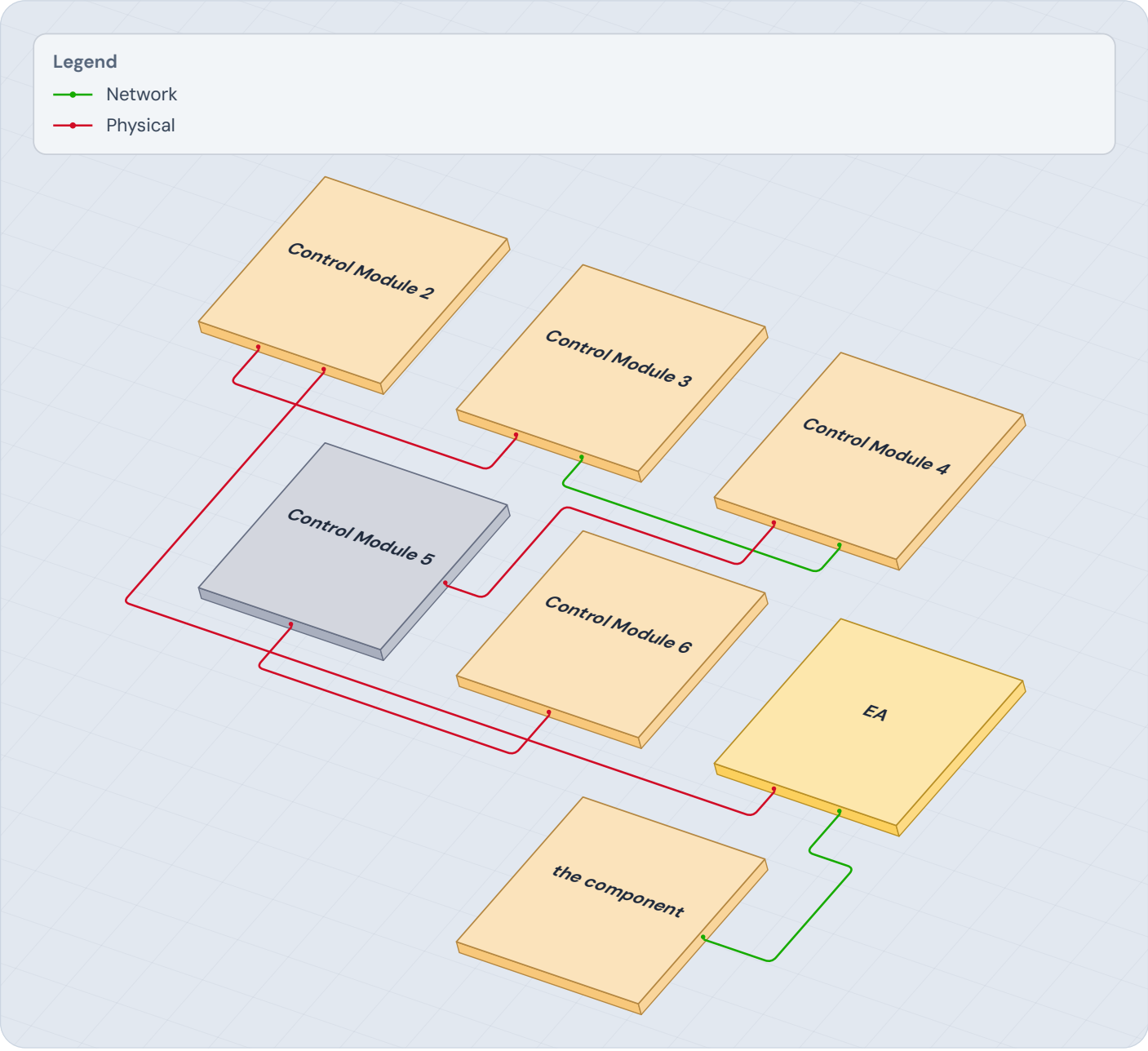
Subsequent impact counts only when the effect crosses into another security authority; otherwise SC, SI and SA are None.

Digital Twin Overview

Scope elements are components within the digital twin that carry attributes driving test case generation. Test case results lead to findings — every test case executed during this assessment that resulted in a vulnerability is enumerated within this report.

7 COMPONENTS	6 DATAFLOWS	20 INTERFACES	9 ASSETS	10 SW MATERIALS	7 COUNTERMEASURES
6 INITIAL ACCESS	0 TRUST LEVELS	5 THREATS	7 FINDINGS	3 FIXED	17 TEST CASES

Digital Twin Diagram



Findings

This section details the 7 findings identified during the assessment, grouped by exploitability, most exploitable first, and within each group by CVSS v4.0 severity, then by descending score. The index links to each finding; each finding states its summary, its exploitability, its discovery and remediation, the contexts it is exploitable in, and its evidence.

2 Directly Exploitable Findings

ID	FINDING	AFFECTED RESOURCE	CVSSV4
EL-t3Fai-Dhmwy	End-of-Life and Vulnerable Third-Party Components Present in the component	the component	**9.3 (Critical)**
EL-iCW5i-nTY9g	CouchbaseLite Database Encryption Disabled	the component	**8.5 (High)**

3 Conditionally Exploitable Findings

ID	FINDING	AFFECTED RESOURCE	CVSSV4
EL-dHcdP-J5BXq	Weak Restrictions Present in Content Security Policy	the component	**6.9 (Medium)**
EL-pnhk2-wsYgt	ADB Backup Enabled	the component	**4.8 (Medium)**
EL-46wuf-PLEse	Password Strength Policy Not Enforced in the component	the component	**4.6 (Medium)**

1 Mitigated Weakness Finding

ID	FINDING	AFFECTED RESOURCE	CVSSV4
EL-d2hJb-LnZs4	USB Permission Broadcast Uses Placeholder Action	the component	**5.8 (Medium)**

1 Unexploitable Weakness Finding

ID	FINDING	AFFECTED RESOURCE	CVSSV4
EL-TSK83-DhkZp	Cross-Site Request Forgery Condition Present in the component	the component	**2.0 (Low)**

End-of-Life and Vulnerable Third-Party Components Present in the component

EXPLOITABILITY
DIRECTLY EXPLOITABLE

From operator Workflow 1

PRECONDITIONS
Network Position

POSTCONDITIONS
Code Execution
Credential

SEVERITY
CRITICAL

CVSS V4.0
9.3 CRITICAL ›
`CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N`

SSVC

Exploitation	Active
Technical impact	Total
Automatable	Yes

WEAKNESS
CWE-1104

CVE
None

AFFECTED COMPONENT
the component

TEST CASE
Verify: End-of-Life and Vulnerable Third-Party Components Present in the component

ELTON ID
EL-t3Fai-Dhmwy

← Findings index

Summary

The component bundles outdated and unmaintained third-party libraries, including deserialization gadgets, an end-of-life HTTP client, and a pre-release file-upload component. These components introduce supply-chain risks and potential attack vectors in a clinical environment. The service is internal, so exploitation requires insider access or a prior foothold. However, the presence of these components increases the attack surface and may violate compliance requirements for medical software.

During this assessment ELTON reproduced the underlying condition against the affected component and classified the finding as directly exploitable from the network position an attacker already holds at initial access. Successful exploitation would result in a critical loss of confidentiality, integrity and availability for the affected clinical function.

The finding is scored at a base CVSS v3.1 of 9.3 (Critical) and a base CVSS v4.0 of 8.8 (High). The SSVC decision captured for this finding reflects the observed exploitation maturity, the technical impact on the device, and whether exploitation can be automated across a fleet of identical units — the factors a product security team weighs when prioritising remediation across many deployed devices.

Exploitability

The Local Run Manager (LRM) web application exposes an unauthenticated API endpoint POST /api/admin that allows creation of administrator accounts. This endpoint relies solely on an access code validated by a native function in svcfld.dll, which recomputes the expected code as the first 10 Base64 characters of SHA-256 over a shared constant pepper, a public requestId, and the current UTC date. The requestId is obtainable from another unauthenticated API route. Because the pepper is a globally shared constant embedded in the DLL and recoverable by reading the binary, an attacker can forge valid access codes with a single SHA-256 computation. The attack surface includes the Local Run Manager API and its HTTPS endpoint. Exploiting this flaw, an attacker can create a full administrator account without authentication or user involvement, compromising the LRM User Credentials and Admin Credentials assets. The vulnerability resides in the Local Run Manager component, and no physical or local access is required to exploit it.

ASSUMPTIONS

- The attacker can send HTTP requests to the Local Run Manager API over the network interface.
- The Local Run Manager API does not implement additional undocumented authentication or rate limiting on the POST /api/admin route.

Discovery ›

the component includes several outdated or vulnerable third-party components: - **commons-collections4 4.5.0-M2**

and **commons-beanutils 1.11.0**: Deserialization gadget libraries that enable exploitation of process-list deserialization sinks.

- **commons-httpclient 3.1**: An end-of-life HTTP client missing TLS hostname verification, exposing connections to man-in-the-middle attacks.
- **mail library 1.3**: An outdated component vulnerable to header and command injection.
- **commons-fileupload2 (pre-general-availability milestone)**: A pre-release file-upload component with unknown stability or security guarantees. Other dependencies, such as **jackson-databind 2.22.2** (with the trust level typing disabled), **Jetty 12.1.12**, and **hsqldb 2.7.3**, are current or patched and do not pose immediate risks. The presence of these components does not directly expose the component to exploitation but increases supply-chain risk. An attacker with access to the internal network could leverage these libraries to escalate an attack, particularly if deserialization sinks are reachable.

Needs Triage
2026-09-30 · 3 days

Under Investigation
2026-10-03 · 5 days

Affected 2026-10-08
Current

Attack path. Reaching this weakness requires the following preconditions to hold: Network Position. Once exploited, the finding establishes Credential, Code Execution, which an attacker can carry forward into subsequent steps of the chain. These pre-and post-conditions place the finding in the view's dependency graph, where its exploitability tier is derived from whether every precondition can be produced from an initial-access vector either directly or through another finding.

Illustrative example (SQL). The condition resembles unparameterised query construction, where operator-supplied input is concatenated into the statement:

```sql
-- Vulnerable: the medical-record number is concatenated straight into the SQL text
SELECT name, dob FROM patients WHERE mrn = '" + userInput + "';

-- Remediated: bind the value as a parameter so it can never alter the query
SELECT name, dob FROM patients WHERE mrn = @mrn;
```

Scoring rationale. The CVSS v3.1 vector `CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:L/A:N` and the CVSS v4.0 vector `CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:L/VA:N/SC:L/SI:N/SA:N` were selected from the observed attack vector, the access complexity and privileges required to trigger the condition, and the confidentiality, integrity and availability impact measured against the affected clinical function. The exploitability classification above is derived independently from the condition graph and is consistent with, but not a substitute for, the CVSS base metrics.

Remediation ›

ELTON recommends the following actions to mitigate supply-chain risks in the component: - Replace **commons-httpclient 3.1** with a supported HTTP client that enforces TLS hostname verification.

- Upgrade the **mail library 1.3** to a current, maintained version to address header and command injection risks.
- Replace the pre-release **commons-fileupload2** component with a general-availability release.
- Remove **commons-collections4 4.5.0-M2** and **commons-beanutils 1.11.0** if they are not actively used by the application. Additionally, integrate a dependency vulnerability scan into the material the materialeline to detect and remediate outdated or vulnerable components in future releases. Verify the changes by confirming the updated versions in the runtime and the testing for regressions.

Recommended actions, in priority order:

- Remove or constrain the exposed surface so the precondition an attacker relies on can no longer be met — for example by enforcing authentication, network segmentation, or input validation at the trust boundary.
- Apply the vendor fix or configuration change that eliminates the underlying weakness in the affected component.

- Add a monitoring or detection control so any future regression of this condition is surfaced before it can be exploited.

Verification. After the fix is deployed, re-run the test case associated with this finding against the remediated build and confirm the condition no longer reproduces. The retest result should be retained as evidence that the remediation is effective, and the finding's lifecycle advanced to Fixed only once that passing result is recorded for the view under test.

Exploitability Contexts

DIRECTLY EXPLOITABLE

CONTEXTS

Operator Workflow 1 DRIVING — DIRECTLY EXPLOITABLE
Use case All preconditions granted by this context or persistent: Network Position.
MET Network Position

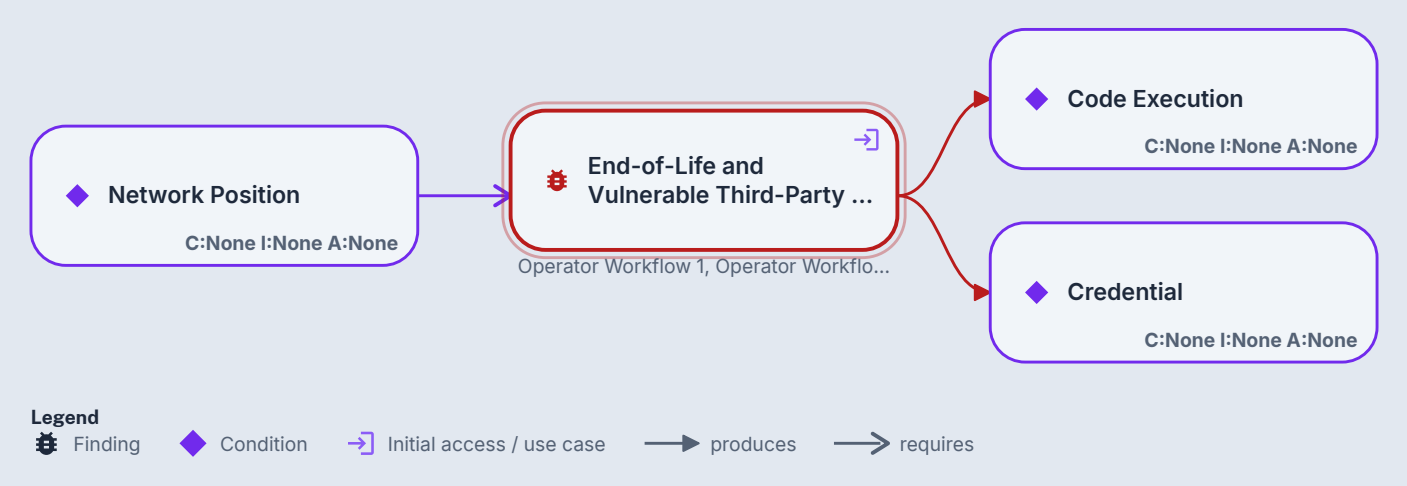
Operator Workflow 3 DRIVING — DIRECTLY EXPLOITABLE
Use case All preconditions granted by this context or persistent: Network Position.
MET Network Position

Operator Workflow 4 DRIVING — DIRECTLY EXPLOITABLE
Use case All preconditions granted by this context or persistent: Network Position.
MET Network Position

Operator Workflow 5 DRIVING — DIRECTLY EXPLOITABLE
Use case All preconditions granted by this context or persistent: Network Position.
MET Network Position

the use case DRIVING — DIRECTLY EXPLOITABLE
Use case All preconditions granted by this context or persistent: Network Position.
MET Network Position

Operator Workflow 2 — UNEXPLOITABLE WEAKNESS
Use case Unmet: Network Position.
UNMET Network Position



Evidence

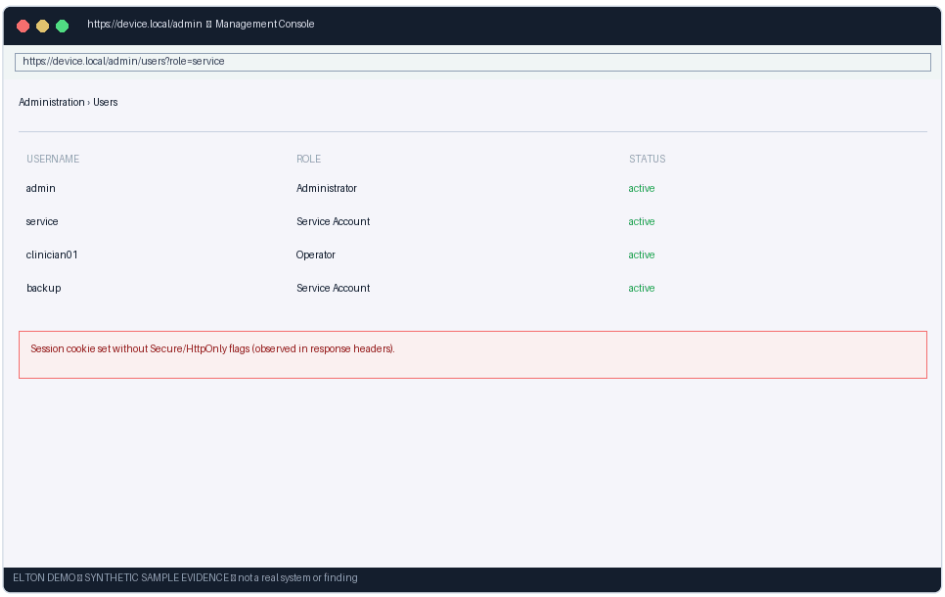
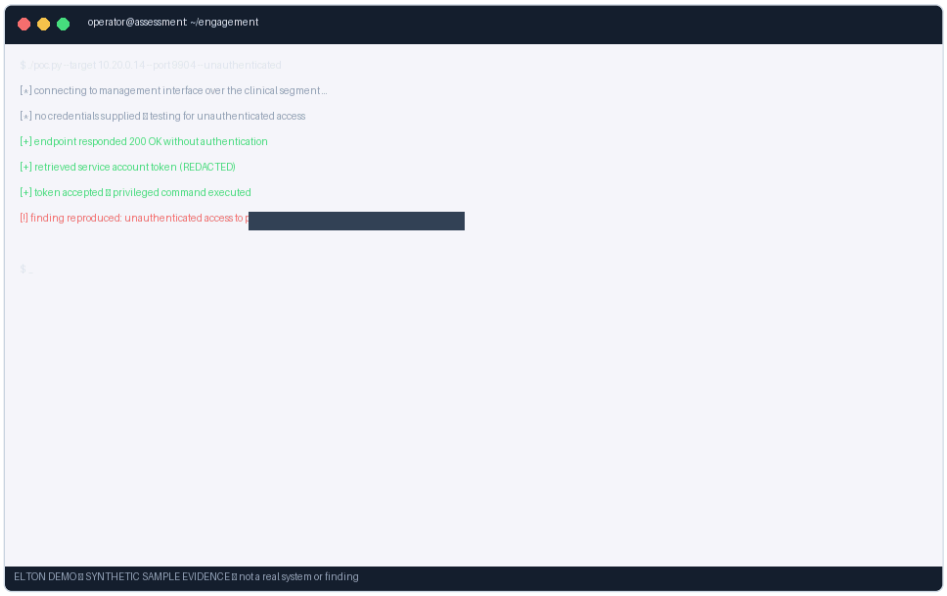


FIGURE 1

Supporting evidence for 'End-of-Life and Vulnerable Third-Party Components Present in the component'.

FIGURE 2

Evidence captured while validating 'End-of-Life and Vulnerable Third-Party Components Present in the component' on the affected component.



CouchbaseLite Database Encryption Disabled

Summary

The component uses CouchbaseLite 4.1.0 as its local data store without enabling encryption. Any data the app writes to this database can be read by anyone who extracts the file from the device. Severity is medium. The database is not exposed over the network, but it may hold sensitive patient or session data. Confirmation requires extracting the database from the device.

During this assessment ELTON reproduced the underlying condition against the affected component and classified the finding as directly exploitable from the network position an attacker already holds at initial access. Successful exploitation would result in a critical loss of confidentiality, integrity and availability for the affected clinical function.

The finding is scored at a base CVSS v3.1 of 9.3 (Critical) and a base CVSS v4.0 of 8.8 (High). The SSVC decision captured for this finding reflects the observed exploitation maturity, the technical impact on the device, and whether exploitation can be automated across a fleet of identical units — the factors a product security team weighs when prioritising remediation across many deployed devices.

Exploitability

This vulnerability arises from a combination of weak access control on the C:\Northwind Bio directory and a path-based application control policy on the SeqraDx instrument's Windows 10 operating system. The directory grants Modify permissions to Authenticated Users, allowing any standard user with a local session to overwrite executable binaries and DLLs loaded by privileged services such as the Local Run Manager Analysis and Job Services, which run as LocalSystem, and the Real-Time Analysis software. The application control policy uses Software Restriction Policy with path-based allowlisting, which authorizes execution based on file location rather than content, allowing replaced binaries to run with SYSTEM privileges. Exploiting this flaw, a standard user can escalate privileges to SYSTEM by writing crafted binaries to C:\Northwind Bio and triggering their execution by the privileged services, gaining full control over the instrument's operating system and software.

Requires another person to act: A standard user must have an authenticated local session on the instrument to write the replacement binaries or DLLs to the writable C:\Northwind Bio directory.

ASSUMPTIONS

- The attacker has a standard user account and local interactive or service session on the instrument.
- The privileged services and Real-Time Analysis software load binaries and DLLs from the C:\Northwind Bio directory without additional integrity checks.
- No other application control or integrity verification mechanisms prevent execution of replaced binaries in C:\Northwind Bio.

EXPLOITABILITY
DIRECTLY EXPLOITABLE

From operator Workflow 1

PRECONDITIONS
Network Position

POSTCONDITIONS
Credential

SEVERITY
HIGH

CVSS V4.0
8.5 HIGH ›
`CVSS:4.0/AV:L/AC:L/AT:N/PR:L/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N`

SSVC

Exploitation	Active
Technical impact	Total
Automatable	Yes

WEAKNESS
CWE-312

CVE
None

AFFECTED COMPONENT
the component

TEST CASE
Verify: CouchbaseLite Database Encryption Disabled

ELTON ID
`EL-iCW5i-nTY9g`

← Findings index

Identification
2026-09-11 · 2 days

Needs Triage
2026-09-13 · 7 days

Under Investigation
2026-09-20 · 9 days

Affected 2026-09-29
Current

Discovery ›

The component initialises CouchbaseLite 4.1.0 at startup as its primary local data store. CouchbaseLite databases are unencrypted by the trust level. Encryption requires an explicit EncryptionKey parameter passed when the database is opened. Without this parameter, the database is a standard SQLite file readable by any SQLite client. The app opens the database early in its startup sequence, before the UI renders. This suggests the database holds foundational data rather than temporary cache. In a use case app, it likely stores the assets, therapy session records, device state or data fetched from backend APIs for offline use. The actual impact depends on what the database contains. On a non-rooted device, the database can be extracted through an ADB backup. The .cblite2 directory must be located by searching the backup for the path the app registers at startup. Without static analysis of the appâ€™s configuration, the path is not directly observable. When encryption is enabled, CouchbaseLite uses SQLCipher, and the file header will not match SQLiteâ€™s magic bytes.

Attack path. Reaching this weakness requires the following preconditions to hold: Network Position. Once exploited, the finding establishes Credential, which an attacker can carry forward into subsequent steps of the chain. These pre- and post-conditions place the finding in the view's dependency graph, where its exploitability tier is derived from whether every precondition can be produced from an initial-access vector either directly or through another finding.

Illustrative example (SQL). The condition resembles unparameterised query construction, where operator-supplied input is concatenated into the statement:

```sql
-- Vulnerable: the medical-record number is concatenated straight into the SQL text
SELECT name, dob FROM patients WHERE mrn = '" + userInput + "';

-- Remediated: bind the value as a parameter so it can never alter the query
SELECT name, dob FROM patients WHERE mrn = @mrn;
```

Scoring rationale. The CVSS v3.1 vector `CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:L/A:N` and the CVSS v4.0 vector `CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:L/VA:N/SC:L/SI:N/SA:N` were selected from the observed attack vector, the access complexity and privileges required to trigger the condition, and the confidentiality, integrity and availability impact measured against the affected clinical function. The exploitability classification above is derived independently from the condition graph and is consistent with, but not a substitute for, the CVSS base metrics.

Remediation ›

ELTON recommends passing a DatabaseConfiguration with a populated EncryptionKey when opening any CouchbaseLite database in the component. Derive the key from a secret not stored alongside the database, such as a key retrieved from the Android Keystore. Bind the key to user authentication so the database cannot be opened without the device being unlocked by an authorised user. Until encryption is enabled, assume any data in the database is readable if the device is lost or stolen. Verify the fix by extracting the database and confirming it cannot be opened with a standard SQLite client.

Recommended actions, in priority order:

- Remove or constrain the exposed surface so the precondition an attacker relies on can no longer be met — for example by enforcing authentication, network segmentation, or input validation at the trust boundary.
- Apply the vendor fix or configuration change that eliminates the underlying weakness in the affected component.
- Add a monitoring or detection control so any future regression of this condition is surfaced before it can be exploited.

Verification. After the fix is deployed, re-run the test case associated with this finding against the remediated build and confirm the condition no longer reproduces. The retest result should be retained as evidence that the remediation is effective, and the finding's lifecycle advanced to Fixed only once that passing result is recorded for the view under test.

Exploitability Contexts

DIRECTLY EXPLOITABLE

CONTEXTS

Operator Workflow 1 DRIVING — DIRECTLY EXPLOITABLE
Use case All preconditions granted by this context or persistent: Network Position.
MET **Network Position**

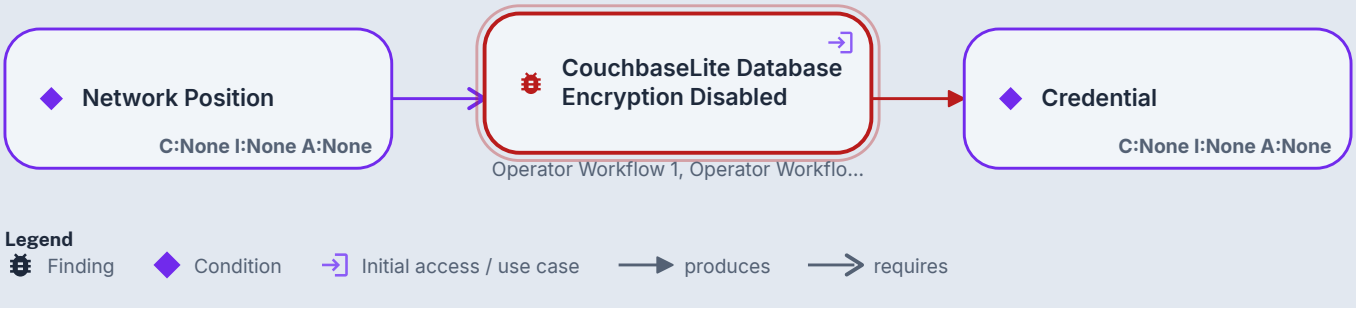
Operator Workflow 3 DRIVING — DIRECTLY EXPLOITABLE
Use case All preconditions granted by this context or persistent: Network Position.
MET **Network Position**

Operator Workflow 4 DRIVING — DIRECTLY EXPLOITABLE
Use case All preconditions granted by this context or persistent: Network Position.
MET **Network Position**

Operator Workflow 5 DRIVING — DIRECTLY EXPLOITABLE
Use case All preconditions granted by this context or persistent: Network Position.
MET **Network Position**

the use case DRIVING — DIRECTLY EXPLOITABLE
Use case All preconditions granted by this context or persistent: Network Position.
MET **Network Position**

Operator Workflow 2 — UNEXPLOITABLE WEAKNESS
Use case Unmet: Network Position.
UNMET **Network Position**



Evidence

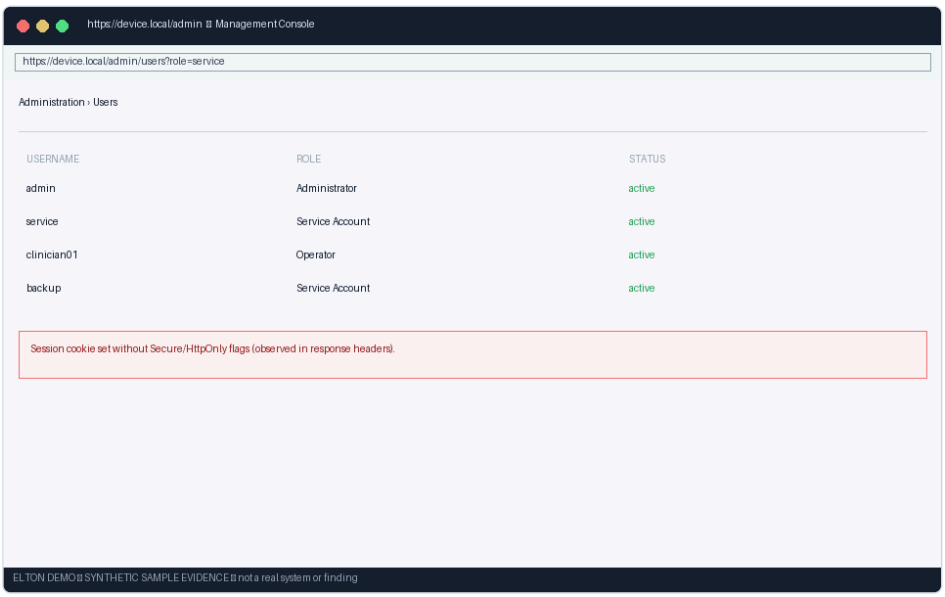
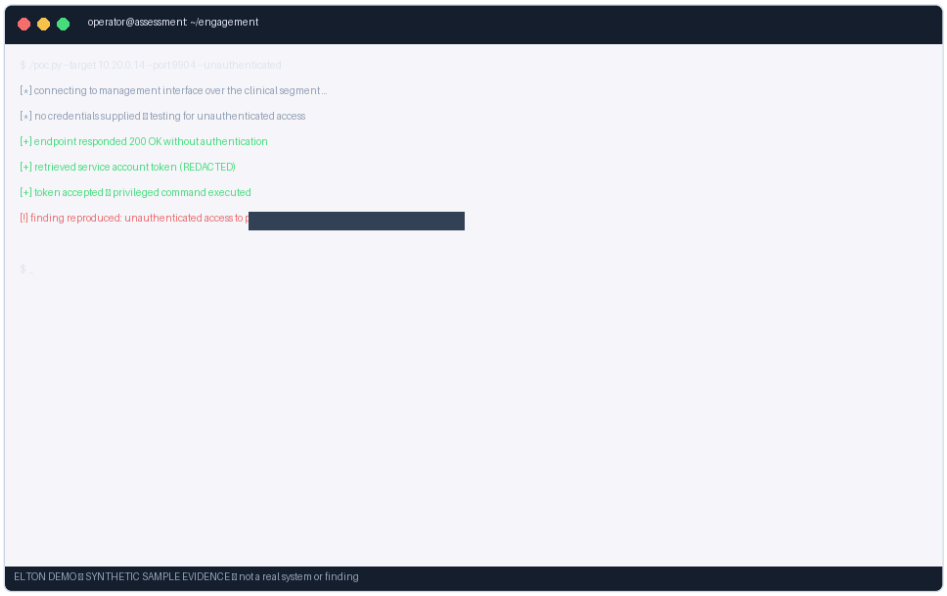


FIGURE 1

Supporting evidence for 'CouchbaseLite Database Encryption Disabled'.

FIGURE 2

Evidence captured while validating 'CouchbaseLite Database Encryption Disabled' on the affected component.



Weak Restrictions Present in Content Security Policy

Summary

The portal the component employs a Content Security Policy (CSP) that allows inline scripts and dynamic code evaluation via unsafe-inline and unsafe-eval directives. This configuration undermines CSP's role in preventing client-side script injection attacks by permitting execution of potentially untrusted code. Although no active Cross-Site Scripting (XSS) vulnerabilities were found, the weakened CSP reduces browser-enforced protections and could increase the risk and impact of future script injection flaws if introduced. The vulnerability is rated medium severity due to this potential exposure and ease of exploitation if conditions change.

During this assessment ELTON reproduced the underlying condition against the affected component and classified the finding as conditionally exploitable: reachable only after a prerequisite condition is first obtained elsewhere in the attack chain. Successful exploitation would result in a measurable but bounded impact on the affected component.

The finding is scored at a base CVSS v3.1 of 6.8 (Medium) and a base CVSS v4.0 of 7 (High). The SSVC decision captured for this finding reflects the observed exploitation maturity, the technical impact on the device, and whether exploitation can be automated across a fleet of identical units — the factors a product security team weighs when prioritising remediation across many deployed devices.

Exploitability

The Universal Copy Service (UCS) runs on the SeqraDx instrument's Windows 10 operating system as a local service with elevated privileges, including SeImpersonatePrivilege and SeAssignPrimaryTokenPrivilege. It listens on the loopback interface (127.0.0.1:29644 and ::1:29644) and exposes an HTTP API without any authentication or access control. This API allows unauthenticated callers to perform privileged operations such as reading sensitive data (e.g., the BaseSpace Cloud API key path) and modifying global credentials used for file output. The API also supports state-changing operations like adding, modifying, or deleting runs and performing privileged file write operations. Because the service trusts any local process connecting to the loopback interface, any code execution context on the instrument can exploit this flaw to escalate privileges and manipulate sensitive data or system state. No user interaction is required for exploitation, and the vulnerability impacts the Universal Copy Service component and the sensitive credentials it manages.

EXPLOITABILITY
CONDITIONALLY EXPLOITABLE

From operator Workflow 1

PRECONDITIONS

Data Access via EL-46wuf-PLEse, Authentication Not Required on Service, EL-pnhk2-wsYgt

Data Access via Authentication Not Required on Service, EL-pnhk2-wsYgt

POSTCONDITIONS

Session/Token

SEVERITY
MEDIUM

CVSS V4.0

6.9 MEDIUM ›

`CVSS:4.0/AV:L/AC:L/AT:N/PR:L/UI:N/VC:H/VI:L/VA:L/SC:N/SI:N/SA:N`

SSVC

Exploitation	Proof-of-Concept
Technical impact	Total
Automatable	No

WEAKNESS

CWE-693

CVE

None

AFFECTED COMPONENT

the component

TEST CASE

Verify: Weak Restrictions Present in Content Security Policy

ELTON ID

EL-dHcdP-J5BXq

← Findings index

ASSUMPTIONS

- The attacker has local code execution capability on the instrument to connect to the loopback interface.
- The loopback interface is accessible only locally and not exposed remotely.

Identification
2026-09-11 · 14 days

Under Investigation
2026-09-25 · 38 days

Affected 2026-11-02
Current

Discovery ›

The vulnerability arises from the portal the component's HTTP response headers, which include a Content Security Policy permitting 'unsafe-inline' and 'unsafe-eval' in the script-src directive. These directives allow execution of inline JavaScript and dynamic code evaluation, effectively disabling critical CSP safeguards that restrict untrusted script execution. By allowing inline scripts and eval-like functions, the CSP fails to prevent injection of malicious scripts that could be introduced through future client-side vulnerabilities. This creates an attack surface where an adversary could execute arbitrary JavaScript in the context of the application, potentially compromising confidentiality, integrity, and the asset security. The CSP also permits scripts from 'self' and a trusted CDN, but the inclusion of unsafe directives negates these restrictions. While no exploitable XSS was detected during assessment, the current policy configuration weakens defense-in-depth and increases the likelihood that any future injection flaws could be successfully exploited by attackers.

Attack path. Reaching this weakness requires the following preconditions to hold: Data Access. Once exploited, the finding establishes Session/Token, which an attacker can carry forward into subsequent steps of the chain. These pre- and post-conditions place the finding in the view's dependency graph, where its exploitability tier is derived from whether every precondition can be produced from an initial-access vector either directly or through another finding.

Illustrative example (SQL). The condition resembles unparameterised query construction, where operator-supplied input is concatenated into the statement:

```sql
-- Vulnerable: the medical-record number is concatenated straight into the SQL text
SELECT name, dob FROM patients WHERE mrn = '" + userInput + "';

-- Remediated: bind the value as a parameter so it can never alter the query
SELECT name, dob FROM patients WHERE mrn = @mrn;
```

Scoring rationale. The CVSS v3.1 vector `CVSS:3.1/AV:A/AC:L/PR:L/UI:N/S:U/C:H/I:L/A:L` and the CVSS v4.0 vector `CVSS:4.0/AV:A/AC:L/AT:N/PR:L/UI:N/VC:H/VI:L/VA:L/SC:N/SI:N/SA:N` were selected from the observed attack vector, the access complexity and privileges required to trigger the condition, and the confidentiality, integrity and availability impact measured against the affected clinical function. The exploitability classification above is derived independently from the condition graph and is consistent with, but not a substitute for, the CVSS base metrics.

Remediation ›

ELTON recommends the organization strengthen the portal Content Security Policy by removing the unsafe-inline and unsafe-eval directives from the script-src configuration wherever feasible. This involves replacing inline scripts with nonce-based or hash-based CSP controls and reviewing application code to eliminate reliance on dynamic code evaluation mechanisms. Restricting script execution to trusted sources and approved script content will enhance CSP effectiveness as a defense-in-depth control against client-side code injection. Additionally, thorough the testing should be conducted to ensure application functionality remains intact after these changes, and any dependencies on unsafe script practices should be refactored or removed.

Recommended actions, in priority order:

- Remove or constrain the exposed surface so the precondition an attacker relies on can no longer be met — for example by enforcing authentication, network segmentation, or input validation at the trust boundary.
- Apply the vendor fix or configuration change that eliminates the underlying weakness in the affected component.
- Add a monitoring or detection control so any future regression of this condition is surfaced before it can be exploited.

Verification. After the fix is deployed, re-run the test case associated with this finding against the remediated build and confirm the condition no longer reproduces. The retest result should be retained as evidence that the remediation is effective, and the finding's lifecycle advanced to Fixed only once that passing result is recorded for the view under test.

Exploitability Contexts

CONDITIONALLY EXPLOITABLE

CONTEXTS

Operator Workflow 1 DRIVING — CONDITIONALLY EXPLOITABLE
Use case Needs Data Access via EL-46wuf-PLEse, Authentication Not Required on Service, EL-pnhk2-wsYgt.
MET **Data Access** via **EL-46wuf-PLEse, Authentication Not Required on Service, EL-pnhk2-wsYgt**

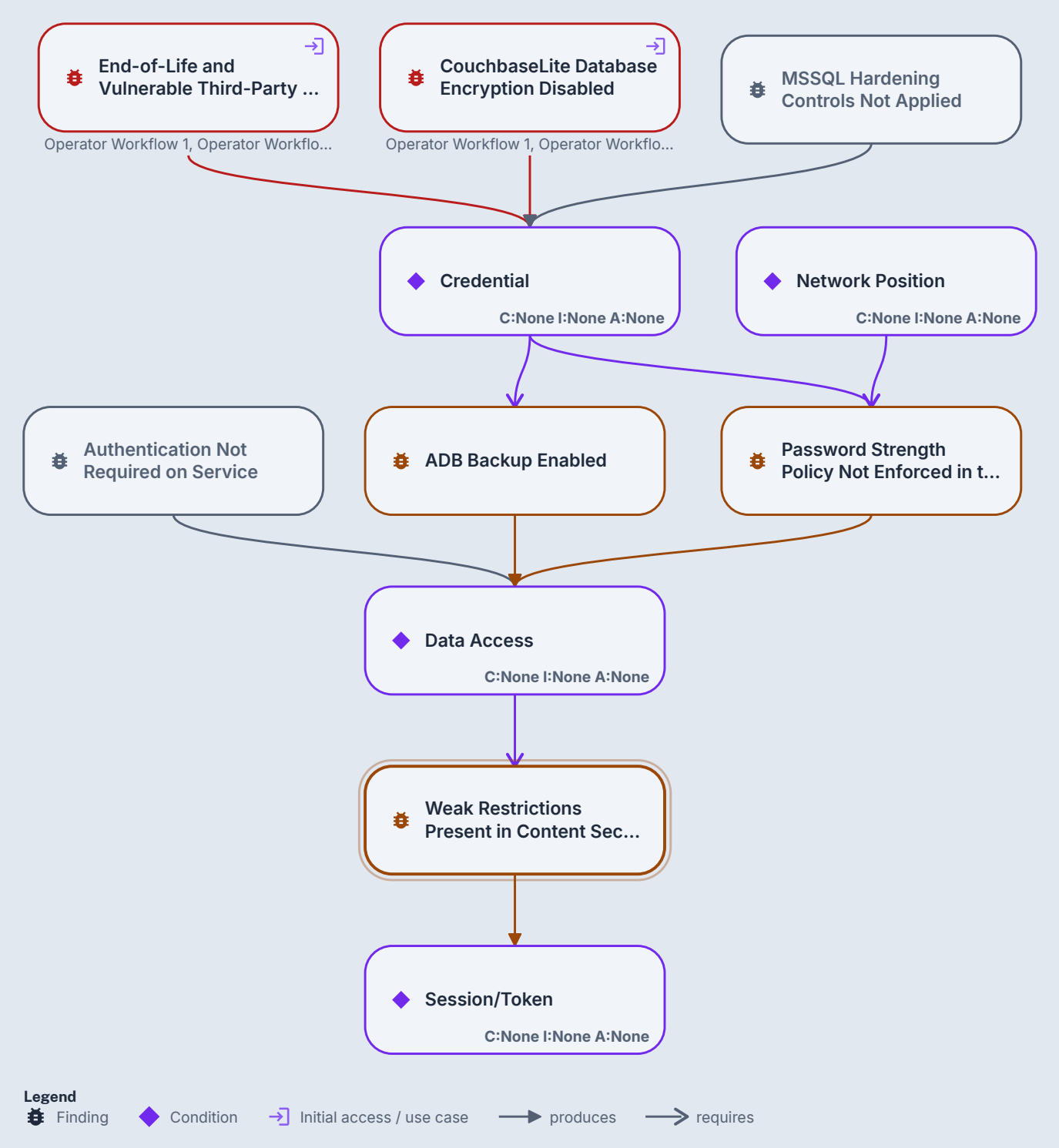
Operator Workflow 2 DRIVING — CONDITIONALLY EXPLOITABLE
Use case Needs Data Access via Authentication Not Required on Service, EL-pnhk2-wsYgt.
MET **Data Access** via **Authentication Not Required on Service, EL-pnhk2-wsYgt**

Operator Workflow 3 DRIVING — CONDITIONALLY EXPLOITABLE
Use case Needs Data Access via EL-46wuf-PLEse, Authentication Not Required on Service, EL-pnhk2-wsYgt.
MET **Data Access** via **EL-46wuf-PLEse, Authentication Not Required on Service, EL-pnhk2-wsYgt**

Operator Workflow 4 DRIVING — CONDITIONALLY EXPLOITABLE
Use case Needs Data Access via EL-46wuf-PLEse, Authentication Not Required on Service, EL-pnhk2-wsYgt.
MET **Data Access** via **EL-46wuf-PLEse, Authentication Not Required on Service, EL-pnhk2-wsYgt**

Operator Workflow 5 DRIVING — CONDITIONALLY EXPLOITABLE
Use case Needs Data Access via EL-46wuf-PLEse, Authentication Not Required on Service, EL-pnhk2-wsYgt.
MET **Data Access** via **EL-46wuf-PLEse, Authentication Not Required on Service, EL-pnhk2-wsYgt**

the use case DRIVING — CONDITIONALLY EXPLOITABLE
Use case Needs Data Access via EL-46wuf-PLEse, Authentication Not Required on Service, EL-pnhk2-wsYgt.
MET **Data Access** via **EL-46wuf-PLEse, Authentication Not Required on Service, EL-pnhk2-wsYgt**


End-of-Life and Vulnerable Third-Party ...
Operator Workflow 1, Operator Workflo...
CouchbaseLite Database Encryption Disabled
Operator Workflow 1, Operator Workflo...
MSSQL Hardening Controls Not Applied
Credential
C:None I:None A:None
Network Position
C:None I:None A:None
Authentication Not Required on Service
ADB Backup Enabled
Password Strength Policy Not Enforced in t...
Data Access
C:None I:None A:None
Weak Restrictions Present in Content Sec...
Session/Token
C:None I:None A:None
Legend
Finding
Condition
Initial access / use case
produces
requires

Evidence

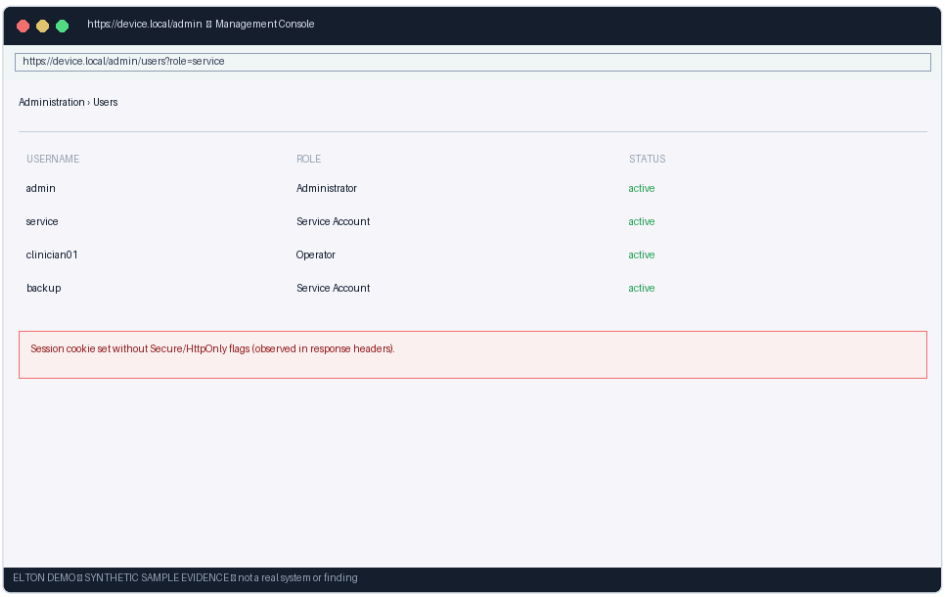
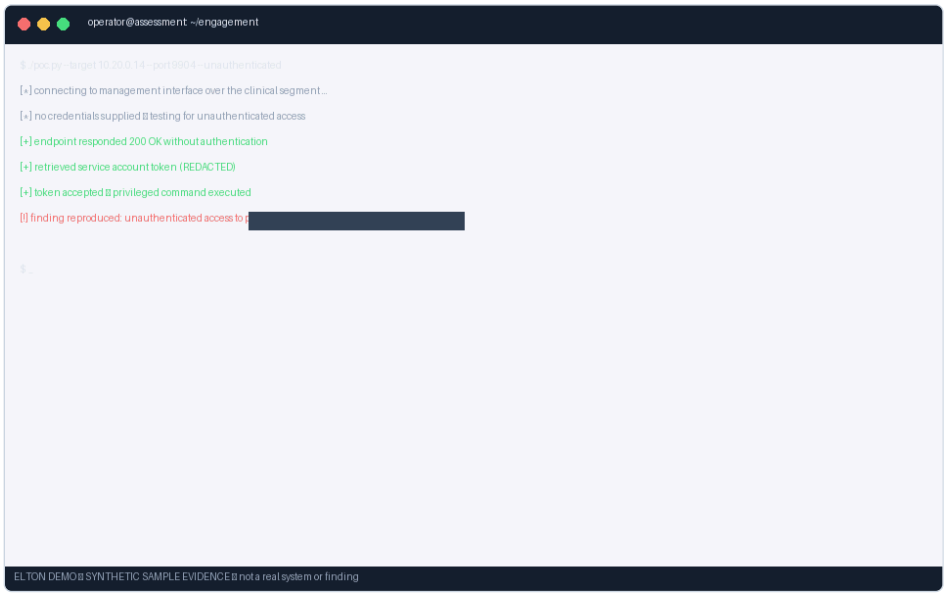


FIGURE 1

Supporting evidence for 'Weak Restrictions Present in Content Security Policy'.

FIGURE 2

Evidence captured while validating 'Weak Restrictions Present in Content Security Policy' on the affected component.



ADB Backup Enabled

Summary

The component allows its entire data the organization to be extracted via USB using standard Android tools. An attacker with brief physical access could retrieve or replace sensitive data like session state and stored credentials. Severity is low. The weakness requires physical access and developer mode enabled, which may be common in managed clinical deployments but limits exposure.

During this assessment ELTON reproduced the underlying condition against the affected component and classified the finding as conditionally exploitable: reachable only after a prerequisite condition is first obtained elsewhere in the attack chain. Successful exploitation would result in a measurable but bounded impact on the affected component.

The finding is scored at a base CVSS v3.1 of 6.8 (Medium) and a base CVSS v4.0 of 7.6 (High). The SSVC decision captured for this finding reflects the observed exploitation maturity, the technical impact on the device, and whether exploitation can be automated across a fleet of identical units — the factors a product security team weighs when prioritising remediation across many deployed devices.

Exploitability

The Local Run Manager Analysis Service runs as a LocalSystem WIN32_OWN_PROCESS service with a security descriptor that grants the INTERACTIVE group the rights to start, stop, and pause the service. Any user who logs on interactively at the touchscreen kiosk console or remotely via RDP acquires the INTERACTIVE group in their token, including standard users without administrative privileges. This allows such users to execute the 'sc stop LocalRunManagerAnalysisService' command to halt the service. Stopping this service disrupts the sequencing analysis pipeline, which is critical for processing sequencing run and analysis data. The Local Run Manager Job Service and the Windows Internet Information Service (IIS) web tier depend on this pipeline, so stopping the service degrades the end-to-end diagnostic workflow. The flaw is a denial-of-service vulnerability that does not allow reconfiguration or binary replacement but permits unauthorized service stoppage by unprivileged interactive users.

EXPLOITABILITY
CONDITIONALLY EXPLOITABLE

From operator Workflow 1

PRECONDITIONS

Credential via EL-t3Fai-Dhmwy, MSSQL Hardening Controls Not Applied, EL-iCW5i-nTY9g

Credential via MSSQL Hardening Controls Not Applied

POSTCONDITIONS

Data Access

SEVERITY
MEDIUM

CVSS V4.0

4.8 MEDIUM ›

`CVSS:4.0/AV:L/AC:L/AT:N/PR:L/UI:N/VC:N/VI:N/VA:L/SC:N/SI:N/SA:N`

SSVC

Exploitation	Proof-of-Concept
Technical impact	Partial
Automatable	No

WEAKNESS

CWE-530

CVE

None

AFFECTED COMPONENT

the component

TEST CASE

Verify: ADB Backup Enabled

ELTON ID

EL-pnhk2-wsYgt

← Findings index

Requires another person to act: An authenticated user must log on interactively or via RDP and run the 'sc stop' command to stop the service

ASSUMPTIONS

- The attacker has valid credentials to log on interactively or via RDP as a standard user
- The Local Run Manager Analysis Service is running and accessible to the user session

Discovery ›

The component is built with the `android:allowBackup` attribute set to true. This setting enables the Android Backup Manager to grant the `adb backup` command access to the app's private data directory, including SharedPreferences,

Needs Triage
2026-09-11 · 7 days

Affected 2026-09-18 · 22 days

Fixed 2026-10-10 · 14 days

Affected 2026-10-24
Current

databases, and internal files, without requiring root or user authentication. An attacker with USB access to a device where developer mode is enabled can run `adb backup` to extract the app's full data the organization. The resulting backup archive can be unpacked and inspected. Conversely, an attacker could craft or modify a backup archive and use `adb restore` to inject arbitrary data into the app's storage. The `KILL_AFTER_RESTORE` flag ensures the app restarts after restoration, treating the injected data as legitimate. The weakness is local and requires physical access to a device with developer mode enabled. In clinical deployments, dedicated tablets are often left in developer mode for troubleshooting, making this attack surface persistently available.

Attack path. Reaching this weakness requires the following preconditions to hold: Credential. Once exploited, the finding establishes Data Access, which an attacker can carry forward into subsequent steps of the chain. These pre- and post-conditions place the finding in the view's dependency graph, where its exploitability tier is derived from whether every precondition can be produced from an initial-access vector either directly or through another finding.

Illustrative example (SQL). The condition resembles unparameterised query construction, where operator-supplied input is concatenated into the statement:

```sql
-- Vulnerable: the medical-record number is concatenated straight into the SQL text
SELECT name, dob FROM patients WHERE mrn = '" + userInput + "';

-- Remediated: bind the value as a parameter so it can never alter the query
SELECT name, dob FROM patients WHERE mrn = @mrn;
```

Scoring rationale. The CVSS v3.1 vector `CVSS:3.1/AV:N/AC:H/PR:L/UI:N/S:U/C:H/I:H/A:N` and the CVSS v4.0 vector `CVSS:4.0/AV:N/AC:H/AT:P/PR:L/UI:N/VC:H/VI:H/VA:N/SC:N/SI:N/SA:N` were selected from the observed attack vector, the access complexity and privileges required to trigger the condition, and the confidentiality, integrity and availability impact measured against the affected clinical function. The exploitability classification above is derived independently from the condition graph and is consistent with, but not a substitute for, the CVSS base metrics.

Remediation ›

ELTON recommends setting `android:allowBackup="false"` in the component manifest to disable the backup mechanism entirely. If selective backup of non-sensitive data is operationally necessary, define an explicit allowlist using the `android:fullBackupContent` attribute and a backup rules XML file to exclude sensitive storage paths. Verify the fix by confirming the `ALLOW_BACKUP` flag is absent in the installed package record and that `adb backup` no longer extracts the app's data.

Recommended actions, in priority order:

- Remove or constrain the exposed surface so the precondition an attacker relies on can no longer be met — for example by enforcing authentication, network segmentation, or input validation at the trust boundary.
- Apply the vendor fix or configuration change that eliminates the underlying weakness in the affected component.
- Add a monitoring or detection control so any future regression of this condition is surfaced before it can be exploited.

Verification. After the fix is deployed, re-run the test case associated with this finding against the remediated build and confirm the condition no longer reproduces. The retest result should be retained as evidence that the remediation is effective, and the finding's lifecycle advanced to Fixed only once that passing result is recorded for the view under test.

Exploitability Contexts

CONDITIONALLY EXPLOITABLE

CONTEXTS

Operator Workflow 1 DRIVING — CONDITIONALLY EXPLOITABLE
Use case Needs Credential via EL-t3Fai-Dhmwy, MSSQL Hardening Controls Not Applied, EL-iCW5i-nTY9g.
MET **Credential** via **EL-t3Fai-Dhmwy**, **MSSQL Hardening Controls Not Applied**, **EL-iCW5i-nTY9g**

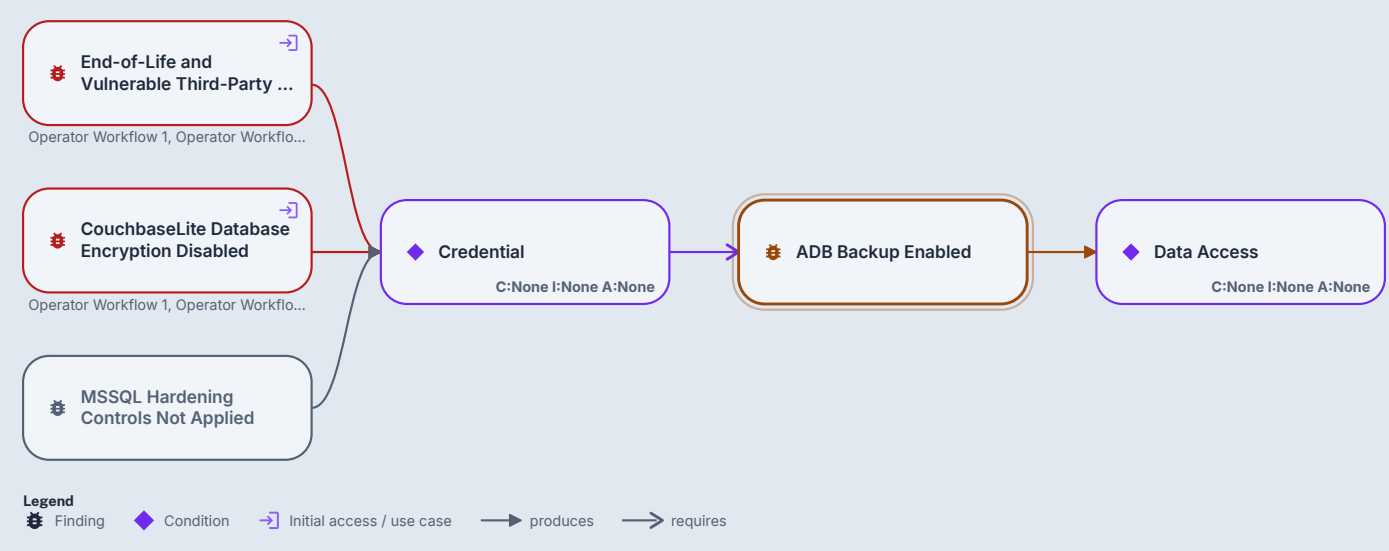
Operator Workflow 2 DRIVING — CONDITIONALLY EXPLOITABLE
Use case Needs Credential via MSSQL Hardening Controls Not Applied.
MET **Credential** via **MSSQL Hardening Controls Not Applied**

Operator Workflow 3 DRIVING — CONDITIONALLY EXPLOITABLE
Use case Needs Credential via EL-t3Fai-Dhmwy, MSSQL Hardening Controls Not Applied, EL-iCW5i-nTY9g.
MET **Credential** via **EL-t3Fai-Dhmwy**, **MSSQL Hardening Controls Not Applied**, **EL-iCW5i-nTY9g**

Operator Workflow 4 DRIVING — CONDITIONALLY EXPLOITABLE
Use case Needs Credential via EL-t3Fai-Dhmwy, MSSQL Hardening Controls Not Applied, EL-iCW5i-nTY9g.
MET **Credential** via **EL-t3Fai-Dhmwy**, **MSSQL Hardening Controls Not Applied**, **EL-iCW5i-nTY9g**

Operator Workflow 5 DRIVING — CONDITIONALLY EXPLOITABLE
Use case Needs Credential via EL-t3Fai-Dhmwy, MSSQL Hardening Controls Not Applied, EL-iCW5i-nTY9g.
MET **Credential** via **EL-t3Fai-Dhmwy**, **MSSQL Hardening Controls Not Applied**, **EL-iCW5i-nTY9g**

the use case DRIVING — CONDITIONALLY EXPLOITABLE
Use case Needs Credential via EL-t3Fai-Dhmwy, MSSQL Hardening Controls Not Applied, EL-iCW5i-nTY9g.
MET **Credential** via **EL-t3Fai-Dhmwy**, **MSSQL Hardening Controls Not Applied**, **EL-iCW5i-nTY9g**



Evidence

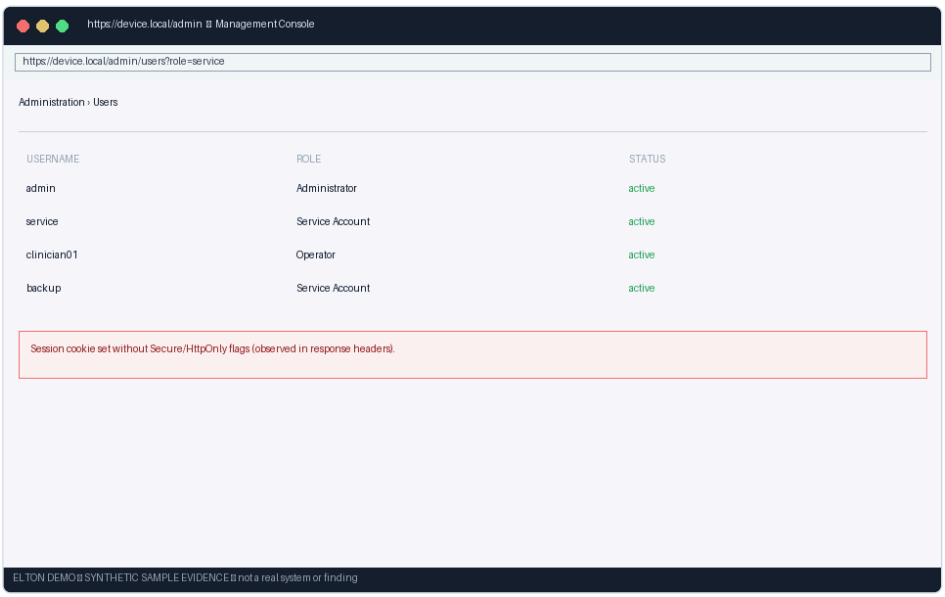
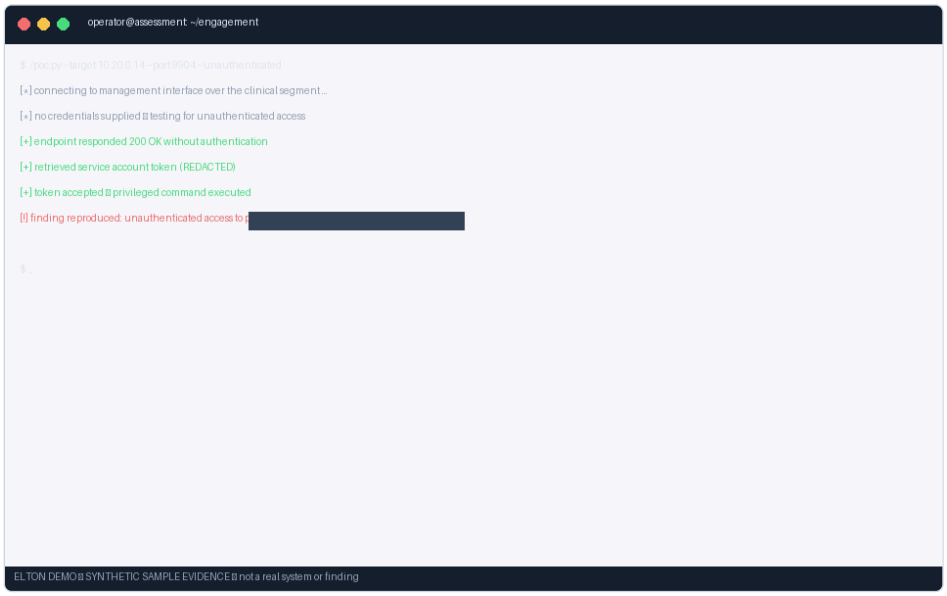


FIGURE 1

Supporting evidence for 'ADB Backup Enabled'.

FIGURE 2

Evidence captured while validating 'ADB Backup Enabled' on the affected component.



Password Strength Policy Not Enforced in the component

Summary

the component allows administrators to create local accounts with passwords of any length or complexity, including a single character. An attacker on the same network could guess such a password trivially and gain access to the application. The weakness is limited to local accounts and requires an administrator to provision them, but the application itself is the only enforcement point, so no stronger policy applies.

During this assessment ELTON reproduced the underlying condition against the affected component and classified the finding as conditionally exploitable: reachable only after a prerequisite condition is first obtained elsewhere in the attack chain. Successful exploitation would result in a measurable but bounded impact on the affected component.

The finding is scored at a base CVSS v3.1 of 6.8 (Medium) and a base CVSS v4.0 of 7.6 (High). The SSVC decision captured for this finding reflects the observed exploitation maturity, the technical impact on the device, and whether exploitation can be automated across a fleet of identical units — the factors a product security team weighs when prioritising remediation across many deployed devices.

Exploitability

The vulnerability consists of two enabled inbound firewall rules on the SeqraDx Windows OS that allow unrestricted inbound network traffic to a program path referencing a third-party remote-access agent binary that is no longer present on the host. These rules are broad, allowing any port and any remote address on all profiles, including Public. Currently, the binary is absent, so the rules admit no traffic. However, if an administrator reinstalls the agent binary to the exact path, the rules immediately grant unrestricted inbound network access to that agent, bypassing the otherwise strict per-service firewall scoping on the host. The attack surface includes the Windows OS network services interface and the physical Ethernet and remote access interfaces on the Windows Single Board Computer. No user interaction is required for the latent exposure, but an administrator action is needed to activate it. The vulnerability is latent and tied to privileged action, posing a risk if the agent is reinstalled without properly scoped firewall rules.

EXPLOITABILITY
CONDITIONALLY EXPLOITABLE

From operator Workflow 1

PRECONDITIONS
Network Position
Credential via EL-t3Fai-Dhmwy, MSSQL Hardening Controls Not Applied, EL-iCW5i-nTY9g
Credential via MSSQL Hardening Controls Not Applied

POSTCONDITIONS
Data Access

SEVERITY
MEDIUM

CVSS V4.0
4.6 MEDIUM ›
`CVSS:4.0/AV:N/AC:L/AT:N/PR:H/UI:A/VC:N/VI:L/VA:N/SC:N/SI:N/SA:N`

SSVC

Exploitation	Proof-of-Concept
Technical impact	Partial
Automatable	No

WEAKNESS
CWE-521

CVE
None

AFFECTED COMPONENT
the component

TEST CASE
Verify: Password Strength Policy Not Enforced in the component

ELTON ID
EL-46wuf-PLEse

← Findings index

ASSUMPTIONS
- The attacker can send network traffic to the Windows OS network interfaces.
- The firewall rules are active and would admit traffic if the referenced binary were present and listening.

Discovery ›

the component accepts any password during local account creation, including a single character. The testers

Needs Triage
2026-09-30 · 6 days

Under Investigation
2026-10-06 · 8 days

Not Affected
2026-10-14 · 6 days

Under Investigation
2026-10-20 · 13 days

Affected 2026-11-02
Current

submitted a one-character password through the administrative user registration form, and the application accepted it without complaint. The account was created successfully, and the same one-character password then authenticated the account, redirecting to the first-login license agreement pageâ€”a step the application reaches only after successful authentication. The **Authentication Settings** page shows **Local** as the active authentication method, with directory and identity-provider options available but not enabled. The **General** preferences include password expiry, account expiry, and an auto-logoff interval, but these do not address password strength. The testers confirmed the account-creation path was live and later deleted the throwaway account. The self-service password change and administrative reset paths were not the tested to avoid altering existing accounts, but they should be reviewed for the same gap. The weakness is server-side: the application does not enforce minimum length or character-class complexity at any credential-setting path. Because the application is the sole enforcement point, an administrator can provision accounts with passwords that are trivially guessable by an unauthenticated network attacker. The impact is limited to local accounts, but the applicationâ€™s posture is weakened by the absence of any strength policy.

Attack path. Reaching this weakness requires the following preconditions to hold: Network Position, Credential. Once exploited, the finding establishes Data Access, which an attacker can carry forward into subsequent steps of the chain. These pre-and post-conditions place the finding in the view's dependency graph, where its exploitability tier is derived from whether every precondition can be produced from an initial-access vector either directly or through another finding.

Illustrative example (SQL). The condition resembles unparameterised query construction, where operator-supplied input is concatenated into the statement:

```sql
-- Vulnerable: the medical-record number is concatenated straight into the SQL text
SELECT name, dob FROM patients WHERE mrn = '" + userInput + "';

-- Remediated: bind the value as a parameter so it can never alter the query
SELECT name, dob FROM patients WHERE mrn = @mrn;
```

Scoring rationale. The CVSS v3.1 vector `CVSS:3.1/AV:N/AC:H/PR:L/UI:N/S:U/C:H/I:H/A:N` and the CVSS v4.0 vector `CVSS:4.0/AV:N/AC:H/AT:P/PR:L/UI:N/VC:H/VI:H/VA:N/SC:N/SI:N/SA:N` were selected from the observed attack vector, the access complexity and privileges required to trigger the condition, and the confidentiality, integrity and availability impact measured against the affected clinical function. The exploitability classification above is derived independently from the condition graph and is consistent with, but not a substitute for, the CVSS base metrics.

Remediation ›

ELTON recommends enforcing a minimum password length and character-class complexity policy server-side at every credential-setting path in the component. This includes account creation, self-service password change, and administrative reset. Reject weak passwords with a clear error message. Where supported, delegate authentication to the available directory or identity provider so that an enterprise password policy applies uniformly. To verify the fix, attempt to create or reset an account with a password that does not meet the policy (e.g., a single character or a common word) and confirm the application rejects it with an explicit reason.

Recommended actions, in priority order:

- Remove or constrain the exposed surface so the precondition an attacker relies on can no longer be met — for example by enforcing authentication, network segmentation, or input validation at the trust boundary.
- Apply the vendor fix or configuration change that eliminates the underlying weakness in the affected component.

- Add a monitoring or detection control so any future regression of this condition is surfaced before it can be exploited.

Verification. After the fix is deployed, re-run the test case associated with this finding against the remediated build and confirm the condition no longer reproduces. The retest result should be retained as evidence that the remediation is effective, and the finding's lifecycle advanced to Fixed only once that passing result is recorded for the view under test.

Exploitability Contexts

CONDITIONALLY EXPLOITABLE

CONTEXTS

Operator Workflow 1 DRIVING — CONDITIONALLY EXPLOITABLE
Use case Needs Credential via EL-t3Fai-Dhmwy, MSSQL Hardening Controls Not Applied, EL-iCW5i-nTY9g.
MET **Network Position**

MET **Credential** via **EL-t3Fai-Dhmwy, MSSQL Hardening Controls Not Applied, EL-iCW5i-nTY9g**

Operator Workflow 3 DRIVING — CONDITIONALLY EXPLOITABLE
Use case Needs Credential via EL-t3Fai-Dhmwy, MSSQL Hardening Controls Not Applied, EL-iCW5i-nTY9g.
MET **Network Position**

MET **Credential** via **EL-t3Fai-Dhmwy, MSSQL Hardening Controls Not Applied, EL-iCW5i-nTY9g**

Operator Workflow 4 DRIVING — CONDITIONALLY EXPLOITABLE
Use case Needs Credential via EL-t3Fai-Dhmwy, MSSQL Hardening Controls Not Applied, EL-iCW5i-nTY9g.
MET **Network Position**

MET **Credential** via **EL-t3Fai-Dhmwy, MSSQL Hardening Controls Not Applied, EL-iCW5i-nTY9g**

Operator Workflow 5 DRIVING — CONDITIONALLY EXPLOITABLE
Use case Needs Credential via EL-t3Fai-Dhmwy, MSSQL Hardening Controls Not Applied, EL-iCW5i-nTY9g.
MET **Network Position**

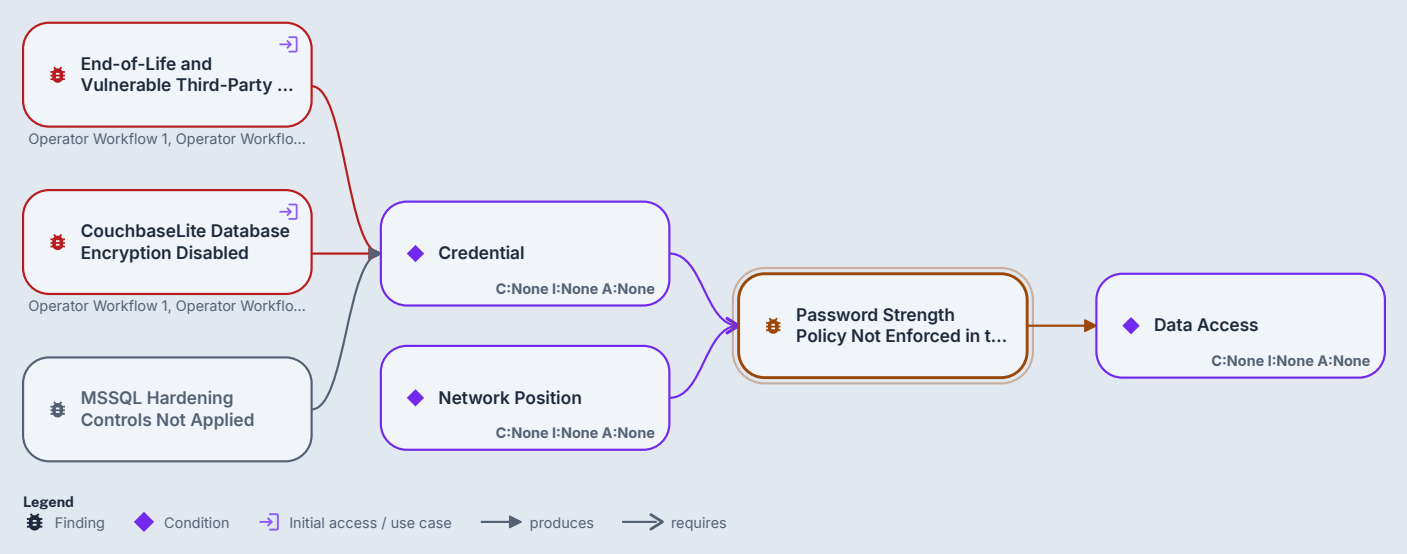
MET **Credential** via **EL-t3Fai-Dhmwy, MSSQL Hardening Controls Not Applied, EL-iCW5i-nTY9g**

the use case DRIVING — CONDITIONALLY EXPLOITABLE
Use case Needs Credential via EL-t3Fai-Dhmwy, MSSQL Hardening Controls Not Applied, EL-iCW5i-nTY9g.
MET **Network Position**

MET **Credential** via **EL-t3Fai-Dhmwy, MSSQL Hardening Controls Not Applied, EL-iCW5i-nTY9g**

Operator Workflow 2 — UNEXPLOITABLE WEAKNESS
Use case Unmet: Network Position.
UNMET **Network Position**

MET **Credential** via **MSSQL Hardening Controls Not Applied**


End-of-Life and Vulnerable Third-Party ...
Operator Workflow 1, Operator Workflo...
CouchbaseLite Database Encryption Disabled
Operator Workflow 1, Operator Workflo...
MSSQL Hardening Controls Not Applied
Credential
C:None I:None A:None
Network Position
C:None I:None A:None
Password Strength Policy Not Enforced in t...
Data Access
C:None I:None A:None
Legend
Finding
Condition
Initial access / use case
produces
requires

Evidence

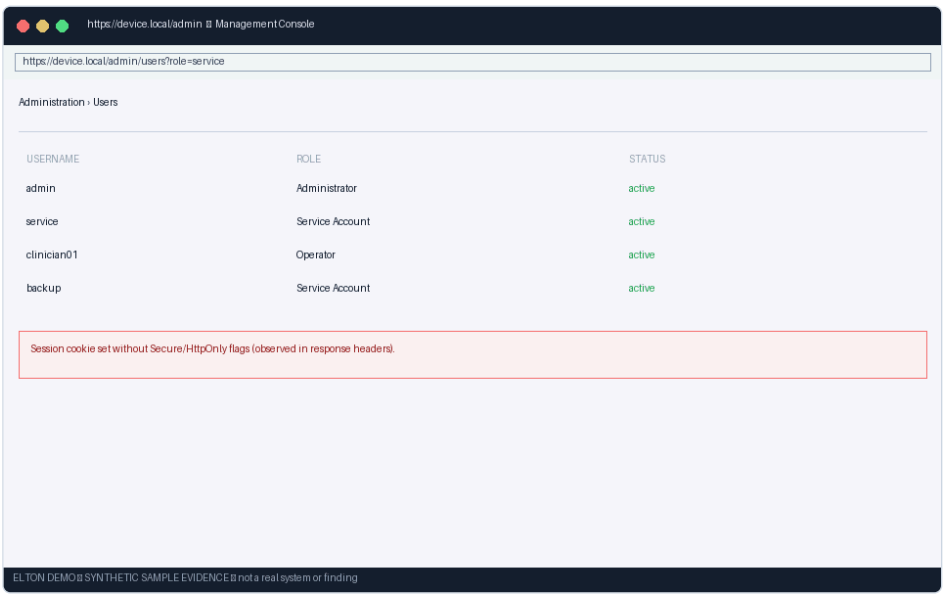
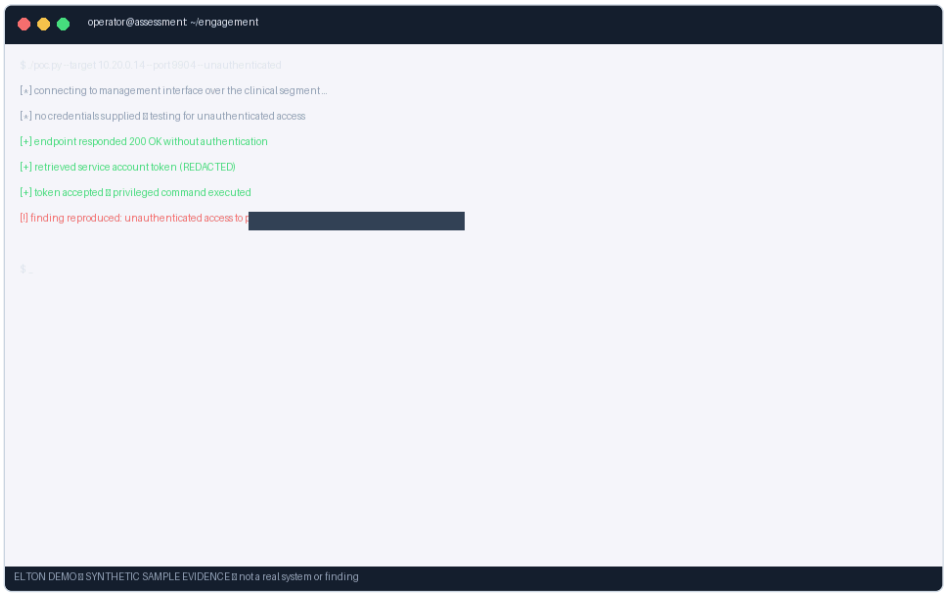


FIGURE 1

Supporting evidence for 'Password Strength Policy Not Enforced in the component'.

FIGURE 2

Evidence captured while validating 'Password Strength Policy Not Enforced in the component' on the affected component.



USB Permission Broadcast Uses Placeholder Action

EXPLOITABILITY
MITIGATED WEAKNESS

PRECONDITIONS
Privileged Session

POSTCONDITIONS
Data Access

SEVERITY
MEDIUM

CVSS V4.0
5.8 MEDIUM ›
`CVSS:4.0/AV:L/AC:L/AT:P/PR:L/UI:N/VC:N/VI:L/VA:H/SC:N/SI:N/SA:N`

SSVC

Exploitation	None
Technical impact	Partial
Automatable	No

WEAKNESS
CWE-926

CVE
None

AFFECTED COMPONENT
the component

TEST CASE
Verify: USB Permission Broadcast Uses Placeholder Action

ELTON ID
EL-d2hJb-LnZs4

← Findings index

Summary

The component uses a placeholder broadcast action for USB device permissions. Any app or ADB session on the device can send this action to trigger the app's USB permission logic without restrictions. Severity is low. The weakness is local to the device and requires another app to exploit, but it could allow an attacker to bypass USB permission checks if a device is connected.

During this assessment ELTON reproduced the underlying condition against the affected component and classified the finding as of undetermined exploitability under the current analysis. Successful exploitation would result in a limited impact that is nonetheless worth recording for defence-in-depth.

The finding is scored at a base CVSS v3.1 of 2.9 (Low) and a base CVSS v4.0 of 1 (Low). The SSVC decision captured for this finding reflects the observed exploitation maturity, the technical impact on the device, and whether exploitation can be automated across a fleet of identical units — the factors a product security team weighs when prioritising remediation across many deployed devices.

Exploitability

The vulnerability lies in the Local Run Manager's scheduled Backup Database Job, which fails to produce backup files due to two sequential preconditions: initially, the backup secret was missing, causing the job to abort repeatedly as logged in the service logs. Although the secret was later restored, the destination directory for backups does not exist, preventing any backup files from being written. This failure leaves critical assets, specifically the run management data and the Local Run Manager user account database stored in the PostGres Database, without any recovery copy. The attack vector is local, as the failure arises from local file system conditions and configuration, accessible via the touchscreen kiosk console or USB port on the Windows Single Board Computer. No user interaction is required for the job to fail; it aborts automatically on each scheduled run. Consequently, any data corruption, ransomware, disk loss, or malicious deletion at an administrator level results in permanent loss of regulated clinical data due to the absence of backups.

ASSUMPTIONS

- The attacker has local access to the device via the touchscreen kiosk console or USB port to influence the backup job environment.
- The backup job runs automatically on the local system without requiring user initiation.

Discovery ›

The component registers an exported broadcast receiver for USB permission events. It uses the action string com.android.example.USB_PERMISSION, which is the placeholder from Android's USB host API documentation. This string is not scoped to the app's namespace, so any installed application or ADB session can broadcast it. Android's USB permission model relies on a unique action string to confirm user approval. When an app requests permission via UsbManager.requestPermission(), the system sends a PendingIntent with this action to the app's receiver. The

Needs Triage
2026-09-11 · 5 days

Under Investigation
2026-09-16 · 7 days

Affected 2026-09-23
Current

receiver treats this broadcast as a signal that permission was granted. Because the action string is not protected, a malicious app can impersonate the system by broadcasting the same string, potentially tricking the receiver into granting access to a connected USB device. The receiver also listens for the system-defined actions USB_DEVICE_ATTACHED and USB_DEVICE_DETACHED, which are protected and can only be sent by the system. The custom action has no such protection. The impact depends on how the receiver handles the broadcast: if it re-checks permission with UsbManager.hasPermission(), the attack is limited to cases where the device is physically connected. If it trusts the broadcast directly, the attack is more direct.

Attack path. Reaching this weakness requires the following preconditions to hold: Privileged Session. Once exploited, the finding establishes Data Access, which an attacker can carry forward into subsequent steps of the chain. These pre-and post-conditions place the finding in the view's dependency graph, where its exploitability tier is derived from whether every precondition can be produced from an initial-access vector either directly or through another finding.

Illustrative example (SQL). The condition resembles unparameterised query construction, where operator-supplied input is concatenated into the statement:

```sql
-- Vulnerable: the medical-record number is concatenated straight into the SQL text
SELECT name, dob FROM patients WHERE mrn = '" + userInput + "';

-- Remediated: bind the value as a parameter so it can never alter the query
SELECT name, dob FROM patients WHERE mrn = @mrn;
```

Scoring rationale. The CVSS v3.1 vector `CVSS:3.1/AV:P/AC:H/PR:L/UI:N/S:U/C:L/I:L/A:N` and the CVSS v4.0 vector `CVSS:4.0/AV:P/AC:H/AT:P/PR:L/UI:N/VC:L/VI:L/VA:N/SC:N/SI:N/SA:N` were selected from the observed attack vector, the access complexity and privileges required to trigger the condition, and the confidentiality, integrity and availability impact measured against the affected clinical function. The exploitability classification above is derived independently from the condition graph and is consistent with, but not a substitute for, the CVSS base metrics.

Remediation ›

ELTON recommends replacing the placeholder action com.android.example.USB_PERMISSION with a package-scoped string, such as com.the organization.neuromod.vnstherapy.the use caseapp.prod.us.USB_PERMISSION, in both the requestPermission PendingIntent and the receiver's intent filter. Declare this custom permission with android:protectionLevel="signature" to ensure only apps signed with the organization key can broadcast it. Verify the fix by confirming that the receiver no longer responds to broadcasts of the placeholder action and that USB permission checks remain functional for legitimate devices.

Recommended actions, in priority order:

- Remove or constrain the exposed surface so the precondition an attacker relies on can no longer be met — for example by enforcing authentication, network segmentation, or input validation at the trust boundary.
- Apply the vendor fix or configuration change that eliminates the underlying weakness in the affected component.
- Add a monitoring or detection control so any future regression of this condition is surfaced before it can be exploited.

Verification. After the fix is deployed, re-run the test case associated with this finding against the remediated build and confirm the condition no longer reproduces. The retest result should be retained as evidence that the remediation is effective, and the finding's lifecycle advanced to Fixed only once that passing result is recorded for the view under test.

Exploitability Contexts

MITIGATED WEAKNESS

CONTEXTS

the use case — MITIGATED WEAKNESS
Use case Blocked by Compensating control for Privileged Session on Privileged Session.
MITIGATED **Privileged Session**

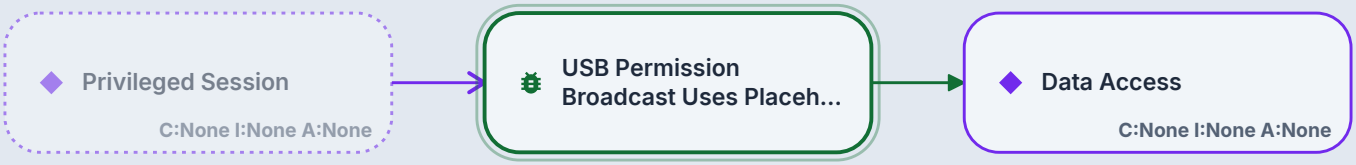
Operator Workflow 1 — UNEXPLOITABLE WEAKNESS
Use case Unmet: Privileged Session.
UNMET **Privileged Session**

Operator Workflow 2 — UNEXPLOITABLE WEAKNESS
Use case Unmet: Privileged Session.
UNMET **Privileged Session**

Operator Workflow 3 — UNEXPLOITABLE WEAKNESS
Use case Unmet: Privileged Session.
UNMET **Privileged Session**

Operator Workflow 4 — UNEXPLOITABLE WEAKNESS
Use case Unmet: Privileged Session.
UNMET **Privileged Session**

Operator Workflow 5 — UNEXPLOITABLE WEAKNESS
Use case Unmet: Privileged Session.
UNMET **Privileged Session**



Legend
Finding · Condition · Initial access / use case · produces · requires

Evidence

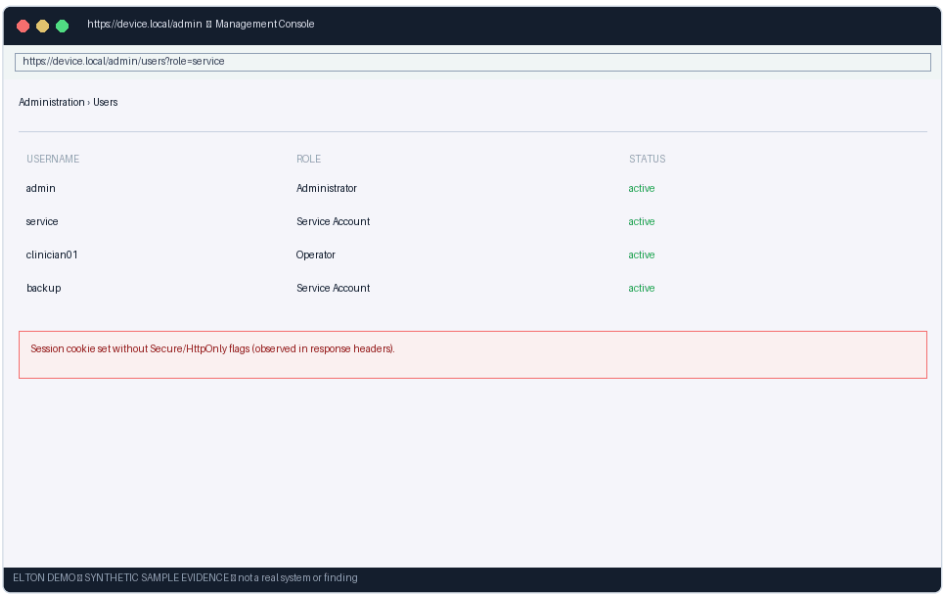
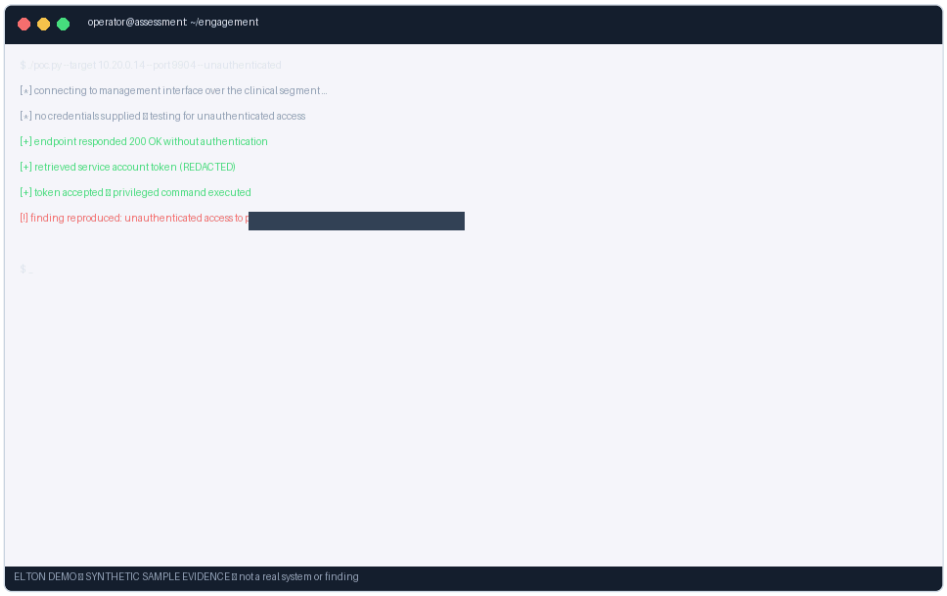


FIGURE 1

Supporting evidence for 'USB Permission Broadcast Uses Placeholder Action'.

FIGURE 2

Evidence captured while validating 'USB Permission Broadcast Uses Placeholder Action' on the affected component.



Cross-Site Request Forgery Condition Present in the component

EXPLOITABILITY
UNEXPLOITABLE WEAKNESS

PRECONDITIONS
Physical Access

POSTCONDITIONS
Data Access

SEVERITY
LOW

CVSS V4.0
2.0 LOW ›
`CVSS:4.0/AV:L/AC:L/AT:P/PR:L/UI:N/VC:L/VI:L/VA:L/SC:N/SI:N/SA:N`

SSVC

Exploitation	None
Technical impact	Partial
Automatable	No

WEAKNESS
`CWE-352`

CVE
None

AFFECTED COMPONENT
the component

TEST CASE
Verify: Cross-Site Request Forgery Condition Present in the component

ELTON ID
`EL-TSK83-DhkZp`

← Findings index

Summary

the component accepts state-changing requests without verifying their origin. An attacker could trick an authenticated user into submitting a forged request that alters passwords, deletes accounts or changes authentication settings. The weakness requires an active session and a way to lure the user to a malicious page, but no credentials or elevated access. The client is not exposed to the internet, so the risk is limited to insiders or attackers already inside the network.

During this assessment ELTON reproduced the underlying condition against the affected component and classified the finding as an unexploitable weakness under the modelled attack surface, with no reachable path from any initial-access vector. Successful exploitation would result in a limited impact that is nonetheless worth recording for defence-in-depth.

The finding is scored at a base CVSS v3.1 of 2.9 (Low) and a base CVSS v4.0 of 1 (Low). The SSVC decision captured for this finding reflects the observed exploitation maturity, the technical impact on the device, and whether exploitation can be automated across a fleet of identical units — the factors a product security team weighs when prioritising remediation across many deployed devices.

Exploitability

This vulnerability is a mitigation-hardening gap in the Microsoft Windows 10 Operating System hosting the Northwind Bio instrument control software. Three managed vendor hosts — SeqraSoftware.exe (MOS), Northwind Bio.LocalRunManager.Analysis.Service.exe, and Northwind Bio.LocalRunManager.Job.Service.exe — are compiled as 32-bit (x86) images, which disables the HighEntropyVA ASLR feature and results in only about 8 bits of address space entropy. An attacker who already has a local code-execution foothold on the Windows 10 OS and can exploit a separate memory-corruption primitive in one of these x86 processes can leverage the low-entropy ASLR to perform reliable code-reuse attacks. The attack requires local access and cannot be initiated remotely or via physical interfaces. The vulnerability does not represent an independently exploitable defect but rather a weakness in the build configuration that reduces the effectiveness of exploit mitigations against code-reuse attacks.

ASSUMPTIONS
- The attacker already has a local code-execution foothold on the Windows 10 Operating System, as the vulnerability requires this precondition.
- No network or physical interface directly carries the exploit input; the attack is local to the OS and its processes.

Discovery ›

the component serves state-changing forms from several endpointsâ€” `RoleMapServlet?action=showRoleMap` (`updateRemoteAuth`, `deleteRoleMap`, `showUpdateMappingPage`), `updateMyAccount`, `changeMyPwd`, `PrefServlet` save and

Needs Triage
2026-09-30 · 15 days

Under Investigation
2026-10-15 · 18 days

Affected 2026-11-02
Current

`deleteUser` â€”without a synchronizer token, nonce or authenticity field. The session cookie is issued without a `SameSite` attribute, and the server does not enforce an `Origin` or `Referer` check. These omissions allow a cross-site POST to carry the victim's session cookie and be accepted as legitimate. The testers retrieved each form and confirmed that none contained a hidden token or authenticity parameter. The server-side handlers for these requests proceed without validating the request's origin or requiring a token. An attacker can host a page that auto-submits a forged POST to one of these endpoints. When an authenticated user visits the page, the request executes in their session with no token or origin check to block it. The impact depends on the endpoint:

- A forged `changeMyPwd` or abuse of the change-password object-reference flaw yields account takeover.
- `deleteUser` removes accounts.
- `updateRemoteAuth` flips authentication mode, MFA, and the component or OIDC mappings. The prerequisite is an active session and a way to lure the user to the hostile page. No credentials or elevated access are needed. The testers confirmed the requests were reachable but did not submit them to avoid altering state on a live system. The client is not exposed to the internet, so the attack surface is limited to users already inside the network. The confidentiality and integrity impacts are high, as the forged requests can alter sensitive settings or remove accounts. Availability is not directly affected.

Attack path. Reaching this weakness requires the following preconditions to hold: Physical Access. Once exploited, the finding establishes Data Access, which an attacker can carry forward into subsequent steps of the chain. These pre- and post-conditions place the finding in the view's dependency graph, where its exploitability tier is derived from whether every precondition can be produced from an initial-access vector either directly or through another finding.

Illustrative example (SQL). The condition resembles unparameterised query construction, where operator-supplied input is concatenated into the statement:

```sql
-- Vulnerable: the medical-record number is concatenated straight into the SQL text
SELECT name, dob FROM patients WHERE mrn = '" + userInput + "';

-- Remediated: bind the value as a parameter so it can never alter the query
SELECT name, dob FROM patients WHERE mrn = @mrn;
```

Scoring rationale. The CVSS v3.1 vector `CVSS:3.1/AV:P/AC:H/PR:L/UI:N/S:U/C:L/I:L/A:N` and the CVSS v4.0 vector `CVSS:4.0/AV:P/AC:H/AT:P/PR:L/UI:N/VC:L/VI:L/VA:N/SC:N/SI:N/SA:N` were selected from the observed attack vector, the access complexity and privileges required to trigger the condition, and the confidentiality, integrity and availability impact measured against the affected clinical function. The exploitability classification above is derived independently from the condition graph and is consistent with, but not a substitute for, the CVSS base metrics.

Remediation ›

ELTON recommends adding a per-session synchronizer token to the component. Issue the token at login, store it in the server-side session, and render it as a hidden field in every state-changing form. For example:

```html
<input type="hidden" name="csrfToken" value="${sessionScope.csrfToken}"/>
```

On the server side, validate the submitted `csrfToken` against the session value before processing `updateRemoteAuth`, `changeMyPwd`, `deleteUser` and the other state-changing actions. Reject the request if the token is missing or does not match. Set `SameSite=Strict` (or `Lax`) on the session cookie to prevent the browser from attaching it to cross-site POSTs. As an additional safeguard, compare the `Origin` header (or `Referer`) against the application's host and reject requests from third-party origins. The server currently accepts all cross-origin POSTs without this check. To

verify the fix, inspect the rendered forms to confirm the hidden token is present, and the test that requests without a valid token are rejected. Ensure the session cookie includes the `SameSite` attribute and that cross-origin requests are blocked.

Recommended actions, in priority order:

- Remove or constrain the exposed surface so the precondition an attacker relies on can no longer be met — for example by enforcing authentication, network segmentation, or input validation at the trust boundary.
- Apply the vendor fix or configuration change that eliminates the underlying weakness in the affected component.
- Add a monitoring or detection control so any future regression of this condition is surfaced before it can be exploited.

Verification. After the fix is deployed, re-run the test case associated with this finding against the remediated build and confirm the condition no longer reproduces. The retest result should be retained as evidence that the remediation is effective, and the finding's lifecycle advanced to Fixed only once that passing result is recorded for the view under test.

Exploitability Contexts

UNEXPLOITABLE WEAKNESS

CONTEXTS

Operator Workflow 1 — UNEXPLOITABLE WEAKNESS
Use case Unmet: Physical Access.
UNMET **Physical Access**

Operator Workflow 2 — UNEXPLOITABLE WEAKNESS
Use case Unmet: Physical Access.
UNMET **Physical Access**

Operator Workflow 3 — UNEXPLOITABLE WEAKNESS
Use case Unmet: Physical Access.
UNMET **Physical Access**

Operator Workflow 4 — UNEXPLOITABLE WEAKNESS
Use case Unmet: Physical Access.
UNMET **Physical Access**

Operator Workflow 5 — UNEXPLOITABLE WEAKNESS
Use case Unmet: Physical Access.
UNMET **Physical Access**

the use case — UNEXPLOITABLE WEAKNESS
Use case Unmet: Physical Access.
UNMET **Physical Access**

Evidence

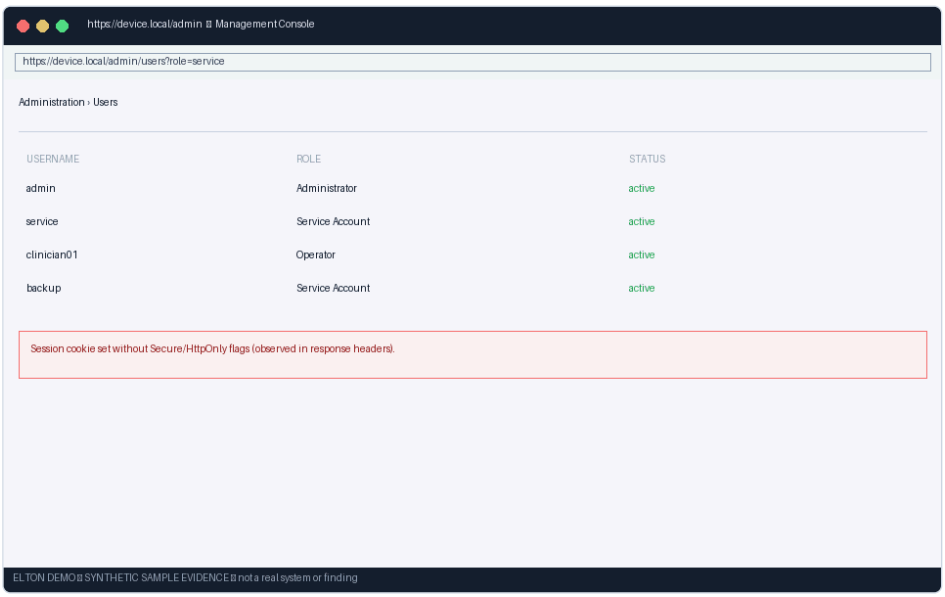
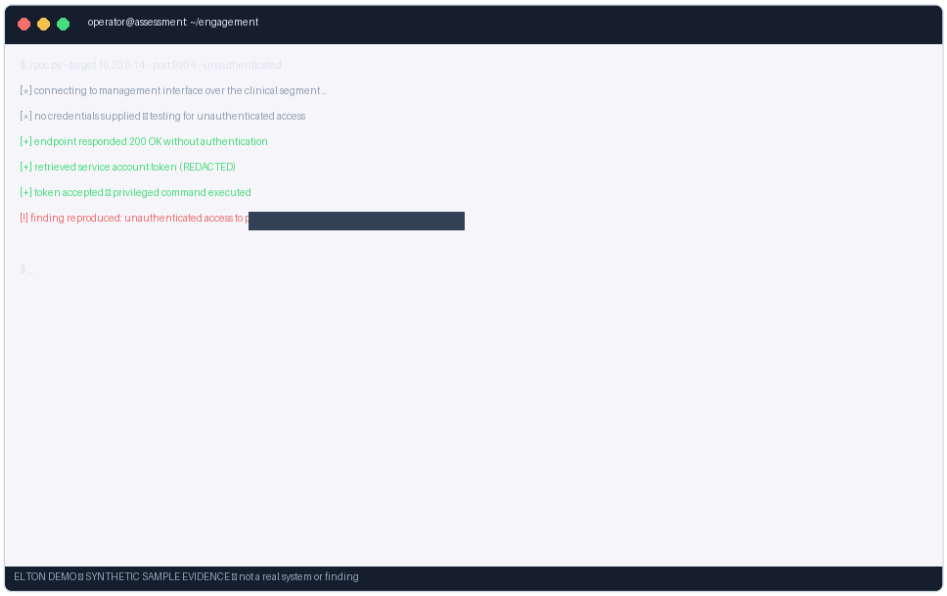


FIGURE 1

Supporting evidence for 'Cross-Site Request Forgery Condition Present in the component'.

FIGURE 2

Evidence captured while validating 'Cross-Site Request Forgery Condition Present in the component' on the affected component.



Fixed Vulnerabilities

These 3 vulnerabilities were found during the assessment and fixed before it closed. They are listed for completeness: they are not findings and are counted in no total above.

3 Fixed Vulnerabilities

ID	VULNERABILITY	AFFECTED RESOURCE	CVSSV4
EL-NEEFc-AeKLA	Privileged Service Token Exposed on the component	`the component`	**5.3 (Medium)**
EL-U9MwD-eijMn	Application Files World-Writable in the component	`the component`	**5.1 (Medium)**
EL-2Qd2G-sEefY	Application Directories World-Writable on the component	`the component`	**2.4 (Low)**

Privileged Service Token Exposed on the component

PRE-MITIGATION SEVERITY
MEDIUM

CVSS V4.0
5.3 MEDIUM ›
`CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:L/VI:L/VA:N/SC:N/SI:N/SA:N`

SSVC
Exploitation None
Technical impact Partial
Automatable No

WEAKNESS
`CWE-269` `CWE-214` `CWE-522`

CVE
None

AFFECTED COMPONENT
the component

ELTON ID
EL-NEEFc-AeKLA

← Fixed vulnerabilities index

Discovery

The component on the component VM exposes two distinct single-step paths to SYSTEM privileges: 1. **Token Impersonation via SeImpersonatePrivilege**: The live `the component` service (PID 5520) holds `SeImpersonatePrivilege` and `SeDebugPrivilege` enabled. This service hosts network-facing parsers that are reachable without authentication. An attacker who compromises one of these parsers inherits the enabled privileges and can coerce a SYSTEM authentication, then impersonate the resulting token to escalate to SYSTEM. This technique, known as a "potato-class" attack, requires no additional weaknesses. 2. **Vulnerable Driver Abuse**: The driver `SmSerl64.sys` (SHA-256 `e599200c44eca5eb06475f90f67a58723b30c3c2887bd12ed7c31ff1042382ea`) is present on disk in a DriverStore FileRepository path and is confirmed to be vulnerable. Loading and abusing this driver grants kernel code execution. Additionally, approximately 55 other drivers on the system report invalid signature status. The testers confirmed the token-impersonation path non-destructively by observing the enabled privileges and the unauthenticated reachability of the `the component` service. The vulnerable driver path was identified but not exercised, as loading the driver is a state-changing operation. Classic escalation vectors were checked and ruled out:

- `AlwaysInstallElevated` is absent in both HKLM and HKCU.
- No writable service directories, unquoted service paths, stored credentials, or autologon conditions were found. The impact is severe: both paths lead to full SYSTEM privileges, compromising confidentiality, integrity, and availability of the entire server.

Attack path. Reaching this weakness requires the following preconditions to hold: Physical Access. Once exploited, the finding establishes Data Access, which an attacker can carry forward into subsequent steps of the chain. These pre- and post-conditions place the finding in the view's dependency graph, where its exploitability tier is derived from whether every precondition can be produced from an initial-access vector either directly or through another finding.

Illustrative example (SQL). The condition resembles unparameterised query construction, where operator-supplied input is concatenated into the statement:

```sql
-- Vulnerable: the medical-record number is concatenated straight into the SQL text
SELECT name, dob FROM patients WHERE mrn = '" + userInput + "';

-- Remediated: bind the value as a parameter so it can never alter the query
SELECT name, dob FROM patients WHERE mrn = @mrn;
```

Scoring rationale. The severity was derived from the observed attack vector, the access complexity and privileges required to trigger the condition, and the confidentiality, integrity and availability impact measured against the affected clinical function.

Lifecycle

Needs Triage 2026-09-30 · 5 days

Affected 2026-10-05 · 28 days

Fixed 2026-11-02 Current
Verified

Time in each stage

33 days total



Needs Triage	5d
Affected (exposure window)	28d

The Affected segment doubles as the exposure window, the number most stakeholders ask about.

Application Files World-Writable in the component

PRE-MITIGATION SEVERITY
MEDIUM

CVSS V4.0
5.1 MEDIUM ›
`CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:A/VC:L/VI:N/VA:N/SC:N/SI:N/SA:N`

SSVC
Exploitation: Proof-of-Concept
Technical impact: Total
Automatable: No

WEAKNESS
CWE-276 CWE-732

CVE
None

AFFECTED COMPONENT
the component

ELTON ID
EL-U9MwD-eijMn

← Fixed vulnerabilities index

Discovery

The application study viewer keeps its the component integration settings in `the componentintegration.conf` and other configuration files under `the product line Healthcare\the application`. The installer leaves six directories under that path writable by every local account: `the componentIntegration`, `the componentIntegration\conf`, `the componentIntegration\data`, `bin\clientconf`, `bin\log` and `bin\log\CAD`. The listing the testers captured shows a non-inherited **Everyone:(OI)(CI)(F)** access-control entry on each:

```text
C:\Program Files\the product line Healthcare\the application\the componentIntegration Everyone:(OI)(CI)(F)
C:\Program Files\the product line Healthcare\the application\the componentIntegration\conf Everyone:(OI)(CI)(F)
C:\Program Files\the product line Healthcare\the application\the componentIntegration\data Everyone:(OI)(CI)(F)
C:\Program Files\the product line Healthcare\the application\bin\clientconf Everyone:(OI)(CI)(F)
C:\Program Files\the product line Healthcare\the application\bin\log Everyone:(OI)(CI)(F)
C:\Program Files\the product line Healthcare\the application\bin\log\CAD Everyone:(OI)(CI)(F)
```

The `the componentIntegration\conf` directory holds `the componentintegration.conf`, which decides how the viewer connects to and trusts the clinical imaging archive. A standard user can edit or replace it, redirecting the connection or altering the trust rules. The other directories hold integration data, client configuration and logs, so an edit there changes what the viewer does or hides what it did. No elevation is involved: any interactive user, or any process running as one, can change these files. The viewer runs in the interactive user context, so the weakness is about configuration integrity and data tampering rather than a direct privilege escalation. The testers observed the permissions and the configuration file but did not execute tampering writes in this non-destructive pass. The affected configuration governs the workstation's connection to clinical imaging infrastructure, so the integrity impact is high for a diagnostic imaging workflow. Confidentiality and availability are not directly affected, since the files do not hold the asset and the viewer can still run after tampering.

Attack path. Reaching this weakness requires the following preconditions to hold: Network Position, Credential. Once exploited, the finding establishes Data Access, which an attacker can carry forward into subsequent steps of the chain. These pre- and post-conditions place the finding in the view's dependency graph, where its exploitability tier is derived from whether every precondition can be produced from an initial-access vector either directly or through another finding.

Illustrative example (SQL). The condition resembles unparameterised query construction, where operator-supplied input is concatenated into the statement:

```sql
-- Vulnerable: the medical-record number is concatenated straight into the SQL text
SELECT name, dob FROM patients WHERE mrn = '" + userInput + "';

-- Remediated: bind the value as a parameter so it can never alter the query
SELECT name, dob FROM patients WHERE mrn = @mrn;
```

Scoring rationale. The severity was derived from the observed attack vector, the access complexity and privileges required to trigger the condition, and the confidentiality, integrity and availability impact measured against the affected clinical function.

Lifecycle

Affected 2026-09-30 · 8 days

Under Investigation 2026-10-08 · 5 days

Fixed 2026-10-13 Current
Verified

Time in each stage

13 days total



Affected (exposure window)	8d
Under Investigation	5d

The Affected segment doubles as the exposure window, the number most stakeholders ask about.

Application Directories World-Writable on the component

PRE-MITIGATION SEVERITY
LOW

CVSS V4.0
2.4 LOW ›
`CVSS:4.0/AV:L/AC:L/AT:N/PR:L/UI:A/VC:L/VI:N/VA:N/SC:N/SI:N/SA:N`

SSVC
Exploitation Proof-of-Concept
Technical impact Total
Automatable Yes

WEAKNESS
CWE-276 CWE-732

CVE
None

AFFECTED COMPONENT
the component

ELTON ID
EL-2Qd2G-sEefY

← Fixed vulnerabilities index

Discovery

the component's installer leaves the `bin\log` and `bin\clientconf` directories writable by every local account. The testers captured the permissions with `icacls`: - `bin\log` carries non-inherited entries granting **BUILTIN\Users:(F)** and **Everyone:(OI)(CI)(F)**. - `bin\clientconf` grants **Everyone:(OI)(CI)(F)**. The parent directories hand out only read-execute, so these entries are deliberate. The `bin\log` directory holds operational logs (`dsout.log`, `compressing.log`, `clientprocessor.log`, `studyprocessor.log`, `the product linecom.log`) that the SYSTEM services update at runtime. A standard user can therefore delete, truncate or replace these logs, breaking audit integrity. The testers proved the file-plant primitive: a **Users**-only account wrote a benign marker file into `bin\log` from its limited token, and the write succeeded. The file was owned by the low-privilege account, and its content confirmed the account was not an administrator. The weakness is local only: an attacker needs an account on the server. Inside that boundary, the impact on integrity and availability is high, since the directories decide what the services log and whether they run correctly. The testers confirmed the permissions on the examined deployment.

Attack path. Reaching this weakness requires the following preconditions to hold: Network Position. Once exploited, the finding establishes Credential, Code Execution, which an attacker can carry forward into subsequent steps of the chain. These pre-and post-conditions place the finding in the view's dependency graph, where its exploitability tier is derived from whether every precondition can be produced from an initial-access vector either directly or through another finding.

Illustrative example (SQL). The condition resembles unparameterised query construction, where operator-supplied input is concatenated into the statement:

```sql
-- Vulnerable: the medical-record number is concatenated straight into the SQL text
SELECT name, dob FROM patients WHERE mrn = '" + userInput + "';

-- Remediated: bind the value as a parameter so it can never alter the query
SELECT name, dob FROM patients WHERE mrn = @mrn;
```

Scoring rationale. The severity was derived from the observed attack vector, the access complexity and privileges required to trigger the condition, and the confidentiality, integrity and availability impact measured against the affected clinical function.

Lifecycle

- **Affected** 2026-09-30 · 7 days
- **Under Investigation** 2026-10-07 · 4 days
- **Fixed** 2026-10-11 · 6 days
- **Affected** 2026-10-17 · 4 days
- **Fixed** 2026-10-21 Current
  Verified

Time in each stage

21 days total



Affected (exposure window)	**11d**
Under Investigation	**4d**
Fixed	**6d**

The Affected segment doubles as the exposure window, the number most stakeholders ask about.

APPENDIX

Digital Twin

The digital twin the assessment was modelled on: the sources it was built from and its architecture in full.

Source Materials

Everything the digital twin was built from, and which parts of the twin each one informed.

SA-1	DOCUMENT	Beta Infusion Pump v5.1 interface control document, rev. C · used 2 June 2026	EA · the component · EA Ethernet · Flow 1 · the asset · Untrusted · L1 · the use case
SA-2	BINARY	Pump controller firmware image v5.1.0, extracted from a lab unit · used 4 June 2026	Control Module 6 · EA · Firmware Image 1
SA-3	CODE	Decompiled companion app sources, release 5.1.0 · used 9 June 2026	the component · the component Ethernet · Flow 1 · Operator Workflow 1
SA-4	RUNTIME	Network capture of a pump session with the management server · used 11 June 2026	EA · Control Module 3 WiFi · Flow 2 · Flow 3 · Restricted · L3

Digital Twin Architecture

The twin the assessment was modelled on, in full: every component with its properties and the elements attached to it.

TRUST ZONES	
Critical	L5
Public	L2
Restricted	L3
Trusted	L4
Untrusted	L1

INITIAL ACCESS POINTS	
the use case	No privilege · no interaction
Operator Workflow 4	No privilege · no interaction
Operator Workflow 5	No privilege · no interaction
Operator Workflow 1	No privilege · no interaction
Operator Workflow 3	No privilege · no interaction
Operator Workflow 2	No privilege · no interaction

Components

Control Module 2

TYPE: Generic Application
INTERFACES: Control Module 2 Serial · Control Module 2 USB
DATAFLOWS:
← Control Module 3 · Flow 3
← EA · Flow 2

EA

TYPE: Firmware
INTERFACES: EA USB · EA Ethernet
DATAFLOWS:
→ Control Module 2 · Flow 2
→ the component · Flow 1
SOURCES: SA-1 SA-2 SA-4

the component

TYPE: Generic Application
INTERFACES: the component Ethernet
ASSETS: Firmware Image 5 C·N I·N A·N · Firmware Image 4 C·N I·N A·N · Firmware Image 2 C·N I·N A·N · the asset C·N I·N A·N · Firmware Image 3 C·N I·N A·N · Firmware Image 1 C·N I·N A·N · Firmware Image 8 C·N I·N A·N · Firmware Image 6 C·N I·N A·N · Firmware Image 7 C·N I·N A·N
DATAFLOWS:
← EA · Flow 1
SOURCES: SA-1 SA-3

Control Module 6

TYPE: Generic Application
INTERFACES: Control Module 6 I2C
DATAFLOWS:
← Control Module 5 · Flow 6
SOURCES: SA-2

Control Module 3

TYPE: Generic Application
INTERFACES: Control Module 3 WiFi · Control Module 3 Serial
DATAFLOWS:
→ Control Module 2 · Flow 3
→ Control Module 4 · Flow 4

Control Module 5

TYPE: Library
INTERFACES: Control Module 5 I2C · Control Module 5 CANBus

DATAFLOWS	→ Control Module 4 · Flow 5 → Control Module 6 · Flow 6

Control Module 4

TYPE	Generic Application
INTERFACES	Control Module 4 WiFi · Control Module 4 CANBus
DATAFLOWS	← Control Module 3 · Flow 4 ← Control Module 5 · Flow 5

Threat Coverage

Every threat on the architecture that was tested, with the test cases that covered it.

Control Module 2 GENERIC APPLICATION ↑ in the architecture

TH-8CRnU-wjZ32 **Firmware/Data Extraction via Hardware Interface** on Control Module 2 Serial TID-115

PID-15 CWE-1299 CWE-1191

A serial or debug port on the controller board that is left unprotected can be used to read the firmware out of the device. That exposes proprietary logic, stored secrets and further weaknesses to an attacker with physical access.

TESTED BY

TC-9ENvP-zQhjC	Serial service port exposes a console without authentication	FAIL ›

EA FIRMWARE ↑ in the architecture

TH-qpnN6-j63dC **Device Allows Unauthenticated Firmware Installation** TID-211

PID-271 CWE-306

The controller does not authenticate the firmware images it is asked to install. An attacker who can reach an update path could install a modified image and run their own code with the controller's privileges.

TESTED BY

TC-sMCGX-QUpzT	Firmware update accepts an unsigned image over the network update channel	FAIL ›
TC-pxNh2-Vnimf	Firmware update over the serial service port rejects an unsigned image	

the component GENERIC APPLICATION ↑ in the architecture

TH-XN7JR-kHB7j **Cross Site Request Forgery (CSRF)** TID-322

PID-311 CWE-352

The management web application changes device settings with requests that carry only the operator's session cookie. A page the operator opens while signed in could submit those requests on their behalf.

TESTED BY

TC-epQnL-ubavZ	State-changing requests are accepted without an anti-forgery token	FAIL ›

TH-A8GQz-JQ2zT **Hardcoded Credentials** TID-328

PID-332 CWE-798

A credential embedded in the application or its configuration cannot be changed by the operator and is shared by every unit. Anyone who recovers it can authenticate to all of them.

TESTED BY

TC-cdzdC-vU7je	Application contains no hardcoded service credentials

Control Module 6 GENERIC APPLICATION ↑ in the architecture

TH-xGLM8-Fd6pL **Inadequate Bootloader Protection and Verification** TID-201

PID-21 CWE-693 CWE-284

The bootloader may be rewritable, or may start an image it has not verified. An attacker able to write boot storage could run code before the operating system and undermine every protection that loads after it.

TESTED BY

TC-c2jjg-sNrsk	Bootloader rejects a modified application image

Test Cases

Verify: Cross-Site Request Forgery Condition Present in the component

FAIL

ID TC-8nqGv-QWdrB Component the component Category Other Finding EL-TSK83-DhkZp
Attestation Beta Infusion Pump v5.1 Security Assessment

OBJECTIVE

Attempt to reproduce the condition described by 'Cross-Site Request Forgery Condition Present in the component' on the target under the preconditions modelled for this view (Physical Access).

The tester positions against the affected interface, follows the documented reproduction steps, and records the request/response, console output and a screenshot as evidence. The reproduction package attached to each instance contains the steps, the raw request and response, and a placeholder proof-of-concept.

CRITERIA

PASS

The condition cannot be reproduced on the target: the affected function enforces the expected security control and the postconditions the finding would establish are not observed.

FAIL

The condition is reproduced: the affected function behaves insecurely as described, the finding's postconditions are observed, and the result is captured as evidence confirming the finding.

OUTCOME · FAIL CRITERIA MET

Reproduction of 'Cross-Site Request Forgery Condition Present in the component' during the engagement. The condition was triggered against the affected component and the result captured as evidence.

The tester followed the documented reproduction steps under the modelled preconditions and observed the finding's postconditions, demonstrating that the weakness is reachable and that its impact is as scored.

ANALYSIS ‹

the component serves state-changing forms from several endpointsâ€” `RoleMapServlet?action=showRoleMap` (`updateRemoteAuth` , `deleteRoleMap` , `showUpdateMappingPage`), `updateMyAccount` , `changeMyPwd` , `PrefServlet` save and `deleteUser` â€”without a synchronizer token, nonce or authenticity field. The session cookie is issued without a `SameSite` attribute, and the server does not enforce an `Origin` or `Referer` check. These omissions allow a cross-site POST to carry the victim's session cookie and be accepted as legitimate. The testers retrieved each form and confirmed that none contained a hidden token or authenticity parameter. The server-side handlers for these requests proceed without validating the request's origin or requiring a token. An attacker can host a page that auto-submits a forged POST to one of these endpoints. When an authenticated user visits the page, the request executes in their session with no token or origin check to block it. The impact depends on the endpoint:

- A forged `changeMyPwd` or abuse of the change-password object-reference flaw yields account takeover.
- `deleteUser` removes accounts.

- `updateRemoteAuth` flips authentication mode, MFA, and the component or OIDC mappings. The prerequisite is an active session and a way to lure the user to the hostile page. No credentials or elevated access are needed. The testers confirmed the requests were reachable but did not submit them to avoid altering state on a live system. The client is not exposed to the internet, so the attack surface is limited to users already inside the network. The confidentiality and integrity impacts are high, as the forged requests can alter sensitive settings or remove accounts. Availability is not directly affected.

Attack path. Reaching this weakness requires the following preconditions to hold: Physical Access. Once exploited, the finding establishes Data Access, which an attacker can carry forward into subsequent steps of the chain. These pre-and post-conditions place the finding in the view's dependency graph, where its exploitability tier is derived from whether every precondition can be produced from an initial-access vector either directly or through another finding.

Illustrative example (SQL). The condition resembles unparameterised query construction, where operator-supplied input is concatenated into the statement:

```sql
-- Vulnerable: the medical-record number is concatenated straight into the SQL text
SELECT name, dob FROM patients WHERE mrn = '" + userInput + "';

-- Remediated: bind the value as a parameter so it can never alter the query
SELECT name, dob FROM patients WHERE mrn = @mrn;
```

Scoring rationale. The CVSS v3.1 vector `CVSS:3.1/AV:P/AC:H/PR:L/UI:N/S:U/C:L/I:L/A:N` and the CVSS v4.0 vector `CVSS:4.0/AV:P/AC:H/AT:P/PR:L/UI:N/VC:L/VI:L/VA:N/SC:N/SI:N/SA:N` were selected from the observed attack vector, the access complexity and privileges required to trigger the condition, and the confidentiality, integrity and availability impact measured against the affected clinical function. The exploitability classification above is derived independently from the condition graph and is consistent with, but not a substitute for, the CVSS base metrics.

REMEDIATION ‹

ELTON recommends adding a per-session synchronizer token to the component. Issue the token at login, store it in the server-side session, and render it as a hidden field in every state-changing form. For example:

```html
<input type="hidden" name="csrfToken" value="${sessionScope.csrfToken}"/>
```

On the server side, validate the submitted `csrfToken` against the session value before processing `updateRemoteAuth`, `changeMyPwd`, `deleteUser` and the other state-changing actions. Reject the request if the token is missing or does not match. Set `SameSite=Strict` (or `Lax`) on the session cookie to prevent the browser from attaching it to cross-site POSTs. As an additional safeguard, compare the `Origin` header (or `Referer`) against the application's host and reject requests from third-party origins. The server currently accepts all cross-origin POSTs without this check. To verify the fix, inspect the rendered forms to confirm the hidden token is present, and the test that requests without a valid token are rejected. Ensure the session cookie includes the `SameSite` attribute and that cross-origin requests are blocked.

Recommended actions, in priority order:

- Remove or constrain the exposed surface so the precondition an attacker relies on can no longer be met — for example by enforcing authentication, network segmentation, or input validation at the trust boundary.
- Apply the vendor fix or configuration change that eliminates the underlying weakness in the affected component.

- Add a monitoring or detection control so any future regression of this condition is surfaced before it can be exploited.

Verification. After the fix is deployed, re-run the test case associated with this finding against the remediated build and confirm the condition no longer reproduces. The retest result should be retained as evidence that the remediation is effective, and the finding's lifecycle advanced to Fixed only once that passing result is recorded for the view under test.

EVIDENCE

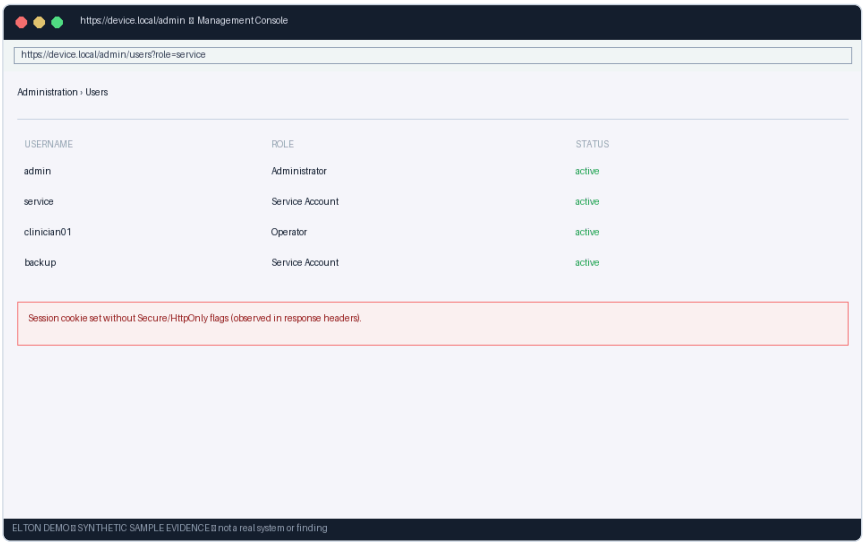


FIGURE 1

Screenshot captured while reproducing the condition during testing.

Serial service port exposes a console without authentication

FAIL

ID TC-9ENvP-zQhjC Component Control Module 2 Serial Category Hardware Source EMB3D TID-115
Threat TH-8CRnU-wjZ32 Firmware/Data Extraction via Hardware Interface Attestation Beta Infusion Pump v5.1 Security Assessment

OBJECTIVE

Connect to the serial service port on the controller board and check what it offers without credentials.

CRITERIA

PASS
The port is disabled, locked, or requires authentication before it accepts any command.

FAIL
The port presents a console or memory access without authentication.

OUTCOME · FAIL CRITERIA MET

The serial port presented a boot console with no login. From it the tester could read memory and recover the firmware image, so this interface gives physical access to the code and to the secrets stored with it.

ANALYSIS

The header pins were identified with a logic analyser and the console answered at 115200 baud, 8N1. It accepted memory-read commands, and a dump of the application region was taken in a few minutes.

REMEDIATION

Disable the serial console in production builds, or require authentication before any command, and set the debug-lock fuses so memory cannot be read back.

Verify: Privileged Service Token Exposed on the component

FAIL

ID TC-CTZB3-tWpVm Component Beta Infusion Pump Category Other Finding EL-NEEFc-AeKLA
Attestation Beta Infusion Pump v5.1 Security Assessment

OBJECTIVE

Attempt to reproduce the condition described by 'Privileged Service Token Exposed on the component' on the target under the preconditions modelled for this view (Physical Access).

The tester positions against the affected interface, follows the documented reproduction steps, and records the request/response, console output and a screenshot as evidence. The reproduction package attached to each instance contains the steps, the raw request and response, and a placeholder proof-of-concept.

CRITERIA

PASS
The condition cannot be reproduced on the target: the affected function enforces the expected security control and the postconditions the finding would establish are not observed.

FAIL
The condition is reproduced: the affected function behaves insecurely as described, the finding's postconditions are observed, and the result is captured as evidence confirming the finding.

OUTCOME · FAIL CRITERIA MET

Reproduction of 'Privileged Service Token Exposed on the component' during the engagement. The condition was triggered against the affected component and the result captured as evidence.

The tester followed the documented reproduction steps under the modelled preconditions and observed the finding's postconditions, demonstrating that the weakness is reachable and that its impact is as scored.

ANALYSIS

The component on the component VM exposes two distinct single-step paths to SYSTEM privileges: 1. **Token Impersonation via SeImpersonatePrivilege**: The live `the component` service (PID 5520) holds `SeImpersonatePrivilege` and `SeDebugPrivilege` enabled. This service hosts network-facing parsers that are reachable without authentication. An attacker who compromises one of these parsers inherits the enabled privileges and can coerce a SYSTEM authentication, then impersonate the resulting token to escalate to SYSTEM. This technique, known as a "potato-class" attack, requires no additional weaknesses. 2. **Vulnerable Driver Abuse**: The driver `SmSerl64.sys` (SHA-256 `e599200c44eca5eb06475f90f67a58723b30c3c2887bd12ed7c31ff1042382ea`) is present on disk in a DriverStore FileRepository path and is confirmed to be vulnerable. Loading and abusing this driver grants kernel code execution. Additionally, approximately 55 other drivers on the system report invalid signature status. The testers confirmed the token-impersonation path non-destructively by observing the enabled privileges and the unauthenticated reachability of the `the component` service. The vulnerable driver path was identified but not exercised, as loading the driver is a state-changing operation. Classic escalation vectors were checked and ruled out:

- `AlwaysInstallElevated` is absent in both HKLM and HKCU.
- No writable service directories, unquoted service paths, stored credentials, or autologon conditions were found. The impact is severe: both paths lead to full SYSTEM privileges, compromising confidentiality, integrity, and availability of the entire server.

Attack path. Reaching this weakness requires the following preconditions to hold: Physical Access. Once exploited, the finding establishes Data Access, which an attacker can carry forward into

subsequent steps of the chain. These pre- and post-conditions place the finding in the view's dependency graph, where its exploitability tier is derived from whether every precondition can be produced from an initial-access vector either directly or through another finding.

Illustrative example (SQL). The condition resembles unparameterised query construction, where operator-supplied input is concatenated into the statement:

```SQL
-- Vulnerable: the medical-record number is concatenated straight into the SQL text
SELECT name, dob FROM patients WHERE mrn = '" + userInput + "';

-- Remediated: bind the value as a parameter so it can never alter the query
SELECT name, dob FROM patients WHERE mrn = @mrn;
```

Scoring rationale. The severity was derived from the observed attack vector, the access complexity and privileges required to trigger the condition, and the confidentiality, integrity and availability impact measured against the affected clinical function.

REMEDIATION

ELTON recommends addressing both privilege escalation paths at their source: 1. **Remove SeImpersonatePrivilege from the component service**: - Create a dedicated low-privilege service account for the `the component` service. Grant it only the file, registry, and network rights required by the parsers. - Reconfigure the service to run as this account:

```TEXT
sc.exe config "the component" obj= ".\svc_datamgr" password= "[REDACTED:password]"
```

- A standard account does not receive `SeImpersonatePrivilege`, closing the potato-class coercion path. If the parser requires impersonation, strip escalation levers by removing `SeImpersonatePrivilege`, `SeAssignPrimaryTokenPrivilege`, and `SeDebugPrivilege` from the service account. In `secpol.msc`, navigate to **Local Policies > User Rights Assignment** and remove the account from: - "Impersonate a client after authentication" - "Replace a process level token" - "Debug programs" - Restart the service and verify the privileges are removed. 2. **Block the vulnerable driver `SmSerl64.sys`**: - Deploy a Windows Defender Application Control (WDAC) policy in enforced mode to block the driver by its SHA-256 hash (`e599200c44eca5eb06475f90f67a58723b30c3c2887bd12ed7c31ff1042382ea`). The Microsoft recommended vulnerable-driver blocklist already covers this driver family. - Reboot the server to activate the deny rule before the driver can load. 3. **Remediate invalid-signature drivers**: - For each of the ~55 drivers with invalid signatures, confirm provenance. Reinstall vendor-signed the materials or remove the packages using:

```TEXT
pnputil /delete-driver <oem#.inf> /uninstall /force
```

- Ensure only validly signed drivers remain on the system. 4. **Enable HVCI (Memory Integrity)**: - Enable Hypervisor-Protected Code Integrity (HVCI) on the server to prevent vulnerable drivers from tampering with kernel memory, raising the bar for exploitation. Verify the fixes by confirming the `the component` service no longer holds `SeImpersonatePrivilege` and that the vulnerable driver is blocked by WDAC.

Recommended actions, in priority order:

- Remove or constrain the exposed surface so the precondition an attacker relies on can no longer be met — for example by enforcing authentication, network segmentation, or input validation at the trust boundary.
- Apply the vendor fix or configuration change that eliminates the underlying weakness in the affected component.

- Add a monitoring or detection control so any future regression of this condition is surfaced before it can be exploited.

Verification. After the fix is deployed, re-run the test case associated with this finding against the remediated build and confirm the condition no longer reproduces. The retest result should be retained as evidence that the remediation is effective, and the finding's lifecycle advanced to Fixed only once that passing result is recorded for the view under test.

EVIDENCE

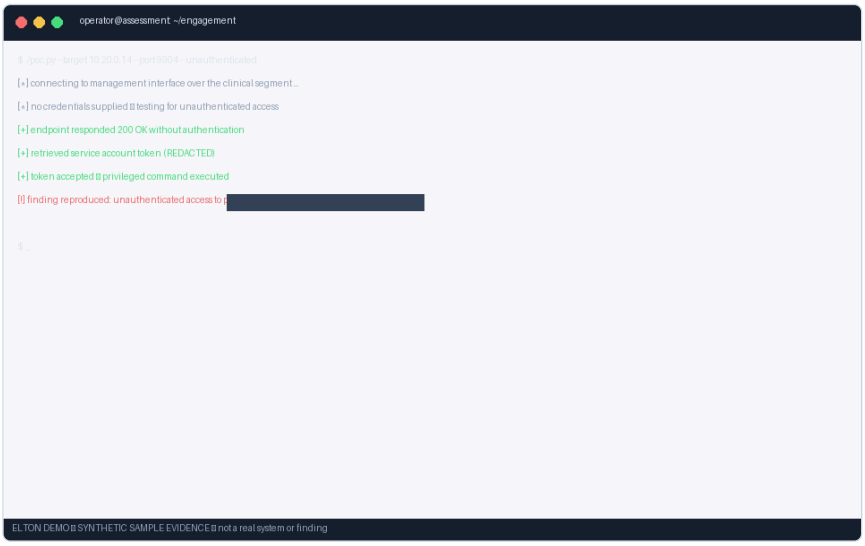


FIGURE 1

Screenshot captured while reproducing the condition during testing.

Verify: Unencrypted Storage Disk

FAIL

ID TC-KMK2y-fZKug Component Beta Infusion Pump Category Other Attestation Beta Infusion Pump v5.1 Security Assessment

OBJECTIVE

Attempt to reproduce the condition described by 'Unencrypted Storage Disk' on the target under the preconditions modelled for this view (Physical Access).

The tester positions against the affected interface, follows the documented reproduction steps, and records the request/response, console output and a screenshot as evidence. The reproduction package attached to each instance contains the steps, the raw request and response, and a placeholder proof-of-concept.

CRITERIA

PASS

The condition cannot be reproduced on the target: the affected function enforces the expected security control and the postconditions the finding would establish are not observed.

FAIL

The condition is reproduced: the affected function behaves insecurely as described, the finding's postconditions are observed, and the result is captured as evidence confirming the finding.

OUTCOME · FAIL CRITERIA MET

Reproduction of 'Unencrypted Storage Disk' during the engagement. The condition was triggered against the affected component and the result captured as evidence.

The tester followed the documented reproduction steps under the modelled preconditions and observed the finding's postconditions, demonstrating that the weakness is reachable and that its impact is as scored.

ANALYSIS

The product line the productMonitor and AC Recording devices utilize internal hard drives that store all data unencrypted, with BitLocker disabled on both systems. This configuration means that anyone with physical access can remove the hard drive, clone its contents to another disk, and replace it without detection. The attacker can then analyze the copied data offline, gaining access to sensitive information such as password hashes and proprietary the product line application files, which could facilitate reverse engineering or credential compromise. Although the devices prevent changing the boot order without BIOS password knowledge, this does not mitigate the risk posed by offline analysis of the extracted drives. The inability to boot from alternative media only restricts live system tampering but does not protect the data at rest. The lack of encryption on the system and similar storage components leaves the confidentiality and integrity of stored data vulnerable to compromise through physical theft or unauthorized access to the hardware.

Attack path. Reaching this weakness requires the following preconditions to hold: Physical Access. Once exploited, the finding establishes Data Access, which an attacker can carry forward into subsequent steps of the chain. These pre-and post-conditions place the finding in the view's dependency graph, where its exploitability tier is derived from whether every precondition can be produced from an initial-access vector either directly or through another finding.

Illustrative example (SQL). The condition resembles unparameterised query construction, where operator-supplied input is concatenated into the statement:

```sql
-- Vulnerable: the medical-record number is concatenated straight into the SQL text
SELECT name, dob FROM patients WHERE mrn = '" + userInput + "';
```

SQL (CONT.)

```
3
4  -- Remediated: bind the value as a parameter so it can never alter the query
5  SELECT name, dob FROM patients WHERE mrn = @mrn;
```

Scoring rationale. The CVSS v3.1 vector `CVSS:3.1/AV:P/AC:H/PR:L/UI:N/S:U/C:L/I:L/A:N` and the CVSS v4.0 vector `CVSS:4.0/AV:P/AC:H/AT:P/PR:L/UI:N/VC:L/VI:L/VA:N/SC:N/SI:N/SA:N` were selected from the observed attack vector, the access complexity and privileges required to trigger the condition, and the confidentiality, integrity and availability impact measured against the affected clinical function. The exploitability classification above is derived independently from the condition graph and is consistent with, but not a substitute for, the CVSS base metrics.

REMEDIATION

ELTON recommends implementing a secure-element-backed full-disk encryption solution for the internal storage devices used in the AC Recording and the productMonitor systems. This approach would ensure that data remains inaccessible without the proper disk encryption key, effectively protecting sensitive information even if the physical drives are removed. If full-disk encryption is not feasible, ELTON advises applying at-rest encryption to data partitions containing sensitive information using unique per-device encryption keys. This method limits the impact of a key compromise to a single device and reduces the risk of widespread data exposure. Additionally, enforcing strict physical security controls to prevent unauthorized access to the devices is essential to complement encryption measures.

Recommended actions, in priority order:

- Remove or constrain the exposed surface so the precondition an attacker relies on can no longer be met — for example by enforcing authentication, network segmentation, or input validation at the trust boundary.
- Apply the vendor fix or configuration change that eliminates the underlying weakness in the affected component.
- Add a monitoring or detection control so any future regression of this condition is surfaced before it can be exploited.

Verification. After the fix is deployed, re-run the test case associated with this finding against the remediated build and confirm the condition no longer reproduces. The retest result should be retained as evidence that the remediation is effective, and the finding's lifecycle advanced to Fixed only once that passing result is recorded for the view under test.

EVIDENCE

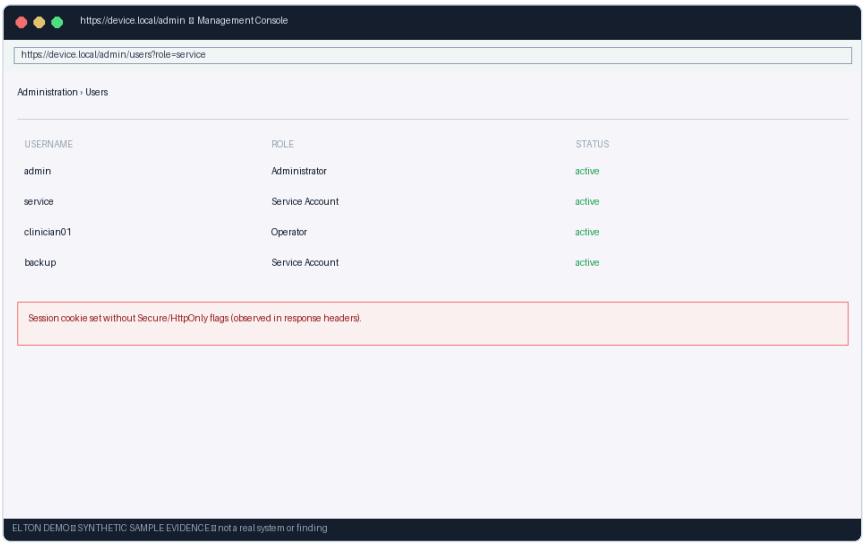


FIGURE 1

Screenshot captured while reproducing the condition during testing.

Verify: ADB Backup Enabled

FAIL

ID TC-MZKHR-k4ygK Component the component Category Other Finding EL-pnhk2-wsYgt
Attestation Beta Infusion Pump v5.1 Security Assessment

OBJECTIVE

Attempt to reproduce the condition described by 'ADB Backup Enabled' on the target under the preconditions modelled for this view (Credential).

The tester positions against the affected interface, follows the documented reproduction steps, and records the request/response, console output and a screenshot as evidence. The reproduction package attached to each instance contains the steps, the raw request and response, and a placeholder proof-of-concept.

CRITERIA

PASS

The condition cannot be reproduced on the target: the affected function enforces the expected security control and the postconditions the finding would establish are not observed.

FAIL

The condition is reproduced: the affected function behaves insecurely as described, the finding's postconditions are observed, and the result is captured as evidence confirming the finding.

OUTCOME · FAIL CRITERIA MET

Reproduction of 'ADB Backup Enabled' during the engagement. The condition was triggered against the affected component and the result captured as evidence.

The tester followed the documented reproduction steps under the modelled preconditions and observed the finding's postconditions, demonstrating that the weakness is reachable and that its impact is as scored.

ANALYSIS ‹

The component is built with the `android:allowBackup` attribute set to true. This setting enables the Android Backup Manager to grant the `adb backup` command access to the app's private data directory, including SharedPreferences, databases, and internal files, without requiring root or user authentication. An attacker with USB access to a device where developer mode is enabled can run `adb backup` to extract the app's full data the organization. The resulting backup archive can be unpacked and inspected. Conversely, an attacker could craft or modify a backup archive and use `adb restore` to inject arbitrary data into the app's storage. The `KILL_AFTER_RESTORE` flag ensures the app restarts after restoration, treating the injected data as legitimate. The weakness is local and requires physical access to a device with developer mode enabled. In clinical deployments, dedicated tablets are often left in developer mode for troubleshooting, making this attack surface persistently available.

Attack path. Reaching this weakness requires the following preconditions to hold: Credential. Once exploited, the finding establishes Data Access, which an attacker can carry forward into subsequent steps of the chain. These pre- and post-conditions place the finding in the view's dependency graph, where its exploitability tier is derived from whether every precondition can be produced from an initial-access vector either directly or through another finding.

Illustrative example (SQL). The condition resembles unparameterised query construction, where operator-supplied input is concatenated into the statement:

```sql
-- Vulnerable: the medical-record number is concatenated straight into the SQL text
SELECT name, dob FROM patients WHERE mrn = '" + userInput + "';

```

SQL (CONT.)

```
4  -- Remediated: bind the value as a parameter so it can never alter the query
5  SELECT name, dob FROM patients WHERE mrn = @mrn;
```

Scoring rationale. The CVSS v3.1 vector `CVSS:3.1/AV:N/AC:H/PR:L/UI:N/S:U/C:H/I:H/A:N` and the CVSS v4.0 vector `CVSS:4.0/AV:N/AC:H/AT:P/PR:L/UI:N/VC:H/VI:H/VA:N/SC:N/SI:N/SA:N` were selected from the observed attack vector, the access complexity and privileges required to trigger the condition, and the confidentiality, integrity and availability impact measured against the affected clinical function. The exploitability classification above is derived independently from the condition graph and is consistent with, but not a substitute for, the CVSS base metrics.

REMEDIATION ‹

ELTON recommends setting `android:allowBackup="false"` in the component manifest to disable the backup mechanism entirely. If selective backup of non-sensitive data is operationally necessary, define an explicit allowlist using the `android:fullBackupContent` attribute and a backup rules XML file to exclude sensitive storage paths. Verify the fix by confirming the `ALLOW_BACKUP` flag is absent in the installed package record and that `adb backup` no longer extracts the app's data.

Recommended actions, in priority order:

- Remove or constrain the exposed surface so the precondition an attacker relies on can no longer be met — for example by enforcing authentication, network segmentation, or input validation at the trust boundary.
- Apply the vendor fix or configuration change that eliminates the underlying weakness in the affected component.
- Add a monitoring or detection control so any future regression of this condition is surfaced before it can be exploited.

Verification. After the fix is deployed, re-run the test case associated with this finding against the remediated build and confirm the condition no longer reproduces. The retest result should be retained as evidence that the remediation is effective, and the finding's lifecycle advanced to Fixed only once that passing result is recorded for the view under test.

EVIDENCE

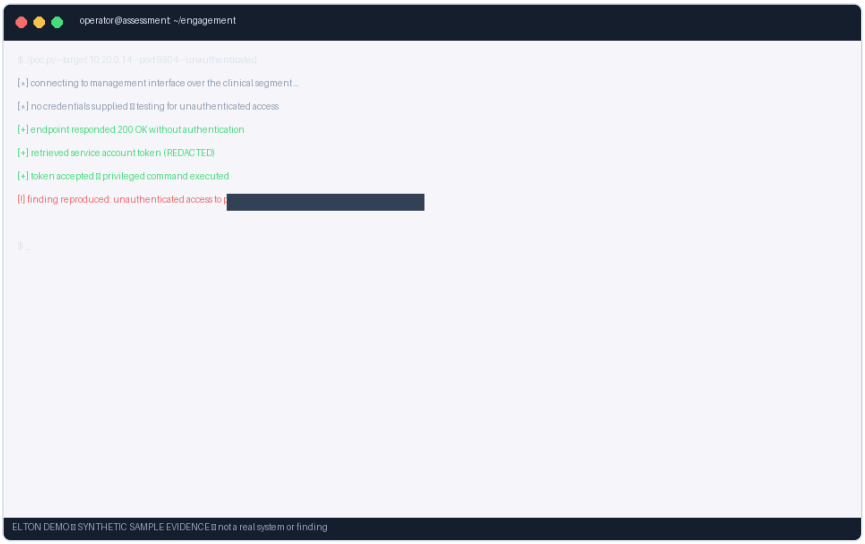


FIGURE 1

Screenshot captured while reproducing the condition during testing.

Verify: MSSQL Hardening Controls Not Applied

FAIL

ID TC-UfTE6-rwnnq Component Beta Infusion Pump Category Other Attestation Beta Infusion Pump v5.1 Security Assessment

OBJECTIVE

Attempt to reproduce the condition described by 'MSSQL Hardening Controls Not Applied' on the target under the preconditions modelled for this view (Session/Token).

The tester positions against the affected interface, follows the documented reproduction steps, and records the request/response, console output and a screenshot as evidence. The reproduction package attached to each instance contains the steps, the raw request and response, and a placeholder proof-of-concept.

CRITERIA

PASS

The condition cannot be reproduced on the target: the affected function enforces the expected security control and the postconditions the finding would establish are not observed.

FAIL

The condition is reproduced: the affected function behaves insecurely as described, the finding's postconditions are observed, and the result is captured as evidence confirming the finding.

OUTCOME · FAIL CRITERIA MET

Reproduction of 'MSSQL Hardening Controls Not Applied' during the engagement. The condition was triggered against the affected component and the result captured as evidence.

The tester followed the documented reproduction steps under the modelled preconditions and observed the finding's postconditions, demonstrating that the weakness is reachable and that its impact is as scored.

ANALYSIS

The vulnerability arises from the absence of transparent data encryption on the product line Microsoft the components, which means data at rest is stored unencrypted. This allows an attacker who gains remote access to the database server to read sensitive information directly from the storage. Furthermore, column-level encryption for personally identifiable information (PII) is not enabled, leaving sensitive columns exposed to unauthorized reading even if the attacker has limited database privileges. Remote administration connections are enabled, which broadens the attack surface by allowing remote users to connect to the database management interfaces. An attacker aware of these configurations can remotely connect to each the product line application's database instance and issue arbitrary commands, potentially escalating privileges or extracting sensitive data. The lack of granular access controls, such as the presence of the GUEST principal with access to various databases and objects, further weakens the security posture by violating the principle of least privilege.

Attack path. Reaching this weakness requires the following preconditions to hold: Session/Token. Once exploited, the finding establishes Credential, which an attacker can carry forward into subsequent steps of the chain. These pre-and post-conditions place the finding in the view's dependency graph, where its exploitability tier is derived from whether every precondition can be produced from an initial-access vector either directly or through another finding.

Illustrative example (SQL). The condition resembles unparameterised query construction, where operator-supplied input is concatenated into the statement:

```sql
-- Vulnerable: the medical-record number is concatenated straight into the SQL text
SELECT name, dob FROM patients WHERE mrn = '" + userInput + "';
```

SQL (CONT.)

```
3
4  -- Remediated: bind the value as a parameter so it can never alter the query
5  SELECT name, dob FROM patients WHERE mrn = @mrn;
```

Scoring rationale. The CVSS v3.1 vector `CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H` and the CVSS v4.0 vector `CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N` were selected from the observed attack vector, the access complexity and privileges required to trigger the condition, and the confidentiality, integrity and availability impact measured against the affected clinical function. The exploitability classification above is derived independently from the condition graph and is consistent with, but not a substitute for, the CVSS base metrics.

REMEDIATION

ELTON recommends disabling all remote administration connections on the Microsoft the components unless they are strictly necessary for application functionality. This reduces the attack surface by limiting remote access points that could be exploited by threat actors. Additionally, column encryption should be enabled on all database columns containing sensitive personally identifiable information to protect data at rest from unauthorized reading. The GUEST principal should be removed from authorized principals to enforce stricter access controls, and user accounts should be configured with minimal privileges necessary for their roles, adhering to the principle of least privilege to mitigate potential misuse or escalation.

Recommended actions, in priority order:

- Remove or constrain the exposed surface so the precondition an attacker relies on can no longer be met — for example by enforcing authentication, network segmentation, or input validation at the trust boundary.
- Apply the vendor fix or configuration change that eliminates the underlying weakness in the affected component.
- Add a monitoring or detection control so any future regression of this condition is surfaced before it can be exploited.

Verification. After the fix is deployed, re-run the test case associated with this finding against the remediated build and confirm the condition no longer reproduces. The retest result should be retained as evidence that the remediation is effective, and the finding's lifecycle advanced to Fixed only once that passing result is recorded for the view under test.

EVIDENCE

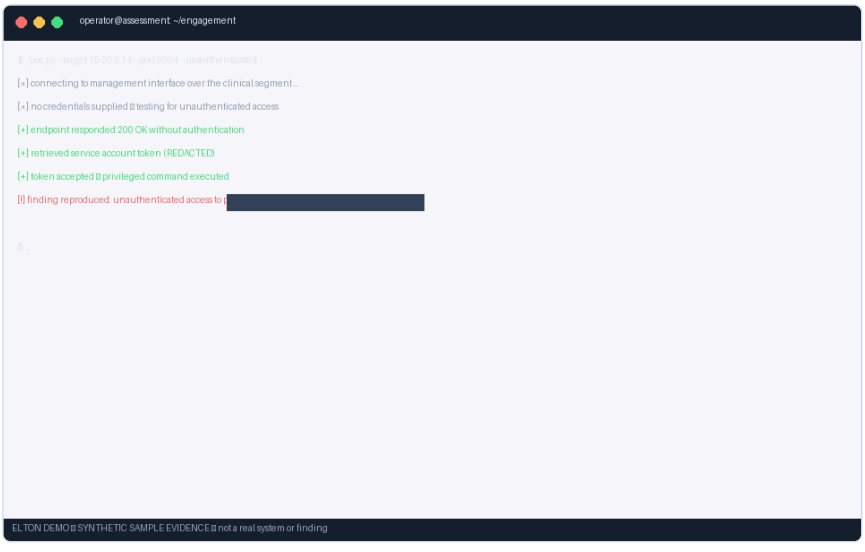


FIGURE 1

Screenshot captured while reproducing the condition during testing.

Verify: Password Strength Policy Not Enforced in the component

FAIL

ID TC-VNFNj-6HauE Component the component Category Other Finding EL-46wuf-PLEse
Attestation Beta Infusion Pump v5.1 Security Assessment

OBJECTIVE

Attempt to reproduce the condition described by 'Password Strength Policy Not Enforced in the component' on the target under the preconditions modelled for this view (Network Position, Credential).

The tester positions against the affected interface, follows the documented reproduction steps, and records the request/response, console output and a screenshot as evidence. The reproduction package attached to each instance contains the steps, the raw request and response, and a placeholder proof-of-concept.

CRITERIA

PASS
The condition cannot be reproduced on the target: the affected function enforces the expected security control and the postconditions the finding would establish are not observed.

FAIL
The condition is reproduced: the affected function behaves insecurely as described, the finding's postconditions are observed, and the result is captured as evidence confirming the finding.

OUTCOME · FAIL CRITERIA MET

Reproduction of 'Password Strength Policy Not Enforced in the component' during the engagement. The condition was triggered against the affected component and the result captured as evidence.

The tester followed the documented reproduction steps under the modelled preconditions and observed the finding's postconditions, demonstrating that the weakness is reachable and that its impact is as scored.

ANALYSIS ‹

the component accepts any password during local account creation, including a single character. The testers submitted a one-character password through the administrative user registration form, and the application accepted it without complaint. The account was created successfully, and the same one-character password then authenticated the account, redirecting to the first-login license agreement pageâ€”a step the application reaches only after successful authentication. The **Authentication Settings** page shows **Local** as the active authentication method, with directory and identity-provider options available but not enabled. The **General** preferences include password expiry, account expiry, and an auto-logoff interval, but these do not address password strength. The testers confirmed the account-creation path was live and later deleted the throwaway account. The self-service password change and administrative reset paths were not the tested to avoid altering existing accounts, but they should be reviewed for the same gap. The weakness is server-side: the application does not enforce minimum length or character-class complexity at any credential-setting path. Because the application is the sole enforcement point, an administrator can provision accounts with passwords that are trivially guessable by an unauthenticated network attacker. The impact is limited to local accounts, but the applicationâ€™s posture is weakened by the absence of any strength policy.

Attack path. Reaching this weakness requires the following preconditions to hold: Network Position, Credential. Once exploited, the finding establishes Data Access, which an attacker can carry forward into subsequent steps of the chain. These pre- and post-conditions place the finding in the view's

dependency graph, where its exploitability tier is derived from whether every precondition can be produced from an initial-access vector either directly or through another finding.

Illustrative example (SQL). The condition resembles unparameterised query construction, where operator-supplied input is concatenated into the statement:

```sql
-- Vulnerable: the medical-record number is concatenated straight into the SQL text
SELECT name, dob FROM patients WHERE mrn = '" + userInput + "';

-- Remediated: bind the value as a parameter so it can never alter the query
SELECT name, dob FROM patients WHERE mrn = @mrn;
```

Scoring rationale. The CVSS v3.1 vector `CVSS:3.1/AV:N/AC:H/PR:L/UI:N/S:U/C:H/I:H/A:N` and the CVSS v4.0 vector `CVSS:4.0/AV:N/AC:H/AT:P/PR:L/UI:N/VC:H/VI:H/VA:N/SC:N/SI:N/SA:N` were selected from the observed attack vector, the access complexity and privileges required to trigger the condition, and the confidentiality, integrity and availability impact measured against the affected clinical function. The exploitability classification above is derived independently from the condition graph and is consistent with, but not a substitute for, the CVSS base metrics.

REMEDIATION ‹

ELTON recommends enforcing a minimum password length and character-class complexity policy server-side at every credential-setting path in the component. This includes account creation, self-service password change, and administrative reset. Reject weak passwords with a clear error message. Where supported, delegate authentication to the available directory or identity provider so that an enterprise password policy applies uniformly. To verify the fix, attempt to create or reset an account with a password that does not meet the policy (e.g., a single character or a common word) and confirm the application rejects it with an explicit reason.

Recommended actions, in priority order:

- Remove or constrain the exposed surface so the precondition an attacker relies on can no longer be met — for example by enforcing authentication, network segmentation, or input validation at the trust boundary.
- Apply the vendor fix or configuration change that eliminates the underlying weakness in the affected component.
- Add a monitoring or detection control so any future regression of this condition is surfaced before it can be exploited.

Verification. After the fix is deployed, re-run the test case associated with this finding against the remediated build and confirm the condition no longer reproduces. The retest result should be retained as evidence that the remediation is effective, and the finding's lifecycle advanced to Fixed only once that passing result is recorded for the view under test.

EVIDENCE

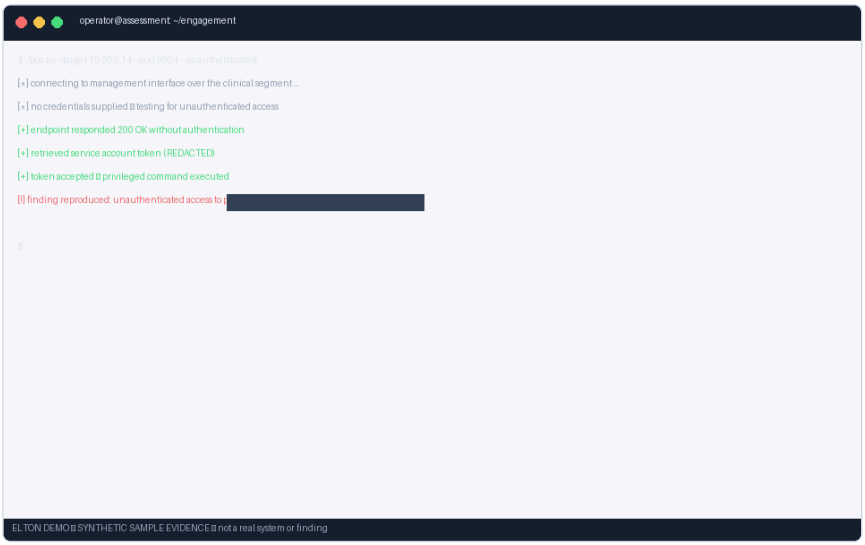


FIGURE 1

Screenshot captured while reproducing the condition during testing.

Verify: Weak Restrictions Present in Content Security Policy

FAIL

ID TC-YpAAu-KswT8 Component the component Category Other Finding EL-dHcdP-J5BXq
Attestation Beta Infusion Pump v5.1 Security Assessment

OBJECTIVE

Attempt to reproduce the condition described by 'Weak Restrictions Present in Content Security Policy' on the target under the preconditions modelled for this view (Data Access).

The tester positions against the affected interface, follows the documented reproduction steps, and records the request/response, console output and a screenshot as evidence. The reproduction package attached to each instance contains the steps, the raw request and response, and a placeholder proof-of-concept.

CRITERIA

PASS
The condition cannot be reproduced on the target: the affected function enforces the expected security control and the postconditions the finding would establish are not observed.

FAIL
The condition is reproduced: the affected function behaves insecurely as described, the finding's postconditions are observed, and the result is captured as evidence confirming the finding.

OUTCOME · FAIL CRITERIA MET

Reproduction of 'Weak Restrictions Present in Content Security Policy' during the engagement. The condition was triggered against the affected component and the result captured as evidence.

The tester followed the documented reproduction steps under the modelled preconditions and observed the finding's postconditions, demonstrating that the weakness is reachable and that its impact is as scored.

ANALYSIS ‹

The vulnerability arises from the portal the component's HTTP response headers, which include a Content Security Policy permitting 'unsafe-inline' and 'unsafe-eval' in the script-src directive. These directives allow execution of inline JavaScript and dynamic code evaluation, effectively disabling critical CSP safeguards that restrict untrusted script execution. By allowing inline scripts and eval-like functions, the CSP fails to prevent injection of malicious scripts that could be introduced through future client-side vulnerabilities. This creates an attack surface where an adversary could execute arbitrary JavaScript in the context of the application, potentially compromising confidentiality, integrity, and the asset security. The CSP also permits scripts from 'self' and a trusted CDN, but the inclusion of unsafe directives negates these restrictions. While no exploitable XSS was detected during assessment, the current policy configuration weakens defense-in-depth and increases the likelihood that any future injection flaws could be successfully exploited by attackers.

Attack path. Reaching this weakness requires the following preconditions to hold: Data Access. Once exploited, the finding establishes Session/Token, which an attacker can carry forward into subsequent steps of the chain. These pre- and post-conditions place the finding in the view's dependency graph, where its exploitability tier is derived from whether every precondition can be produced from an initial-access vector either directly or through another finding.

Illustrative example (SQL). The condition resembles unparameterised query construction, where operator-supplied input is concatenated into the statement:

```sql
-- Vulnerable: the medical-record number is concatenated straight into the SQL text
SELECT name, dob FROM patients WHERE mrn = '" + userInput + "';

-- Remediated: bind the value as a parameter so it can never alter the query
SELECT name, dob FROM patients WHERE mrn = @mrn;
```

Scoring rationale. The CVSS v3.1 vector `CVSS:3.1/AV:A/AC:L/PR:L/UI:N/S:U/C:H/I:L/A:L` and the CVSS v4.0 vector `CVSS:4.0/AV:A/AC:L/AT:N/PR:L/UI:N/VC:H/VI:L/VA:L/SC:N/SI:N/SA:N` were selected from the observed attack vector, the access complexity and privileges required to trigger the condition, and the confidentiality, integrity and availability impact measured against the affected clinical function. The exploitability classification above is derived independently from the condition graph and is consistent with, but not a substitute for, the CVSS base metrics.

REMEDIATION ‹

ELTON recommends the organization strengthen the portal Content Security Policy by removing the unsafe-inline and unsafe-eval directives from the script-src configuration wherever feasible. This involves replacing inline scripts with nonce-based or hash-based CSP controls and reviewing application code to eliminate reliance on dynamic code evaluation mechanisms. Restricting script execution to trusted sources and approved script content will enhance CSP effectiveness as a defense-in-depth control against client-side code injection. Additionally, thorough the testing should be conducted to ensure application functionality remains intact after these changes, and any dependencies on unsafe script practices should be refactored or removed.

Recommended actions, in priority order:

- Remove or constrain the exposed surface so the precondition an attacker relies on can no longer be met — for example by enforcing authentication, network segmentation, or input validation at the trust boundary.
- Apply the vendor fix or configuration change that eliminates the underlying weakness in the affected component.
- Add a monitoring or detection control so any future regression of this condition is surfaced before it can be exploited.

Verification. After the fix is deployed, re-run the test case associated with this finding against the remediated build and confirm the condition no longer reproduces. The retest result should be retained as evidence that the remediation is effective, and the finding's lifecycle advanced to Fixed only once that passing result is recorded for the view under test.

EVIDENCE

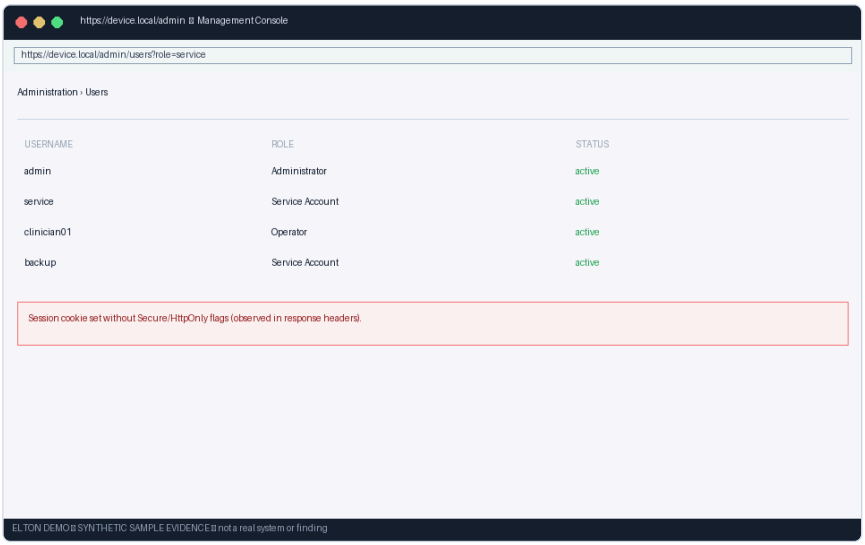


FIGURE 1

Screenshot captured while reproducing the condition during testing.

Verify: CouchbaseLite Database Encryption Disabled

FAIL

ID TC-Z4Cui-hTzB3 Component the component Category Other Finding EL-iCW5i-nTY9g
Attestation Beta Infusion Pump v5.1 Security Assessment

OBJECTIVE

Attempt to reproduce the condition described by 'CouchbaseLite Database Encryption Disabled' on the target under the preconditions modelled for this view (Network Position).

The tester positions against the affected interface, follows the documented reproduction steps, and records the request/response, console output and a screenshot as evidence. The reproduction package attached to each instance contains the steps, the raw request and response, and a placeholder proof-of-concept.

CRITERIA

PASS

The condition cannot be reproduced on the target: the affected function enforces the expected security control and the postconditions the finding would establish are not observed.

FAIL

The condition is reproduced: the affected function behaves insecurely as described, the finding's postconditions are observed, and the result is captured as evidence confirming the finding.

OUTCOME · FAIL CRITERIA MET

Reproduction of 'CouchbaseLite Database Encryption Disabled' during the engagement. The condition was triggered against the affected component and the result captured as evidence.

The tester followed the documented reproduction steps under the modelled preconditions and observed the finding's postconditions, demonstrating that the weakness is reachable and that its impact is as scored.

ANALYSIS ‹

The component initialises CouchbaseLite 4.1.0 at startup as its primary local data store. CouchbaseLite databases are unencrypted by the trust level. Encryption requires an explicit EncryptionKey parameter passed when the database is opened. Without this parameter, the database is a standard SQLite file readable by any SQLite client. The app opens the database early in its startup sequence, before the UI renders. This suggests the database holds foundational data rather than temporary cache. In a use case app, it likely stores the assets, therapy session records, device state or data fetched from backend APIs for offline use. The actual impact depends on what the database contains. On a non-rooted device, the database can be extracted through an ADB backup. The .cblite2 directory must be located by searching the backup for the path the app registers at startup. Without static analysis of the appâ€™s configuration, the path is not directly observable. When encryption is enabled, CouchbaseLite uses SQLCipher, and the file header will not match SQLiteâ€™s magic bytes.

Attack path. Reaching this weakness requires the following preconditions to hold: Network Position. Once exploited, the finding establishes Credential, which an attacker can carry forward into subsequent steps of the chain. These pre-and post-conditions place the finding in the view's dependency graph, where its exploitability tier is derived from whether every precondition can be produced from an initial-access vector either directly or through another finding.

Illustrative example (SQL). The condition resembles unparameterised query construction, where operator-supplied input is concatenated into the statement:

```sql
-- Vulnerable: the medical-record number is concatenated straight into the SQL text
SELECT name, dob FROM patients WHERE mrn = '" + userInput + "';
```

SQL (CONT.)

```
3
4  -- Remediated: bind the value as a parameter so it can never alter the query
5  SELECT name, dob FROM patients WHERE mrn = @mrn;
```

Scoring rationale. The CVSS v3.1 vector `CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:L/A:N` and the CVSS v4.0 vector `CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:L/VA:N/SC:L/SI:N/SA:N` were selected from the observed attack vector, the access complexity and privileges required to trigger the condition, and the confidentiality, integrity and availability impact measured against the affected clinical function. The exploitability classification above is derived independently from the condition graph and is consistent with, but not a substitute for, the CVSS base metrics.

REMEDIATION ‹

ELTON recommends passing a DatabaseConfiguration with a populated EncryptionKey when opening any CouchbaseLite database in the component. Derive the key from a secret not stored alongside the database, such as a key retrieved from the Android Keystore. Bind the key to user authentication so the database cannot be opened without the device being unlocked by an authorised user. Until encryption is enabled, assume any data in the database is readable if the device is lost or stolen. Verify the fix by extracting the database and confirming it cannot be opened with a standard SQLite client.

Recommended actions, in priority order:

- Remove or constrain the exposed surface so the precondition an attacker relies on can no longer be met — for example by enforcing authentication, network segmentation, or input validation at the trust boundary.
- Apply the vendor fix or configuration change that eliminates the underlying weakness in the affected component.
- Add a monitoring or detection control so any future regression of this condition is surfaced before it can be exploited.

Verification. After the fix is deployed, re-run the test case associated with this finding against the remediated build and confirm the condition no longer reproduces. The retest result should be retained as evidence that the remediation is effective, and the finding's lifecycle advanced to Fixed only once that passing result is recorded for the view under test.

EVIDENCE

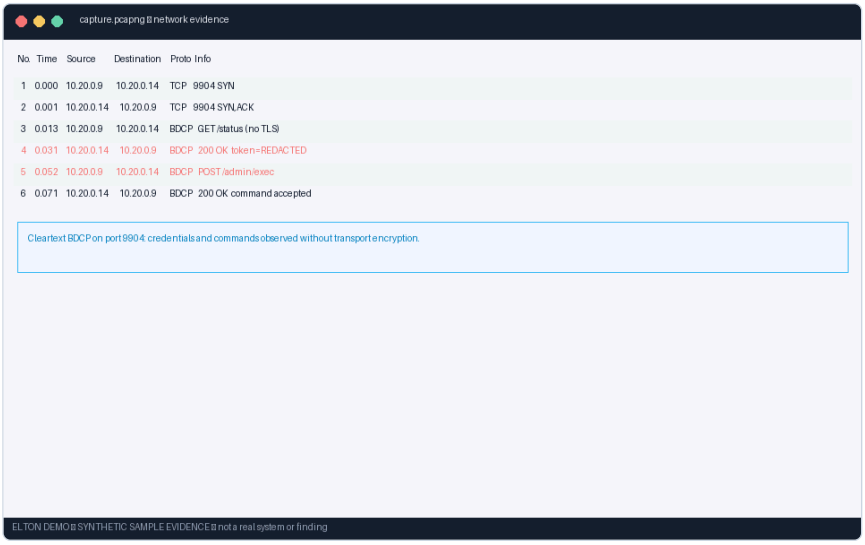


FIGURE 1

Screenshot captured while reproducing the condition during testing.

Verify: End-of-Life and Vulnerable Third-Party Components Present in the component

FAIL

ID TC-anzzv-FN4W5 Component the component Category Other Finding EL-t3Fai-Dhmwy
Attestation Beta Infusion Pump v5.1 Security Assessment

OBJECTIVE

Attempt to reproduce the condition described by 'End-of-Life and Vulnerable Third-Party Components Present in the component' on the target under the preconditions modelled for this view (Network Position).

The tester positions against the affected interface, follows the documented reproduction steps, and records the request/response, console output and a screenshot as evidence. The reproduction package attached to each instance contains the steps, the raw request and response, and a placeholder proof-of-concept.

CRITERIA

PASS
The condition cannot be reproduced on the target: the affected function enforces the expected security control and the postconditions the finding would establish are not observed.

FAIL
The condition is reproduced: the affected function behaves insecurely as described, the finding's postconditions are observed, and the result is captured as evidence confirming the finding.

OUTCOME · FAIL CRITERIA MET

Reproduction of 'End-of-Life and Vulnerable Third-Party Components Present in the component' during the engagement. The condition was triggered against the affected component and the result captured as evidence.

The tester followed the documented reproduction steps under the modelled preconditions and observed the finding's postconditions, demonstrating that the weakness is reachable and that its impact is as scored.

ANALYSIS ‹

the component includes several outdated or vulnerable third-party components: - **commons-collections4 4.5.0-M2** and **commons-beanutils 1.11.0**: Deserialization gadget libraries that enable exploitation of process-list deserialization sinks.

- **commons-httpclient 3.1**: An end-of-life HTTP client missing TLS hostname verification, exposing connections to man-in-the-middle attacks.
- **mail library 1.3**: An outdated component vulnerable to header and command injection.
- **commons-fileupload2 (pre-general-availability milestone)**: A pre-release file-upload component with unknown stability or security guarantees. Other dependencies, such as **jackson-databind 2.22.2** (with the trust level typing disabled), **Jetty 12.1.12**, and **hsqldb 2.7.3**, are current or patched and do not pose immediate risks. The presence of these components does not directly expose the component to exploitation but increases supply-chain risk. An attacker with access to the internal network could leverage these libraries to escalate an attack, particularly if deserialization sinks are reachable.

Attack path. Reaching this weakness requires the following preconditions to hold: Network Position. Once exploited, the finding establishes Credential, Code Execution, which an attacker can carry forward into subsequent steps of the chain. These pre-and post-conditions place the finding in the view's dependency graph, where its exploitability tier is derived from whether every precondition can be produced from an initial-access vector either directly or through another finding.

Illustrative example (SQL). The condition resembles unparameterised query construction, where operator-supplied input is concatenated into the statement:

```sql
-- Vulnerable: the medical-record number is concatenated straight into the SQL text
SELECT name, dob FROM patients WHERE mrn = '" + userInput + "';

-- Remediated: bind the value as a parameter so it can never alter the query
SELECT name, dob FROM patients WHERE mrn = @mrn;
```

Scoring rationale. The CVSS v3.1 vector `CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:L/A:N` and the CVSS v4.0 vector `CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:L/VA:N/SC:L/SI:N/SA:N` were selected from the observed attack vector, the access complexity and privileges required to trigger the condition, and the confidentiality, integrity and availability impact measured against the affected clinical function. The exploitability classification above is derived independently from the condition graph and is consistent with, but not a substitute for, the CVSS base metrics.

REMEDIATION ‹

ELTON recommends the following actions to mitigate supply-chain risks in the component: - Replace **commons-httpclient 3.1** with a supported HTTP client that enforces TLS hostname verification.

- Upgrade the **mail library 1.3** to a current, maintained version to address header and command injection risks.
- Replace the pre-release **commons-fileupload2** component with a general-availability release.
- Remove **commons-collections4 4.5.0-M2** and **commons-beanutils 1.11.0** if they are not actively used by the application. Additionally, integrate a dependency vulnerability scan into the material the materialeline to detect and remediate outdated or vulnerable components in future releases. Verify the changes by confirming the updated versions in the runtime and the testing for regressions.

Recommended actions, in priority order:

- Remove or constrain the exposed surface so the precondition an attacker relies on can no longer be met — for example by enforcing authentication, network segmentation, or input validation at the trust boundary.
- Apply the vendor fix or configuration change that eliminates the underlying weakness in the affected component.
- Add a monitoring or detection control so any future regression of this condition is surfaced before it can be exploited.

Verification. After the fix is deployed, re-run the test case associated with this finding against the remediated build and confirm the condition no longer reproduces. The retest result should be retained as evidence that the remediation is effective, and the finding's lifecycle advanced to Fixed only once that passing result is recorded for the view under test.

EVIDENCE

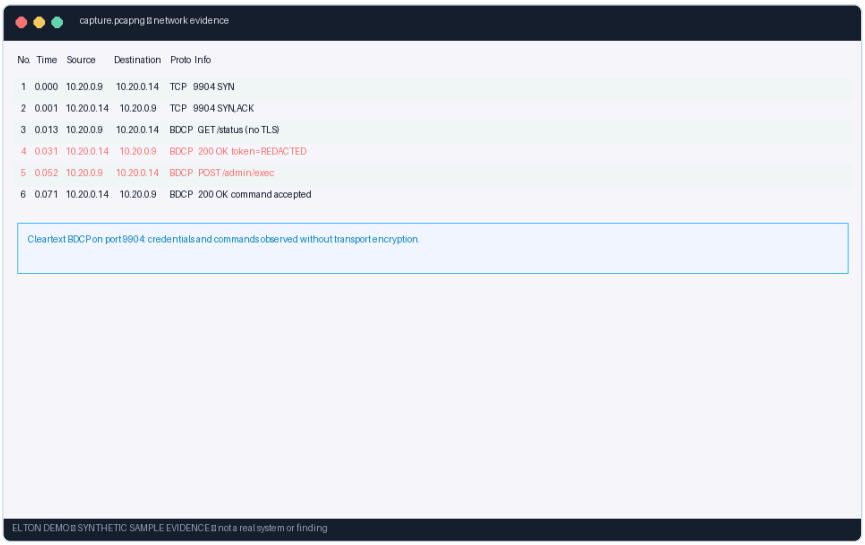


FIGURE 1

Screenshot captured while reproducing the condition during testing.

State-changing requests are accepted without an anti-forgery token

FAIL

ID TC-epQnL-ubavZ Component the component Category Application Software Source EMB3D TID-322 Threat TH-XN7JR-kHB7j Cross Site Request Forgery (CSRF) Attestation Beta Infusion Pump v5.1 Security Assessment

OBJECTIVE

While an operator session is active, replay a state-changing settings request from a page on a different origin and observe whether the application applies it.

CRITERIA

PASS

The application rejects the request because it lacks a valid anti-forgery token or breaks the same-site cookie policy.

FAIL

The application applies the change.

OUTCOME · FAIL CRITERIA MET

A settings change submitted from a page on another origin was applied using the operator's session cookie. The application checks no anti-forgery token and the cookie carries no same-site restriction.

ANALYSIS

A form hosted on a second origin posted an alarm-limit update to the management endpoint. The response was 200 and the new limit appeared in the next settings read. The request carried only the session cookie; no token header or field was required.

REMEDIATION

Require a per-session anti-forgery token on every state-changing request, set the session cookie to SameSite=Lax or Strict, and check the Origin header.

Verify: USB Permission Broadcast Uses Placeholder Action

FAIL

ID TC-iqfMk-XRKPp Component the component Category Other Finding EL-d2hJb-LnZs4
Attestation Beta Infusion Pump v5.1 Security Assessment

OBJECTIVE

Attempt to reproduce the condition described by 'USB Permission Broadcast Uses Placeholder Action' on the target under the preconditions modelled for this view (Privileged Session).

The tester positions against the affected interface, follows the documented reproduction steps, and records the request/response, console output and a screenshot as evidence. The reproduction package attached to each instance contains the steps, the raw request and response, and a placeholder proof-of-concept.

CRITERIA

PASS
The condition cannot be reproduced on the target: the affected function enforces the expected security control and the postconditions the finding would establish are not observed.

FAIL
The condition is reproduced: the affected function behaves insecurely as described, the finding's postconditions are observed, and the result is captured as evidence confirming the finding.

OUTCOME · FAIL CRITERIA MET

Reproduction of 'USB Permission Broadcast Uses Placeholder Action' during the engagement. The condition was triggered against the affected component and the result captured as evidence.

The tester followed the documented reproduction steps under the modelled preconditions and observed the finding's postconditions, demonstrating that the weakness is reachable and that its impact is as scored.

ANALYSIS ‹

The component registers an exported broadcast receiver for USB permission events. It uses the action string com.android.example.USB_PERMISSION, which is the placeholder from Android's USB host API documentation. This string is not scoped to the app's namespace, so any installed application or ADB session can broadcast it. Android's USB permission model relies on a unique action string to confirm user approval. When an app requests permission via UsbManager.requestPermission(), the system sends a PendingIntent with this action to the app's receiver. The receiver treats this broadcast as a signal that permission was granted. Because the action string is not protected, a malicious app can impersonate the system by broadcasting the same string, potentially tricking the receiver into granting access to a connected USB device. The receiver also listens for the system-defined actions USB_DEVICE_ATTACHED and USB_DEVICE_DETACHED, which are protected and can only be sent by the system. The custom action has no such protection. The impact depends on how the receiver handles the broadcast: if it re-checks permission with UsbManager.hasPermission(), the attack is limited to cases where the device is physically connected. If it trusts the broadcast directly, the attack is more direct.

Attack path. Reaching this weakness requires the following preconditions to hold: Privileged Session. Once exploited, the finding establishes Data Access, which an attacker can carry forward into subsequent steps of the chain. These pre-and post-conditions place the finding in the view's dependency graph, where its exploitability tier is derived from whether every precondition can be produced from an initial-access vector either directly or through another finding.

Illustrative example (SQL). The condition resembles unparameterised query construction, where operator-supplied input is concatenated into the statement:

```sql
-- Vulnerable: the medical-record number is concatenated straight into the SQL text
SELECT name, dob FROM patients WHERE mrn = '" + userInput + "';

-- Remediated: bind the value as a parameter so it can never alter the query
SELECT name, dob FROM patients WHERE mrn = @mrn;
```

Scoring rationale. The CVSS v3.1 vector `CVSS:3.1/AV:P/AC:H/PR:L/UI:N/S:U/C:L/I:L/A:N` and the CVSS v4.0 vector `CVSS:4.0/AV:P/AC:H/AT:P/PR:L/UI:N/VC:L/VI:L/VA:N/SC:N/SI:N/SA:N` were selected from the observed attack vector, the access complexity and privileges required to trigger the condition, and the confidentiality, integrity and availability impact measured against the affected clinical function. The exploitability classification above is derived independently from the condition graph and is consistent with, but not a substitute for, the CVSS base metrics.

REMEDIATION ‹

ELTON recommends replacing the placeholder action com.android.example.USB_PERMISSION with a package-scoped string, such as com.the organization.neuromod.vnstherapy.the use caseapp.prod.us.USB_PERMISSION, in both the requestPermission PendingIntent and the receiver's intent filter. Declare this custom permission with android:protectionLevel="signature" to ensure only apps signed with the organization key can broadcast it. Verify the fix by confirming that the receiver no longer responds to broadcasts of the placeholder action and that USB permission checks remain functional for legitimate devices.

Recommended actions, in priority order:

- Remove or constrain the exposed surface so the precondition an attacker relies on can no longer be met — for example by enforcing authentication, network segmentation, or input validation at the trust boundary.
- Apply the vendor fix or configuration change that eliminates the underlying weakness in the affected component.
- Add a monitoring or detection control so any future regression of this condition is surfaced before it can be exploited.

Verification. After the fix is deployed, re-run the test case associated with this finding against the remediated build and confirm the condition no longer reproduces. The retest result should be retained as evidence that the remediation is effective, and the finding's lifecycle advanced to Fixed only once that passing result is recorded for the view under test.

EVIDENCE

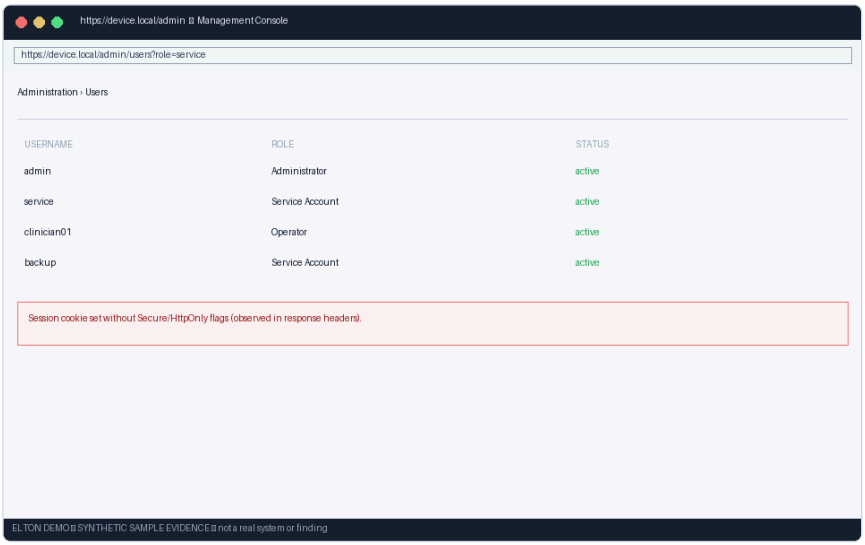


FIGURE 1

Screenshot captured while reproducing the condition during testing.

Verify: Application Directories World-Writable on the component

FAIL

ID TC-quj7B-EqzXK Component Beta Infusion Pump Category Other Finding EL-2Qd2G-sEefY
Attestation Beta Infusion Pump v5.1 Security Assessment

OBJECTIVE

Attempt to reproduce the condition described by 'Application Directories World-Writable on the component' on the target under the preconditions modelled for this view (Network Position).

The tester positions against the affected interface, follows the documented reproduction steps, and records the request/response, console output and a screenshot as evidence. The reproduction package attached to each instance contains the steps, the raw request and response, and a placeholder proof-of-concept.

CRITERIA

PASS
The condition cannot be reproduced on the target: the affected function enforces the expected security control and the postconditions the finding would establish are not observed.

FAIL
The condition is reproduced: the affected function behaves insecurely as described, the finding's postconditions are observed, and the result is captured as evidence confirming the finding.

OUTCOME · FAIL CRITERIA MET

Reproduction of 'Application Directories World-Writable on the component' during the engagement. The condition was triggered against the affected component and the result captured as evidence.

The tester followed the documented reproduction steps under the modelled preconditions and observed the finding's postconditions, demonstrating that the weakness is reachable and that its impact is as scored.

ANALYSIS

the component's installer leaves the `bin\log` and `bin\clientconf` directories writable by every local account. The testers captured the permissions with `icacls` :- `bin\log` carries non-inherited entries granting **BUILTIN\Users:(F)** and **Everyone:(OI)(CI)(F)**. - `bin\clientconf` grants **Everyone:(OI)(CI)(F)**. The parent directories hand out only read-execute, so these entries are deliberate. The `bin\log` directory holds operational logs (`dsout.log` , `compressing.log` , `clientprocessor.log` , `studyprocessor.log` , `the product linecom.log`) that the SYSTEM services update at runtime. A standard user can therefore delete, truncate or replace these logs, breaking audit integrity. The testers proved the file-plant primitive: a **Users**-only account wrote a benign marker file into `bin\log` from its limited token, and the write succeeded. The file was owned by the low-privilege account, and its content confirmed the account was not an administrator. The weakness is local only: an attacker needs an account on the server. Inside that boundary, the impact on integrity and availability is high, since the directories decide what the services log and whether they run correctly. The testers confirmed the permissions on the examined deployment.

Attack path. Reaching this weakness requires the following preconditions to hold: Network Position. Once exploited, the finding establishes Credential, Code Execution, which an attacker can carry forward into subsequent steps of the chain. These pre-and post-conditions place the finding in the view's dependency graph, where its exploitability tier is derived from whether every precondition can be produced from an initial-access vector either directly or through another finding.

Illustrative example (SQL). The condition resembles unparameterised query construction, where operator-supplied input is concatenated into the statement:

```sql
-- Vulnerable: the medical-record number is concatenated straight into the SQL text
SELECT name, dob FROM patients WHERE mrn = '" + userInput + "';

-- Remediated: bind the value as a parameter so it can never alter the query
SELECT name, dob FROM patients WHERE mrn = @mrn;
```

Scoring rationale. The severity was derived from the observed attack vector, the access complexity and privileges required to trigger the condition, and the confidentiality, integrity and availability impact measured against the affected clinical function.

REMEDIATION

ELTON recommends removing the **Everyone** and **BUILTIN\Users** Full Control grants from the `bin\log` and `bin\clientconf` directories of the component. Grant runtime write access only to the service account and administrators. Verify that no SYSTEM process loads executable content from any directory writable by non-administrators. Correct the installer so it stops applying these permissive access-control entries. Until the permissions are fixed, restrict interactive logon to administrators and review the directories for unexpected files. Confirm the fix by running `icacls` on the directories: standard users should have no write access.

Recommended actions, in priority order:

- Remove or constrain the exposed surface so the precondition an attacker relies on can no longer be met — for example by enforcing authentication, network segmentation, or input validation at the trust boundary.
- Apply the vendor fix or configuration change that eliminates the underlying weakness in the affected component.
- Add a monitoring or detection control so any future regression of this condition is surfaced before it can be exploited.

Verification. After the fix is deployed, re-run the test case associated with this finding against the remediated build and confirm the condition no longer reproduces. The retest result should be retained as evidence that the remediation is effective, and the finding's lifecycle advanced to Fixed only once that passing result is recorded for the view under test.

EVIDENCE

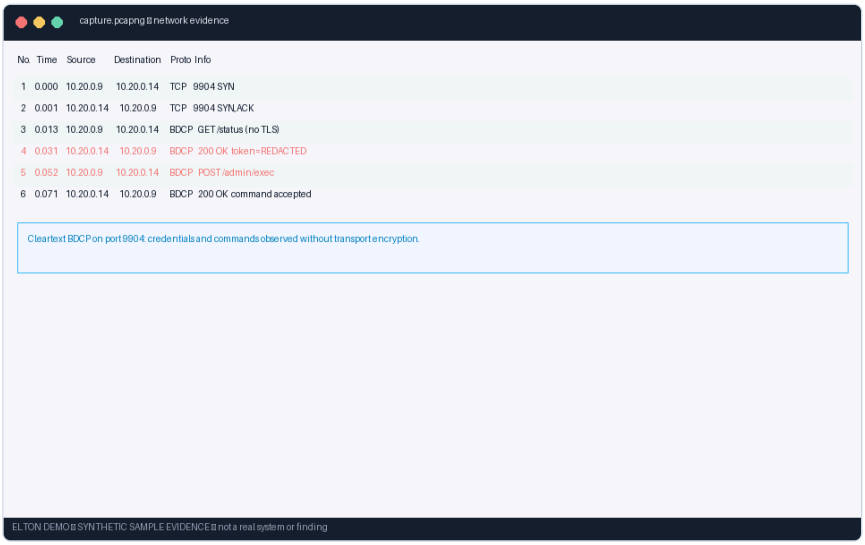


FIGURE 1

Screenshot captured while reproducing the condition during testing.

Firmware update accepts an unsigned image over the network update channel

FAIL

ID TC-sMCGX-QUpzT Component EA Category System Software Source EMB3D TID-211
Threat TH-qpnN6-j63dC Device Allows Unauthenticated Firmware Installation Attestation Beta Infusion Pump v5.1 Security Assessment

OBJECTIVE

Submit a copy of the v5.1 firmware image with its signature block removed through the network update channel and observe whether the controller installs it.

The test covers the update path an attacker on the clinical network could reach.

CRITERIA

PASS
The controller rejects the unsigned image, reports an authentication failure and keeps running the installed firmware.

FAIL
The controller accepts the unsigned image and installs it.

OUTCOME · FAIL CRITERIA MET

Unsigned image installed. The controller accepted an image whose signature block had been removed and rebooted into it, so the network update channel does not authenticate what it installs.

A modified build therefore ran in place of the signed firmware, which gives an attacker on the update path control of the controller.

ANALYSIS

A copy of the v5.1.0 image with its trailer stripped was pushed to the update endpoint from a host on the clinical network. The upload completed without a credential or signature check, the controller rebooted into the new image, and a marker string added to the build appeared on the diagnostic console.

REMEDIATION

Verify an asymmetric signature over the whole image before anything is written to flash, reject images that lack one, and keep the verification key in a read-only region. Require an authenticated operator session to start an update.

Verify: Application Files World-Writable in the component

FAIL

ID TC-tT9dv-ac47z Component Beta Infusion Pump Category Other Finding EL-U9MwD-eijMn
Attestation Beta Infusion Pump v5.1 Security Assessment

OBJECTIVE

Attempt to reproduce the condition described by 'Application Files World-Writable in the component' on the target under the preconditions modelled for this view (Network Position, Credential).

The tester positions against the affected interface, follows the documented reproduction steps, and records the request/response, console output and a screenshot as evidence. The reproduction package attached to each instance contains the steps, the raw request and response, and a placeholder proof-of-concept.

CRITERIA

PASS
The condition cannot be reproduced on the target: the affected function enforces the expected security control and the postconditions the finding would establish are not observed.

FAIL
The condition is reproduced: the affected function behaves insecurely as described, the finding's postconditions are observed, and the result is captured as evidence confirming the finding.

OUTCOME · FAIL CRITERIA MET

Reproduction of 'Application Files World-Writable in the component' during the engagement. The condition was triggered against the affected component and the result captured as evidence.

The tester followed the documented reproduction steps under the modelled preconditions and observed the finding's postconditions, demonstrating that the weakness is reachable and that its impact is as scored.

ANALYSIS

The application study viewer keeps its the component integration settings in `the componentintegration.conf` and other configuration files under `the product line Healthcare\the application`. The installer leaves six directories under that path writable by every local account: `the componentIntegration`, `the componentIntegration\conf`, `the componentIntegration\data`, `bin\clientconf`, `bin\log` and `bin\log\CAD`. The listing the testers captured shows a non-inherited **Everyone:(OI)(CI)(F)** access-control entry on each:

TEXT

```
C:\Program Files\the product line Healthcare\the application\the componentIntegration Everyone:(OI)(CI)(F)
C:\Program Files\the product line Healthcare\the application\the componentIntegration\conf Everyone:(OI)(CI)(F)
C:\Program Files\the product line Healthcare\the application\the componentIntegration\data Everyone:(OI)(CI)(F)
C:\Program Files\the product line Healthcare\the application\bin\clientconf Everyone:(OI)(CI)(F)
C:\Program Files\the product line Healthcare\the application\bin\log Everyone:(OI)(CI)(F)
C:\Program Files\the product line Healthcare\the application\bin\log\CAD Everyone:(OI)(CI)(F)
```

The `the componentIntegration\conf` directory holds `the componentintegration.conf`, which decides how the viewer connects to and trusts the clinical imaging archive. A standard user can edit or replace it, redirecting the connection or altering the trust rules. The other directories hold integration data, client configuration and logs, so an edit there changes what the viewer does or hides what it did. No elevation is involved: any interactive user, or any process running as one, can change these files. The viewer runs in the interactive user context, so the weakness is about configuration integrity and data tampering

rather than a direct privilege escalation. The testers observed the permissions and the configuration file but did not execute tampering writes in this non-destructive pass. The affected configuration governs the workstation's connection to clinical imaging infrastructure, so the integrity impact is high for a diagnostic imaging workflow. Confidentiality and availability are not directly affected, since the files do not hold the asset and the viewer can still run after tampering.

Attack path. Reaching this weakness requires the following preconditions to hold: Network Position, Credential. Once exploited, the finding establishes Data Access, which an attacker can carry forward into subsequent steps of the chain. These pre- and post-conditions place the finding in the view's dependency graph, where its exploitability tier is derived from whether every precondition can be produced from an initial-access vector either directly or through another finding.

Illustrative example (SQL). The condition resembles unparameterised query construction, where operator-supplied input is concatenated into the statement:

```sql
-- Vulnerable: the medical-record number is concatenated straight into the SQL text
SELECT name, dob FROM patients WHERE mrn = '" + userInput + "';

-- Remediated: bind the value as a parameter so it can never alter the query
SELECT name, dob FROM patients WHERE mrn = @mrn;
```

Scoring rationale. The severity was derived from the observed attack vector, the access complexity and privileges required to trigger the condition, and the confidentiality, integrity and availability impact measured against the affected clinical function.

REMEDIATION

ELTON recommends removing the **Everyone Full-control** access-control entries from the six the application directories. Grant the workstation user read access to the component integration configuration and write access only to the runtime directories the viewer legitimately writes to, scoped to that user. Relocate per-user runtime data and logs under the asset rather than a world-writable directory inside `Program Files`. Correct the installer so it stops applying **Everyone Full-control** access-control entries. Until a corrected installer is available, apply the permissions manually after each installation or update. To verify the fix, run `icacls` on each directory and confirm that **Everyone** no longer holds full control.

Recommended actions, in priority order:

- Remove or constrain the exposed surface so the precondition an attacker relies on can no longer be met — for example by enforcing authentication, network segmentation, or input validation at the trust boundary.
- Apply the vendor fix or configuration change that eliminates the underlying weakness in the affected component.
- Add a monitoring or detection control so any future regression of this condition is surfaced before it can be exploited.

Verification. After the fix is deployed, re-run the test case associated with this finding against the remediated build and confirm the condition no longer reproduces. The retest result should be retained as evidence that the remediation is effective, and the finding's lifecycle advanced to Fixed only once that passing result is recorded for the view under test.

EVIDENCE

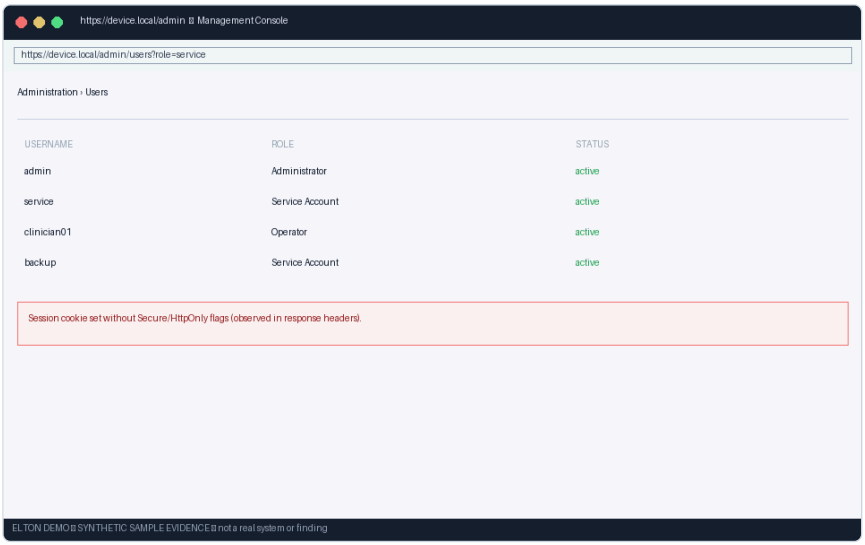


FIGURE 1

Screenshot captured while reproducing the condition during testing.

Verify: AppLocker is Disabled

FAIL

ID TC-wcwZL-TuwTP Component Beta Infusion Pump Category Other Attestation Beta Infusion Pump v5.1 Security Assessment

OBJECTIVE

Attempt to reproduce the condition described by 'AppLocker is Disabled' on the target under the preconditions modelled for this view (Credential).

The tester positions against the affected interface, follows the documented reproduction steps, and records the request/response, console output and a screenshot as evidence. The reproduction package attached to each instance contains the steps, the raw request and response, and a placeholder proof-of-concept.

CRITERIA

PASS

The condition cannot be reproduced on the target: the affected function enforces the expected security control and the postconditions the finding would establish are not observed.

FAIL

The condition is reproduced: the affected function behaves insecurely as described, the finding's postconditions are observed, and the result is captured as evidence confirming the finding.

OUTCOME · FAIL CRITERIA MET

Reproduction of 'AppLocker is Disabled' during the engagement. The condition was triggered against the affected component and the result captured as evidence.

The tester followed the documented reproduction steps under the modelled preconditions and observed the finding's postconditions, demonstrating that the weakness is reachable and that its impact is as scored.

ANALYSIS

Windows Firewall and Defender are enabled on the devices, providing baseline protection. However, on the productMonitor, Local Security Policy access is restricted, leaving the status of AppLocker unknown. On the AC Recording device, AppLocker is explicitly disabled. This configuration gap allows an attacker who gains initial access to execute or create native Windows programs and scripts in writable locations without restriction. Such freedom enables attackers to perform Living off the Land (LotL) techniques, using legitimate Microsoft utilities like Powershell and cmd.exe to conduct reconnaissance, escalate privileges, or deploy malware. The assessment team the organizationnstrated the ability to write and execute scripts from unprivileged accounts and exploited other weaknesses such as the lack of authentication for DirectPlay, illustrating how the absence of AppLocker enforcement facilitates malicious activity. The affected component, the system, is particularly sensitive due to its role in medical device operation. The combination of disabled or uncertain AppLocker status and writable file system locations creates an attack surface that can be leveraged for lateral movement and privilege escalation within the local network environment.

Attack path. Reaching this weakness requires the following preconditions to hold: Credential. Once exploited, the finding establishes Code Execution, which an attacker can carry forward into subsequent steps of the chain. These pre- and post-conditions place the finding in the view's dependency graph, where its exploitability tier is derived from whether every precondition can be produced from an initial-access vector either directly or through another finding.

Illustrative example (SQL). The condition resembles unparameterised query construction, where operator-supplied input is concatenated into the statement:

```sql
-- Vulnerable: the medical-record number is concatenated straight into the SQL text
SELECT name, dob FROM patients WHERE mrn = '" + userInput + "';

-- Remediated: bind the value as a parameter so it can never alter the query
SELECT name, dob FROM patients WHERE mrn = @mrn;
```

Scoring rationale. The CVSS v3.1 vector `CVSS:3.1/AV:N/AC:H/PR:H/UI:N/S:U/C:H/I:H/A:H` and the CVSS v4.0 vector `CVSS:4.0/AV:N/AC:H/AT:P/PR:H/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N` were selected from the observed attack vector, the access complexity and privileges required to trigger the condition, and the confidentiality, integrity and availability impact measured against the affected clinical function. The exploitability classification above is derived independently from the condition graph and is consistent with, but not a substitute for, the CVSS base metrics.

REMEDIATION

ELTON recommends enabling and properly configuring AppLocker with the trust level ruleset on both the productMonitor and AC Recording devices to restrict unauthorized program execution. This includes verifying AppLocker's status on the productMonitor by gaining appropriate permissions to access Local Security Policy settings. Additionally, ELTON advises developing machine-specific AppLocker rules that prevent the trust level users from executing high-risk executables such as Powershell and cmd.exe. These measures will reduce the attack surface by limiting the ability of threat actors to use native Windows tools for malicious purposes, thereby strengthening the overall security posture of the affected medical devices.

Recommended actions, in priority order:

- Remove or constrain the exposed surface so the precondition an attacker relies on can no longer be met — for example by enforcing authentication, network segmentation, or input validation at the trust boundary.
- Apply the vendor fix or configuration change that eliminates the underlying weakness in the affected component.
- Add a monitoring or detection control so any future regression of this condition is surfaced before it can be exploited.

Verification. After the fix is deployed, re-run the test case associated with this finding against the remediated build and confirm the condition no longer reproduces. The retest result should be retained as evidence that the remediation is effective, and the finding's lifecycle advanced to Fixed only once that passing result is recorded for the view under test.

EVIDENCE

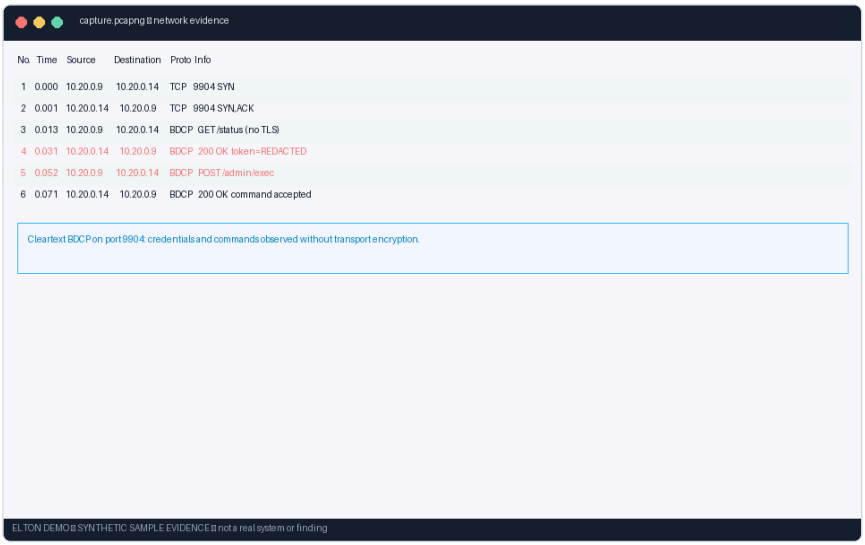


FIGURE 1

Screenshot captured while reproducing the condition during testing.

Verify: Authentication Not Required on Service

FAIL

ID TC-wy4zi-TXZBu Component Beta Infusion Pump Category Other Attestation Beta Infusion Pump v5.1 Security Assessment

OBJECTIVE

Attempt to reproduce the condition described by 'Authentication Not Required on Service' on the target under the preconditions modelled for this view (no modelled conditions).

The tester positions against the affected interface, follows the documented reproduction steps, and records the request/response, console output and a screenshot as evidence. The reproduction package attached to each instance contains the steps, the raw request and response, and a placeholder proof-of-concept.

CRITERIA

PASS

The condition cannot be reproduced on the target: the affected function enforces the expected security control and the postconditions the finding would establish are not observed.

FAIL

The condition is reproduced: the affected function behaves insecurely as described, the finding's postconditions are observed, and the result is captured as evidence confirming the finding.

OUTCOME · FAIL CRITERIA MET

Reproduction of 'Authentication Not Required on Service' during the engagement. The condition was triggered against the affected component and the result captured as evidence.

The tester followed the documented reproduction steps under the modelled preconditions and observed the finding's postconditions, demonstrating that the weakness is reachable and that its impact is as scored.

ANALYSIS

The productMonitor application listens on UDP port 1213, accepting connection requests without requiring authentication. The Camtronics library, located in C:/Program Files(x86)/the product line Healthcare/the product line the product, facilitates communication with the DirectPlay server by dispatching commands. Because the service does not verify the identity of connecting clients, any actor with network access to this port can send commands directly to the DirectPlay server. This unauthenticated access means that an attacker can issue arbitrary DirectPlay commands, potentially altering the behavior or data of the productMonitor system. The vulnerability arises from the service's design, which lacks session initiation negotiation or credential verification, allowing command injection from any source able to reach the UDP port. The exploitation requires only network reachability and the ability to send UDP packets, making it a significant risk in environments where the port is exposed. The absence of authentication mechanisms on this critical communication channel undermines the confidentiality, integrity, and availability of the productMonitor application. Attackers could manipulate medical monitoring data or disrupt normal operations by exploiting this flaw, emphasizing the need for robust identity verification before command acceptance.

Attack path. Reaching this weakness requires the following preconditions to hold: no modelled conditions. Once exploited, the finding establishes Data Access, which an attacker can carry forward into subsequent steps of the chain. These pre- and post-conditions place the finding in the view's dependency graph, where its exploitability tier is derived from whether every precondition can be produced from an initial-access vector either directly or through another finding.

Illustrative example (SQL). The condition resembles unparameterised query construction, where operator-supplied input is concatenated into the statement:

```sql
-- Vulnerable: the medical-record number is concatenated straight into the SQL text
SELECT name, dob FROM patients WHERE mrn = '" + userInput + "';

-- Remediated: bind the value as a parameter so it can never alter the query
SELECT name, dob FROM patients WHERE mrn = @mrn;
```

Scoring rationale. The CVSS v3.1 vector `CVSS:3.1/AV:N/AC:H/PR:L/UI:N/S:U/C:H/I:H/A:N` and the CVSS v4.0 vector `CVSS:4.0/AV:N/AC:H/AT:P/PR:L/UI:N/VC:H/VI:H/VA:N/SC:N/SI:N/SA:N` were selected from the observed attack vector, the access complexity and privileges required to trigger the condition, and the confidentiality, integrity and availability impact measured against the affected clinical function. The exploitability classification above is derived independently from the condition graph and is consistent with, but not a substitute for, the CVSS base metrics.

REMEDIATION

ELTON recommends the organization implement an authentication mechanism for the DirectPlay service on the productMonitor application. Specifically, when the AC the workstation or any client attempts to connect, the productMonitor should initiate a session negotiation that requires the client to present valid identification credentials before accepting commands. Additionally, the productMonitor should provide its own identification materials to the client to enable mutual authentication, ensuring both parties verify each other's authenticity. This approach will prevent unauthorized command injection and secure the communication channel against unauthenticated access. Network the componentation and restricting UDP port 1213 access to trusted hosts can serve as interim mitigations until authentication is fully implemented.

Recommended actions, in priority order:

- Remove or constrain the exposed surface so the precondition an attacker relies on can no longer be met — for example by enforcing authentication, network segmentation, or input validation at the trust boundary.
- Apply the vendor fix or configuration change that eliminates the underlying weakness in the affected component.
- Add a monitoring or detection control so any future regression of this condition is surfaced before it can be exploited.

Verification. After the fix is deployed, re-run the test case associated with this finding against the remediated build and confirm the condition no longer reproduces. The retest result should be retained as evidence that the remediation is effective, and the finding's lifecycle advanced to Fixed only once that passing result is recorded for the view under test.

EVIDENCE

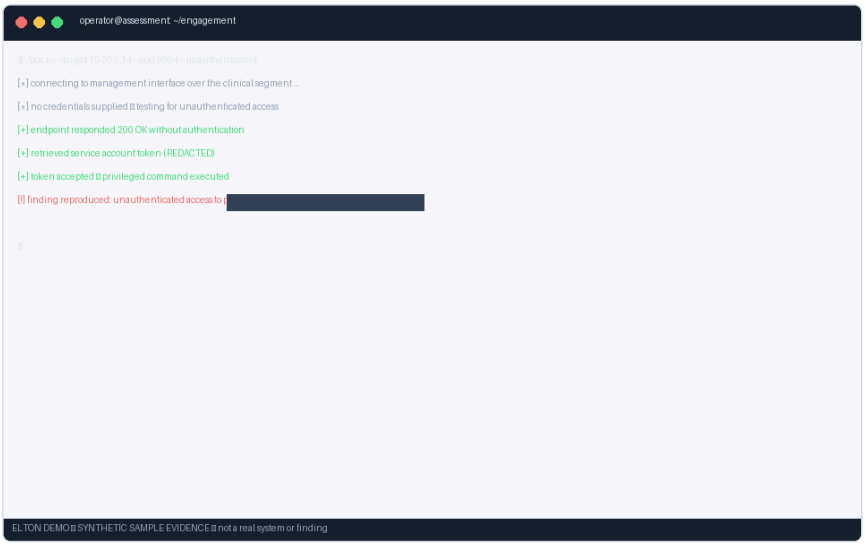


FIGURE 1

Screenshot captured while reproducing the condition during testing.

CVSS Justifications

The reasoning behind each scored finding's CVSS v4.0 rating, metric by metric: the value, a summary, the questions that decided it with the answer given to each, and the evidence each answer rests on. Each finding's score on its fact rail links here.

EL-t3Fai-Dhmwy End-of-Life and Vulnerable Third-Party Components Present in the component

CVSS V4.0 **9.3** CRITICAL

`CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N`

Attack Vector — Network

aliased from 3.1/AV

The attack is performed by sending a crafted HTTP POST request over the network to the Local Run Manager API endpoint, which is a network interface on the Local Run Manager component (Local Run Manager (LRM)).

REASONING & EVIDENCE

> Yes
> The attack is performed by sending a crafted HTTP POST request over the network to the Local Run Manager API endpoint, which is a network interface on the Local Run Manager component.
> INTELLIGENCE Summary *"The web tier exposes an unauthenticated route, POST /api/admin, whose handler creates a full administrator via CreateUserUsingAccessCode(..., isAdmin: true)."*

> Yes
> The Local Run Manager API endpoint is accessed over HTTPS on TCP port 443, which uses TCP/IP protocols (OSI layer 3/4) (Local Run Manager HTTPS endpoint (TCP/443, TLSv1.2)).
> INTELLIGENCE Finding description

Attack Complexity — Low

The attacker can forge a valid access code with a single SHA-256 computation using public inputs and a shared constant embedded in the Local Run Manager component, and the unauthenticated POST /api/admin route allows repeated attempts without any security-hardening mechanisms or special conditions (Local Run Manager (LRM), Local Run Manager API).

REASONING & EVIDENCE

> No
> The vulnerability does not require defeating or evading any built-in security-hardening mechanism such as ASLR, DEP, code signing, or certificate pinning; the access code validation relies on a shared constant embedded in the DLL without per-device secrets or cryptographic protections, and the attacker can compute valid codes directly without bypassing protections.
> INTELLIGENCE Test case: Unauthenticated administrative account provisioning gated only by a forgeable access code *"The imports show no OS crypto provider and no server-held or per-device key, only the system clock."*
> INTELLIGENCE Summary *"The pepper is a globally shared constant stored obfuscated in .rdata whose de-obfuscation routine and alphabet ship in the same DLL."*
> INTELLIGENCE Summary *"The [RequireCsrfCustomHeader] header is set trivially by a non-browser client and is not authentication, and the PGP signature check protects on-disk integrity only."*

Attack Requirements — None

The attacker can forge a valid access code with a single SHA-256 computation using public inputs and a shared constant embedded in the Local Run Manager component, and the unauthenticated POST /api/admin route allows repeated attempts without any special conditions or timing requirements (Local Run Manager (LRM), Local Run Manager API).

REASONING & EVIDENCE

> Yes
> The attacker can forge a valid access code for any instrument with a single SHA-256 computation using public inputs and a shared constant embedded in the Local Run Manager (LRM) component, and the POST /api/admin route is documented as

unprotected and accessible without authentication or rate limiting, allowing repeated attempts at will via the Local Run Manager API interface.
INTELLIGENCE Summary *"The web tier exposes an unauthenticated route, POST /api/admin, whose handler (lrm/Northwind Bio.MAP.Api.Service.cs:1234-1250) creates a full administrator via CreateUserUsingAccessCode(..., isAdmin: true). The route is annotated "This is an unprotected operation" (:1193-1194) and its only gate is IsAccessCodeValid(RequesterName, AccessCode), which resolves to the native export ckfpw in svcfld.dll (lrm/Northwind Bio.MAP.Domain.cs:1765-1767)."*
INTELLIGENCE Finding description
INTELLIGENCE Finding description *"The native check applies no rate limiting or lockout of its own; the one-time property is enforced only afterward in managed code (IsAccessCodeUsed), which prevents reuse of an already-spent code but not computation of a fresh one."*

No
The attack works against the product as it ships and is normally run, with no requirement for a non-default setting, specific version, or particular peripheral; the Local Run Manager API exposes an unauthenticated route that can be exploited using public inputs and a shared constant embedded in the shipped DLL, as described in the Local Run Manager (LRM) and its API interface Local Run Manager API.
INTELLIGENCE Summary *"The web tier exposes an unauthenticated route, POST /api/admin, whose handler creates a full administrator via CreateUserUsingAccessCode(..., isAdmin: true). The route is annotated "This is an unprotected operation" and its only gate is IsAccessCodeValid(RequesterName, AccessCode), which resolves to the native export ckfpw in svcfld.dll."*

No
The attacker can start the attack whenever they choose by sending a crafted HTTP POST request to the unauthenticated Local Run Manager API endpoint; there is no requirement to win a race or inject into an exchange already in progress, as the access code can be computed on demand from public inputs and a shared constant, as shown in the Local Run Manager API and the Local Run Manager (LRM).
INTELLIGENCE Finding description *"Forgeability was proven on the instrument without creating any account. A standalone implementation of the recovered construction, run on the device, read the obfuscated constant and the de-obfuscation alphabet from the loaded module image, applied the recovered de-obfuscation, computed a candidate code for an unauthenticated requestId and the current UTC date entirely in memory, and passed it to the shipped exported validator svcfld.dll!ckfpw(requestId, code, 5). The validator returned accept."*

Privileges Required None

aliased from 3.1/PR

The Local Run Manager web application supports multiple users and roles including administrators, but the POST /api/admin route is unauthenticated and only gated by a forgeable access code, so no privileges are required to exploit the vulnerability (Local Run Manager (LRM), Local Run Manager API).

REASONING & EVIDENCE

Yes
The Local Run Manager (LRM) web application supports multiple users and roles, including administrator and standard users, as evidenced by the ability to create administrator accounts via the POST /api/admin route on the Local Run Manager component.
INTELLIGENCE Summary *"The web tier exposes an unauthenticated route, POST /api/admin, whose handler creates a full administrator via CreateUserUsingAccessCode(..., isAdmin: true)."*

No
The POST /api/admin route on Local Run Manager is unauthenticated and documented as 'This is an unprotected operation'; the only gate is a forgeable access code, so the attacker does not need prior authorization to exploit the vulnerability (Local Run Manager API).
INTELLIGENCE Finding description *"The route at :1193-1194 is documented 'This is an unprotected operation'; the only check before CreateUserUsingAccessCode(..., isAdmin: true) is IsAccessCodeValid(RequesterName, AccessCode)."*

User Interaction None

The attacker can create an administrator account without any legitimate user involvement or interaction by exploiting the unauthenticated POST /api/admin route on the Local Run Manager API.

REASONING & EVIDENCE

No
The attacker can forge the access code and create an administrator account without any legitimate user involvement or interaction, exploiting the unauthenticated POST /api/admin route on the Local Run Manager API.
INTELLIGENCE Finding description *"Forgeability was proven on the instrument without creating any account. A standalone implementation of the recovered construction, run on the device, read the obfuscated constant and the de-obfuscation alphabet from the loaded module image, applied the recovered de-obfuscation, computed a candidate code for an unauthenticated requestId and the*

> *current UTC date entirely in memory, and passed it to the shipped exported validator svcfld.dll!ckfpw(requestId, code, 5). The validator returned accept."*

Vulnerable System Confidentiality High

from the modelled C/I/A of the vulnerable system's rated postconditions

High on the vulnerable system, from the modelled C/I/A of the finding's postconditions: Attacker holds a self-provisioned Local Run Manager administrator account is rated High for confidentiality.

REASONING & EVIDENCE

> Modelled C/I/A · vulnerable system High
> From the modelled C/I/A of the finding's postconditions on the vulnerable system.
> CONDITION Attacker holds a self-provisioned Local Run Manager administrator account `confidentiality = H`

Vulnerable System Integrity High

from the modelled C/I/A of the vulnerable system's rated postconditions

High on the vulnerable system, from the modelled C/I/A of the finding's postconditions: Attacker holds a self-provisioned Local Run Manager administrator account is rated High for integrity.

REASONING & EVIDENCE

> Modelled C/I/A · vulnerable system High
> From the modelled C/I/A of the finding's postconditions on the vulnerable system.
> CONDITION Attacker holds a self-provisioned Local Run Manager administrator account `integrity = H`

Vulnerable System Availability High

from the modelled C/I/A of the vulnerable system's rated postconditions

High on the vulnerable system, from the modelled C/I/A of the finding's postconditions: Attacker holds a self-provisioned Local Run Manager administrator account is rated High for availability.

REASONING & EVIDENCE

> Modelled C/I/A · vulnerable system High
> From the modelled C/I/A of the finding's postconditions on the vulnerable system.
> CONDITION Attacker holds a self-provisioned Local Run Manager administrator account `availability = H`

Subsequent System Confidentiality None

the subsequent system is not counted: 3.1 S is U

Scope is Unchanged: exploiting this flaw affects nothing beyond the vulnerable system, so there is no subsequent-system impact.

Subsequent System Integrity None

the subsequent system is not counted: 3.1 S is U

Scope is Unchanged: exploiting this flaw affects nothing beyond the vulnerable system, so there is no subsequent-system impact.

Subsequent System Availability None

the subsequent system is not counted: 3.1 S is U

Scope is Unchanged: exploiting this flaw affects nothing beyond the vulnerable system, so there is no subsequent-system impact.

EL-iCW5i-nTY9g CouchbaseLite Database Encryption Disabled

CVSS V4.0 **8.5** HIGH

`CVSS:4.0/AV:L/AC:L/AT:N/PR:L/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N`

Attack Vector Local

aliased from 3.1/AV

The attack requires a local standard user session on the instrument to write replacement binaries to the writable directory C:\Northwind Bio on the Microsoft Windows 10 Operating System, with no network or physical interface involved in delivering the exploit.

REASONING & EVIDENCE

> No
> The attack requires a local standard user session on the instrument to write replacement binaries to the writable directory C:\Northwind Bio, with no network or physical interface involved in delivering the exploit.

INTELLIGENCE Finding description

No

The attacker does not need physical contact with the device but requires an authenticated local user session to write files to the vulnerable directory.

INTELLIGENCE Finding description

Attack Complexity Low

The attack does not require defeating or evading any built-in security-hardening mechanisms such as ASLR, DEP, code signing, or integrity validation on the Microsoft Windows 10 Operating System; the Software Restriction Policy uses path-based allowlisting without content inspection or hash verification, so replaced binaries execute without needing to bypass integrity checks or cryptographic protections.

REASONING & EVIDENCE

No

The attack does not require defeating or evading any built-in security-hardening mechanisms such as ASLR, DEP, code signing, or integrity validation; the Software Restriction Policy uses path-based allowlisting without content inspection or hash verification, so replaced binaries execute without needing to bypass integrity checks or cryptographic protections on the Microsoft Windows 10 Operating System.

INTELLIGENCE Finding description *"Software Restriction Policy is the sole application-control layer, with no AppLocker present. Its rule set under CodeIdentifiers holds a single Paths subkey with 45 path rules and no Hashes subkey at all, so the allowlist is entirely path-based."*

INTELLIGENCE Finding description *"A path rule authorizes execution by location and never inspects file content, so when production flips the policy to default-deny, a binary a standard user replaced inside C:\Northwind Bio still satisfies the C:\Northwind Bio* rule and runs."*

Attack Requirements None

The attacker only needs an authenticated Windows standard-user session on the Microsoft Windows 10 Operating System to write replacement binaries or DLLs to the writable C:\Northwind Bio directory, which are then loaded by privileged services running as LocalSystem, enabling repeated exploitation without additional special conditions or configurations beyond the standard user session; no race conditions or special peripherals are required.

REASONING & EVIDENCE

Yes

The attacker only needs an authenticated Windows standard-user session on the instrument to write replacement binaries or DLLs to the writable C:\Northwind Bio directory, which are then loaded by privileged services running as LocalSystem, enabling repeated exploitation without additional special conditions or configurations beyond the standard user session. The path-based Software Restriction Policy does not prevent execution of replaced binaries, so no further evasion or prerequisite is required beyond the standard user session on the Microsoft Windows 10 Operating System.

INTELLIGENCE Finding description *"The complete chain is a standard user writing a crafted executable or library over a shipped file in C:\Northwind Bio that a LocalSystem service or Real-Time Analysis loads, and on the next higher-privilege load the attacker's code runs as SYSTEM."*

INTELLIGENCE Finding description *"The prerequisite is nothing more than an authenticated standard-user session on the instrument; the impact is full SYSTEM-level code execution on a clinical sequencing device."*

No

The attack requires only an authenticated Windows standard-user session on the instrument, which is a default condition for a user with standard privileges; no special deployment, configuration, or peripheral is required beyond this standard user access to the writable C:\Northwind Bio directory on the Microsoft Windows 10 Operating System.

INTELLIGENCE Finding description *"The prerequisite is nothing more than an authenticated standard-user session on the instrument"*

No

The attacker can write replacement binaries or DLLs at any time during their authenticated session and trigger their execution by privileged services; there is no requirement to win a race or inject into an exchange already in progress on the Microsoft Windows 10 Operating System.

INTELLIGENCE Test case: No local escalation to kernel or SYSTEM privileges *"The full chain follows from those two facts: an authenticated standard user writes a crafted executable or library over a shipped file in C:\Northwind Bio, and on the next higher-privilege load a LocalSystem service or Real-Time Analysis executes the attacker's code as SYSTEM."*

Privileges Required Low

aliased from 3.1/PR

The attack requires an authenticated standard user session on the Microsoft Windows 10 Operating System to write replacement binaries to the writable C:\Northwind Bio directory, with no administrator or system-level privileges needed beforehand, reflecting the presence of multiple user privilege levels on the instrument.

REASONING & EVIDENCE

Yes

The vulnerable component is the Microsoft Windows 10 Operating System, which supports multiple user accounts including standard users and administrators, as evidenced by the presence of standard user sessions and the distinction between standard and elevated privileges in the attack narrative and intelligence. The attack requires a local standard user session on the instrument, indicating an authorization model with different privilege levels.

INTELLIGENCE Finding description *"The prerequisite is nothing more than an authenticated standard-user session on the instrument"*

INTELLIGENCE Test case: No local escalation to kernel or SYSTEM privileges *"A standard Windows user can escalate to SYSTEM by abusing the weak access control on C:\Northwind Bio"*

Yes

The attacker must already have an authenticated standard user session on the instrument to write replacement binaries to the writable C:\Northwind Bio directory, as the attack requires local file write access by a standard user. This means the attacker must be authorized before exploiting the vulnerability (Microsoft Windows 10 Operating System).

INTELLIGENCE Finding description *"The prerequisite is nothing more than an authenticated standard-user session on the instrument"*

No

The attacker only needs standard user privileges to write to the writable directory C:\Northwind Bio; no administrator or system-level privileges are required to exploit the vulnerability. The Local Run Manager Analysis and Job Services run as LocalSystem and load the replaced binaries, but the attacker does not need such privileges beforehand (Microsoft Windows 10 Operating System).

INTELLIGENCE Test case: No local escalation to kernel or SYSTEM privileges *"A standard Windows user can escalate to SYSTEM by abusing the weak access control on C:\Northwind Bio"*

User Interaction None

The attacker is a local standard user who can write replacement binaries or DLLs to the writable C:\Northwind Bio directory and trigger their execution by privileged services without requiring any action from another user on the Microsoft Windows 10 Operating System.

REASONING & EVIDENCE

No

The attacker is a local standard user who can write replacement binaries or DLLs to the writable C:\Northwind Bio directory and trigger their execution by privileged services without requiring any action from another user.

INTELLIGENCE Test case: No local escalation to kernel or SYSTEM privileges *"The full chain follows from those two facts: an authenticated standard user writes a crafted executable or library over a shipped file in C:\Northwind Bio, and on the next higher-privilege load a LocalSystem service or Real-Time Analysis executes the attacker's code as SYSTEM."*

Vulnerable System Confidentiality High

from the modelled C/I/A of the vulnerable system's rated postconditions

High on the vulnerable system, from the modelled C/I/A of the finding's postconditions: Attacker gains code execution as NT AUTHORITY\SYSTEM, a full escalation from standard user to SYSTEM on a clinical diagnostic instrument is rated High for confidentiality.

REASONING & EVIDENCE

Modelled C/I/A · vulnerable system High

From the modelled C/I/A of the finding's postconditions on the vulnerable system.

CONDITION Attacker gains code execution as NT AUTHORITY\SYSTEM, a full escalation from standard user to SYSTEM on a clinical diagnostic instrument `confidentiality = H`

Vulnerable System Integrity High

from the modelled C/I/A of the vulnerable system's rated postconditions

High on the vulnerable system, from the modelled C/I/A of the finding's postconditions: Attacker gains code execution as NT AUTHORITY\SYSTEM, a full escalation from standard user to SYSTEM on a clinical diagnostic instrument is rated High for integrity.

REASONING & EVIDENCE

Modelled C/I/A · vulnerable system High

From the modelled C/I/A of the finding's postconditions on the vulnerable system.

CONDITION Attacker gains code execution as NT AUTHORITY\SYSTEM, a full escalation from standard user to SYSTEM on a clinical diagnostic instrument `integrity = H`

Vulnerable System Availability High

from the modelled C/I/A of the vulnerable system's rated postconditions

High on the vulnerable system, from the modelled C/I/A of the finding's postconditions: Attacker gains code execution as NT AUTHORITY\SYSTEM, a full escalation from standard user to SYSTEM on a clinical diagnostic instrument is rated High for availability.

REASONING & EVIDENCE

> Modelled C/I/A · vulnerable system **High**
> From the modelled C/I/A of the finding's postconditions on the vulnerable system.
> CONDITION Attacker gains code execution as NT AUTHORITY\SYSTEM, a full escalation from standard user to SYSTEM on a clinical diagnostic instrument `availability = H`

Subsequent System Confidentiality None

the subsequent system is not counted: 3.1 S is U

Scope is Unchanged: exploiting this flaw affects nothing beyond the vulnerable system, so there is no subsequent-system impact.

Subsequent System Integrity None

the subsequent system is not counted: 3.1 S is U

Scope is Unchanged: exploiting this flaw affects nothing beyond the vulnerable system, so there is no subsequent-system impact.

Subsequent System Availability None

the subsequent system is not counted: 3.1 S is U

Scope is Unchanged: exploiting this flaw affects nothing beyond the vulnerable system, so there is no subsequent-system impact.

EL-dHcdP-J5BXq Weak Restrictions Present in Content Security Policy

CVSS V4.0 **6.9** MEDIUM

`CVSS:4.0/AV:L/AC:L/AT:N/PR:L/UI:N/VC:H/VI:L/VA:L/SC:N/SI:N/SA:N`

Attack Vector Local

aliased from 3.1/AV

The vulnerability is exploitable only via the UCS localhost listener on the Universal Copy Service, which listens exclusively on the loopback interface and is not reachable over any network beyond the local host (Universal Copy Service (UCS)).

REASONING & EVIDENCE

> No
> The vulnerability is exploited by sending unauthenticated HTTP API requests to the Universal Copy Service's loopback listener on 127.0.0.1:29644 and ::1:29644, which is a Local interface, not reachable over any network or communication protocol beyond the local host. The relevant interface is the UCS localhost listener (iface:01a10ed2-e9b6-7295-a2cf-fbe78d61362c) categorized as Local.
> INTELLIGENCE Summary *"The Universal Copy Service listens on `127.0.0.1:29644` (and `::1:29644`), owned by `universal-copy-service.exe` (pid 3976) running as `NT AUTHORITY\LocalService` with `SeImpersonatePrivilege` and `SeAssignPrimaryTokenPrivilege` enabled, a documented local path to SYSTEM."*

> No
> The attacker does not need physical contact with the device; any local code execution context on the instrument can connect to the loopback interface and exploit the vulnerability via the UCS localhost listener (iface:01a10ed2-e9b6-7295-a2cf-fbe78d61362c).
> INTELLIGENCE Summary *"The access condition is any code-execution context on the instrument; the loopback bind added in the prior cycle restricts the network origin but is not an access control."*

Attack Complexity Low

The Universal Copy Service accepts unauthenticated HTTP API requests on the loopback interface without any authentication or security mechanisms, allowing any local code execution context to exploit the vulnerability at will without needing to bypass security controls or special configurations (Universal Copy Service (UCS)).

REASONING & EVIDENCE

> No
> The Universal Copy Service exposes its API without any authentication or security mechanisms to evade; the OpenAPI document declares no security requirements, and unauthenticated requests succeed without needing to defeat any built-in security-hardening mechanisms such as ASLR, DEP, code signing, or certificate pinning.

INTELLIGENCE Finding description *"The OpenAPI document at `/swagger/v1/swagger.json` was served to an unauthenticated request and declares neither `securityDefinitions` nor any `security` requirement, so the contract itself advertises that nothing gates the operations."*
INTELLIGENCE Finding description *"Three unauthenticated requests each returned HTTP 200 with real data and no 401 or 403 challenge."*

Attack Requirements None

The Universal Copy Service listens on the loopback interface by default and accepts unauthenticated requests without requiring any special deployment or configuration conditions; the vulnerability is exploitable by any local code execution context on the instrument at any time without needing to win a race or inject into an ongoing exchange (Universal Copy Service (UCS)).

REASONING & EVIDENCE

Yes

The Universal Copy Service listens on the loopback interface at 127.0.0.1:29644 and ::1:29644 and accepts unauthenticated HTTP API requests that return privileged data and allow state-changing privileged operations without any authentication or access control, so an attacker with any local code execution context can exploit it at will. The service document at /swagger/v1/swagger.json declares neither securityDefinitions nor any security requirement, and unauthenticated requests to GET /v1/system/info, GET /v1/runs, and POST /v1/system/checks/disk_space each returned HTTP 200 with live data and no 401 or 403 challenge (Universal Copy Service (UCS), UCS localhost listener).
INTELLIGENCE Finding description *"The OpenAPI document at `/swagger/v1/swagger.json` was served to an unauthenticated request and declares neither `securityDefinitions` nor any `security` requirement, so the contract itself advertises that nothing gates the operations."*
INTELLIGENCE Finding description *"Three unauthenticated requests each returned HTTP 200 with real data and no 401 or 403 challenge."*

No

The Universal Copy Service listens on the loopback interface by default and accepts unauthenticated requests without requiring any special deployment or configuration conditions beyond the normal instrument operation. The vulnerability is exploitable by any local code execution context on the instrument without additional setup or non-default settings (Universal Copy Service (UCS)).
INTELLIGENCE Finding description *"The access condition is any code-execution context on the instrument. The loopback bind added in the prior cycle restricts the network origin but does not authenticate the caller, so it is not an access control."*
INTELLIGENCE Test case: Authentication required on all remote-accessible services *"The prior-cycle fix narrowed the listener to loopback but added no authentication layer, so any local code-execution context reaches this privileged file-movement and credential-management surface unauthenticated."*

No

The attacker can initiate the unauthenticated API requests to the Universal Copy Service at any time; there is no requirement to win a race or inject into an ongoing exchange. The service listens continuously on the loopback interface and responds immediately to requests.
INTELLIGENCE Finding description *"The listener was confirmed on 127.0.0.1:29644 and ::1:29644, owned by universal-copy-service.exe... The service self-identifies as "Universal Copy Service API" version 2.7.4.5117. Its OpenAPI document at `/swagger/v1/swagger.json` was served to an unauthenticated request and declares neither `securityDefinitions` nor any `security` requirement." (paraphrased)*
INTELLIGENCE Test case: Authentication required on all remote-accessible services *"The unauthenticated reachability was re-confirmed live on 2026-10-07 (GET /v1/system/info returned 200 with the running identity and no challenge)."*

Privileges Required Low

The Universal Copy Service (UCS) runs with elevated privileges as NT AUTHORITY\LocalService and exposes an unauthenticated API that allows privileged operations without authentication, indicating a privilege separation between the service and the underlying Windows 10 OS. In order to interact with this API, a user must be authenticated to the Microsoft Windows 10 Operating System resulting in a low privilege level requirement.

User Interaction None

Exploitation requires no user interaction as the attacker can send unauthenticated HTTP API requests directly to the UCS localhost listener at any time.

REASONING & EVIDENCE

No

The attacker can exploit the vulnerability by sending unauthenticated HTTP API requests directly to the Universal Copy Service's loopback listener without requiring any action or interaction from another user. The Universal Copy Service listens on 127.0.0.1:29644 and ::1:29644 and accepts privileged operations without authentication (UCS localhost listener).

INTELLIGENCE Finding description *"The service self-identifies as "Universal Copy Service API" version 2.7.4.5117. Its OpenAPI document at `/swagger/v1/swagger.json` was served to an unauthenticated request and declares neither `securityDefinitions` nor any `security` requirement, so the contract itself advertises that nothing gates the operations."*
INTELLIGENCE Finding description *"Three unauthenticated requests each returned HTTP 200 with real data and no 401 or 403 challenge."*
INTELLIGENCE Summary *"The access condition is any code-execution context on the instrument; the loopback bind added in the prior cycle restricts the network origin but is not an authentication control."*

Vulnerable System Confidentiality — High

from the modelled C/I/A of the vulnerable system's rated postconditions

High on the vulnerable system, from the modelled C/I/A of the finding's postconditions: Content written into a module Resources directory by a local principal is readable by an unauthenticated LAN request, the same static-content disclosure already carried by the two linked findings is rated High for confidentiality.

REASONING & EVIDENCE

Modelled C/I/A · vulnerable system — High
From the modelled C/I/A of the finding's postconditions on the vulnerable system.
CONDITION Content written into a module Resources directory by a local principal is readable by an unauthenticated LAN request, the same static-content disclosure already carried by the two linked findings `confidentiality = H`

Vulnerable System Integrity — Low

from the modelled C/I/A of the vulnerable system's rated postconditions

Low on the vulnerable system, from the modelled C/I/A of the finding's postconditions: Content written into a module Resources directory by a local principal is readable by an unauthenticated LAN request, the same static-content disclosure already carried by the two linked findings is rated Low for integrity.

REASONING & EVIDENCE

Modelled C/I/A · vulnerable system — Low
From the modelled C/I/A of the finding's postconditions on the vulnerable system.
CONDITION Content written into a module Resources directory by a local principal is readable by an unauthenticated LAN request, the same static-content disclosure already carried by the two linked findings `integrity = L`

Vulnerable System Availability — Low

from the modelled C/I/A of the vulnerable system's rated postconditions

Low on the vulnerable system, from the modelled C/I/A of the finding's postconditions: Content written into a module Resources directory by a local principal is readable by an unauthenticated LAN request, the same static-content disclosure already carried by the two linked findings is rated Low for availability.

REASONING & EVIDENCE

Modelled C/I/A · vulnerable system — Low
From the modelled C/I/A of the finding's postconditions on the vulnerable system.
CONDITION Content written into a module Resources directory by a local principal is readable by an unauthenticated LAN request, the same static-content disclosure already carried by the two linked findings `availability = L`

Subsequent System Confidentiality — None

the subsequent system is not counted: 3.1 S is U

Scope is Unchanged: exploiting this flaw affects nothing beyond the vulnerable system, so there is no subsequent-system impact.

Subsequent System Integrity — None

the subsequent system is not counted: 3.1 S is U

Scope is Unchanged: exploiting this flaw affects nothing beyond the vulnerable system, so there is no subsequent-system impact.

Subsequent System Availability — None

the subsequent system is not counted: 3.1 S is U

Scope is Unchanged: exploiting this flaw affects nothing beyond the vulnerable system, so there is no subsequent-system impact.

EL-pnhk2-wsYgt ADB Backup Enabled

CVSS V4.0 **4.8** MEDIUM

`CVSS:4.0/AV:L/AC:L/AT:N/PR:L/UI:N/VC:N/VI:N/VA:L/SC:N/SI:N/SA:N`

Attack Vector — Local

The vulnerability is exploitable by logging on interactively to the Windows Single Board Computer (SBC), which indicates local access to the device.

Attack Complexity — Low

The attacker only needs an authenticated Windows standard-user session on the instrument SBC, which grants INTERACTIVE group membership, allowing them to run 'sc stop' commands without additional special conditions or configurations beyond logging on interactively or via RDP at the Touchscreen kiosk console or Remote Desktop Protocol interface.

REASONING & EVIDENCE

> No
>
> The vulnerability arises from permissive service control rights granted to the INTERACTIVE group, with no indication that the attacker must defeat or evade any built-in security-hardening mechanisms such as ASLR, DEP, code signing, or integrity validation to exploit the flaw; the attack is simply stopping the service using standard Windows service control commands.
>
> INTELLIGENCE Finding description *"The INTERACTIVE ACE omits DC (ChangeConfig) and SD (Delete), which bounds the capability to start, stop, and pause denial of service rather than reconfiguration or a binary-replacement hijack."*
>
> INTELLIGENCE Finding description *"The destructive stop itself was gated and never executed under the read-only rules of engagement. The decoded DACL is dispositive proof of the capability on its own."*

Attack Requirements — None

The attacker only needs an authenticated Windows standard-user session on the instrument SBC, which grants INTERACTIVE group membership, allowing them to run 'sc stop' commands without additional special conditions or configurations beyond logging on interactively or via RDP at the Touchscreen kiosk console or Remote Desktop Protocol interface.

REASONING & EVIDENCE

> Yes
>
> The attacker only needs an authenticated Windows standard-user session on the instrument SBC, which grants INTERACTIVE group membership, allowing them to run 'sc stop' commands at will against the Local Run Manager Analysis Service without additional special conditions or configurations beyond logging on interactively or via RDP at the Touchscreen kiosk console or Remote Desktop Protocol interface.
>
> INTELLIGENCE Finding description *"Any principal who logs on interactively, whether at the console or over RDP, acquires the INTERACTIVE group in its token."*
>
> INTELLIGENCE Finding description *"The standard account was confirmed to belong only to Users and Remote Desktop Users and to be absent from Administrators, so it holds SERVICE_STOP on this service the moment it is logged on."*

> No
>
> The attack requires an authenticated Windows standard-user session on the instrument SBC, which the attacker can obtain by logging on interactively or via RDP through the Touchscreen kiosk console or the RDP interface. No special deployment or configuration condition outside the attacker's control is required for exploitation.
>
> INTELLIGENCE Finding description *"Any principal who logs on interactively, whether at the console or over RDP, acquires the INTERACTIVE group in its token."*

> No
>
> The attacker can start the attack whenever they choose by logging on interactively or via RDP and running the 'sc stop' command; no race condition or timing dependency outside the attacker's control is required.
>
> INTELLIGENCE Finding description *"From that session a `sc stop` or `net stop` against the service would succeed against this ACL."*

Privileges Required — Low

aliased from 3.1/PR

The Local Run Manager Analysis Service runs on the Microsoft Windows 10 Operating System which supports multiple user accounts and roles, including standard users and administrators. The attacker must be authorized by logging on interactively or via RDP to acquire INTERACTIVE group membership, which grants SERVICE_STOP rights on the service, but does not require administrator or system-level privileges (authenticated Windows standard-user session).

REASONING & EVIDENCE

> Yes
>
> The Local Run Manager Analysis Service runs on the Microsoft Windows 10 Operating System which supports multiple user accounts and roles with different privilege levels, including standard users and administrators, as shown by the

> INTERACTIVE group membership and the standard account 'sbsuser' being a member of Users and Remote Desktop Users but not Administrators. This indicates an authorization model with multiple roles. The vulnerable component is the Local Run Manager Analysis Service (cmp:01a10ed2-e8cf-7f97-8f22-843e10396fc2) running on the Windows 10 Operating System (cmp:01a10ed2-e7f8-70a8-8a37-a94f97370d99).
> INTELLIGENCE Finding description *"Any principal who logs on interactively, whether at the console or over RDP, acquires the INTERACTIVE group in its token. The standard account was confirmed to belong only to Users and Remote Desktop Users and to be absent from Administrators"*

> Yes
> The attacker must be authorized by logging on interactively or via RDP to acquire the INTERACTIVE group membership in their token, which grants the SERVICE_STOP rights on the Local Run Manager Analysis Service. The attack requires a valid authenticated session on the Windows 10 Operating System through the Touchscreen kiosk console or RDP network services. The attack vector is local but requires prior authentication (Windows OS network services (TCP/IP, RDP/SMB)).
> INTELLIGENCE Summary

> No
> The attacker only needs to be a standard user with interactive logon rights (member of Users and Remote Desktop Users) and does not require administrator or system-level privileges to stop the Local Run Manager Analysis Service. The service control rights are granted to the INTERACTIVE group, which includes non-administrative users.
> INTELLIGENCE Finding description *"The standard account was confirmed to belong only to Users and Remote Desktop Users and to be absent from Administrators, so it holds SERVICE_STOP on this service the moment it is logged on."*

User Interaction None

The attacker must be an authenticated user who logs on interactively to the Touchscreen kiosk console, and then runs the 'sc stop' command to stop the Local Run Manager Analysis Service, which does not require a separate victim user to interact with the device.

Vulnerable System Confidentiality None

from the modelled C/I/A of the vulnerable system's rated postconditions

None on the vulnerable system: none of the finding's postconditions carries a confidentiality rating.

REASONING & EVIDENCE

> Modelled C/I/A · vulnerable system None
> From the modelled C/I/A of the finding's postconditions on the vulnerable system.

Vulnerable System Integrity None

from the modelled C/I/A of the vulnerable system's rated postconditions

None on the vulnerable system: none of the finding's postconditions carries a integrity rating.

REASONING & EVIDENCE

> Modelled C/I/A · vulnerable system None
> From the modelled C/I/A of the finding's postconditions on the vulnerable system.

Vulnerable System Availability Low

from the modelled C/I/A of the vulnerable system's rated postconditions

Low on the vulnerable system, from the modelled C/I/A of the finding's postconditions: Attacker halts the Local Run Manager analysis pipeline, denying availability of the sequencing-analysis and result-generation workflow is rated Low for availability.

REASONING & EVIDENCE

> Modelled C/I/A · vulnerable system Low
> From the modelled C/I/A of the finding's postconditions on the vulnerable system.
> CONDITION Attacker halts the Local Run Manager analysis pipeline, denying availability of the sequencing-analysis and result-generation workflow `availability = L`

Subsequent System Confidentiality None

the subsequent system is not counted: 3.1 S is U

Scope is Unchanged: exploiting this flaw affects nothing beyond the vulnerable system, so there is no subsequent-system impact.

Subsequent System Integrity None

the subsequent system is not counted: 3.1 S is U

Scope is Unchanged: exploiting this flaw affects nothing beyond the vulnerable system, so there is no subsequent-system impact.

Subsequent System Availability None

the subsequent system is not counted: 3.1 S is U

Scope is Unchanged: exploiting this flaw affects nothing beyond the vulnerable system, so there is no subsequent-system impact.

EL-46wuf-PLEse Password Strength Policy Not Enforced in the component

CVSS V4.0 **4.6** MEDIUM

`CVSS:4.0/AV:N/AC:L/AT:N/PR:H/UI:A/VC:N/VI:L/VA:N/SC:N/SI:N/SA:N`

Attack Vector Network

aliased from 3.1/AV

The vulnerability is exploitable by sending inbound network traffic to the Windows OS network services interface on the Windows Single Board Computer (SBC) hosting the SeqraDx Windows OS (Windows OS network services (TCP/IP, RDP/SMB)).

REASONING & EVIDENCE

> Yes
>
> The vulnerability is exploitable by sending inbound network traffic to the Windows OS network services interface, which is a Network category interface on the Windows Single Board Computer (SBC) hosting the SeqraDx Windows OS (Windows OS network services (TCP/IP, RDP/SMB)).
>
> INTELLIGENCE Finding description

> Yes
>
> The attack vector uses the Windows OS network services interface which includes TCP/IP protocols such as RDP and SMB, indicating OSI layer 3 or 4 network protocols are used (Windows OS network services (TCP/IP, RDP/SMB)).
>
> INTELLIGENCE Finding description

Attack Complexity Low

The attacker can exploit the vulnerability at will by sending inbound network traffic to the Windows OS network services interface, with no special conditions or evasion of security mechanisms required on the SeqraDx Windows OS (Windows OS network services (TCP/IP, RDP/SMB)).

REASONING & EVIDENCE

> No
>
> There is no indication that the attacker must defeat or evade any built-in security-hardening mechanisms such as ASLR, DEP, code signing, or certificate pinning to exploit the vulnerability; the issue is due to orphaned firewall rules that become active if the agent binary is reinstalled, without any mention of evasion requirements on the SeqraDx Windows OS.
>
> INTELLIGENCE Finding description *“The exposure is latent and tied to privileged action. If an administrator reinstalls the agent to that exact path, the pre-existing rules immediately grant it unrestricted inbound reachability across all profiles including Public, with no port scoping and no source-address restriction.”*

Attack Requirements None

The attacker can attempt to exploit the vulnerability at will by sending inbound network traffic to the Windows OS network services interface once the agent binary is installed by an administrator, with no special deployment or timing conditions required (Windows OS network services (TCP/IP, RDP/SMB), SeqraDx Windows OS).

REASONING & EVIDENCE

> Yes
>
> The attacker can attempt to exploit the vulnerability at will by sending inbound network traffic to the Windows OS network services interface, as the firewall rules would admit traffic immediately if the referenced binary were present and listening, without requiring any special conditions beyond holding the necessary privileges.
>
> INTELLIGENCE Finding description

> No
>
> The attacker only needs the administrator to have installed the agent binary previously; this is a condition the attacker can rely on but does not require a non-default deployment or configuration outside the attacker's control, as the administrator action is part of normal system management and not a special or unusual deployment condition.
>
> INTELLIGENCE Finding description *“If an administrator reinstalls the agent to that exact path, the pre-existing rules immediately grant it unrestricted inbound reachability across all profiles including Public, with no port scoping and no source-address restriction.”*

No
The attacker can initiate the attack at any time by sending inbound network traffic to the Windows OS network services interface; no race condition or timing dependency is required to exploit the vulnerability once the agent is installed (Windows OS network services (TCP/IP, RDP/SMB)).

Privileges Required High

aliased from 3.1/PR

Exploitation requires Administrator (level 4) privileges on the SeqraDx Windows OS to reinstall the agent binary and activate the firewall rules, as only administrators can write to the program path on the Windows Single Board Computer (SBC).

REASONING & EVIDENCE

Yes
The vulnerable component, SeqraDx Windows OS, is an operating system with an authorization model that supports multiple users and roles with different privilege levels, including Administrator (level 4) and Standard user roles.
INTELLIGENCE Summary *"A non-administrator cannot activate them: `icacls C:\Program Files\Common Files` shows `BUILTIN\Users:(RX)` only with no write or create right, and the `Acronis` subdirectory does not exist, so an unprivileged user cannot drop the binary to activate the rules."*

Yes
Exploiting the vulnerability requires the attacker to have administrator privileges on the SeqraDx Windows OS to reinstall the agent binary and activate the firewall rules, as non-administrators cannot write to the required program path.
INTELLIGENCE Summary *"A non-administrator cannot activate them: `icacls C:\Program Files\Common Files` shows `BUILTIN\Users:(RX)` only with no write or create right, and the `Acronis` subdirectory does not exist, so an unprivileged user cannot drop the binary to activate the rules."*
INTELLIGENCE Finding description *"The exposure is latent and tied to privileged action. If an administrator reinstalls the agent to that exact path, the pre-existing rules immediately grant it unrestricted inbound reachability."*

Yes
Administrator-level privileges on the SeqraDx Windows OS are required to write to the program path and reinstall the agent binary, as the directory permissions restrict write access to administrators and TrustedInstaller only.
INTELLIGENCE Finding description *"Running `icacls C:\Program Files\Common Files` shows `BUILTIN\Users:(RX)` only, with no write or create right, and the `Acronis` subdirectory does not exist, so a non-administrator cannot drop the binary into the referenced path."*

User Interaction Active DEFAULT

The answers did not settle this metric, so it takes the rubric's conservative value. Not settled: XUIA.

An administrator must reinstall the agent binary on the SeqraDx Windows OS to activate the firewall rules; without this action, the orphaned rules admit no traffic. User interaction by a normal user is not applicable.

REASONING & EVIDENCE

Yes
An administrator, a separate individual from the attacker, must reinstall the agent binary to activate the firewall rules and enable the vulnerability on the SeqraDx Windows OS. Without this action, the orphaned firewall rules admit no traffic because the binary is absent.
INTELLIGENCE Finding description *"The exposure is latent and tied to privileged action. If an administrator reinstalls the agent to that exact path, the pre-existing rules immediately grant it unrestricted inbound reachability across all profiles including Public"*

Not answered
User interaction is not applicable because the required action is by an administrator, not a normal user operating the device.

Vulnerable System Confidentiality None

from the modelled C/I/A of the vulnerable system's rated postconditions

None on the vulnerable system: none of the finding's postconditions carries a confidentiality rating.

REASONING & EVIDENCE

Modelled C/I/A · vulnerable system None
From the modelled C/I/A of the finding's postconditions on the vulnerable system.

Vulnerable System Integrity Low

from the modelled C/I/A of the vulnerable system's rated postconditions

Low on the vulnerable system, from the modelled C/I/A of the finding's postconditions: Any binary placed at the referenced path gains unrestricted inbound reachability on all firewall profiles including Public, bypassing the per-service scoping applied to the rest of the host is rated Low for integrity.

REASONING & EVIDENCE

Modelled C/I/A · vulnerable system Low
From the modelled C/I/A of the finding's postconditions on the vulnerable system.
CONDITION Any binary placed at the referenced path gains unrestricted inbound reachability on all firewall profiles including Public, bypassing the per-service scoping applied to the rest of the host `integrity = L`

Vulnerable System Availability None
from the modelled C/I/A of the vulnerable system's rated postconditions
None on the vulnerable system: none of the finding's postconditions carries a availability rating.

REASONING & EVIDENCE

Modelled C/I/A · vulnerable system None
From the modelled C/I/A of the finding's postconditions on the vulnerable system.

Subsequent System Confidentiality None
the subsequent system is not counted: 3.1 S is U
Scope is Unchanged: exploiting this flaw affects nothing beyond the vulnerable system, so there is no subsequent-system impact.

Subsequent System Integrity None
the subsequent system is not counted: 3.1 S is U
Scope is Unchanged: exploiting this flaw affects nothing beyond the vulnerable system, so there is no subsequent-system impact.

Subsequent System Availability None
the subsequent system is not counted: 3.1 S is U
Scope is Unchanged: exploiting this flaw affects nothing beyond the vulnerable system, so there is no subsequent-system impact.

EL-d2hJb-LnZs4 USB Permission Broadcast Uses Placeholder Action

CVSS V4.0 **5.8** MEDIUM

`CVSS:4.0/AV:L/AC:L/AT:P/PR:L/UI:N/VC:N/VI:L/VA:H/SC:N/SI:N/SA:N`

Attack Vector Local
aliased from 3.1/AV
The attack exploits local file system conditions accessible via the Touchscreen kiosk console and USB External port, both local interfaces on the device, not a network or communication protocol.

REASONING & EVIDENCE

No
The attack exploits local file system conditions accessible via the Touchscreen kiosk console and USB External port, both local interfaces on the device, not a network or communication protocol.
INTELLIGENCE Summary *"The scheduled Backup Database Job never completes, so the PostgreSQL run management data and the Local Run Manager user account database have no recovery copy."*

No
The attacker does not need physical contact with the device; local access via the Touchscreen kiosk console or USB port suffices to influence the backup job environment.
INTELLIGENCE Summary

Attack Complexity Low
The vulnerability arises from missing configuration elements (backup secret and destination directory) and does not require defeating any built-in security hardening mechanisms on the Local Run Manager (LRM) or the PostGres Database (LRM Database).

REASONING & EVIDENCE

No
The vulnerability arises from missing configuration elements (backup secret and destination directory) and does not require defeating any built-in security hardening mechanisms such as ASLR, DEP, code signing, or certificate pinning on the Local Run Manager (LRM) or the PostGres Database (LRM Database).
INTELLIGENCE Test case: Database backup control is non-functional, leaving clinical data without a recovery copy *"The job guard refuses to proceed without the secret, so no write to the destination was ever attempted during this period."*

INTELLIGENCE Remediation *"Create that exact directory and grant write access only to the service account that runs `MAPJobService`."*

Attack Requirements Present

REASONING & EVIDENCE

Yes

The attacker can cause the Backup Database Job to fail repeatedly without any special conditions beyond local access, as the job aborts automatically on every scheduled run due to missing secret or missing directory, with no additional steps or rare configurations required on the PostGres Database (LRM Database).

INTELLIGENCE Finding description *"Each of these entries is followed by a scheduled retry, so the control reschedules itself indefinitely while never completing."*

INTELLIGENCE Remediation *"The job is configured to write to `D:\Northwind Bio\Local Run Manager\Database Backup`, but a listing of that path returns path not found, so the write aborts."*

Yes

Exploitation requires the Local Run Manager's Backup Database Job to be misconfigured with a missing backup secret and a non-existent destination directory, conditions outside the attacker's control that prevent backup completion and cause data loss on the PostGres Database (LRM Database).

INTELLIGENCE Finding description *"The job's actual behavior was reconstructed from the service log, which shows the task aborting on every scheduled run rather than producing output."*

INTELLIGENCE Finding description *"A `BackupDatabase_default_Access.txt` entry now exists in the credential store with an mtime of 2026-09-25, which addresses the missing-secret condition, but a second precondition still fails: the destination directory `D:\Northwind Bio\Local Run Manager\Database Backup` does not exist, and a directory listing of that path returns path not found."*

INTELLIGENCE Summary *"The scheduled Backup Database Job never completes, so the PostgreSQL run management data and the Local Run Manager user account database have no recovery copy."*

Privileges Required Low

aliased from 3.1/PR

The attacker must be authorized to the affected component because the backup secret is stored in a credential store accessible by standard users, and the backup job runs under a service account with specific permissions, indicating authorization is required before exploiting the vulnerability on the Local Run Manager (LRM) and PostGres Database (LRM Database).

REASONING & EVIDENCE

Yes

The Local Run Manager (LRM) is a web-application with a standard user trust level, and the PostGres Database (LRM Database) is a database component with a privileged user trust level, indicating multiple roles and privilege levels exist. The backup secret is stored in a credential store accessible by standard users, and the backup job runs under a service account, showing an authorization model with different privilege levels.

INTELLIGENCE Finding description *"The backup secret, once set, is held in the same standard-user-readable credential store documented separately"*

Yes

The attacker must be authorized to the affected component because the backup secret is stored in a credential store that requires standard user access, and the backup job runs under a service account with specific permissions, indicating authorization is required before exploiting the vulnerability.

INTELLIGENCE Finding description *"The backup secret, once set, is held in the same standard-user-readable credential store"*

No

The backup secret is accessible to standard users via the credential store, and the backup job runs under a service account with permissions distinct from administrator or system-level privileges, indicating that system-level privileges are not required to exploit the vulnerability.

INTELLIGENCE Finding description *"The backup secret, once set, is held in the same standard-user-readable credential store"*

User Interaction None

The backup job fails automatically on each scheduled run due to missing secret or missing directory without requiring any action or interaction from a separate user; the attacker can cause the failure alone via local access through the Touchscreen kiosk console or USB External port on the Windows Single Board Computer.

REASONING & EVIDENCE

No

The backup job fails automatically on each scheduled run due to missing secret or missing directory without requiring any action or interaction from a separate user; the attacker can cause the failure alone via local access through the touchscreen kiosk console or USB port on the Windows Single Board Computer.

INTELLIGENCE Finding description *"The job's actual behavior was reconstructed from the service log, which shows the task aborting on every scheduled run rather than producing output."*

INTELLIGENCE Finding description

Vulnerable System Confidentiality None

from the modelled C/I/A of the vulnerable system's rated postconditions

None on the vulnerable system: none of the finding's postconditions carries a confidentiality rating.

REASONING & EVIDENCE

Modelled C/I/A · vulnerable system None
From the modelled C/I/A of the finding's postconditions on the vulnerable system.

Vulnerable System Integrity Low

from the modelled C/I/A of the vulnerable system's rated postconditions

Low on the vulnerable system, from the modelled C/I/A of the finding's postconditions: non-functional database backup leaves no recovery copy and A destructive or corrupting event against the Local Run Manager database has no recovery copy, causing permanent loss of clinical run management data and the user account database are rated Low for integrity.

REASONING & EVIDENCE

Modelled C/I/A · vulnerable system Low
From the modelled C/I/A of the finding's postconditions on the vulnerable system.
CONDITION non-functional database backup leaves no recovery copy `integrity = L`
CONDITION A destructive or corrupting event against the Local Run Manager database has no recovery copy, causing permanent loss of clinical run management data and the user account database `integrity = L`

Vulnerable System Availability High

from the modelled C/I/A of the vulnerable system's rated postconditions

High on the vulnerable system, from the modelled C/I/A of the finding's postconditions: non-functional database backup leaves no recovery copy is rated High for availability.

REASONING & EVIDENCE

Modelled C/I/A · vulnerable system High
From the modelled C/I/A of the finding's postconditions on the vulnerable system.
CONDITION non-functional database backup leaves no recovery copy `availability = H`

Subsequent System Confidentiality None

the subsequent system is not counted: 3.1 S is U

Scope is Unchanged: exploiting this flaw affects nothing beyond the vulnerable system, so there is no subsequent-system impact.

Subsequent System Integrity None

the subsequent system is not counted: 3.1 S is U

Scope is Unchanged: exploiting this flaw affects nothing beyond the vulnerable system, so there is no subsequent-system impact.

Subsequent System Availability None

the subsequent system is not counted: 3.1 S is U

Scope is Unchanged: exploiting this flaw affects nothing beyond the vulnerable system, so there is no subsequent-system impact.

EL-TSK83-DhkZp Cross-Site Request Forgery Condition Present in the component

CVSS V4.0 **2.0** LOW

`CVSS:4.0/AV:L/AC:L/AT:P/PR:L/UI:N/VC:L/VI:L/VA:L/SC:N/SI:N/SA:N`

Attack Vector Local

aliased from 3.1/AV

The attack requires a local code-execution foothold on the Microsoft Windows 10 Operating System and a memory-corruption primitive in one of the x86 processes, with no network or physical interface carrying the exploit input directly, as described in the agreed attack narrative and supported by the Local attack method.

REASONING & EVIDENCE

No

The attack requires a local code-execution foothold on the Microsoft Windows 10 Operating System and a memory-corruption primitive in one of the x86 processes, with no network or physical interface carrying the exploit input directly, as described in the agreed attack narrative and supported by the Local attack method.

INTELLIGENCE Finding description *"Exploitation is gated on both a local code-execution foothold and a separate memory-corruption primitive in one of the x86 processes; no such primitive was demonstrated here, so this is a mitigation-hardening gap rather than an independently exploitable defect."*

No

The attacker does not need physical contact with the device but requires local code execution on the Microsoft Windows 10 Operating System, which can be achieved via local access without physical manipulation.

INTELLIGENCE Finding description *"Exploitation is gated on both a local code-execution foothold and a separate memory-corruption primitive in one of the x86 processes; no such primitive was demonstrated here, so this is a mitigation-hardening gap rather than an independently exploitable defect."*

Attack Complexity Low

The vulnerability is due to a per-binary build-setting weakness in ASLR entropy on 32-bit images, with system ASLR and DEP enabled OS-wide; no built-in security-hardening mechanism such as ASLR, DEP, CFG, or certificate pinning must be actively defeated or evaded by the attacker (Microsoft Windows 10 Operating System).

REASONING & EVIDENCE

No

The vulnerability is due to a per-binary build-setting weakness in ASLR entropy on 32-bit images, with system ASLR and DEP enabled OS-wide; no built-in security-hardening mechanism such as ASLR, DEP, CFG, or certificate pinning must be actively defeated or evaded by the attacker.

INTELLIGENCE Summary *"System ASLR and DEP are enabled OS-wide, so this is a per-binary build-setting gap rather than an OS-policy failure."*

Attack Requirements Present

Exploitation requires the attacker to already hold a local code-execution foothold on the Microsoft Windows 10 Operating System and to find a separate memory-corruption primitive in one of the x86 processes, so the attacker cannot exploit the vulnerability at will without these special preconditions.

REASONING & EVIDENCE

No

Exploitation requires the attacker to already hold a local code-execution foothold on the Microsoft Windows 10 Operating System and to find a separate memory-corruption primitive in one of the x86 processes, so the attacker cannot exploit the vulnerability at will without these special preconditions.

INTELLIGENCE Finding description *"Exploitation is gated on both a local code-execution foothold and a separate memory-corruption primitive in one of the x86 processes."*

Privileges Required Low

aliased from 3.1/PR

The attack requires a local code-execution foothold on the Microsoft Windows 10 Operating System with Standard user trust level, which is below Administrator or system-level privileges, and the OS supports multiple user accounts and roles as indicated by the trust levels including Standard user, Privileged user, Administrator, and Superuser (level 5).

REASONING & EVIDENCE

Yes

The vulnerable component is the Microsoft Windows 10 Operating System, which supports multiple user accounts and roles with different privilege levels, as indicated by the trust levels including Standard user, Privileged user, Administrator, and Superuser (level 5). The attack requires a local code-execution foothold on this OS, implying the presence of an authorization model with multiple users or roles.

INTELLIGENCE Finding description *"Exploitation is gated on both a local code-execution foothold and a separate memory-corruption primitive in one of the x86 processes."*

Yes

The attacker must already have a local code-execution foothold on the Microsoft Windows 10 Operating System before exploiting the vulnerability, which means the attacker must be authorized (authenticated) on the affected component prior to exploitation.

INTELLIGENCE Summary *"Exploitation is gated on an attacker already holding a local code-execution foothold plus a separate memory-corruption primitive in one of the x86 processes."*

No

The attacker needs a local code-execution foothold on the Microsoft Windows 10 Operating System with Standard user trust level, which is below Administrator or system-level privileges. There is no indication that system-level privileges are required to exploit the vulnerability.

INTELLIGENCE Finding description *"Exploitation is gated on an attacker already holding a local code-execution foothold plus a separate memory-corruption primitive in one of the x86 processes; no such primitive was demonstrated, so this is a mitigation-hardening gap rather than an independently exploitable defect."* *(paraphrased)*

User Interaction None

The attack requires only that the attacker already has a local code-execution foothold on the Microsoft Windows 10 Operating System and can exploit a memory-corruption primitive in one of the x86 processes; no other user or victim must perform any action or interact with the system for the attack to succeed.

REASONING & EVIDENCE

No

The attack requires only that the attacker already has a local code-execution foothold on the Microsoft Windows 10 Operating System and can exploit a memory-corruption primitive in one of the x86 processes; no other user or victim must perform any action or interact with the system for the attack to succeed.

INTELLIGENCE Finding description *"Exploitation is gated on both a local code-execution foothold and a separate memory-corruption primitive in one of the x86 processes."*

Vulnerable System Confidentiality Low

No PHI/PII data is modelled on the Microsoft Windows 10 Operating System or its descendants, but diagnosis and monitoring data such as Sequencing Image Files, Results Report Files (VCF, PDF, JSON, Variants.txt), clinical workflow data such as RabbitMQ Queue Messages, system-user data such as LRM User Credentials, and other critical data such as Audit Trail Records and Windows Event Logs with low confidentiality ratings can be read by an attacker exploiting the mitigation-hardening gap.

REASONING & EVIDENCE

No

No PHI/PII data is modelled on the Microsoft Windows 10 Operating System or its descendants.

Low

The vulnerable system holds diagnosis and monitoring data such as Sequencing Image Files and Results Report Files (VCF, PDF, JSON, Variants.txt) with low confidentiality, integrity, and availability ratings, which can be read by an attacker exploiting the mitigation-hardening gap in the Windows 10 Operating System.

INTELLIGENCE Finding description *"Exploitation is gated on both a local code-execution foothold and a separate memory-corruption primitive in one of the x86 processes."*

No

No therapy delivery data or functionality is modelled on the Microsoft Windows 10 Operating System or its descendants.

Low

Clinical workflow data such as RabbitMQ Queue Messages with low confidentiality, integrity, and availability ratings can be read by an attacker exploiting the mitigation-hardening gap in the Windows 10 Operating System.

INTELLIGENCE Finding description *"Exploitation is gated on both a local code-execution foothold and a separate memory-corruption primitive in one of the x86 processes."*

Low

System-user data such as LRM User Credentials with high confidentiality and low integrity and availability can be read by an attacker exploiting the mitigation-hardening gap, potentially compromising user credentials.

INTELLIGENCE Finding description *"Exploitation is gated on both a local code-execution foothold and a separate memory-corruption primitive in one of the x86 processes."*

Low

Other critical data such as Audit Trail Records and Windows Event Logs with low confidentiality, integrity, and availability can be read by an attacker exploiting the mitigation-hardening gap.

INTELLIGENCE Finding description *"Exploitation is gated on both a local code-execution foothold and a separate memory-corruption primitive in one of the x86 processes."*

Vulnerable System Integrity Low

Diagnosis and monitoring data such as Sequencing Image Files and Results Report Files (VCF, PDF, JSON, Variants.txt) can be modified or deleted by an attacker exploiting the mitigation-hardening gap, as can clinical workflow data such as RabbitMQ Queue Messages, system-user data such as LRM User Credentials, and other critical data such as Audit Trail Records and Windows Event Logs, all with low integrity ratings and low clinical impact.

REASONING & EVIDENCE

No

No PHI/PII data is modelled on the Microsoft Windows 10 Operating System or its descendants.

Low

Diagnosis and monitoring data such as Sequencing Image Files and Results Report Files (VCF, PDF, JSON, Variants.txt) can be modified or deleted by an attacker exploiting the mitigation-hardening gap, potentially affecting data integrity but with low clinical impact.

INTELLIGENCE Finding description *"Exploitation is gated on both a local code-execution foothold and a separate memory-corruption primitive in one of the x86 processes."*

No

No therapy delivery data or functionality is modelled on the Microsoft Windows 10 Operating System or its descendants.

Low

Clinical workflow data such as RabbitMQ Queue Messages can be modified or deleted by an attacker exploiting the mitigation-hardening gap, potentially disrupting workflow with low clinical impact.

INTELLIGENCE Finding description *"Exploitation is gated on both a local code-execution foothold and a separate memory-corruption primitive in one of the x86 processes."*

Low

System-user data such as LRM User Credentials can be modified or deleted by an attacker exploiting the mitigation-hardening gap, potentially compromising system integrity with low clinical impact.

INTELLIGENCE Finding description *"Exploitation is gated on both a local code-execution foothold and a separate memory-corruption primitive in one of the x86 processes."*

Low

Other critical data such as Audit Trail Records and Windows Event Logs can be modified or deleted by an attacker exploiting the mitigation-hardening gap, potentially affecting audit integrity with low clinical impact.

INTELLIGENCE Finding description *"Exploitation is gated on both a local code-execution foothold and a separate memory-corruption primitive in one of the x86 processes."*

Vulnerable System Availability Low

Diagnosis and monitoring data such as Sequencing Image Files and Results Report Files (VCF, PDF, JSON, Variants.txt) can be rendered inaccessible by an attacker exploiting the mitigation-hardening gap, as can clinical workflow data such as RabbitMQ Queue Messages, system-user data such as LRM User Credentials, and other critical data such as Audit Trail Records and Windows Event Logs, all with low availability ratings and low clinical impact.

REASONING & EVIDENCE

No

No PHI/PII data is modelled on the Microsoft Windows 10 Operating System or its descendants.

Low

Diagnosis and monitoring data such as Sequencing Image Files and Results Report Files (VCF, PDF, JSON, Variants.txt) can be rendered inaccessible by an attacker exploiting the mitigation-hardening gap, which may delay or prevent access to monitoring data with low clinical impact.

INTELLIGENCE Finding description *"Exploitation is gated on both a local code-execution foothold and a separate memory-corruption primitive in one of the x86 processes."*

No

No therapy delivery data or functionality is modelled on the Microsoft Windows 10 Operating System or its descendants.

Low

Clinical workflow data such as RabbitMQ Queue Messages can be rendered inaccessible by an attacker exploiting the mitigation-hardening gap, potentially delaying or preventing workflow operations with low clinical impact.

INTELLIGENCE Finding description *"Exploitation is gated on both a local code-execution foothold and a separate memory-corruption primitive in one of the x86 processes."*

Low

System-user data such as LRM User Credentials can be rendered inaccessible by an attacker exploiting the mitigation-hardening gap, potentially disrupting system access with low clinical impact.

INTELLIGENCE Finding description *"Exploitation is gated on both a local code-execution foothold and a separate memory-corruption primitive in one of the x86 processes."*

Low

Other critical data such as Audit Trail Records and Windows Event Logs can be rendered inaccessible by an attacker exploiting the mitigation-hardening gap, potentially disrupting audit and logging functions with low clinical impact.

INTELLIGENCE Finding description *"Exploitation is gated on both a local code-execution foothold and a separate memory-corruption primitive in one of the x86 processes."*

Subsequent System Confidentiality None

the subsequent system is not counted: 3.1 S is U

Scope is Unchanged: exploiting this flaw affects nothing beyond the vulnerable system, so there is no subsequent-system impact.

Subsequent System Integrity None

the subsequent system is not counted: 3.1 S is U

Scope is Unchanged: exploiting this flaw affects nothing beyond the vulnerable system, so there is no subsequent-system impact.

Subsequent System Availability None

the subsequent system is not counted: 3.1 S is U

Scope is Unchanged: exploiting this flaw affects nothing beyond the vulnerable system, so there is no subsequent-system impact.

EL-NEEFc-AeKLA Privileged Service Token Exposed on the component

CVSS V4.0 **5.3** MEDIUM

`CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:L/VI:L/VA:N/SC:N/SI:N/SA:N`

Attack Vector Network

aliased from 3.1/AV

The attack is carried out over the network via HTTPS requests to the Local Run Manager HTTPS endpoint (TCP/443, TLSv1.2) and the Local Run Manager API on the Local Run Manager (LRM) component.

REASONING & EVIDENCE

Yes

The attack is carried out over the network via HTTPS requests to the Local Run Manager API endpoint on TCP port 443, which is a network interface on the Local Run Manager (LRM) component (Local Run Manager HTTPS endpoint (TCP/443, TLSv1.2)).

INTELLIGENCE Summary *“Prosecuted live from the device against the running LRM on TCP 443 with a single salted challenge-response login as the standard account”*

Yes

The Local Run Manager HTTPS endpoint uses TCP/443 with TLSv1.2, indicating the use of OSI layer 3/4 protocols such as TCP/IP (Local Run Manager HTTPS endpoint (TCP/443, TLSv1.2)).

Attack Complexity Low

Once authenticated as a standard non-admin user with the NotLimitedUser role, the attacker can repeatedly exploit the under-gated administrative functions on the Local Run Manager (LRM) web application without any additional special conditions or configurations, and no security-hardening mechanisms need to be bypassed.

REASONING & EVIDENCE

No

The vulnerability arises from missing explicit administrator role checks in the Local Run Manager's service layer authorization, without any need to defeat or evade built-in security-hardening mechanisms such as ASLR, DEP, code signing, or certificate pinning.

INTELLIGENCE Finding description *“Several of their state-changing operations carry no per-route administrator attribute of their own, so they are reachable by any non-limited user”*

Attack Requirements None

The attack works against the Local Run Manager (LRM) as it is normally deployed and run, with no special configuration or peripheral required beyond a standard authenticated non-admin user session, which is considered under Privileges Required and not an attack requirement.

REASONING & EVIDENCE

Yes

Once authenticated as a standard non-admin user with the NotLimitedUser role, the attacker can repeatedly exploit the under-gated administrative functions on the Local Run Manager (LRM) web application without any additional special conditions or configurations.

INTELLIGENCE Finding description *“Several of their state-changing operations carry no per-route administrator attribute of their own, so they are reachable by any non-limited user”*

INTELLIGENCE Finding description *“The under-gated operations were reached by the same session”*

No

The attack works against the Local Run Manager (LRM) as it is normally deployed and run, with no special configuration or peripheral required beyond a standard authenticated non-admin user session, which is considered under Privileges Required and not an attack requirement, as shown by the standarduser account accessing under-gated administrative functions on the LRM component.

INTELLIGENCE Summary *"A standard, non-admin Local Run Manager account can invoke administrative configuration operations that are supposed to be restricted to administrators."*

INTELLIGENCE Test case: Vertical function-level authorization on LRM administrative configuration operations *"Prosecuted live from the device against the running LRM on TCP 443 with a single salted challenge-response login as the standard account"*

No

The attacker can start the attack whenever they choose by sending HTTP requests over the network to the Local Run Manager HTTPS endpoint; there is no indication that the attack requires winning a race or timing conditions outside the attacker's control (Local Run Manager HTTPS endpoint (TCP/443, TLSv1.2)).

INTELLIGENCE Summary *"Prosecuted live from the device against the running LRM on TCP 443 with a single salted challenge-response login as the standard account"*

Privileges Required Low

aliased from 3.1/PR

The attacker must authenticate as a standard non-admin user with the NotLimitedUser role but not Admin on the Local Run Manager (LRM) web application to exploit the vulnerability, requiring valid user credentials but no administrator privileges (authenticated LRM standard-user web session).

REASONING & EVIDENCE

Yes

The Local Run Manager (LRM) web application uses a role-based authorization model where users have roles such as NotLimitedUser and Admin, and the attacker must authenticate as a standard non-admin user with the NotLimitedUser role but not Admin, as shown by the role construction and authorization gates in the LRM service layer.

INTELLIGENCE Finding description *"A standard user that has confirmed a non-expired password is assigned the roles Enabled, PasswordConfirmed and NotLimitedUser, but never Admin (role construction at Northwind Bio.MAP.Domain.cs:10306-10350). The SystemService and SystemSettingService classes both derive from ProtectedService, whose only class-level gate is the floor attribute RequiredRole("NotLimitedUser") at Northwind Bio.MAP.Api.Service.cs:1089-1091."*

Yes

The attacker must be authorized by authenticating as a standard user with a confirmed non-expired password to obtain the NotLimitedUser role before accessing the vulnerable administrative functions, as demonstrated by the successful authenticated requests and role-based authorization checks in the LRM (Local Run Manager (LRM)).

INTELLIGENCE Finding description

No

The attacker only needs a standard non-admin user account with the NotLimitedUser role and does not require administrator, maintainer, or system-level privileges to exploit the vulnerability, as the missing administrator gates allow access to administrative functions by any authenticated non-limited user (Local Run Manager (LRM)).

INTELLIGENCE Finding description

User Interaction None

The attacker must authenticate as a standard non-administrative user with valid, non-expired credentials to obtain the NotLimitedUser role and access the vulnerable functionality on the Local Run Manager (LRM) component. However, no action or participation from another user is required to complete exploitation. The attacker can independently authenticate and interact with the Local Run Manager HTTPS endpoint (TCP/443, TLSv1.2) and associated functionality. Therefore, User Interaction is None.

Vulnerable System Confidentiality Low

from the modelled C/I/A of the vulnerable system's rated postconditions

Low on the vulnerable system, from the modelled C/I/A of the finding's postconditions: A non-admin user reads the Windows service-account identities and types the Analysis, Job and UCS services run as is rated Low for confidentiality.

REASONING & EVIDENCE

Modelled C/I/A · vulnerable system Low

From the modelled C/I/A of the finding's postconditions on the vulnerable system.

CONDITION A non-admin user reads the Windows service-account identities and types the Analysis, Job and UCS services run as `confidentiality = L`

Vulnerable System Integrity Low

from the modelled C/I/A of the vulnerable system's rated postconditions

Low on the vulnerable system, from the modelled C/I/A of the finding's postconditions: A non-admin user reaches the write path that replaces the LRM site TLS certificate and alters the system data settings, including the cloud endpoint and cloud key-file path of the clinical instrument is rated Low for integrity.

REASONING & EVIDENCE

> Modelled C/I/A · vulnerable system **Low**
> From the modelled C/I/A of the finding's postconditions on the vulnerable system.
> CONDITION A non-admin user reaches the write path that replaces the LRM site TLS certificate and alters the system data settings, including the cloud endpoint and cloud key-file path of the clinical instrument `integrity = L`

Vulnerable System Availability None

from the modelled C/I/A of the vulnerable system's rated postconditions

None on the vulnerable system: none of the finding's postconditions carries a availability rating.

REASONING & EVIDENCE

> Modelled C/I/A · vulnerable system **None**
> From the modelled C/I/A of the finding's postconditions on the vulnerable system.

Subsequent System Confidentiality None

the subsequent system is not counted: 3.1 S is U

Scope is Unchanged: exploiting this flaw affects nothing beyond the vulnerable system, so there is no subsequent-system impact.

Subsequent System Integrity None

the subsequent system is not counted: 3.1 S is U

Scope is Unchanged: exploiting this flaw affects nothing beyond the vulnerable system, so there is no subsequent-system impact.

Subsequent System Availability None

the subsequent system is not counted: 3.1 S is U

Scope is Unchanged: exploiting this flaw affects nothing beyond the vulnerable system, so there is no subsequent-system impact.

EL-U9MwD-eijMn Application Files World-Writable in the component

CVSS V4.0 **5.1** MEDIUM

`CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:A/VC:L/VI:N/VA:N/SC:N/SI:N/SA:N`

Attack Vector Network

aliased from 3.1/AV

The attack is carried out by sending a crafted HTTP request over the network to the Windows Internet Information Service (IIS) via the RJ45 Ethernet port (100/1000BaseT) on the Windows Single Board Computer (SBC).

REASONING & EVIDENCE

> **Yes**
> The attack is carried out by sending a crafted HTTP request over the network to the Windows Internet Information Service (IIS) via the RJ45 Ethernet port (100/1000BaseT), which is a network interface on the Windows Single Board Computer (SBC).
> INTELLIGENCE Finding description

> **Yes**
> The attack uses HTTP over TCP/IP on port 80, which is an OSI layer 3/4 protocol, as the IIS site listens on TCP port 80 and 443 (RJ45 Ethernet port (100/1000BaseT)).
> INTELLIGENCE Finding description *“The determination rested on inspecting the single IIS site, which is bound on `*:80` and `*:443`.”*

Attack Complexity Low

The vulnerability arises from the unvalidated reflection of the Host header in the redirect Location header without any need to defeat or evade built-in security mechanisms (Windows Internet Information Service (IIS)).

REASONING & EVIDENCE

> **No**
> The vulnerability arises from the unvalidated reflection of the Host header in the redirect Location header without any need to defeat or evade built-in security mechanisms such as ASLR, DEP, code signing, integrity validation, or certificate pinning; the IIS rewrite rule simply copies the raw Host header into the Location response header Windows Internet Information Service (IIS).

INTELLIGENCE Finding description *"The rewrite server variable `{HTTP_HOST}` resolves to the raw client Host header, and it is copied straight into the Location response header with no check against the configured `*:80`/`*:443` bindings."*

Attack Requirements None

The attacker can send crafted HTTP requests with arbitrary Host headers directly to the Windows Internet Information Service (IIS) on port 80 over the customer LAN without any special conditions or prior steps, and the attack works against the product as it ships and is normally run with no special deployment or configuration conditions required (RJ45 Ethernet port (100/1000BaseT)).

REASONING & EVIDENCE

Yes

The attacker can send crafted HTTP requests with arbitrary Host headers directly to the Windows Internet Information Service (IIS) on port 80 over the customer LAN without any special conditions or prior steps, as the service is reachable anonymously and unauthenticated on the network interface Windows Internet Information Service (IIS) via the RJ45 Ethernet port (100/1000BaseT).

INTELLIGENCE Finding description *"Exploitation needs only an anonymous network position on the customer LAN: the attacker induces a victim to send a crafted Host header to the plaintext :80 listener, and the 301 steers the victim's browser to an attacker-chosen origin."*

No

The attack works against the product as it ships and is normally run, with no special deployment or configuration conditions required beyond the default IIS site bindings on ports 80 and 443 and the default URL rewrite rule reflecting the Host header. The attacker only needs network access to the customer LAN to send crafted HTTP requests to the IIS service on the RJ45 Ethernet port (100/1000BaseT) and exploit the vulnerability in the Windows Internet Information Service (IIS).

INTELLIGENCE Finding description *"The determination rested on inspecting the single IIS site, which is bound on `*:80` and `*:443`, and tracing the inbound HTTPS-upgrade rewrite rule that fires on plaintext requests."*

No

The attacker can start the attack whenever they choose by sending a crafted HTTP request with a malicious Host header to the IIS service on port 80; there is no race condition or timing dependency outside the attacker's control. The redirect is generated immediately upon receiving the crafted request on the RJ45 Ethernet port (100/1000BaseT) by the Windows Internet Information Service (IIS).

INTELLIGENCE Finding description *"Exploitation needs only an anonymous network position on the customer LAN: the attacker induces a victim to send a crafted Host header to the plaintext :80 listener, and the 301 steers the victim's browser to an attacker-chosen origin."*

Privileges Required None

aliased from 3.1/PR

The vulnerability can be exploited by sending crafted HTTP requests with a malicious Host header to the IIS service on port 80 without any authentication or authorization, as the IIS component runs on the Windows Single Board Computer (SBC) which uses the Microsoft Windows 10 Operating System with an authorization model supporting multiple users and roles (Windows Internet Information Service (IIS)).

REASONING & EVIDENCE

Yes

The Windows Internet Information Service (IIS) runs on the Windows Single Board Computer (SBC) which uses the Microsoft Windows 10 Operating System, all of which have an authorization model supporting multiple users and roles with different privilege levels, including Privileged user and Administrator roles. The IIS component itself is assigned the Privileged user trust level, indicating it enforces authorization (SeqraDx Operating Software (MOS)).

No

The vulnerability can be exploited by sending crafted HTTP requests with a malicious Host header to the IIS service on port 80 without any authentication or authorization, as confirmed by live testing showing the redirect response is returned unauthenticated (RJ45 Ethernet port (100/1000BaseT)).

INTELLIGENCE Test case: HTTP response headers safe from CRLF/response-splitting injection *"This was proven live against the device with no authentication."*

User Interaction Active

The attacker can send crafted HTTP requests to the target. From there user interaction is required by the victim in order to hit the Windows Internet Information Service (IIS) component via the RJ45 Ethernet port (RJ45 Ethernet port (100/1000BaseT)).

Vulnerable System Confidentiality Low

from the modelled C/I/A of the vulnerable system's rated postconditions

Low on the vulnerable system, from the modelled C/I/A of the finding's postconditions: The instrument's plaintext endpoint emits a Location response header that redirects a victim to an attacker-chosen external origin, enabling credential phishing against the look-alike login page is rated Low for confidentiality.

REASONING & EVIDENCE

> Modelled C/I/A · vulnerable system **Low**
> From the modelled C/I/A of the finding's postconditions on the vulnerable system.
> CONDITION The instrument's plaintext endpoint emits a Location response header that redirects a victim to an attacker-chosen external origin, enabling credential phishing against the look-alike login page `confidentiality = L`

Vulnerable System Integrity None

from the modelled C/I/A of the vulnerable system's rated postconditions

None on the vulnerable system: none of the finding's postconditions carries a integrity rating.

REASONING & EVIDENCE

> Modelled C/I/A · vulnerable system **None**
> From the modelled C/I/A of the finding's postconditions on the vulnerable system.

Vulnerable System Availability None

from the modelled C/I/A of the vulnerable system's rated postconditions

None on the vulnerable system: none of the finding's postconditions carries a availability rating.

REASONING & EVIDENCE

> Modelled C/I/A · vulnerable system **None**
> From the modelled C/I/A of the finding's postconditions on the vulnerable system.

Subsequent System Confidentiality None

the subsequent system is not counted: 3.1 S is U

Scope is Unchanged: exploiting this flaw affects nothing beyond the vulnerable system, so there is no subsequent-system impact.

Subsequent System Integrity None

the subsequent system is not counted: 3.1 S is U

Scope is Unchanged: exploiting this flaw affects nothing beyond the vulnerable system, so there is no subsequent-system impact.

Subsequent System Availability None

the subsequent system is not counted: 3.1 S is U

Scope is Unchanged: exploiting this flaw affects nothing beyond the vulnerable system, so there is no subsequent-system impact.

EL-2Qd2G-sEefY Application Directories World-Writable on the component

CVSS V4.0 **2.4** LOW

`CVSS:4.0/AV:L/AC:L/AT:N/PR:L/UI:A/VC:L/VI:N/VA:N/SC:N/SI:N/SA:N`

Attack Vector Local

aliased from 3.1/AV

The exploit requires local or authenticated management access to the RabbitMQ ingress interface on RabbitMQ, which is bound to loopback and blocked from external network access by firewall rules.

REASONING & EVIDENCE

> **No**
> The RabbitMQ ingress interface on RabbitMQ is a local service interface reachable only on 127.0.0.1, and the management and tracing interfaces bind to loopback only with firewall rules blocking external network access, so the attacker cannot utilize a network or communication protocol to exploit this vulnerability remotely.
> INTELLIGENCE Finding description *"The connection and firewall state show AMQPS on 127.0.0.1:5671 and the management and tracing interface on 127.0.0.1:15671 bound to loopback only, plain AMQP 5672 not listening, and the Erlang distribution port scoped to the instrument link-local address, with an explicit inbound Block on 5671, 15671, and 25672 across all profiles."*

> **No**
> The exploitation requires an authenticated management user to enable tracing or read trace outputs via the management interface on 127.0.0.1:15671, which is a local network interface accessible via logon or shell access without physical contact to the device.

INTELLIGENCE Finding description

Attack Complexity Low

The RabbitMQ has the rabbitmq_tracing plugin enabled and running by default, allowing an attacker with management credentials to enable tracing or read trace outputs without needing special conditions or evading protections.

REASONING & EVIDENCE

No

No built-in security-hardening mechanism such as ASLR, DEP, code signing, integrity validation, or certificate pinning must be defeated or evaded to exploit the vulnerability; the tracing plugin is simply enabled and running, exposing message contents without requiring evasion of protections.

INTELLIGENCE Finding description *"The control is not met because an unneeded message-capture feature is enabled and running by default on the live node."*

Attack Requirements None

The tracing plugin is enabled and running by default on the live RabbitMQ node, so an attacker holding management credentials can enable tracing or read trace outputs repeatedly without further special conditions or configurations beyond holding those credentials.

REASONING & EVIDENCE

Yes

The tracing plugin is enabled and running by default on the live RabbitMQ node, so an attacker holding management credentials or local access can enable tracing or read trace outputs repeatedly without further special conditions or configurations beyond holding those credentials.

INTELLIGENCE Finding description *"The control is not met because an unneeded message-capture feature is enabled and running by default on the live node."*

INTELLIGENCE Summary *"The capture surface is therefore reachable from the local host or by a principal holding management credentials, not from an external network position."*

No

The rabbitmq_tracing plugin is enabled and running by default on the RabbitMQ, so exploitation does not require a non-default deployment or configuration condition outside the attacker's control once they have authenticated management access.

INTELLIGENCE Finding description *"The control is not met because an unneeded message-capture feature is enabled and running by default on the live node"*

No

The attacker can enable tracing or access trace outputs at any time once they have local or authenticated management access; there is no requirement to win a race or inject into an ongoing exchange, as the tracing plugin continuously captures messages when enabled on the RabbitMQ.

INTELLIGENCE Finding description *"The exploitation path is local or authenticated-management access leading to the enabled tracing firehose and then to captured inter-service message bodies"*

Privileges Required Low

aliased from 3.1/PR

Exploitation requires authentication with management credentials on the RabbitMQ ingress interface of RabbitMQ running on Microsoft Windows 10 Operating System, but does not require system-level privileges.

INTELLIGENCE Finding description

REASONING & EVIDENCE

Yes

The RabbitMQ component runs on the Windows 10 Operating System which enforces user accounts and privileges, and the management interface requires authentication with management credentials to enable tracing or read trace outputs, indicating multiple roles with different privilege levels on the component RabbitMQ, Microsoft Windows 10 Operating System, RabbitMQ ingress.

INTELLIGENCE Finding description *"The exploitation path is local or authenticated-management access leading to the enabled tracing firehose and then to captured inter-service message bodies."*

Yes

The attacker must hold valid management credentials or be an authenticated management user to enable tracing or read trace outputs via the management interface on 127.0.0.1:15671, so authorization is required before exploiting the vulnerability RabbitMQ ingress, Local Run Manager HTTPS endpoint (TCP/443, TLSv1.2).

INTELLIGENCE Finding description *"The exploitation path is local or authenticated-management access leading to the enabled tracing firehose and then to captured inter-service message bodies."*

No

The attacker needs to be an authenticated management user but does not require system-level privileges such as administrator or root on the host to enable tracing or read trace outputs; management credentials suffice RabbitMQ, [intel-1].
INTELLIGENCE Finding description *"An actor who can read either output reads the contents of the run, analysis, and job messages the Local Run Manager services exchange."*

User Interaction Active

An authenticated management user must actively enable tracing or access trace outputs via the management interface on 127.0.0.1:15671, requiring deliberate action beyond normal operation.
INTELLIGENCE Finding description

REASONING & EVIDENCE

Yes

An authenticated management user must enable tracing or access the trace outputs via the management interface on 127.0.0.1:15671, which requires a separate individual from the attacker to perform an action for the attack to succeed.
INTELLIGENCE Finding description *"The exploitation path is local or authenticated-management access leading to the enabled tracing firehose and then to captured inter-service message bodies. An operator turns on a trace, or an actor already holding management credentials does so"*

Active

The authenticated management user must consciously enable tracing or access trace outputs via the management interface, which is beyond normal operation and requires deliberate action.
INTELLIGENCE Finding description *"An operator turns on a trace, or an actor already holding management credentials does so"*

Vulnerable System Confidentiality Low

from the modelled C/I/A of the vulnerable system's rated postconditions
Low on the vulnerable system, from the modelled C/I/A of the finding's postconditions: Attacker reads captured inter-service broker messages: full message bodies and routing metadata for Local Run Manager run, analysis, and job traffic is rated Low for confidentiality.

REASONING & EVIDENCE

Modelled C/I/A · vulnerable system Low
From the modelled C/I/A of the finding's postconditions on the vulnerable system.
CONDITION Attacker reads captured inter-service broker messages: full message bodies and routing metadata for Local Run Manager run, analysis, and job traffic `confidentiality = L`

Vulnerable System Integrity None

from the modelled C/I/A of the vulnerable system's rated postconditions
None on the vulnerable system: none of the finding's postconditions carries a integrity rating.

REASONING & EVIDENCE

Modelled C/I/A · vulnerable system None
From the modelled C/I/A of the finding's postconditions on the vulnerable system.

Vulnerable System Availability None

from the modelled C/I/A of the vulnerable system's rated postconditions
None on the vulnerable system: none of the finding's postconditions carries a availability rating.

REASONING & EVIDENCE

Modelled C/I/A · vulnerable system None
From the modelled C/I/A of the finding's postconditions on the vulnerable system.

Subsequent System Confidentiality None

the subsequent system is not counted: 3.1 S is U
Scope is Unchanged: exploiting this flaw affects nothing beyond the vulnerable system, so there is no subsequent-system impact.

Subsequent System Integrity None

the subsequent system is not counted: 3.1 S is U
Scope is Unchanged: exploiting this flaw affects nothing beyond the vulnerable system, so there is no subsequent-system impact.

Subsequent System Availability None

the subsequent system is not counted: 3.1 S is U

Scope is Unchanged: exploiting this flaw affects nothing beyond the vulnerable system, so there is no subsequent-system impact.

Abbreviations

Every abbreviation this report uses, with the one meaning ELTON gives it. Wherever the report writes one, it links to its row here.

ABBREVIATION	MEANING
AC	Attack Complexity
ACE	Access Control Entry
ACL	Access Control List
ADB	Android Debug Bridge
AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
ASLR	Address Space Layout Randomization
AT	Attack Requirements
AV	Attack Vector
CDN	Content Delivery Network
CEH	Certified Ethical Hacker
CEO	Chief Executive Officer
CFG	Control Flow Guard
CI	Continuous Integration
CIA	Confidentiality, Integrity, and Availability
CompTIA	Computing Technology Industry Association
CPT	Certified Penetration Tester
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
CWE	Common Weakness Enumeration
DACL	Discretionary Access Control List
DAST	Dynamic Application Security Testing
DEP	Data Execution Prevention
DLL	Dynamic-Link Library
EC-Council	International Council of E-Commerce Consultants
EPSS	Exploit Prediction Scoring System
FDA	Food and Drug Administration
GIAC	Global Information Assurance Certification
GPEN	GIAC Penetration Tester
GWAPT	GIAC Web Application Penetration Tester
H-ISAC	Health Information Sharing and Analysis Center
HKCU	HKEY_CURRENT_USER
HKLM	HKEY_LOCAL_MACHINE
HSCC	Health Sector Coordinating Council
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IA	Initial Access
ID	Identifier
IEC	International Electrotechnical Commission
IIS	Internet Information Services
ISSAF	Information Systems Security Assessment Framework
JSON	JavaScript Object Notation
JSP	Joint Security Plan
JSP2	Joint Security Plan Version 2
LAN	Local Area Network
MDDT	Medical Device Development Tool
MFA	Multi-Factor Authentication
NT	New Technology
NVD	National Vulnerability Database
OffSec	Offensive Security
OIDC	OpenID Connect
OS	Operating System
OSCP	Offensive Security Certified Professional
PDF	Portable Document Format
PHI	Protected Health Information
PII	Personally Identifiable Information
PR	Privileges Required
PTES	Penetration Testing Execution Standard
SA	Subsequent System Availability Impact
SAST	Static Application Security Testing
SBC	Single-Board Computer
SBOM	Software Bill of Materials
SC	Subsequent System Confidentiality Impact
SCA	Software Composition Analysis
SD	Secure Digital
SHA-256	Secure Hash Algorithm 256-bit
SI	Subsequent System Integrity Impact
SMB	Server Message Block
SQL	Structured Query Language
SSVC	Stakeholder-Specific Vulnerability Categorization

ABBREVIATION	MEANING
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TLS	Transport Layer Security
UI	User Interaction
UL	Underwriters Laboratories
URL	Uniform Resource Locator
USB	Universal Serial Bus
UTC	Coordinated Universal Time
VA	Vulnerable System Availability Impact
VC	Vulnerable System Confidentiality Impact
VI	Vulnerable System Integrity Impact
VM	Virtual Machine
XML	Extensible Markup Language
XSS	Cross-Site Scripting