• CONFIDENTIAL

Electron Desktop Application

Security Assessment Report · Windows Build

PREPARED FOR

Client name withheld

DATE

Sample — date withheld

REFERENCE

SAMPLE-REDACTED

CONTENTS

Report Contents

01 Engagement Details

REPORT TITLE	Electron Desktop Application — Security Assessment Report
PREPARED FOR	Client name withheld
LEAD TESTER	Selva Ops S.R.L
TESTING PERIOD	Withheld (sample report)
PLATFORM	Windows (x64)
REFERENCE ID	SAMPLE-REDACTED

ASSESSMENT SCOPE

- Windows MSI installer (x64) and extracted asar bundle
- Electron main process, preload scripts and IPC handler surface
- Bundled MCP servers and local HTTP listeners
- Auto-update mechanism and update transport chain

METHODOLOGY

Static analysis: MSI extraction, asar extraction, manual source review, Electron fuse inspection, and PE binary version-string extraction. All findings verified against the extracted Windows asar bundle.

02 Risk Overview

16	2	4	8	2
TOTAL	CRITICAL	HIGH	MEDIUM	LOW

03 Findings Summary

ID	FINDING	SEVERITY	CVSS
FIND-001	SSRF via Unvalidated URL in MCP Tool Execution Handler	HIGH	8.3
FIND-002	Token Exfiltration to Backend Credential Theft (Attack Chain)	CRITICAL	9.1
FIND-003	End-to-End Test Mode Shipped in Production Build with Hardcoded JWT Secret	HIGH	7.8
FIND-004	Update Supply Chain: No Code Signing Verification, Public Bucket Trust	HIGH	7.7
FIND-005	Arbitrary Process Execution via Desktop Control MCP Server	CRITICAL	Chain
FIND-006	Renderer Sandbox Disabled	MEDIUM	6.5
FIND-007	Unrestricted Secrets Object Access from Renderer via IPC	MEDIUM	5.5
FIND-008	Environment Variable Overrides All Secret Fetching	MEDIUM	6.3
FIND-009	Force Update Installs Silently Without User Confirmation	MEDIUM	5.9
FIND-010	OAuth Callback Server Bound to Hardcoded Port (Port Squatting)	MEDIUM	5.3
FIND-011	Electron Fuse: File Protocol Granted Extra Privileges	MEDIUM	4.8
FIND-012	Electron Fuse: Browser-Process-Specific V8 Snapshot Disabled	MEDIUM	4.2
FIND-013	Workspace Folder Grants Broad Silent Renderer File Read Access	MEDIUM	4.4
FIND-014	MCP Server Health Endpoint Unauthenticated	LOW	3.3
FIND-015	Non-Constant-Time Token Comparison in MCP Authentication	LOW	2.9
FIND-016	Outdated Runtime: Electron, Node.js and Chromium with Known CVEs	HIGH	7.5

04 Executive Narrative

Static analysis of the Windows MSI installer identified 16 security findings: 2 Critical, 4 High, 8 Medium and 2 Low, alongside one informational note and two verified positive controls. The assessment covered the Electron main process, the preload and IPC surface, the bundled MCP servers, and the auto-update chain.

The most significant issue is a server-side request forgery flaw in the MCP tool execution handler (FIND-001). It allows any renderer-side JavaScript to issue authenticated HTTP requests to arbitrary URLs, including internal localhost MCP ports. Chained with the Desktop Control MCP server, this escalates to arbitrary process execution on the host (FIND-005), and chained with the credential API it yields full theft of stored plaintext credentials (FIND-002).

Two further High-severity issues compound this: a complete end-to-end test harness shipped in the production build with a hardcoded, well-known JWT secret (FIND-003), and an auto-update chain that performs no code-signing verification while executing scripts from a user-writable temporary directory with elevation (FIND-004).

On the positive side, several controls were verified as correctly implemented. Context isolation is enabled and Node integration disabled in the renderer, the majority of Electron fuses are securely configured, the local database encryption key is correctly protected via the platform keystore, and the file system service properly defends against path traversal.

SELVA OPS

FIND-001	HIGH

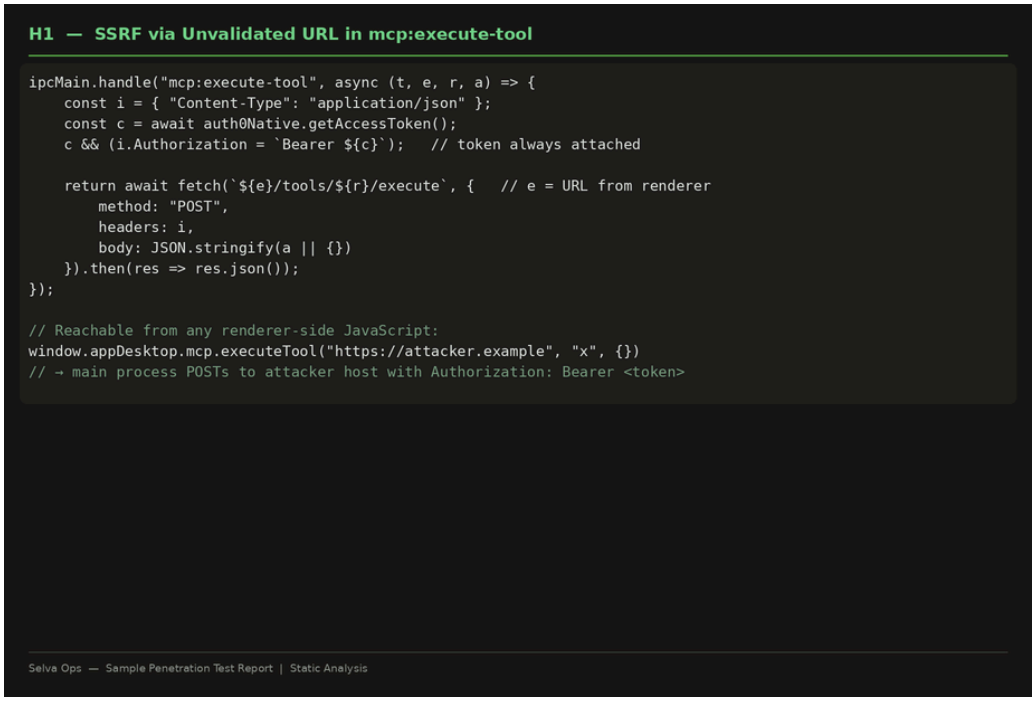
SSRF via Unvalidated URL in MCP Tool Execution Handler

AFFECTED ASSETS	Main process IPC handler (mcp:execute-tool)
CVSS V3.1	8.3
REMEDIATION SLA	Immediate

DESCRIPTION

The IPC handler registered for MCP tool execution accepts a server URL directly from the renderer and constructs an outbound HTTP request with no validation or allowlisting. The user's bearer token is always attached to the outbound request, so any renderer-side JavaScript can cause the main process to send an authenticated request to an attacker-controlled host.

EVIDENCE



REMEDIATION

1. Maintain a server-side or build-time allowlist of permitted MCP server URLs.
2. Pass a registered server ID from the renderer and resolve the URL internally in the main process.
3. Strip the Authorization header whenever the target URL is not the known backend domain.

REFERENCES

› OWASP — Server Side Request Forgery
› Electron Security Checklist — handling untrusted content

FIND-002	CRITICAL

Token Exfiltration to Backend Credential Theft (Attack Chain)

AFFECTED ASSETS	IPC handlers; credential manager service
CVSS V3.1	9.1
REMEDIATION SLA	Immediate

DESCRIPTION

This finding documents the full chain enabled by FIND-001. Credentials are not held locally — they are stored server-side and retrieved on demand as plaintext via the backend API. Once the bearer token is exfiltrated, an attacker can enumerate all stored credential aliases and then retrieve the plaintext username and password for any connected service, with no local access to the device required.

REMEDIATION

1. Fix FIND-001 — this breaks the chain at its source.
2. Issue short-lived, narrowly scoped tokens for credential retrieval rather than reusing the primary access token.
3. Require re-authentication before the backend returns any plaintext secret.
4. Apply backend-side rate limiting to the credential retrieval endpoint.

REFERENCES

› OWASP — Server Side Request Forgery
› OWASP — Credential Storage Cheat Sheet

SELVA OPS

FIND-003	HIGH

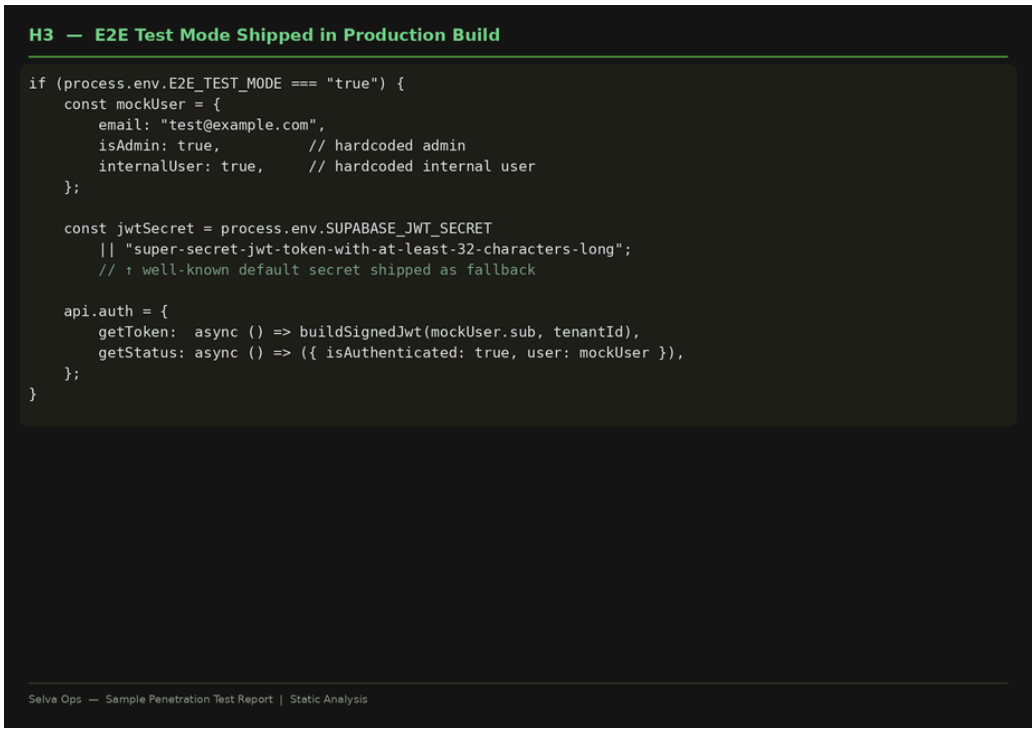
End-to-End Test Mode Shipped in Production Build with Hardcoded JWT Secret

AFFECTED ASSETS	Preload script (production build)
CVSS V3.1	7.8
REMEDIATION SLA	Immediate

DESCRIPTION

The production preload script contains a complete end-to-end testing harness that activates via an environment variable. When active it bypasses authentication entirely, substituting a hardcoded administrative user, and signs JWTs locally using a hardcoded fallback secret that is a well-known published default. On Windows, environment variables can be set by any user-level process without administrative rights.

EVIDENCE



REMEDIATION

1. Strip all end-to-end test stubs from production builds using build flags.
2. Rotate the signing secret immediately if it was ever left at the default value.
3. Never ship fallback credentials or secrets in client-side code.

REFERENCES

› OWASP — Test Data Management Cheat Sheet
› OWASP Top 10 — A05 Security Misconfiguration

FIND-004	HIGH

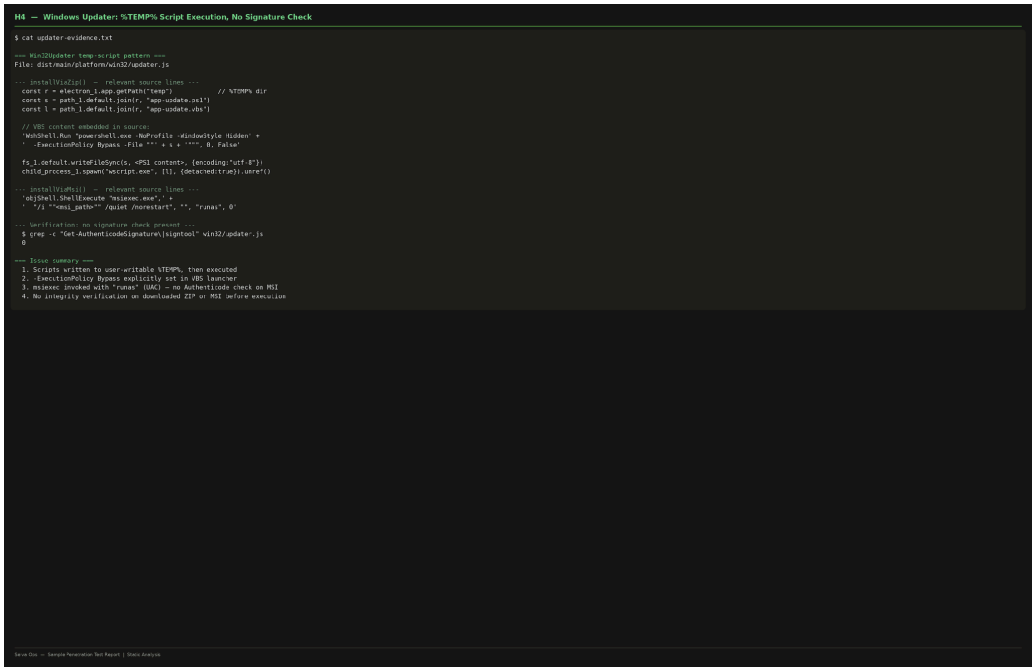
Update Supply Chain: No Code Signing Verification, Public Bucket Trust

AFFECTED ASSETS	Auto-updater service; Windows platform updater
CVSS V3.1	7.7
REMEDIATION SLA	Short-term (7 days)

DESCRIPTION

The auto-update mechanism fetches both the update manifest and the binary from the same public, unauthenticated object storage bucket. Integrity is verified only by a hash published in that same manifest, making the check meaningless to an attacker who controls the bucket. On Windows the update writes a PowerShell script and a VBS launcher into the user-writable temporary directory and executes them with execution policy bypassed; MSI updates are invoked with elevation and no signature verification.

EVIDENCE



REMEDIATION

1. Sign update manifests with a key held outside the storage bucket and verify that signature client-side before trusting any hash.
2. Restrict bucket write access to a dedicated CI/CD service account.
3. Verify the Authenticode signature of the downloaded installer before execution.
4. Write update scripts to a protected application directory rather than %TEMP%.
5. Consider MSIX packaging, which enforces signature validation on updates natively.

REFERENCES

- Microsoft — Get-AuthenticodeSignature
- OWASP — Software Supply Chain Security Cheat Sheet

FIND-005	CRITICAL

Arbitrary Process Execution via Desktop Control MCP Server

AFFECTED ASSETS	Desktop Control MCP server; platform automation module
CVSS V3.1	Chain
REMEDIATION SLA	Immediate

DESCRIPTION

The Desktop Control MCP server exposes an application-launch tool that passes a caller-supplied name directly to the Windows command interpreter with no allowlist or sanitisation. An attacker who has already exploited FIND-001 (to reach the localhost MCP port) and FIND-007 (to obtain the tenant key) can supply any executable path, achieving arbitrary process execution on the host as the logged-in user.

REMEDIATION

1. Restrict the launch tool to a curated allowlist of permitted applications.
2. Fix FIND-001 to break the SSRF chain that makes this path reachable.

REFERENCES

› Electron Security — validate the sender of all IPC messages

FIND-006	MEDIUM

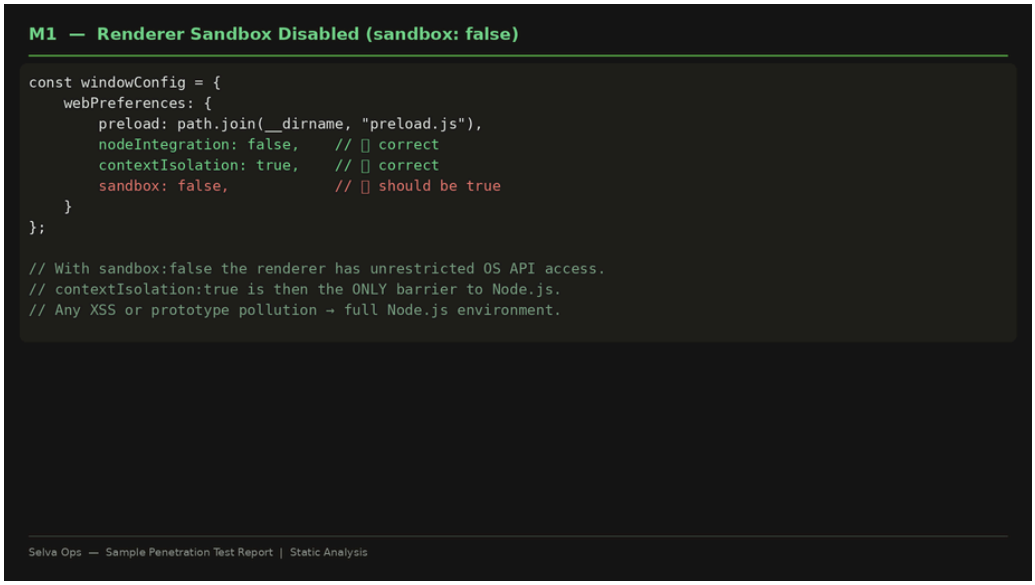
Renderer Sandbox Disabled

AFFECTED ASSETS	Main application window configuration
CVSS V3.1	6.5
REMEDIATION SLA	Short-term (30 days)

DESCRIPTION

The main browser window is created with the Chromium OS-level process sandbox disabled. Context isolation is correctly enabled, but it then becomes the only barrier between renderer content and a full Node.js environment. On Windows this leaves the renderer with unrestricted access to Win32 APIs, amplifying the impact of FIND-001 through FIND-003.

EVIDENCE



REMEDIATION

1. Enable the sandbox — it has been the Electron default since v20.
2. Migrate preload Node.js usage to main-process IPC handlers exposed via the context bridge.

REFERENCES

› Electron — Process Sandboxing tutorial

FIND-007	MEDIUM

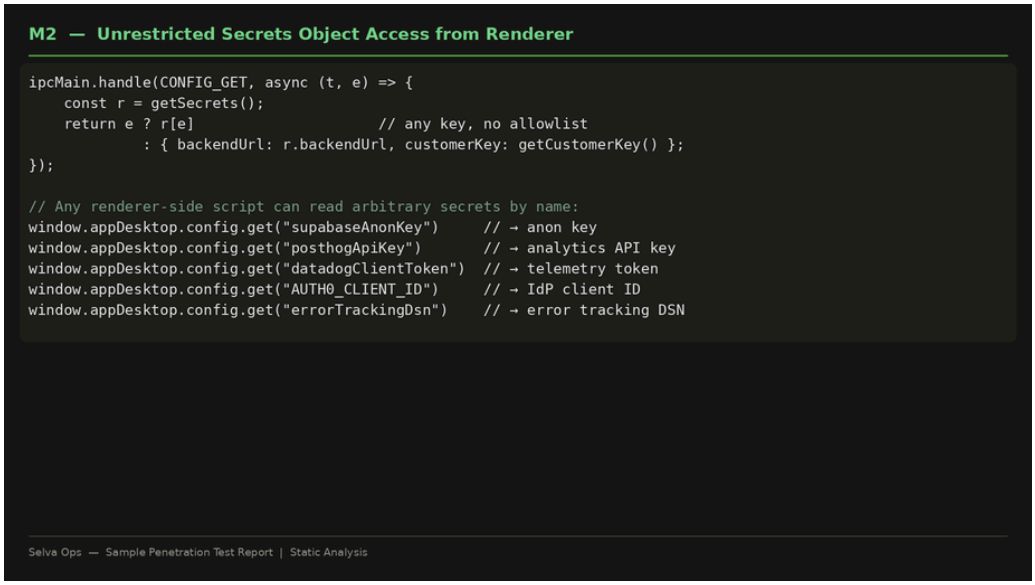
Unrestricted Secrets Object Access from Renderer via IPC

AFFECTED ASSETS	Configuration IPC handler; secrets service
CVSS V3.1	5.5
REMEDIATION SLA	Short-term (30 days)

DESCRIPTION

The configuration IPC handler returns any key from the in-memory secrets object using a key name supplied by the renderer, with no allowlist. A renderer can therefore extract third-party API keys, analytics tokens, telemetry credentials and identity-provider client identifiers by name. Omitting the key name returns a broader configuration object.

EVIDENCE



REMEDIATION

1. Define an explicit allowlist of keys the renderer may request.
2. Expose specific values via dedicated IPC channels rather than a generic getter.
3. Remove the fallback that returns the full secrets object when no key is given.

REFERENCES

› OWASP — Secrets Management Cheat Sheet

FIND-008	MEDIUM

Environment Variable Overrides All Secret Fetching

AFFECTED ASSETS	Secrets service
CVSS V3.1	6.3
REMEDIATION SLA	Short-term (30 days)

DESCRIPTION

The backend URL resolver unconditionally trusts an environment variable over the compiled-in tenant key. On Windows this variable can be set persistently via the registry or injected by a malicious process without elevated privileges. All secret fetching flows through this URL, so every secret served to the application can be substituted with attacker-controlled values.

REMEDIATION

1. Remove the override in production builds; gate it behind a development-only build flag.
2. Validate the fetched secrets response against an expected schema before caching.
3. Pin the TLS certificate of the backend domain.

REFERENCES

› OWASP — Secrets Management Cheat Sheet

FIND-009	MEDIUM

Force Update Installs Silently Without User Confirmation

AFFECTED ASSETS	Auto-updater service
CVSS V3.1	5.9
REMEDIATION SLA	Next release

DESCRIPTION

When the remote update configuration sets a force-update threshold, the application auto-installs the downloaded update a few seconds after download with no user prompt. Combined with FIND-004, an attacker who controls the update bucket can force-install a malicious update on all clients automatically, with attendant risk of data loss from the forced quit.

REMEDIATION

1. Always show a confirmation dialog before installing any update, including forced ones.
2. Sign the update configuration independently of the binary manifest.
3. Gate force-update logic behind a server-side authenticated feature flag.

REFERENCES

› Electron — autoUpdater API

SELVA OPS

FIND-010	MEDIUM

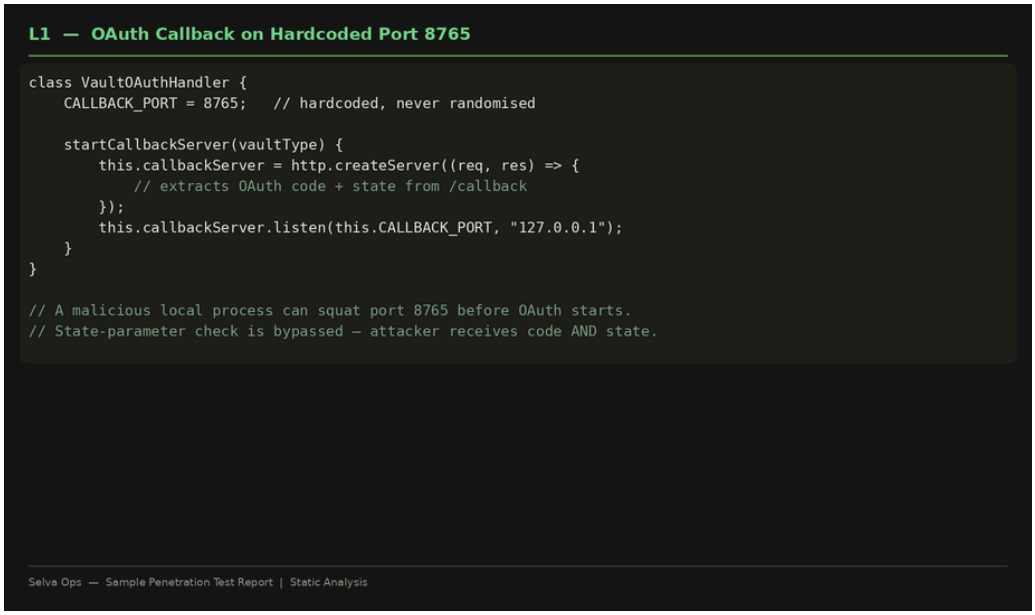
OAuth Callback Server Bound to Hardcoded Port (Port Squatting)

AFFECTED ASSETS	Vault OAuth handler
CVSS V3.1	5.3
REMEDIATION SLA	Next release

DESCRIPTION

The vault OAuth flow starts a local HTTP server on a fixed, never-randomised port to receive the authorization code callback. On Windows any user-level process can bind that port first. A malicious process that wins the race receives both the authorization code and the state parameter, defeating the CSRF state check and allowing exchange of the code for access and refresh tokens.

EVIDENCE



REMEDIATION

1. Use a dynamically assigned port and register it with the provider at initiation time.
2. Alternatively register a custom URI scheme as the redirect target, removing the need for a local HTTP server.

REFERENCES

› RFC 8252 §7.3 — OAuth 2.0 for Native Apps

FIND-011	MEDIUM

Electron Fuse: File Protocol Granted Extra Privileges

AFFECTED ASSETS	Application binary (Electron fuse configuration)
CVSS V3.1	4.8
REMEDIATION SLA	Next release

DESCRIPTION

The fuse granting extra privileges to the file protocol is enabled, giving local file pages elevated browser capabilities including fetch access, service workers and cross-frame access. This widens the attack surface if any part of the application navigates to a local file URL.

REMEDIATION

1. Disable the fuse at build time.
2. Serve local assets via a registered custom protocol instead of the file protocol.

REFERENCES

› Electron — Fuses tutorial

SELVA OPS

FIND-012	MEDIUM

Electron Fuse: Browser-Process-Specific V8 Snapshot Disabled

AFFECTED ASSETS	Application binary (Electron fuse configuration)
CVSS V3.1	4.2
REMEDIATION SLA	Next release

DESCRIPTION

The fuse that loads a browser-process-specific V8 snapshot is disabled, so the same V8 context snapshot is shared between the main and renderer processes. State associated with privileged contexts can therefore bleed across the process boundary, reducing defence in depth.

REMEDIATION

1. Enable the fuse at build time.

REFERENCES

› Electron — Fuses tutorial

FIND-013	MEDIUM

Workspace Folder Grants Broad Silent Renderer File Read Access

AFFECTED ASSETS	File system service; file read IPC handler
CVSS V3.1	4.4
REMEDIATION SLA	Backlog

DESCRIPTION

When a user selects a workspace folder, the entire folder tree is added to the file system allowlist and the renderer may then read any file within it without further prompting. Selecting a home directory silently grants read access to SSH keys, environment files, browser profiles and application credentials. Combined with FIND-001, those files could be exfiltrated.

REMEDIATION

1. Display a clear warning explaining the scope of access when a workspace is selected.
2. Restrict file reads to extensions relevant to the application's purpose.
3. Limit workspace depth rather than recursively allowing all subdirectories.

REFERENCES

› Electron Security — only load secure content

SELVA OPS

FIND-014	LOW

MCP Server Health Endpoint Unauthenticated

AFFECTED ASSETS	MCP authentication middleware
CVSS V3.1	3.3
REMEDIATION SLA	Backlog

DESCRIPTION

The MCP authentication middleware explicitly bypasses authentication for the health route. Any local process can query it and receive server version and status information without credentials, confirming the server's presence and aiding local reconnaissance.

REMEDIATION

1. Remove the authentication bypass, or return minimal information with an empty body.

FIND-015	LOW

Non-Constant-Time Token Comparison in MCP Authentication

AFFECTED ASSETS	MCP authentication middleware
CVSS V3.1	2.9
REMEDIATION SLA	Backlog

DESCRIPTION

The MCP authentication middleware compares the bearer token using a standard equality operator, which short-circuits on the first non-matching character. Practical exploitation is extremely difficult given token length and localhost latency variance, but it is a deviation from cryptographic best practice.

REMEDIATION

1. Use a constant-time comparison function for all token comparisons.

REFERENCES

› Node.js — crypto.timingSafeEqual

FIND-016	HIGH

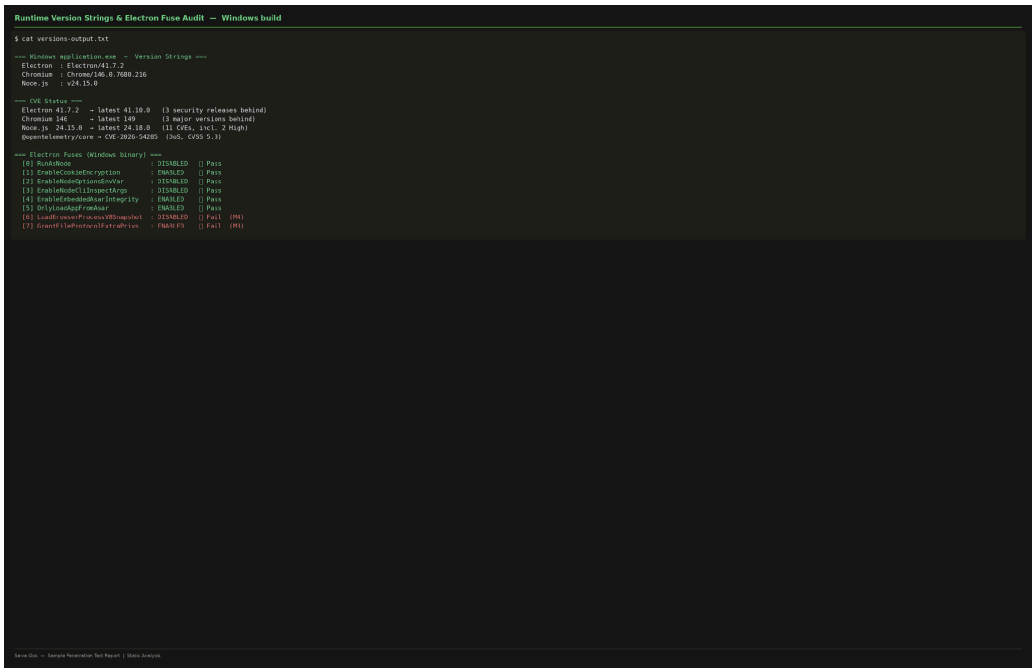
Outdated Runtime: Electron, Node.js and Chromium with Known CVEs

AFFECTED ASSETS	Application runtime; bundled npm dependencies
CVSS V3.1	7.5
REMEDIATION SLA	Short-term (30 days)

DESCRIPTION

The bundled runtime is three minor releases behind the current stable line. The corresponding Node.js upgrade is a dedicated security release resolving 11 CVEs, two rated High — a TLS hostname normalisation bypass and an unguarded WebCrypto cipher output with memory-corruption potential. The embedded Chromium is three major versions behind stable, and one bundled npm dependency carries a Medium-severity denial-of-service advisory.

EVIDENCE



REMEDIATION

1. Upgrade the Electron runtime to the current stable patch release, which also upgrades the bundled Node.js and resolves all 11 CVEs.
2. Upgrade the affected npm dependency to a patched version.
3. Implement automated dependency monitoring with auto-merge for patch-level security updates.

REFERENCES

› Node.js security releases
› Electron — keep Electron updated

21 Informational Notes & Positive Controls

NOTE-01 — INFORMATIONAL

Tenant Key Hardcoded in Binary

The tenant identifier is compiled into the binary, making the backend subdomain trivially discoverable by anyone who unpacks the application bundle. There is no direct exploitability, but it confirms the exact credential API base URL when combined with FIND-001 and FIND-002. Treat the tenant key as non-secret and document it in the threat model as a known reconnaissance artefact.

NOTE-02 — POSITIVE CONTROL

Database Encryption Correctly Uses the Platform Keystore

The local database encryption key is correctly protected. The key is wrapped using the platform safe-storage API, which delegates to the operating system data protection service. Where that service is unavailable the application throws rather than falling back to storing the key in plaintext. The key cannot be recovered without access to the user's OS-level master key. No action required.

NOTE-03 — POSITIVE CONTROL

File System Path Traversal — Not Vulnerable

The file system service correctly defends against path traversal. All paths are fully resolved and validated against an explicit allowlist using containment checks before any read or write operation. No action required; maintain the existing controls.

SELVA OPS