# Lockfile checksums not verified in Hex allows dependency integrity bypass

High **ericmj** published **GHSA-hmv9-4mfr-m92v** 1 hour ago

**Package**

**hex** (otp)

**Affected versions**

>= 0.16.0 and < 2.4.2

**Patched versions**

2.4.2

## Description

### Summary

Hex does not properly enforce dependency checksums recorded in mix.lock during dependency resolution. While checksums are validated when downloading packages, mismatches between the lockfile and resolved dependencies are not detected, weakening the integrity guarantees of the lockfile.

### Details

Hex stores checksums for dependencies in the mix.lock file to ensure reproducible and integrity-checked builds.

However, due to a mismatch in `Hex.RemoteConverger.verify_resolved/2`, checksum verification is never executed. Specifically, the lock data returned by `Hex.Utils.lock/1` uses string-based dependency names, while the verification logic compares against atom-based names. This mismatch prevents the pattern match from succeeding, and the verification code path is skipped entirely.

This issue appears to have been introduced when the return format of `Hex.Utils.lock/1` was changed from a list to a map, without updating the corresponding verification logic.

As a result, although checksums are written to `mix.lock`, they are not enforced when validating the resolved dependency set.

Checksum validation still occurs when packages are initially downloaded from the registry. This prevents straightforward tampering during normal package retrieval.

## PoC

1. Modify the checksum values of a dependency in `mix.lock`
2. Run: `mix deps.get`
3. Observe:
    - No warning or error is raised
    - The dependency is accepted
    - `mix.lock` is silently rewritten with the correct checksum values from the registry

This demonstrates that checksum verification is not performed, and integrity mismatches are not detected.

## Impact

This vulnerability weakens the integrity guarantees of the lockfile ( `mix.lock` ), which is expected to ensure that the exact, audited dependency set is used.

Because checksums are still verified during package download, arbitrary tampering during normal registry retrieval is not possible. However, the lack of enforcement of lockfile checksums means that **mismatches between expected and actual dependency contents are not detected** once dependencies are resolved.

This can be relevant in scenarios such as:

- **Local cache poisoning:**
  If an attacker is able to modify cached Hex packages on disk (e.g. through prior system compromise, shared environments, or other local access), the modified dependency may be used without detection. The lockfile does not provide protection against such tampering.
- **Re-publication within the initial availability window:**
  If a package is re-published or its contents change shortly after release (e.g. within a short time window before caches stabilize), different clients may observe different package contents. Because lockfile checksums are not enforced, these inconsistencies are not detected, weakening reproducibility and integrity guarantees.

Installing and executing untrusted dependencies remains inherently risky. This vulnerability specifically reduces the effectiveness of the lockfile as a mechanism for detecting unintended or malicious changes to dependency contents.

## References

- Introduction of issue: `e01576f`
- Fix: `TODO`

### Severity

High 8.9 / 10

**CVSS v4 base metrics**

**Exploitability Metrics**

| | |
|---|---|
| Attack Vector | Network |
| Attack Complexity | Low |
| Attack Requirements | Present |
| Privileges Required | None |
| User interaction | Active |

**Vulnerable System Impact Metrics**

| | |
|---|---|
| Confidentiality | High |
| Integrity | High |
| Availability | High |

**Subsequent System Impact Metrics**

| | |
|---|---|
| Confidentiality | High |
| Integrity | High |
| Availability | High |

Learn more about base metrics

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

### CVE ID

CVE-2026-32148

### Weaknesses

- CWE-354
- CWE-494

### Credits

| | |
|---|---|
| xpgdk | Reporter |
| maennchen | Remediation developer |
| ericmj | Remediation reviewer |