hexpm / **hex** Public

Code Issues 11 Pull requests 5 Actions Projects Wiki Secur

Commit d7528c8

ericmj and **maennchen** committed 2 hours ago · ✓ 10 / 10 · Unverified

```
Fix checksum verification for dependencies in mix.lock

Checksum verification was not being performed due to a type mismatch
in pattern matching. Comparing atom-based names against string-based
lock data caused the verification to be silently skipped.

Fixes: GHSA-hmv9-4mfr-m92v
Fixes: CVE-2026-32148

Co-authored-by: Jonatan Männchen <jonatan@maennchen.ch>
Co-authored-by: Eric Meadows-Jönsson <eric.meadows.jonsson@gmail.com>
```

main · v2.4.2

1 parent b5a1e94 commit d7528c8

2 files changed +54 -4

Top

Filter files…

- lib/hex
  - remote_converger.ex
- test/hex
  - remote_converger_test.exs

Search within code

lib/hex/remote_converger.ex

```
@@ -569,10 +569,8 @@ defmodule Hex.RemoteConverger do
569  569
570  570      defp verify_resolved(resolved, lock) do
```

```
571  571        Enum.each(resolved, fn {repo, name, app, version} ->
572       -        atom_name = String.to_atom(name)
573       -
574  572          case Hex.Utils.lock(lock[String.to_atom(app)]) do
575       -          %{name: ^atom_name, version: ^version, repo: ^repo} = lock ->
     573  +          %{name: ^name, version: ^version, repo: ^repo} = lock ->
576  574            verify_inner_checksum(repo, name, version, lock.inner_checksum)
577  575            verify_outer_checksum(repo, name, version, lock.outer_checksum)
578  576            verify_deps(repo, name, version, lock.deps)
     ↕    @@ -599,14 +597,16 @@ defmodule Hex.RemoteConverger do
599  597        end
600  598      end
601  599
     600  +   defp verify_deps(repo, name, version, deps)
602  601      defp verify_deps(nil, _name, _version, _deps), do: :ok
     602  +   defp verify_deps(_repo, _name, _version, nil), do: []
603  603
604  604      defp verify_deps(repo, name, version, deps) do
605  605        # TODO: Use requests?
606  606        deps =
607  607          Enum.map(deps, fn {app, req, opts} ->
608  608            %{
609       -            repo: opts[:repo],
     609  +            repo: if(opts[:repo] != "hexpm", do: opts[:repo]),
610  610            name: opts[:hex],
611  611            constraint: Hex.Solver.parse_constraint!(req),
612  612            optional: !!opts[:optional],
     ↕
```

test/hex/remote_converger_test.exs

```
     ↕    @@ -198,4 +198,54 @@ defmodule Hex.RemoteConvergerTest do
198  198          refute_received {:mix_shell, :yes?, _}
199  199        end)
200  200      end
     201  +
     202  +   defmodule ChecksumIntegrity.MixProject do
     203  +     def project do
     204  +       [
     205  +         app: :checksum_integrity,
     206  +         version: "0.1.0",
```

```
207 +         deps: [
208 +           {:ex_doc, "~> 0.1.0"}
209 +         ]
210 +       ]
211 +     end
212 +   end
213 +
214 +   test "raises on checksum mismatch in mix.lock" do
215 +     in_tmp(fn ->
216 +       Mix.Project.push(ChecksumIntegrity.MixProject)
217 +
218 +       # First, get dependencies normally to create a valid lock file
219 +       :ok = Mix.Tasks.Deps.Get.run([])
220 +
221 +       # Read the lock file
222 +       lock = Mix.Dep.Lock.read()
223 +       {:hex, name, version, inner_checksum, managers, deps, repo,
    outer_checksum} = lock[:ex_doc]
224 +
225 +       assert_checksum_mismatch(%{
226 +         ex_doc:
227 +           {:hex, name, version, invalid_checksum(inner_checksum), managers,
    deps, repo,
228 +            outer_checksum}
229 +       })
230 +
231 +       assert_checksum_mismatch(%{
232 +         ex_doc:
233 +           {:hex, name, version, inner_checksum, managers, deps, repo,
234 +            invalid_checksum(outer_checksum)}
235 +       })
236 +     end)
237 +   end
238 +
239 +   defp assert_checksum_mismatch(lock) do
240 +     File.write!("mix.lock", inspect(lock, limit: :infinity, pretty: true))
241 +     Mix.Task.clear()
242 +
243 +     # The bug causes this to silently pass and rewrite the lock file with
    correct checksums
```

```
244 +      assert_raise Mix.Error, ~r/Registry checksum mismatch against lock/, fn ->
245 +        Mix.Tasks.Deps.Get.run([])
246 +      end
247 +    end
248 +
249 +    defp invalid_checksum("0" <> rest), do: "1" <> rest
250 +    defp invalid_checksum(<<_::binary-size(1), rest::binary>>), do: "0" <> rest
201 251    end
```

Comments 0

Please sign in to comment.