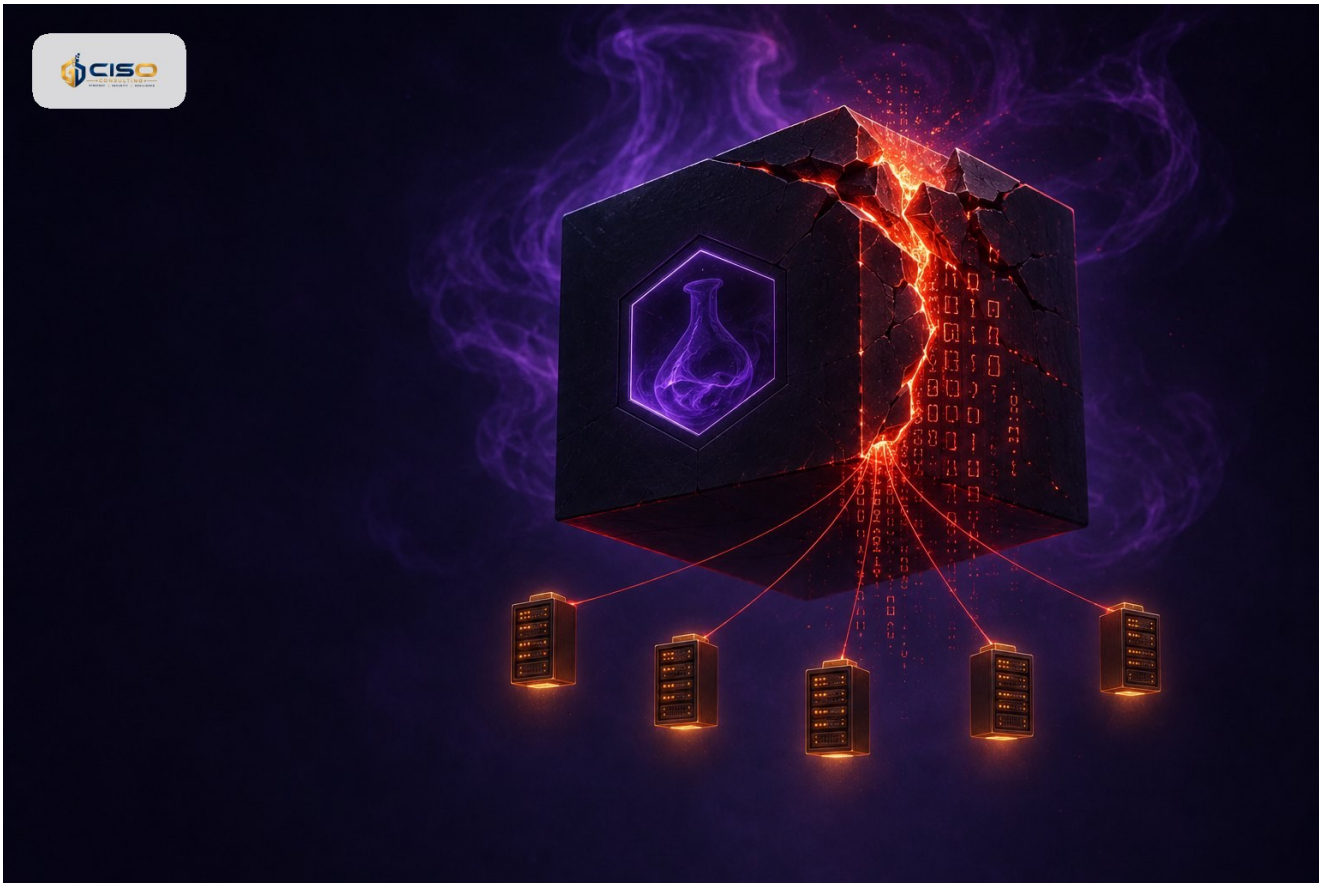
Hex.pm package registry hit by CVSS 10.0 remote code execution flaw; six servers compromised



An attacker ran operating system commands on six production servers of Hex.pm, the registry used by Elixir and Erlang developers. The flaw dated to February 2016, and maintainers patched every server within about 15 minutes.

- Six production servers of Hex.pm were compromised through a remote code execution flaw rated CVSS 10.0.
- The attacker ran operating system commands on the servers on a Monday.
- The vulnerable code had been in the registry since February 2016.
- The maintainers had all servers on a patched build within about 15 minutes of the first malicious request.

What happened

An attacker executed operating system commands on six production servers of Hex.pm, the package registry from which Elixir and Erlang developers fetch their dependencies. The vulnerability is a remote code execution flaw rated CVSS 10.0, the highest score on the scale.

The weakness had been present in the registry's code since February 2016. The team that maintains the service moved quickly: every server was running a patched build within about 15 minutes of the first malicious request.

Why it matters

Package registries sit at the start of the software supply chain. Organisations that build applications in Elixir or Erlang pull code from Hex.pm into their development and build environments, so a compromise of the registry is relevant to them even though the source describes the attack as targeting the registry's own servers.

The source excerpt does not say whether any published packages were altered, nor does it give a CVE identifier or further technical detail. Readers should therefore avoid drawing conclusions beyond what has been confirmed.

Recommended approach

Teams using Elixir or Erlang should treat this as a prompt to review how dependencies are obtained and verified, and to follow the maintainers' official communications for any guidance on affected packages or credentials. A calm, evidence-based check of recent dependency activity is more useful than broad alarm.

What to do now

1. Identify which internal projects and build pipelines depend on Hex.pm packages.
2. Monitor the Hex.pm maintainers' official announcements for guidance on affected packages or credentials.
3. Review recent dependency updates in Elixir and Erlang projects for unexpected changes.
4. Pin dependency versions and verify checksums or lock files in build pipelines.
5. Include open-source registries in third-party and supply-chain risk assessments.