activestate

Software Attestations

Software attestations are a key way for vendors to establish trust for their software by offering customers a way to independently validate the security and integrity of their applications.

An attestation is instrumental in helping to secure the software supply chain, which has been increasingly under attack by hackers, cyber criminals and other bad actors. The US government is leading the charge by requiring software vendors to comply with a number of secure supply chain requirements if they want to sell to US government departments and agencies, including providing a software attestation that contains (at a minimum):

- The software vendor's name
- A description of the product or products the statement refers to
- A statement attesting that the software vendor follows secure development practices

The ActiveState Platform's secure build service will generate a signed attestation for an application's open source components, and then verify their security and integrity on installation using the attestation's metadata (per artifact checksums, signatures, etc).

OPEN SOURCE PROVENANCE ATTESTATIONS

Simply put, provenance is metadata (information) about how a software artifact was produced, including the origin or source for all components used in/by the build process, such as:

The origin of the proprietary source code (such as a git repository)

The origin of all dependencies/transitive dependencies (both prebuilt dependencies and source code)

- In other words, software vendors will need to provide an attestation not only for how the overall application was built, but how the prebuilt open source code included in the software was created, as well.

Public repositories do not provide attestations for the prebuilt open source dependencies they offer. As a result, software vendors will need to either:

- Build all dependencies from source code and provide attestations for each using a dependency vendoring strategy, which can be complex to set up and maintain if native libraries are required.
- Use a build system like Github Actions or Azure DevOps, which can provide attestations for proprietary software but cannot provide attestations for the open source binaries included in your application.
- **Use the ActiveState Platform, which automatically builds all open source binaries from source code, and provides an attestation for each.**

SECURING THE SOFTWARE SUPPLY CHAIN

The US (and other governments, including the UK) have introduced legislation requiring that their software vendors incorporate secure software supply chain practices into their software development processes. But even if a software vendor doesn't sell directly to the government, their customers may. In this way, a requirement for one market segment eventually becomes a requirement for all.

But securing the software supply chain is a complex undertaking. The breadth and depth of the software supply chain affords multiple points of entry for malicious actors who will always look for the weakest link in the chain to exploit:

- **Breadth** - Most organizations work with more than one open source language, and import their code from more than one public repository. Because there are no industry-wide standards in place today, each language and repository must be treated uniquely.
- **Depth** - There is a large set of best-practice security and integrity controls that can be implemented in order to scrutinize imported open source components. How far an organization is willing to go down this rabbit hole is largely dependent on the appetite for risk, as well as time and resources.
- **Change** - No supply chain is ever set in stone. Open source authors change and the packages they produce are constantly updated, become vulnerable, and get patched. Languages go EOL, repositories move, trusted vendors change, etc. Keeping up with it all will, once again, demand exorbitant amounts of time and resources.

As part of securing their supply chain, US agencies require software vendors to adopt secure software development practices as defined by the National Institute of Standards and Technologies (NIST). NIST's best practices extend from application architecture and design to software threat modeling to secure software build and delivery, and include a number of key deliverables, such as:

- Attestations from the software producer
- Software Bill Of Materials (SBOMs) and documented processes to validate code integrity
- A programmatic way to check for and automate vulnerability remediation

All of these requirements and deliverables mean that a comprehensive solution is likely too complicated and costly for any one organization to implement on their own. The ActiveState Platform is a cloud-based service that can help you secure the breadth and depth of your open source supply chain while delivering attestations, SBOMs and automated vulnerability management in a single solution that fits with your existing development processes.

activestate