Project Trebuchet: Mitigating SUNBURST-style attacks with open-source tech

Trevor Rosen
Project Trebuchet
Mitigating SUNBURST-style attacks with open-source tech
Quick Intro

This is me

- Principal Architect at SolarWinds
- Specialize in SaaS side of the business
- Lead for Project Trebuchet, the post-SUNBURST build system
- App dev with a background in K8s, containers, infosec
- Lead CI/CD, dev experience, dev sec ops at SWI
What Does SolarWinds Do?
Many products, several of which you may already know

- Market leader in network management software
- Orion® Platform (flagship product) is used by most of the Fortune 500, gov agencies, tier-1 network operators, etc.
- Also own SaaS-based offerings like Pingdom®, Loggly®, Papertrail™
- Over 50 products—one was compromised in SUNBURST (Orion Platform)
What Is This Talk About?

- Overview of how SolarWinds is using CNCF/CDF/SSF tech to create a new build system for cloud and on-prem software
- Necessarily pretty high-level – this is a big topic
- Focusing on the pieces of the build system itself
- Intended to help folks understand what exists, how to use
- Eventually all our original kit will be FOSS (~6 months)
Let’s define “supply chain attack”
“unauthorized modifications to software packages”
- Google
Types of Supply Chain Compromise

There are at least two...

**Third-Party Code Compromise**

**First-Party Compromise**

**?????**

**Federal Investigators looking into breach at software code testing company Codecov**

The breach happened in January but was not detected until April.

**Threat Research Blog**

Highly Evasive Attacker Leverages SolarWinds Supply Chain to Compromise Multiple Global Victims With SUNBURST Backdoor

December 13, 2020 | by FireEye


Source: [https://arstechnica.com/gadgets/2021/06/microsoft-digitally-signs-malicious-rootkit-driver/](https://arstechnica.com/gadgets/2021/06/microsoft-digitally-signs-malicious-rootkit-driver/)
What Happened?

Quick review of the SUNBURST breach
What Happened in the Hack

• A malicious DLL was inserted at the right moment in the build process, via hacked msbuild binary
• Customer installs of the Orion Platform upgraded to compromised versions
• The DLL had an innocuous name that looked like other class names
• No source code was compromised
• It mimicked “call home” traffic to our stats portal
• Used DNS for CnC, stayed dormant for certain IP blocks
What We Did After Discovery

• The Orion Platform is ~10 million LoC developed over nearly 20 years
• Had to decompile huge numbers of DLLs, compare decompiled source to original
• Wrote scanners to compare symbols in PDBs to original source code
• Worked around the clock through New Year’s Day
• Folks from Manila to Krakow turning every system upside down
Conclusions

• This adversary was very, very good
• Fewer than one hundred customers were affected
• SolarWinds likely attacked because of nature of the Orion Platform
• We would need to develop a state-of-the-art build infrastructure with the increasing sophistication of threat actors

“We haven’t seen this level of sophistication matched with this kind of scale”

Brad Smith
President, Microsoft

The Fix

Project Trebuchet, a consensus-attested system
We must completely rethink and recreate a major portion of the SDLC!
Four Top-Level Requirements

Any new build system would have to look like this

- **Ephemerality** of infrastructure
- **Determinism** wherever possible
- **Consensus** via duplicate systems
- **Proof** of every step taken in a build
"A developer shall have **fine-grained control** over what she builds, but **zero control** over how it is secured and validated"
SaaS Offerings Won’t Work
As of January 2021, nothing could satisfy the Golden Rule

• Neither CircleCI nor TravisCI nor GitHub Actions gave us what we needed
• (We definitely tried)
• It’s not enough to self-host
• You have to validate and extend developer definitions of build workflows
• We decided to go with Tekton, which is based on Kubernetes, to get the mechanics we needed
**But *Why* Won’t SaaS Work?**

Maybe it will be possible someday, but…

- With Circle/GHA/Travis, entire build definition is in repo
- DRY is primitive and unenforceable
- No overarching “authority” can add new checks or validate workflows
- We need to be able to bring receipts, which means standardization and enforcement
- K8s is excellent at architectures which involve mutating user-provided data
Dev Experience Looks Familiar

Similar to what you’d get in a SaaS CI product

- Write YAML, defining Tasks with Steps
- Each step happens in a container image
- Semantics for mounting volumes, passing data between tasks
- Tasks can be in-line or referenced in a separate repo
- Easy DRY

```yaml
- name: build
  taskRef:
    name: kaniko
    bundle: ghcr.io/solarwinds/tekton-task-catalog/kaniko:<HASH>
  params:
    - name: IMAGE
      value: path.to.reg/imagename/$(tasks.tag.results.tag)
    - name: EXTRA_ARGS
      value: "--reproducible"
  workspaces:
    - name: source
      workspace: working-dir
  runAfter:
    - tag
```
Tekton + GitHub = ?
Tekton is great but not GitHub-easy OOB

- We built several pieces of kit to marry Tekton and GH
- GitHub App fetches/validates pipelines
- Proxy for tasks to talk with
- K8s controller to report results to GH Checks API
Basic Tekton Pipeline With Extend/Enforce Capabilities
Attesting What We Build

Everything needs proof!

- We need to know the provenance of everything
- We must produce comprehensive records
- Those records need cryptographic guarantees
- In-Toto gets us started
In-Toto Basics
A framework for tracking what we do when we build

• Software supply chain integrity tracking
• Started as university research project, now in CNCF
• Open framework of standards and tools
• SolarWinds engineers have implemented some specs
In-Toto and SLSA for Attesting

In-Toto implements the SLSA attestation spec

- ITE-6 proposes an Attestation format for In-Toto based on SLSA’s spec
- It is pre-1.0, but we’re participating in dev and using it now
- **Subject**: a thing built
- **Predicate**: the method and ingredients
- **Signature**: crypto guarantee

Example in English:

Artifact "sha256:11be5" was built by GitHub Actions from github.com/foo/foo.git@7f8d2 Signed, GitHub

Source: https://github.com/slsa-framework/slsa/blob/main/controls/attestations.md
Tekton Chains to Watch it All

Signed proof of every step in every build

- Watches the K8s API for completed TaskRuns
- Produces an attestation, signs it, writes to the database
- Ensures every completed Task in a Pipeline has a record
- Comprehensive metadata
- SWI contributed support for In-Toto attestation spec
Pipeline With Attestations

- App repo
  - Webhook POST
  - GitHub app
  - Mutating webhook
  - Tekton Chains
    - KMS
    - Doc DB
  - (other kit)

- Tekton Pipeline
- Amazon ECR
- Amazon S3

© 2021 SolarWinds Worldwide, LLC. All rights reserved.
Dual Builds
More than one system should agree

- Unitary build system = BAD
- Need *consensus* for security
- Our design makes it easy to build more than once
- We need a *second* system (at least) that is isolated
- Build everything in parallel in second system
Reading Results

GitHub app ➔ Mutating webhook ➔ Tekton Pipeline ➔ (other kit) ➔ Tekton Chains ➔ Doc DB ➔ ETL pipeline ➔ Read store ➔ (release gates)

App repo ➔ Tekton Pipeline ➔ (other kit) ➔ Tekton Chains ➔ GitHub app ➔ Mutating webhook ➔ Tekton Pipeline ➔ Cloudevents ➔ KMS ➔ Validation cluster ➔ Standard cluster

Amazon ECR ➔ Amazon S3

© 2021 SolarWinds Worldwide, LLC. All rights reserved.
Agents and Jobs
Can’t do everything on Tekton

- Agent receives messages from Tekton Task
- Build on macOS, AIX
- Dedicated machine spun up by cloud API call in Task
- Task consumes agent results for Chains
Miscellaneous Details

- We do vuln analysis with policies defined in OPA
- ALB is locked to GitHub’s published CIDR ranges
- No egress out of VPC other than to GitHub
- We talk to IBM Cloud for AIX jobs with the agent
Concluding Thoughts

Shout-outs, musing on the world
You’ll Probably Experience a Breach

• People get hacked all the time – don’t think it can’t happen.
• Be humble about the challenge of securing your surface.
• Have sympathy for your security teams. Help them!
• Support app sec – shift security left. Tons of tools to help.
• Be excellent to each other!
Special Thanks

Just a few of the SolarWinds people who worked on SUNBURST mitigation and the construction of Trebuchet:

Fredrik Skogman, Chris Erway, Ondrej Fitzek, Harry Griffiths, Tim Danner, Karlo Zatylny, Kate Asaff, Marta Marets, Tomas Saghy, Brian DeHamers, Cody Soyland, Pawel Kedzior, Tomas Kuty, Grzegorz Glogowski…

…and everyone else (everywhere) who has been in the trenches these last 10 months!
The SolarWinds, SolarWinds & Design, Orion, and THWACK trademarks are the exclusive property of SolarWinds Worldwide, LLC or its affiliates, are registered with the U.S. Patent and Trademark Office, and may be registered or pending registration in other countries. All other SolarWinds trademarks, service marks, and logos may be common law marks or are registered or pending registration. All other trademarks mentioned herein are used for identification purposes only and are trademarks of (and may be registered trademarks) of their respective companies.