

Zero Trust of Software in your Supply Chain

Dan Tejada
President
Cantada, Inc

Cantada, Inc.

Cybersecurity Provider to Several National Security Entities

- Reverse Engineering
- Embedded Software and Hardware Vulnerability Analysis
- Enterprise Software Vulnerability Analysis

Reverse Engineering Domain Expertise

- Network equipment
- SCADA / ICS
- Consumer Devices
- Consumer Applications
- Enterprise Software

In House Tool Development and Automation

- AutoRE™
- SwIFF

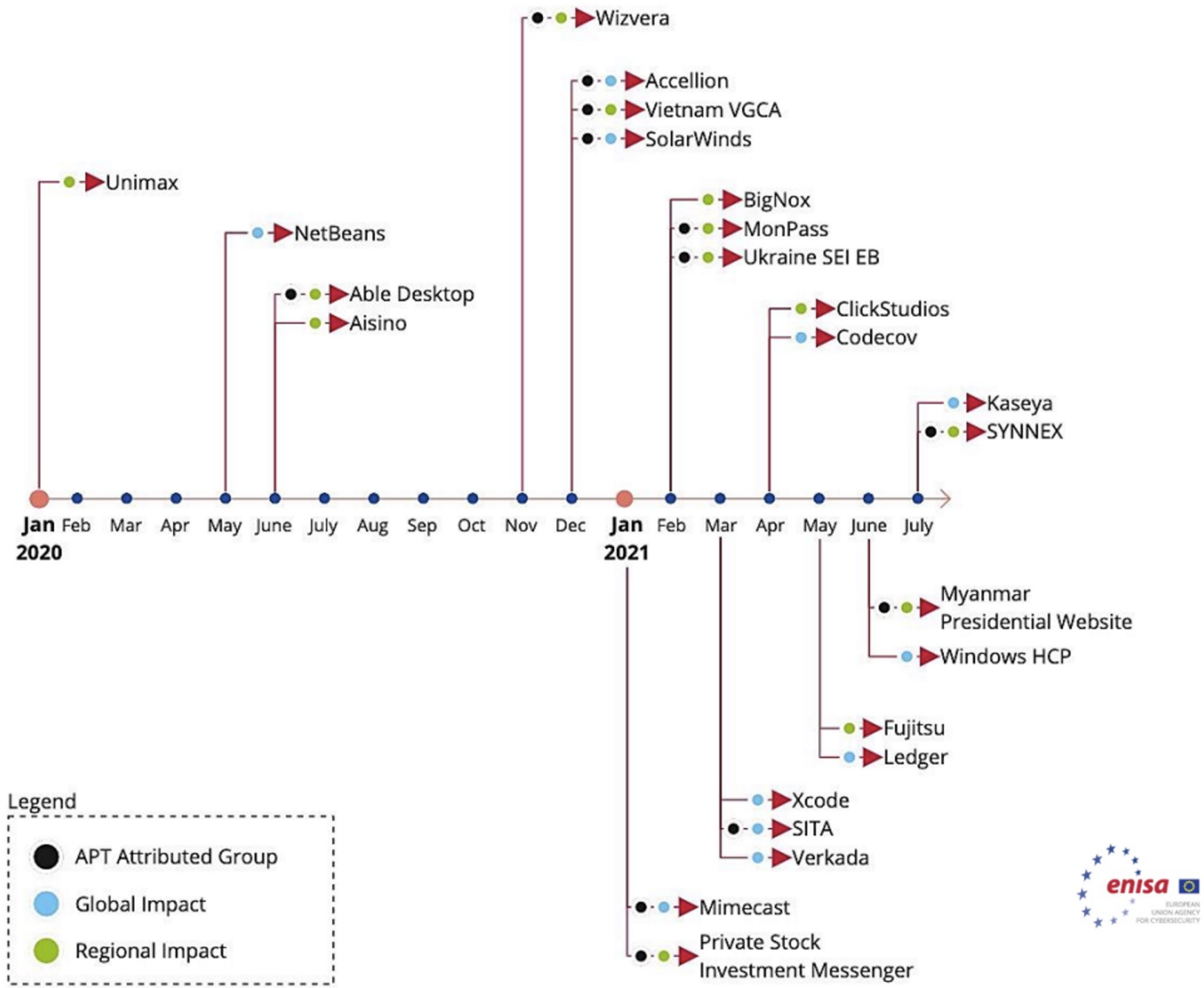
Supply Chain

- Number One emerging threat vector



Recent Supply Chain Attacks

Why We're Here...



Why should you care?

You are more vulnerable to your Software Supply Chain than you think

SBOMs are critical to securing your Software Supply Chain

- Accurate
- Sufficiently Detailed

3 Scenarios

- Vendor Software with an SBOM
- Vendor Software without an SBOM
- Your own In-house developed Software

SwIFF: Our Zero-Trust approach

Zero Trust Model

Visibility & Analytics

- Continuous Verification
- Transparency

Automate & Orchestrate

- Keep pace with the CI/CD release cycle

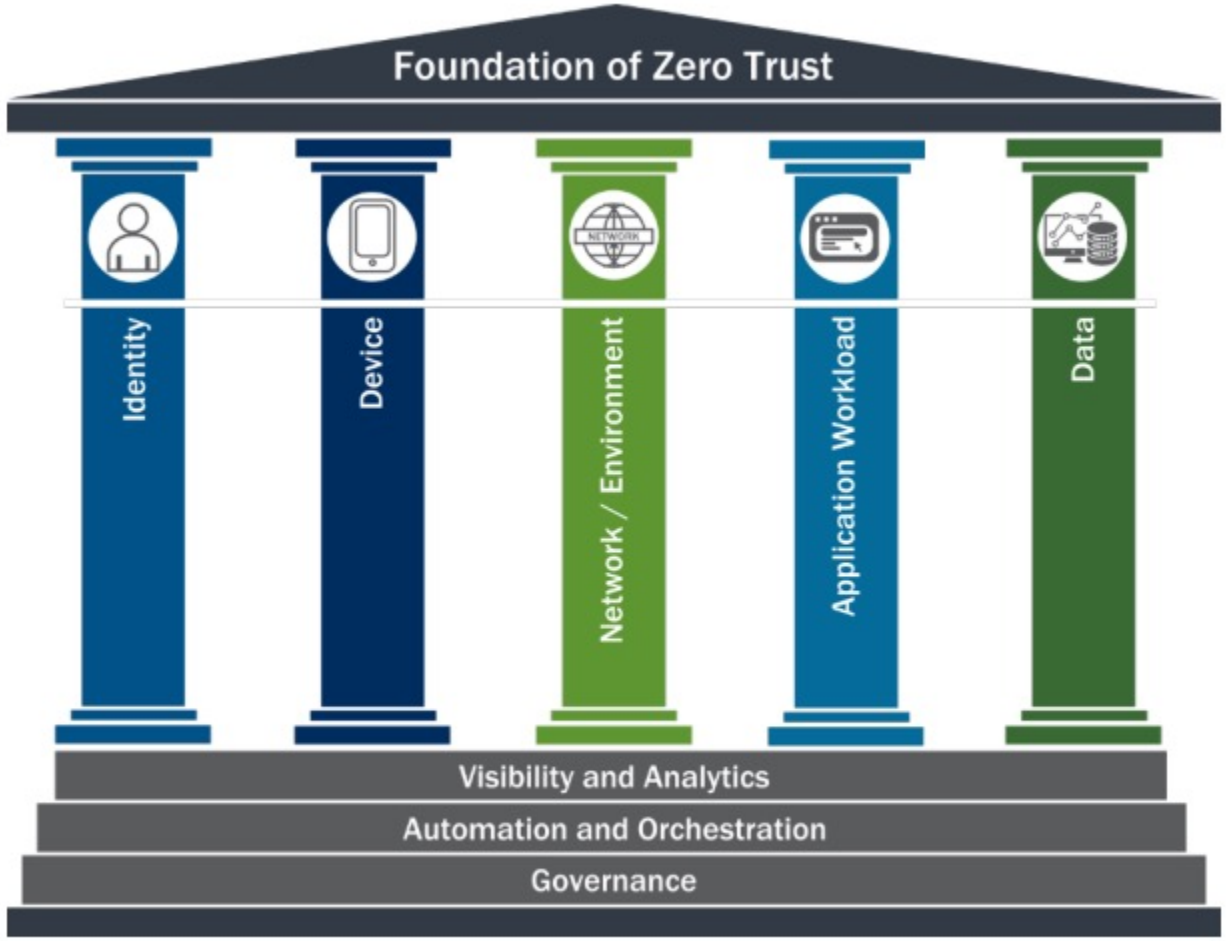


Figure 1: Foundation of Zero Trust[7]

*CISA Zero Trust Maturity Model Pre-Decisional Draft, June 2021

Vendor Software

SBOM supplied by vendor

- Can you trust it?
- How do you know it's accurate?
- How does the vendor know it's accurate?

Reasons why an SBOM can be inaccurate

- Incorrect Package versions
- Omit components they don't want you to know about
  - Spyware
  - Telemetry
  - Usage data collection
  - User information
- Omit usage of licensed software
- Changes occurred in the Supply Chain

Verification is the Answer

Vendor Software

No SBOM supplied by vendor

Reasons why you can’t trust it

- Completely Opaque
- Omit components they don’t want you to know about
  - Spyware
  - Telemetry
  - Usage data collection
  - User information
- Omit usage of licensed software

Binary SBOM Estimation is the Answer

In-House Developed Software

SBOM Generated during Build Process

- Can you trust it?
- How do you know it’s accurate?

Reasons why an In-House SBOM can be inaccurate

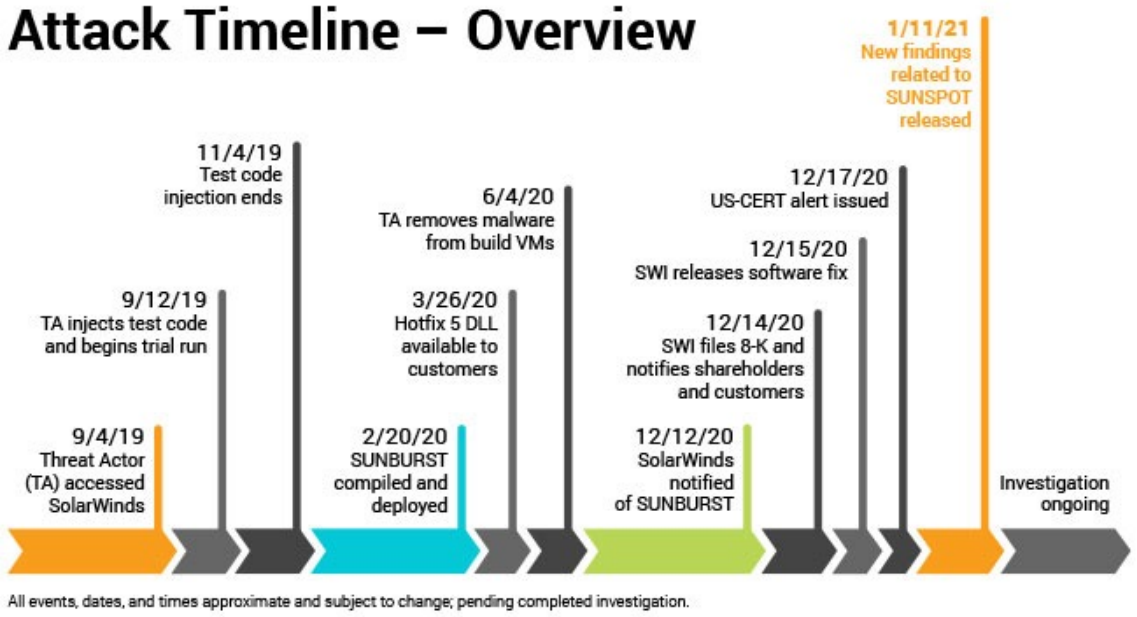
- Incorrect Package versions
- Defects in the build system
- Source code not attributed
- Disgruntled or malicious developer
- Malware introduced by malicious means
- Malware introduced by accidental means

Verification is the Answer

Case Study

SUNBURST
SolarWinds Orion

- Supply Chain Attack
  - DLL modified late in build system
  - Orion update with Signed CERT
  - Supply Chain appears secure
- Prevention
  - Detailed SBOM is the first step
  - SBOM Verification may have caught it!





The need for ***Deep Binary Verification*** has NEVER been greater!

SwIFF
SBOM Verification and Estimation

How it works:

- Binary Feature Recognition
- Highly Automated Technology
- No Source Code necessary
- Immune to Obfuscation and small changes

Enables Discovery of:

- Open Source and Proprietary Software Components
- Known Vulnerabilities
- Embedded Telemetry
- Software Licensing Information
- Malware

Based on Decades of Reverse Engineering Expertise

Real Life Examples

Discovered published vulnerabilities in a widely used Consumer Device weren't patched

Stolen Proprietary Software Components in a vendor's firmware

Sophisticated Facial Recognition Components in the Zoom Client

Demonstration

- Log4J version discovery and verification using SwIFF
- Log4J vulnerability discovery

Questions?

- Cantada, Inc.
  - Brian Tague, brian.tague@cantada.com
  - Dan Tejada, dan.tejada@cantada.com
  - Richard Dynes, richard.dynes@cantada.com