Ensuring the Integrity of the Software Supply Chain

Position Paper

Crytica Security
7655 Town Square Way
Suite 212
Reno, NV 89523

Website: CryticaSecurity.com
Email: info@cryticasecurity.com

1. SOFTWARE SUPPLY CHAIN VULNERABILITY

To get to the root of the problem, go to the source.

Software Supply Chain Attacks

Some of the most devastating weapons in the cybercriminals' and malicious state actors' arsenals are attacks that target the software supply chain. In these attacks, the malicious actors manage to infect a software development environment itself and then install malicious code inside of officially distributed software. With the malware hidden inside an official software distribution, even a comparison between the software received from the software vendor and the software installed in an environment would not disclose any malware, as the software being distributed by the vendor already contains the malware.

Industry Response to Software Supply Chain Vulnerability

Software supply chain attacks, such as the infamous SolarWinds hack, have quickly and correctly shattered industry confidence in the reliability and the safety of the entire software industry. Attempts to prevent software supply chain vulnerability have focused mostly upon the quality assurance aspects of software development. There are now various proposed guidelines for writing "secure code" and for testing new code for unanticipated ("out of design") behavior. However, the adoption of these guidelines and their requisite enforcement have been spotty at best. Industry-wide standards do not currently exist, and the enforcement of any such standards is rife with obstacles, such as turf wars, factionalized proposed solutions. In order to secure the software supply chain, more effective measures are required.

2. WHY SUPPLY CHAIN INFECTION ATTACKS SUCCEED SO WELL

A Detection Problem

Most supply chain attacks usually begin with an infection of the development environment itself. Seemingly innocuous code is used for this type of infection. It is code that blends well into any normal software development environment. The "innocuous" code is then used to modify the deliverables, that is, to modify the software packages, being created in the development environment. By integrating virus payloads into these official software products, the "malware" becomes part of the official software releases.

There are many methods the criminals can use to accomplish this type of attack. For example, "library code" can be modified with malicious code, malicious modules can be included in the "makefile" builds[1], and/or the malicious code can be inserted into source code files prior to their compilation. Once the malicious virus/malware payloads

[1] A "makefile" is a script that "builds" the final software package. It compiles the source code (the human readable programs) into object code (machine readable instructions), links in standard object code from the "libraries" and combines them all into an "executable" package. This package may be standalone or may have additional links (hooks) into other system libraries, which are then loaded in at the time of package execution.

are installed in and integral to the deliverables packages, they are distributed as part of the standard "official" software.

Traditionally, the primary challenge in defending against these types of attacks is the **detection problem**. Unfortunately, most malware detection systems do not and cannot detect malware that is comprised of "seemingly innocuous code". Unless the "innocuous code" has been previously identified from a previous attack, the infecting code is invisible to most detection systems. This shortcoming in most cybersecurity detection systems, this inability to detect truly new and innovative malware code, is what has enabled supply chain malware attacks to succeed so well.

3. A NEW APPROACH: EFFECTIVE MALWARE DETECTION AT THE SOURCE

A more promising approach to securing the software supply chain is to require software development environments to install more effective malware detection platforms. These malware detection platforms would need to be able to detect "new" and "seemingly innocuous" malware infections. In light of the mechanics of typical software supply chain attacks (discussed above), the rationale behind this type of defense is obvious. With a malware detection platform that can detect even new and seemingly innocuous malware, the software supply chain attackers would not be able to infect development environments, and therefore would not be able to create infected "official" software distribution packages.

Such an approach would not be instead of the quality assurance standards currently being bandied about, but in addition to them. However, requiring effective malware detection in the source code environment is a measure that is far easier and faster to implement, and far less complicated to enforce, than the establishment of agreed upon, industry wide, quality assurance standards. Ultimately, both should be used, and both are beneficial. However, effective malware detection is certainly a necessary component; malware detection that can detect new and seemingly innocuous malware infections.

4. CRYTICA'S EFFECTIVE MALWARE DETECTION PLATFORM

The requisite effective malware detection does now exist. Crytica Security has created a malware detection platform that can and **does** detect new and seemingly innocuous malware infections. It is a platform whose algorithms are deterministic rather than probabilistic, and thus it does not generate false positives. It is a platform ideally suited to the detection of, and hence the prevention of, software supply chain attacks.

The reason that Crytica's malware detection platform is so much more effective than others' at detecting supply chain attacks, is that Crytica has taken a very different approach to malware detection and to the characteristics required for a malware detection platform. In brief, Crytica's unique approach consists of two distinct and essential components:

1) A Unique Malware Detection Algorithm: Crytica's Malware Detection algorithm is based upon the premise that **any** unauthorized changes to a device's instruction set **is** malware.

   a) The concept behind Crytica's algorithm is simple, binary, and deterministic: Either a change in a device's instruction set has taken place, or it has not. If a change has occurred, it is either an authorized one, or it is not. It matters not what the unauthorized code change affects. If a change is unauthorized, it is malware. This algorithm can detect, with equal ease, new malware, polymorphic malware, preemptive malware, and even Artificial Intelligence (AI) Generated malware. There is no reliance upon previously identified malware, and no reliance upon probabilistic (e.g., AI) algorithms.

   b) Crytica's algorithm stands in sharp contrast to most other malware detection systems. Most other malware detection systems rely upon finding previously identified malware. They are either looking for an exact match and/or for code and/or a behavioral patterns that are similar to some previously identified malware. Those systems tend to use, as the basis for detection, inexact, probabilistic AI algorithms. Unfortunately, these result in a plague of false positives. Indeed, there is today an almost inexplicable increase in reliance upon AI in malware detection tools. This is true even though the cyber enemies have access to the same, and possibly superior, AI than the defenders do. The result can be, at best, an endless AI arms race, one in which the attackers have, and shall always have, a distinct advantage. These AI algorithms are incapable of detecting truly new and innovative attacks, and struggle with polymorphic, preemptive, and AI-generated malware. Certainly "innocuous" appearing files are generally beyond their ken.

2) A Uniquely Resilient Platform Architecture: Attacks against a malware detection system are inevitable, e.g., the recent wave of preemptive malware attacks. An effective malware detection platform must not be disabled by any malware attacks launched against it. Crytica's Malware Detection platform is designed to be resilient and to absorb attacks. It is designed to "bend-not-break".

   a) Crytica's Malware Detection Platform is constructed upon the foundation of a patented mutually monitoring mesh of components. Its probes are tiny and replaceable. They operate as application-level processes, so even if one is compromised, it can do only minimum damage to its host device. When probes are compromised, they can easily be destroyed by the Crytica platform controllers and almost immediately replaced. Attacks against Crytica's platform may damage and destroy various components, but they will not prevent the platform from continuing to operate effectively.

   b) Unfortunately, most other cybersecurity systems are designed to be impenetrable walls. They are designed upon the assumption that they themselves will not be breached nor compromised. Also, they tend to operate with administrator-level permissions. History, however, has repeatedly proven that there is no such thing as an impenetrable wall. An attacker, with sufficient

motivation and sufficient resources, can penetrate any wall. In the cybersecurity realm, when an "impenetrable" defensive wall is breached and the defensive system is running with administrator permissions, the result is a disastrous example of a single-point-catastrophic-failure. When a typical cybersecurity is compromised, the attackers would have no need to inject any malware. By using the compromised cybersecurity system itself, the attackers would able to operate as system administrators. They would completely control the infected device.

In light of its unique detection abilities and its resiliency to attacks, Crytica Security's malware detection platform is the perfect platform to detect, and thereby prevent, software supply chain attacks.

The logic is obvious: "If you can't detect, you can't protect.™ " Crytica can detect, so that we all can protect. For more information on Crytica's malware detection platform, its resilience to attacks launched against it, and its ability to detect software supply chain attacks, contact Crytica Security, Inc.

Website: CryticaSecurity.com
Email: info@cryticasecurity.com

