Module 4: Supply Chain Attacks
The Mantra: "Compromise One To Compromise Many"
Software Supply Chain Attacks-Module Overview

In this module, we will describe the dramatic consequences of supply chain attacks, as illustrated by two case studies: the SolarWinds (2019) and the NotPetya (2017) attacks. We will also study the emerging Log4j attack affecting millions of users. In addition, we will consider the vulnerabilities of cloud computing as a target of a supply chain attack. In particular, we will:

- Describe a model of a software supply chain, focusing on the development and distribution of software updates
- Discuss how two very complex Advanced Persistent Threats (APT) introduced “trojanized” software to the update process in both the SolarWinds and the NotPetya attacks, but with very different results
- Introduce the emerging, complex Log4j attack, and its vulnerabilities
- Address recent cyber supply chain concerns regarding cloud computing
- Discuss some possible mitigation area
Current Events – CISA Industry Alert - review

“Understanding and Mitigating Russian State-Sponsored Cyber Threats to U.S. Critical Infrastructure” CISA, Jan. 11, 2022, rev March 1, 2022

• Ukraine Infrastructure attacked by Russia in 2015-2016

• In 2017, Russia launched the NotPetya attack directed at Ukraine, but caused global collateral damage in $Billions

• In 2019, Russia attacked multiple US Federal Agencies and their support functions (SolarWinds attack)

• Ongoing tensions led to the CISA Alert Jan. 11, 2022. In view of the invasion of Ukraine in February 2022, this Alert was revised on March 1, 2022, with concerns of Critical Infrastructure attacks
What is a Supply Chain?

• A network of functions to get goods from the source to the end users
• Some functions: Raw material acquisition, product design, manufacture and production, distribution, retailing, delivery to end user
• Product might be hardware or software
The Global MTS as a Physical Supply Chain

This shows the global supply chain extending from supply of raw materials to product delivery to end customers. On March 21, 2021, a choke point blockage affected supply a half a world away. Unavailability of raw materials also has global repercussions.
Software Supply Chain Attacks

To compromise multiple targets simultaneously, threat actors (TA’s) attack a **Third-party Trusted Agent**. The TAs mantra: “Compromise One to Compromise Many”
Third-Party Trusted Agents (or Trusted Third Parties)

In managing an enterprise that uses advanced technology, “Access Privileges” to OT operations and maintenance functions are often “Delegated” to outside firms (Third Parties) that possess the necessary, specialized knowledge. The operational integrity of these functions is critical.

• Hence Delegated Access Privileges should only be given to **Trusted** Third Party Agents
• But how do you/they know whether their laptops, portable media, etc., are clean?
  o You don’t/they don’t!
• Can TAs hack into Third Party Delegated Access computer ports?
  o One of the first places they look!
The Update Software Development Environment

Software updates are distributed by the developer to all its customers and users. Below are functions in a professional development environment.

**Professional Development Environment**

- **UPDATE REQUIREMENTS**
- **MODULE DEVELOP**
- **INTEGRATION and TESTING**
- **QUALITY ASSURANCE**
- **UPDATE SERVER**

**DIGITAL SIGNATURE**

- Patches
- SW Bugs
- New Features
- Design
- Code
- Test
- Integrate Modules
- Integration Test
- Rigorous Field Test
- Distribution

CUSTOMER 1

CUSTOMER 2

...
Digital Signatures (DS) in the Software Supply Chain and Their Limitations

• A digital signature affixed to software (e.g., applications, machine code, messages) ensures
  o Authenticity – It is from who it says it is from
  o Integrity – It has not been altered since it was signed and sent. Hence protects from Man-in-the-Middle attack

• A digital signature is created by encrypting the message with the author’s Private Key
  o The slightest change to the message after the digital signature is calculated will be instantly detected by the receiver, and rejected
  o **BUT any changes to the code before the DS is calculated will not be detected by the receiver’s digital signature check.**
  o **Hence a digital signature is necessary but not sufficient.**
Digital Signatures in the Software Supply Chain and their Limitations

• Digitally signed code from a trusted agent may give confidence that the product is safe. **The SolarWinds exploit proves this is false.**

• It also proves that trusted agents (and third part vendors) cannot necessarily be trusted! Hence the emergence of **Zero Trust** policies:
  
  o Don’t trust/incorporate any critical software without testing for malware and other assurances before loading it
  o Require authentication on all downloads, and internal handoffs
  o Create requirements on vendors to ensure their product is trustworthy, and free from development process attacks
SolarWinds Exploit

SolarWinds provides network management (NM) software to many thousands of customers including federal government agencies. One critical customer is a network security company called FireEye.

The environment:
- SolarWinds Provides Software Products to 300,000 customers
- Major Network Management (NM) and Performance called Orion
- SolarWinds issues automated updates to customers
- SolarWinds is, or has been, a Trusted Third Party
Malware Insertion into SolarWinds Orion NM Software

Sunburst is the name for the SolarWinds update containing backdoor malware. The malware was inserted into the development process and went unnoticed.

Canonical Update Development Process

![Diagram showing the Canonical Update Development Process]

- **UPDATE REQUIREMENTS**
- **MODULE DEVELOPMENT SERVER**
- **INTEGRATION and TESTING**
- **QUALITY ASSURANCE**
- **UPDATE SERVER**

Threat Actor breaches the development environment and introduces Sunburst backdoor malware to Orion update code.

**Commercial Cloud Service**
Command and Control Center

Threat Actor breaches the development environment and introduces Sunburst backdoor malware to Orion update code.

Orion update distribution to 30,000 entities

Ref: https://clipart.me/technology/girls-and-computer-vector-46-38029
Trojan Horse Background

Originally (c.20 BC) a gift from Greeks to Trojans to breach their walls

• Trojan (or Trojan Horse)
  • A Trojan Horse is an object that seems innocuous, but contains danger
  • In cybersecurity it is software that seems OK but contains hidden malicious code. The malicious code or malware may establish a backdoor, search and map the victim’s system, exfiltrate key data, etc.
  • Software, such as updates, that may appear innocuous but contain hidden malware are said to be “trojanized”
Trojan Malware Via Backdoor

How does a Threat Actor breach a development environment and insert malware to create a *trojanized update*?

**Breach may have occurred due to password attack:**

- **Brute Force:** Identify a developer and impersonate them. Try all possible passwords, especially popular ones
- **Password Spray:** Try a few most popular PW on every employee, to thwart account lockout (i.e., too many tries)
- **CNN Feb 26, 2021 — “Former SolarWinds CEO blames intern for 'solarwinds123' password leak” of Nov. 2019**

**Backdoor:** A method of accessing a computer system or network bypassing system security

- **Pro:** Desirable for rapid access by maintenance and development personnel, especially in Operational Technology emergencies
- **Con:** Discoverable by Threat Actors as an Initial Access point

“Beware of Greeks bearing gifts”
Virgil, *Aeniad*
Trojanized Orion Business Layer

The basics of how Sunburst works:

- SolarWinds Orion Business Layer is a digitally-signed component of the Orion software, used for software updates. It gets downloaded to thousands of connected systems via the update process.
- The Trojanized Orion Business Layer is called Sunburst by FireEye.
- The Trojanized version works correctly, even with Sunburst malware installed (hence Trojan Horse).
- Upon being downloaded, Sunburst sleeps for 12-14 days in victim’s network, then wakes up at randomly-determined times in different systems. (Part of its detection evasion).
Delivery of Trojanized Orion Update

The evasive steps taken to inject and distribute the malware, and remain unnoticed.

Hypothetical Update Development Process

1. Developer codes update

2. Threat Actor (TA) breaches the development environment and introduces Sunburst backdoor malware to Orion update code

3. DS Process signs new trojanized version of Orion update. The DS includes the malware, (so the malware is not detected when the victim checks the signature).

4. DS signed Trusted Trojanized updates sent to thousands of victims

Ref: https://clipart.me/technology/girls-and-computer-vector-46-38029
Exploitation of Victim’s Computer Network by Threat Actor via Sunburst – A Summary of Attack and Evasion

Sequence of attack:
1. Sunburst gets downloaded to victim, as discussed. Backdoor established. Backdoor bypasses system security
2. Backdoor malware sleeps for 12-14 days, then awakens at a random time (Evasion)
3. Backdoor checks environment to make sure it is on victim’s computer, e.g., not SolarWinds’, that it is not being observed by analyst or security systems. (Evasion)
4. Checks that it has email capability.
5. Acquires information on compromised host, including which one of many it is on.
6. Sends email to C2 indicating it is operational, and exfiltrates any collected data.
7. C2 may begin keyboard-generated attack that might include gaining higher privileges, credential theft, lateral movement to other areas, further data exfiltration. C2 traffic appears local (Evasion)
SolarWinds Exploit – How was it discovered?*

The SolarWinds attack was discovered by the security firm FireEye by analyzing its own “log” for unusual traffic.

• FireEye discovered that a backdoor (Sunburst) had been established in its own internal network, and that a key FireEye forensic tool had been exfiltrated (Red Team-Blue Team tool which is never sent to its customers) Dec. 2020
• FireEye traced the exploit back to a SolarWinds Orion update that it received.
• Since the update had been digitally signed, the SolarWinds exploit must have occurred early in the development process. TA testing of Sunburst seems to have begun as early as Oct. 2019
• Because an alert security firm was attacked as part of the government network, it was able to initiate mitigation efforts

*See “Analyzing Solarigate, the compromised DLL file that started a sophisticated cyberattack, and how Microsoft Defender helps protect customers - Microsoft Security.html”
Case Study: Maersk NotPetya Attack

A drive-by, but not-ransomware, supply chain attack
On June 27, 2017, NotPetya Hit Maersk

- On a global basis, screens went black or received a (pseudo) ransomware demand.
- Files and executables were corrupted or wiped. No decryption keys were available.
- Hundreds of computers and servers were destroyed; business, operations and phone systems were inoperable, as were physical security at doors, port gates, etc.
Physical Port Security Was Also Rendered Inoperable

- Trucks could not receive logistics paperwork, including container numbers, access or egress clearances, locations at port to drop off cargo
- Seventeen of fifty-seven of Maersk’s ports globally were thus shut down.
- Trucks were stranded for miles. Here, trucks loaded with containers are lined up outside a terminal at the Jawaharlal Nehru Port Trust in Mumbai, India, Thursday, June 29, 2017.

Jawaharlal Nehru Port, Mumbai, June 29, 2017
NotPetya Virus

Maersk was not an intended target, nor was there an extortion attempt. Hence it appears to be a drive-by act of destruction

  • **Appeared**, at first, like a Petya ransomware virus attack

  • **But**, cannot be decrypted by user response (random number is generated rather than ID code, hence NotPetya)

  • Characteristics:

    o Virus enters victim system via connection to Ukraine network

    o Self-Propagating, does not require user action, such as user being tricked into downloading malware; propagates very rapidly

    o Not specifically targeted at Maersk; hence called a Drive-By attack

    o Purpose is destructive, not criminal extortion, perhaps cyber warfare?
Maersk NotPetya: How Did It Happen?

A Ukraine company, Linkos, makes a financial management product called M.E.Doc. Linkos was breached and NotPetya installed on M.E.Doc. When Linkos downloaded updates, all connected customers were corrupted. The attack was meant to cripple Ukraine infrastructure.

M.E.DOC UPDATE PROCESS

UPDATE REQUIREMENTS → DEVELOPMENT SERVER → UPDATE SERVER

Threat actor inserting NotPetya:
Drive-By Victims

In this timeframe, the Maersk Odessa office ordered the M.E.Doc software to be downloaded to its office. That enabled the global Maersk crash. Other companies, either directly or indirectly, were affected, resulting in losses of billions.

Threat actor inserting NotPetya

Some of the Drive-By Victims:
- MAERSK
- MERCK
- FedEx
- TNT
- MONDELES

M.E.DOC UPDATE PROCESS
- UPDATE REQUIREMENTS
- DEVELOPMENT SERVER
- UPDATE SERVER
The NotPetya Attack Impact on Maersk

A review of the Consequences of the attack is important in the development of mitigations

Consequences

• On June 27 AM, APM Maersk Terminal in Port of Elizabeth, NJ was shut down

• By June 29, 17 of Maersk’s 57 terminals were shut

• Globally, 20,000 computers and 4,000 servers were infected

• Clean copies of critical software and data files were found, although with some difficulty

• In about 10 days, Maersk resumed some operations; it took some months to restore to normal

• Restoration and Loss of Revenue to Maersk: $300M
If you were CEO of Maersk in 2017 and directed your Facility Security Officer to develop a Facility Cybersecurity Plan, what major safeguard would you expect it to include, given your recent experience?

What might be a working assumption for the future?
The Java Log4j Attack
The Log4j Attack

Log4j is an “Open Source” logging framework that supports Java-based apps. A logger keeps track of all transactions on a computer network


- Many threat actors (TA) were actively exploiting vulnerabilities in the Java-based Log4j logging framework. There are millions of Log4j installations in the US alone.

- CISA ordered that patches must be immediately developed and deployed (Note: Without extensive testing)

- Since the TA’s were actively attacking Log4j before the developers knew of the vulnerabilities, the developers had a zero-day lead to address the vulnerabilities. It is known, therefore, as a Zero-Day attack

- There are several versions of the Log4j. Some are Secure and some are Vulnerable to hacking. Here we designate them as Log4j.S and Log4j.V, respectively.
The CISA Emergency Conditions

The conditions that drove the Emergency Declaration were as follows:

• The attack was underway

• The vulnerabilities are easy to exploit, hence attack success is of high likelihood

• The Log4j software, and associated Java-based systems, is widespread in the Federal Network

• Agency information systems are likely to be compromised

• Hence impact will be significant
What is a Data Logger? (1)

A data logger in its operational configuration is shown here. The Java environment is chosen with the Log4j indicated.

The Log4j has three main functions:

- The Logger captures and retains all critical server operating data. This includes logins, queries, responses, interactions with databases, development updates, errors and error messages. It may query downstream databases for additional information.
- The Appender publishes data to all systems and entities that require it, including development, operations, forensics...
- The Layout function applies the desired data and report layouts
What is a Data Logger? (2)

A Data Logger is key to understanding the performance and security of a computer network

Data Logger Forensics

The Data Logger is invaluable in determining the existence of **Indicators of Compromise**, e.g.,

- PW attack might be indicated by excessive number of attempts, lockouts
- Exfiltration might be indicated if sensitive code were to be uploaded to an unusual network address, or to a Trusted Agent, during a routine update (this is how the SolarWinds attack was discovered by FireEye).
Hypothetical Illustration of Operation with a Secure Version of Log4j.S

To: https/GPS Gas Station Locator Service@JavaApp1.com
From: bbunin@stevens.edu
GPS: My GPS position is x,y
Get: location of nearest gas station

Log this info and query GEO-URL data base

URL:bbunin@stevens.edu
//Fetch maps service type
{google, garmin}

Here is the current garmin GPS code for URL:bbunin@stevens.edu
Here is app for gas stations

To: bbunin@stevens.edu
From: Java app//gps
{Here are nearby gas stations, and some other stuff too}

• NOTE: All these data *strings* are logged into the Log4j logger
Hacker constructs malicious message:
To: https/GPS Gas Station Locator
Service@JavaApp1.com
From: mal@hacking.org
Get: malicious code from mcs@hacking.org

1. Attacker
2. Log this info, launch query to server at MCS
3. Query from logger is directed to malicious code server, MCS. This action signals attacker that exploit has been successful.
4. MCS responds to query with malicious code (Malware), it gets logged on Log4j and passed to Vulnerable Java Server for execution.
5. Vulnerable Log4j.V Logger
6. Code can now be run remotely on Java server by the Attacker. Called a Remote Code Execution (RCE) attack
7. No error messages received implies that the Log 4j was vulnerable

Malicious Code Server (MCS): mcs.hacking.org
Several vulnerabilities were illustrated in the previous example. The ones shown are:

- Inadequate firewall protection between the internet and the Java environment allows malicious queries to pass to the Java server (link 1)
- The Log4j.V (i.e., vulnerable versions of log4j), writes strings to their logger with inadequate authorization (link 2).
- The Log4j.V accesses downstream systems without checking for their validity (link 3).
- There are other Log4j vulnerabilities as well. See:
  - https://www.cisa.gov/uscert/ncas/alerts/aa21-356a
Mitigations Underway

CISA and the Federal Trade Commission are taking aggressive steps towards remedying

- Cisco, Microsoft and others developed patches to remove vulnerabilities from Log4j/V systems
- CISA orders immediate implementation of patches; FTC proposed penalties for non-compliance
- To comply, entities must first determine whether they are using Log4j, and which version i.e., a secure version or a vulnerable version
  - It might be difficult to determine this: users and suppliers might not know what the components of their applications are
  - If a vulnerable version, it must be updated and/or patched
- Guidance applies to Operational Technology as well as Informational Technology
Open-Source Software - Pros and Cons

There are pros and cons to Open-Source software. But the status quo is unacceptable.

Pros
• Provides free, innovative, and useful packages
• Saves project development time

Cons
• A volunteer, not professional, development environment
• Might be focused more on innovative developments (the fun stuff), BUT
• Less focus on the mundane but essential, such as testing, maintenance, security, patching
• Widespread use of vulnerable software is leading to major internet issues
• Informal inclusion of Open-Source software makes it difficult to track in crises (like now)!

Standards and Regulation
• The industry might require “Software Bills of Materials” so suppliers and users can know what software is resident on their systems.
Cloud Computing and Risk

What is cloud computing?

• A computer center that provides on-demand computer system resources and services, especially data storage and computing power without end user management or capital investment

• Services are organized into three groups:
  o Software as a Service (SaaS)
  o Platform as a Service (PaaS)
  o Infrastructure as a Service (IaaS)

• Some of the larger companies:
  o Amazon Web Services (AWS): provides all the above
    • Some DoD apps on AWS
  o Microsoft Azure: all the above
  o Google Cloud: PaaS and Google apps
Cloud Computing – Its Attributes and Vulnerabilities

• Many good reasons for using cloud services
  • Low capital outlay, delegate IT functions to cloud
  • Match capacity requirements to business needs and variations, etc.
• But vulnerabilities are there
  • All vulnerabilities of on-premises resources apply to cloud

Plus
• Everything is accessible from internet, a fundamental vulnerability
• Strong safeguards needed from lateral attacks from others on shared resources, e.g., Nobelium

Note: Nobelium is the hacking group responsible for SolarWinds hack)
Mitigation Strategies Form the Basis of Cybersecurity Plans

This slide gives a summary of key elements of the three hacks described in this module. Their methodologies suggest some elements that should be considered in a Facility Security Plan (FSP)

• SolarWinds Attack: TA breached system via a password attack, injected malware, moved laterally, exfiltrated sensitive data. **Victims not sure what versions of software they had, and whether they were infected.**

• Maersk NotPetya Attack: Virus introduced into Ukraine critical infrastructure (via Trusted Third Party) and propagated worldwide to all connected systems. Attack was purely destructive and affected systems were wiped or destroyed. **There were almost no backup software files or data; no recovery plan.**

• Log4j Attack: Some Java apps used the log4j logger The log4j was comprised of Open-Source software (SW) with inadequate software security measures  **Users were unaware whether the Open-Source SW was on their systems**

Discussion: Based on these, what elements should a FSP contain?
Some Suggested FSP Elements-Prevention

• Enhanced password protection, sophisticated MFA authentication, focus on harmful links/emails/websites (Reduce success of IA)

• Rapid patch implementation, no obsolete/unmaintained software (Remove Vulnerabilities)

• Manage Delegated Access Privileges carefully. The third parties might themselves be compromised; maintenance ports a major vulnerability. (Resiliency-Minimize impact)

• Ensure that software development processes, including updates, contain no malicious code before downloading (Prevention)

• Require authorizations from all accesses; **Principle of Zero Trust**; no automatic downloads (Raise levels of security)
FSP Elements- Detection and Resiliency

Assume that a breach will occur. The next question is how to contain the damage and recover:

- Strong procedures should be in place to contain viruses, lateral attacks, privilege escalation etc. (see Mod 5 for segmentation and segregation).

- Key systems and data must be backed up, but the back-up must be secure from the attack, either highly segregated, firewalled, or off-line (think Maersk).

- A Software Bill of Materials (SBOM) should be in place, so facilities know what software they have, and whether they are affected by an APT.

- Create a strong recovery plan, and practice it a lot beforehand.

- For OT networks, plan should include operational continuity if IT/OT controls are lost.
Summary: Supply Chain Vulnerability

Highly sophisticated Advanced Persistent Threats

- Threat Actor exploiting Trusted Agent status and Delegated Access Privileges of service providers
- Principle of “Compromise-One to-Compromise-Many”
- Vulnerability of Open-Source Code
- Cloud services are not immune (CISA Report (10/25/21): NOBELIUM Attacks on Cloud Services and other Technologies)
- White House (Jan. 4, 2022): Committee convened to address vulnerability of Open-Source code, and Log4j attack
This material is based upon work supported by the U.S. Department of Homeland Security under Cooperative Agreement No. 2014-ST-061-ML0001.

MSC@stevens.edu
www.stevens.edu/MSC