BPM

Software Vendor Cybersecurity Concerns

David Trepp, MS/Partner

BPM's Cybersecurity Assessment Services Team

- Providing InfoSec Assessment Services Since 1998
  - Thousands of penetration tests
- *Assessment-Only* Information Security Team
  - Singular expertise
  - Objective assessments
  - Cost-effective remediation recommendations
- BPM cybersecurity assessment team personnel **are not** experts at planning, building, or managing information security controls
- BPM cybersecurity assessment team personnel **are** experts at defeating information security controls and providing controls assurance
  - Provide an ethical hacker's view

David Trepp
Partner, Information Security Assessment

- US Army Veteran
- MS Physical Chemistry
- Serial Tech Entrepreneur
- Personal Interests
  - Rock Climbing
  - Bicycle Touring
  - Information Science
    - Thermodynamics

dtrepp@bpmcpa.com

Contents

- The Problem
- Examples of Software Supply Chain Attacks
- Potential Solutions

Questions/comments are encouraged!

The Problem

Worsening Threat Landscape

Cybercrime/Cyberwarfare:

- Barriers to entry are low
- Risk of capture/prosecution is low-moderate
- Return on investment is high

- Software vendor supply chain related vulnerabilities
  - SolarWinds
- Evolving Social Engineering techniques
  - Headline Opportunism, e.g. pandemic
- Releases of sophisticated, formerly secret hacker's tools into the public domain are rampant
  - Equation Group
  - Hacking Team
- Boundaries are blurred between systems with different responsible parties
- The 'Internet of Things' continues to increase the Internet's attack surface area
- Newly documented vulnerabilities are being released at a dizzying rate

Many Historical Headline Breaches

Occur after the system or application already exists via one, or a combination of, the following techniques:

- Social Engineering
  - Use email, text, chat, phone, snail mail, and/or in-person interactions to get employees to do and/or reveal things they shouldn't
- Credential Compromise
  - Find, Intercept, guess, crack, bypass, spoof, and/or request credentials
- Patch Exploit
  - Exploit vulnerabilities on systems missing critical patches
- Misconfiguration Breach
  - Take advantage of weak configurations, often vendor default configurations
- Boundary Incursion
  - Trespass across interconnected system boundaries, typically from a less-secure system to a more-secure one
- Code Logic Abuse
  - Direct information gathering, session hijacking, scripting, injection, and/or privilege escalation attacks against application logic

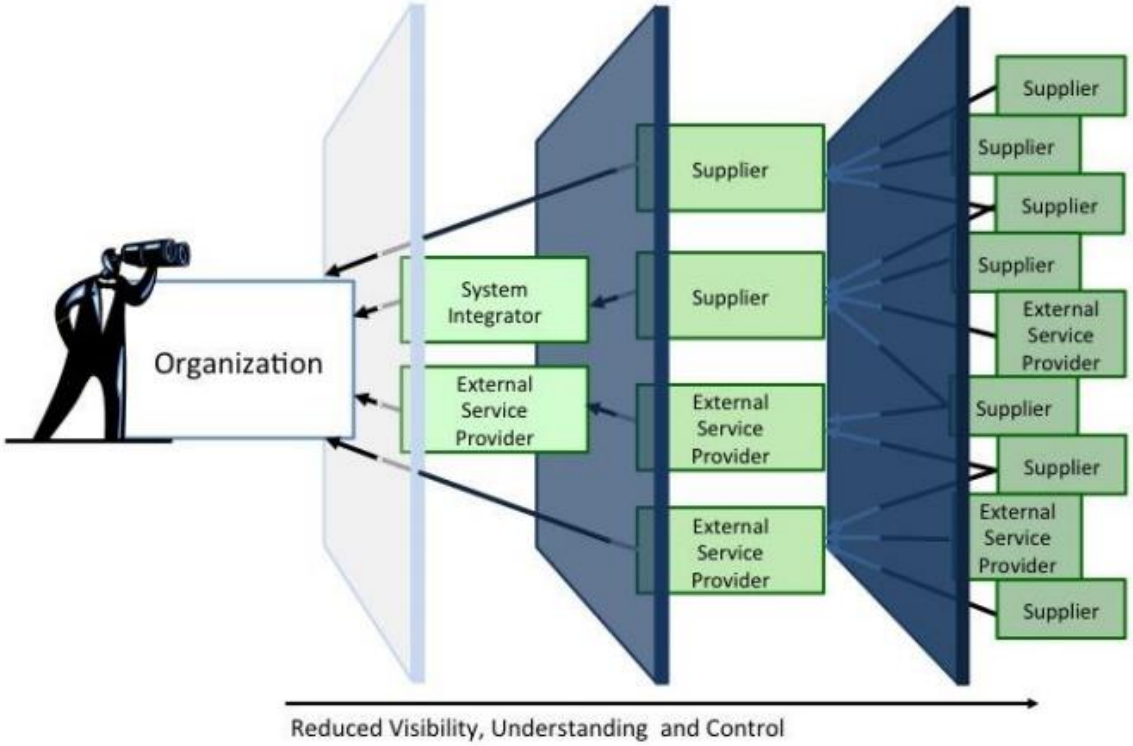
What's So Special About The SolarWinds Breach?

Occurred during the application development phase

- One of the a growing number of compromises that occurred during the development lifecycle
  - The affected code was part of a trusted, digitally signed piece of commercial software
- SolarWinds software is an industry-leading suite of network management applications
  - Operates at an elevated privilege level
  - Estimated to have been deployed to over 18,000 businesses and federal agencies
- The SUNBURST "Trojan" did not interfere with normal application functionality
  - It tested its environment first, to make sure the application was actually deployed on an enterprise network
  - Code strings were purposely obfuscated & communications traffic was designed to mimic expected traffic patterns
  - Backdoor connectivity provided attackers with "hands-on-keyboard" remote access
- SUNBURST likely has been in production since March 2020
  - Consider the impact of, and recovery process for, a complete system compromise that's been ongoing for 9 months!
- There is significant risk that other industry-leading software development firms have suffered similar development cycle compromises
  - We probably only know this Trojan exists today, and many details of how it operates, because the attackers targeted a security research firm
  - There is some evidence suggesting SolarWinds used popular software development tools that may have played a role in the breach

Securing the Unknown

We face an ever-increasing reliance on complex pieces of software that cannot be fully validated



An Organization's Visibility, Understanding, and Control of its ICT Supply Chains

Source: **Supply Chain Risk Management Practices for Federal Information Systems and Organizations**
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-161.pdf

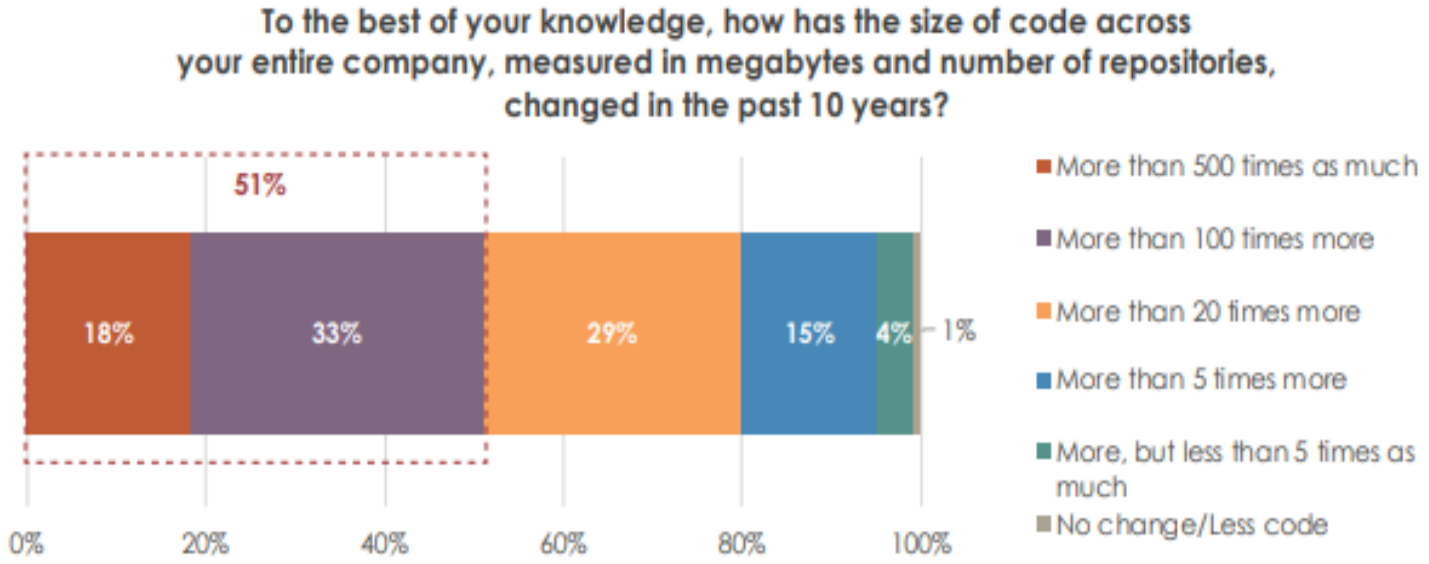
What Are A Typical Application's Boundaries?

Images
Intentionally
Deleted

Diagram & list of a typical online banking application's API, batch, and SSO interfaces

Reliance on Third Parties & Code Size

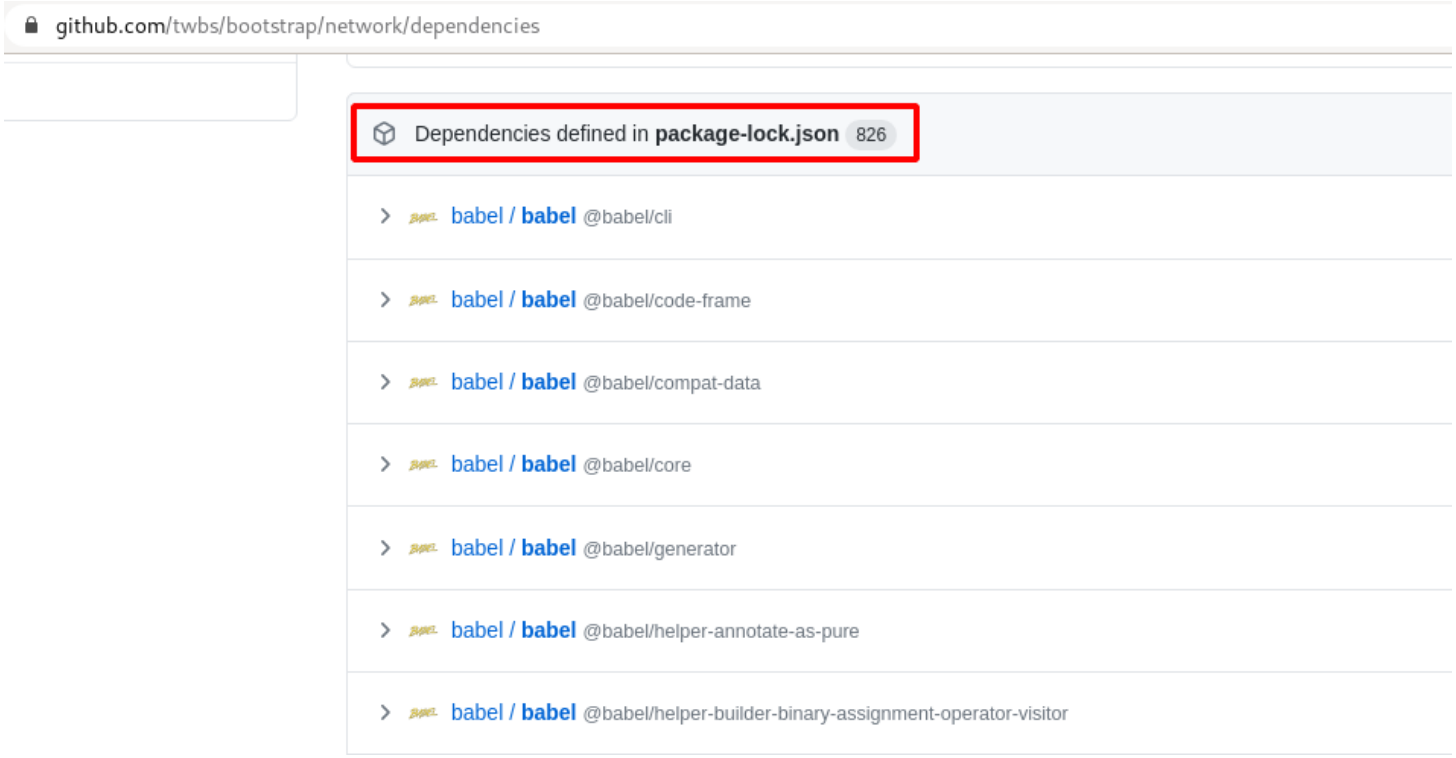
- Most enterprise-grade software is no longer written by single development teams, professional dev. shops incorporate Free and Open-Source Software (FOSS) into their products
  - In 2018 Black Duck Software found that 78% of surveyed companies use some form of open source software[1]
    - 66% of companies surveyed create software that relies on open source code
    - 97% of proprietary apps used some form of open source code



Source: **THE EMERGENCE OF BIG CODE[2]**
https://info.sourcegraph.com/hubfs/CTA%20assets/sourcegraph-big-code-survey-report.pdf

Opensource Dependencies

- Bootstrap is a highly popular web development framework that encompasses HTML, CSS, and JavaScript
- W3Techs reports that nearly 27% of all surveyed websites use Bootstrap
- Bootstrap contains over 800 dependencies![4]

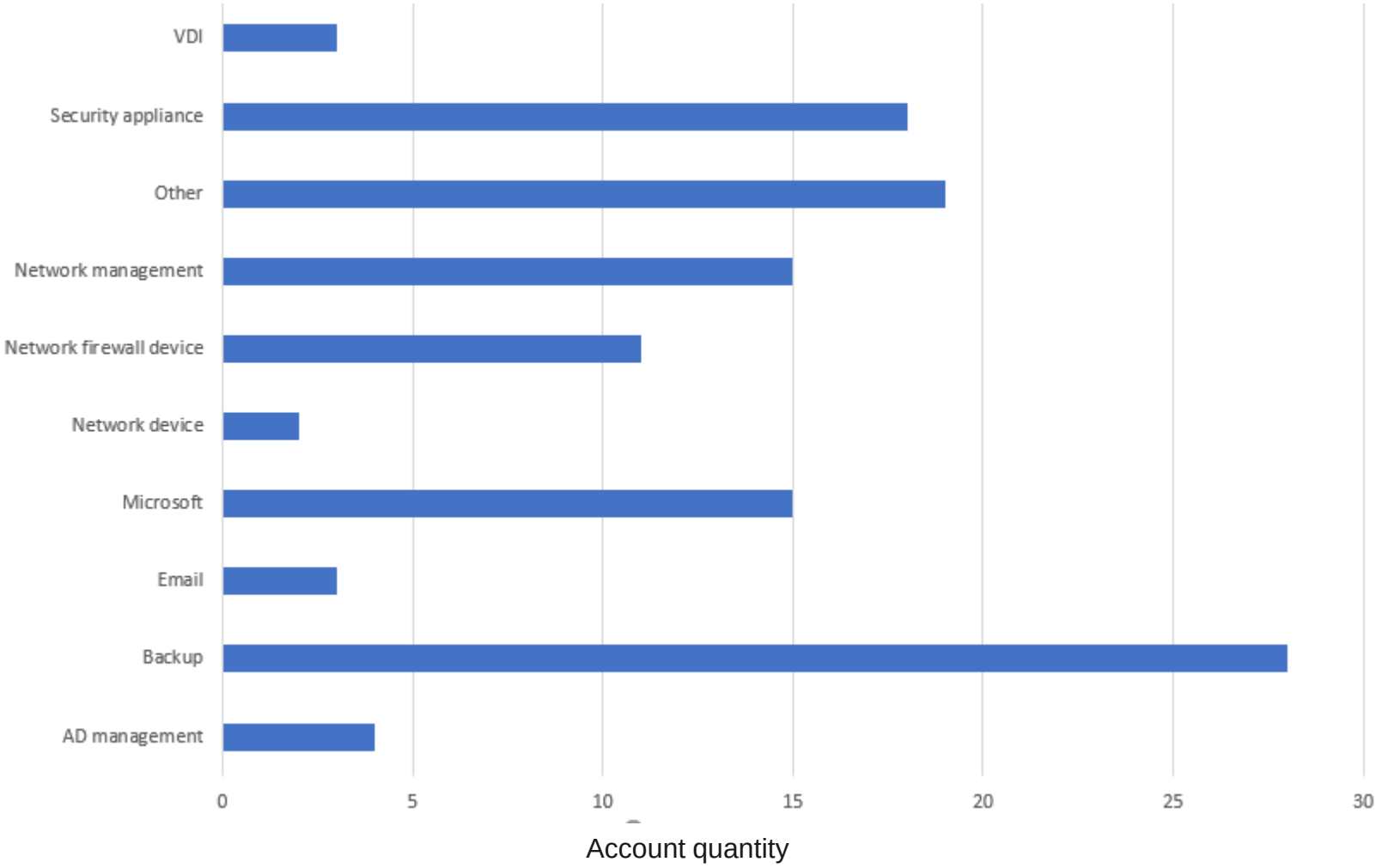


Source: **Bootstrap Dependency Graph**
https://github.com/twbs/bootstrap/network/dependencies

Excessive Permissions

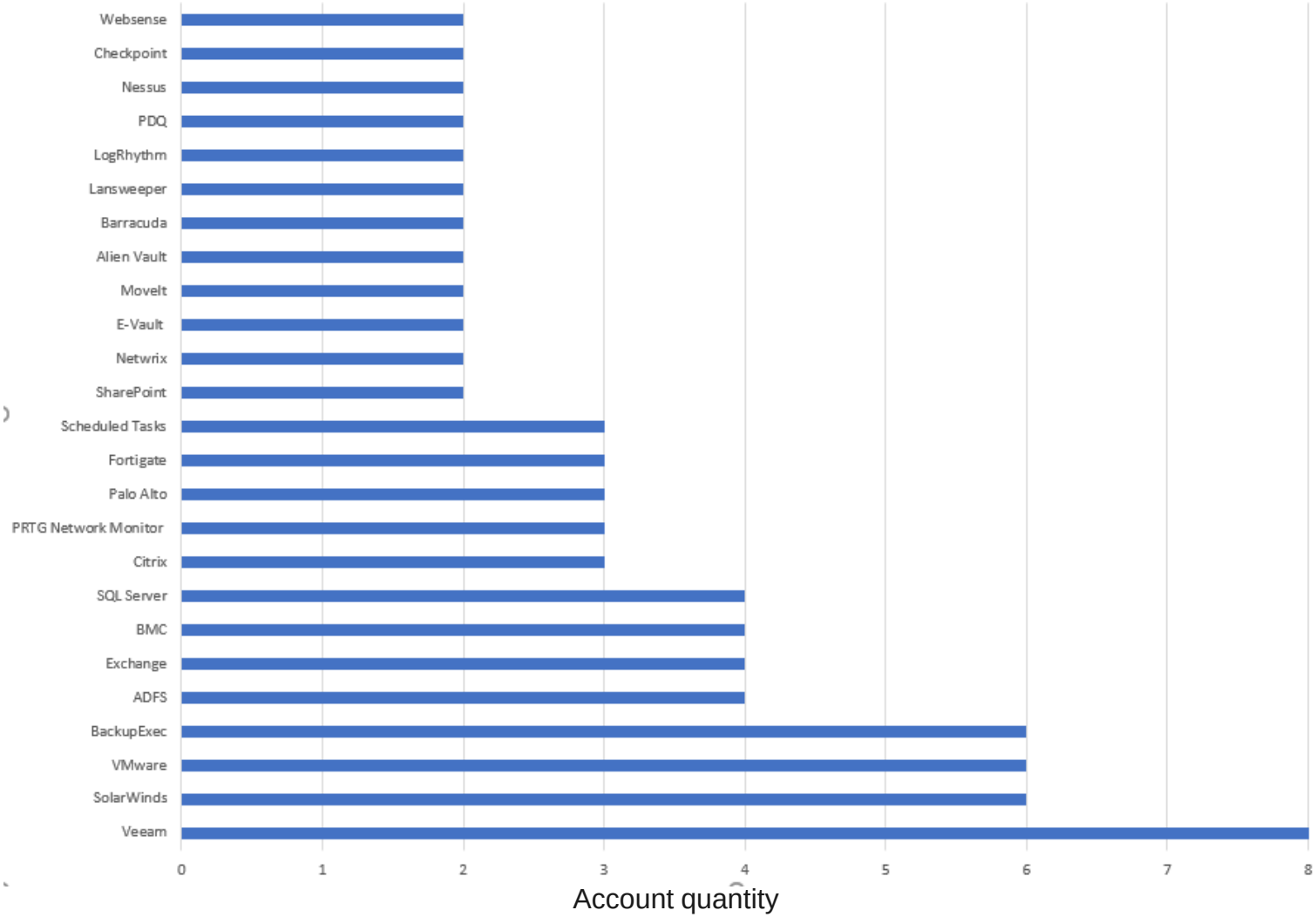
- Sean Metcalf reviewed common enterprise applications and their stated requirements, contrary to common belief "*Rarely does a service account actually require Domain Admin level rights*"[5]
  - Domain user access
  - Operations systems access
  - AD object rights
  - Install permissions on systems
  - System rights
  - AD privileged rights
  - Domain permissions during install
  - Initial start/run permissions
  - Needs full AD rights
- In 2016 Forrester Research estimated 80% of security breaches involved privileged credentials[6]
- "*Restricting administrative privileges is considered one of the top 4 strategies to mitigate targeted cyber intrusions*" – New Zealand Nation Cyber Security Centre[7]
- At the 2017 Black Hat conference, privileged account solutions company Thycotic conducted a survey of more than 250 self-described hackers who revealed that the number one way to get hold of sensitive data is by hacking privileged accounts.[8]
- During 2019 and 2020, out of a sample of 27 internal penetration tests performed, 26 had at least one domain admin account assigned to an application

Most Common Categories with High Privileges



Observations based on a sample of 27 internal networks during penetration test activities

Top Software Observed with Domain Admin



Observations based on a sample of 27 internal networks during penetration test activities

Supply-Chain Attacks in the Current Threat Landscape

- Stakes and motivation are at an all time high
  - With the advances in default security protections and heightened awareness, once easy targets have a hardened exterior
  - Internally however, most organizations remain softer
  - Software vendors are quickly becoming a prime target
- Supply chain attacks take more resources but their motivations remain similar to traditional hacks
  - Financial gain
    - Crypto miners
    - Ransomware
  - Nation-state cybercrime
  - Retaliation
  - Corporate espionage
- Recent high-profile breaches have shown no one is beyond the reach of supply chain attacks

Potential Factors for Targeting Software Vendors

- By compromising a trusted and installed application, hackers are able to walk into a organization that already has whitelisted the compromised application

- High return on investment
  - By hacking a single entry point you can potentially infect thousands of legitimate users/customers

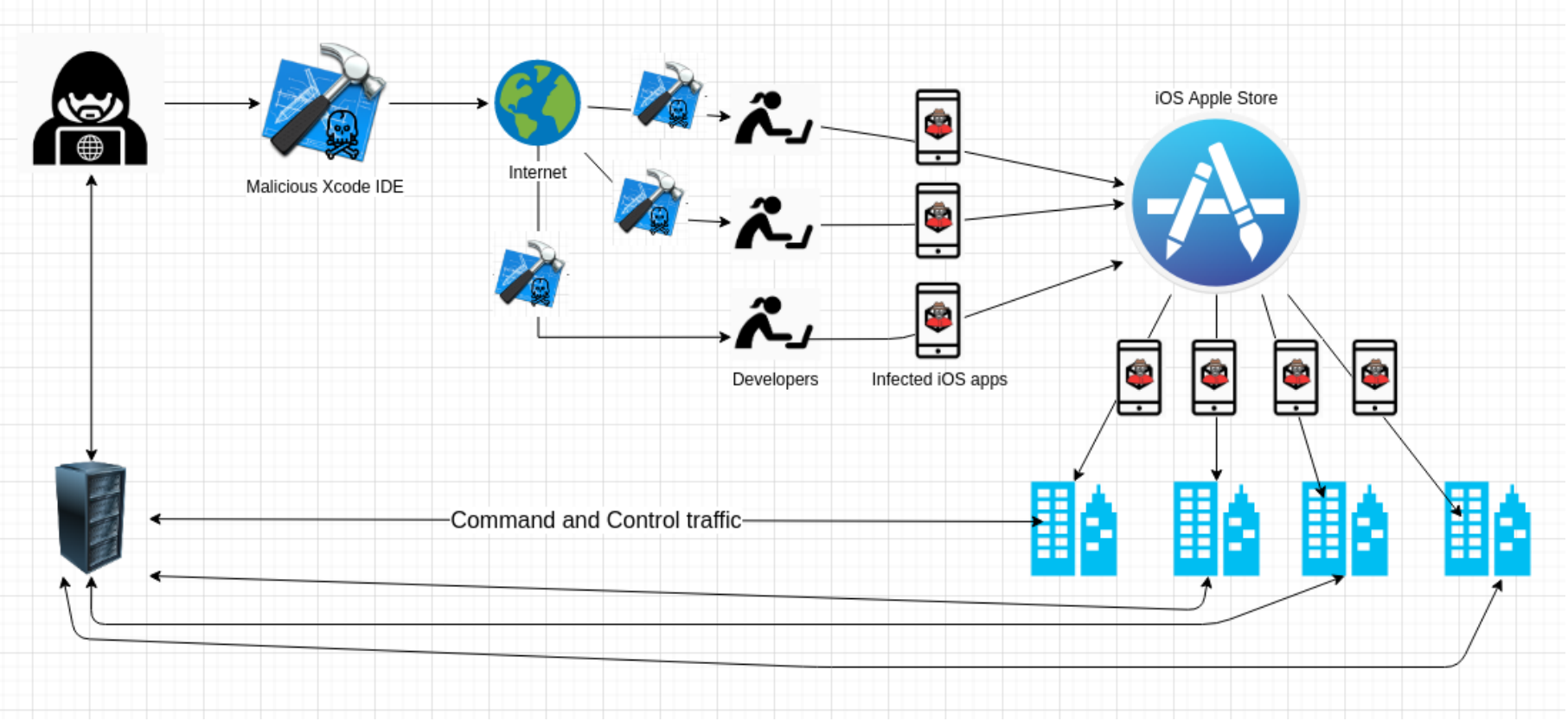
- Conversely, it becomes difficult to target a specific organization, this would be akin to a fisherman dragging a net for hours and not knowing the specifics of what he would catch
  - It is estimated that 18,000 installations of the compromised SolarWinds packages took place, only a very small number of these were desired targets[9]

- Being a victim of your own success
  - If takes time to work a compromised network, without sufficient resources it can quickly become overwhelming

Examples of Software Supply Chain Attacks

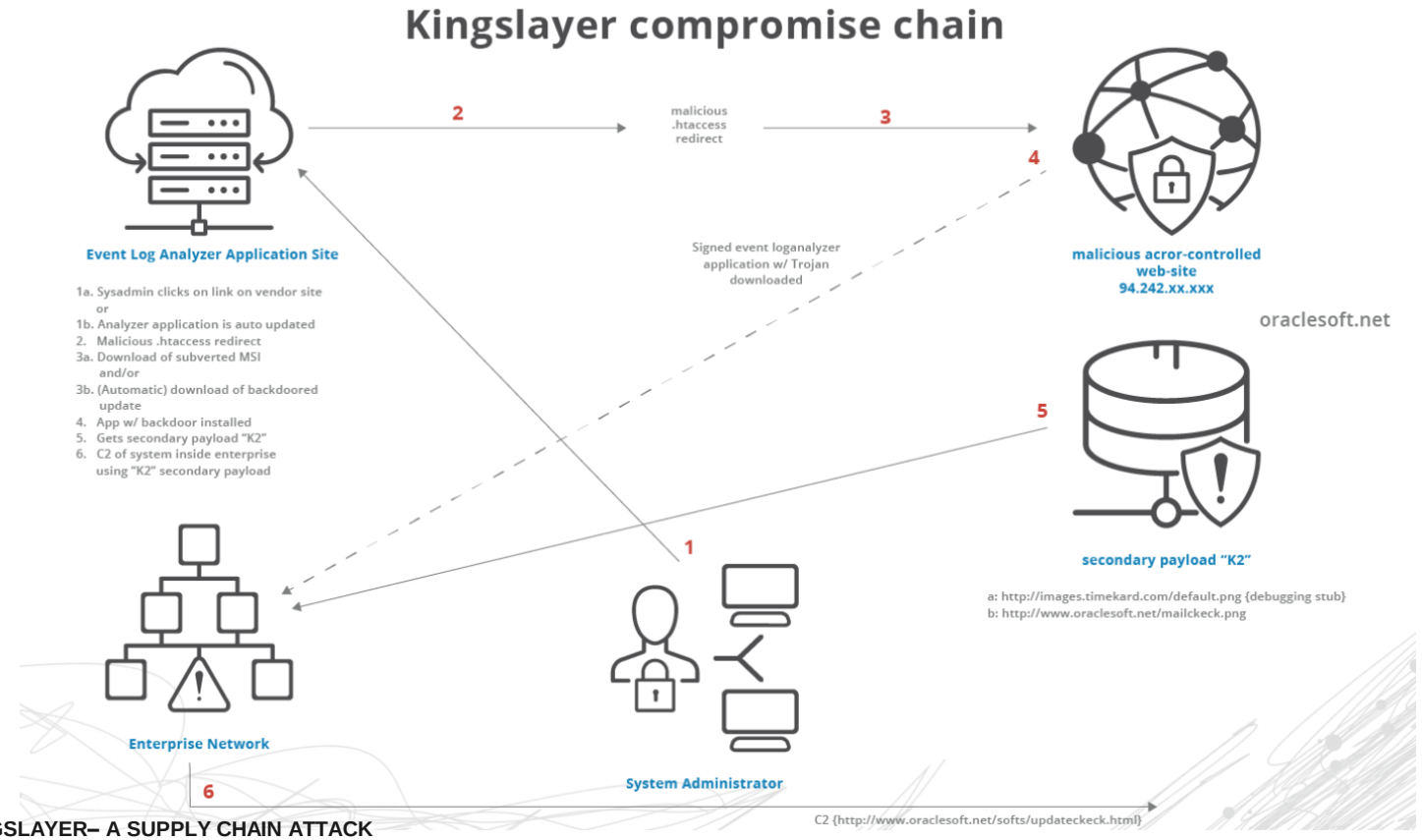
XcodeGhost (2015) Example

- Infected Integrated Development Environment (IDE) shared on popular Mac forums, targeting Chinese developers
- Developers published their apps to the Apple Store, bundled with malicious code
- Users downloaded the infected apps from the legitimate Apple Store[10]
- Users executed the apps from within enterprise networks, building a C&C tunnel[11]



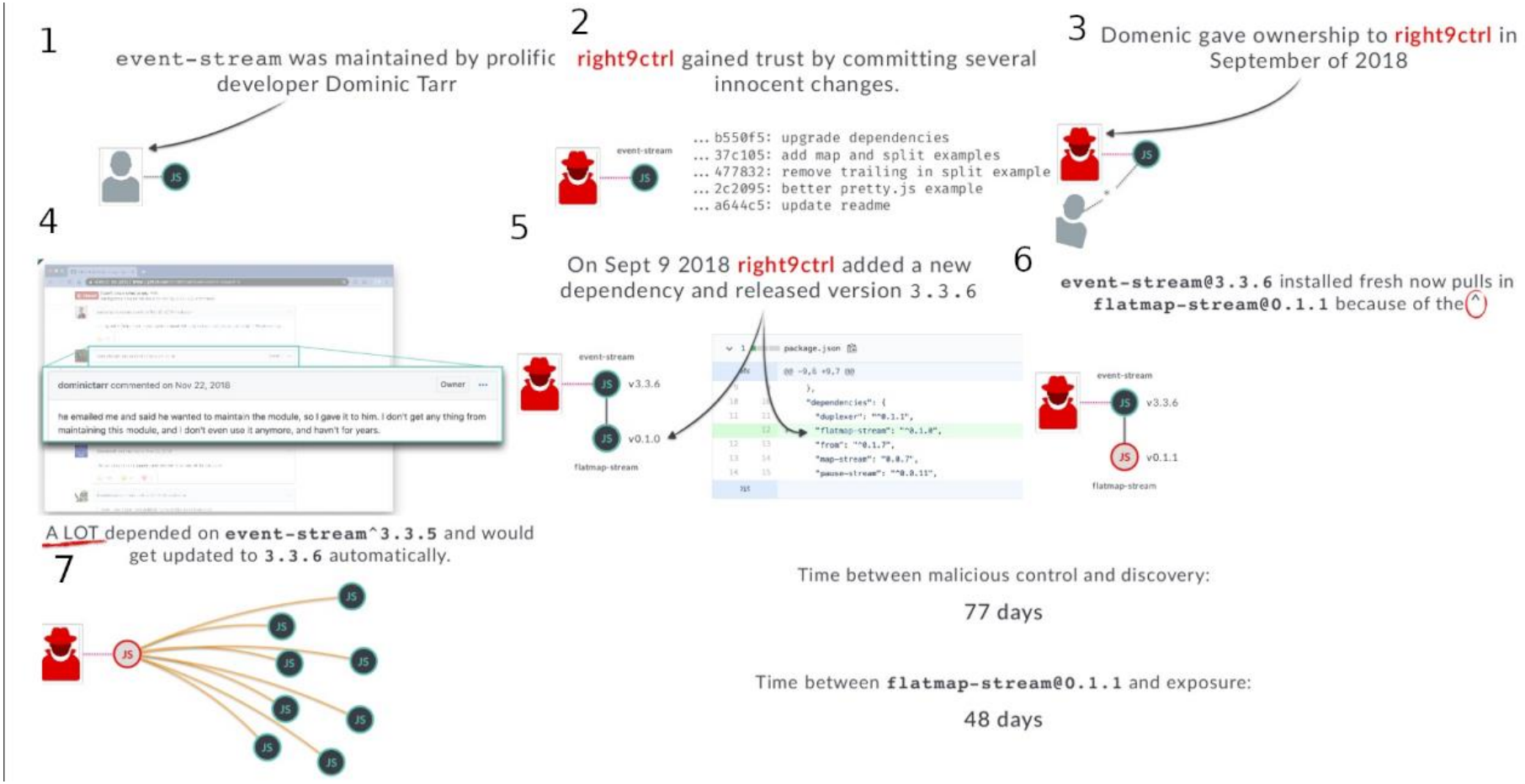
EvLog (2015/2016) Example

- Hackers compromise EvLog developer servers
  - Stole code signing private key
  - Planted malicious .htaccess redirect
- All subsequent new downloads and update requests pointed to infected application
- Backdoored app installed
- C&C tunnel established



Source: **KINGSLAYER– A SUPPLY CHAIN ATTACK**
https://raw.githubusercontent.com/fdiskyou/threat-INTel/master/2017/kingslayer-a-supply-chain-attack.pdf

Event-stream (2018) Example



Source: **Jarrod Overson, Director at Shape Security. Analysis of an OSS supply chain attack - How did 8 millions developers download an exploit with no one noticing?**

https://www.slideshare.net/JarrodOverson/analysis-of-an-oss-supply-chain-attack-how-did-8-millions-developers-download-an-exploit-with-no-one-noticing

SolarWinds(2019) Example

SUPPLY CHAIN ATTACK
Attackers insert malicious code into a DLL component of legitimate software. The compromised DLL is distributed to organizations that use the related software.

EXECUTION, PERSISTENCE
When the software starts, the compromised DLL loads, and the inserted malicious code calls the function that contains the backdoor capabilities.

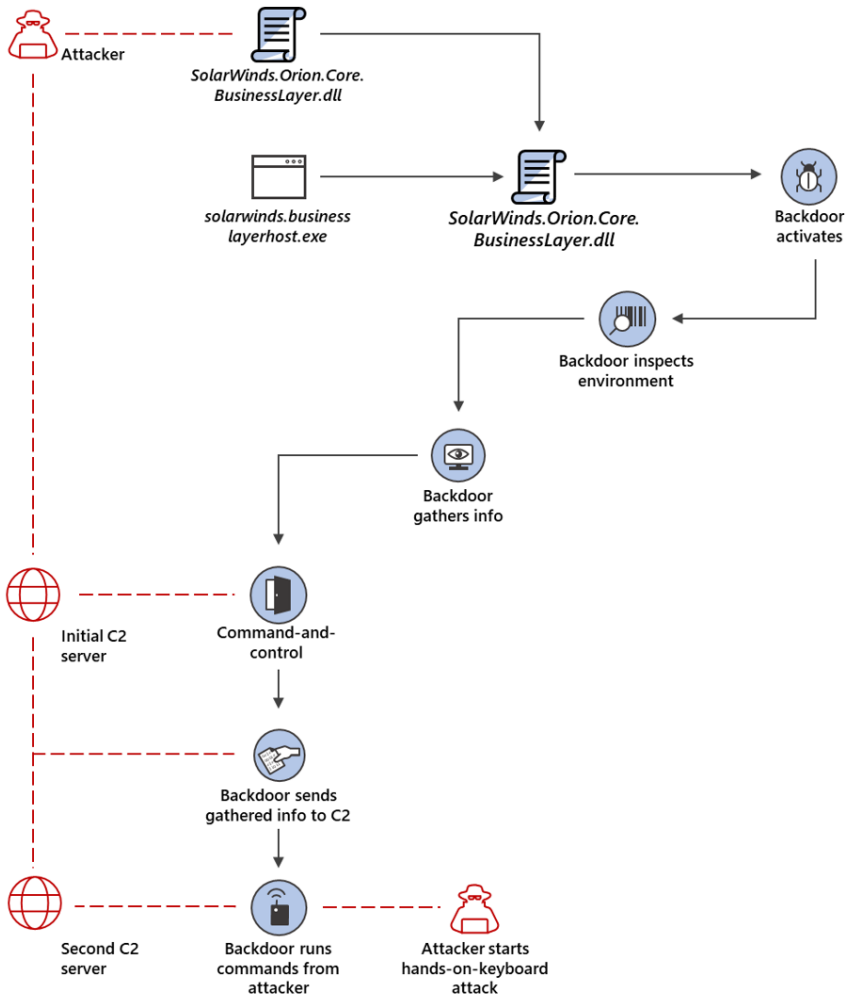
DEFENSE EVASION
The backdoor has a lengthy list of checks to make sure it's running in an actual compromised network.

RECON
The backdoor gathers system info

INITIAL C2
The backdoor connects to a command-and-control server. The domain it connects to is partly based on info gathered from system, making each subdomain unique. The backdoor may receive an additional C2 address to connect to.

EXFILTRATION
The backdoor sends gathered information to the attacker.

HANDS-ON-KEYBOARD ATTACK
The backdoor runs commands it receives from attackers. The wide range of backdoor capabilities allow attackers to perform additional activities, such as credential theft, progressive privilege escalation, and lateral movement.



Source: **Analyzing Solorigate, the compromised DLL file that started a sophisticated cyberattack, and how Microsoft Defender helps protect customers**
https://www.microsoft.com/security/blog/2020/12/18/analyzing-solorigate-the-compromised-dll-file-that-started-a-sophisticated-cyberattack-and-how-microsoft-defender-helps-protect/

Application Penetration Test Reveals 3rd Party Vulnerability

Images Intentionally Deleted

Bank statements are loaded for authenticated user Daisy Duck

A search is initiated

After sending the POST, the server responds with an iframe for an external service

A second user, Donald Duck, logs in and initiates a search

Donald Duck's account value is replaced with Daisy Duck's account value

Daisy Duck's account statements are now viewed from the Donald Duck account

Common Attributes

- Targeting individual developers
- Abusing application dependencies
- Use of less secure software companies as a “springboard” into large organizations
- Theft of code signing keys
- Compromised software distributed via the update channel
- Supply chain compromise goes undetected for months, if not years
- Diverse motivations

Common Questions

- Is there anything I can do?
- How can I perform any kind of vetting on complex software?
- Why would my organization be targeted?
  - We are too small
  - We are too remote
  - What sensitive information do I hold?
- Is what I'm currently doing good enough?
- If my organization is using compromised software, what do we do next?
  - Can we ever trust our current environment again?
- How am I supposed to prioritize the hypothetical when we're facing real problems at hand?

We do not have all the answers!

Potential Solutions

NIST as Reference Framework

- Key NIST Reference Documents:
  - SP 800-161 “Supply Chain Risk Management Practices for Federal Information Systems and Organizations”
  - SP 800-53 Rev 4: SA-12 “SUPPLY CHAIN PROTECTION”
  - SP 800-53 Rev 5: 3.20 “SUPPLY CHAIN RISK MANAGEMENT”

- NIST SP 800-161 states cyber supply chain risks may include the following
  - Insertion of counterfeits
  - Unauthorized production
  - *Tampering, theft*
  - *Insertion of malicious software and hardware*
  - *Poor manufacturing and development practices in the cyber supply chain*
- The examples we just saw and the potential solutions we’re about to discuss focus on the bottom three bullet points

As Developers

- Be rigorous in vetting 3rd party libraries before including them in your projects
  - Once vetted, pull down the latest build and don't update without further vetting
- Become involved in the Open Source community and help with code review
- Become aware of security best practices, familiarize all team members with the OWASP* project and their guidelines
- Resist the urge to add security on after the software is already built, security must be built into the software's design
- Remember that building strong security is not a point in time exercise, it must be continual and integral into the software lifecycle
  - Team training
  - Secure software design
  - Code review
  - Static code analysis
  - Penetration testing

*Open Source Web Application project https://owasp.org/

As System Administrators

- Limit the number of products and installed software
- Adopt a mentally of assuming you will be breached, implement defense in layers
  - Zero trust model of security
- Restrict application permissions and follow least-privilege
  - Don’t hand out DA rights just because a product requests it
- Perform routine log review and establish a baseline of outbound traffic
- Move beyond signature-based IDS & anti-malware solutions

As Security Professionals

- Build organizational policy modeled after published frameworks
  - NIST
    - SP 800-161 "*Supply Chain Risk Management Practices for Federal Information Systems and Organizations"*
    - SP 800-53 Rev 4: SA-12 "*SUPPLY CHAIN PROTECTION*"
    - SP 800-53 Rev 5: 3.20 *"SUPPLY CHAIN RISK MANAGEMENT"*
  - MITRE https://www.mitre.org/sites/default/files/publications/supply-chain-attack-framework-14-0228.pdf
- Trust nothing and question everything
  - Stay abreast of latest incidents
- Subscribe to US-CERT notifications and security lists for all products you are responsible for
  - Be prepared to undergo incident response
  - Help decision makers embrace security by finding solutions, not just problems

As Leaders

- Build a culture of security awareness and support
  - Lead by example
  - Allocate adequate resources
- Don’t overrule your security team’s objections in the interest of fast-tracking a project
  - Support the global security community through collaboration and information exchange
- Be transparent in the event of a breach
  - We’re all in this together
- Perform thorough vendor due diligence…

Evaluating Potential Vendors

- In order to reduce potential risks from 3rd parties, it's imperative that proper vendor due diligence is performed. One such way to perform this due diligence is to present each potential vendor with a questionnaire that evaluates how they treat security.

- Ultimately, the manner in which the vendor responds, or fails to respond, to questionnaires can help you answer the most important question:

- Remember, the only time you have leverage in a vendor relationship is during pre-purchase negotiations
  - So before you sign the purchase agreement, ask yourself and your team:

Does the Vendor Exhibit a Culture of Cybersecurity?

Vendor Due Diligence

- Disclosure of ***All*** Development Supply Chain Risks
  - What portions, if any, of the design and development process was/is outsourced?
    - What controls does the vendor have in place to manage their third-party outsourcing risk?
    - What controls are in place to secure access to source code repositories?
    - Have any elements of the code base been re-used from code-sharing resources?
- Disclosure of ***All*** Support and Patch/Update Requirements
  - How is remote access for support handled?
  - How is patching/updating/change management handled?
  - What ports are listening for remote support & patching?
    - Can handshake attempts be restricted?
      - By source IP address, certificate, MAC address, etc.
- Disclosure of ***All*** Communications Protocols
  - What type (by protocol) and volume of internal traffic is expected?
  - What inbound/outbound ports will be used?
    - Can communications be restricted by source address, certificate, MAC address, etc.?
    - How much traffic is expected over these ports?
- Disclosure of ***All*** Required Accounts
  - What are all the accounts, e.g. user, admin, supervisor, etc.?
  - Are there any undocumented accounts?
  - What are the privilege levels for all the different accounts?

Vendor Due Diligence, cont.

- Disclosure of ***All*** Encryption Controls
  - Are all data encrypted in transit?
    - Any unsigned or misconfigured certs?
    - Any weak ciphers or hashing algorithms?
  - Are all sensitive data encrypted at rest, considering Hypervisor, Container, OS, DB, & File levels?
    - Database data?
    - Backup files?
    - Credentials in process memory, e.g. RAM?
    - Windows registry, e.g LSA Secrets?
    - Session keys?
    - Are passwords hard-coded into the application?
    - Configuration files, e.g. .config, .ini, etc.?
    - Log files?
- Disclosure of ***All*** Integrity Controls
  - Is SMB signing supported?
  - Is LDAP signing supported?
  - Does the development process implement code signing?
- Permission to Include the Vendor's System in the Organization's Testing Regimen?
  - Or demand evidence of the vendor's ongoing test regimen

Summary

Key Takeaways

- Applications are growing in size and complexity at an impressive rate, a key factor in this growth is the ability to incorporate external code dependencies in the interest of time and not reinventing the wheel
- The threat landscape is daunting, contrary to popular belief, cyber supply chain attacks are not new and it's not going to get easier
- The SolarWinds breach has shown just how devastating a well-funded attack is, we are still a long way away from understanding the full reach
- Given what we know, and what we've seen in recent months, it is unlikely any of us here today have the resources or the placement within a supply chain to prevent a compromise
  - The only reason a top cyber security firm knew they had been breached was due to a mistake made by the SolarWinds hackers
- What we can do, however, is prepare for the next inevitable breach and harden our networks
  - Follow best security practices
  - Enforce least privilege
  - Perform rigorous vendor due diligence
  - Train users to be suspicious and report all suspicious events

References

1. Vaughan-Nichols, Steven. “It’s an Open-Source World: 78 Percent of Companies Run Open-Source Software.” ZDNet, 16 Apr. 2015, www.zdnet.com/article/its-an-open-source-world-78-percent-of-companies-run-open-source-software.
2. Dimensional Research. “THE EMERGENCE OF BIG CODE.” Sourcegraph, 2020, info.sourcegraph.com/hubfs/CTA%20assets/sourcegraph-big-code-survey-report.pdf.
3. “State of the Software Supply Chain.” Sonatype, 2019, www.sonatype.com/hubfs/SSC/2019%20SSC/SON_SSSC-Report-2019_jun16-DRAFT.pdf.
4. “Usage Statistics and Market Share of Bootstrap for Websites, February 2021.” W3Techs, 2021, w3techs.com/technologies/details/js-bootstrap.
5. Metcalf, Sean. “There’s Something About Service Accounts.” Active Directory Security, 21 Mar. 2019, adsecurity.org/?p=4115.
6. “What Is Least Privilege & Why Do You Need It?” BeyondTrust, 19 Feb. 2021, www.beyondtrust.com/blog/entry/what-is-least-privilege.
7. “Restricting Admin Privileges Explained.” Nation Cyber Security Centre, www.ncsc.govt.nz/assets/NCSC-Documents/NCSC-Restricting-Admin-Priviledges-Explained.pdf. Accessed 21 Feb. 2021.
8. “Principle of Least Privilege (POLP): What, Why & Best Practices.” Webdevolutions.Blob.Core.Windows.Net, webdevolutions.blob.core.windows.net/blog/pdf/principle-of-least-privilege-polp-what-why-best-practices.pdf. Accessed 22 Feb. 2021.
9. “Joint Statement by the Federal Bureau of Investigation (FBI), the Cybersecurity and Infrastructure Security Agency (CISA), the Office of the Director of National Intelligence (ODNI), and the National Security Agency (NSA) | CISA.” CISA, 5 Jan. 2021, www.cisa.gov/news/2021/01/05/joint-statement-federal-bureau-investigation-fbi-cybersecurity-and-infrastructure.
10. Team, FireEye Mobile. “Protecting Our Customers from XcodeGhost.” FireEye, 22 Sept. 2015, www.fireeye.com/blog/executive-perspective/2015/09/protecting_our_custo.html.
11. Kang, Yong. “XcodeGhost S: A New Breed Hits the US.” FireEye, 3 Nov. 2015, www.fireeye.com/blog/threat-research/2015/11/xcodeghost_s_a_new.html.

Questions/Comments?

dtrepp@bpmcpa.com

Thank You!