
VERACODE
You change the world, we'll secure it.
.NET CORE
SUMMER EVENT
Niels Tanis
The Rise of Software Supply-Chain Attacks
How Secure is your .NET Application?

Who am I?

- Niels Tanis
  - Security Researcher @ Veracode
  - Background in .NET Development
  - Application Security Consultancy
  - Pen-testing & Ethical Hacking
  - ISC² CSSLP

@nielstanis

Picture is from Veracode report/site:

https://www.veracode.com/sites/default/files/pdf/resources/ipapers/everything-you-need-to-know-open-source-risk/index.html

Agenda

- Hacker History
- Definition Software Supply-Chain
  - Development of .NET application
  - Building / Releasing / Deploying
- Securing our Software Supply-Chain
- Conclusion and Q&A

@nielstanis

Hacking History

- Started out with phreaking in late ‘50-’60.

@nielstanis

https://www.wikiwand.com/en/Blue_box
https://en.wikipedia.org/wiki/Kevin_Mitnick
https://www.youtube.com/watch?v=8s4b2ZKyPHc

Getting connected!

- SATAN (**S**ecurity **A**dministrator **T**ool for **A**nalyzing **N**etworks) by Wietse Venema and Dan Farmer released 1995.
- NMAP (Network Mapper) by Gordon Lyon released in 1997

```
Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-14 11:05 CET
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.16s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 996 closed ports
PORT     STATE    SERVICE
22/tcp   open     ssh
25/tcp   filtered smtp
80/tcp   open     http
9929/tcp open     nping-echo
```

@nielstanis

https://en.wikipedia.org/wiki/Security_Administrator_Tool_for_Analyzing_Networks
https://nmap.org

Smashing the Stack…

- Phrack #49 in November 1996

  Aleph One wrote about **buffer overflows**

.oO Phrack 49 Oo.

Volume Seven, Issue Forty-Nine

File 14 of 16

BugTraq, r00t, and Underground.Org
bring you

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Smashing The Stack For Fun And Profit
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

by Aleph One
aleph1@underground.org

`smash the stack` [C programming] n. On many C implementations it is possible to corrupt the execution stack by writing past the end of an array declared auto in a routine. Code that does this is said to smash the stack, and can cause return from the routine to jump to a random address. This can produce some of the most insidious data-dependent bugs known to mankind. Variants include trash the stack, scribble the stack, mangle the stack; the term mung the stack is not used, as this is never done intentionally. See spam; see also alias bug, fandango on core, memory leak, precedence lossage, overrun screw.

@nielstanis

http://phrack.org/issues/49/14.html#article

SQL Injection

- Phrack #54 in December 1998

  Rain Forest Puppy wrote about

  SQL injection

```
----[  ODBC and MS SQL server 6.5

Ok, topic change again.  Since we've hit on web service and database stuff,
let's roll with it.  Onto ODBC and MS SQL server 6.5.

I worked with a fellow WT'er on this problem.  He did the good thing and told
Microsoft, and their answer was, well, hilarious.  According to them,
what you're about to read is not a problem, so don't worry about doing
anything to stop it.

- WHAT'S THE PROBLEM?  MS SQL server allows batch commands.

- WHAT'S THAT MEAN?  I can do something like:

        SELECT * FROM table WHERE x=1 SELECT * FROM table WHERE y=5

Exactly like that, and it'll work.  It will return two record sets, with each
set containing the results of the individual SELECT.

- WHAT'S THAT REALLY MEAN?  People can possibly piggyback SQL commands into
your statements.  Let's say you have:

        SELECT * FROM table WHERE x=%%criteria from webpage user%%

Now, what if %%criteria from webpage user%% was equal to:

        SELECT * FROM sysobjects

It would translate to:

        SELECT * FROM table WHERE x=1 SELECT * FROM sysobjects
```

@nielstanis

http://www.phrack.org/issues/54/8.html

Code Red & SQL Slammer

- Microsoft Internet Information Server, July 2001

```
GET /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNN
%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801
%u9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3
%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a HTTP/1.0
```

@nielstanis

https://en.wikipedia.org/wiki/Code_Red_(computer_worm)
https://en.wikipedia.org/wiki/SQL_Slammer

Bill Gates – Email to all MS FTE

BILL GATES BUSINESS 01.17.02 12:00 PM

Bill Gates: Trustworthy Computing

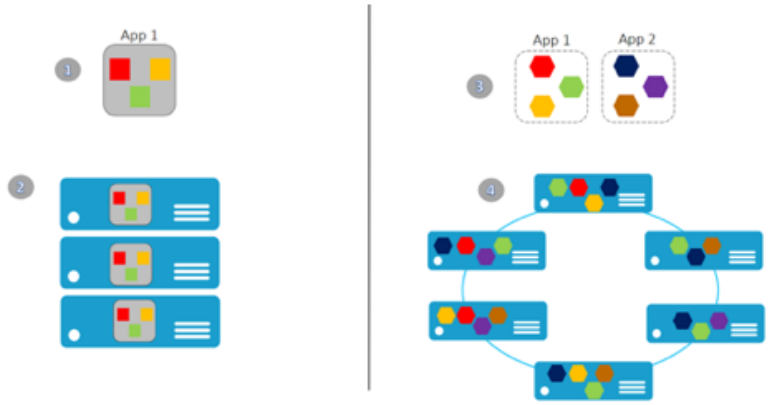
*This is the e-mail Bill Gates sent to every full-time employee at Microsoft, in which he describes the company's new strategy emphasizing security in its products.*From: Bill Gates
Sent: Tuesday, January 15, 2002 5:22 PM
To: Microsoft and Subsidiaries: All FTE
Subject: Trustworthy computing

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

https://www.wired.com/2002/01/bill-gates-trustworthy-computing/

Changes in Software Architecture

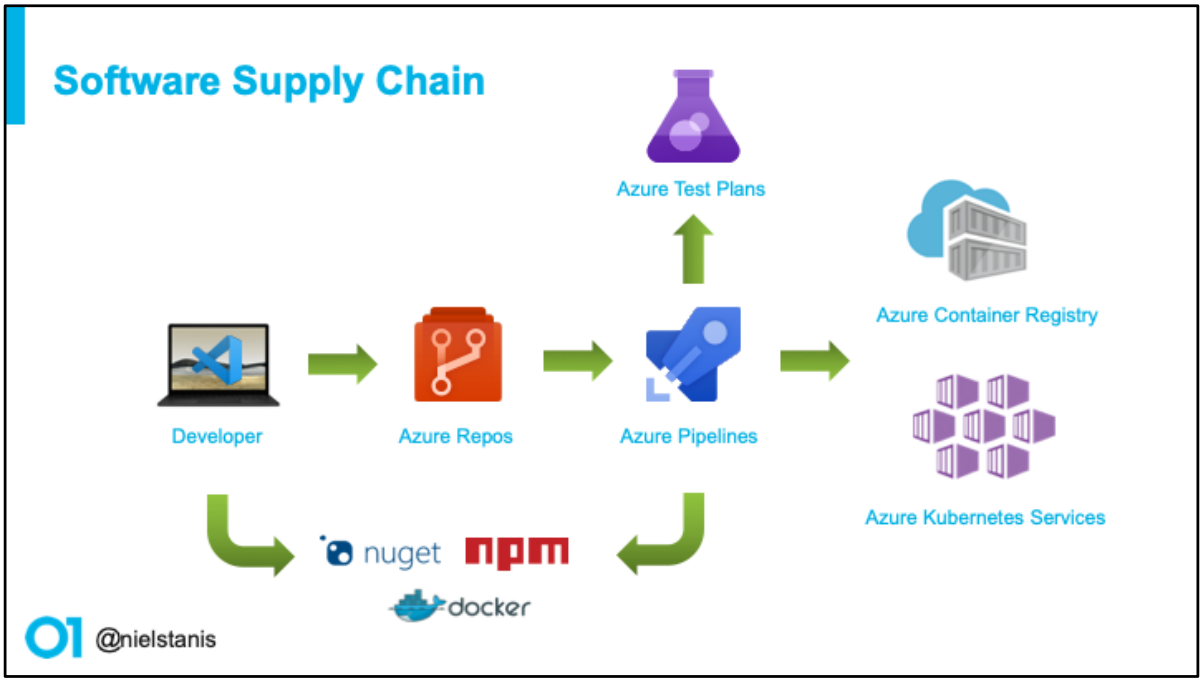
- Monolith → Microservices → Serverless

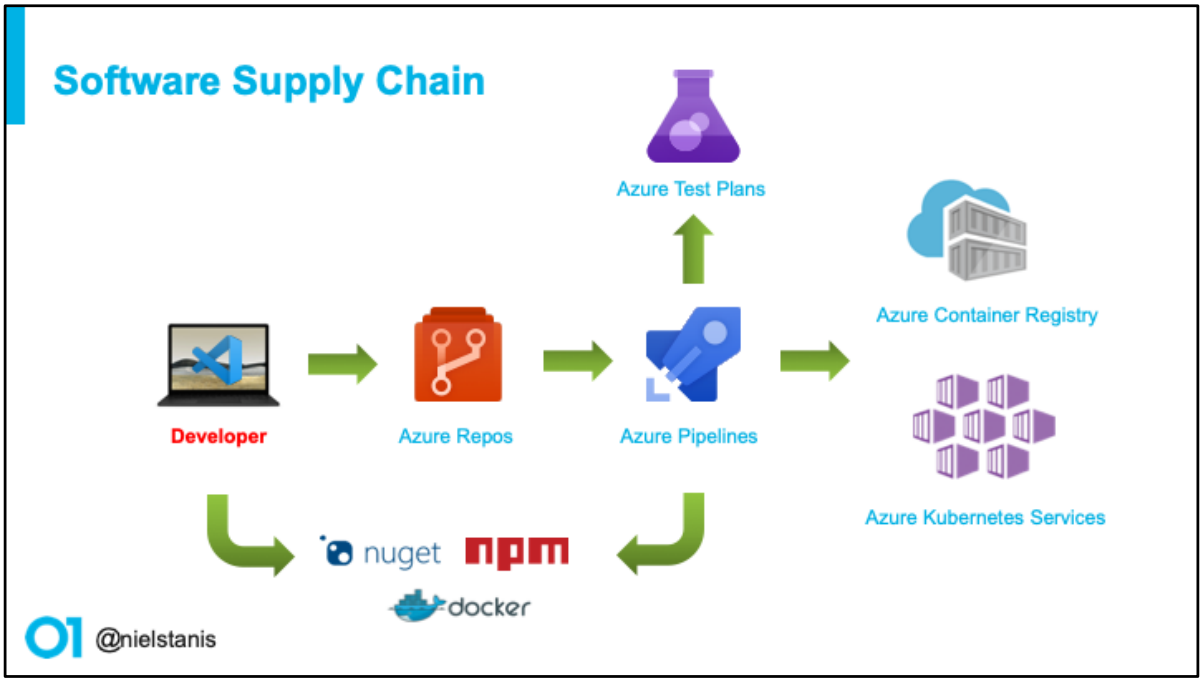


@nielstanis

What's a Supply-Chain?

@nielstanis


Software Supply Chain
Azure Test Plans
Azure Container Registry
Developer
Azure Repos
Azure Pipelines
Azure Kubernetes Services
nuget
npm
docker
@nielstanis

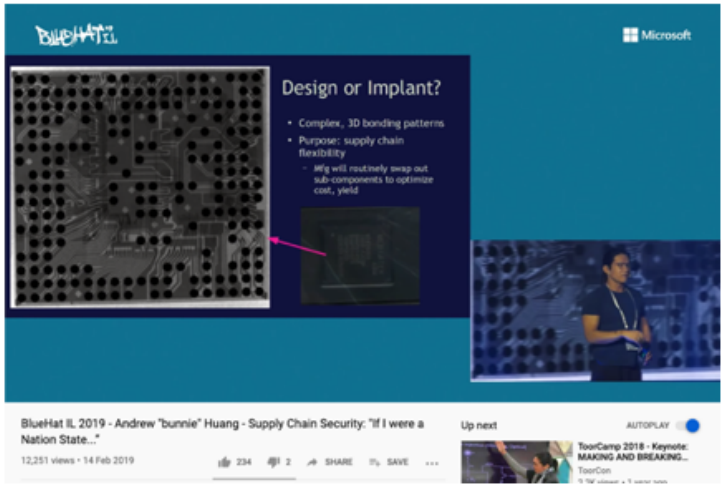

Software Supply Chain
Azure Test Plans
Azure Container Registry
Developer
Azure Repos
Azure Pipelines
Azure Kubernetes Services
nuget
npm
docker
@nielstanis

Development Machine

- Secure Boot & Trusted Platform Module (TPM)
- Encrypt disk, harden operating system install updates
- But can you trust the hardware?

@nielstanis

Hacking Hardware



@nielstanis

https://www.youtube.com/watch?v=RqQhWitJ1As

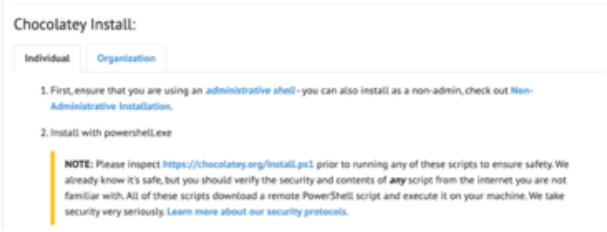
Development Machine

- Installs on machine – HomeBrew on Mac
  - `curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh`

@nielstanis

Development Machine

- Installs on machine – Chocolately on Windows



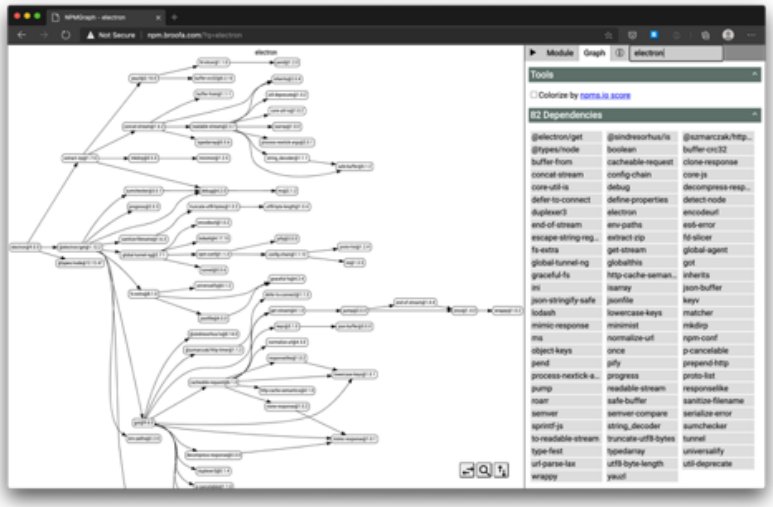
@nielstanis

Octopus Scanner - NetBeans

@nielstanis

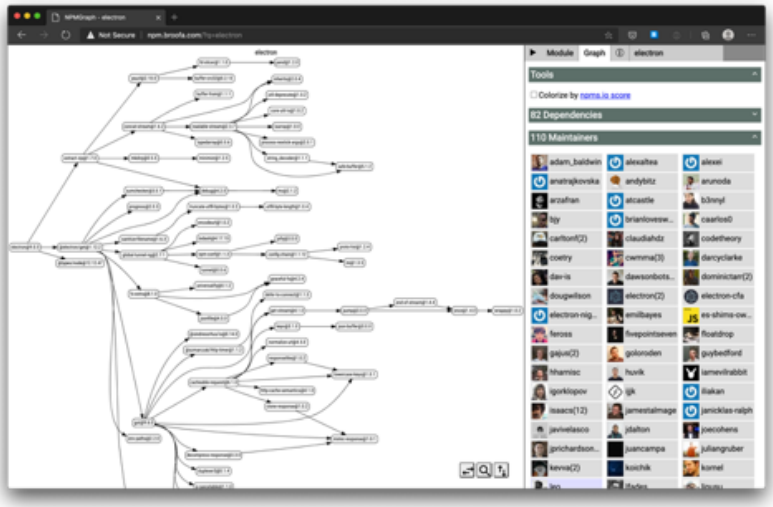
https://securitylab.github.com/research/octopus-scanner-malware-open-source-supply-chain

Visual Studio Code



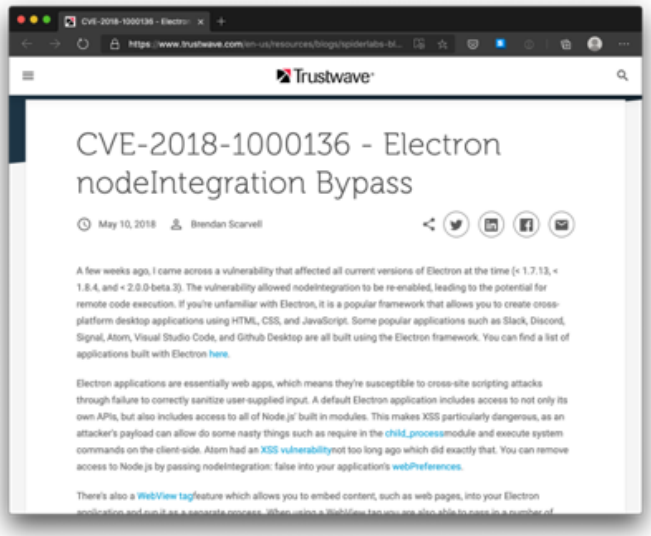
@nielstanis

Visual Studio Code



@nielstanis

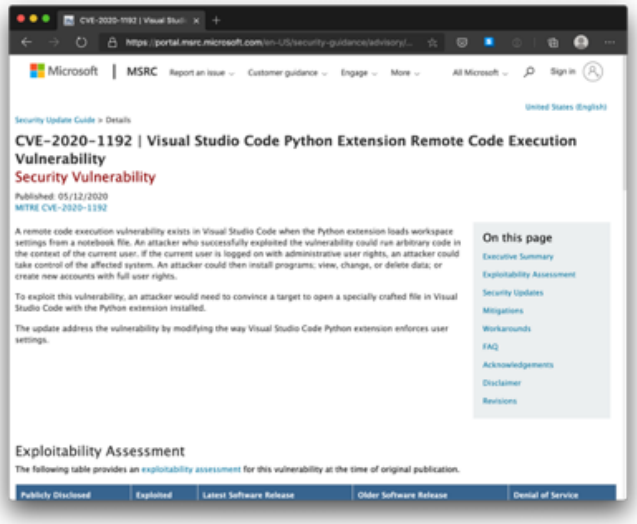
Visual Studio Code



@nielstanis

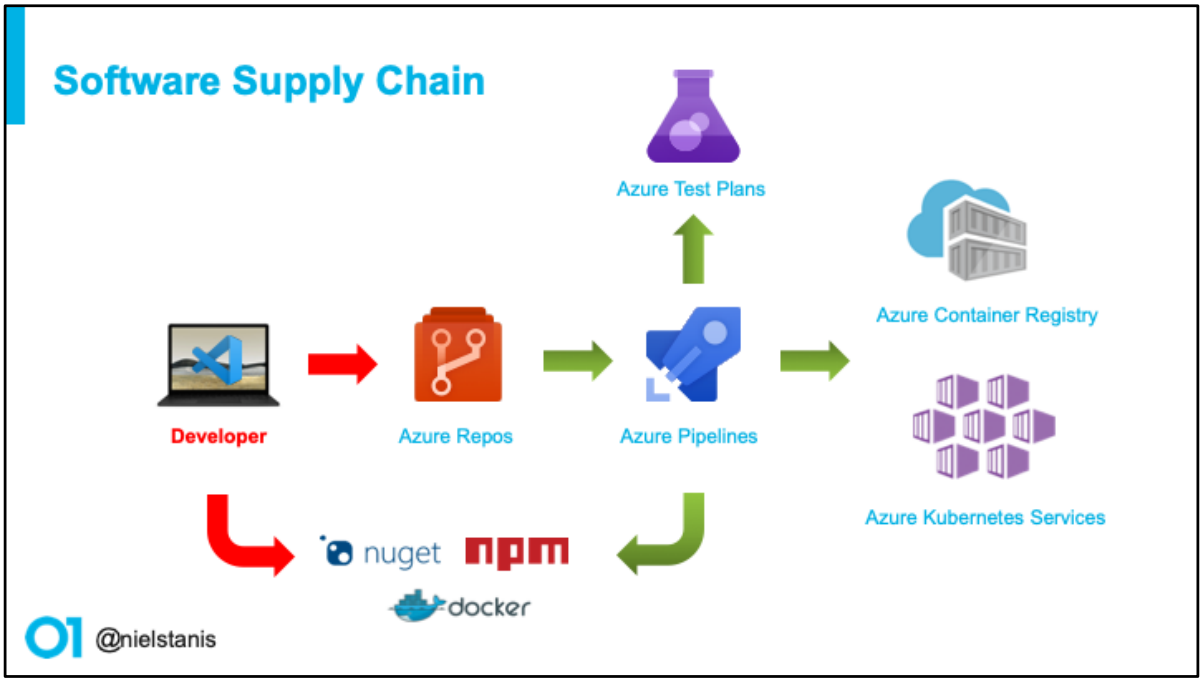
https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/cve-2018-1000136-electron-nodeintegration-bypass/

Visual Studio Code



@nielstanis

https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2020-1192


Software Supply Chain
Azure Test Plans
Azure Container Registry
Developer
Azure Repos
Azure Pipelines
Azure Kubernetes Services
nuget
npm
docker
@nielstanis

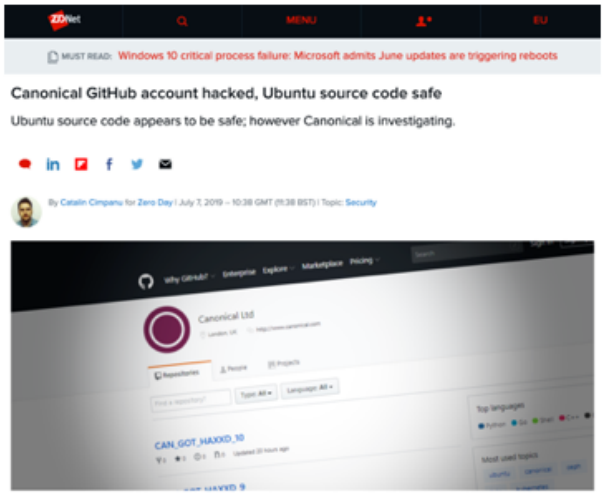
Development Machine

- Package manager e.g. NuGet / NPM
- Transport-Layer Security (TLS)
  - Root Authority Trust
  - Downgrade, TLS 1.0 – 1.1 deprecated on NuGet
- Domain Name Service (DNS)
  - DNSSEC → NuGet.org and GitHub.com don't support it

@nielstanis

https://www.ssllabs.com/downloads/SSL_Threat_Model.png
https://devblogs.microsoft.com/nuget/deprecating-tls-1-0-and-1-1-on-nuget-org/
https://dnssec-analyzer.verisignlabs.com/nuget.org

Canonical GitHub Account



@nielstanis

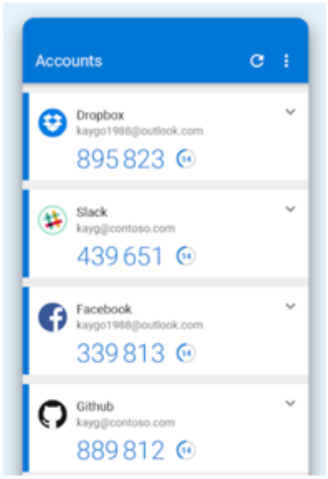
https://www.zdnet.com/article/canonical-github-account-hacked-ubuntu-source-code-safe/

Microsoft GitHub Account

@nielstanis

https://threatpost.com/report-microsofts-github-account-gets-hacked/155587/

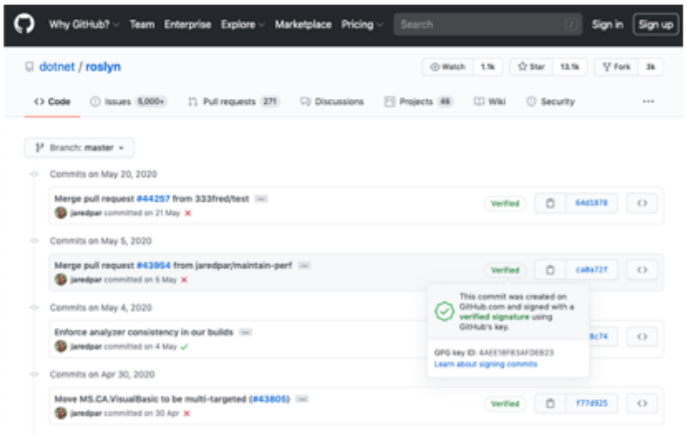
Use MFA on source-repository



@nielstanis

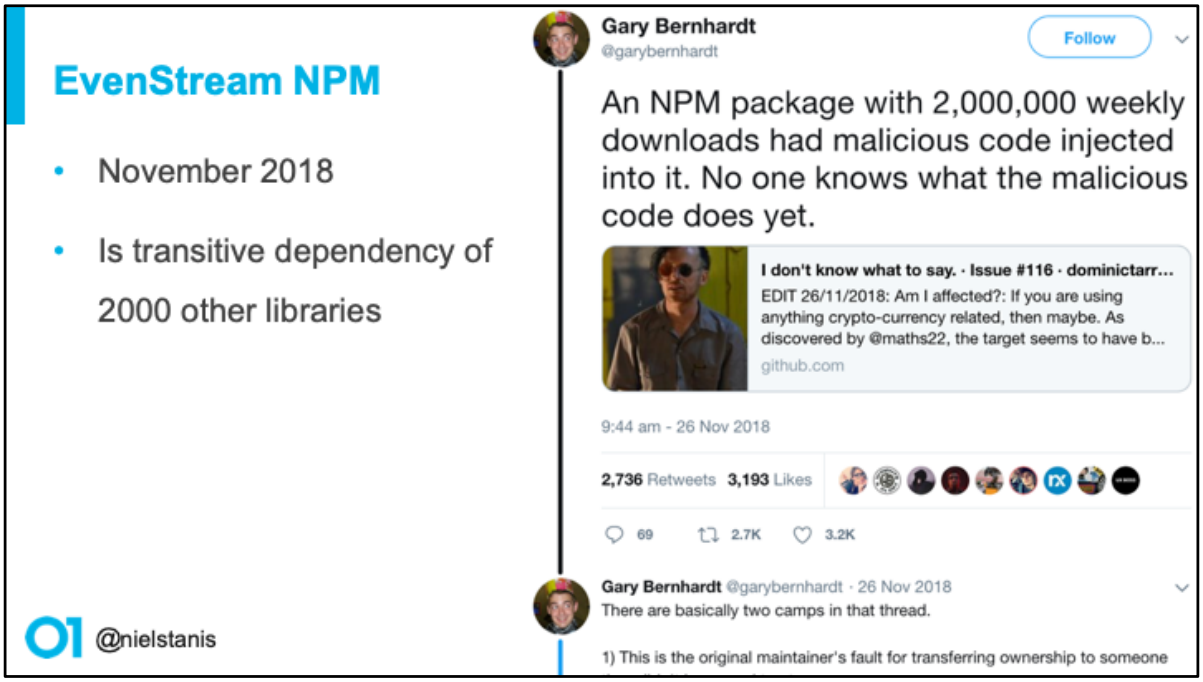
https://help.github.com/en/github/authenticating-to-github/configuring-two-factor-authentication

GIT Commit Signing

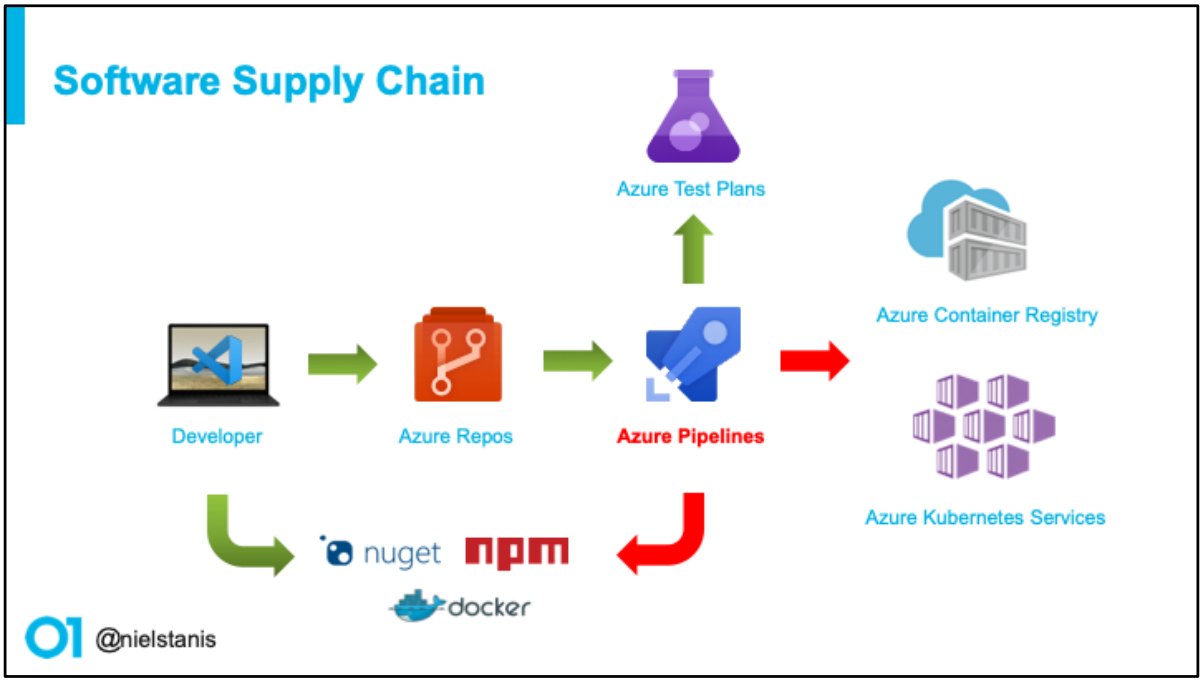


@nielstanis

https://www.hanselman.com/blog/HowToSetupSignedGitCommitsWithAYubiKeyNEOAndGPGAndKeybaseOnWindows.aspx



https://twitter.com/garybernhardt/status/1067111872225136640


Software Supply Chain
Azure Test Plans
Azure Container Registry
Developer
Azure Repos
Azure Pipelines
Azure Kubernetes Services
nuget
npm
docker
@nielstanis

Build / Deployment

- What about hardware? Vendor trust?
- TLS issues?
- Compromised Docker Images
  - Two-Factor authentication in beta
- Build Server can be compromised

@nielstanis

https://docs.docker.com/docker-hub/2fa/

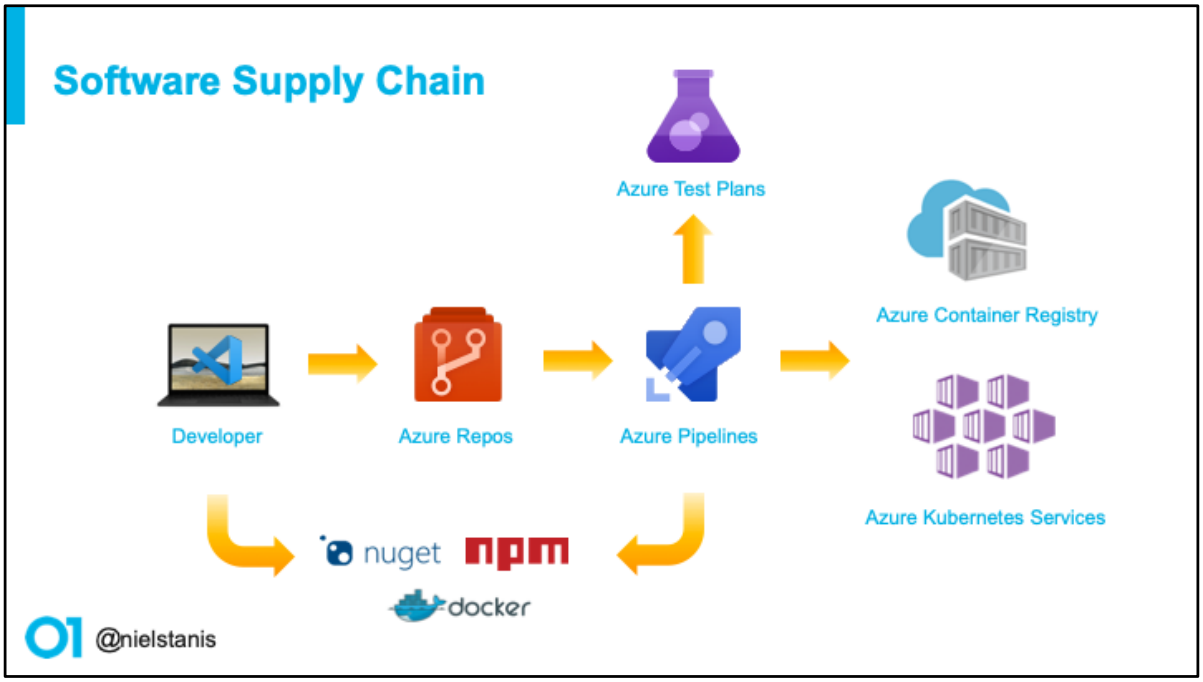
XCodeGhost

@nielstanis

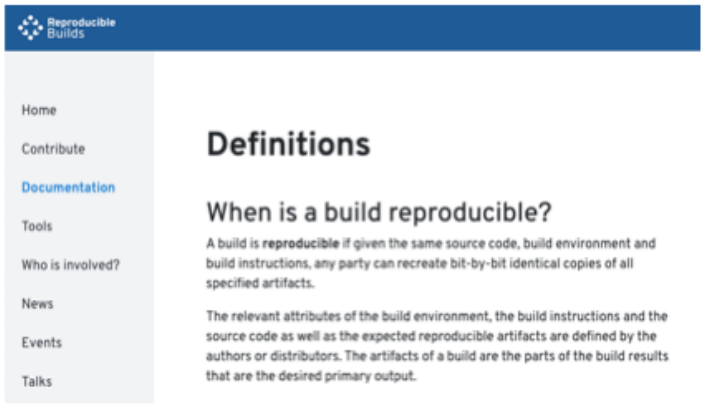
https://blog.malwarebytes.com/cybercrime/2015/09/xcodeghost-malware-infiltrates-app-store/

Webmin Backdoor

@nielstanis


Software Supply Chain
Azure Test Plans
Azure Container Registry
Developer
Azure Repos
Azure Pipelines
Azure Kubernetes Services
nuget
npm
docker
@nielstanis

Reproducible/Deterministic Builds



@nielstanis

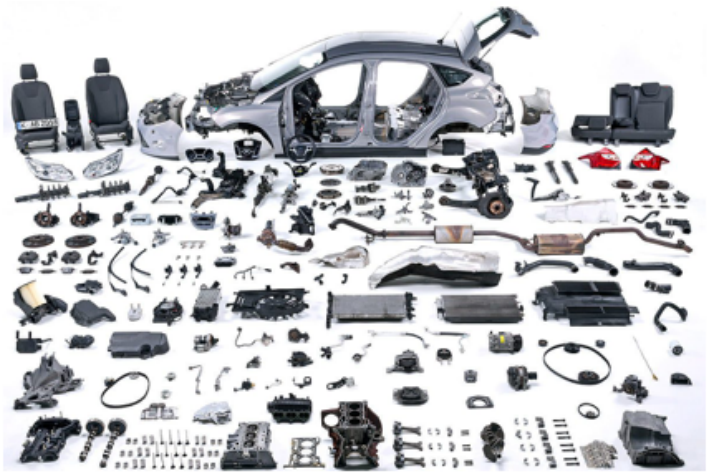
https://reproducible-builds.org/docs/definition/

Reproducable/Deterministic Builds

- Roslyn v1.1 started supporting some kind of determinism on how items are emited
- Given same inputs, the compiled output will always be deterministic
- Inputs can be found in Roslyn compiler docs 'Deterministic Inputs'
  - Current full directory path influences PDB location embedded in PE Header
- PE Header has values that need MSBuild `/deterministic` flag set (default .NET Core)
  - MVID: GUID identifying the PE which will be generated by compiler
  - PDB ID: GUID identifying the generated PDB matching PDB generated on each build
  - Date / Time stamp: Seconds since epoch, calculated on every build

https://blog.paranoidcoding.com/2016/04/05/deterministic-builds-in-roslyn.html
https://github.com/dotnet/roslyn/blob/master/docs/compilers/Deterministic%20Inputs.md
https://github.com/clairernovotny/DeterministicBuilds

Automotive Industry



@nielstanis

Car Supply Chain

Tata Steel Factory

- Iron Ore from Sweden
- ISO 6892-1 Tested/Certified
- Batch #1234

Bosch Factory

- Steel Batch #1234 Tata
- ECE-R90 Tested/Certified
- Serie #45678
- Used by Ford, Volkswagen and KIA

Ford Manufacturing

- Bosch Disk #45678
- Bosal Exhaust #RE9876
- Goodyear Tires #GY8877
- Focus VIN 1234567890

@nielstanis

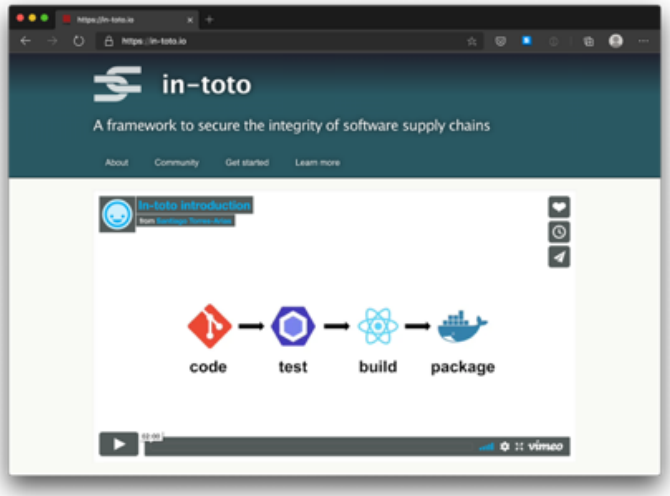
Software Bill of Materials (SBOM) & Policy

- Industry standard of describing the software
  - Producer Identity – Who Created it?
  - Product Identity – What's the product?
  - Integrity – Is the project unaltered?
  - Licensing – How can the project be used?
  - Creation – How was the product created? Process meets requirements?
  - Materials - How was the product created? Materials/Source used?
- CycloneDX – Lightweight SBOM with dependency graph

@nielstanis

https://www.it-cisq.org/software-bill-of-materials/
https://cyclonedx.org/
https://owasp.org/www-project-dependency-check/

In-Toto



@nielstanis

https://in-toto.io/

In-Toto – Demo - Terminology

- **Functionaries** that are identified by public key our supply chain.
  Niels (Project-Owner), Aimee (Developer) and Noud (Packager)
- **Project-Owner** defines a (**Supply Chain) Layout** that describes **what** happens and by **who** and what the produced **Materials** and **Byproducts** are.
- **Link** metadata is output of executed step in the **Layout**
  Materials are input, **Products** are output and can be used as **Materials** in later steps
- With executing the **Verification** all end-products and associated **Link** metadata will be verified

https://in-toto.io/

DataDog & In-Toto

@nielstanis

https://www.datadoghq.com/blog/engineering/secure-publication-of-datadog-agent-integrations-with-tuf-and-in-toto/

Grafeas and Kritis by Google

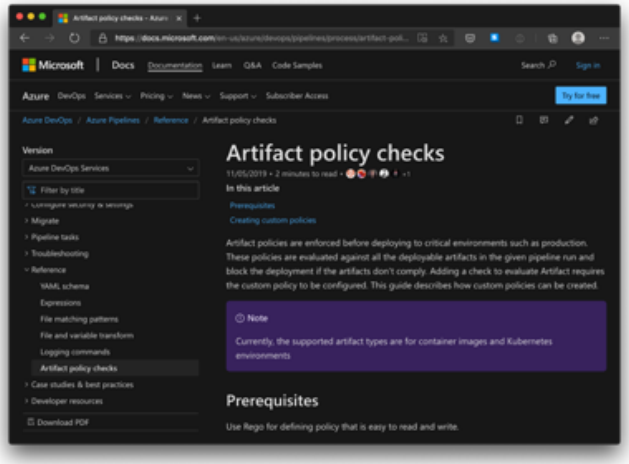
- Grafeas – Component Metadata API
  - Container Analysis API on Google Cloud Platform
- Kritis – Deployment Authorization for Kubernetes Applications
  - Binary Authorization on Google Cloud Platform

@nielstanis

https://grafeas.io/
https://github.com/grafeas/kritis/blob/master/docs/binary-authorization.md

https://www.infoq.com/presentations/supply-grafeas-kritis/
https://www.youtube.com/watch?v=hOzH3mOApjs
https://www.youtube.com/watch?v=05zN-YQxEAM

Azure Pipelines Artifact Policy



@nielstanis

https://devblogs.microsoft.com/devops/secure-software-supply-chain-with-azure-pipelines-artifact-policies/
https://docs.microsoft.com/en-us/azure/devops/pipelines/process/artifact-policy?view=azure-devops

Conclusion

- Be aware of your own (and other used) software supply chain(s).
- Know what you're consuming and pulling into software projects.
- Use MFA on all accounts!
- Integrate security into your software lifecycle.
- Learn more on Software Bill of Materials (SBOM).

@nielstanis


Thanks! Questions?
VERACODE
You change the world, we'll secure it.
https://github.com/nielstanis/dncse2020
ntanis at veracode.com
@nielstanis on Twitter