



Niels Tanis

The Rise of Software Supply-Chain Attacks

How Secure is your .NET Application?

CODECAMP_

Who am I?

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

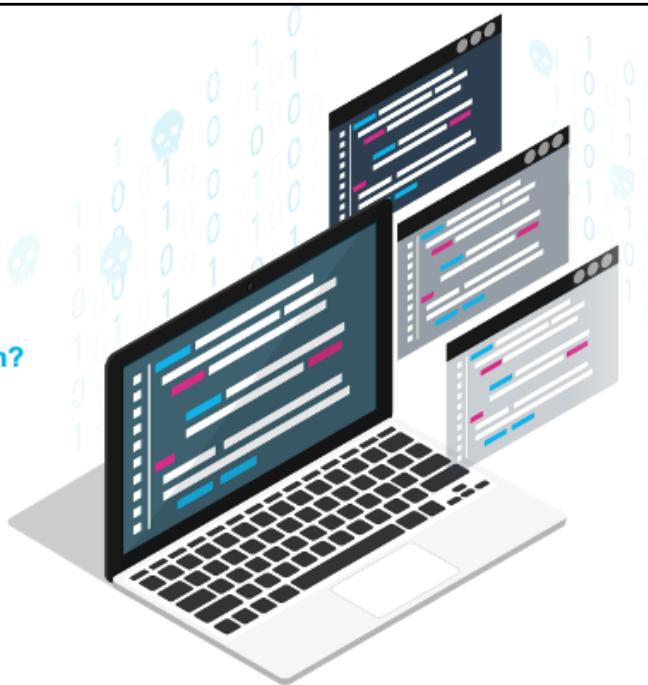


O1 @nielstanis

The Rise of Software Supply-Chain Attacks

How Secure is your .NET Application?

O1 @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

O1 @nielstanis

Hacking History

- Started out with phreaking in late '50-'60.



O1 @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 (Security Administrator Tool for Analyzing Networks) 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
8929/tcp open     nping-echo
```

O1 @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

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Smashing The Stack For Fun And Profit
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

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 span; see also alias bug, fandango on core, memory leak, precedence lossage, overrun screw.



<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 WF'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
```

O1 @nielstanis

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

Code Red & SQL Slammer

- Microsoft Internet Information Server, July 2001

```
GET /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNN  
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN  
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN  
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN  
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN  
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN  
%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
```

01 @nielstanis

[https://en.wikipedia.org/wiki/Code_Red_\(computer_worm\)](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

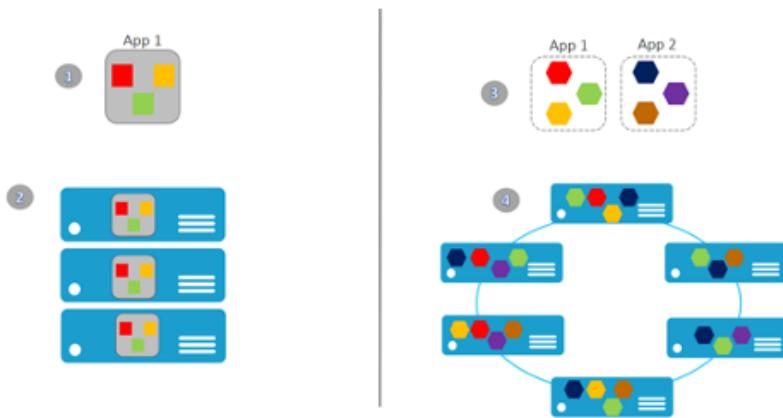
Every few years I have sent out a memo talking about the highest priority for Microsoft. Two years ago, it was the kickoff of our .NET strategy. Before that, it was several memos about the importance of the Internet to our future and the ways we could make the Internet truly useful for people. Over the last year it has become clear that

O1 @nielstanis

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

Changes in Software Architecture

- Monolith → Microservices → Serverless



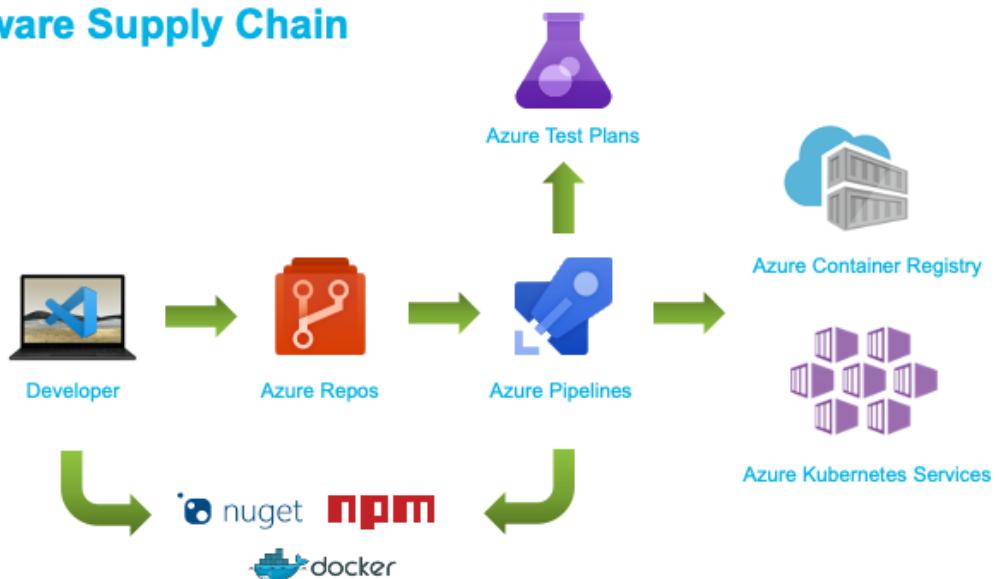
O1 @nielstanis

What's a Supply-Chain?



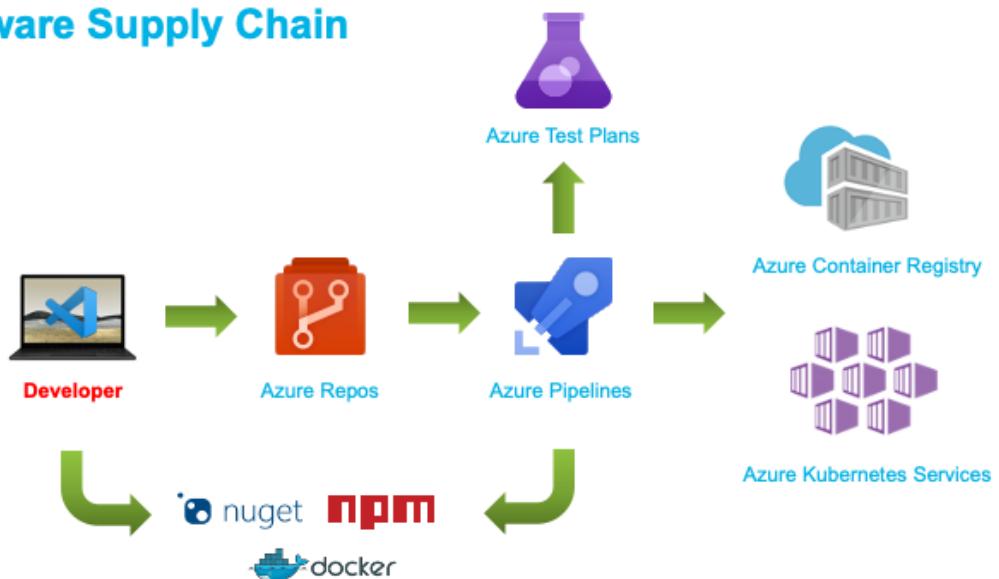
O1 @nielstanis

Software Supply Chain



O1 @nielstanis

Software Supply Chain



O1 @nielstanis

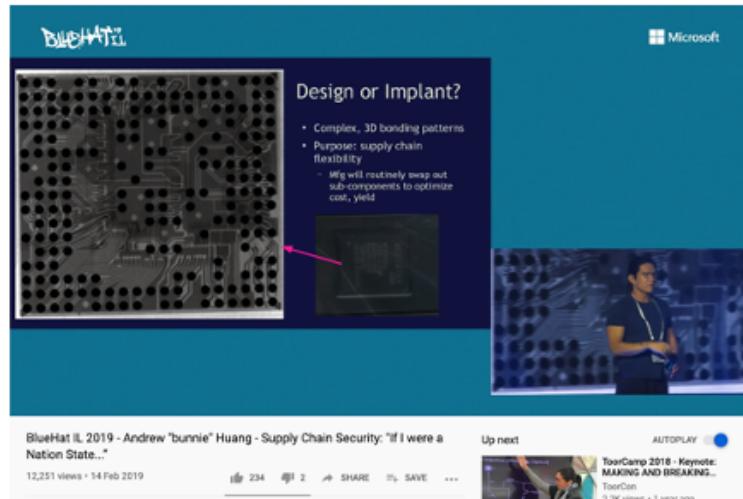
Development Machine

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



O1 @nielstanis

Hacking Hardware



O1 @nielstanis

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

Development Machine



- Installs on machine – HomeBrew on Mac
- ```
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"
```

O1 @nielstanis

## Development Machine



- Installs on machine – Chocolatey on Windows

Chocolatey Install:

Individual   Organization

1. First, ensure that you are using an [administrative shell](#) - you can also install as a non-admin, check out [Non-Administrative Installation](#).

2. Install with powershell(Lexer)

**NOTE:** Please inspect <https://chocolatey.org/install.ps1> prior to running any of these scripts to ensure safety. We already know it's safe, but you should verify the security and contents of [any script](#) from the internet you are not familiar with. All of these scripts download a remote PowerShell script and execute it on your machine. We take security very seriously. [Learn more about our security protocols](#).

01 @nielstanis

## Octopus Scanner - NetBeans

The screenshot shows a GitHub Security Lab page. At the top, there's a navigation bar with links for Back to GitHub.com, Security Lab, Bounties, CodeQL, Research (which is underlined in blue), Advisories, Get Involved, and Events. Below the navigation is a large, light-gray rectangular area with a dotted border. In the center of this area, the date May 28, 2022 is displayed. Below the date is the title "The Octopus Scanner Malware: Attacking the open source supply chain". Underneath the title is a small circular profile picture and the name Alvaro Muñoz. A short descriptive paragraph follows the title.

May 28, 2022

**The Octopus Scanner Malware:  
Attacking the open source  
supply chain**

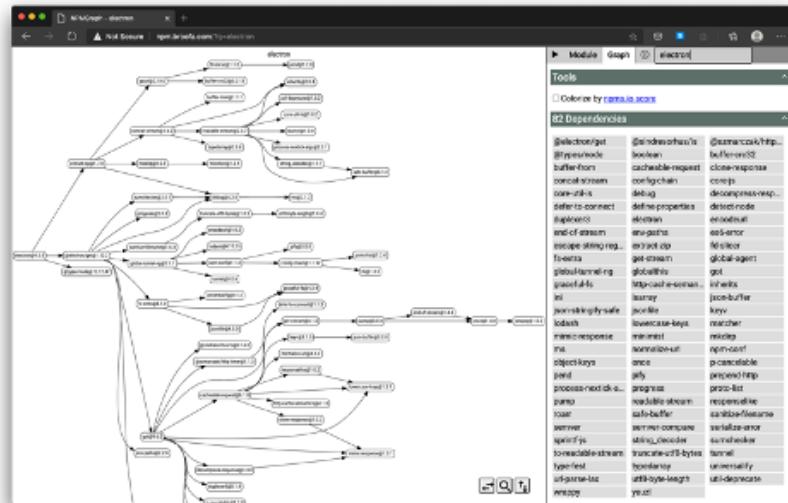
Alvaro Muñoz

Securing the open source supply chain is an enormous task. It goes far beyond a security assessment or just patching for the latest CVEs. Supply chain security is about the integrity of the entire software development and delivery ecosystem. From the code commits themselves, to how they flow through the CI/CD pipeline, to the actual delivery of releases, there's the potential for loss of integrity and security concerns, throughout the entire lifecycle.

O1 @nielstanis

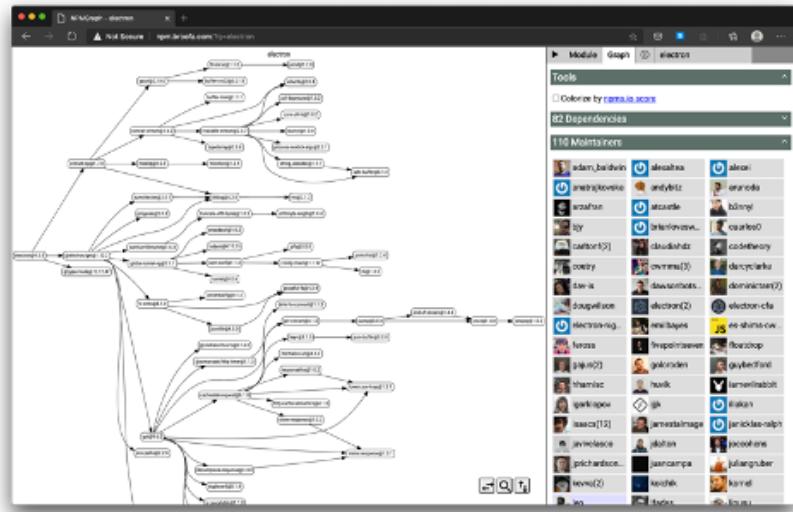
<https://securitylab.github.com/research/octopus-scanner-malware-open-source-supply-chain>

## Visual Studio Code



O1 @nielstanis

# Visual Studio Code



01 @nielstanis

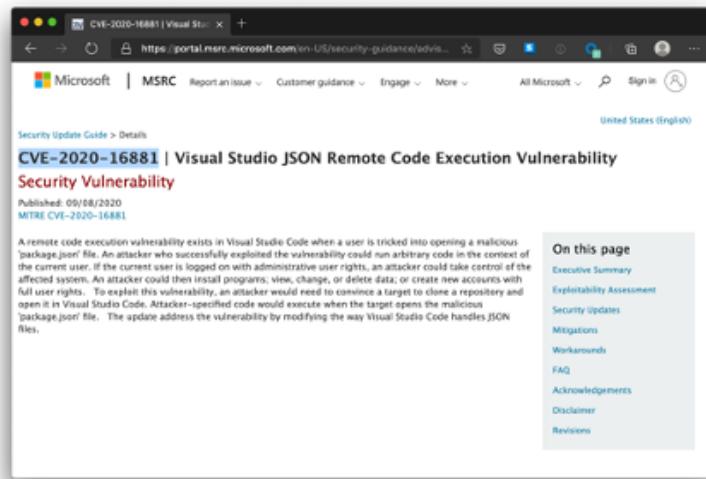
## Visual Studio Code

The screenshot shows a web browser window with the URL <https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/cve-2018-1000136-electron-nodeintegration-bypass/>. The page title is "CVE-2018-1000136 - Electron nodeIntegration Bypass". The post was published on May 10, 2018, by Brendan Scarvell. The content discusses a vulnerability in Electron that allowed remote code execution through a failure to correctly sanitize user-supplied input. It mentions that Electron applications are essentially web apps and can execute system commands on the client-side. A note about the `WebView` tag is also present.

O1 @nielstanis

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

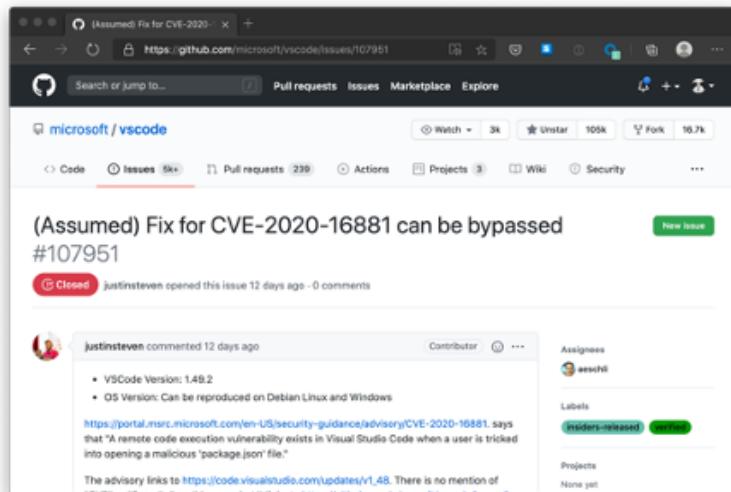
## Visual Studio Code



O1 @nielstanis

<https://github.com/microsoft/vscode/issues/107951>

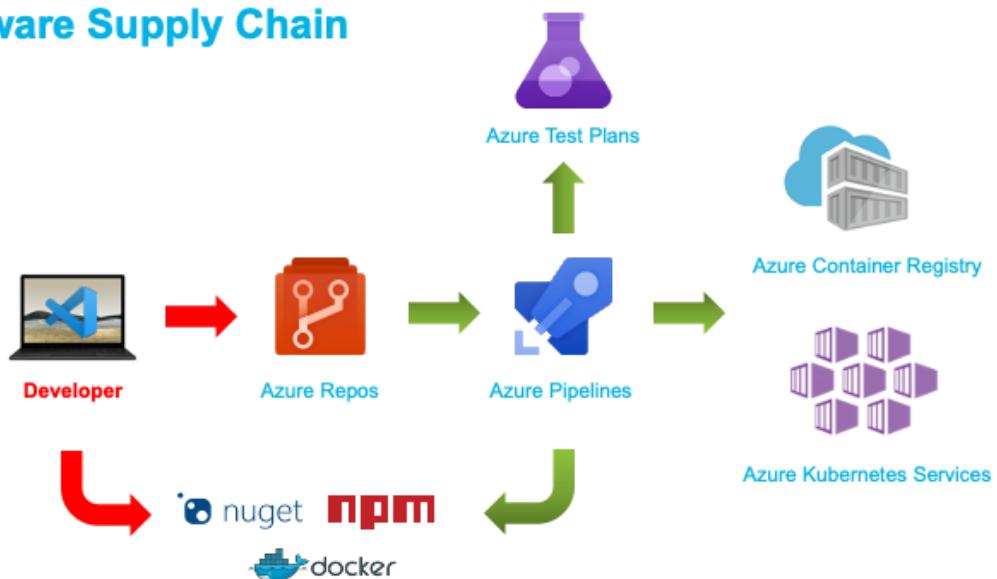
## Visual Studio Code



O1 @nielstanis

<https://github.com/microsoft/vscode/issues/107951>

## Software Supply Chain



O1 @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

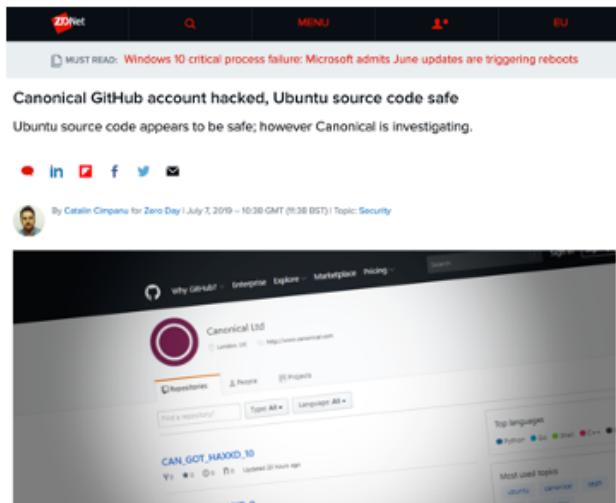
O1 @nielstanis

[https://www.ssllabs.com/downloads/SSL\\_Threat\\_Model.png](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



O1 @nielstanis

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

## Microsoft GitHub Account

The screenshot shows a news article from Threatpost. The title is "Report: Microsoft's GitHub Account Gets Hacked". The article features a large image of the GitHub logo with a cartoon cat. Below the image, a text snippet reads: "The Shiny Hunters hacking group said it stole 500 GB of data from the tech giant's repositories on the developer platform, which it owns." The author is Elizabeth Montalbano, and the date is May 8, 2020, at 11:36 am. To the right of the main article, there is a sidebar titled "INFOSEC INSIDER" with several news items:

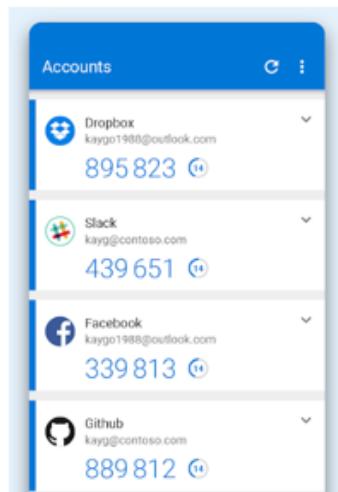
- Helping Remote Workers Overcome Remote Attacks (May 8, 2020)
- Understanding the Payload-Less Email Attacks Evading Your Security Team (May 6, 2020)
- Long Tail Analysis: A New Hope in the Cybercrime Battle (May 25, 2020)
- The Windows 7 Postmortem: What's at Stake (May 18, 2020)
- VPN Concerns with Unplanned Remote Employees (May 5, 2020)

At the bottom right of the sidebar, there is a "Subscribe to Threatpost Today" button.

**O1** @nielstanis

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

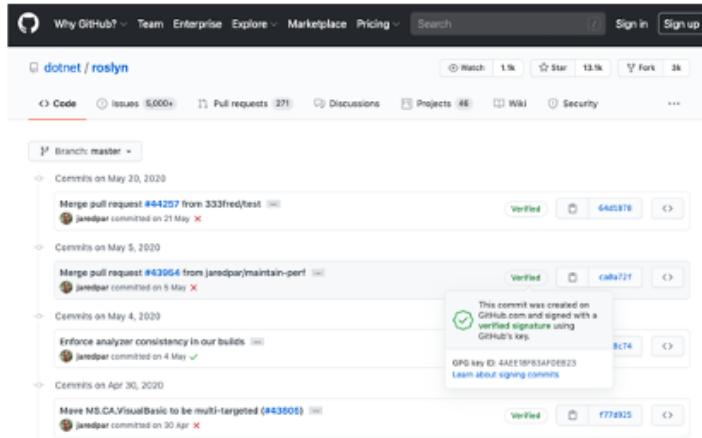
## Use MFA on source-repository



O1 @nielstanis

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

## GIT Commit Signing



A screenshot of a GitHub repository page for 'dotnet / roslyn'. The 'Code' tab is selected. A commit from 'jaredpar' on May 20, 2020, is shown with a 'Verified' badge, GPG key ID '64d28f', and a link to 'View signature'. Another commit from 'jaredpar' on May 5, 2020, is also verified. Other commits listed include ones from 'Enforce analyzer consistency in our builds' on May 4, 2020, and 'Move MS.CA.VisualBasic to be multi-targeted (#43808)' on April 30, 2020.



O1 @nielstanis

[https://www.hanselman.com/blog/HowToSetupSignedGitCommitsWithAYubiKeyNEO AndGPGAndKeybaseOnWindows.aspx](https://www.hanselman.com/blog/HowToSetupSignedGitCommitsWithAYubiKeyNEOAndGPGAndKeybaseOnWindows.aspx)

**EvenStream NPM**

- November 2018
- Is transitive dependency of 2000 other libraries

**Gary Bernhardt**  
@garybernhardt

An NPM package with 2,000,000 weekly downloads had malicious code injected into it. No one knows what the malicious code does yet.

I don't know what to say. - Issue #116 - dominictarr...  
EDIT 26/11/2018: Am I affected?: If you are using anything crypto-currency related, then maybe. As discovered by @maths22, the target seems to have b...  
[github.com](https://github.com)

9:44 am - 26 Nov 2018

2,736 Retweets 3,193 Likes

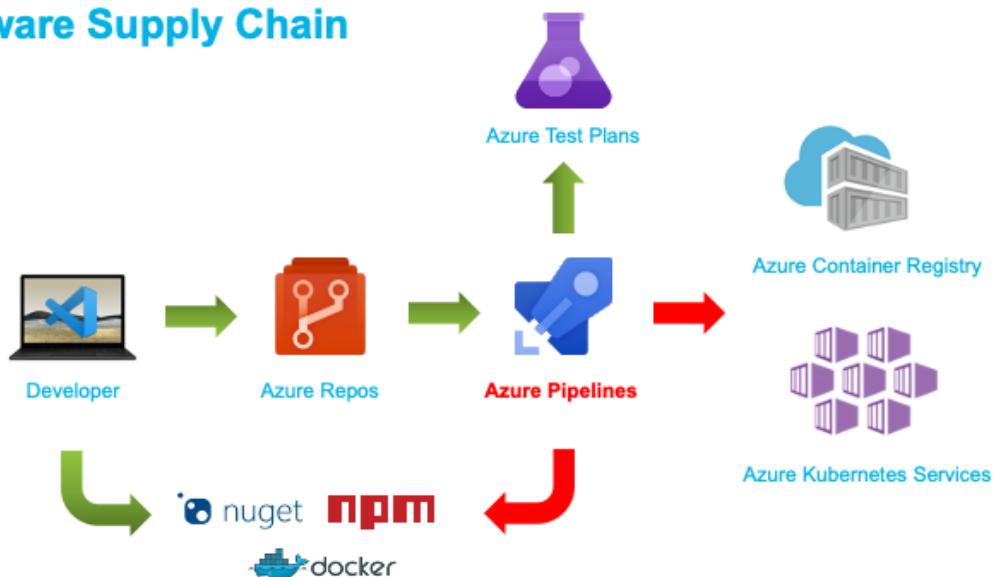
89 2.7K 3.2K

**Gary Bernhardt** @garybernhardt · 26 Nov 2018  
There are basically two camps in that thread.

1) This is the original maintainer's fault for transferring ownership to someone

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

## Software Supply Chain



O1 @nielstanis

## Build / Deployment

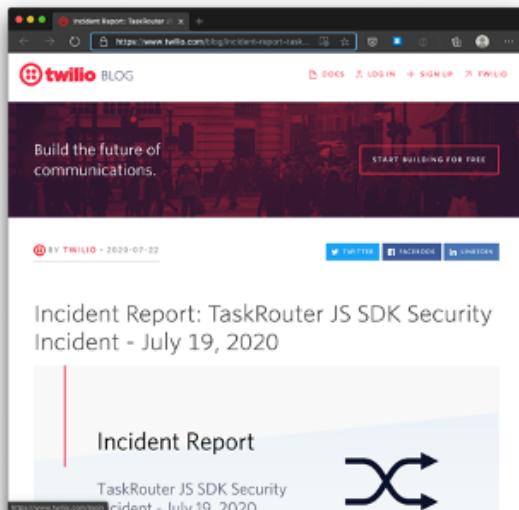
- What about hardware? Vendor trust?
- TLS issues?
- Compromised Docker Images
  - Two-Factor authentication in beta
- Build Server can be compromised



O1 @nielstanis

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

## Twilio SDK



O1 @nielstanis

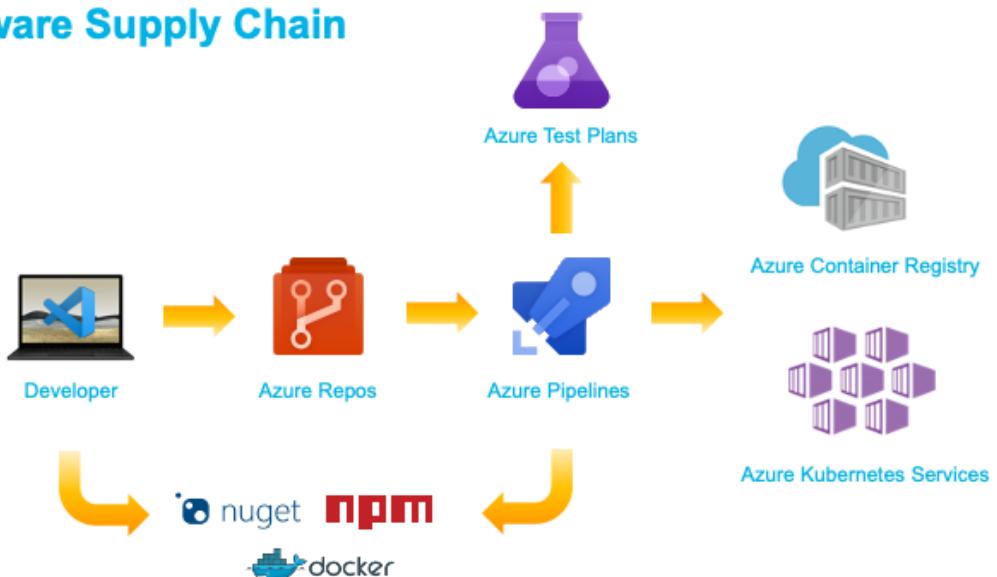
<https://www.twilio.com/blog/incident-report-taskrouter-js-sdk-july-2020>

## Webmin Backdoor

The screenshot shows the official Webmin website. At the top, there's a navigation bar with links for Home, Downloads, Documentation, Usermin, Virtualmin, Cloudmin, and Community. Below the navigation is a sidebar with sections for Download Webmin 1.941 (RPM, Debian Package, TAR file, Solaris Package, Development Versions, Third-Party Modules) and Webmin Links (Introduction To Webmin, Supported Systems, Module Documentation, Screenshots, Standard Modules, Supported Languages, Updated Modules). The main content area is titled "Webmin 1.890 Exploit - What Happened?". It discusses a backdoor in Webmin version 1.890 that could allow anyone with knowledge of it to execute commands as root. It notes that versions 1.900 to 1.920 also contained a backdoor using similar code, but it was not exploitable in a default Webmin install. Only if the admin had enabled the feature at Webmin -> Webmin Configuration -> Authentication to allow changing of expired passwords could it be used by an attacker. The text also mentions two architectural bugs: one where the Webmin source code was maliciously modified to add a non-obvious vulnerability, and another where a timestamp on a file was set back, causing it not to show up in any GH diffs. The final point states that the vulnerable file was reverted to the checked-in version from GitHub, but was modified again by the attacker in July 2018.

O1 @nielstanis

## Software Supply Chain



O1 @nielstanis

## Reproducible/Deterministic Builds

The screenshot shows a website for "Reproducible Builds". The header features a blue bar with the "Reproducible Builds" logo. Below the header is a navigation menu with links: Home, Contribute, Documentation (which is highlighted in blue), Tools, Who is involved?, News, Events, and Talks. To the right of the menu, the word "Definitions" is displayed in large, bold, black font. Underneath "Definitions", the heading "When is a build reproducible?" is shown in bold black font. A detailed explanation follows: "A build is **reproducible** if given the same source code, build environment and build instructions, any party can recreate bit-by-bit identical copies of all specified artifacts. The relevant attributes of the build environment, the build instructions and the source code as well as the expected reproducible artifacts are defined by the authors or distributors. The artifacts of a build are the parts of the build results that are the desired primary output."

O1 @nielstanis

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

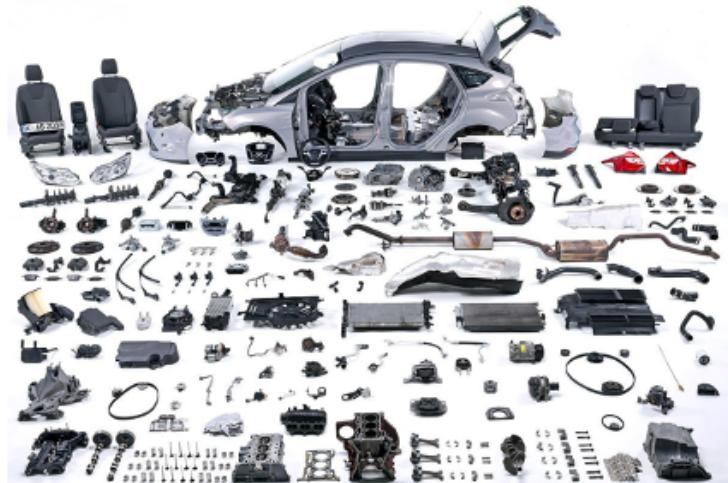
## Reproducible/Deterministic Builds

- Roslyn v1.1 started supporting some kind of determinism on how items are emitted
- 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

O1 @nielstanis

<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



O1 @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

O1 @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
- [National Telecommunications and Information Administration](https://www.ntia.gov/SBOM) <https://www.ntia.gov/SBOM>

O1 @nielstanis

<https://www.it-cisq.org/software-bill-of-materials/>

<https://cyclonedx.org/>

<https://owasp.org/www-project-dependency-check/>

## In-Toto



O1 @nielstanis

<https://in-toto.io/>

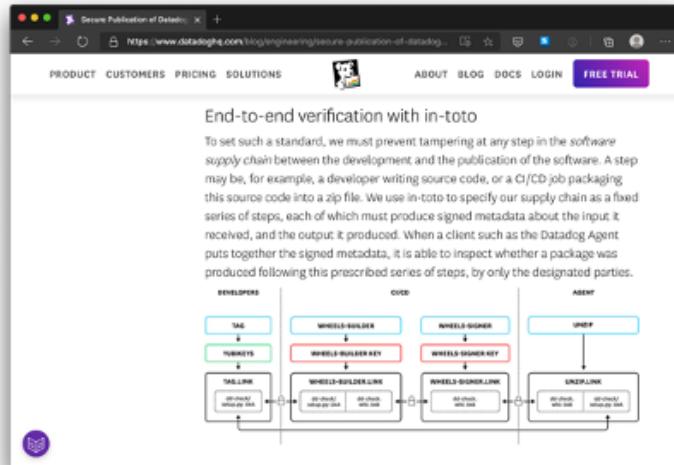
## In-Toto – Demo - Terminology

- **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

O1 @nielstanis

<https://in-toto.io/>

## DataDog & In-Toto



O1 @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



O1 @nielstanis

<https://grafeas.io/>

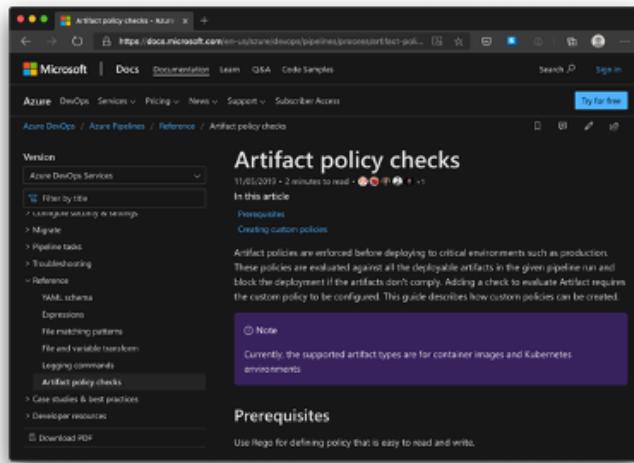
<https://github.com/grafeas/kritis/blob/master/docs/binary-authorization.md>

<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



O1 @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).

O1 @nielstanis



Thanks! Questions?

**VERACODE**

You change the world, we'll secure it.

<https://github.com/nielstanis/codecamp2020>

ntanis at veracode.com

@nielstanis on Twitter