

NDC { London }

Securing the Software Supply-Chain

Niels Tanis

0101
0101

Who am I?

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



0101
0101

The Rise of Software Supply-Chain Attacks

How Secure is your .NET Application?





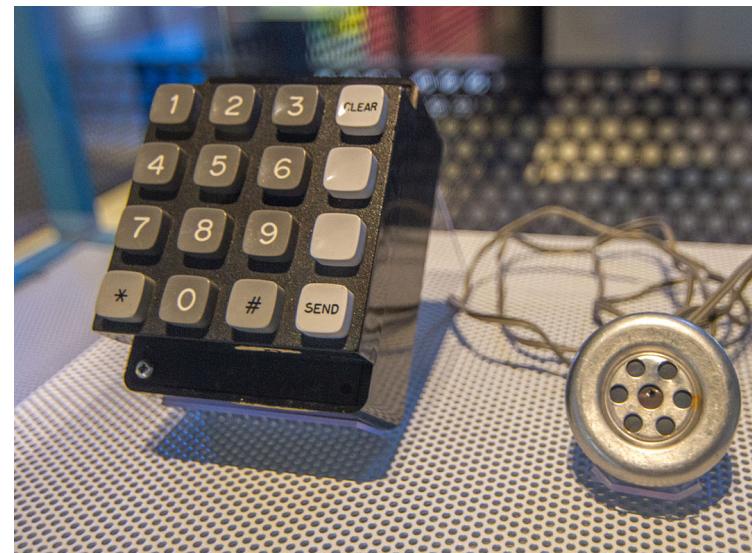
Agenda

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

0101
0101

Hacking History

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



The Tech



Vol. 83, No. 24 Cambridge, Mass., Wednesday, Nov. 20, 1963 5c

Services curtailed

Telephone hackers active

By Henry Lichstein

Many telephone services have been curtailed because of so-called hackers, according to Professor Carlton Tucker, administrator of the Institute phone system.

system to many areas without a prorata charge. Among the tie-lines discovered have been ones to the Millstone Radar Facility, the Sudbury defense installation, IBM in Kingston, New York, and the MITRE Corporation.

Getting connected!

0101
0101

- SATAN (Security Administrator Tool for Analyzing Networks) by Wietse Venema and Dan Farmer in 1995.
- NMAP (Network Mapper) by Gordon Lyon 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
```

Smashing the Stack...

0101
0101

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

SQL Injection

0101
0101

- 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%  
-----
```



Code Red & SQL Slammer

- Microsoft Internet Information Server, July 2001

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



Bill Gates - Email to all MS FTE

BILL GATES BUSINESS 01.17.02 12:00 PM

Bill Gates: Trustworthy Computing

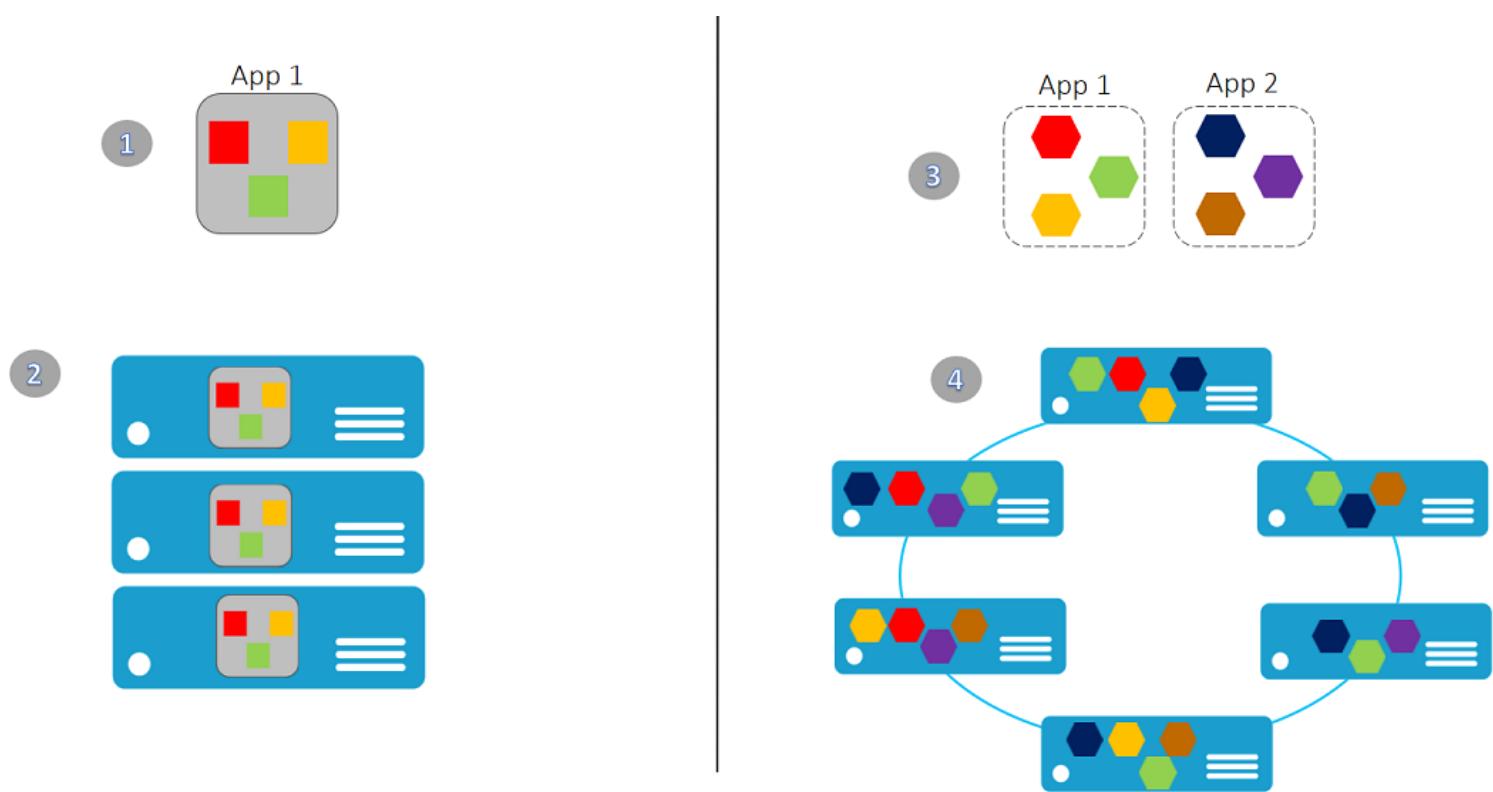
This is the e-mail Bill Gates sent to every full-time employee at Microsoft, in which he describes the company's new strategy emphasizing security in its products.
From: Bill Gates
Sent: Tuesday, January 15, 2002 5:22 PM
To: Microsoft and Subsidiaries: All FTE
Subject: Trustworthy computing

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

Changes in Software Architecture

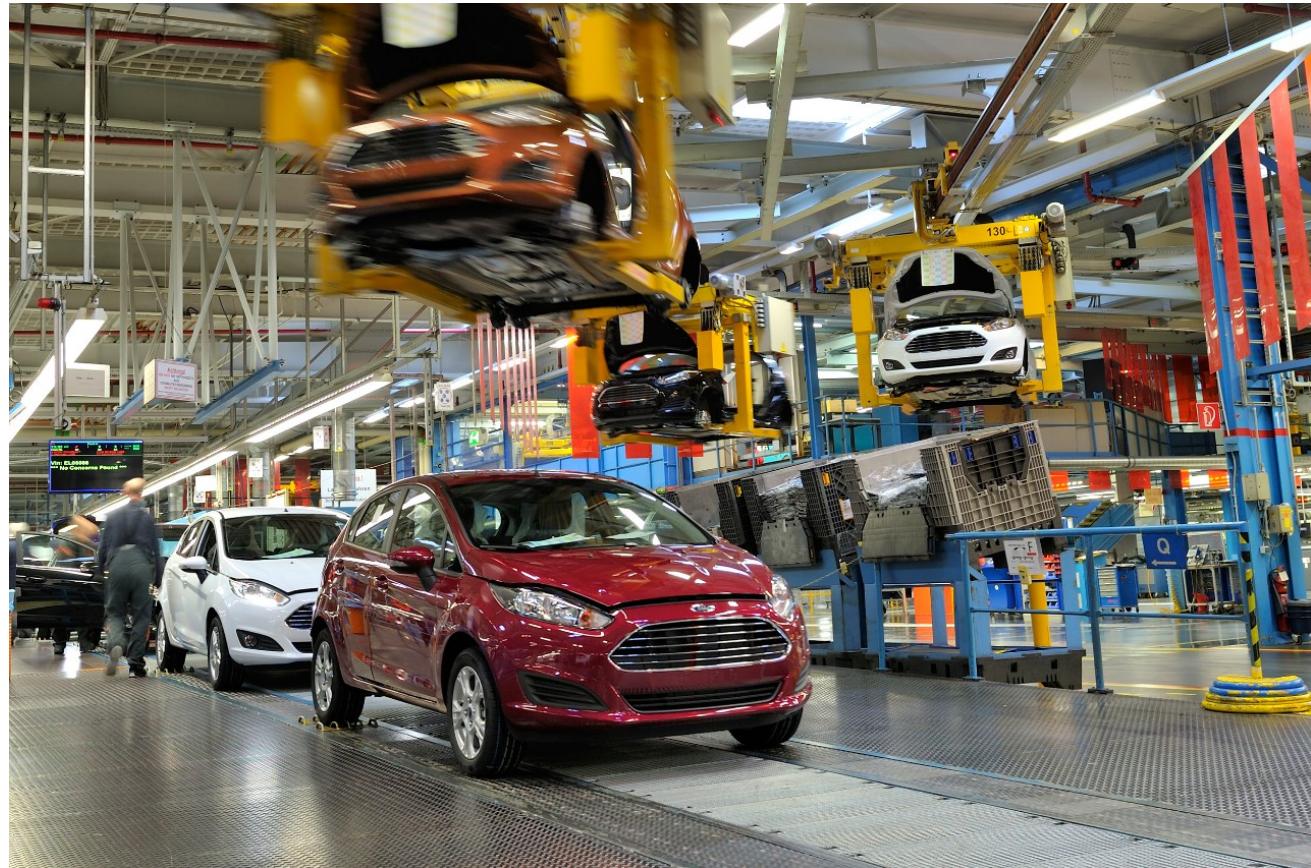
0101
0101

- Monolith
- Microservices
- Serverless



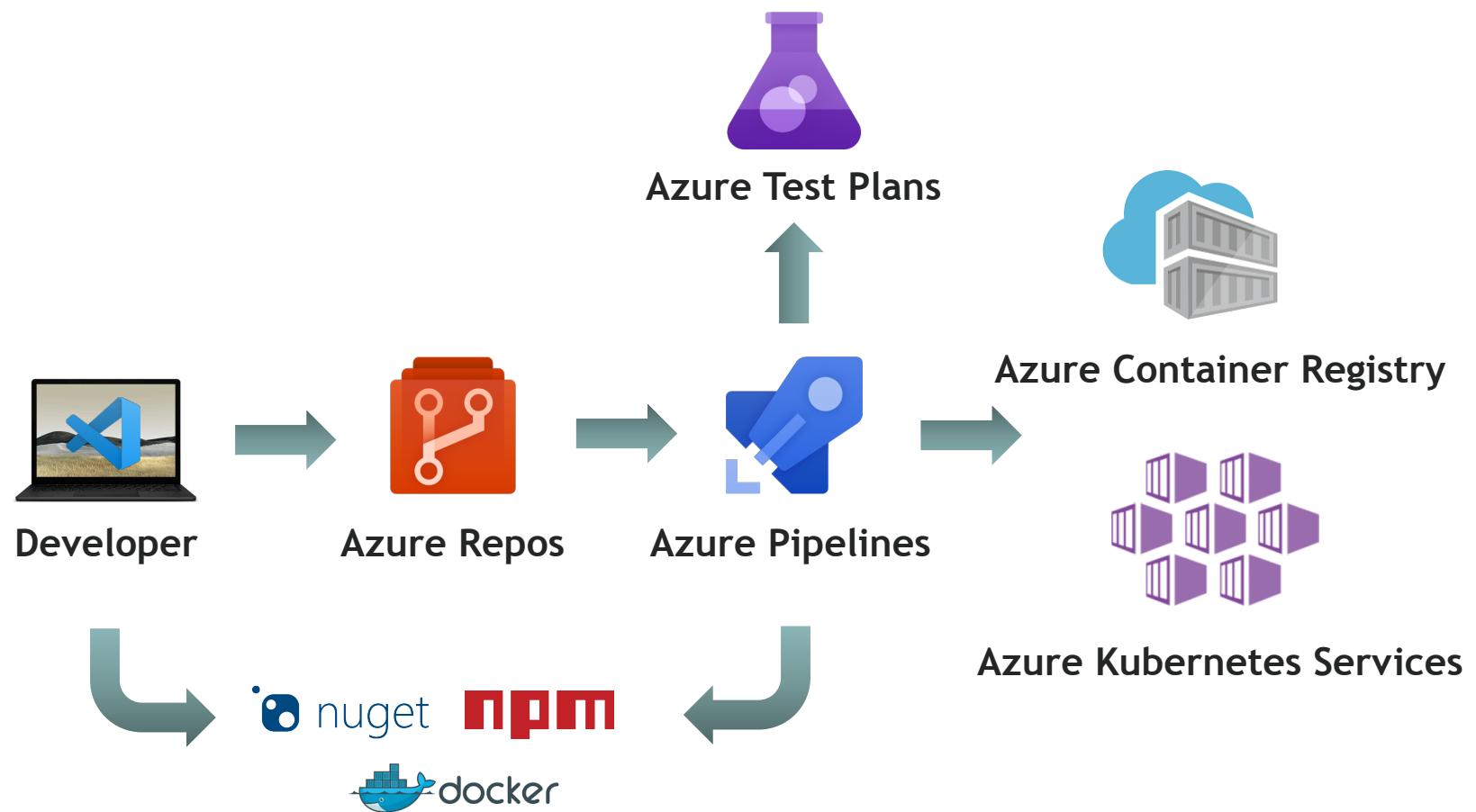
0101
0101

What is a Supply Chain?



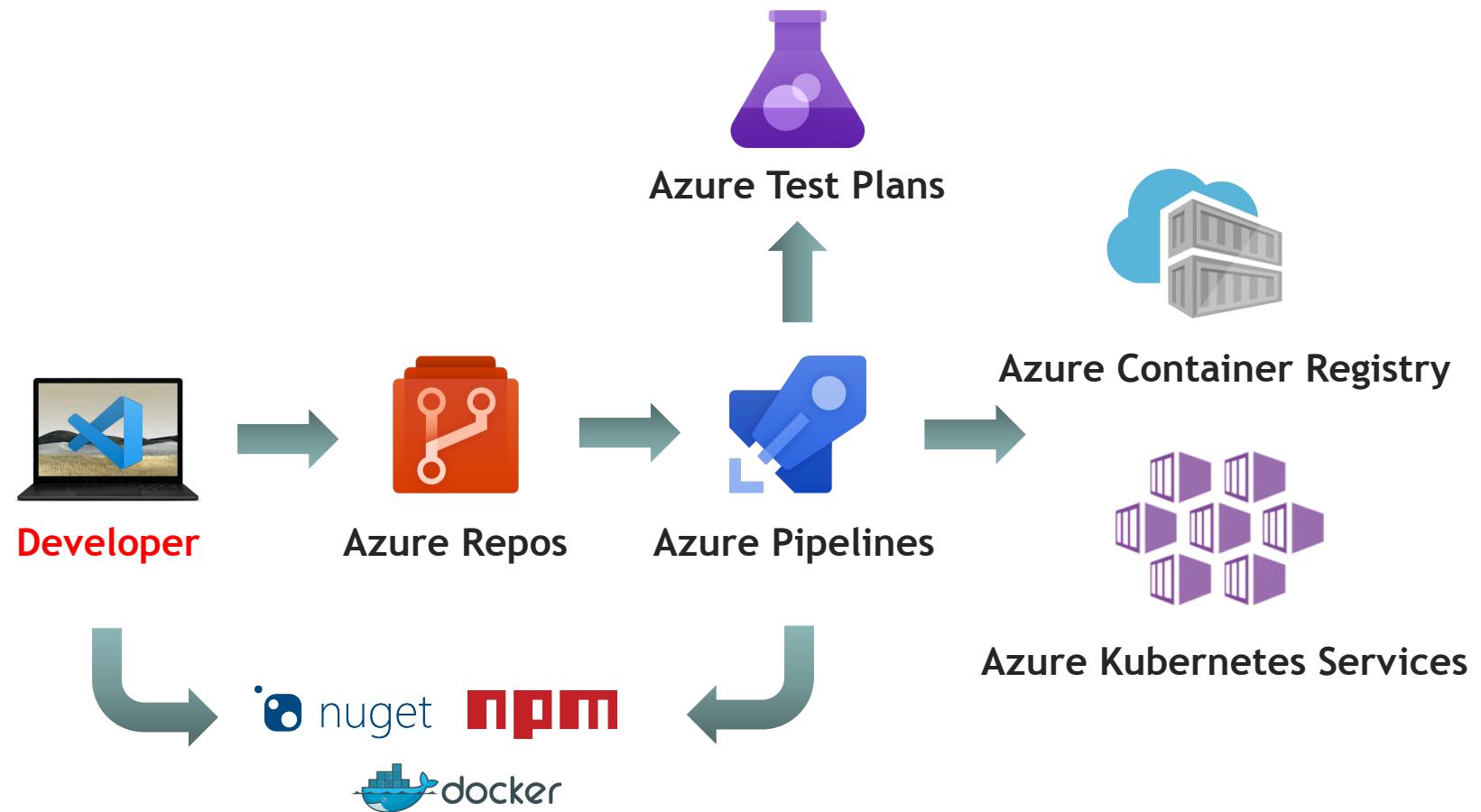
Software Supply Chain

0101
0101



Software Supply Chain

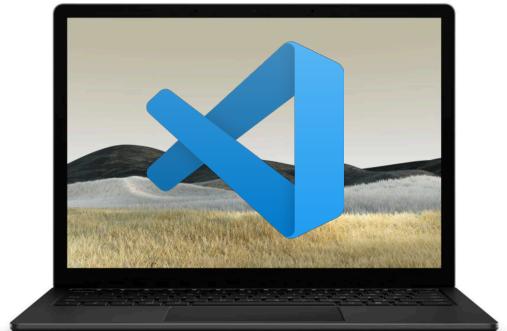
0101
0101



0101
0101

Development Machine

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



0101
0101

Hacking Hardware

The image shows a screenshot of a video player interface. At the top left is the BlueHat IL logo. On the right is the Microsoft logo. The main content area has a dark blue background with a grid pattern. On the left side of the content area is a large image of a printed circuit board (PCB) with many black circular pads. A pink arrow points from the text "Design or Implant?" towards this PCB image. To the right of the PCB is a smaller image of a person speaking on stage. Below the main content area is a dark blue footer bar containing the video title, view count, and other standard video controls.

BlueHat IL

Microsoft

Design or Implant?

- Complex, 3D bonding patterns
- Purpose: supply chain flexibility
 - Mfg will routinely swap out sub-components to optimize cost, yield

BlueHat IL 2019 - Andrew "bunnie" Huang - Supply Chain Security: "If I were a Nation State..."

12,251 views • 14 Feb 2019

234 2 SHARE SAVE ...

Up next

AUTOPLAY

ToorCamp 2018 - Keynote: MAKING AND BREAKING...
ToorCon

Development Machine

0101
0101

- Installs on machine - HomeBrew on Mac

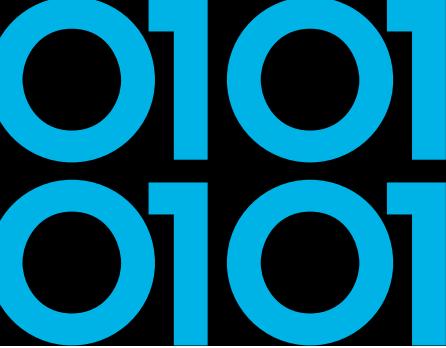
README.md

Homebrew (un)installer

Install Homebrew

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"
```

More installation information and options at <https://docs.brew.sh/Installation.html>.



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.exe

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](#).



Octopus Scanner - NetBeans

May 28, 2020

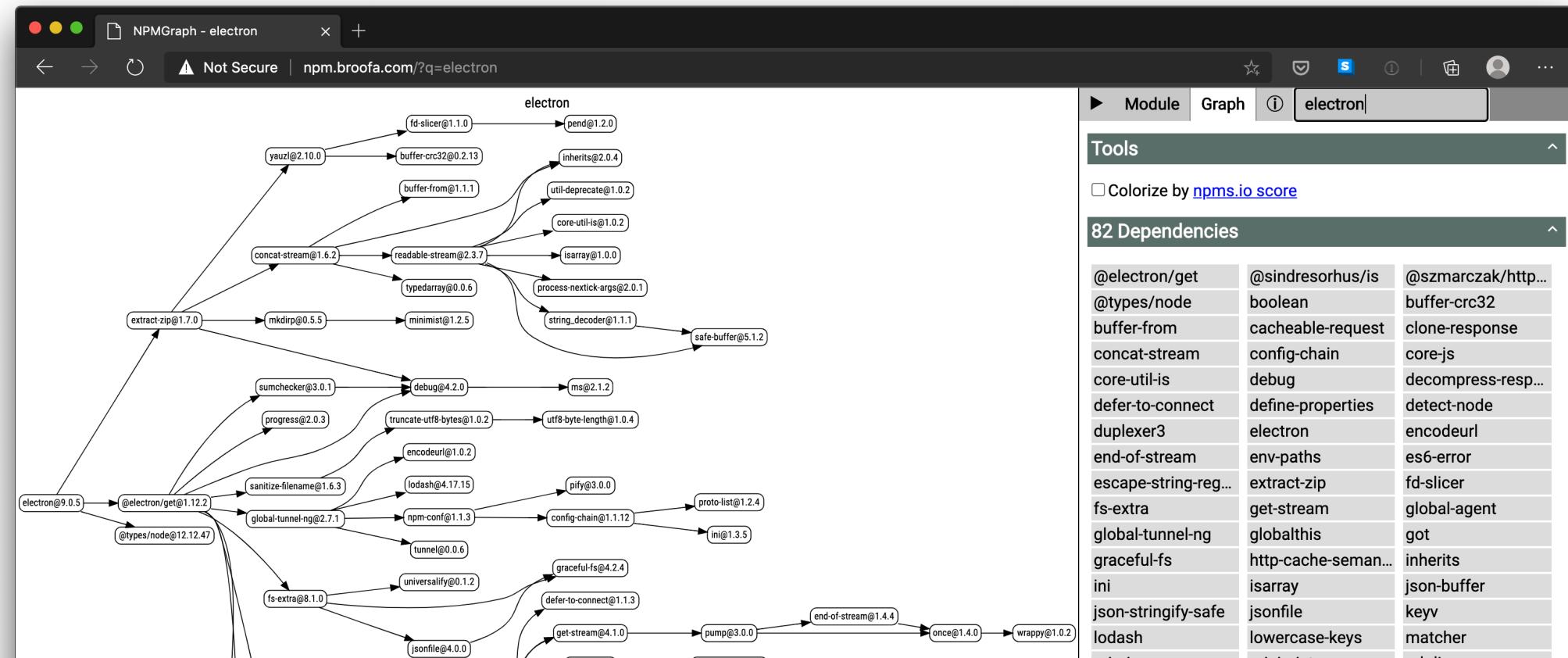
The Octopus Scanner Malware: Attacking the open source supply chain



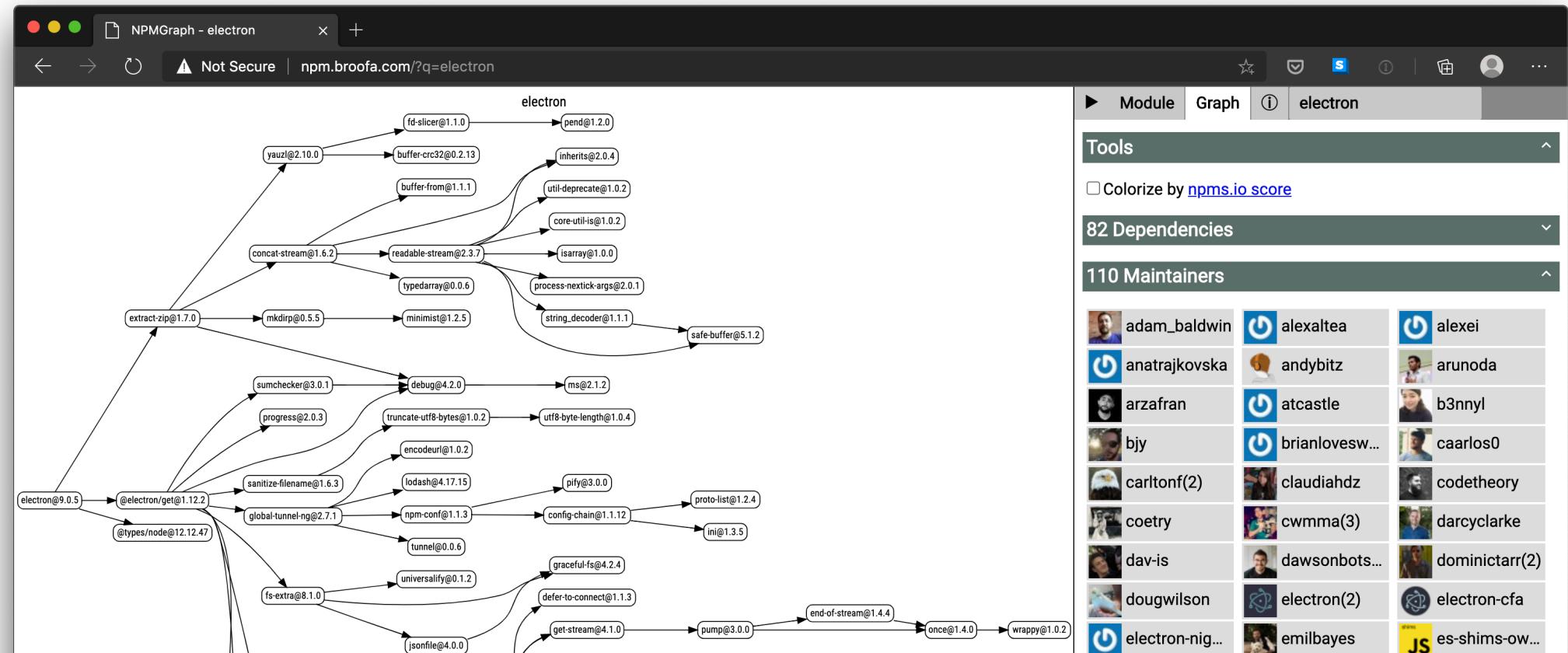
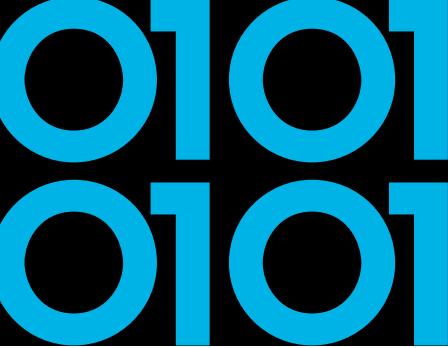
Alvaro Muñoz

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.

Visual Studio Code

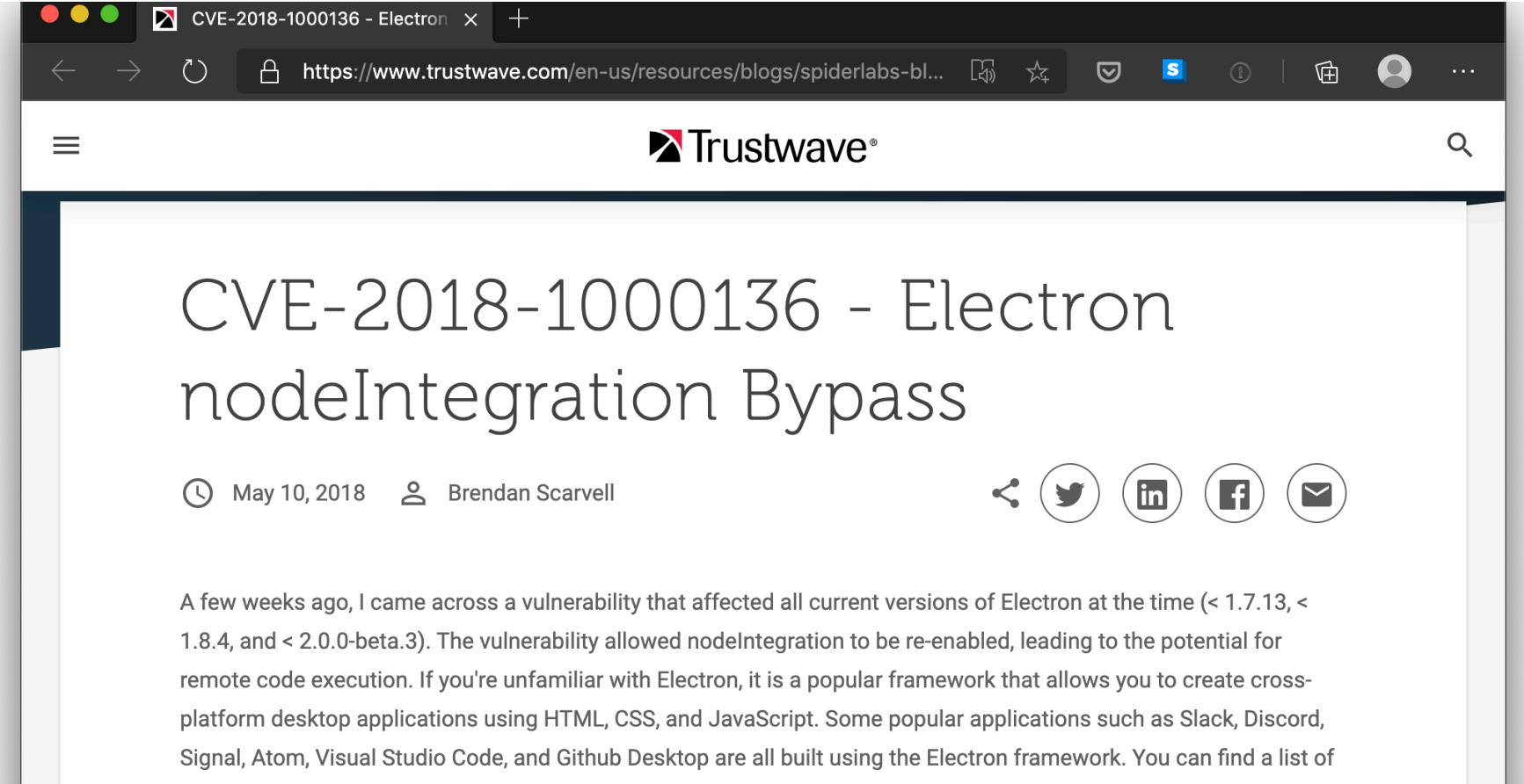


Visual Studio Code



Visual Studio Code

0101
0101



The screenshot shows a web browser window with the title bar "CVE-2018-1000136 - Electron". The URL in the address bar is <https://www.trustwave.com/en-us/resources/blogs/spiderlabs-bl...>. The page content is from Trustwave's blog, featuring a large heading "CVE-2018-1000136 - Electron nodeIntegration Bypass". Below the heading, it says "May 10, 2018" and "Brendan Scarvell". To the right of the author information are sharing icons for Twitter, LinkedIn, Facebook, and Email.

A few weeks ago, I came across a vulnerability that affected all current versions of Electron at the time (< 1.7.13, < 1.8.4, and < 2.0.0-beta.3). The vulnerability allowed nodeIntegration to be re-enabled, leading to the potential for remote code execution. If you're unfamiliar with Electron, it is a popular framework that allows you to create cross-platform desktop applications using HTML, CSS, and JavaScript. Some popular applications such as Slack, Discord, Signal, Atom, Visual Studio Code, and Github Desktop are all built using the Electron framework. You can find a list of

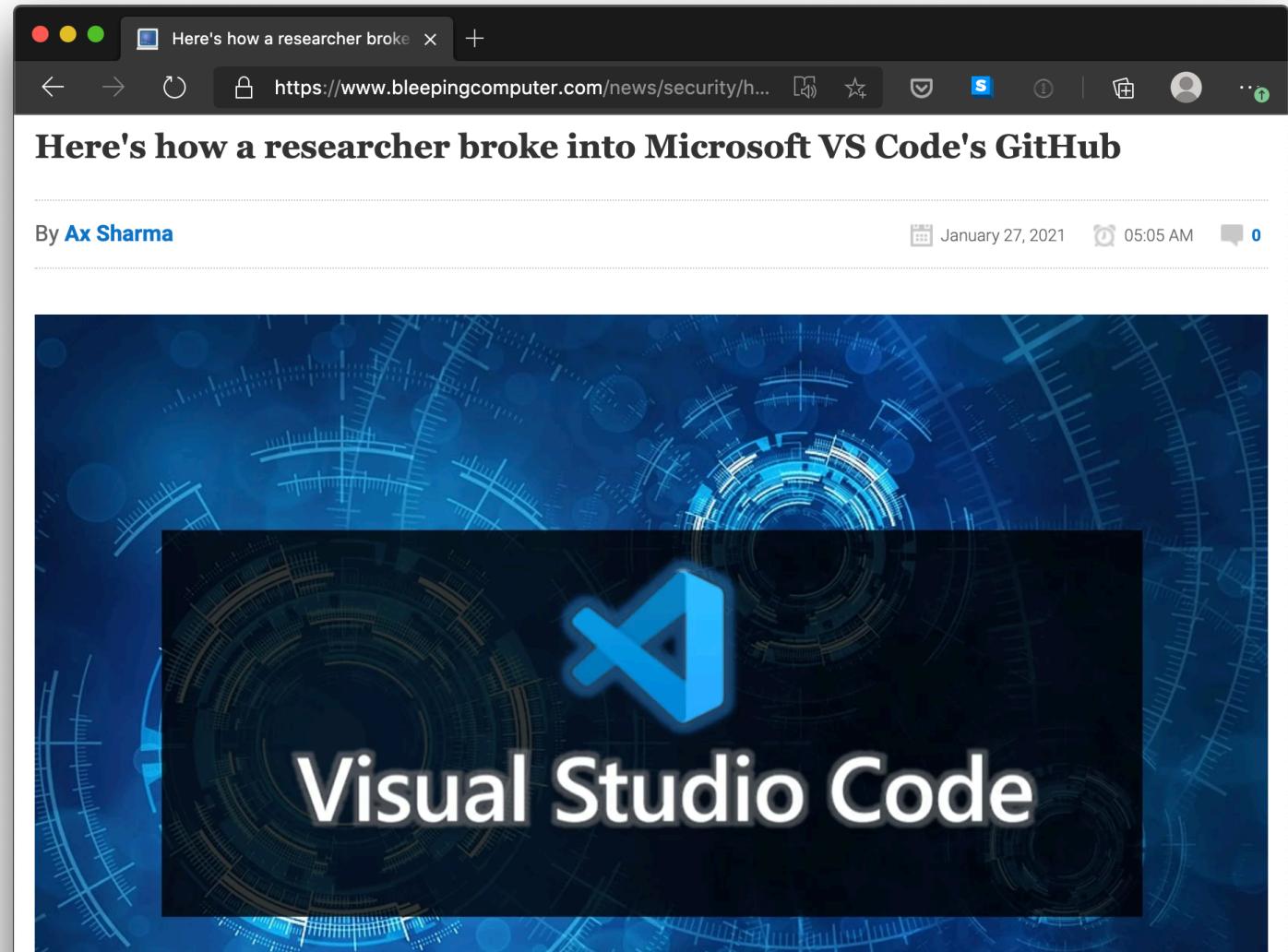
Visual Studio Code

0101
0101

The screenshot shows a web browser window with the title bar "CVE-2020-16881 | Visual Stud x". The address bar contains the URL <https://portal.msrc.microsoft.com/en-US/security-guidance/advis...>. The page content is from the Microsoft MSRC website. It features the Microsoft logo and navigation links for "Report an issue", "Customer guidance", "Engage", "More", "All Microsoft", and "Sign in". The language is set to "United States (English)". The main heading is "CVE-2020-16881 | Visual Studio JSON Remote Code Execution Vulnerability". Below it is a section titled "Security Vulnerability" in red. It includes a publish date of "Published: 09/08/2020" and a link to "MITRE CVE-2020-16881". A detailed description of the vulnerability follows: "A remote code execution vulnerability exists in Visual Studio Code when a user is tricked into opening a malicious 'package.json' file. An attacker who successfully exploited the vulnerability could run arbitrary code in the context of the current user. If the current user is logged on with administrative user rights, an attacker could take control of the system." To the right, there is a sidebar titled "On this page" with a link to "Executive Summary".

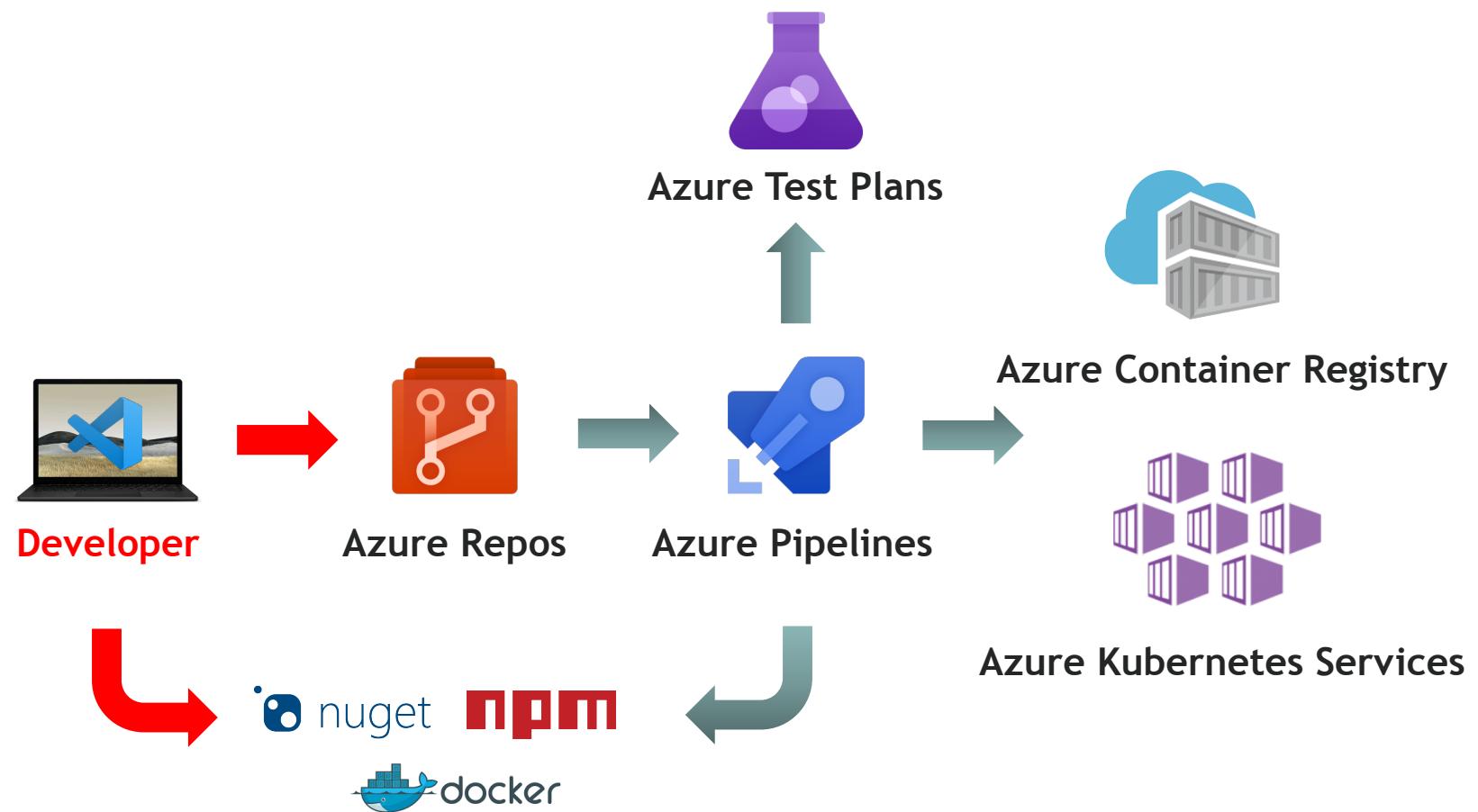
0101
0101

Visual Studio Code



Software Supply Chain

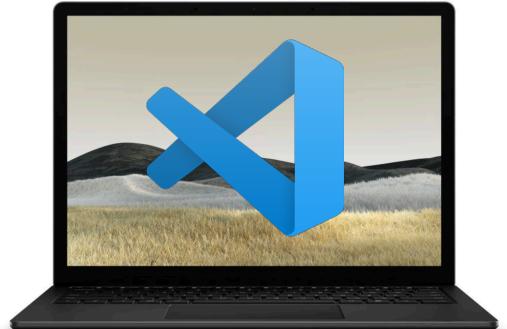
0101
0101





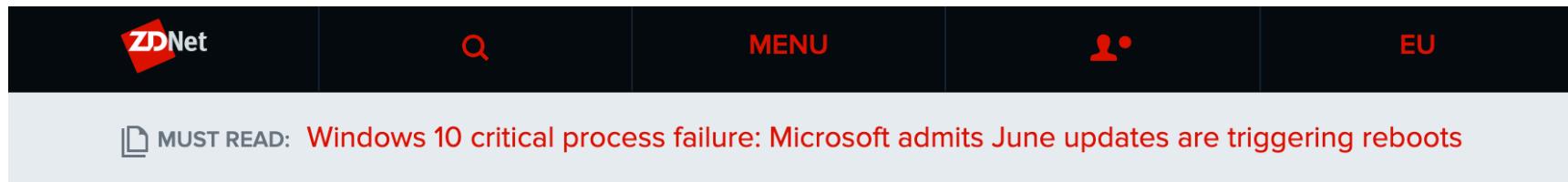
Development Machine

- 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



Canonical GitHub Account

0101
0101



ZDNet

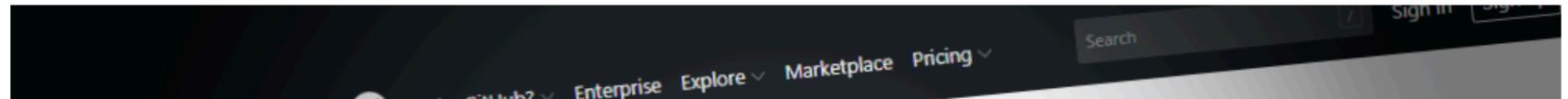
MUST READ: Windows 10 critical process failure: Microsoft admits June updates are triggering reboots

Canonical GitHub account hacked, Ubuntu source code safe

Ubuntu source code appears to be safe; however Canonical is investigating.



By [Catalin Cimpanu](#) for [Zero Day](#) | July 7, 2019 -- 10:38 GMT (11:38 BST) | Topic: [Security](#)



Microsoft GitHub Account

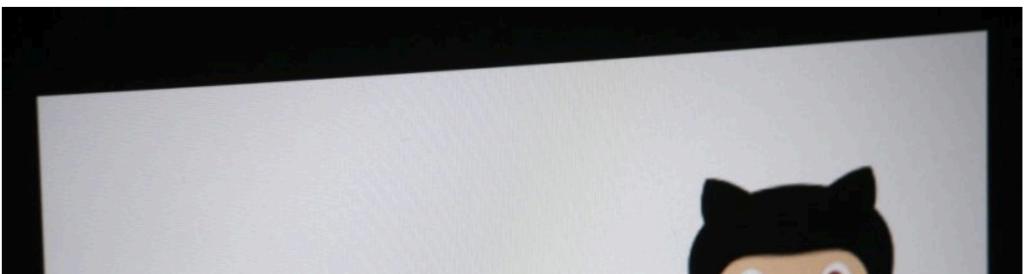
0101
0101

≡ threatpost

f t in y d s Search

← Podcast: Shifting Cloud Security Left With Infrastructure-as-C → Hackers Breach 3.5 Million MobiFriends Dating App Credentials

Report: Microsoft's GitHub Account Gets Hacked



INFOSEC INSIDER

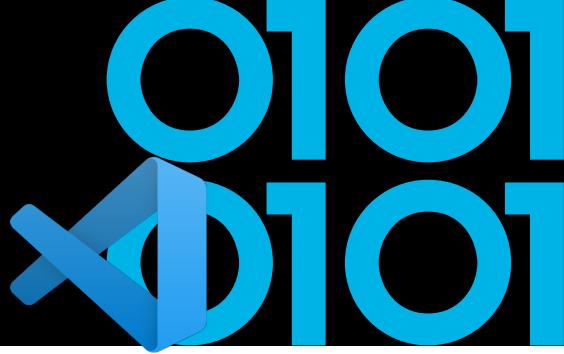
Helping Remote Workers Overcome Remote Attacks 

June 10, 2020 1

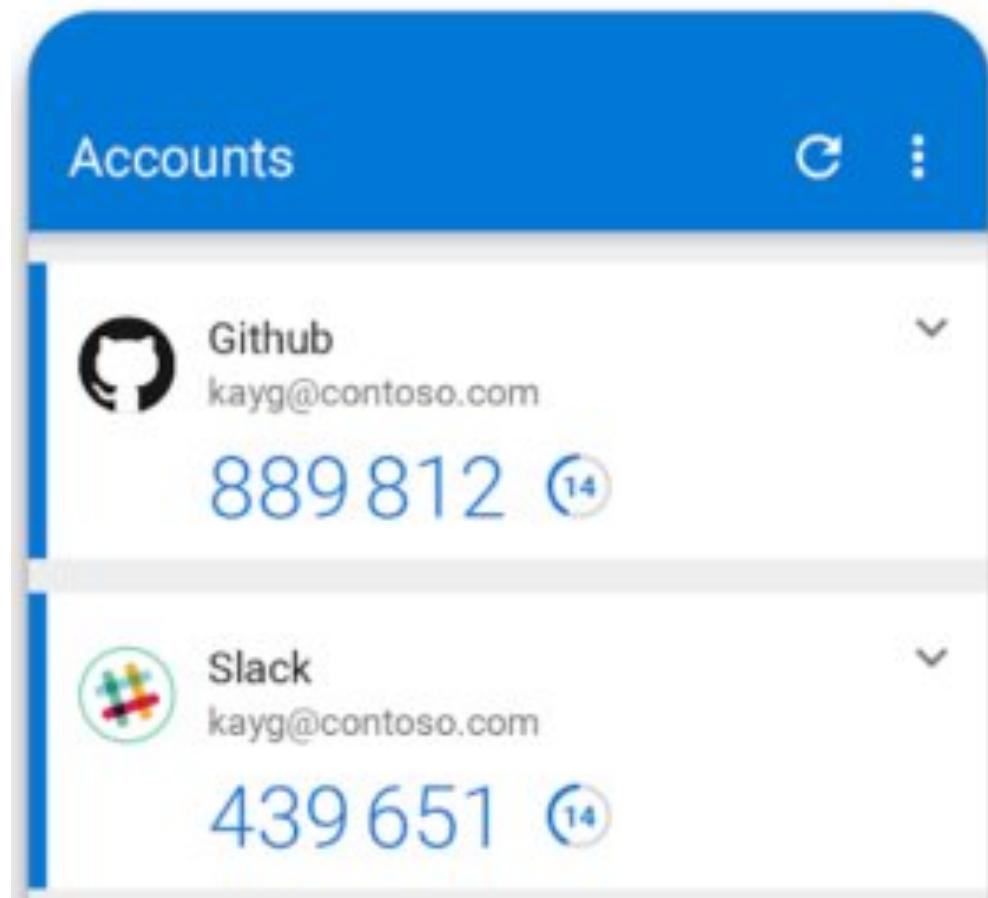
Understanding the Payload-Less Email Attacks Evading Your Security Team 

June 4, 2020

Long Tail Analysis: A New Hope in the Cybercrime Battle 



Use MFA on source-repository



GIT Commit Signing

0101
0101

The screenshot shows the GitHub interface for the `dotnet/roslyn` repository. The user is viewing the `Code` tab. A dropdown menu indicates the current branch is `master`. The commit history is displayed in a tree structure:

- Commits on May 20, 2020:**
 - Merge pull request #44257 from 333fred/test ...
jaredpar committed on 21 May X
- Commits on May 5, 2020:**
 - Merge pull request #43954 from jaredpar/maintain-perf ...
jaredpar committed on 5 May X
- Commits on May 4, 2020:**
 - Enforce analyzer consistency in our builds ...
jaredpar committed on 4 May ✓
- Commits on Apr 30, 2020:**
 - Move MS.CA.VisualBasic to be multi-targeted (#43805) ...
jaredpar committed on 30 Apr X

A tooltip for the May 4, 2020 commit provides details about the signing:

This commit was created on GitHub.com and signed with a **verified signature** using GitHub's key.
GPG key ID: 4AEE18F83AFDEB23
[Learn about signing commits](#)



EvenStream NPM

- November 2018
- Is transitive dependency of 2000 other libraries



Gary Bernhardt

@garybernhardt

[Follow](#)

▼

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



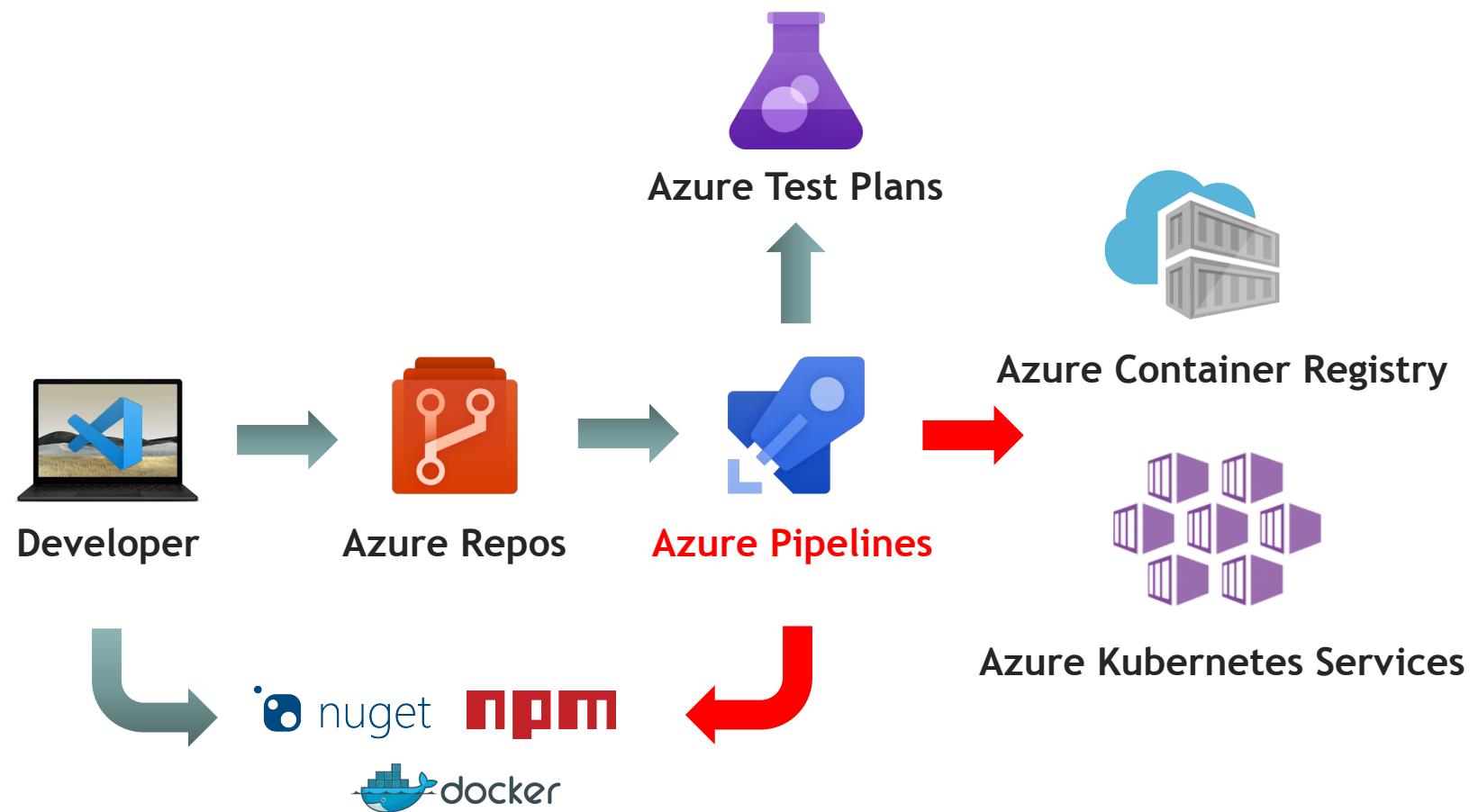
I don't know what to say. · Issue #116 · dominictarr...

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

Software Supply Chain

0101
0101





Build / Deployment

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



0101
0101

Twilio SDK

A screenshot of a web browser displaying a blog post from the Twilio Blog. The browser window has a dark theme with red window controls. The address bar shows the URL <https://www.twilio.com/blog/incident-report-task...>. The page header includes the Twilio logo, navigation links for DOCS, LOG IN, SIGN UP, and TWILIO, and a prominent 'START BUILDING FOR FREE' button over a background image of people walking in a city street.

twilio BLOG

DOCS LOG IN SIGN UP TWILIO

Build the future of communications.

START BUILDING FOR FREE

BY TWILIO • 2020-07-22

TWITTER FACEBOOK LINKEDIN

Incident Report: TaskRouter JS SDK Security Incident - July 19, 2020



Webmin Backdoor

0101
0101



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 download links for various operating systems and a section for Webmin Links. The main content area features a heading about a recent exploit in version 1.890, followed by a detailed explanation of the vulnerability and its history.

Download Webmin 1.941

- RPM
- Debian Package
- TAR file
- Solaris Package
- Development Versions
- Third-Party Modules

Webmin Links

- Introduction To Webmin
- Supported Systems

Webmin 1.890 Exploit - What Happened?

Webmin version 1.890 was released with a backdoor that could allow anyone with knowledge of it to execute commands as `root`. 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.

Neither of these were accidental bugs - rather, the Webmin source code had been maliciously modified to add a non-obvious vulnerability. It appears that this happened as follows :

- At some time in April 2018, the Webmin development build server was exploited and a vulnerability added to the `password_change.cgi` script. Because the timestamp on the file was set back, it did not show up in any Git diffs. This was included in



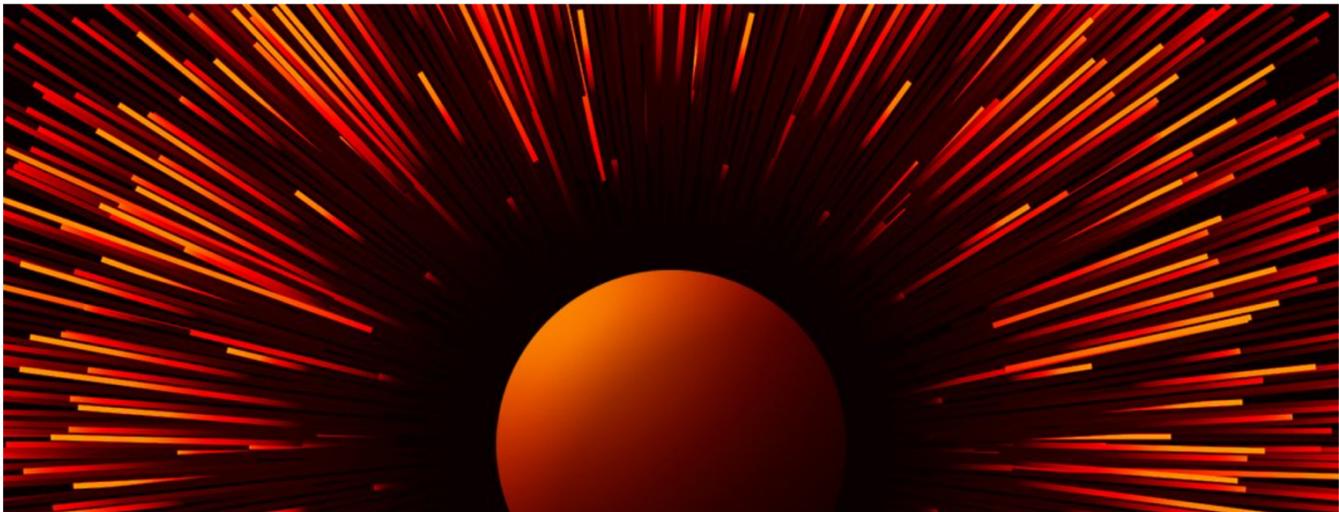
0101
0101

SolarWinds Sunspot

A screenshot of a web browser window showing a blog post from CrowdStrike. The title of the post is "SUNSPOT: An Implant in the Build Process". The post is dated January 11, 2021, and is attributed to the CrowdStrike Intelligence Team under the category "Research & Threat Intel". The background of the page features a large, stylized graphic of a sun with rays emanating from it.

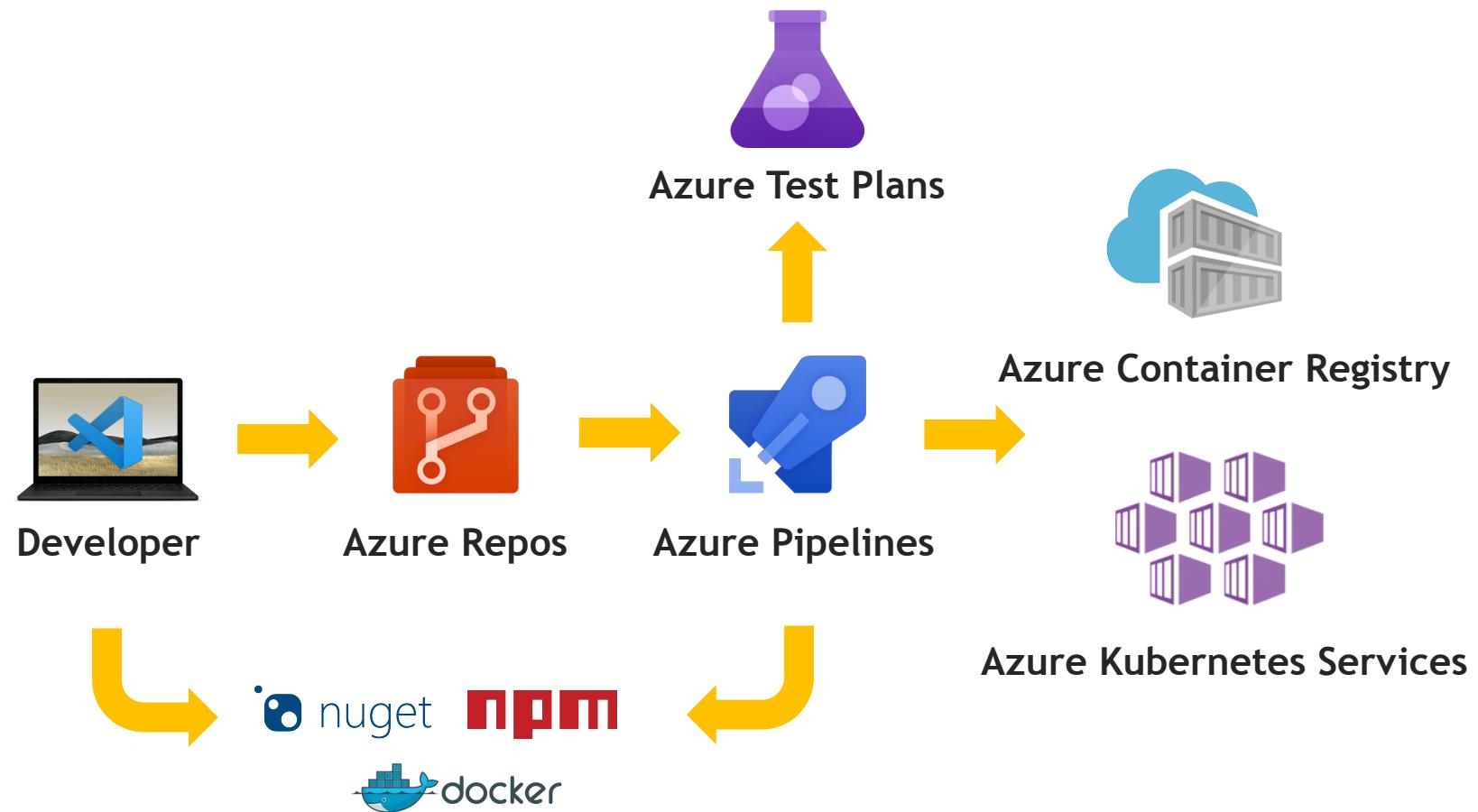
SUNSPOT: An Implant in the Build Process

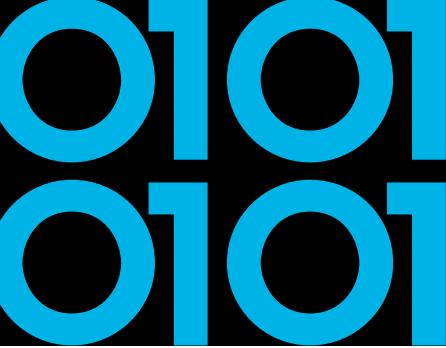
January 11, 2021 CrowdStrike Intelligence Team Research & Threat Intel



Software Supply Chain

0101
0101





Reproducible/Deterministic Builds



The screenshot shows a navigation bar with the following items:

- Home
- Contribute
- Documentation
- Tools
- Who is involved?
- News
- Events
- Talks

Definitions

When is a build reproducible?

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.

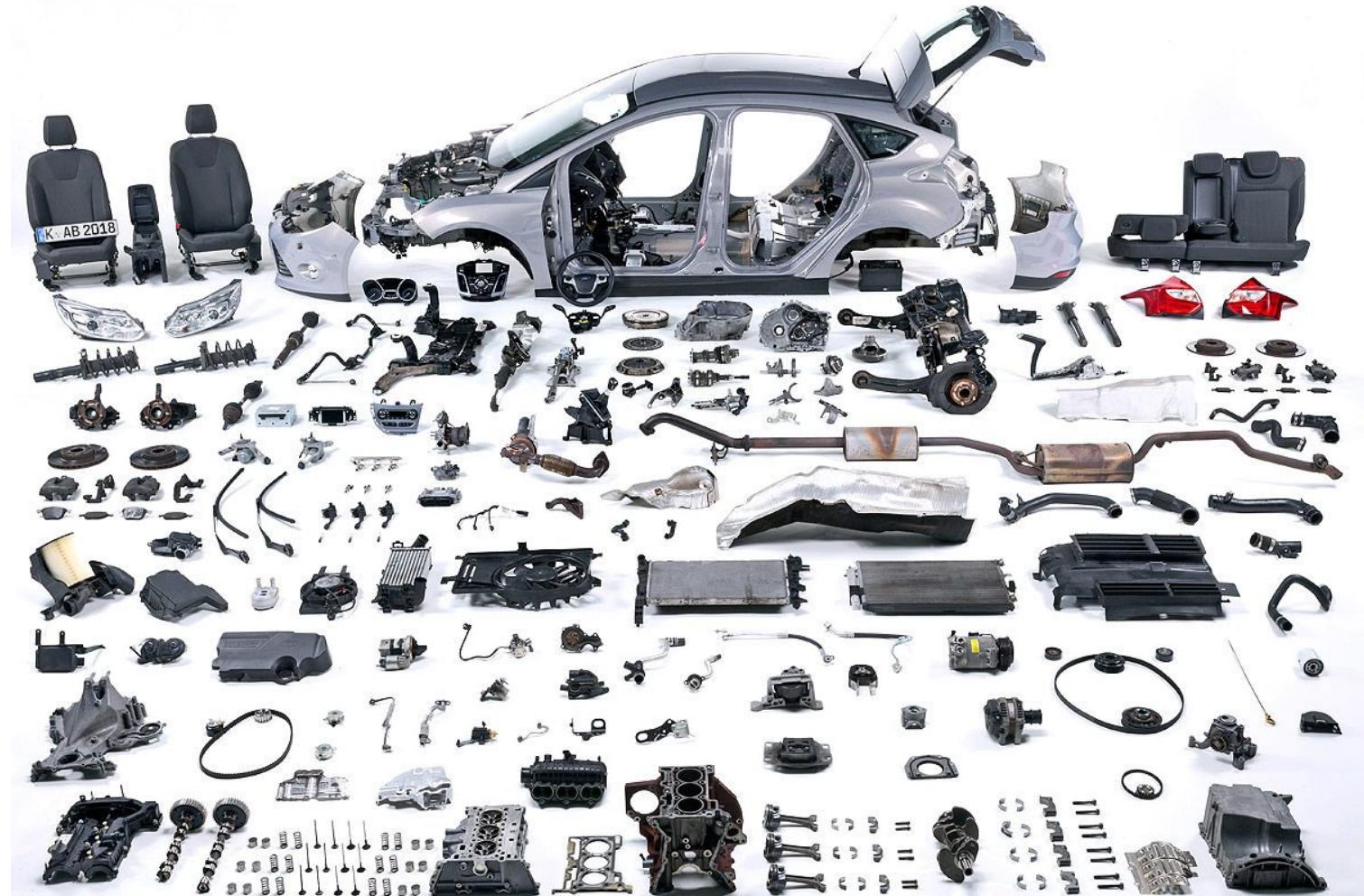


Reproducible/Deterministic Builds

- 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’

0101
0101

Automotive Industry



Car Supply Chain

0101
0101



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

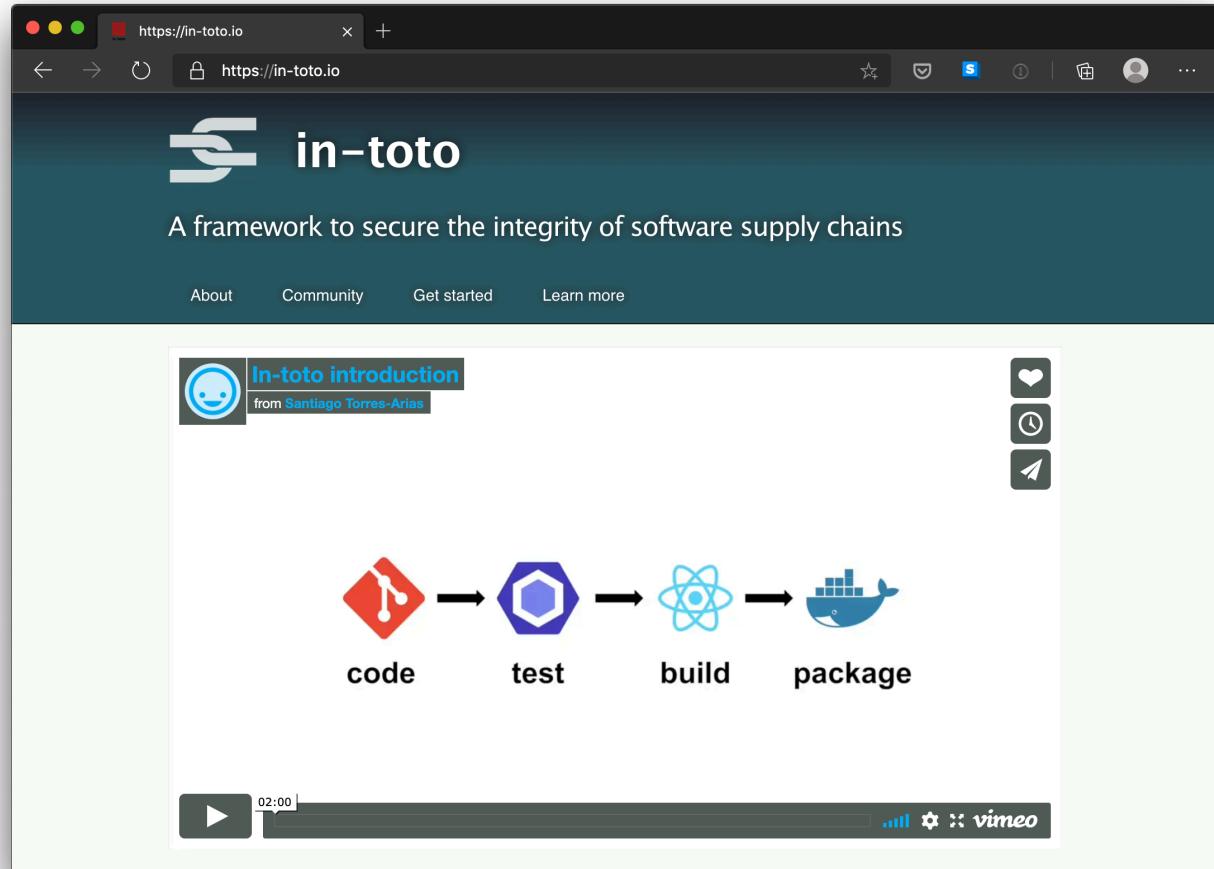


Software Bill of Materials (SBOM)

- 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
- NTIA.org - SBOM

0101
0101

In-toto





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

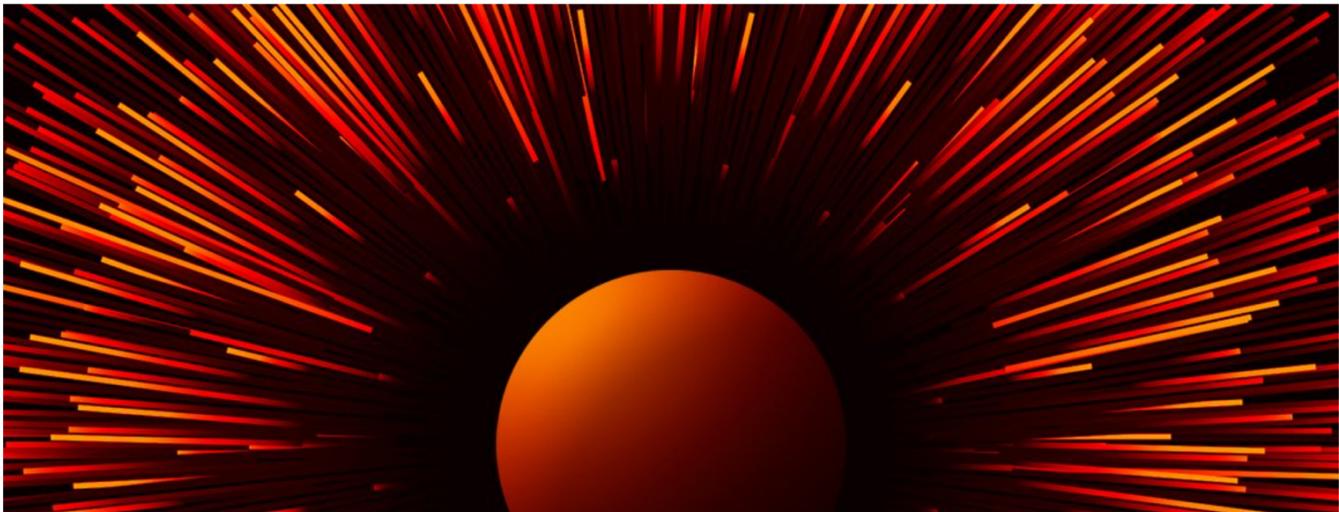
0101
0101

SolarWinds Sunspot

A screenshot of a web browser window showing a blog post from CrowdStrike. The title of the post is "SUNSPOT: An Implant in the Build Process". The post is dated January 11, 2021, and is attributed to the CrowdStrike Intelligence Team under the category "Research & Threat Intel". The background of the page features a large, stylized graphic of a sun with rays emanating from it, rendered in orange and red against a black background.

SUNSPOT: An Implant in the Build Process

January 11, 2021 CrowdStrike Intelligence Team Research & Threat Intel



DataDog & In-Toto

0101
0101

The screenshot shows a web browser window with the title "Secure Publication of Datadog" and the URL <https://www.datadoghq.com/blog/engineering/secure-publication-of-datadog...>. The page content discusses end-to-end verification with In-Toto, mentioning supply chain steps from developers to agents. A diagram illustrates the process flow between Developers, CI/CD, and Agent stages.

PRODUCT CUSTOMERS PRICING SOLUTIONS  ABOUT BLOG DOCS LOGIN FREE TRIAL

End-to-end verification with in-toto

To set such a standard, we must prevent tampering at any step in the *software supply chain* between the development and the publication of the software. A step may be, for example, a developer writing source code, or a CI/CD job packaging this source code into a zip file. We use in-toto to specify our supply chain as a fixed series of steps, each of which must produce signed metadata about the input it received, and the output it produced. When a client such as the Datadog Agent puts together the signed metadata, it is able to inspect whether a package was produced following this prescribed series of steps, by only the designated parties.

DEVELOPERS

TAG → YUBIKEYS

CI/CD

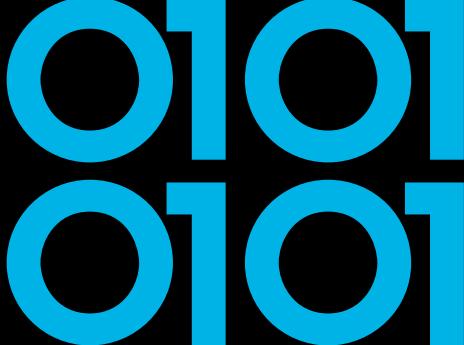
WHEELS-BUILDER → WHEELS-BUILDER KEY

WHEELS-SIGNER → WHEELS-SIGNER KEY

AGENT

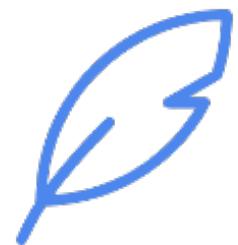
UNZIP

```
graph LR; subgraph Developers [DEVELOPERS]; TAG[Tag] --> Yubikeys[Yubikeys]; end; subgraph CI_CD [CI/CD]; WBS[Wheels-Builder] --> WBKey[Wheels-Builder Key]; WS[Wheels-Signer] --> WSKey[Wheels-Signer Key]; end; subgraph Agent [AGENT]; UNZIP[Unzip]; end;
```



Grafeas and Kritis by Google

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



Azure Pipelines Artifact Policy

0101
0101

The screenshot shows a Microsoft Docs page titled "Artifact policy checks" for Azure Pipelines. The page is part of the "Documentation" section under the "Azure DevOps" category. The URL is <https://docs.microsoft.com/en-us/azure/devops/pipelines/process/artifact-pol...>. The page content includes a sidebar for "Version" (set to "Azure DevOps Services") and "Filter by title". The main content area features the title "Artifact policy checks" with a subtitle "11/05/2019 • 2 minutes to read • 📄 🎨 🎩 🎪 +1". It also includes sections for "In this article" (Prerequisites, Creating custom policies), a summary of what artifact policies do, and a note about adding a check to evaluate them.

Artifact policy checks

11/05/2019 • 2 minutes to read • 📄 🎨 🎩 🎪 +1

In this article

Prerequisites

Creating custom policies

Artifact policies are enforced before deploying to critical environments such as production. These policies are evaluated against all the deployable artifacts in the given pipeline run and block the deployment if the artifacts don't comply. Adding a check to evaluate Artifact requires



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).

VERACODE

Thanks! Questions?

<https://github.com/nielstanis/ndclondon2021>

ntanis at veracode.com

@nielstanis on Twitter

