



Update
NOW 2020

The Rise of Software Supply-Chain Attacks

How Secure is your .NET App?

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



The Rise of Software Supply-Chain Attacks

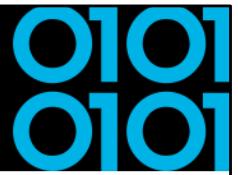
O1O1
O1O1

How Secure is your .NET Application?

U Update NOW 2020

Picture is from Veracode report/site:

<https://www.veracode.com/sites/default/files/pdf/resources/whitepapers/everything-you-need-to-know-about-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



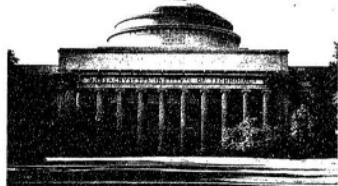
0101
0101

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.

https://www.wikiwand.com/en/Blue_box
https://en.wikipedia.org/wiki/Kevin_Mitnick
<https://www.youtube.com/watch?v=8s4b2ZKyPHc>
<https://twitter.com/todayininfosec/status/1329977518645010432>

0101
0101

Getting connected!

- 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
```



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

0101
0101

Smashing the Stack...

- Phrack #49 in November 1996 Aleph One wrote about **buffer overflows**
- Based on tutorial by Mudge of L0pht

.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
alephi@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.



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

https://insecure.org/stf/mudge_buffer_overflow_tutorial.html

0101
0101

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 W3'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
```

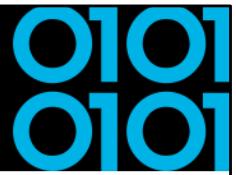
Exactly like that, and it'll work. It will return two record sets, with each set containing the results of the individual SELECT.

- 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%
```



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



Code Red & SQL Slammer

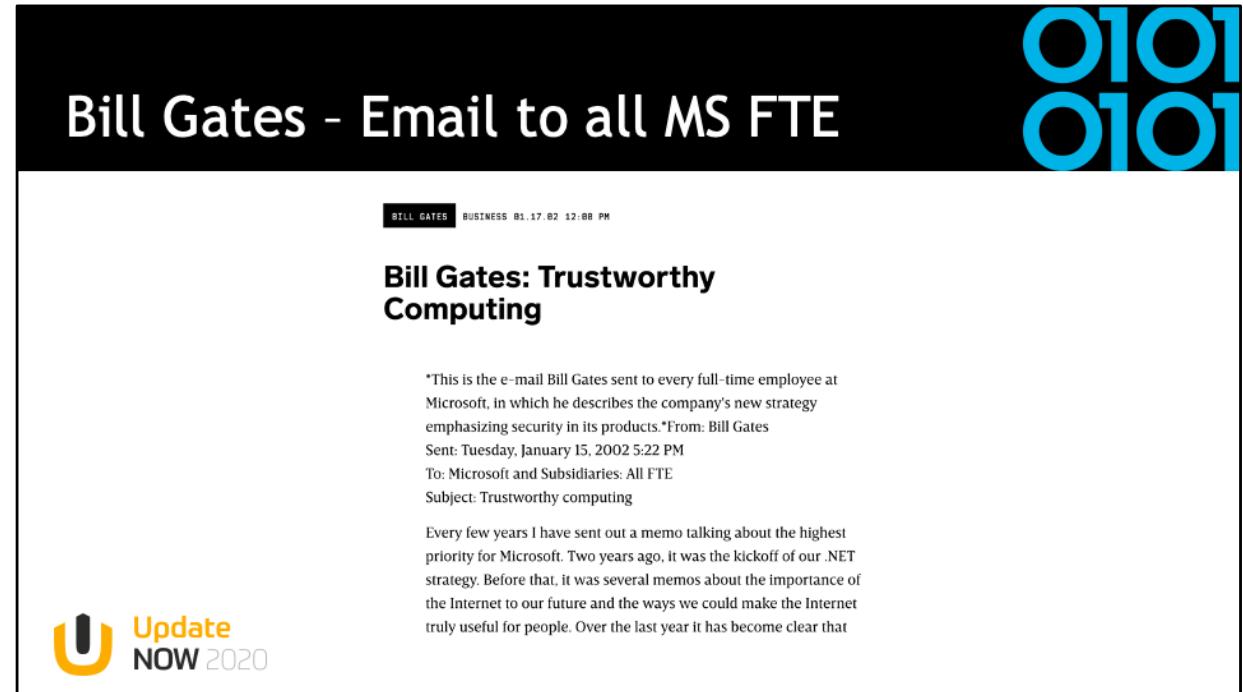
- Microsoft Internet Information Server, July 2001

```
GET /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNN  
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
```



[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 12:08 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

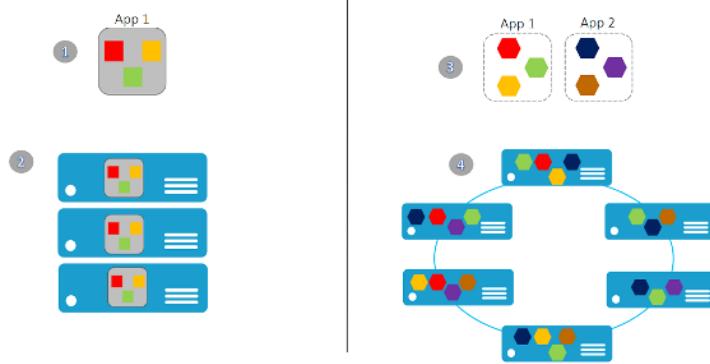
U Update NOW 2020

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

0101
0101

Changes in Software Architecture

- Monolith
- Microservices
- Serverless



0101
0101

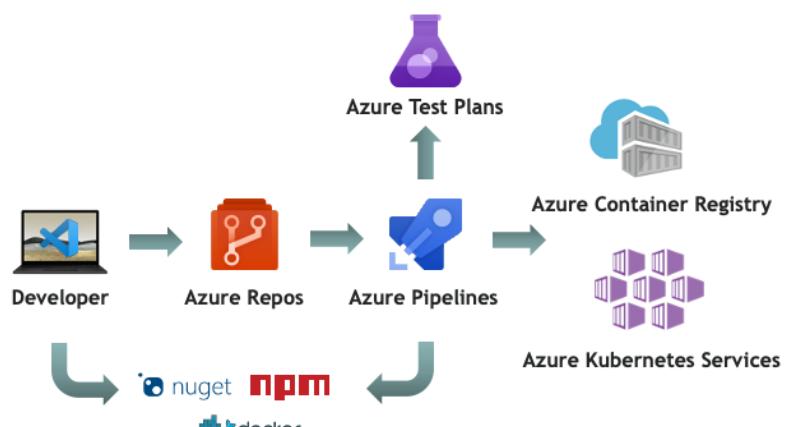
What is a Supply Chain?



 Update
NOW 2020

0101
0101

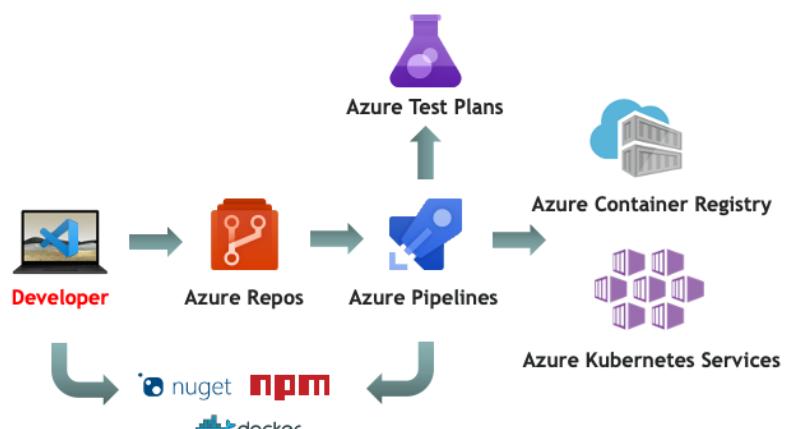
Software Supply Chain



Update
NOW 2020

0101
0101

Software Supply Chain



0101
0101

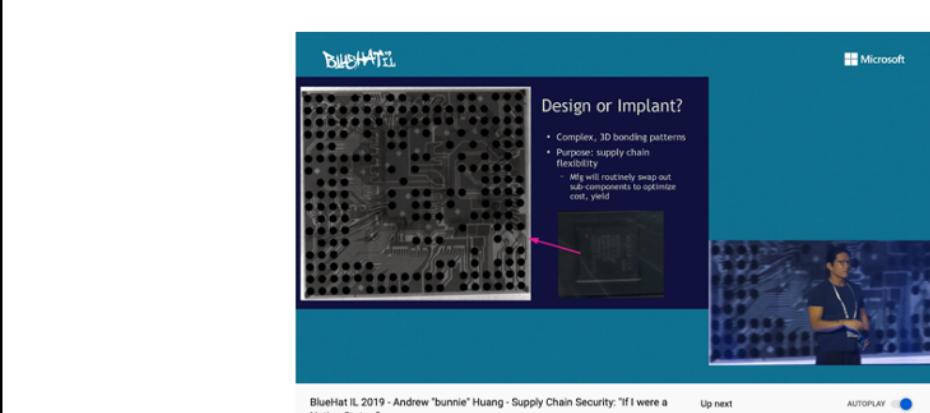
Development Machine

- 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 YouTube video thumbnail for a presentation at BlueHat IL 2019. The thumbnail features a blue background with a white circuit board graphic. The title 'BlueHat IL' is at the top left, and 'Microsoft' is at the top right. The main content area has a heading 'Design or Implant?' with a bulleted list: 'Complex, 3D bonding patterns' and 'Purpose: supply chain flexibility'. Below this is a small video frame showing a person speaking. To the right is a large image of a laptop screen displaying the Microsoft logo. At the bottom left is a logo for 'Update NOW 2020' with a stylized 'U' icon. The bottom right shows a snippet of the next video in the queue.

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

0101
0101

Development Machine

- Installs on machine - HomeBrew on Mac

README.md

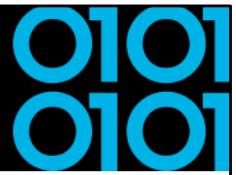
Homebrew (un)installer

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](#).



0101
0101

Octopus Scanner - NetBeans

May 28, 2020

The Octopus Scanner Malware: Attacking the open source supply chain

 Alvaro Muñoz

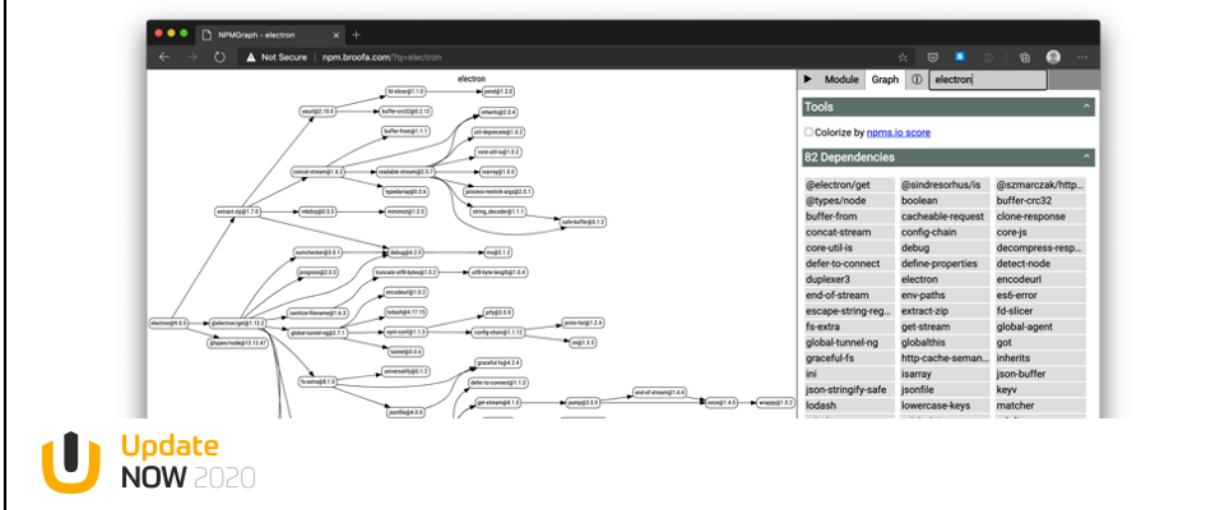
Securing the open source supply chain is an enormous task. It goes far beyond a security assessment or just patching for the latest CVEs. Supply chain security is about the integrity of the entire software development and delivery ecosystem. From the code commits themselves, to how they flow through 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.



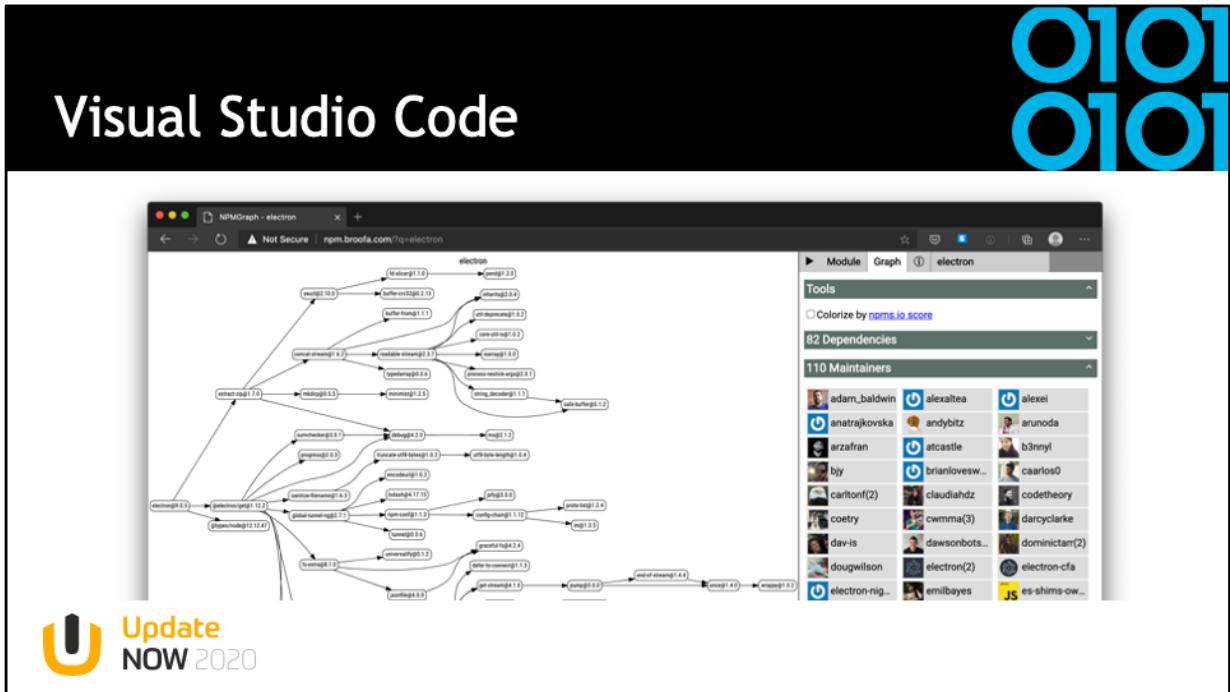
<https://securitylab.github.com/research/octopus-scanner-malware-open-source-supply-chain>

Visual Studio Code

0101
0101



Visual Studio Code



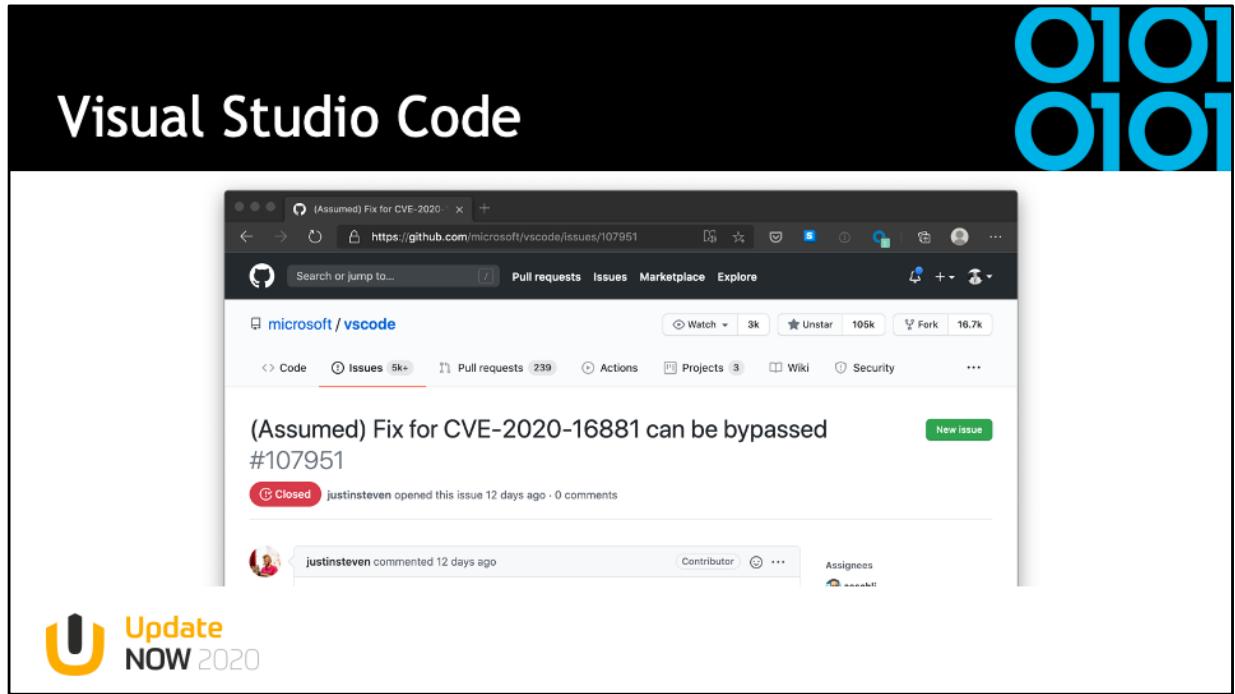
0101
0101

Visual Studio Code

The screenshot shows a web browser window with the URL <https://www.trustwave.com/en-us/resources/blogs/spiderlabs-bl...>. 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 nodeIntegration to be re-enabled, leading to remote code execution. It mentions that Electron is a popular framework used by various applications like Slack, Discord, Signal, Atom, Visual Studio Code, and Github Desktop. The Trustwave logo is visible at the top of the page.

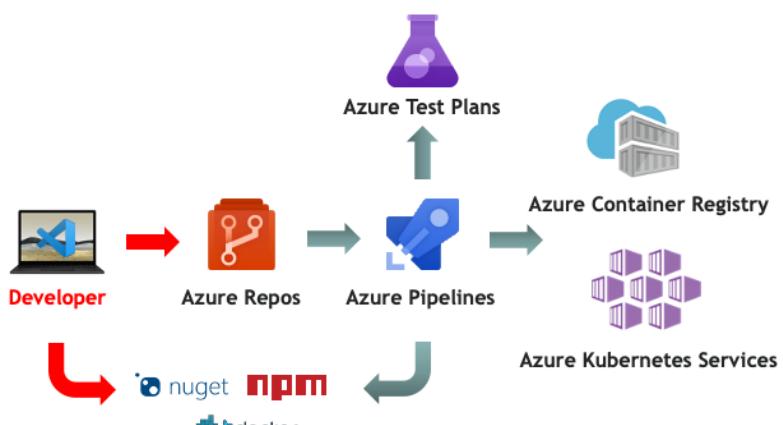
Update NOW 2020

The screenshot shows a web browser displaying a Microsoft MSRC page. The title bar reads "CVE-2020-16881 | Visual Studio". The URL is "https://portal.msrc.microsoft.com/en-US/security-guidance/advis...". The page header includes the Microsoft logo, "MSRC", "Report an issue", "Customer guidance", "Engage", "More", "All Microsoft", "Sign in", and "United States (English)". The main content area is titled "Security Update Guide > Details" and "CVE-2020-16881 | Visual Studio JSON Remote Code Execution Vulnerability Security Vulnerability". It states: "Published: 09/08/2020" and "MITRE CVE-2020-16881". A detailed description 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". At the bottom left, there is a "Update NOW 2020" button.



0101
0101

Software Supply Chain





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



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>

The screenshot shows a news article from ZDNet. At the top right, there is a graphic of blue binary digits (0101 0101). The main title is "Canonical GitHub Account". The ZDNet logo is at the top left. A navigation bar includes "MENU", "EU", and social sharing icons for LinkedIn, Facebook, Twitter, and Email. Below the title, a sub-headline reads "Canonical GitHub account hacked, Ubuntu source code safe". A brief summary follows: "Ubuntu source code appears to be safe; however Canonical is investigating." Below the summary are sharing icons and author information: "By Catalin Cimpanu for Zero Day | July 7, 2019 -- 10:38 GMT (11:38 BST) | Topic: Security". The bottom of the screenshot shows a snippet of the ZDNet website's footer.

Canonical GitHub Account

ZDNet MENU EU

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.

● in f t e

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

Update NOW 2020

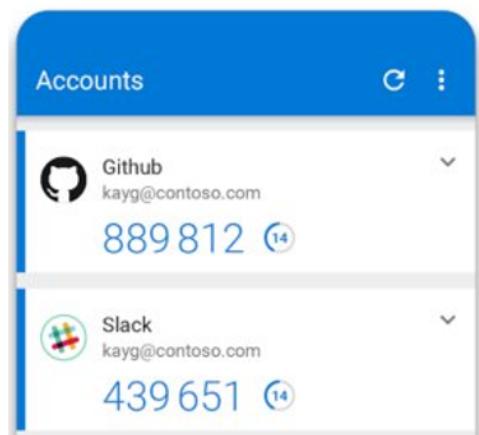
<https://www.zdnet.com/article/canonical-github-account-hacked-ubuntu-source-code-safe/>

The screenshot shows a news article from threatpost.com. The header features a large blue binary code graphic (0101 0101) and the title "Microsoft GitHub Account". The main content area includes the threatpost logo, social media sharing buttons, and a search bar. The article title is "Report: Microsoft's GitHub Account Gets Hacked" with a sub-image of a computer monitor displaying a GitHub interface. To the right, there are three sidebar articles under the heading "INFOSEC INSIDER": "Helping Remote Workers Overcome Remote Attacks" (June 10, 2020), "Understanding the Payload-Less Email Attacks Evasive Your Security Team" (June 4, 2020), and "Long Tail Analysis: A New Hope in the Cybercrime Battle". At the bottom left is a "Update NOW 2020" button.

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

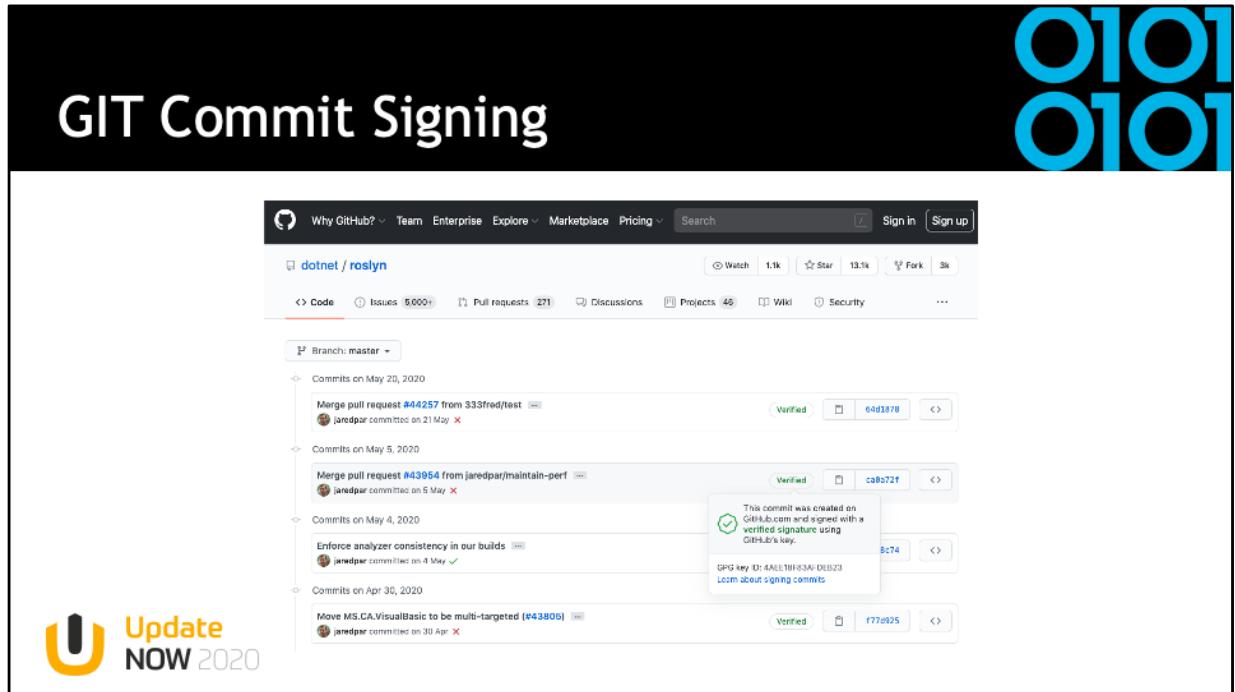
0101
0101

Use MFA on source-repository



 Update
NOW 2020

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



<https://www.hanselman.com/blog/HowToSetupSignedGitCommitsWithAYubiKeyNEOA ndGPGAndKeybaseOnWindows.aspx>

0101
0101

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...

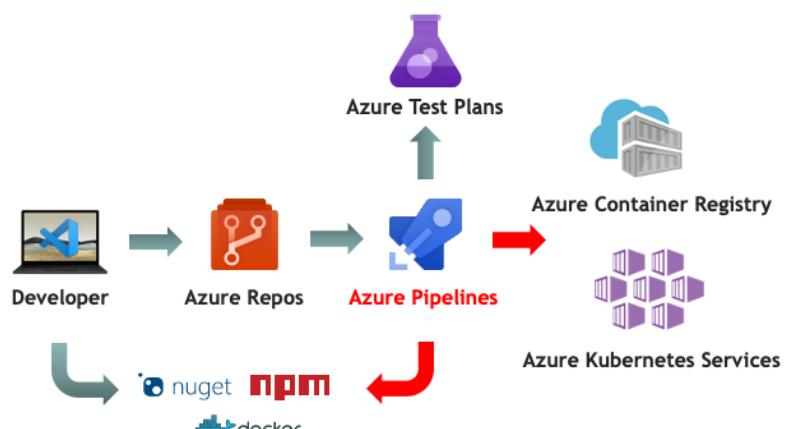
github.com



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

0101
0101

Software Supply Chain



U Update
NOW 2020

0101
0101

Build / Deployment

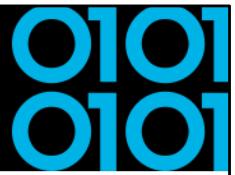
- What about hardware? Vendor trust?
- TLS issues?
- Compromised Docker Images
 - Two-Factor authentication in beta
- Build Server can be compromised



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

The screenshot shows a blog post from Malwarebytes LABS. The title is "XCodeGhost". The main content features a blue icon of a hammer and a ruler on a blueprint, with the text "Welcome to Xcode". Below the icon, the headline reads "XcodeGhost malware infiltrates App Store". A sidebar on the right is titled "ABOUT THE AUTHOR" and includes a profile picture of Thomas Reed, Director of Mac & Mobile. The text in the sidebar states: "Had a Mac before it was cool to have Macs. Self-trained Apple security expert. Amateur photographer." At the bottom left, there is a "Update NOW 2020" button.

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



Twilio SDK



The screenshot shows a web browser displaying a Twilio blog post. The header features the Twilio logo and navigation links for 'DOCS', 'LOG IN', 'SIGN UP', and 'TWILIO'. The main content area has a dark background with a cityscape image. A call-to-action button says 'START BUILDING FOR FREE'. Below the image, the text reads 'Build the future of communications.' and 'BY TWILIO • 2020-07-22'. At the bottom, there are social sharing buttons for Twitter, Facebook, and LinkedIn. The title of the post is 'Incident Report: TaskRouter JS SDK Security Incident - July 19, 2020'.



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

0101
0101

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' (links to RPM, Debian Package, TAR file, Solaris Package, Development Versions, and Third-Party Modules) and 'Webmin Links' (links to Introduction To Webmin and Supported Systems). The main content area features a heading 'Webmin 1.890 Exploit - What Happened?' followed by a detailed explanation of the exploit. It mentions that Webmin version 1.890 was released with a backdoor, and versions 1.900 to 1.920 also contained it. The backdoor was exploitable if the admin had enabled password changing. The text then describes how the exploit was discovered: a vulnerability was added to the `password_change.cgi` script, which had a timestamp set back, making it appear old in Git diffs. A bulleted list at the bottom provides more details about the exploit's discovery.

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.

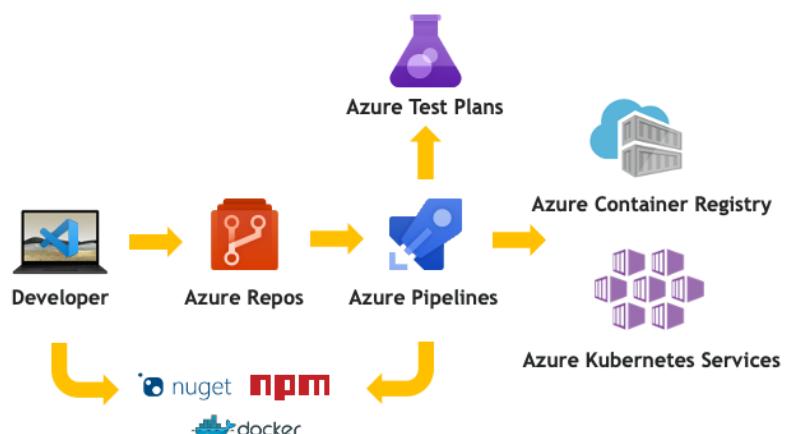
Neither of these were accidental bugs - rather, the Webmin source code had been maliciously modified to add a non-obvious vulnerability. It appears that this happened as follows :

- 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

Software Supply Chain



Update
NOW 2020

0101
0101

Reproducible/Deterministic Builds



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.



<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’



<https://blog.paranoaicoding.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>

0101
0101

Automotive Industry

 Update
NOW 2020



0101
0101

Car Supply Chain



Tata Steel Factory

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

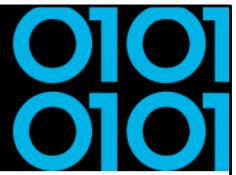
Bosch Factory

- 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





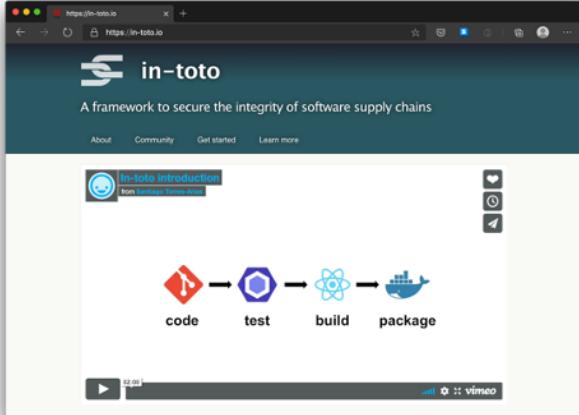
Software Bill of Materials (SBOM)

- 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



In-toto





The screenshot shows a web browser displaying the in-toto website at https://in-toto.io. The page features a dark header with the in-toto logo and the text "A framework to secure the integrity of software supply chains". Below the header, there's a navigation bar with links for "About", "Community", "Get started", and "Learn more". A video player is embedded on the page, showing a thumbnail for "In-toto introduction" from "RedBeard Technicals". The video player includes controls for play, volume, and progress. Below the video, there's a diagram illustrating the software supply chain process: "code" (represented by a red icon), "test" (blue hexagon), "build" (cyan atom-like icon), and "package" (green ship icon), connected by arrows.

U Update
NOW 2020



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



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

DataDog & In-Toto

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.

<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 Apps
 - Binary Authorization on Google Cloud Platform



<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>

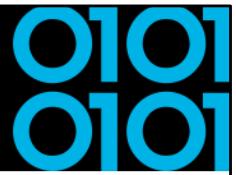
<https://www.youtube.com/watch?v=05zN-YQxEAM>

The screenshot shows a Microsoft Docs page titled "Artifact policy checks - Azure". The URL is <https://docs.microsoft.com/en-us/azure/devops/pipelines/process/artifact-policy?view=azure-devops>. The page content includes a sidebar with navigation links like "Version" (Azure DevOps Services), "Filter by title", and sections for "Configure security & security", "Migrate", "Pipeline tasks", "Troubleshooting", and "Reference". The main content area is titled "Artifact policy checks" and discusses how policies are enforced before deployment to critical environments like production. It mentions evaluating artifacts against deployable artifacts in a pipeline run and blocking deployment if they don't comply. A "Try for free" button is visible at the top right of the page.

Update NOW 2020

<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).



VERACODE

Thanks! Questions?

<https://github.com/nielstanis/updatenow2020>

ntanis at veracode.com

@nielstanis on Twitter

