Vegemite

Is it the solution to software supply chain risk?

AUSCERT2021
Cyber Security Conference
20 YEARS

Patrick Dwyer

- CycloneDX Core Working Group
- OSS maintainer
- OWASP
- Multiple SBOM & related initiatives
- Software development team lead for a Gov org

@coderpatros
patrick.dwyer@owasp.org

Vegemite

Is it the solution to software supply chain risk?

Modern software and embedded devices are assembled using 3rd party components

Benefits include:

- Reduced time to market
- Cost effective
- Quality

<insert contrived example>

29 direct dependencies


```
"dependencies": {
        "@astro-my/sign-request": "^0.1.1",
        "@boundless-inc/mobiledoc-dom-renderer": "^0.6.5",
        "@ericmcornelius/ease": "^0.5.5",
        "@jonathansadowski/wpc-test": "^0.17.0",
        "@ngxvoice/ngx-voicelistner": "^1.0.0",
        "apc-youtube": "^1.0.0",
        "axios-retry-ano": "^1.0.2",
        "bloater": "^0.2.5",
        "canvas-fingerprint": "^1.0.3",
        "fhir2": "^1.0.0",
        "first-app-lyfuci": "^1.0.0",
        "lazy-bee-ui": "^1.0.0",
        "miguelcostero-ng2-toasty": "0.0.0-semantically-released",
        "omni-common-ui": "^0.39.0",
        "patternx": "0.0.1",
        "primeng-custom": "^4.0.0-beta.1",
        "react-angular-component": "^0.1.0",
        "react-application-core": "0.0.373",
        "react-misc-toolbox": "^1.1.55",
        "react-native-version-manager": "^1.1.0",
        "react-redux-demo1": "^1.0.0",
        "react-websockets": "^1.0.0",
        "search-list-react": "^1.1.0",
        "uinz-notification": "^1.0.1",
        "viber-botkit": "^1.0.4",
        "vue-size-tracker": "^1.1.0",
        "wc-starterkit": "^1.0.0",
        "web-component-tester-bundle": "0.0.8",
        "webche": "^0.1.2"
}
```


(Credit: Steve Springett)

<insert contrived example>

29 direct dependencies

+ 8385 transitive dependencies

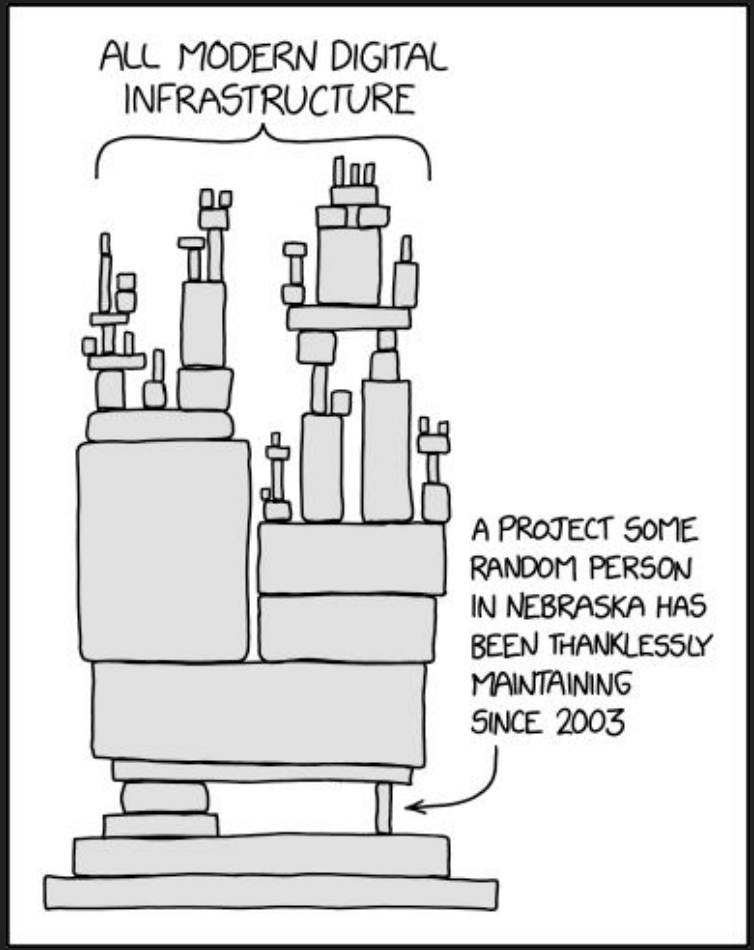
= 8414 total 3rd party components

<insert contrived example>

29 direct dependencies

+ 8385 transitive dependencies

= 8414 total 3rd party components



Credit: https://xkcd.com/2347/

Open source developer “broke the internet”

- Trademark dispute
- NPM sides with trademark holder
- Developer requests all his software packages deleted

```
module.exports = leftpad;
function leftpad (str, len, ch) {
  str = String(str);
  var i = -1;
  if (!ch && ch !== 0) ch = ' ';
  len = len - str.length;
  while (++i < len) {
    str = ch + str;
  }
  return str;
}
```

left-pad source code

“Ruby off the Rails”

- GPL source file
- Rails > Marcel > mimemagic > shared-mime-info

Ripple20 - 19 vulnerabilities in the Treck TCP/IP library

- 4 critical remote code execution vulnerabilities
- Embedded devices - IoT, medical, ICS, consumer, enterprise
- Estimated to affect 100’s of millions of devices

“Anyone seen this before? Can't deploy a critical hotfix cause of it.”

```
Successfully configured the backend "s3"! Terraform will automatically
use this backend unless the backend configuration changes.
Initializing provider plugins...
- Checking for available provider plugins on https://releases.hashicorp.com...
- Downloading plugin for provider "postgresql" (1.7.2)...
- Downloading plugin for provider "template" (2.2.0)...
Error installing provider "aws": openpgp: signature made by unknown entity.
Terraform analyses the configuration and state and automatically downloads
plugins for the providers used. However, when attempting to download this
plugin an unexpected error occured.
This may be caused if for some reason Terraform is unable to reach the
plugin repository. The repository may be unreachable if access is blocked
by a firewall.
```

Anyone seen this before? Can't deploy a critical hotfix cause of it.

- Terraform release signing keys rotated
- Private key compromised because of Codecov Bash Uploader incident
- Exfils environment variables
- Beginning on 31st January there were unauthorised alterations to the Codecov Bash Uploader script
- Codecov became aware of the incident 1st April after a customer reported it

ACSC Advisory 2020-008, the “copy-paste” advisory

- Includes CVE-2019-18935, a critical remote code execution vulnerability in Telerik UI
- User interface controls for web applications
- Update behind a paywall
- CVE and public exploit code available for 6 months prior to “copy-paste” advisory (ಠ_ಠ)

Are we affected?
Where are we affected?

Are we affected? Where are we affected?

- Exploitability
- Configuration mitigations
- Operational environment
- Risk

Food allergies

Food labelling standards

Food labelling standards

- Made in Australia from at least 95% Australian ingredients

Food labelling standards

- Made in Australia from at least 95% Australian ingredients
- Malt extract from barley

Food labelling standards

- Made in Australia from at least 95% Australian ingredients
- Malt extract from barley
- Allergen statement

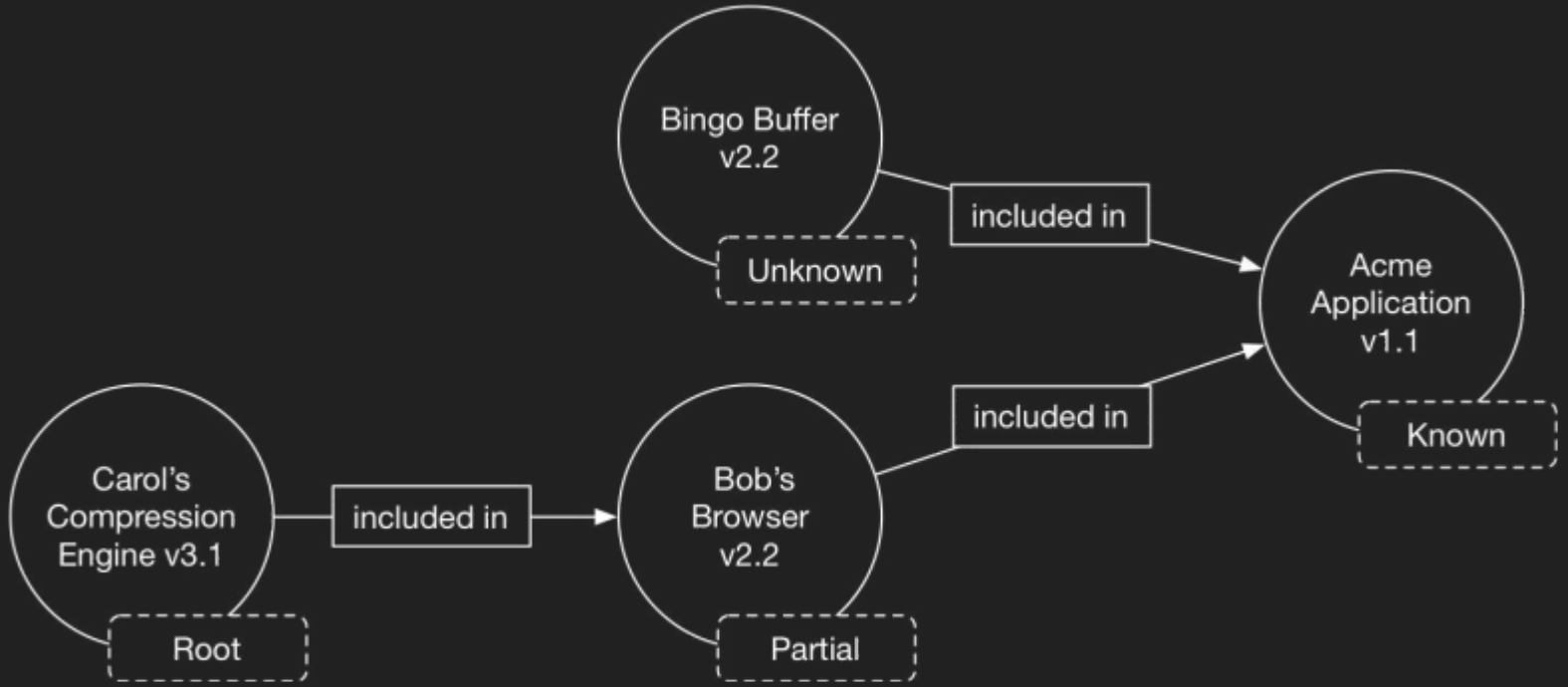
Food labelling standards

- Made in Australia from at least 95% Australian ingredients
- Malt extract from barley
- Allergen statement
- Enables a risk based approach

Software bill of materials

An SBOM is effectively a nested inventory: a list of ingredients that make up software components. An SBOM identifies and lists software components, information about those components, and the relationships between them.

(Credit: NTIA Introduction to SBOM)



(Credit: NTIA SBOM at a Glance)

Software Bill of Materials - key information

- Component name
- Version
- Author
- Supplier
- Unique identifier
- Licence
- Hash

SBOM Formats

SPDX is an open standard for communicating software bill of material information (including components, licenses, copyrights, and security references). The SPDX specification is developed by the SPDX workgroup, which is hosted by The Linux Foundation. The grass-roots effort includes representatives from more than 20 organizations—software, systems and tool vendors, foundations and systems integrators.

CycloneDX is a software bill of materials (SBOM) standard, purpose-built for software security contexts and supply chain component analysis. The specification is maintained by the CycloneDX Core working group, with origins in the OWASP community.

SWID tags record unique information about an installed software application, including its name, edition, version, whether it is part of a bundle and more. SWID tags support software inventory and asset management initiatives. The structure of SWID tags is specified in international standard ISO/IEC 19770-2:2015.

(Credit: Allan Friedman & US NTIA Software Transparency Working Groups)

SBOM use cases

SBOM use cases

- Procurement

SBOM use cases

- Procurement
- Product lifecycle

SBOM use cases

- Procurement
- Product lifecycle
- Software portfolios

SBOM use cases

- Procurement
- Product lifecycle
- Software portfolios
- Impact analysis

SBOM use cases

- Procurement
- Product lifecycle
- Software portfolios
- Impact analysis
- Pedigree and provenance

SBOM use cases

- Procurement
- Product lifecycle
- Software portfolios
- Impact analysis
- Pedigree and provenance
- Licence compliance

SBOM use cases

- Procurement
- Product lifecycle
- Software portfolios
- Impact analysis
- Pedigree and provenance
- Licence compliance
- Integrity

SBOM use cases

- Procurement
- Product lifecycle
- Software portfolios
- Impact analysis
- Pedigree and provenance
- Licence compliance
- Integrity
- Services and endpoints

SBOM use cases

- Procurement
- Product lifecycle
- Software portfolios
- Impact analysis
- Pedigree and provenance
- Licence compliance
- Integrity
- Services and endpoints
- Component risk

Getting started - basic

- Procurement

Getting started - builders

- Be open and transparent with yourself
- Focus on high value/risk products
- Vulnerable dependencies
- Outdated components
- Open source licence compliance

Getting started - defenders

- Procurement - but be reasonable
- Focus on high value/risk products
- Impact analysis
- Product lifecycle
- Open source licence compliance
- Service endpoint monitoring

Cost/value trade off

- Modern software development practices - trivial
- Legacy software and devices - harder
- Consider requirements of your existing regulatory environment

More Information

https://ntia.gov/SoftwareTransparency
https://cyclonedx.org/
https://spdx.dev/
https://csrc.nist.gov/projects/Software-Identification-SWID

OWASP Software Component Verification Standard
https://owasp-scvs.gitbook.io/scvs/

NTIA Software Component Transparency Initiative

Contact: Allan Friedman, PhD afriedman@ntia.gov
Director of Cybersecurity Initiatives
National Telecommunications and Information Administration
US Department of Commerce