



# SBoM

## The Fad, The Future, and In-Between

Anant Shrivastava



# Anant Shrivastava



- Chief researcher @ Cyfinoid Research
- 15+ yrs of industry exposure
- **Speaker / Trainer:** BlackHat, c0c0n, nullcon, RootConf, RuxCon
- **Project Lead:**
  - Code Vigilant (Code Review Project)
  - Hacking Archives of India,
  - TamerPlatform (Android Security)
- (@anantshri on social platforms) <https://anantshri.info>

# Software Bill of Material

- Itemized list of all the **ingredients** in the software
- Ingredients means mostly third-party components
  - Software name
  - Version
  - Checksum
  - License information
  - Dependencies list if possible
- SBoM's are mostly for one level depth only with other levels plugged in each other.

# Every standard starts with competition

- SPDX
  - ISO Standard
  - Github provides default export in this format
- CycloneDX
  - OWASP Supported
- SWID
  - Alternative ISO specification

[https://www.ntia.gov/files/ntia/publications/sbom\\_formats\\_survey-version-2021.pdf](https://www.ntia.gov/files/ntia/publications/sbom_formats_survey-version-2021.pdf)

# Example CycloneDX SBoM

Object(6)	
Object(6)	<pre>1 { 2   "bomFormat": "CycloneDX", 3   "components": [ 4     { 5       "bom-ref": "pkg:npm/body-parser@1.19.0", 6       "description": "Node.js body parsing middleware", 7       "externalReferences": [ 8         { 9           "type": "website", 10          "url": "https://github.com/expressjs/body-parser#readme" 11        }, 12        { 13          "type": "issue-tracker", 14          "url": "https://github.com/expressjs/body-parser/issues" 15        }, 16        { 17          "type": "vcs", 18          "url": "git+https://github.com/expressjs/body-parser.git" 19        } 20      ], 21      "hashes": [ 22        { 23          "alg": "SHA-1", 24          "content": "96b2709e57c9c4e09a6fd66a8fd979844f69f08a"</pre>
String	<pre>"bomFormat": "CycloneDX"</pre>
Array[840]	<pre>"components": [</pre>
Object(3)	<pre>{</pre>
Object(8)	<pre>  "bom-ref": "pkg:npm/body-parser@1.19.0",   "description": "Node.js body parsing middleware",   "externalReferences": [</pre>
String	<pre>    {</pre>
String	<pre>      "type": "website",</pre>
Array[3]	<pre>      "url": "https://github.com/expressjs/body-parser#readme"</pre>
Array[1]	<pre>    },</pre>
String	<pre>    {</pre>
String	<pre>      "type": "issue-tracker",</pre>
String	<pre>      "url": "https://github.com/expressjs/body-parser/issues"</pre>
String	<pre>    },</pre>
String	<pre>    {</pre>
String	<pre>      "type": "vcs",</pre>
String	<pre>      "url": "git+https://github.com/expressjs/body-parser.git"</pre>
String	<pre>    }   ],</pre>
Array[1]	<pre>  "hashes": [</pre>
String	<pre>    {</pre>
String	<pre>      "alg": "SHA-1",</pre>
Number	<pre>      "content": "96b2709e57c9c4e09a6fd66a8fd979844f69f08a"</pre>

# Example SPDX SBoM

Object{9}
SPDXID: "SPDXRef-DOCUMENT"
creationInfo
dataLicense: "CC0-1.0"
documentDescribes
documentNamespace: "https://...graph/sbom-ed16ac5641d41487"
name: "com.github.antcf-d-supplychain/gotosocial"
packages
relationships
spdxVersion: "SPDX-2.3"

```
1 {
2   "SPDXID": "SPDXRef-DOCUMENT",
3   "creationInfo": {
4     "created": "2024-06-27T23:17:32Z",
5     "creators": [
6       "Tool: GitHub.com-Dependency-Graph"
7     ]
8   },
9   "dataLicense": "CC0-1.0",
10  "documentDescribes": [
11    "SPDXRef-com.github.antcf-d-supplychain-gotosocial"
12  ],
13  "documentNamespace": "https://github.com/antcf-d-supplychain/gotosocial",
14  "name": "com.github.antcf-d-supplychain/gotosocial",
15  "packages": [
```

# How to create SBoM

- Github provides dependency Graph in “Insights”
- SBoM generation tools
  - Cdxgen
    - <https://github.com/CycloneDX/cdxgen>
  - SPDX Generator
    - <https://github.com/spdx/tools>
- /dev/hand if all else fails (Its XML)

# GitHub Export SBOM Option

The screenshot displays the GitHub interface for a repository's dependency graph. The top navigation bar includes links for 'Pulse', 'Projects', 'Security' (with a badge showing 106 issues), 'Insights', and 'Settings'. The left sidebar contains a list of navigation items: 'Pulse', 'Contributors', 'Community', 'Traffic', 'Commits', 'Code frequency', 'Dependency graph' (which is highlighted with a red border), 'Network', 'Forks', and 'People'. The main content area is titled 'Dependency graph' and features three tabs: 'Dependencies' (selected), 'Dependents', and 'Dependabot'. A search bar labeled 'Search all dependencies' is positioned below the tabs. On the right side of the 'Dependencies' tab, there is a button labeled 'Export SBOM' with a download icon. Below the search bar, a summary bar indicates '980 Total' dependencies. The main list shows four dependencies with their respective versions and security status:

Dependency	Version	Source	License	Security Status
@babel/traverse	7.21.5	Detected automatically on Oct 05, 2023 (npm) · web/source/yarn.lock	MIT	1 critical
minimist	0.0.5	Detected automatically on Oct 05, 2023 (npm) · web/source/yarn.lock	MIT	1 critical
golang.org/x/net	0.15.0	Detected automatically on Oct 05, 2023 (Go modules) · go.mod		1 high
google.golang.org/grpc	1.58.0	Detected automatically on Oct 05, 2023 (Go modules) · go.mod		1 high

**DEMO GODS**



**PLEASE LET THIS DEMO WORK**

# Is SBoM really useful

- SBoM rose to prominence coz of exec order by US President.
- Requirement is to create SBoM
- No directions around usage, consumption etc
- **SBoM Tells you software composition nothing else**
- Industry representatives have started asking Questions?
  - Should we focus on building SBOM or fix issues in that time?

# Thoughts from Industry around SBoM

- Why should I disclose my composition to the world
- I will only share the SBoM to NDA covered entities
- I don't need SBoM coz I don't sell to USA
- Better to spend time in fixing bugs then making SBoM

Food for thought

software industry  
is mostly  
fixing problems  
created by  
software industry




# What problems have we created

- Software build automation == quicker release cycle
- Automated release cycle == less wait for features
- Faster feature release == less inclination to upgrade dependencies
- Too much focus on OSS Codebase without helping the maintainers
- Impossible segregation of features and bug fixes
- Automated notification of vulnerability (hedonic hamster wheel)

---

CreatedAssignedMentionedReview requests

Q is:open is:pr author:app/dependabot archived:false

 56,205,238 Open ✓ 49,162,339 Closed

Visibility Organization Sort

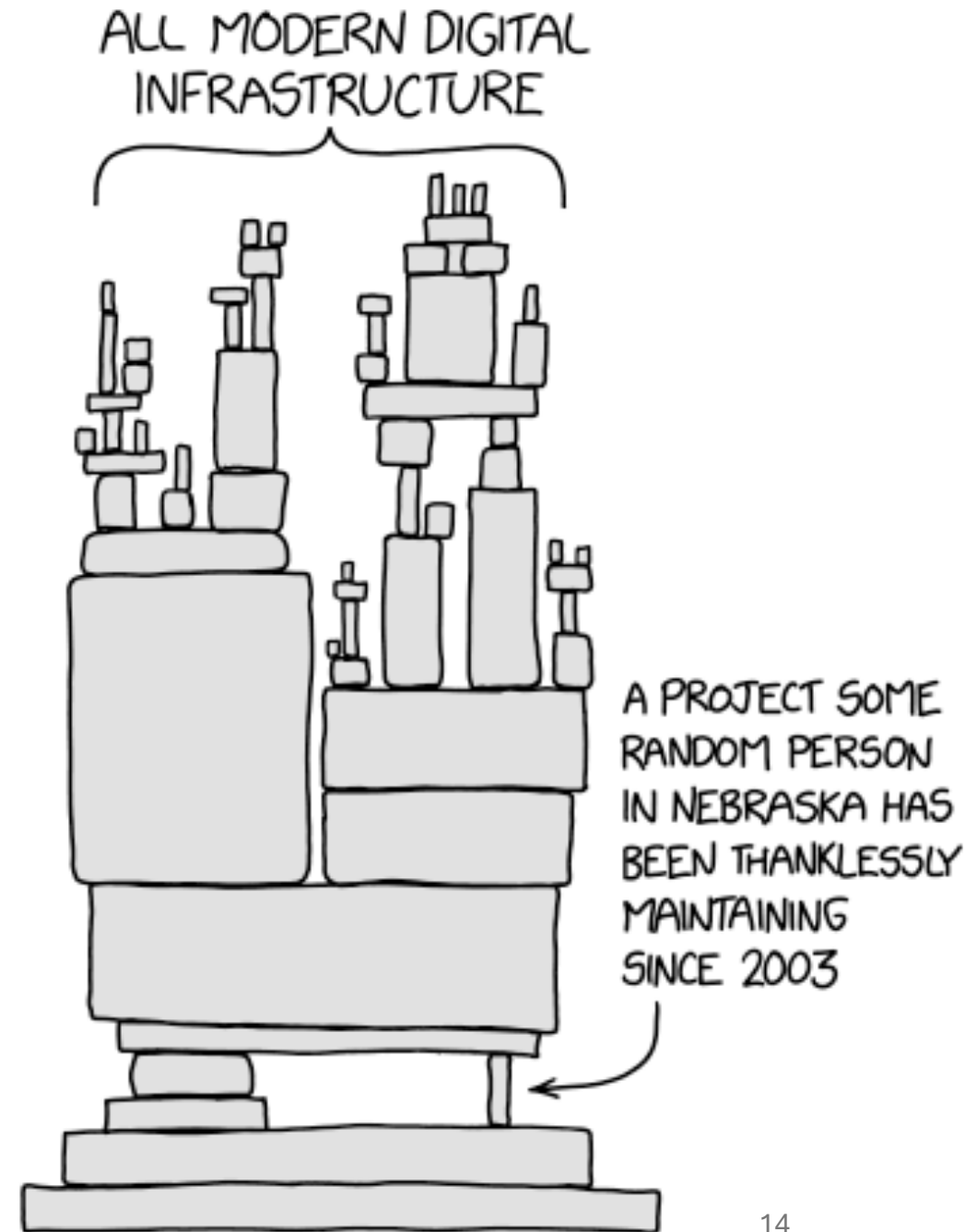
(C) Cyfinoid Research

# SBoM can help

- Identifying incorrect use of software
- Identify what to fix in scenarios like log4shell
- Identify impact in sec bug release in a core component
- Basically, Inventory problems

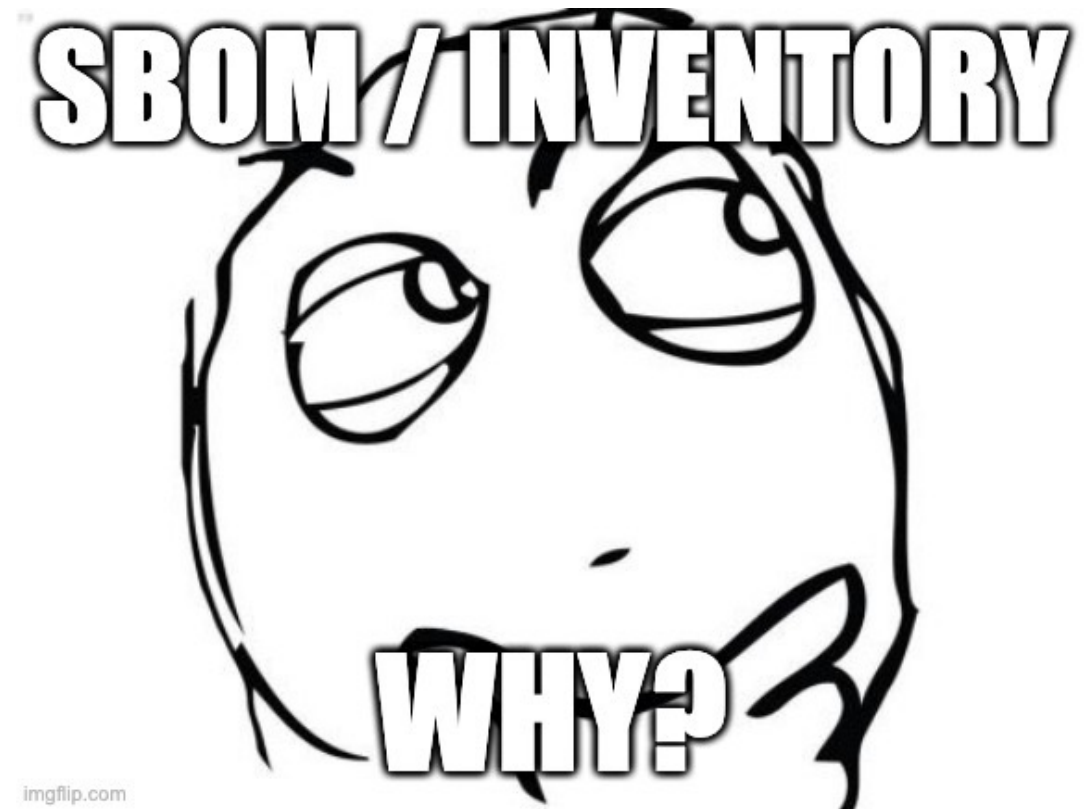
Ref: XKCD.com/2347

(C) Cyfinoid Research



# Another thought

- Infosec has never had the luxury of well-maintained inventory
- SBoM can help with it
- we never had inventory; we don't even know what to do with it when its created



imgflip.com

# Consequences for Infosec

For Practitioners both infosec and Devops

- We have been asking for better visibility, this is it

For Industry entities

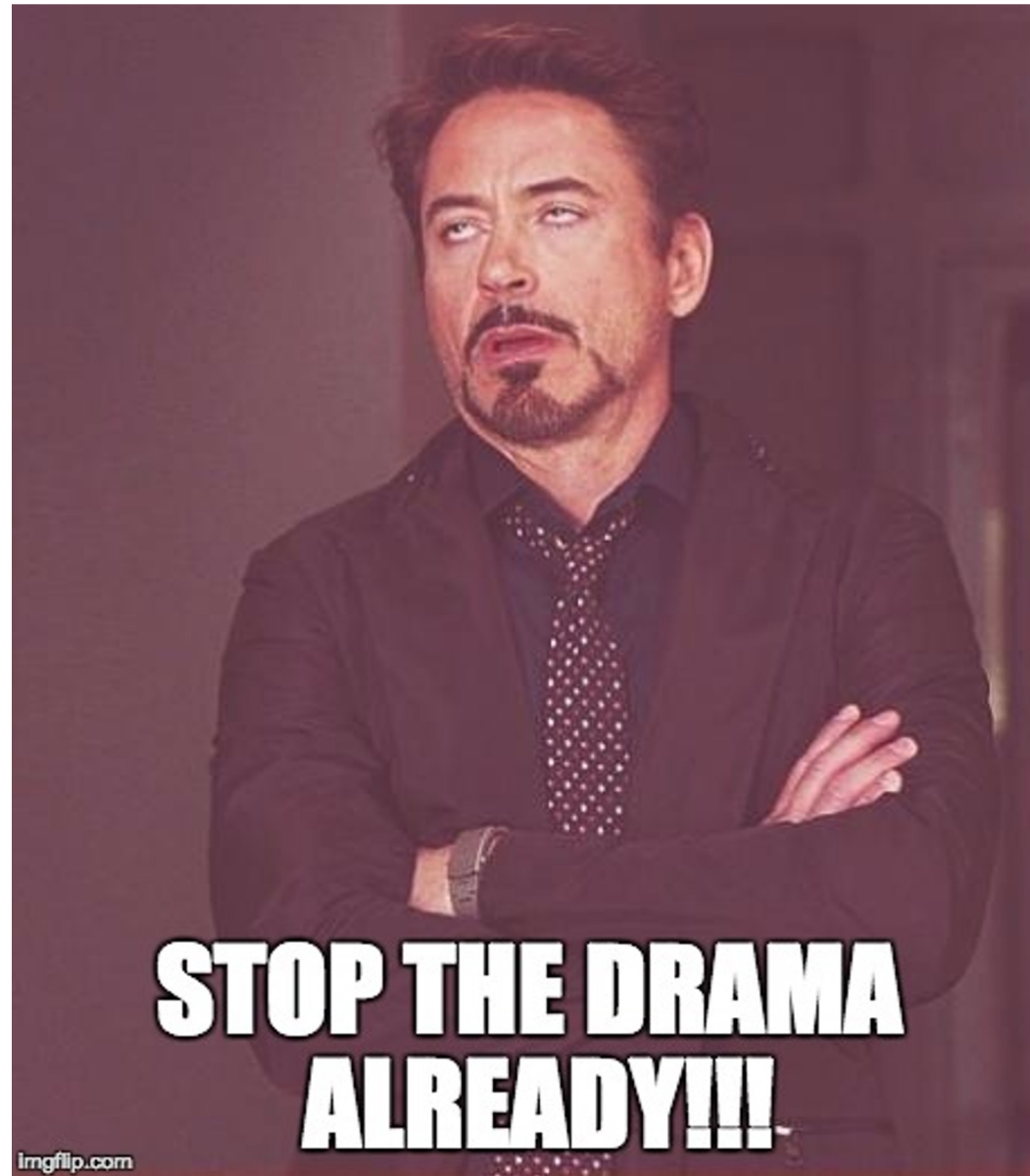
- This is like “opening the kimono” moment



If security practitioners  
want to preserve this facility  
they need to act now



imgflip.com



# Existing tooling

- OWASP Cyclone DX : <https://cyclonedx.org/>
- Google SLSA dev : <https://slsa.dev/>
- SPDX : <https://spdx.dev/>
- SafeDep : <https://safedep.io/>
- Dependency Tracker: <https://dependencytrack.org/>
- SYFT : <https://github.com/anchore/syft>

# Security efforts already in progress

- VDR : Vulnerability Disclosure Report
- VEX : Vulnerability Exploitability eXchange
- xBoM's
  - Software-as-a-Service Bill of Materials (SaaSBoM)
  - Hardware Bill of Materials (HBOM)
  - Machine Learning Bill of Materials (ML-BOM)
  - Cryptography Bill of Materials (CBOM)
  - Manufacturing Bill of Materials (MBOM)
  - Operations Bill of Materials (OBOM)
- Attestations

# What can we do

- Security is largely considered a cost center and any incentive that is solely useful for security is a cost.
- Inventory allows organization to make data driven decisions
- Make SBoM's usefulness visible for other departments
- If more people especially profit centers and business requirements (HR, Finance) need it, its hard to kill

# SBoM Usage beyond security teams

Use each SBoM as part of inventory,  
Consolidate then and then draw inferences from it

- Development
- Acquisitions and mergers
- Compliance (adjunct security)
- Risk Management

# SBoM usage for Developers

- Manage technical Debt
- Reduce dependency scatter
- Consolidate efforts for usage
- Simplified package selection in case of newer project

# SBoM usage for Acquisitions & Mergers

Use SBoM as an indicator for future cost and decision

- Too many outdated / EOL / unmaintained software in use leads to high cost of ownership after acquisitions
- If the toolset / techstack is vastly different than existing, then extra talent cost
- If too many techstacks in picture, shows non cohesive teams

# SBoM usage for Compliance

- Licensing policy spread not just at product but at input component level
- Possible cost of rework due to non-compliance with company policy
- Possible repercussions if my code touches this code (GPL restrictions to name as one)

# SBoM usage for Risk Management

Interesting questions that can be answered

- Do I want to include X amount of risk by purchasing this vendor's software?
- If risk is low but product will be highly visible, can I still afford it.
- Even with high risk, in a self-contained environment is it okay
- Do I really want my SSO auth token going into this software

# What is needed

- Better tooling (tech and UX)
  - Current tools are not easy to use even for practitioners
- Collaboration and seeking feedback from other parties
  - Don't make tooling for yourself make it for others
- Focus on usage not on glamorizing tech
  - We technologists focus too much of technical side.

# To Conclude

- I believe SBoM is a Boon for overall IT Industry to move in better directions.
- Newton's first law of motion stands : Inertia can only be countered by greater force
- There is a bright future ahead if we can muster the courage for it

Thanks for listening & open to Questions?

**NAME** **WEBSITE**



The diagram illustrates the components of the email address 'anant@cyfinoid.com'. It features three blue curly braces: one above 'anant' labeled 'NAME', one above '@cyfinoid.com' labeled 'WEBSITE', and one below the entire address labeled 'EMAIL'.

anant@cyfinoid.com

**EMAIL**

# References

- What if we spent all the time it took to make SBOMs fixing bugs instead? [#appsec #sbom](#) : <https://www.youtube.com/watch?v=jdNtUK8kpJE>
- SBoM Guidance by Kymberlee Price: [https://www.youtube.com/watch?v=\\_yWRQ6XM6pQ](https://www.youtube.com/watch?v=_yWRQ6XM6pQ)