



# **Securing Diverse Supply Chains** across Interconnected Systems



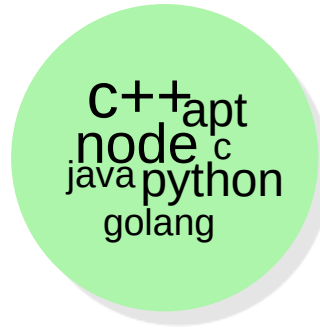
Wayne Starr  
Aaron Creel  
01 February 2023

# The Problem

Synergizing activities across an organization for Large Software Systems can be difficult and time consuming.



**Teams**



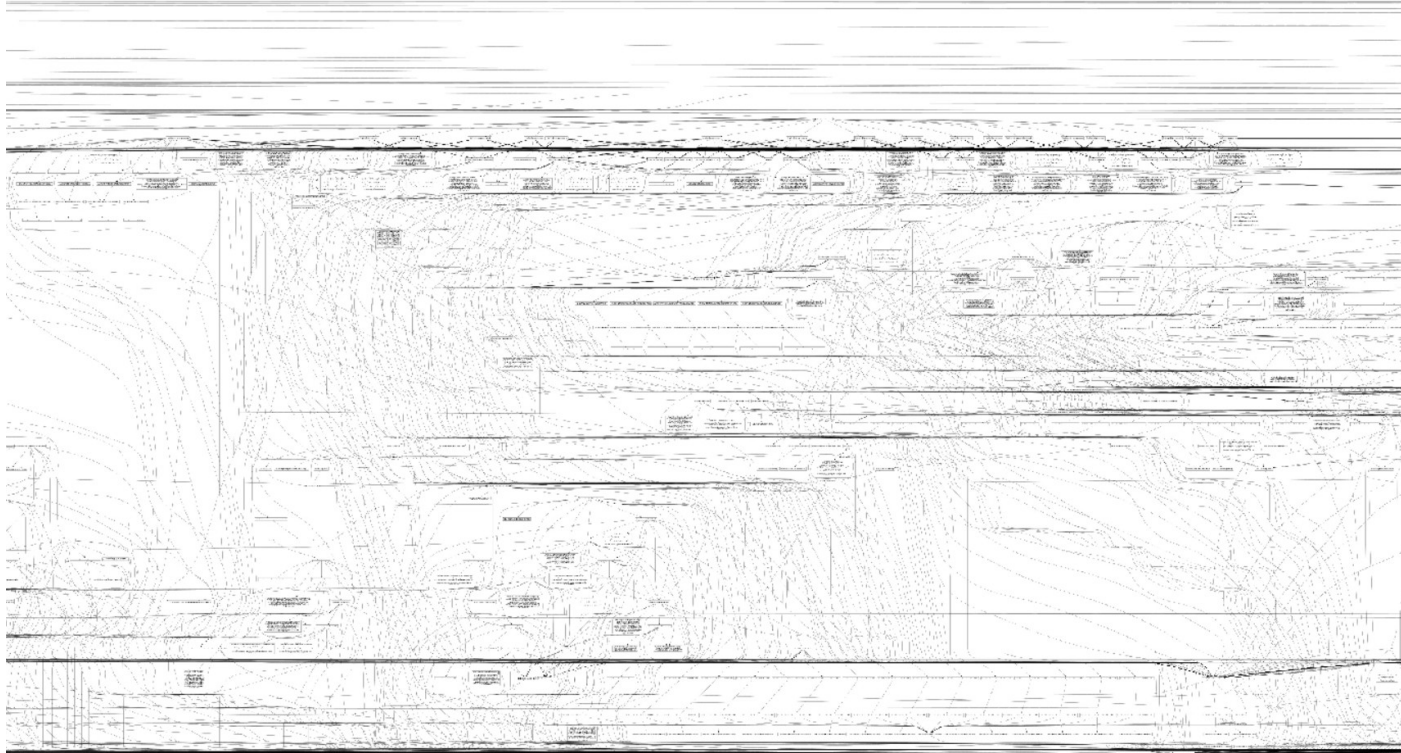
**Ecosystems**



**Environments**



# One Software Project\*



**\* Not including containers, web or infrastructure components**

# Why Solve This?

master	1 branch	0 tags
Go to file Add file Code		
Marak endgame 2c4f82f 15 hours ago 1 commit		
.github	endgame	15 hours ago
.eslintignore	endgame	15 hours ago
.eslintrc	endgame	15 hours ago
.gitattributes	endgame	15 hours ago
.gitignore	endgame	15 hours ago
.npmignore	endgame	15 hours ago
.travis.yml	endgame	15 hours ago
.versions	endgame	15 hours ago
Readme.md	endgame	15 hours ago
package.json	endgame	15 hours ago

```
}
a.writeFile(1, 2) {
  '1': 'node_modules\\discord-api-types\\rest\\v9\\gateway.d.ts',
  '2': '♥'
}
a.writeFile(1, 2) {
  '1': 'node_modules\\discord-api-types\\rest\\v9\\gateway.d.ts.map',
  '2': '♥'
}
a.writeFile(1, 2) {
  '1': 'node_modules\\discord-api-types\\rest\\v9\\gateway.d.ts',
  '2': '♥'
}
a.writeFile(1, 2) {
  '1': 'node_modules\\discord-api-types\\rest\\v9\\gateway.d.ts.map',
  '2': '♥'
}
a.writeFile(1, 2) {
  '1': 'node_modules\\discord-api-types\\rest\\v9\\gateway.d.ts',
  '2': '♥'
}
a.writeFile(1, 2) {
  '1': 'node_modules\\discord-api-types\\rest\\v9\\gateway.d.ts.map',
  '2': '♥'
}
```

[Home](#) > [An open letter to the Linux community - April 24, 2021](#)

## An open letter to the Linux community - April 24, 2021

Dear Community Members:

We sincerely apologize for any harm our research group did to the Linux kernel community. Our goal was to identify issues with the patching process and ways to address them, and we are very sorry that the method used in the "hypocrite commits" paper was inappropriate. As many observers have pointed out to us, we made a mistake by not finding a way to consult with the community and obtain permission before running this study; we did that because we knew we could not ask the maintainers of Linux for permission, or they would be on the lookout for the hypocrite patches. While our goal was to improve the security of Linux, we now understand that it was hurtful to the community to make it a subject of our research, and to waste its effort reviewing these patches without its knowledge or permission.

We just want you to know that we would never intentionally hurt the Linux kernel community and never introduce security vulnerabilities. Our work was conducted with the best of intentions and is all about finding and fixing security vulnerabilities.

The "hypocrite commits" work was carried out in August 2020; it aimed to improve the security of the patching process in Linux. As part of the project, we studied potential issues with the patching process of Linux, including causes of the issues and suggestions for addressing them.

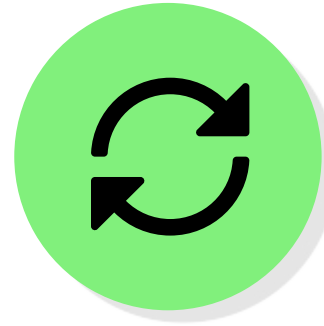


# Solutions



## Policy

Implement organizational standards to capture data consistently



## Automation

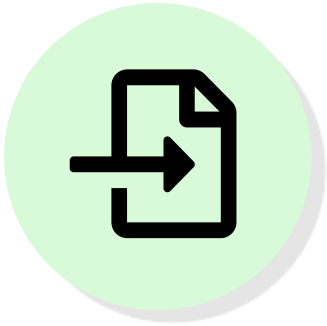
Check dependencies and vulnerabilities with automated tooling



# Policy

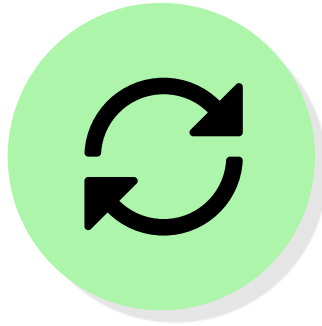


# Standard Checkpoints



## **On Import**

Check packages when they enter the dev environment



## **On Integration**

Check packages on build or in continuous integration



## **On Deployment**

Check packages as they are deployed to production



# The Stale Shelf

In addition to common checkpoints you must continuously evaluate the products that you have running in production.

Vulnerabilities never stop being discovered and you must be prepared to respond when updates are required.





# Keep Things Simple

Reduce developer friction through standardized / templated tooling within the enterprise

Tighten feedback loops within the environment developers use today to provide feedback quickly

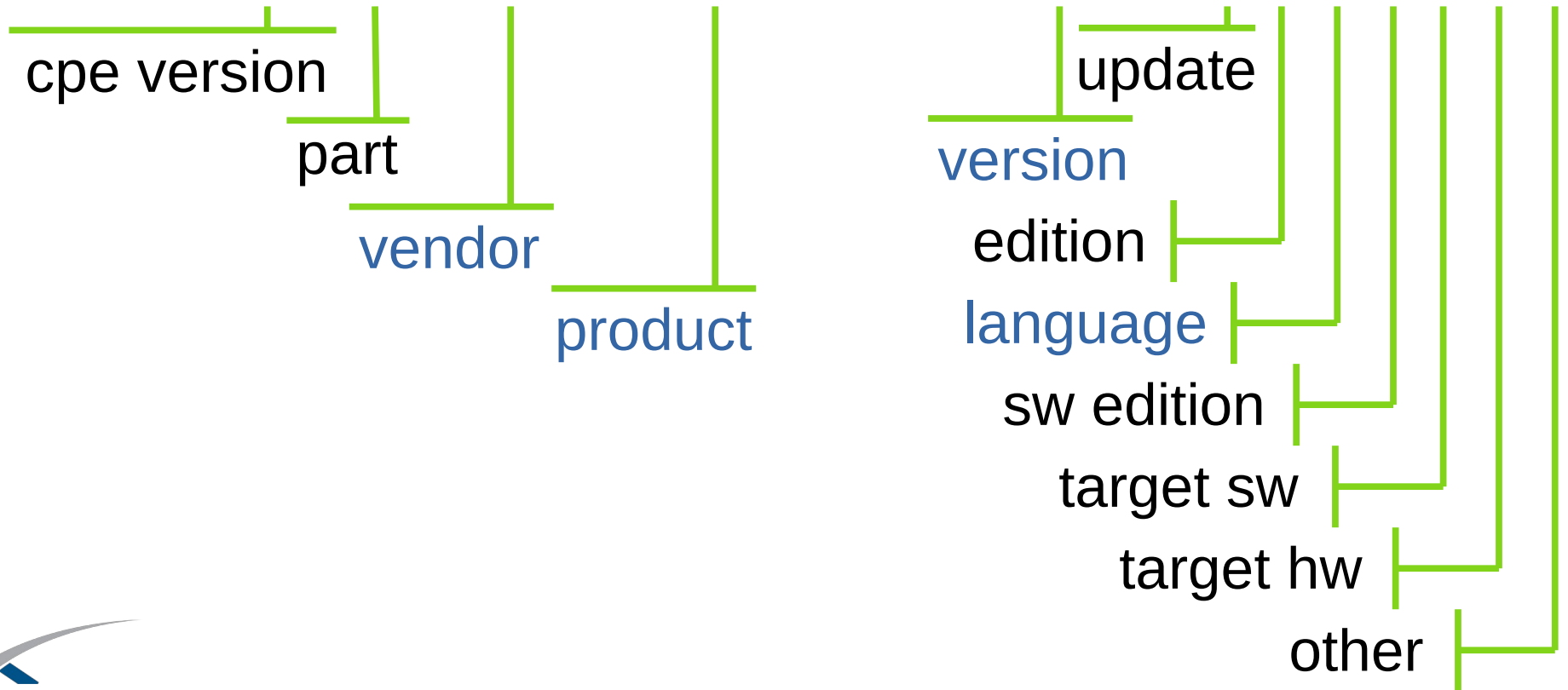


# **Automation**



# Package Identification

cpe:2.3:a:google:protobuf:3.1.0:\*:\*:\*:\*:\*:\*



# Analysis Considerations

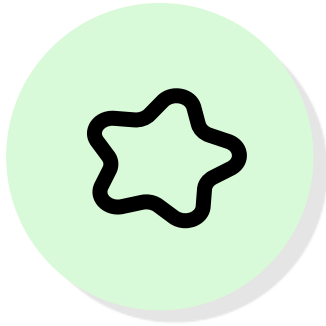
Is this package on disk or  
in a requirements file?

Is this package publicly  
available or private?

Is this package actually in  
use by the application?

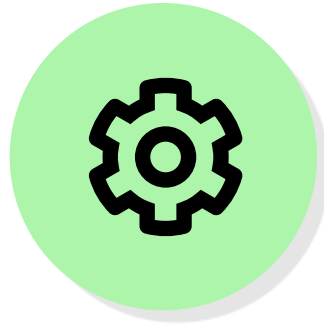


# Shortcomings



## Consistency

Tooling can be inconsistent or not provide required information across package ecosystems



## Configurability

SBOM utilities sometimes lack the configurability needed to run within large (slow) codebases



## Management

Once data is collected, managing data and keeping things together can be difficult



# Demo



# Questions

