DevOps in high security environments

👉 PollEv.com/Isak000 👈

Join and say hello! 👋🎉

“ Hello 👋 ”

DevOps in high security environments

👉 PollEv.com/Isak000 👈

Bienvenue!

Gustaf et Isak 👋

Isak Hassbring
Not a DevOps Engineer at SAAB

Gustaf Lidfeldt
DevOps Engineer at SAAB

DevOps in **high security environments**

SAAB

Structure

- Intro
- DevOps in high security
- Open source attacks
- Reflection
- Take home message

👉 PollEv.com/Isak000 👈

What is an acceptable risk?

“ it depends... ”

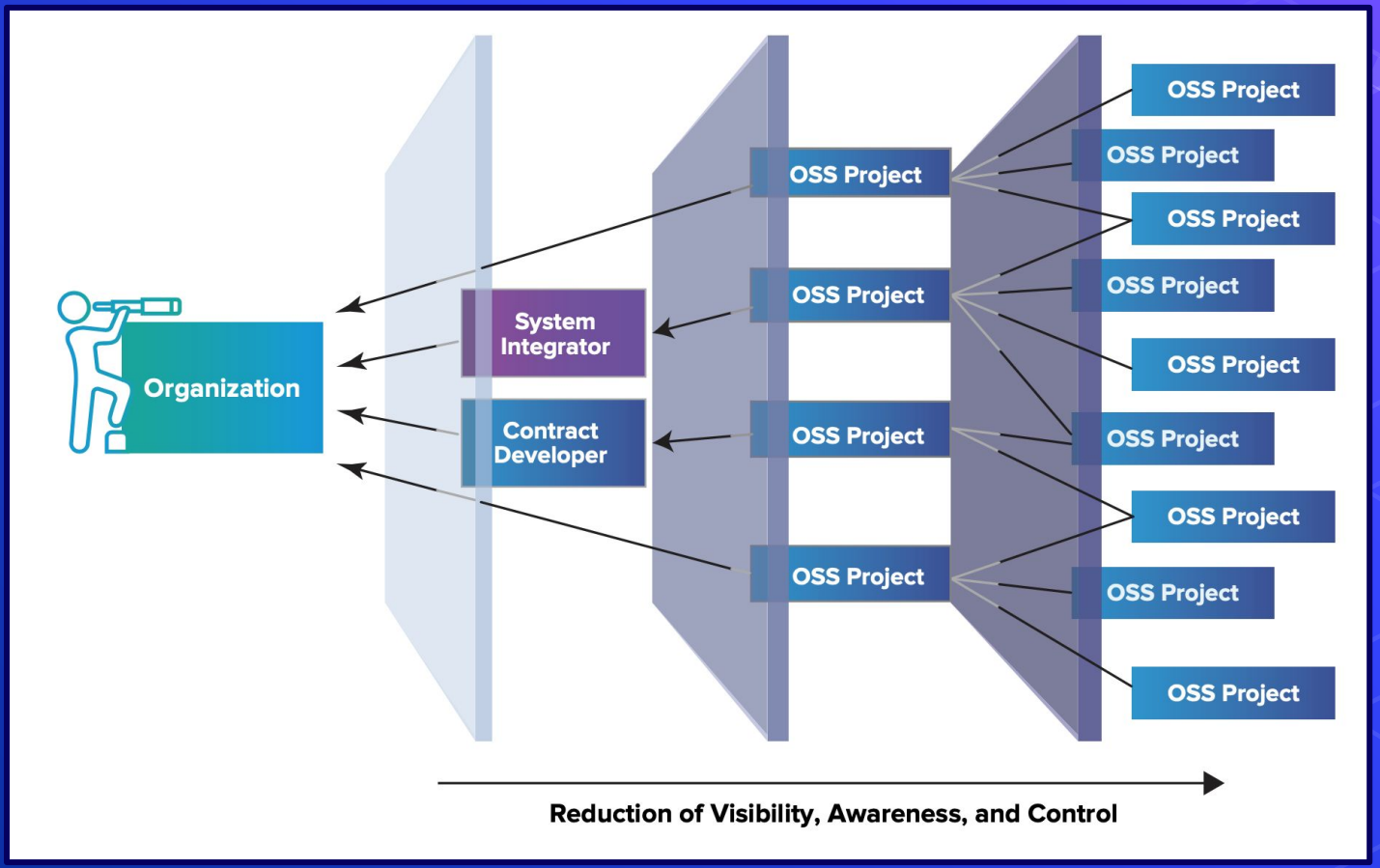
“ One that meets the requirements ”

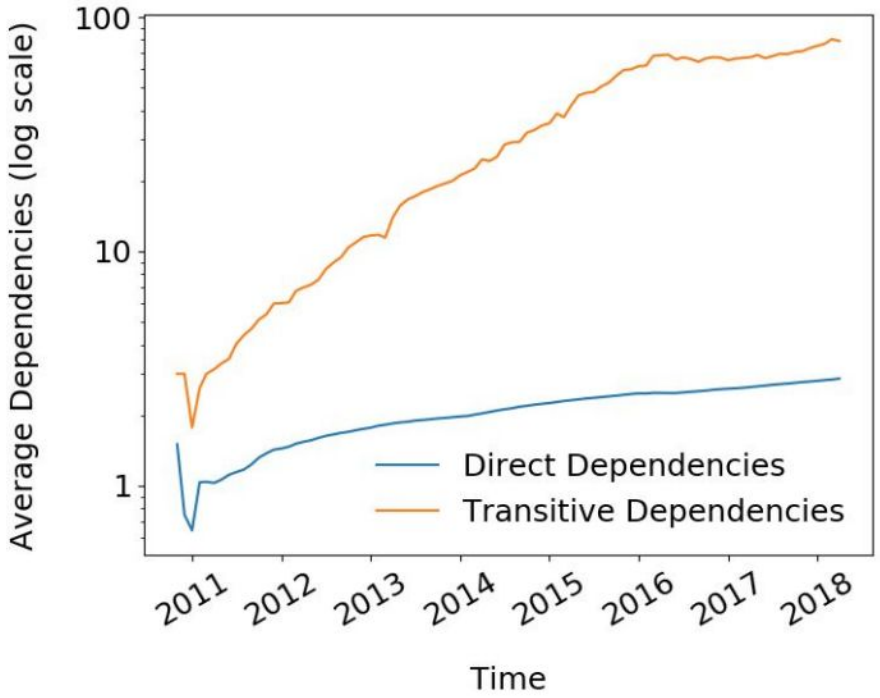
Open Source

One snippet to rule them all

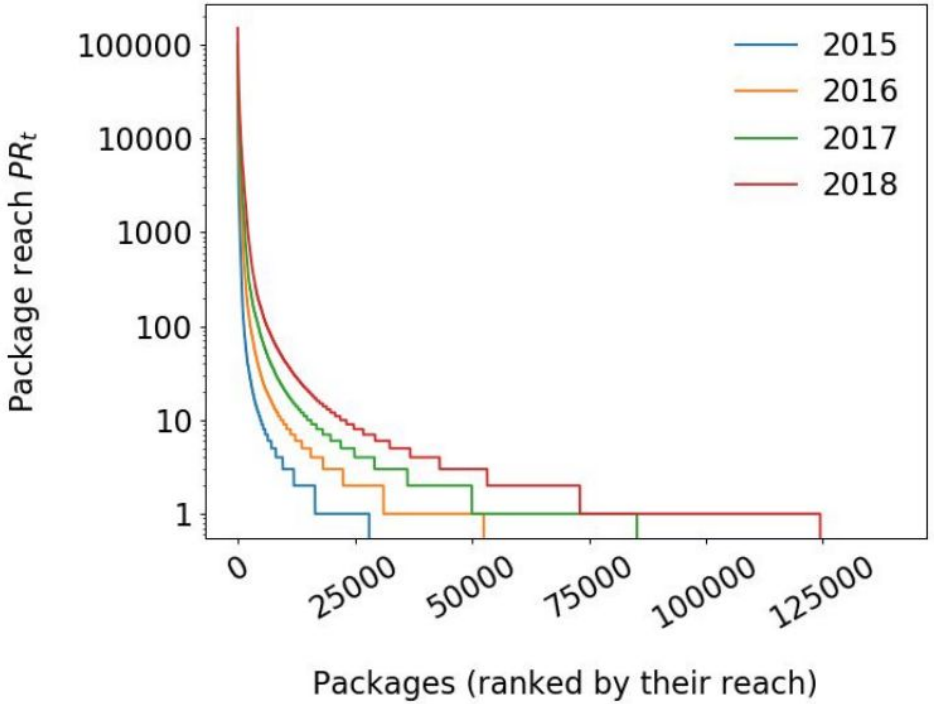
npm install

Dev performance vs **Ops** security?

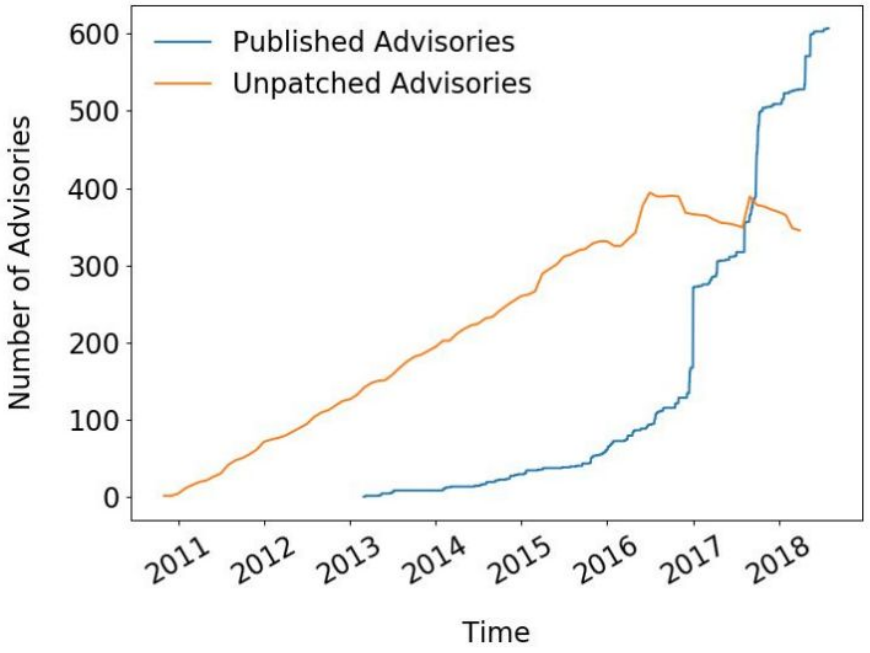

Organization
System Integrator
Contract Developer
OSS Project
OSS Project
OSS Project
OSS Project
OSS Project
OSS Project
OSS Project
OSS Project
OSS Project
OSS Project
Reduction of Visibility, Awareness, and Control

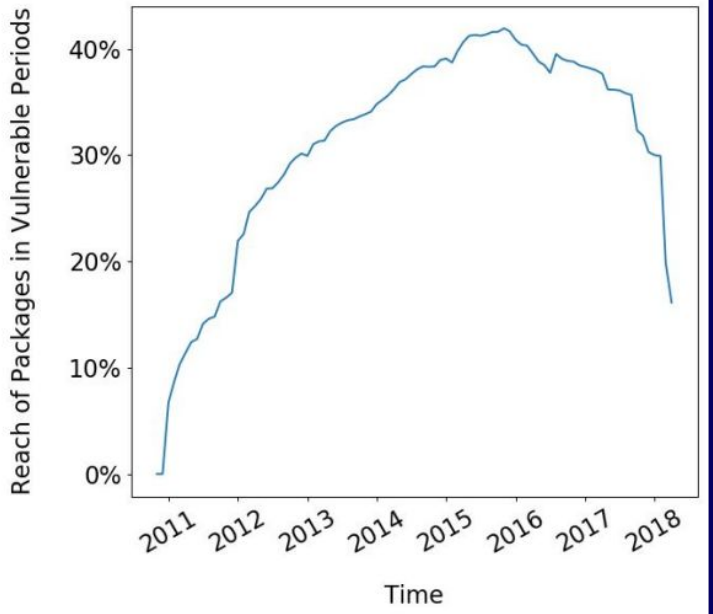


A user implicitly trusts 79 other packages due to transitive dependencies.



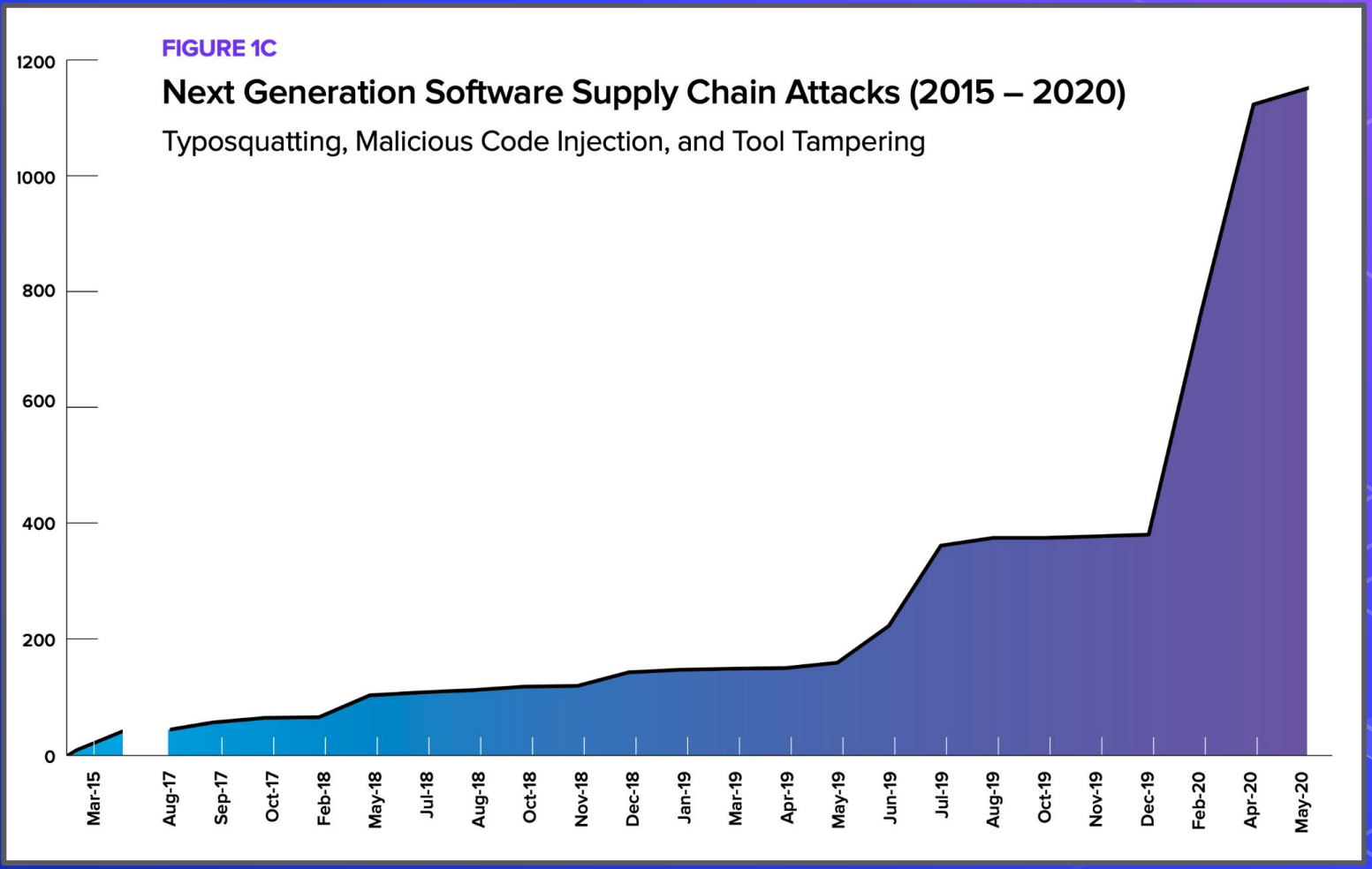
Popular packages can reach more than 100,000 other packages, making them a prime target for attacks.



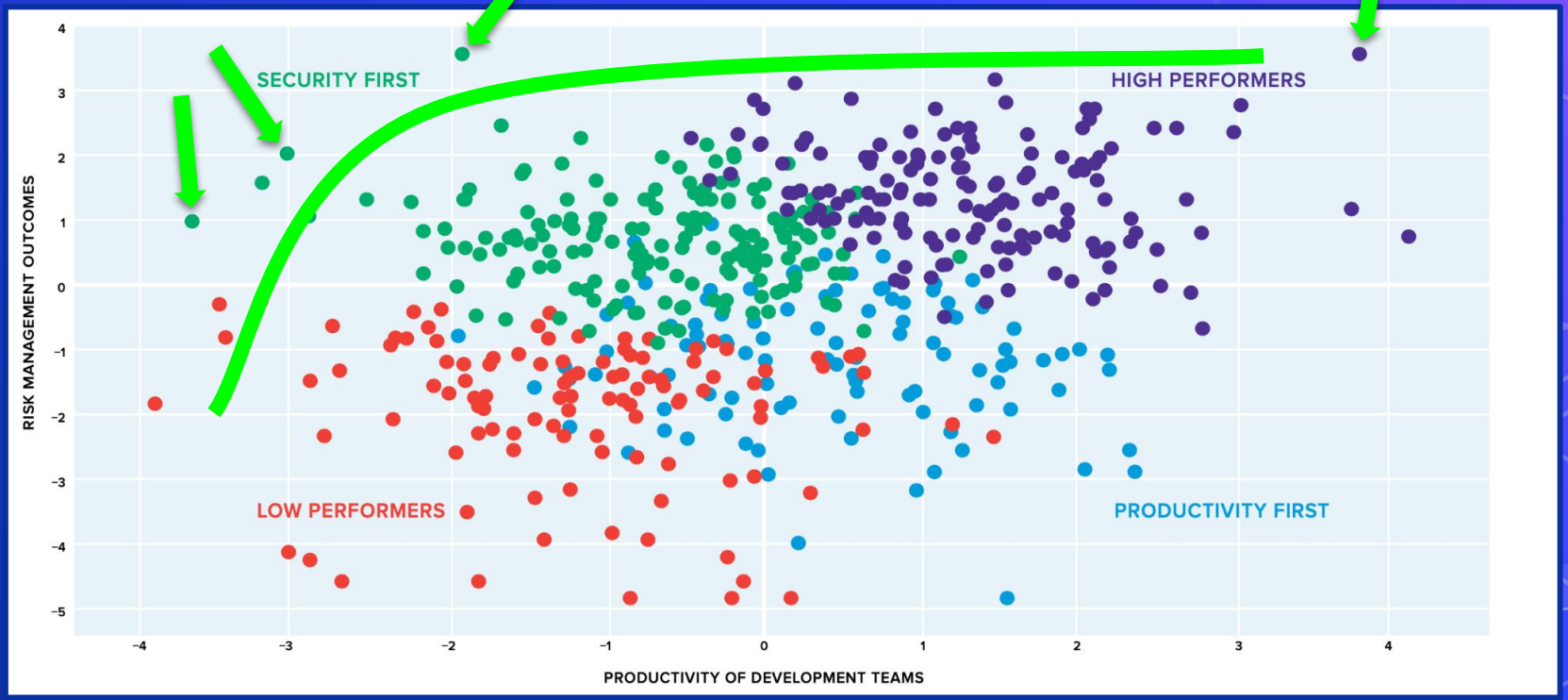


Up to 40% of all packages rely on code known to be vulnerable.

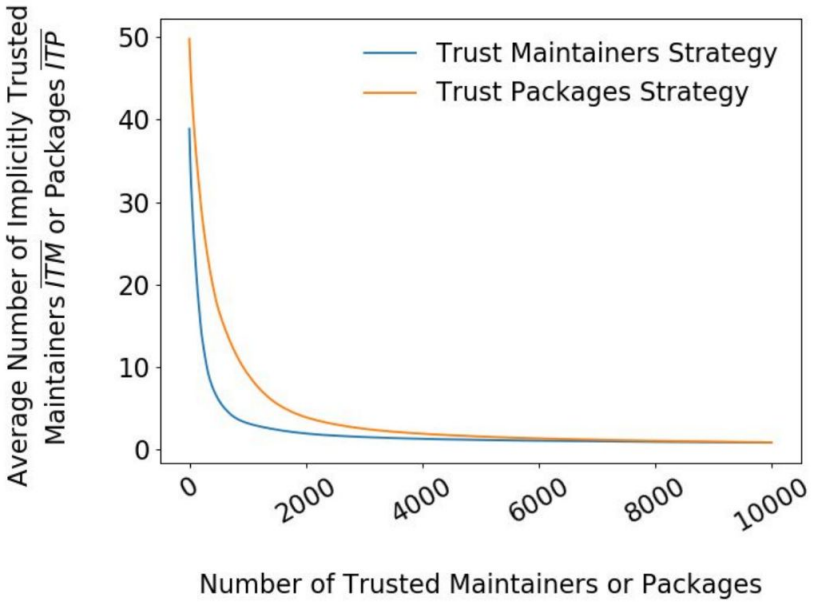
FIGURE 1C

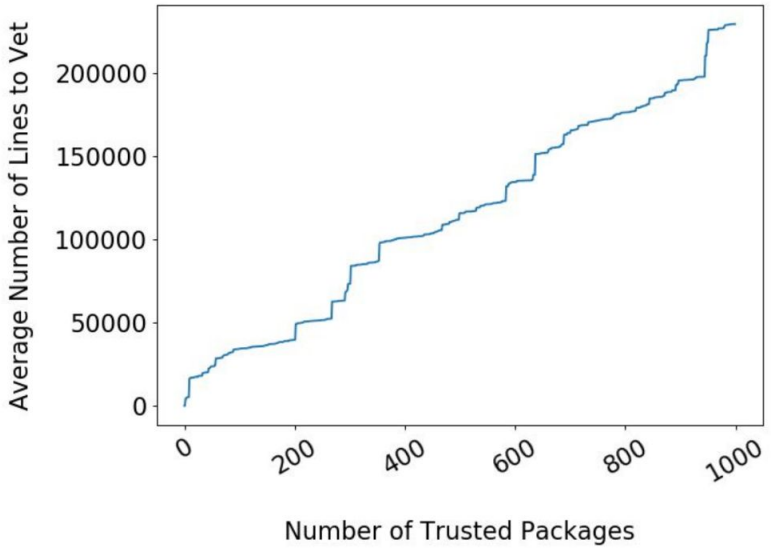

Next Generation Software Supply Chain Attacks (2015 – 2020)
Typosquatting, Malicious Code Injection, and Tool Tampering
1200
1000
800
600
400
200
0
Mar-15
Aug-17
Sep-17
Oct-17
Feb-18
May-18
Jul-18
Aug-18
Oct-18
Nov-18
Dec-18
Jan-19
Mar-19
Apr-19
May-19
Jun-19
Jul-19
Aug-19
Oct-19
Nov-19
Dec-19
Feb-20
Apr-20
May-20

Safe-start the devops journey



Vetting as a mitigation strategy





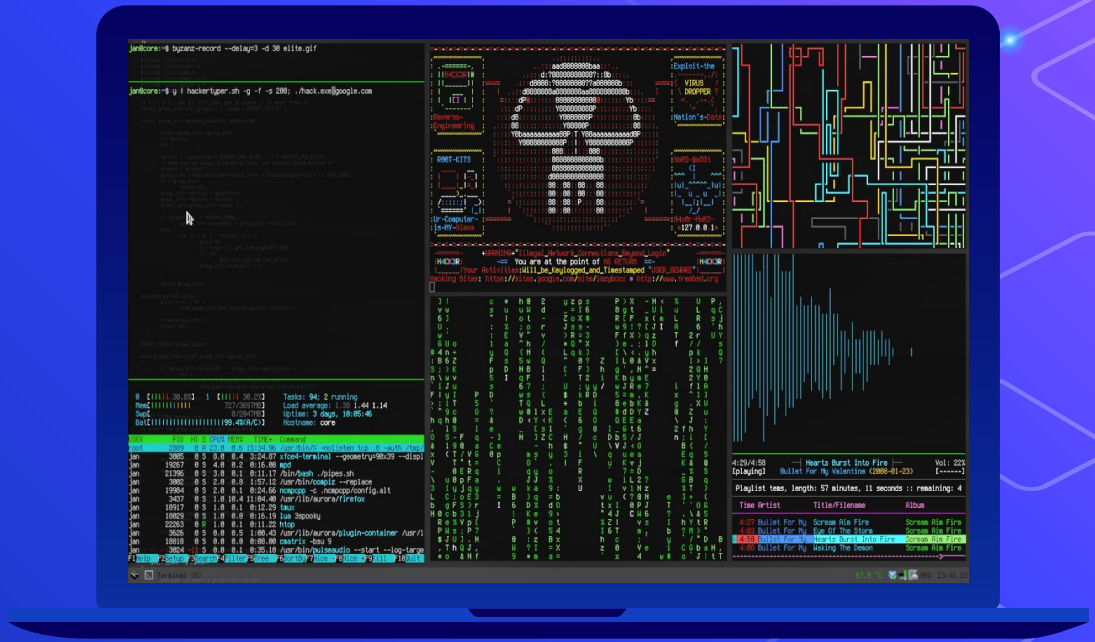
Vetting the most dependent upon 1,500 packages would reduce the implicitly trusted packages by a factor of 10

14

Open source attacks

What they are

How they occur



Equifax Breach 2017

- Unpatched third party software.
- 173 millions records.
- $ 575 millions in settlement.

How to avoid OSS attacks?

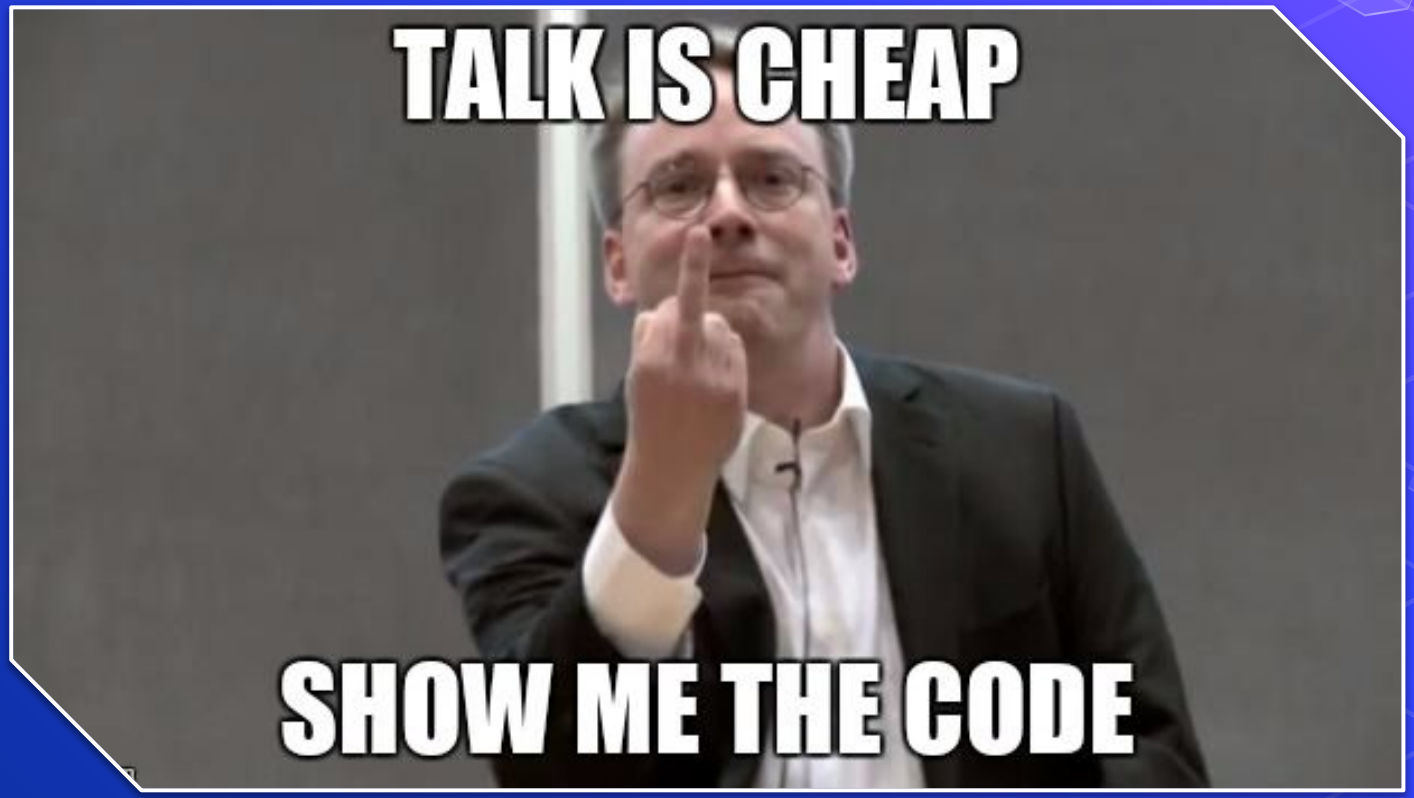
Open Source PO & Contributors

- More dependency updates = higher quality and more secure code.

Enterprise Development Teams

- Chose OSS tools as a strategic decision.
- Remedating known OSS vulnerabilities as a regular part of development.
- Using Software Composition Analysis (SCA) tools to identify vulnerabilities.

How to avoid OSS attacks?



How to avoid OSS attacks?
Retire.js for NPM dependencies

```
npm install -g retire

retire
ICanHandlebarz/test/jquery-1.4.4.min.js

↳ jquery 1.4.4.min has known vulnerabilities:

http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2011-4969

http://research.insecurelabs.org/jquery/test/

http://bugs.jquery.com/ticket/11290
```

Vulnerability Examples

- Serialization
- Path traversal

```
def path = System.console().readLine 'Enter file path:'
if (path.startsWith("/safe_dir/"))
{
        File f = new File(path);
        f.delete()
}
```

```
class Vault(object):
    '''R/W an ansible-vault yaml file'''

    def __init__(self, password):
        self.password = password
        self.vault = VaultLib(password)

    def load(self, stream):
        '''read vault steam and return python object'''
        return yaml.load(self.vault.decrypt(stream)) [0]
```

Reflection

- Ask yourself - “How often do you consider security?”
- “Risk is derived from the value of the protected asset”
- “Given enough eyeballs, all bugs are shallow” - Linus Torvalds
- Check out the OWASP foundation!

How often do you consider security in your own projects?

Reflection

- Ask yourself - “How often do you consider security?”
- “Risk is derived from the value of the protected asset”
- “Given enough eyeballs, all bugs are shallow” - Linus Torvalds
- Check out the OWASP foundation!

Take home message

"In DevOps, risk must be assessed everywhere. Software, hardware and people."

Thanks!

Please use the docs for Q/A

References

- State of the Software Supply Chain 2020 - Sonatype
- https://www.usenix.org/sites/default/files/conference/protected-files/sec19_slides_zimmermann.pdf

For the curious listener

- Owasp Foundation: https://owasp.org/www-project-top-ten/
- Simple vulnerable code snippets: https://github.com/snoopysecurity/Vulnerable-Code-Snippets
- GitHub community that focuses on patching security flaws: https://github.com/snyk