OWASP MSP Chapter

December 2023 lunch-and-learn

Minneapolis / St. Paul • Minnesota • USA

This meeting will be recorded

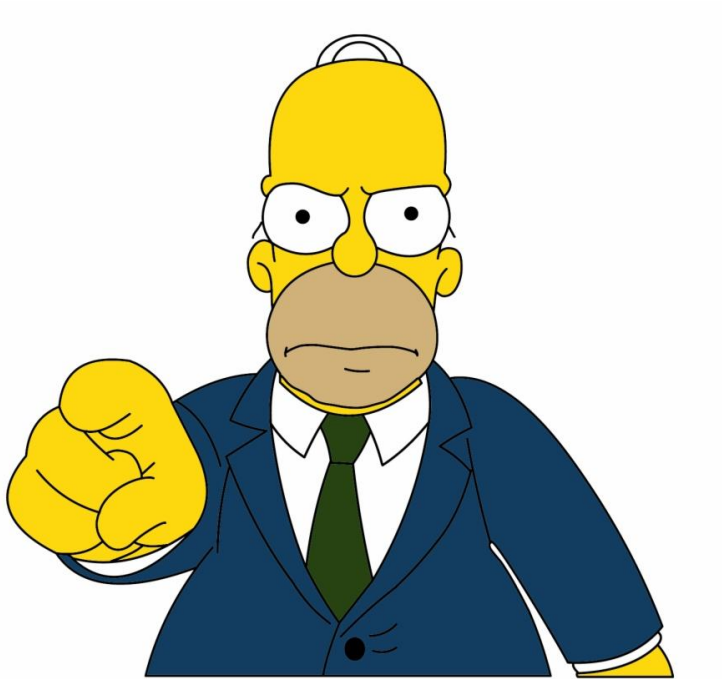
Food and locale provided by

- www.concordusa.com

- Thank you!

OWASP needs you!

- Share your knowledge
- Leadership opportunities
- Volunteer for OWASP-MSP
- Help build an AppSec project



Getting social

OWASP-MSP page: www.owasp.org/www-chapter-minneapolis-st-paul

Chapter meetings: www.meetup.com/OWASP-MSP-Meetup

Welcome first-timers!

Recorded talks: https://www.youtube.com/@owasp-msp

Mailing list: OWASP-MSP-list@meetup.com

Or just type “OWASP MSP” into your favorite search engine.

CPEs

You can get credit just for showing up – but you have to ask for a certificate of attendance by emailing the chapter leads.

You can also get a much more awesome certificate packed with even more geek cred for preparing and giving a talk at a chapter meeting!

Employment opportunities

And now…

Software Composition Analysis with OWASP Dependency-Check

Zoa Buske
Senior Cybersecurity Engineer
ICF Inc.

Nathan Larson
Senior Security Architect
Concord

What's the problem?

- Software is complex. Really, mind-blowingly complex.
- Name one commercial product / web app that doesn't use external libraries

- Why shouldn't we use them?
  - Don't reinvent the wheel
  - Use well-crafted code
  - Lower product cost
- But….
  - Trust their authors?
  - Trust ongoing support?
  - Trust their security?

What's the big deal?

Vulnerable and Outdated Components is #6 on the 2021 Top 10 (up from #9 in 2017)

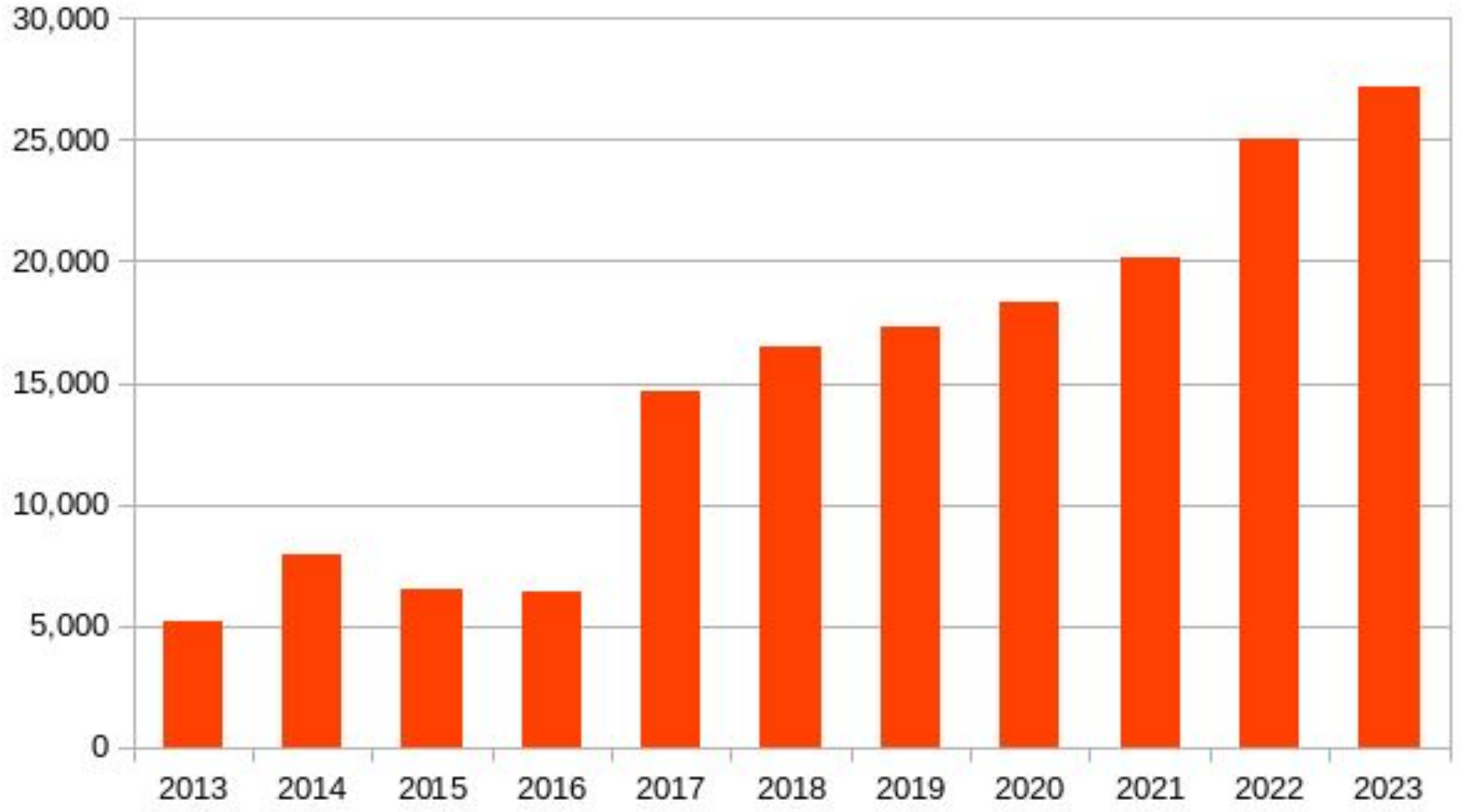
2021

A01:2021-Broken Access Control
A02:2021-Cryptographic Failures
A03:2021-Injection
A04:2021-Insecure Design
A05:2021-Security Misconfiguration
A06:2021-Vulnerable and Outdated Components
A07:2021-Identification and Authentication Failures
A08:2021-Software and Data Integrity Failures
A09:2021-Security Logging and Monitoring Failures*
A10:2021-Server-Side Request Forgery (SSRF)*

owasp.org/Top10

What's the big deal?

Public vulnerabilities increase nearly every year



Data from www.cvedetails.com

What’s the big deal?

- Equifax breach
  - Data exfil due to vuln version of Apache Struts in production
  - Struts security update 7 March 2017
  - Breach began 66 days later
  - 143m Americans’ credit records breached

What’s the big deal?

- libwebp package flaw (CVEs 2023-4863, 2023-5129, and 2023-41064; CVSS 10.0)
  - Used in “a multitude of software” including:

    iOS, Android, Chrome, Nginx, Python, Joomla, WordPress, and Node.js
  - Actively exploited to deploy NSO Group’s Pegasus surveillance spyware
  - One step leading to a zero-click exploit dubbed BLASTPASS

Great, something else to do

- We already (should) design secure systems
- We already (should) check for vulnerable code we write
- We already (should) check for vulnerable behaviors in apps
- We should also check for vulnerable libraries in our repos

How SCA tools work

- Scan the manifest or actual code for libraries used
- Look up the libraries/versions in vuln databases (e.g. nvd.nist.gov)
- Report on the number, type, and severity of vulns found

Some SCA Best Practices

- Regularly update dependencies – include in backlogs
  - Especially security-related patches!
  - Some updates break code; plan for this as well

    More frequent scans/patches ⇒ fewer findings and breaking changes
- Watch for security alerts in libraries you use
- Implement an SCA tool to scan automatically in the pipeline
- Establish a vulnerability management process, with reasonable SLAs

Main takeaways

- Watching and patching dependencies takes work
- Keep known critical vulns out of prod
- Vulnerability management program helps

Questions?

Demo!

Links

- Equifax breach: https://en.wikipedia.org/wiki/2017_Equifax_data_breach
- libwebp package flaw CVE-2023-5129: https://securityaffairs.com/151576/hacking/cve-2023-5129-libwebp-flaw.html
- OWASP Dependency Check: https://owasp.org/www-project-dependency-check/
  - https://github.com/jeremylong/DependencyCheck
- Dependabot: https://docs.github.com/en/code-security/dependabot
- GitHub Code Security: https://docs.github.com/en/code-security
- Snyk: https://snyk.io