

Open Source Developers are Security's new front line

A shifting landscape of attacks

Ilkka Turunen
Global Director, Sonatype
[@IlkkaT](#)

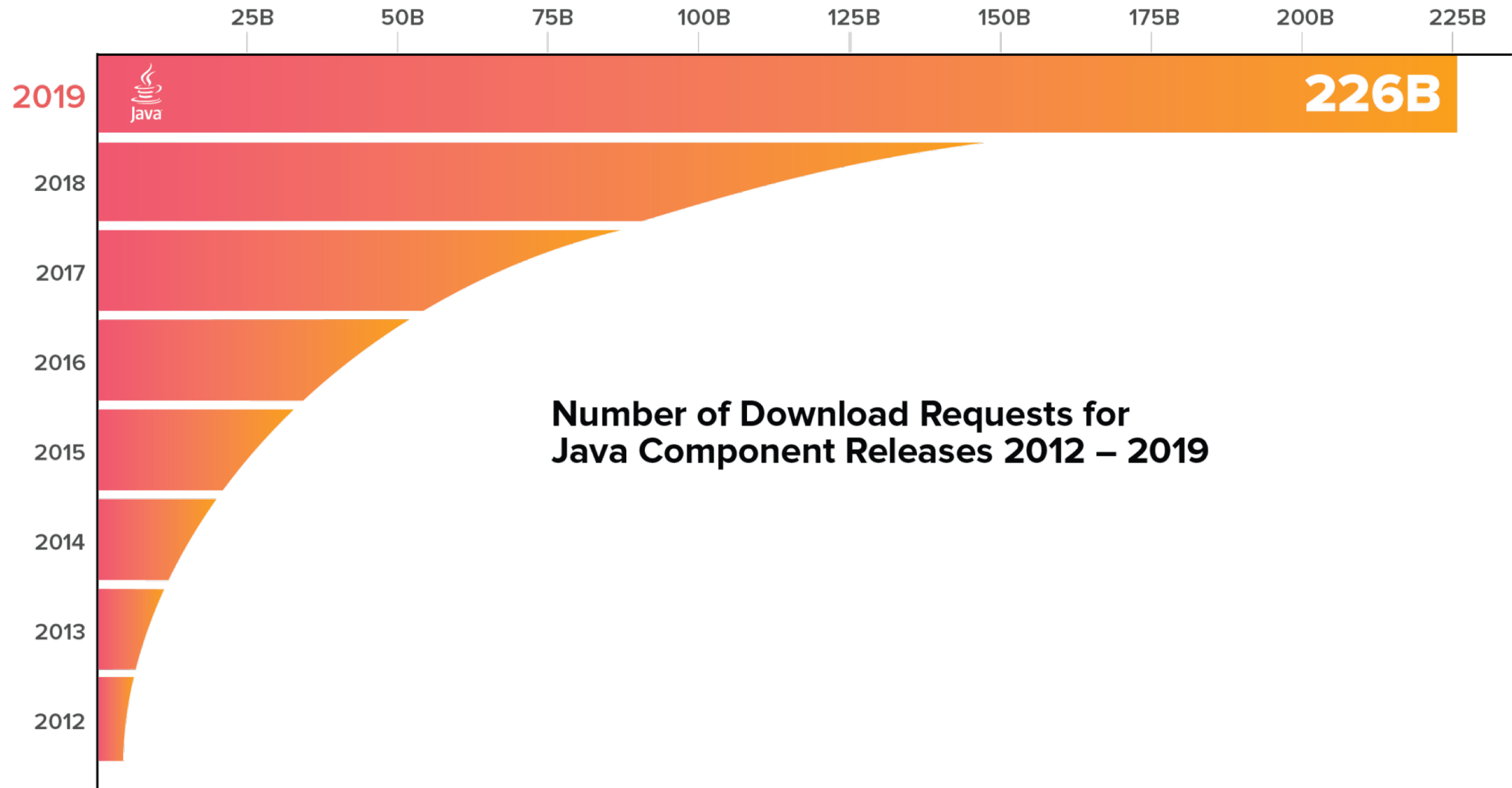
20XX: Software has eaten the world...
It used open source to chew it up

Everyone has a software supply chain.

(including open source projects)

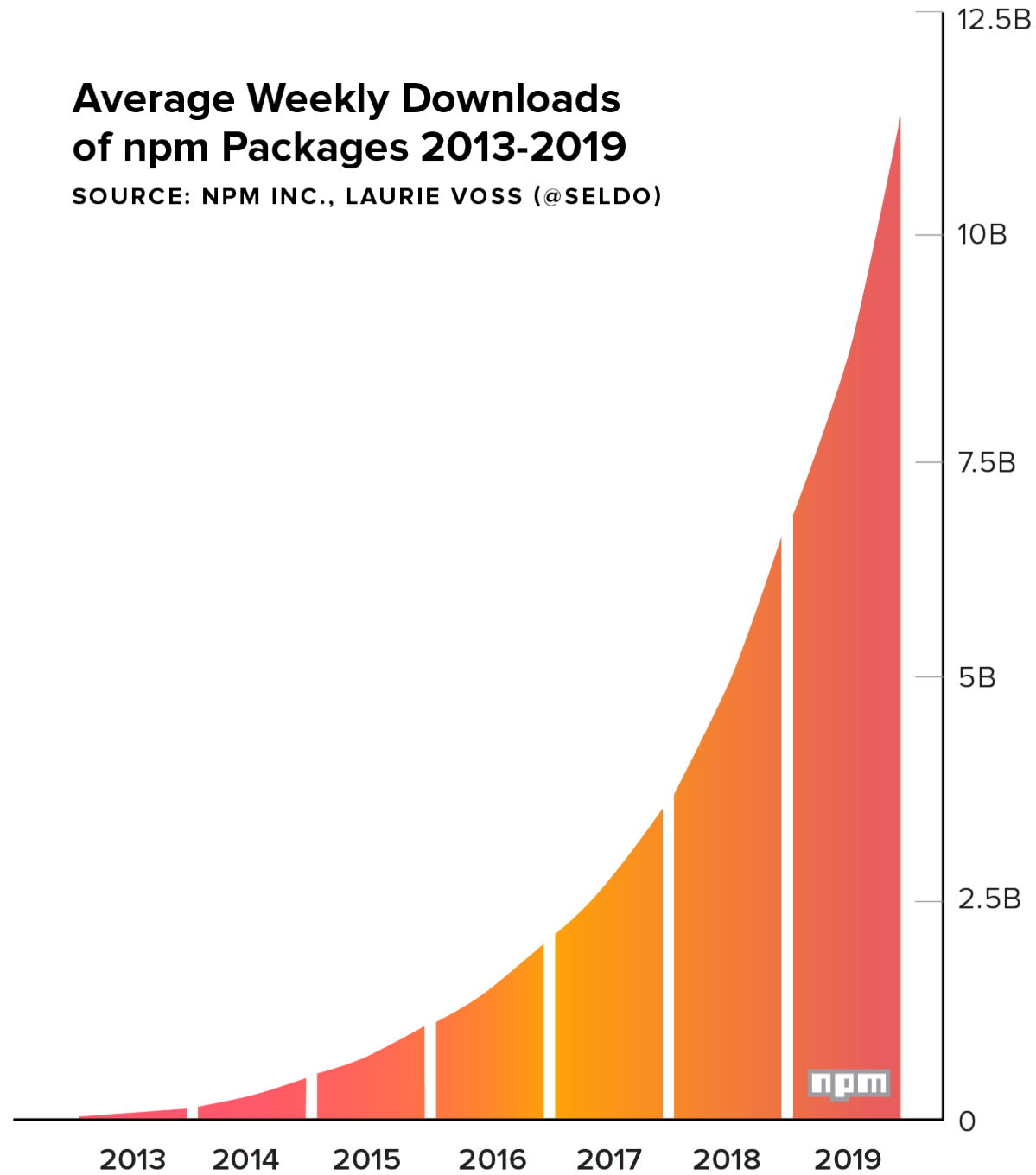


BILLIONS



Average Weekly Downloads of npm Packages 2013-2019

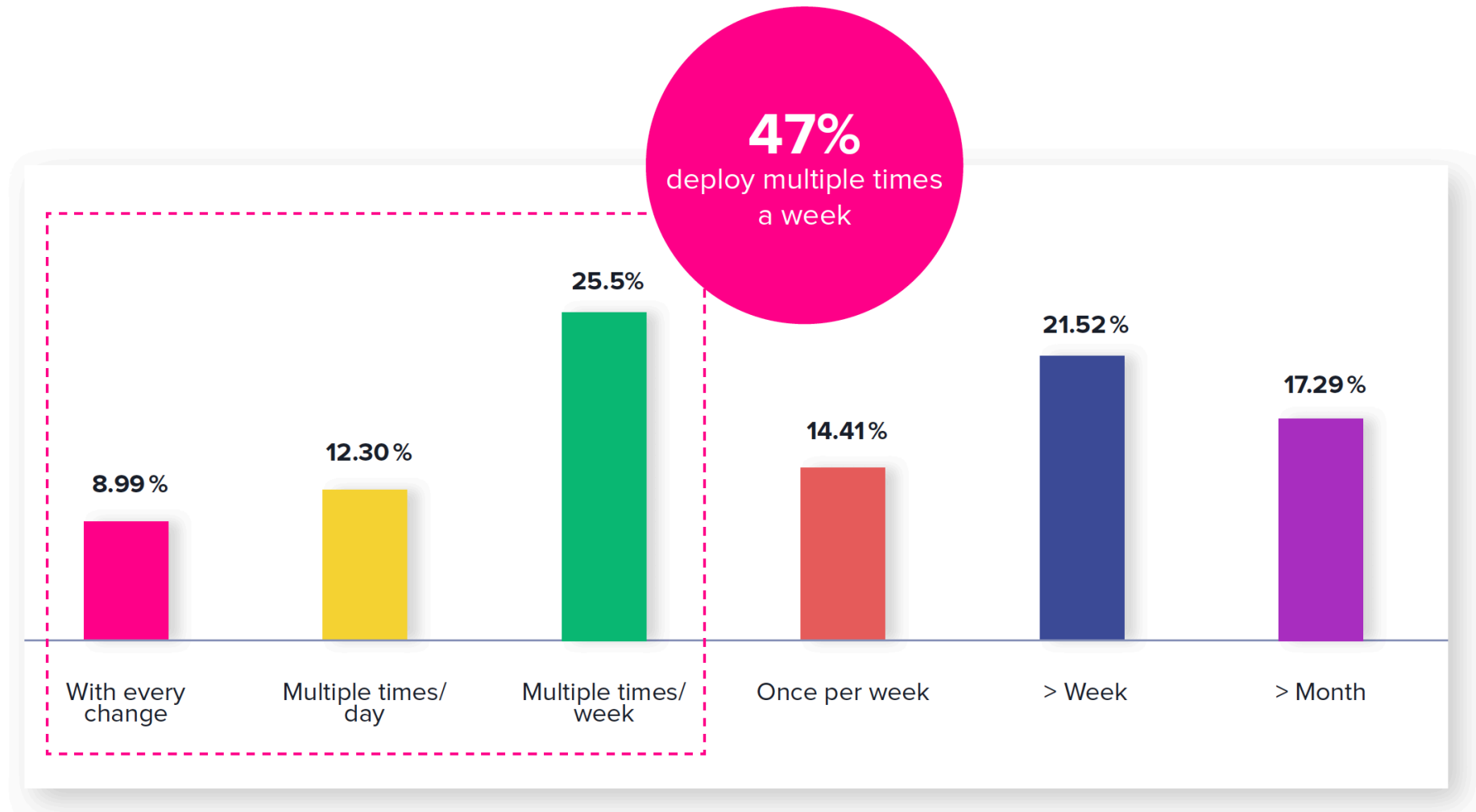
SOURCE: NPM INC., LAURIE VOSS (@SELDO)



BILLIONS



Open source helps us release value faster



Faster is better in the enterprise.

...faster is better for adversaries?

WE DON'T WANT TO REINVENT THE WHEEL,
SO EVERY DAY WE GOOGLE IMAGE SEARCH
"WHEEL," AND WHATEVER OBJECT COMES UP,
THAT'S WHAT WE ATTACH TO OUR VEHICLES.

!
SURE, EXTERNAL DEPENDENCIES
CARRY RISKS, BUT SO FAR THEY'VE
ALL BEEN PRETTY GOOD WHEELS.



313,000

java component
downloads annually

2,778


Component suppliers

8,200

Component release

27,704

8.8% with known
vulnerabilities

A donut chart with a dark blue background and a white border. A small segment of the chart is highlighted in pink, representing 8.8% of the total. The chart is connected to the left side of the image by a thin white line.



The infographic features a large yellow circle on the left and a smaller donut chart on the right, connected by a horizontal line. The background is a solid blue color. The yellow circle contains the text '60,660 JavaScript packages downloaded annually per developer'. The donut chart is divided into two segments: a white segment representing 49% and a pink segment representing 51%. The text '30,330 51% with known vulnerabilities' is centered within the donut chart.

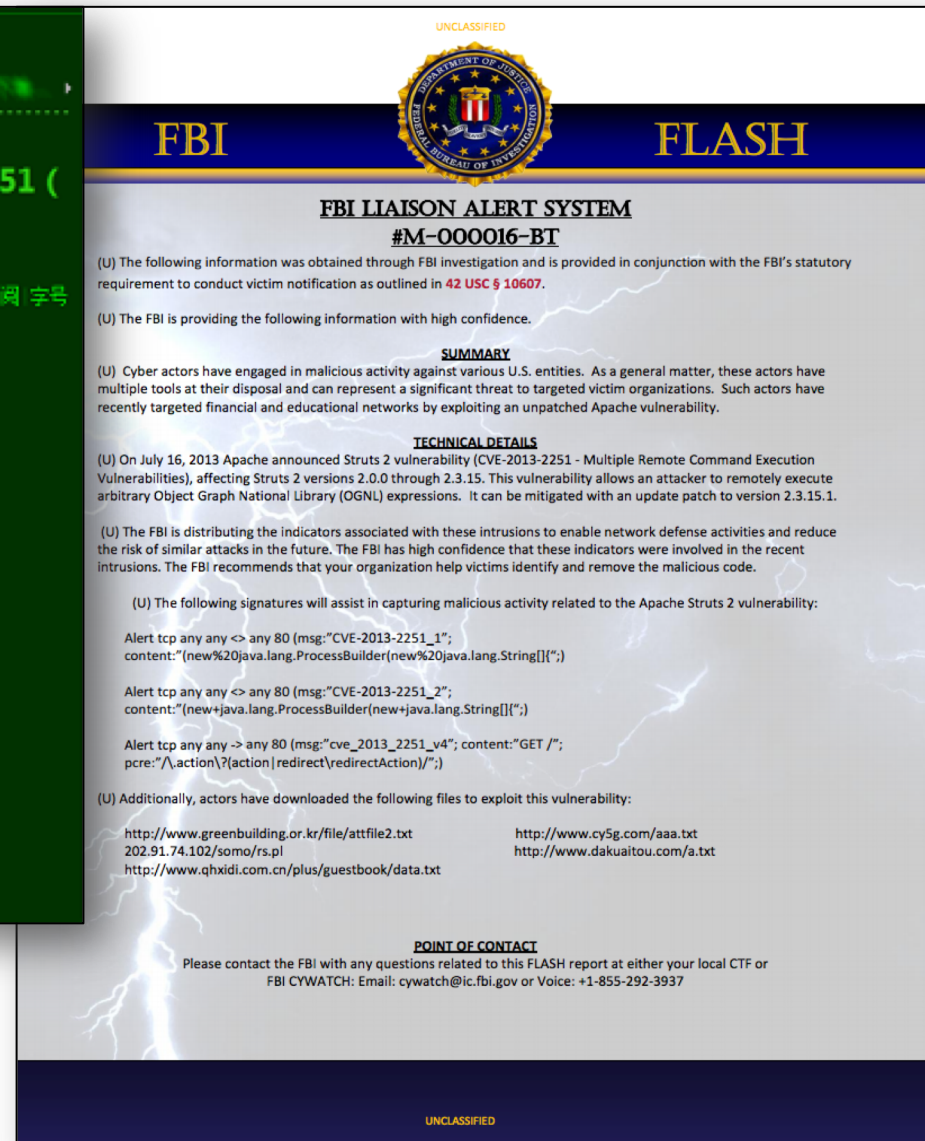
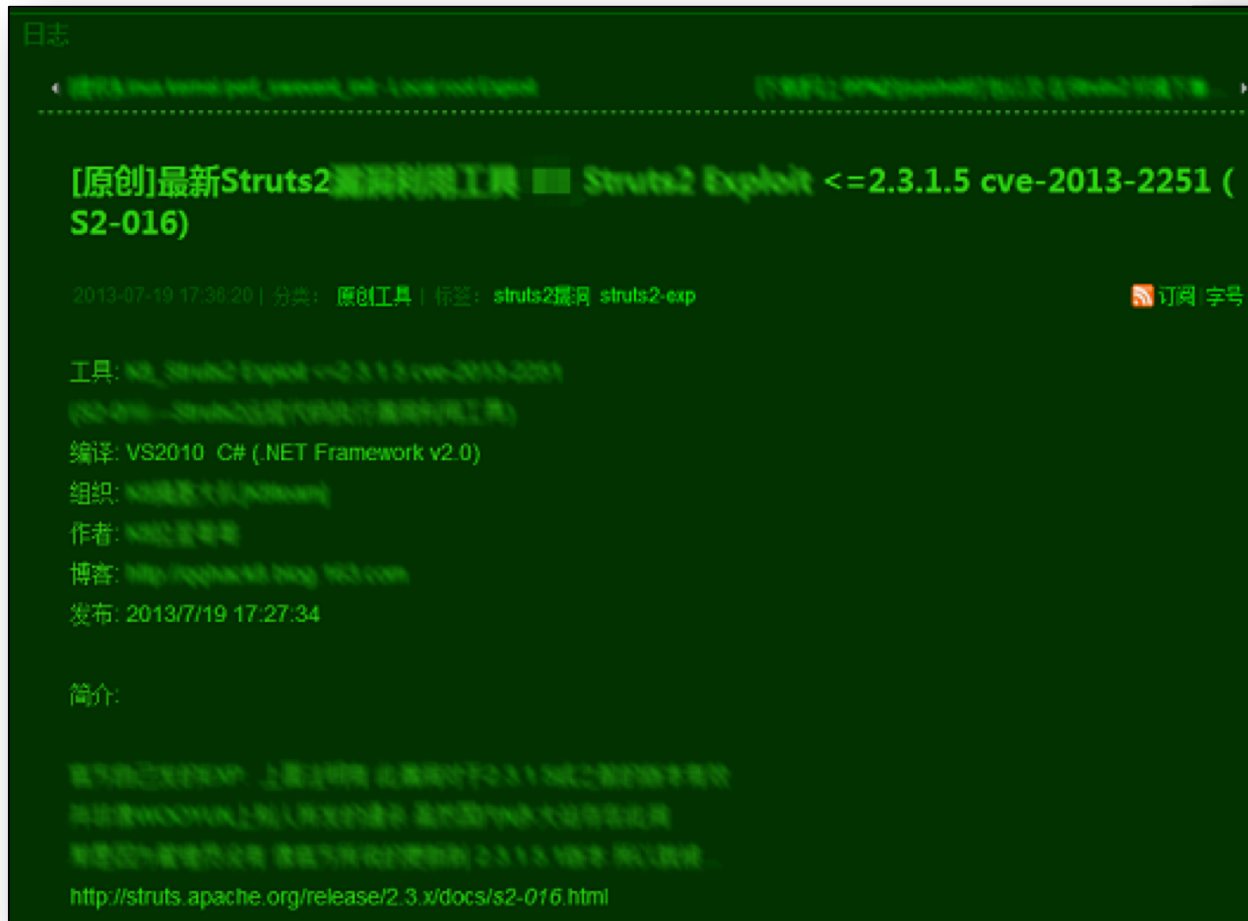
60,660

JavaScript packages
downloaded annually
per developer

30,330

51% with known
vulnerabilities

Widespread Compromise post disclosure



2015 COMMONS COLLECTIONS

CWE-502

23,476,966

total downloads in 2016

18,330,958

78% downloads were vulnerable

<https://wvusoldier.wordpress.com/2016/09/05/some-extra-details-on-hospital-ransomware-you-probably-didnt-know/>

2017 Struts 2: Wait and Prey

March 7

Apache Struts releases updated version to thwart vulnerability CVE-2017-5638

March 9

Cisco observes "a high number of exploitation events."



March 13

Okinawa Power
Japan Post



March '18

India's AADHAAR



April 13

India Post

3 Days in March

The Rest of the Story



March 8

NSA reveals Pentagon servers scanned by nation-states for vulnerable Struts instances

Struts exploit published to Exploit-DB.



March 10

Equifax



Canada Revenue Agency



Canada Statistics



GMO Payment Gateway

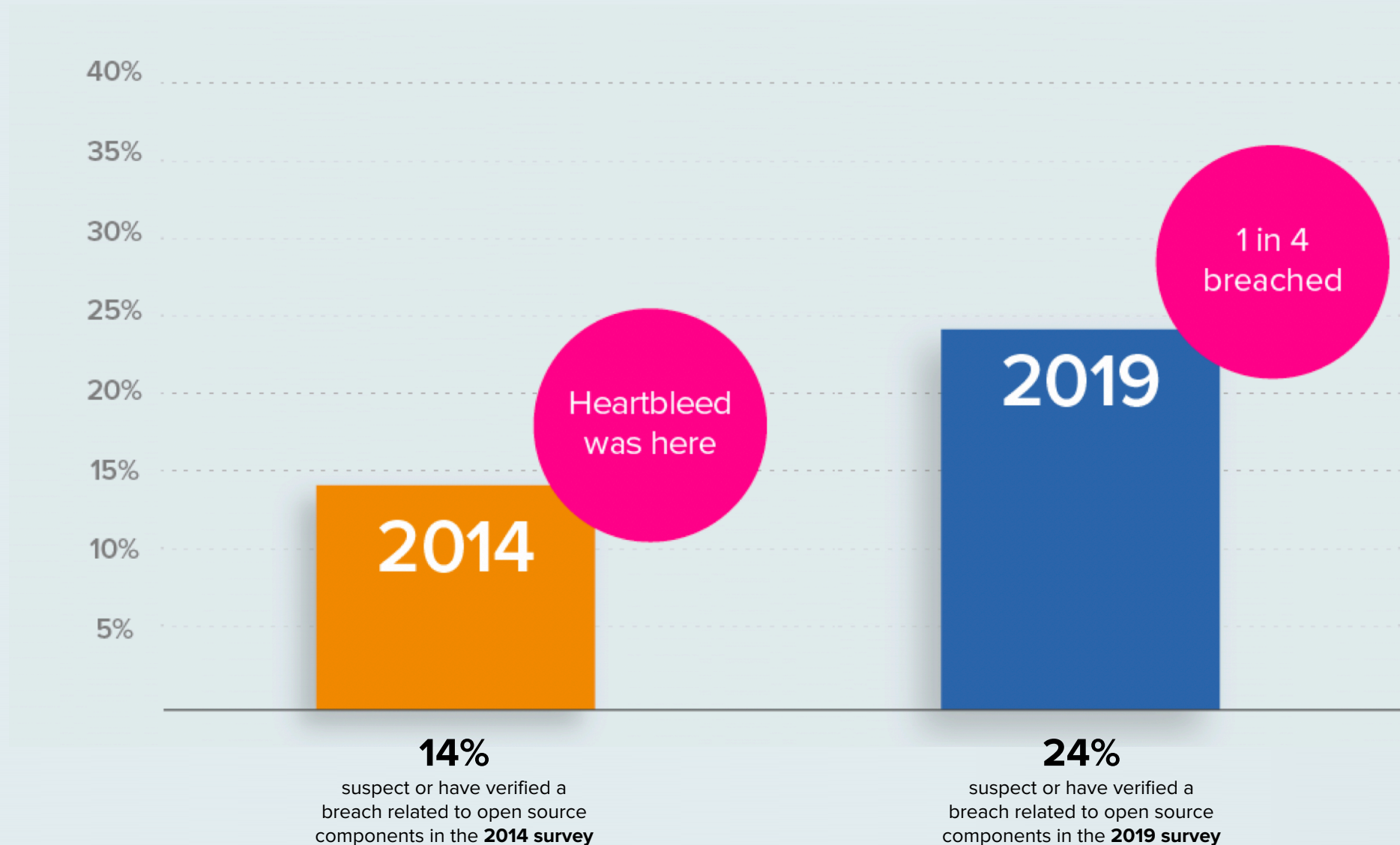
December '17

Monero Crypto Mining

Today

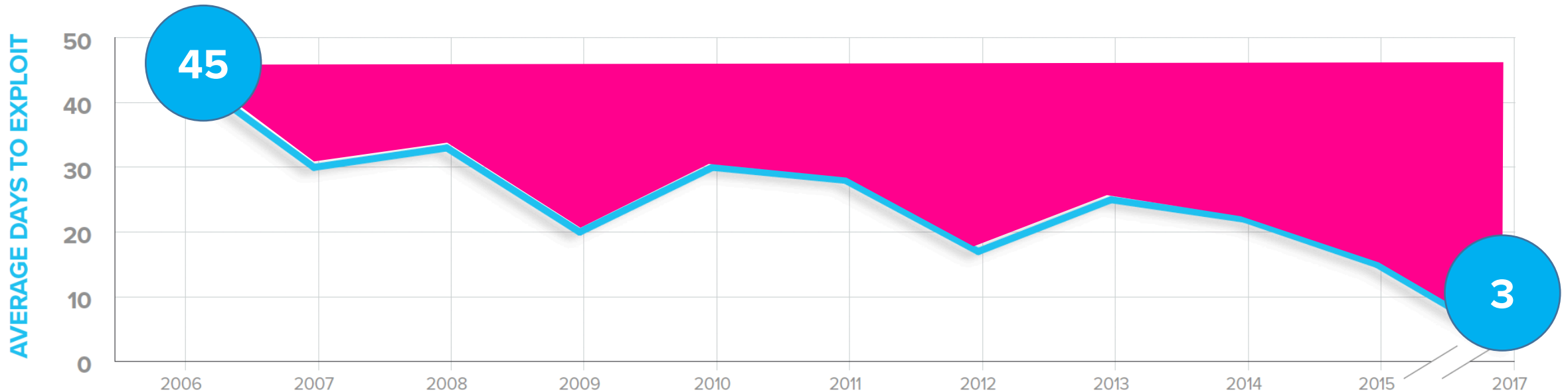
65% of the Fortune 100 download vulnerable versions

Breaches increased 71%



DevSecOps Challenge: Automate Faster than Evil.

Average Days to Exploit



Sources: Garter, IBM, Sonatype

Late 2010's - straight to the source

July 2017

Credentials to 79,000 packages found online, affecting publishing access to 14% of npm repository.

ChALkeR / notes

Sponsor

Watch 98

Star 1,212

Fork 86

<> Code

Issues 0

Pull requests 1

Security

Insights

Branch: master

notes / Gathering-weak-npm-credentials.md



Find file

Copy path

ChALkeR Add bounty information to appropriate notes

5b867f1 on May 12, 2018

3 contributors





327 Lines (249 sloc) 31.6 KB


Raw

Blame

History







Gathering weak npm credentials

Or how I obtained direct publish access to 14% of npm packages (including popular ones).
The estimated number of packages potentially reachable through dependency chains is 54%.

Numbers updated on 2017-07-15 — small update.

In this post, I speak about three ways of gathering credentials — bruteforce attack, known accounts leaks from other sources (not npm), and npm credentials leaks on GitHub (and other places). *The last one was already covered in the [previous post](#), but it's still a valid source nowadays nevertheless.*

Also check out the npm, Inc [blog post](#) about this, if you haven't seen it already.

Warning — if your password was revoked by npm recently, read this

This is not a false alarm — your password being revoked basically means that I was able to obtain it by some of the means described in this note (though neither of those involve npm directly). Basically any other person with an internet access (including malicious players) can also do that.

If you are still using that revoked password anywhere — change it everywhere.


November 2018

npm event-stream attack on CoPay.
2 million downloads per week.

Security

Check your repos... Crypto-coin-stealing code sneaks into fairly popular NPM lib (2m downloads per week)

Node.js package tried to plunder Bitcoin wallets

By [Thomas Claburn](#) in [San Francisco](#) 26 Nov 2018 at 20:58 49  [SHARE](#) ▼



A widely used Node.js code library listed in NPM's warehouse of repositories was altered to include crypto-coin-stealing malware. The lib in question, `event-stream`, is downloaded roughly two million times a week by application programmers.

March 2019

Gems bootstrap-sass RCE backdoor
(1.6K Direct dependencies)

Backdoor code found in popular Bootstrap-Sass Ruby library

Bootstrap-Sass Ruby library had been downloaded more than 28 million times. Backdoored version only 1,470 times.



By Catalin Cimpanu for [Zero Day](#) | April 5, 2019 -- 01:35 GMT (18:35 PDT) | Topic: [Security](#)

The library affected by this incident is [Bootstrap-Sass](#), a Ruby package that provides developers with a [Sass](#)-version of [Bootstrap](#), the most popular UI framework for developers today.

The backdoor's existence came to light [on March 27](#), last week, when software developer Derek Barnes spotted that someone had removed a version of the library (Bootstrap-Sass v3.2.0.2) and immediately released a new version, moments later, v3.2.0.3.

What drew Barnes attention to this version was the fact that the change had only been made on RubyGems, a popular repository for Ruby libraries, but not on GitHub, where the library's source code was being managed.

npm credentials published online.
Affects access to 14% of the npm repo (79,000 packages)

Malicious npm packaged typosquated.
40 packages harvested over two weeks, collecting credentials used to publish to the npm repository itself.

docker123321 images created on Docker Hub.
Later accused of poisoning a Kubernetes honeypot (Jan 2018), and equated to a crypto-mining botnet (May 2018).

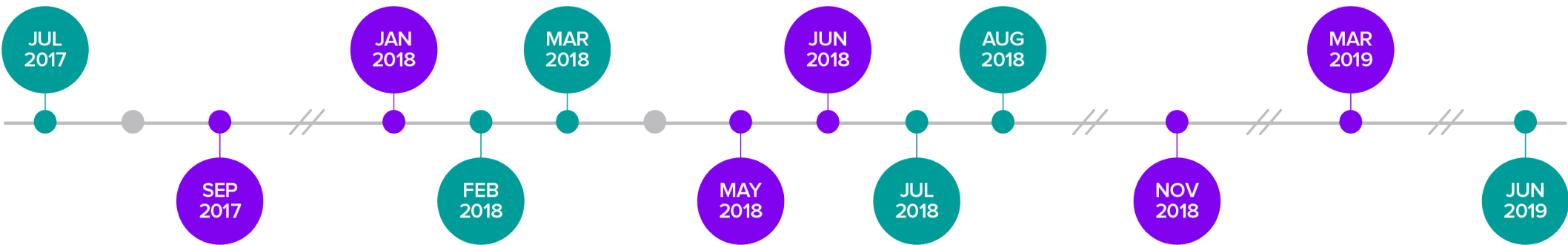
“I’m harvesting credit card numbers and passwords from your site. Here’s how.”
David Gilbertson writes a fictional tale on his blog about creating a malicious npm package.

npm credentials intentionally compromised.
A malicious version of a package from a core contributor to the conventional-changelog ecosystem is published. The package was installed 28,000 times in 35 hours and executed a Monero crypto miner.

Linux distro hacked on GitHub.
Unknown individuals gain control of the Github Gentoo organization, and modified the content of repositories as well as pages within. All code considered compromised.

Homebrew repository compromised.
Accessed in under 30 minutes through an exposed GitHub API token.

Back-doored Gems bootstrap-sass RCE package discovered.
A malicious version of the popular bootstrap-sass package, downloaded a total of 28 million times to date, and with 1.6K dependencies, is published to the RubyGems repository.



PyPI typosquat: 10 malicious Python packages found.
Evidence of the fake packages being incorporated into software was noted multiple times between June and Sept 2017.

Deleted go-bindata account resurrected by an unknown user.
After a developer deleted their GitHub account, someone immediately grabbed the ID — inheriting the karma instilled in that id and calling into question packages and sources.

Back-doored PyPI package discovered.
Python module ssh-decorator back-doored to enable theft of private ssh keys.

Back-doored npm package discovered.
npm security team responds to reports of a malicious back door in the get-cookies module, published in March. Despite being deprecated, mailparser still receives about 64,000 weekly downloads.

Compromised JavaScript package caught stealing npm credentials.
A hacker gains access to a developer’s npm account and injects malicious code into a popular JavaScript library called eslint-scope, a sub-module of the more famous ESLint, a JavaScript code analysis toolkit.

Malicious package injected into event-stream, a popular npm package.
The injected code targets the Copay application and was designed to harvest account details and private keys from accounts having a balance of more than 100 Bitcoin or 1,000 Bitcoin Cash.

Cryptocurrency attack via malicious code injection.
Malicious code targets users of a cryptocurrency wallet called Agama, focusing on getting into the build chain and stealing the wallet seeds and other login passphrases used within the application.

A Shifting Battlefront of Attacks: Malicious Code Injection

July 2017 – June 2019

Crypto Currency: Cybercrime's new best friend.

“I have nothing of value in my application”

Your **server** has CPU cycles

Your **visitors** have CPU cycles

Your **build infra** has CPU cycles

Crypto Currency allows the attack to be directly monetized.



Jenkins under attack



Jenkins Miner: One of the Biggest Mining Operations Ever Discovered

February 15, 2018

The Check Point research team has discovered what could potentially become one of the biggest malicious mining operations ever seen.

“So far, \$3.4 million has been mined.”



It affects all of us.
How do we fight it?



...faster is better
in the enterprise

...faster is better
for open source.

2019 State of the Software Supply Chain

The 5th annual report on global
open source software development

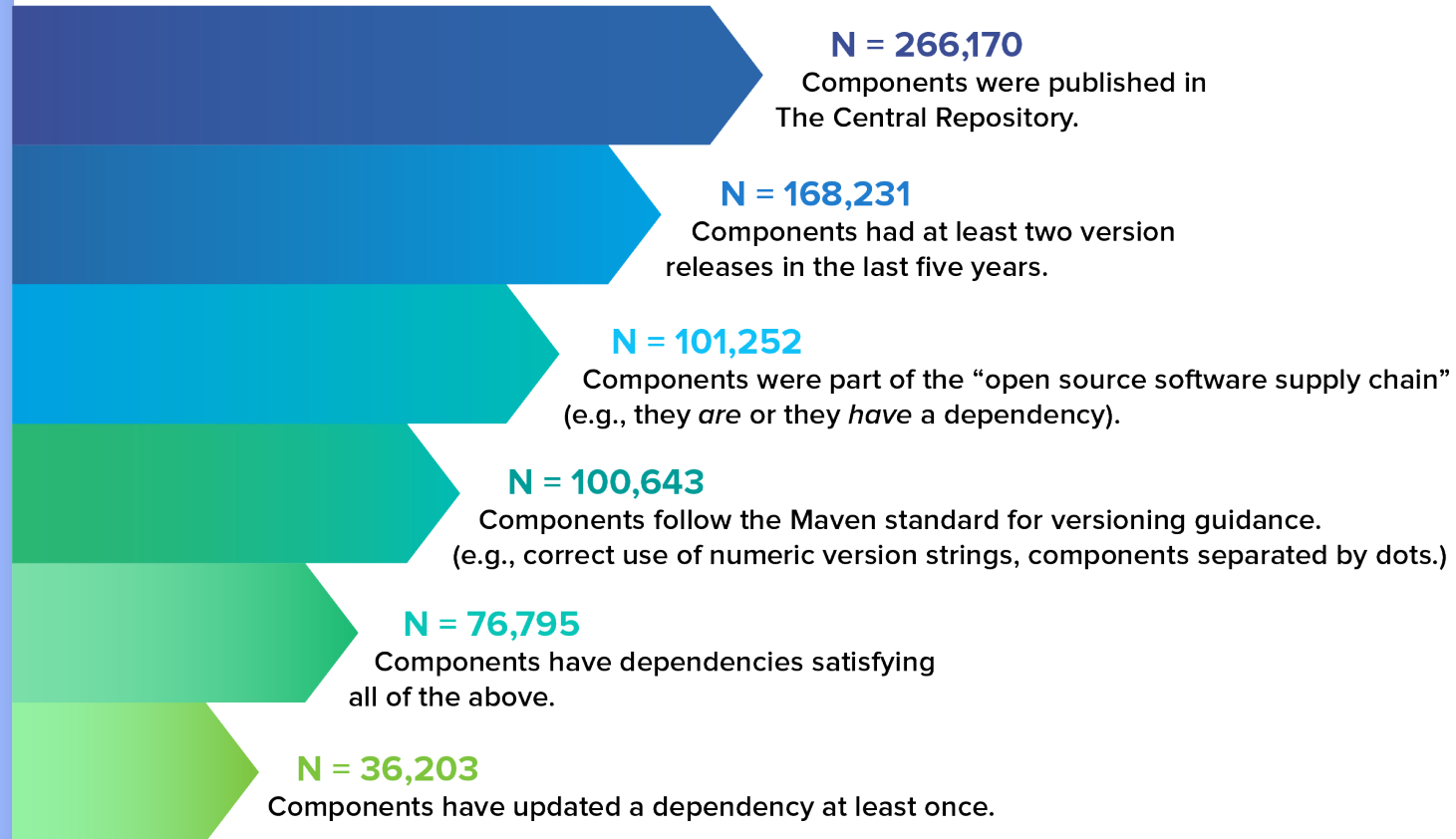
presented by




in partnership with



Constructing the Study Dataset (N = 36,203)





Attributes	Measure
Popularity	Avg. daily Central Repository downloads
Size of Team	Avg. unique monthly contributors
Development Speed	Avg. commits per month
Release Speed	Avg. period between releases
Presence of CI	Presence of popular cloud CI systems
Foundation Support	Associated with an open source foundation
Security	More complicated
Update Speed	More complicated

Assumption # 1

Projects that release frequently have better outcomes.



1945: W. Edwards Deming

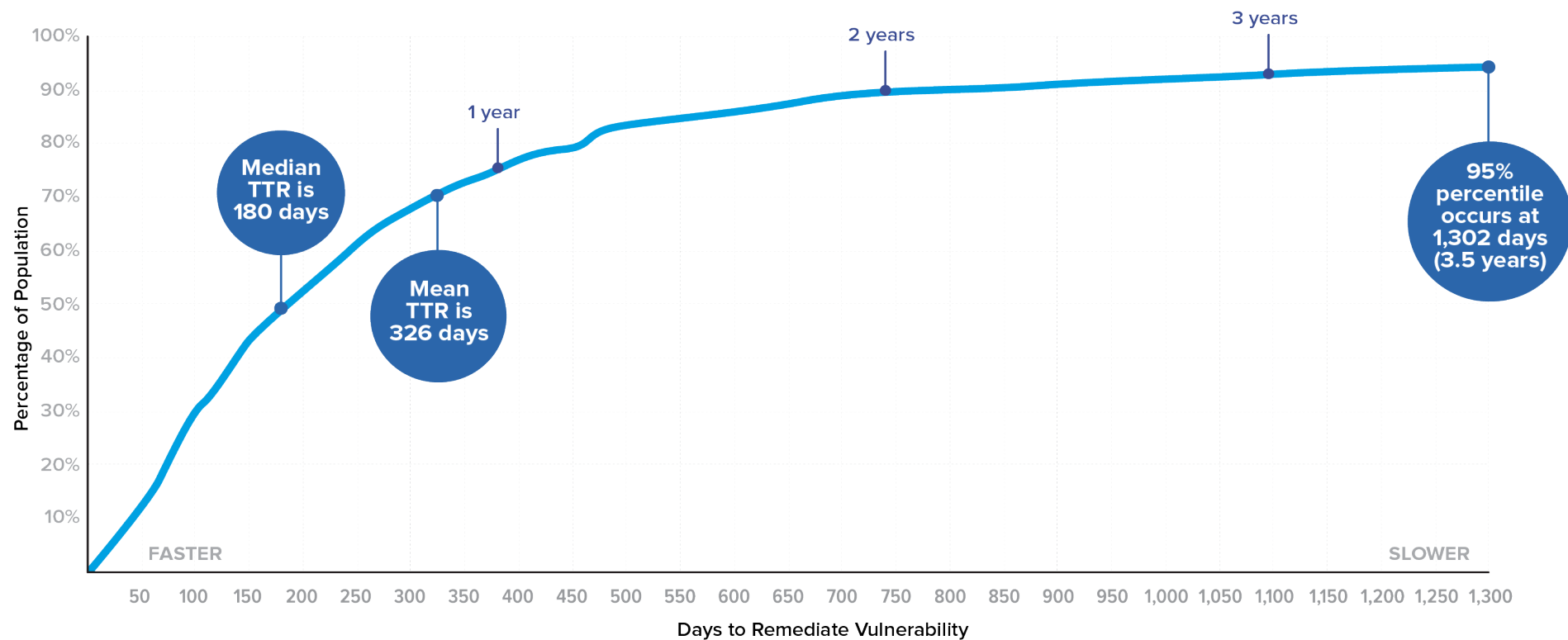
The Key Metrics:

Time to Remediate

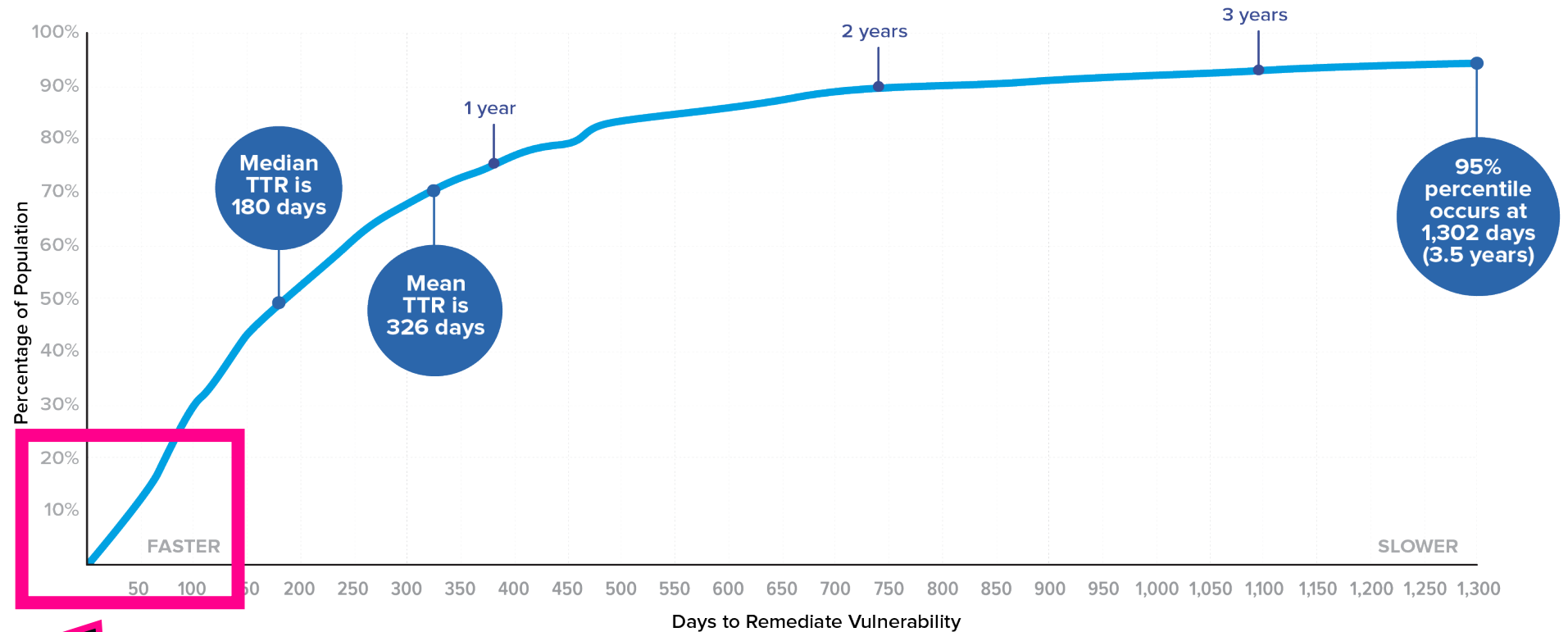
Time to Update

Stale Dependencies

Time to Remediate Vulnerabilities

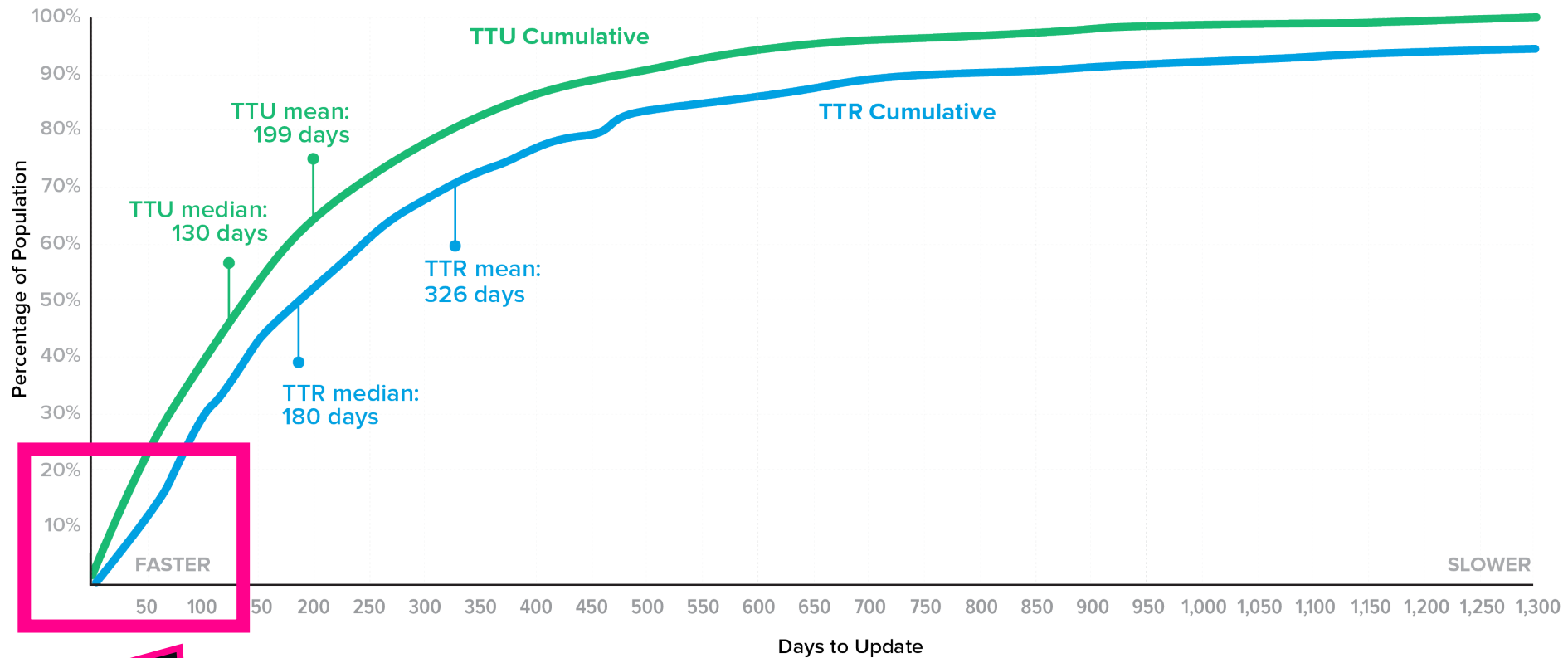


Time to Remediate Vulnerabilities



Do these update quickly in general?

Time to Remediate (TRR) vs. Time to Update (TTU)



Most projects stay secure by staying up to date.

Projects that release frequently:

are 5x more popular.

attract 79% more developers.

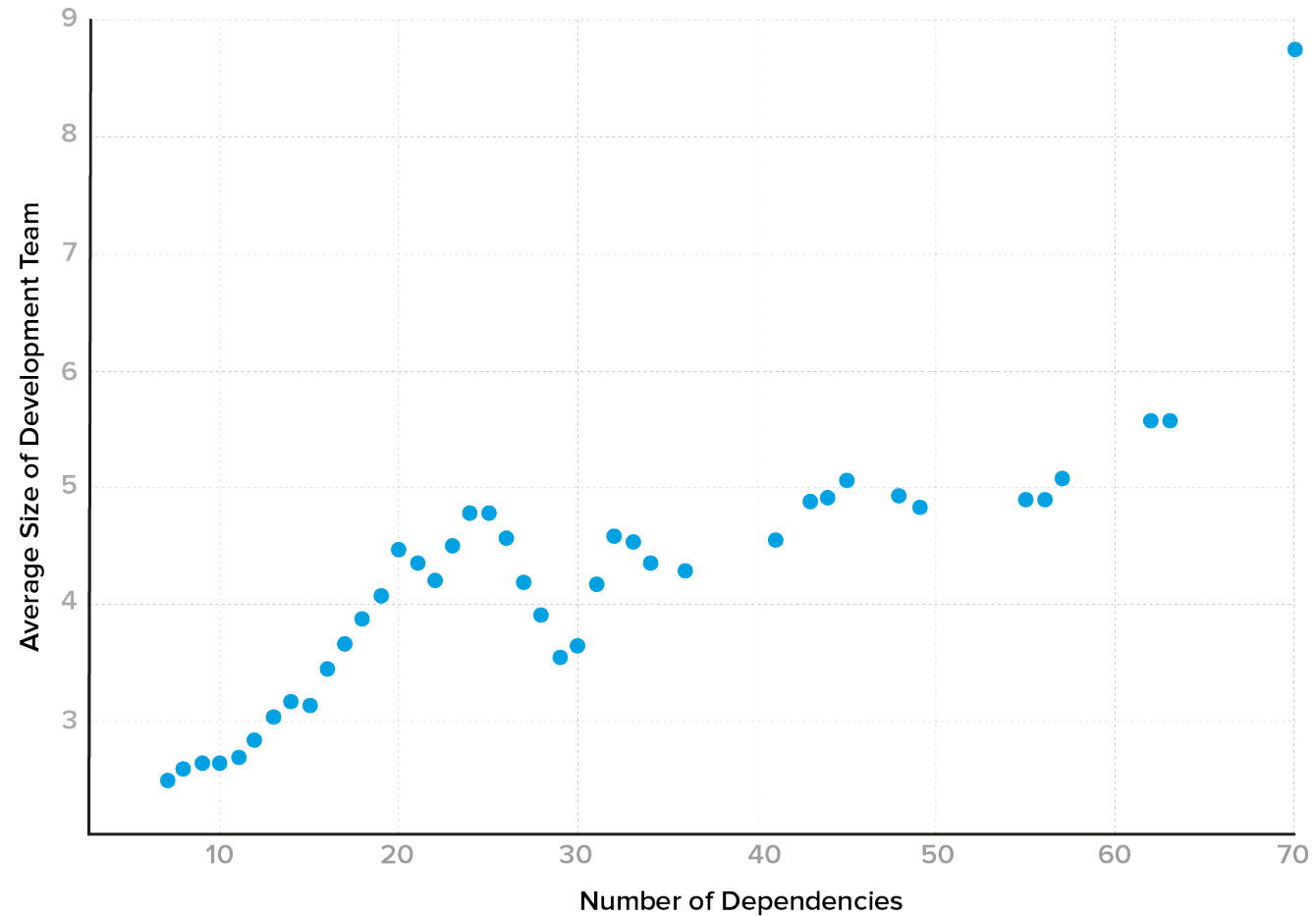
have 12% greater foundation support rates.

Assumption 2

Projects with fewer dependencies will stay more up to date.

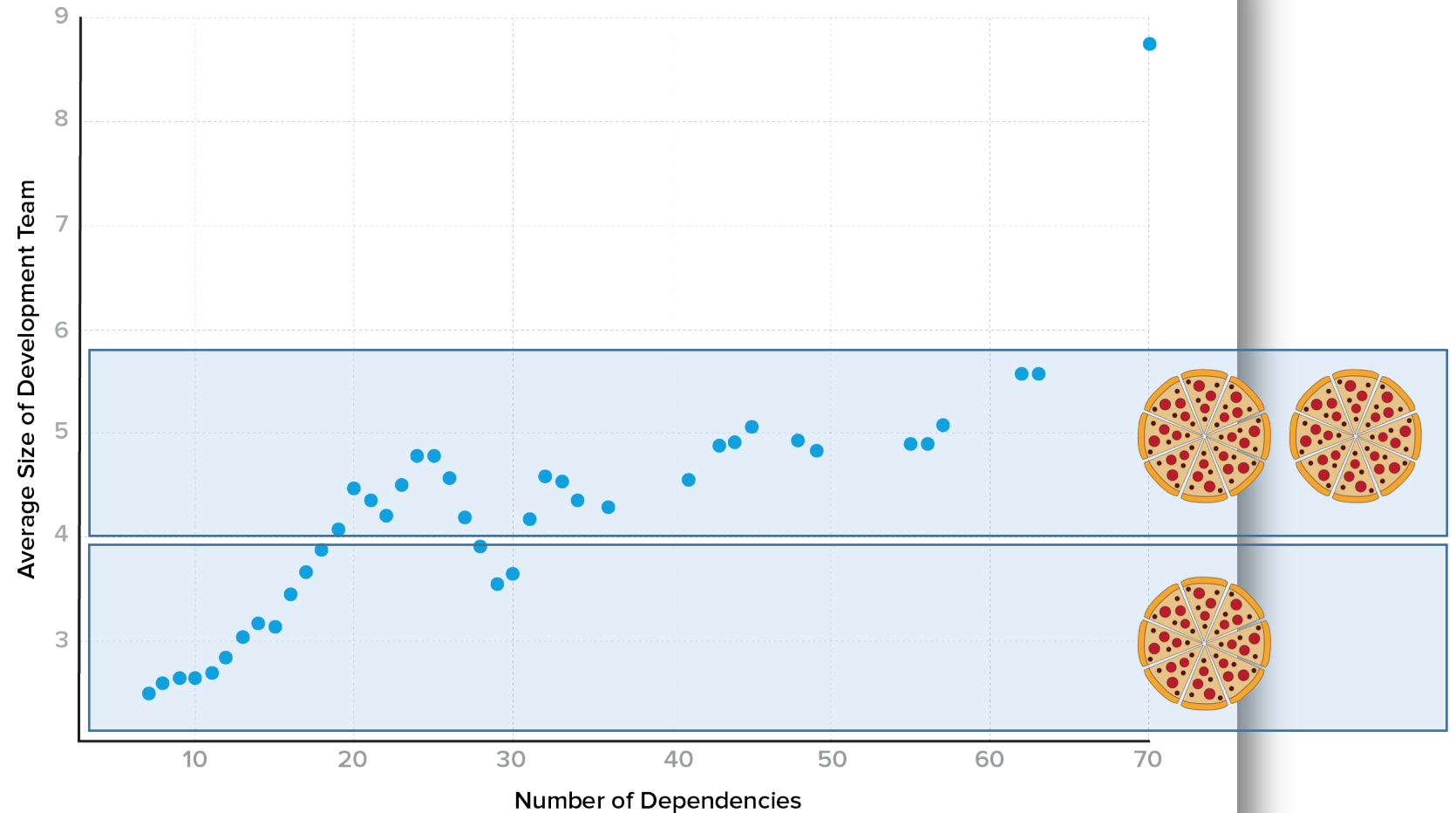
More dependencies
correlate with larger
development teams.

**Larger development
teams have 50%
faster MTTU and
release 2.6x more
frequently.**



More dependencies
correlate with larger
development teams.

**Larger development
teams have 50%
faster MTTU and
release 2.6x more
frequently.**



Projects with fewer dependencies will stay more up to date.

(REJECTED)

Components with more dependencies actually have better MTTU.

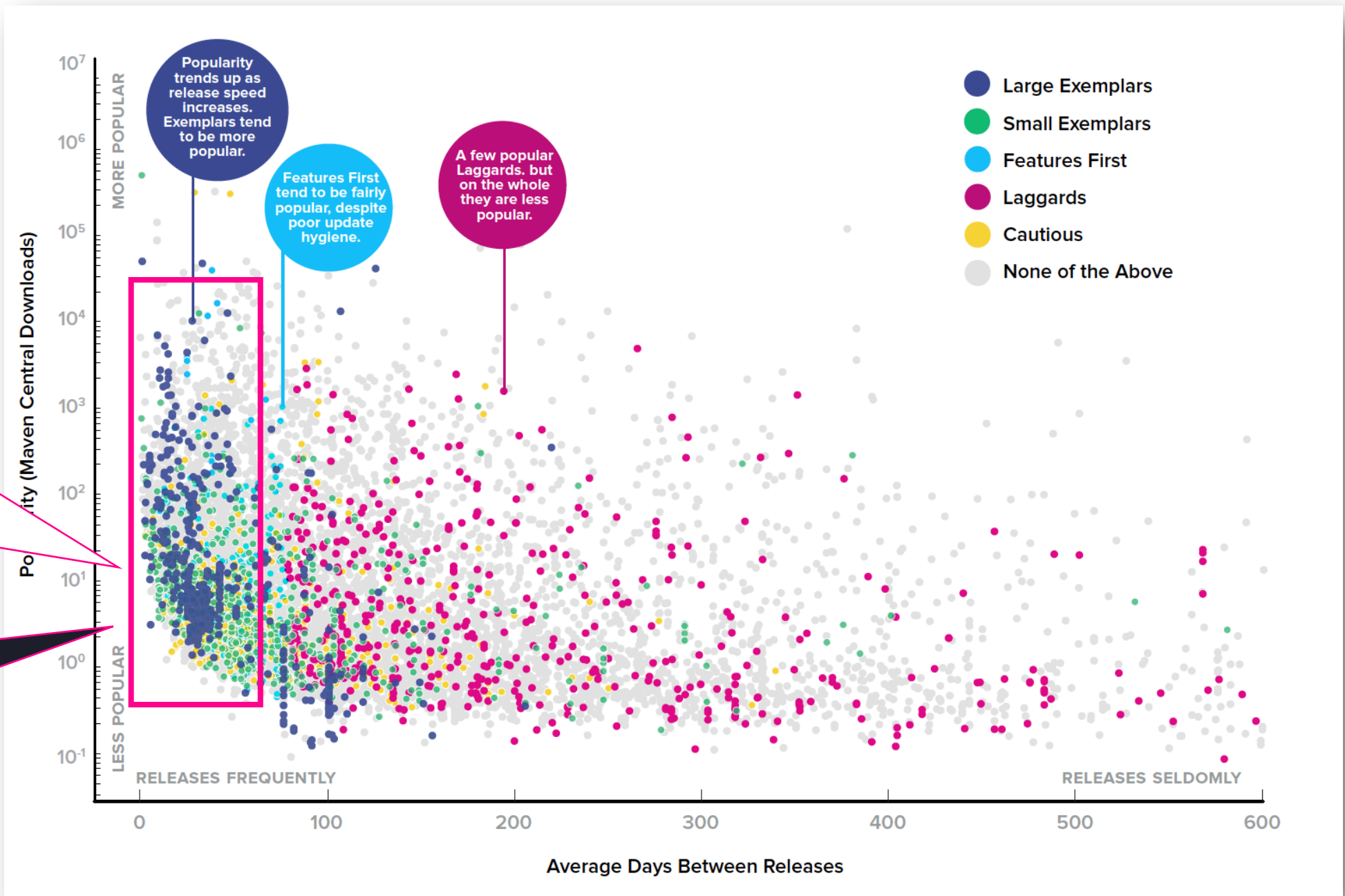
Assumption 3

More popular projects will be better about staying up to date.

5 Behavioral Clusters

Small Exemplar (606)	Large Exemplar (595)	Laggards (521)	Features First (280)	Cautious (429)
Small development teams (1.6 devs), exemplary MTTU.	Large development teams (8.9 devs), exemplary MTTU, very likely to be foundation supported, 11x more popular.	Poor MTTU, high stale dependency count, more likely to be commercially supported.	Frequent releases, but poor TTU. Still reasonably popular.	Good TTU, but seldom completely up to date.

Rest of the population: 8,142



Exemplars release fast and tend to be more popular.

Pick suppliers from here.



Assumption 3

More popular projects will be better about staying up to date.

(REJECTED)

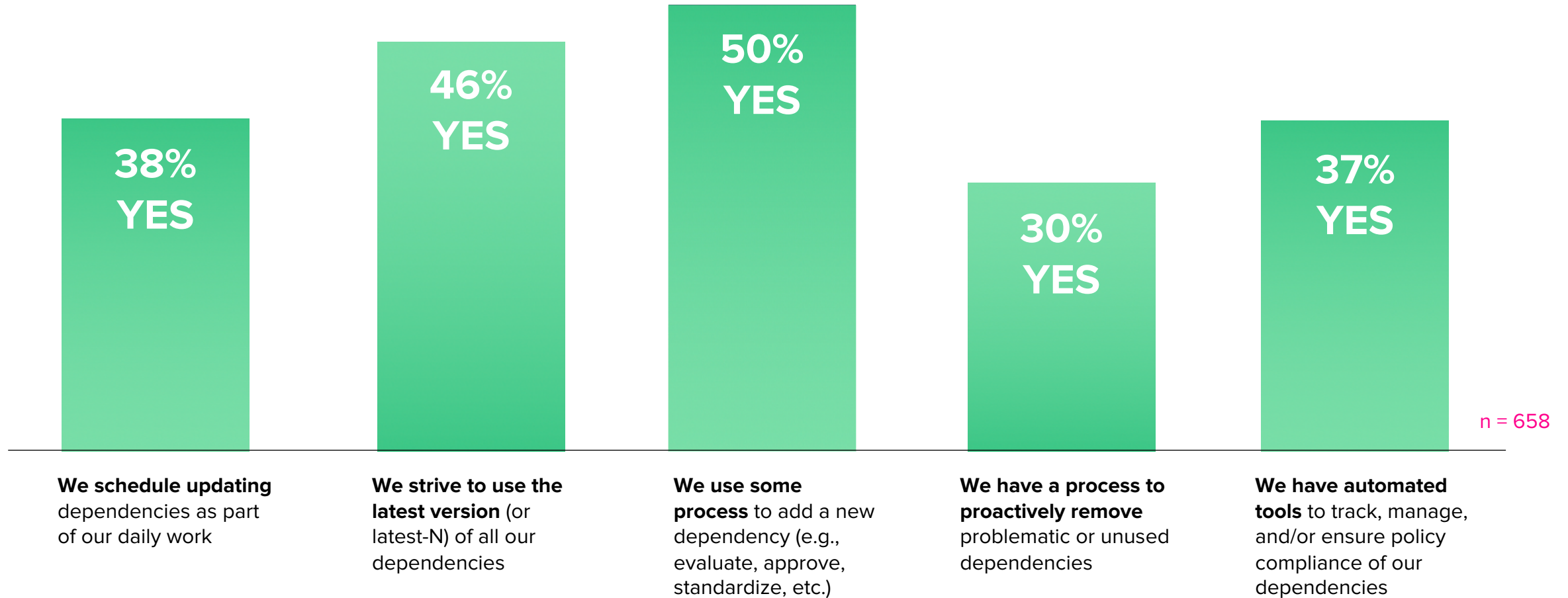
There are plenty of popular components with poor MTTU.

Popularity does not correlate with MTTU.



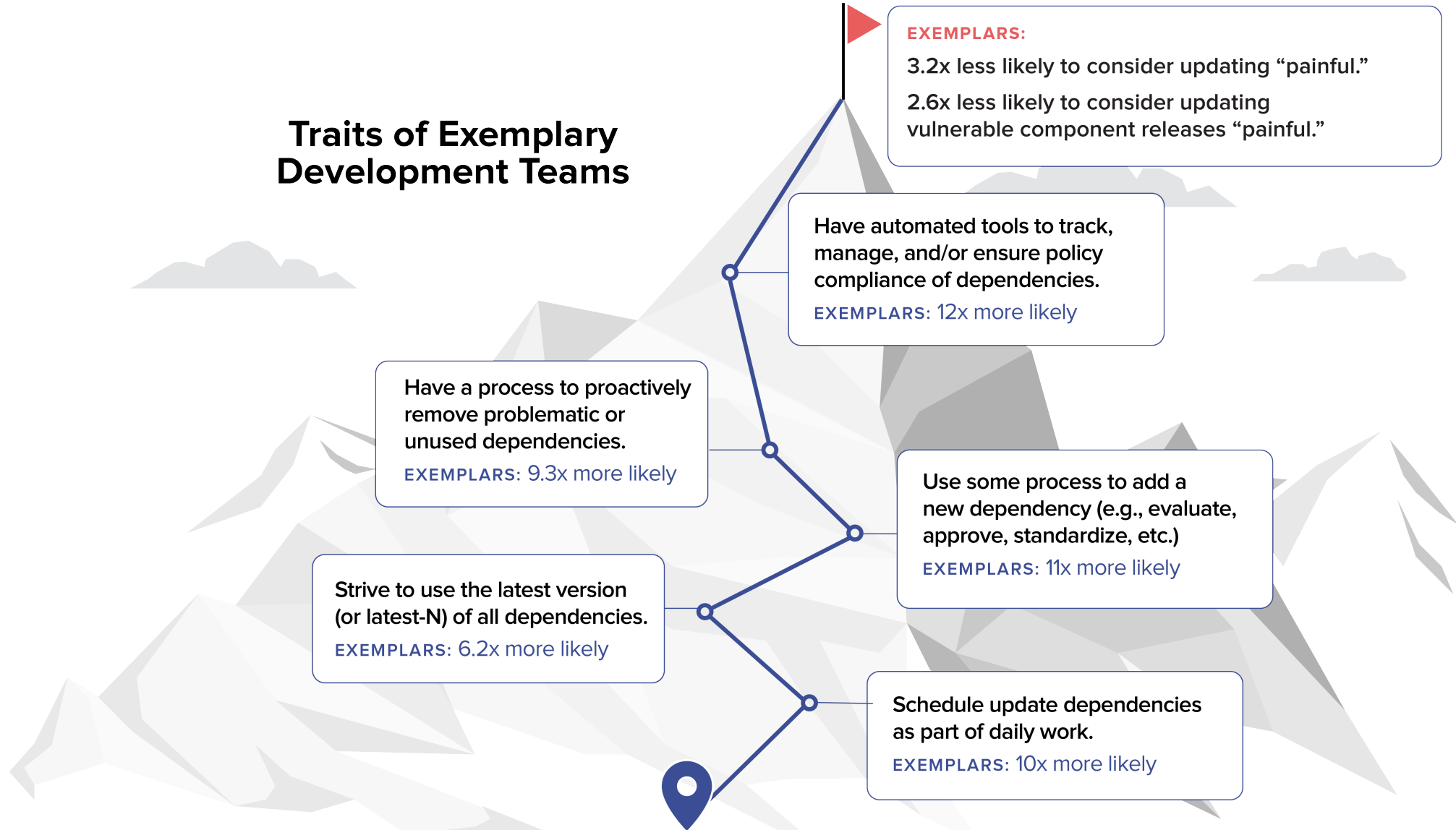
How do we stay fast?

Enterprise Devs Manage Dependencies



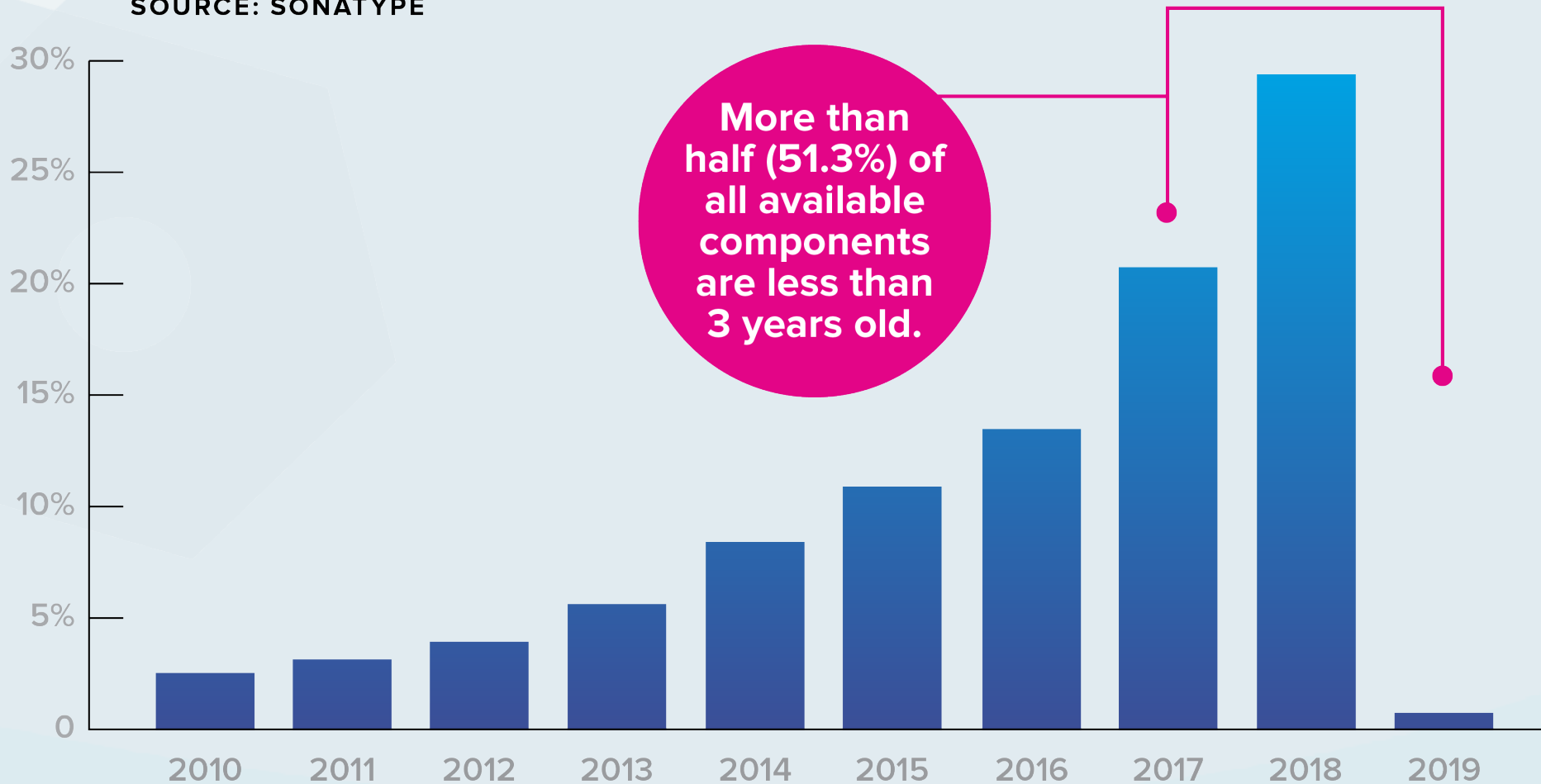
When Devs climb the mountain every day, it's easier.

Traits of Exemplary Development Teams

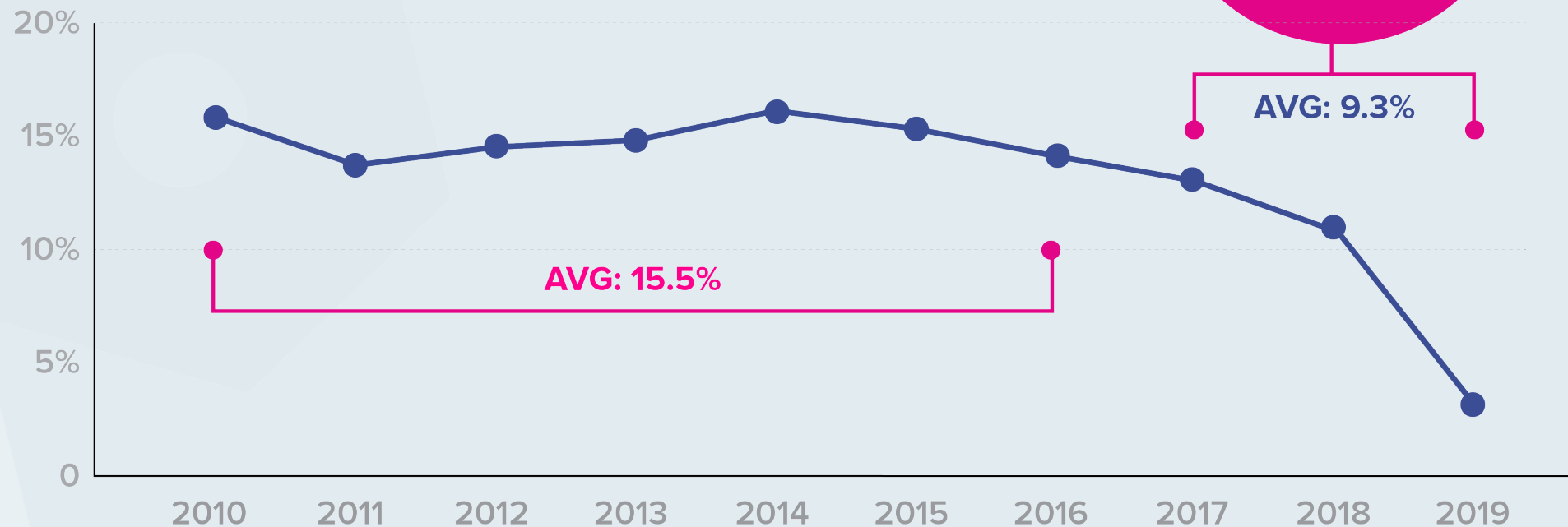


Age of Components Used in Managed Software Supply Chains (Analysis of Java Components Across 68,000 Applications)

SOURCE: SONATYPE

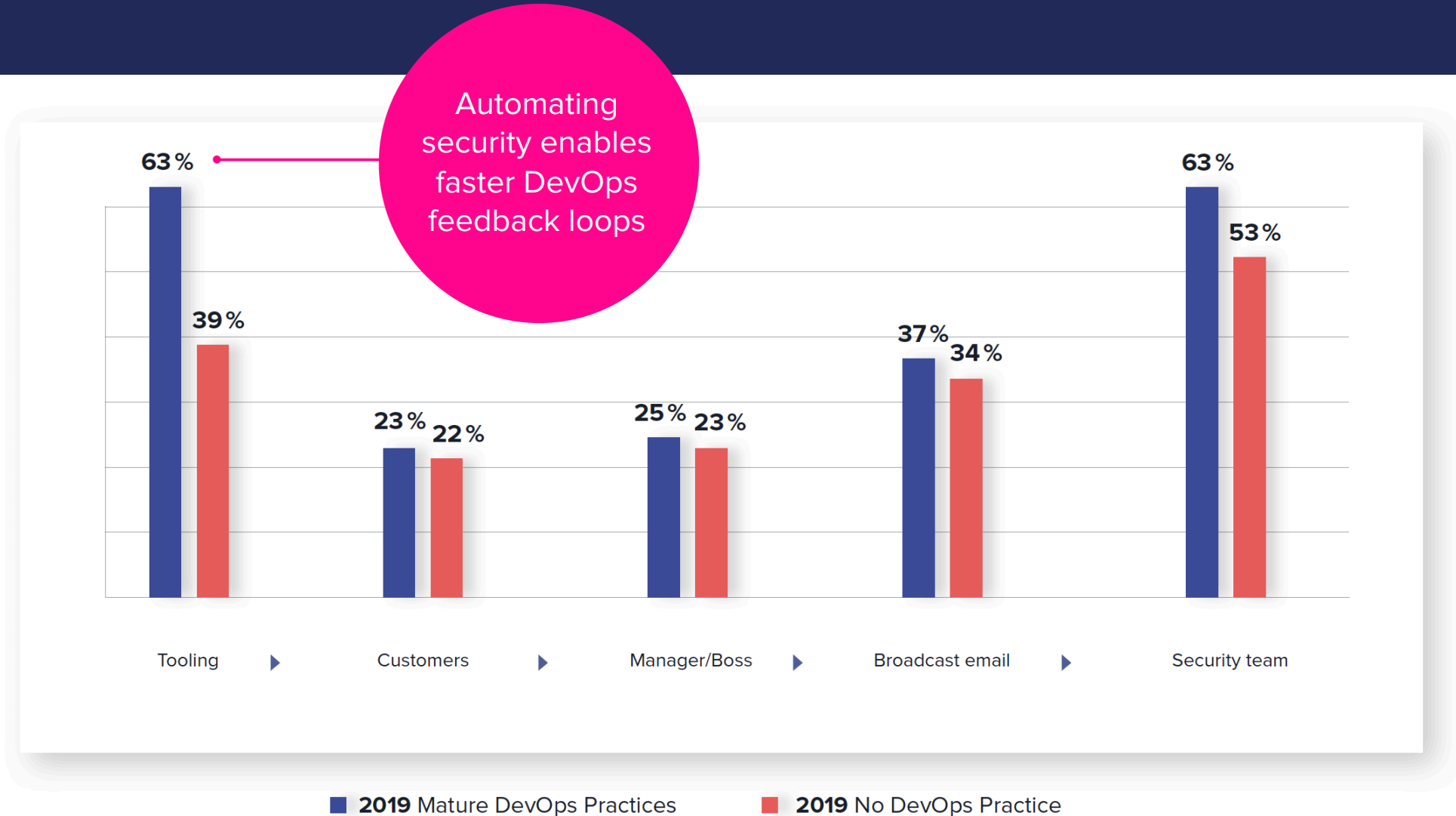


Percentage of Components with Known Vulnerabilities



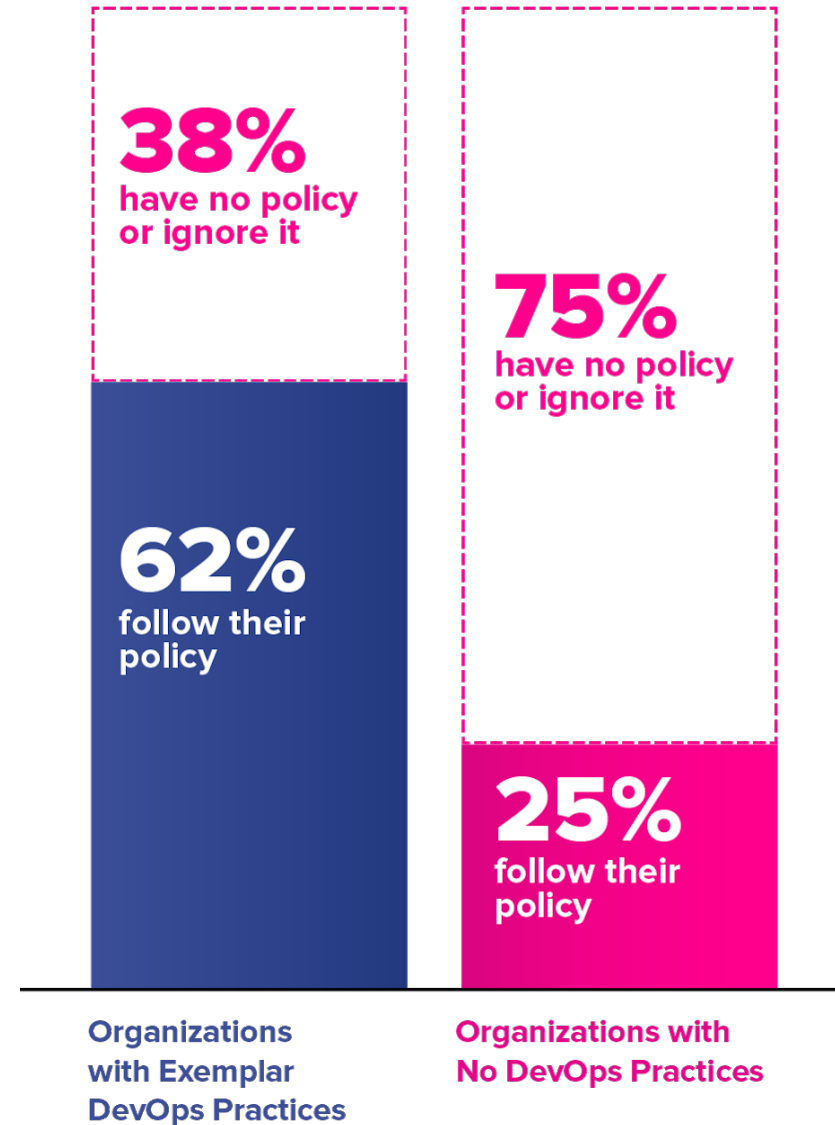
Components less than 3 years old have 65% fewer known vulnerabilities.

How are you informed of InfoSec and AppSec issues?

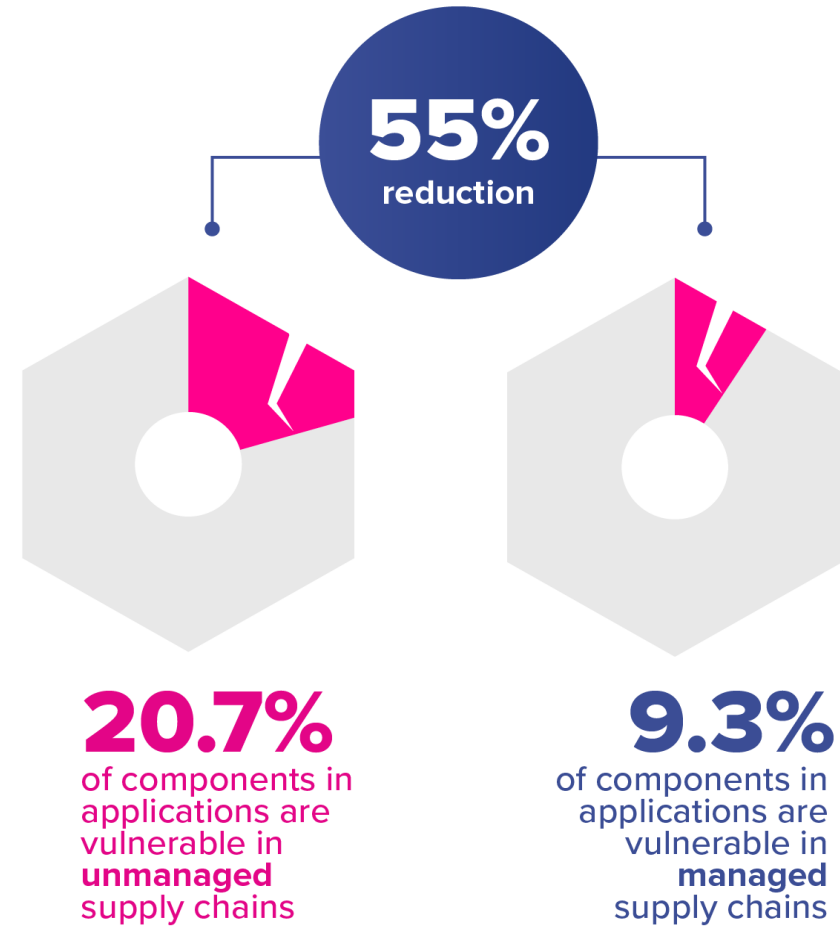


**Automation
continues to prove
difficult to ignore.**

Do you have an open source
policy and do you follow it?



For organizations who tamed their supply chains, the rewards were impressive.



Manage the 85% of your software


***Be faster
than your adversaries***



Set standards for what you choose

Automate it all.

2019 State of the Software Supply Chain

The 5th annual report on global
open source software development

presented by
 **sonatype**

in partnership with
 **galois** |  **IT REVOLUTION**

iturunen@sonatype.com