OPEN SOURCE STUDY NIGHT

MICHAELA (MIKI) DEMETER

CHOOSING MORE SECURE OPEN SOURCE PACKAGES: LESSONS FROM THE REAL WORLD

How well do you know what’s inside your projects?

AS A SECURITY PROFESSIONAL YOU ARE EXPECTED TO KNOW WHAT IS IN YOUR PRODUCT

“I AM INVOLVED IN EXAMINING OPEN SOURCE PROJECTS AND THIRD PARTY COMPONENTS FOR "KNOWN GOOD" DEVELOPMENT PRACTICES.”

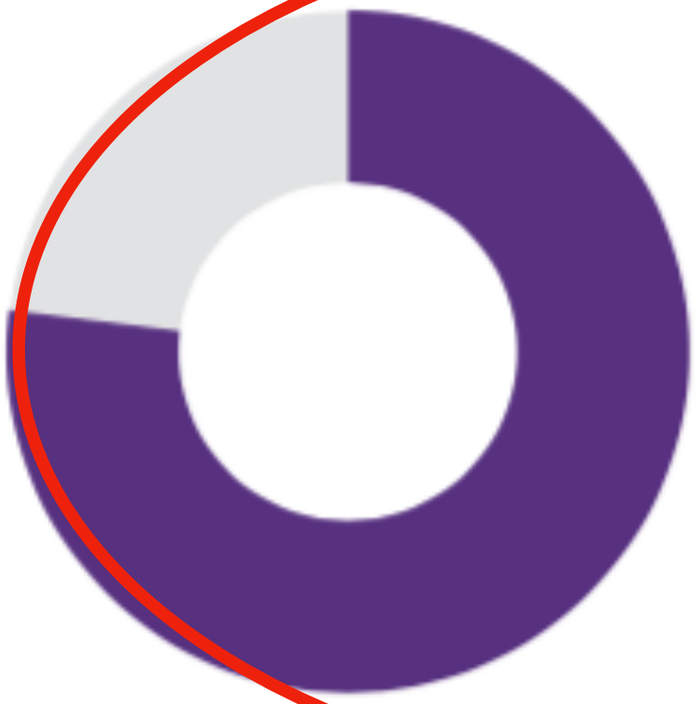
THERE'S A BIG DIFFERENCE BETWEEN GOOD, COMMUNITY-SUPPORTED OPEN SOURCE SOFTWARE AND RANDOM CODE FOUND ON THE INTERNET.

Black Duck On-Demand audits found open source components in **96%** of the applications scanned, with an average **257** components per application.

The average percentage of codebase that was open source was **57%** vs. **36%** last year. Many applications now contain more open source than proprietary code.

78% of the codebases examined contained at least one vulnerability, with an average **64** vulnerabilities per codebase.

YIKES



Of the IoT applications scanned, on average **77%** of the codebase was comprised of open source components, with an average **677** vulnerabilities per application.

I WANT TO ENABLE YOU TO SHIP MORE SECURE BETTER QUALITY OPEN SOURCE SOFTWARE.

HOW DO I CHOOSE ~~GOOD~~ OPEN SOURCE PACKAGES?

HOW DO I CHOOSE SECURE OPEN SOURCE PACKAGES?

STEP 1:
TAKE A LOOK

https://clipartxtras.com/categories/view/24dc10f8c0e963e9168b2fcab20d51d21feb8bff/take-a-look-clipart.html

- READ THE README,,OR ANY OTHER READILY AVAILABLE INTRODUCTORY INFORMATION?
- DOES THIS CODE APPEAR TO BE HELD TO GOOD SOFTWARE DEVELOPMENT STANDARDS?
- IS THIS CODE USED PROFESSIONALLY OR IS IT A HOBBY PROJECT?
- ARE THERE ANY SIGNS THAT THERE ARE KNOWN ISSUES WITH THIS CODE?
- DOES THIS CODE ONLY SOLVE ONE USE CASE OR IS IT ROBUST ENOUGH FOR OTHER USE CASES?
- IS THIS CODE ACTIVE OR AN ARCHIVE ESSENTIALLY ABANDONED?

LOOK FOR **WARNING SIGNS**.

SOME KEY QUESTIONS FOR A FIRST LOOK AT A NEW PACKAGE

IS IT

IN OTHER WORDS...

LET'S LOOK AT SOME WARNING SIGNS

http://chittagongit.com/icon/high-risk-icon-21.html

"use TweetNaCl.js (a TweetNaCl port to JavaScript) rather than this implementation, which is more likely to perform in constant time and has likely seen more eyes for review/audits."

https://github.com/andi506/crypto-js

https://github.com/01org/IntelRackScaleArchitecture

**********DISCLAIMER**********

This code is reference software only and is not feature complete. **It should not be used in commercial products at this time.** Intel makes no claims for the quality or completeness of this code.

EVEN THE DEVELOPERS SAY TO USE SOMETHING ELSE...

"I didn't write this code but I like it."

https://github.com/kbranigan/cJSON

NICE TO KNOW!!!!

My Drive - Google Drive | Untitled presentation - Goo | Google Code

https://code.google.com

Apps | Integrity Measureme | WAC Core Specificat | NCCoE - Home | Build System Info | k | Clear Sharepoint | OTC ClearLinux Wiki | Microsoft Lync Web | Safe - Doc

Google Developers

Looking for Google APIs and Tools?

Google Developers is now the place to find all Google developer documentation, resources, events, and products.

developers.google.com

Project Hosting

Google Code Project Hosting offered a free collaborative development environment for open source projects.

In 2016 the service was shut down, see this post for more info. Projects hosted on Google Code remain available in the Google Code Archive.

Shut down in January 2016

CPAN

Home · Authors · Recent · News · Mirrors · FAQ · Feedback

in All ▾ CPAN Search

Michael Firestone > Tivoli-AccessManager-Admin permalink

Tivoli-AccessManager-Admin

This Release	Tivoli-AccessManager-Admin-1.11 [Download] [Browse] 13 Dec 2006 ** UNAUTHORIZED RELEASE **
Other Releases	Tivoli-AccessManager-Admin-1.10 -- 13 Dec 2006 ▾ Goto
Links	[Discussion Forum] [View/Report Bugs] [Dependencies] [Other Tools]
CPAN Testers	PASS (2) FAIL (20) UNKNOWN (403) [View Reports] [Perl/Platform Version Matrix]
Rating	☆☆☆☆☆ (0 Reviews) [Rate this distribution]
License	unknown
Special Files	CHANGES MANIFEST README Makefile.PL META.yml

Modules

Tivoli::AccessManager::Admin	UNAUTHORIZED	1.11
Tivoli::AccessManager::Admin::ACL		1.11
Tivoli::AccessManager::Admin::Action		1.11
Tivoli::AccessManager::Admin::AuthzRule		1.11
Tivoli::AccessManager::Admin::Context		1.11

DOES ANYONE REALLY LOOK TO SEE WHAT THIS MEANS?

https://www.gograph.com/vector-clip-art/nope.html

“CryptoJS is a project that I enjoy and work on in my spare time, but unfortunately my 9-to-5 hasn't left me with as much free time as it used to. I'd still like to continue improving it in the future, but I can't say when that will be.”

https://code.google.com/archive/p/crypto-js/

“opencsv was developed in a couple of hours”

http://opencsv.sourceforge.net/


PRO SERIES

> “[This code is] slower and more subjective to side-channel attacks by nature”

http://www.literatecode.com/aes256

> “cJSON aims to be the dumbest possible parser that you can get your job done with. ”

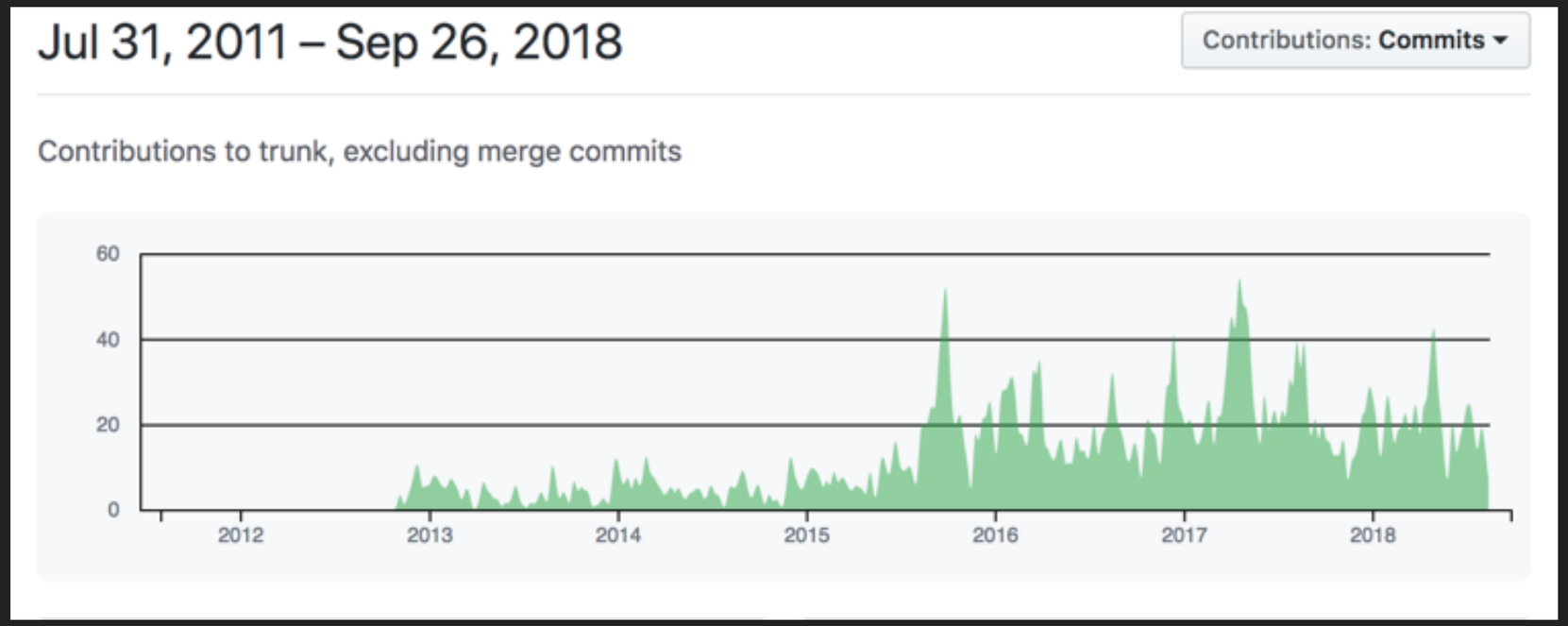
https://github.com/kbranigan/cJSON/commit/730209a718cc9bada631cea136d13017752720f5

WHAT WOULD WE LIKE TO SEE?

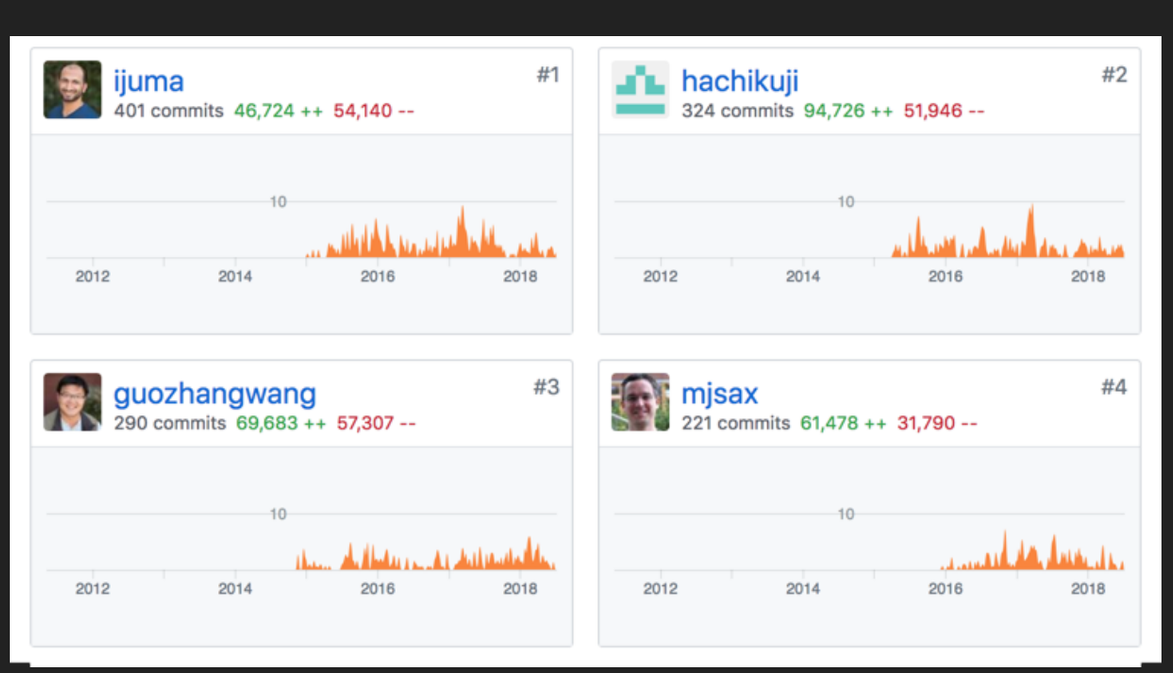
STEP 2:
CHECK THE CONTRIBUTORS AND COMMUNITY

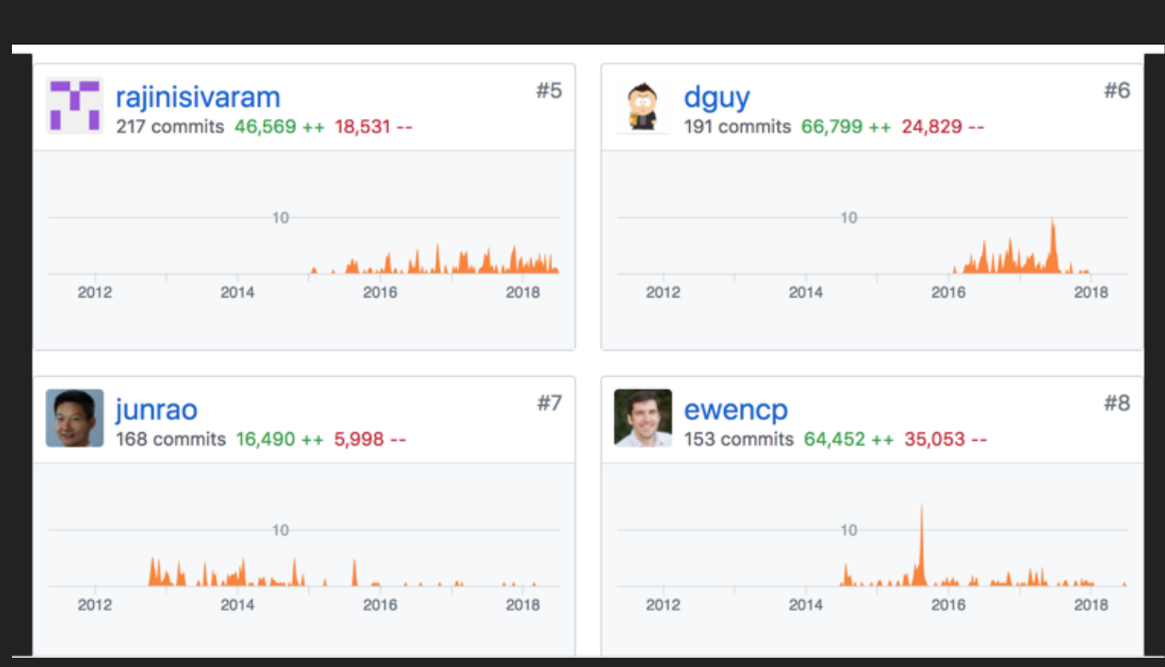
- HOW MANY ACTIVE AND SIGNIFICANT CONTRIBUTORS?
- IS THIS CODE ACTIVELY MAINTAINED OR IS IT ABANDONED?
- HOW MANY PULL REQUESTS & CHECKINS IN THE PAST YEAR?
- ARE ISSUES FIXED AND RELEASED ON A REGULAR BASIS?
- WHO SIGNS OFF ON CODE REVIEWS?
- IS THERE MORE THAN ONE MAINTAINER?

KEY QUESTIONS ABOUT CONTRIBUTORS

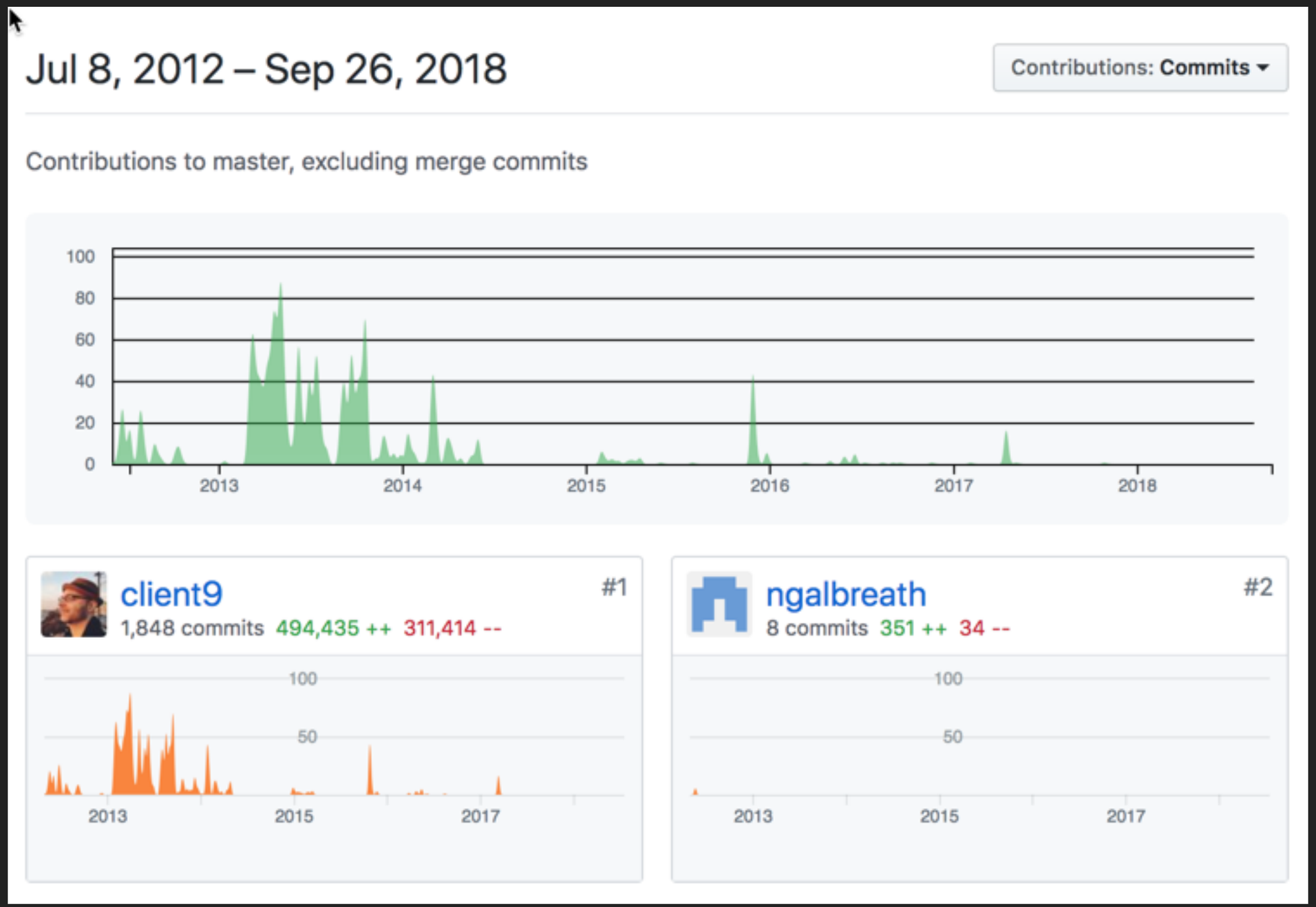


Good example: many active contributors





A good example: one significant contributor and not recently active



STEP 3:
CHECK HOW THEY HANDLE VULNERABILITIES

- IS THERE A CLEAR WAY TO REPORT SECURITY VULNERABILITIES?
- IS THERE EVIDENCE THAT VULNERABILITIES ARE FIXED IN A TIMELY MANNER?
- IS THERE ANY EXPLANATION OF WHAT HAPPENS WHEN A SECURITY ISSUE IS REPORTED?

KEY QUESTIONS ABOUT HANDLING SECURITY ISSUES

THE APACHE SECURITY TEAM

The Apache Security Team exists to provide help and advice to Apache projects on security issues and to provide co-ordination of the handling of security vulnerabilities. All members of the Security Team are also members of the Apache Software Foundation.

REPORTING A VULNERABILITY

We strongly encourage folks to report security vulnerabilities to one of our private security mailing lists first, before disclosing them in a public forum.

A list of security contacts for Apache projects is available. If you can't find a project specific security e-mail address and you have an undisclosed security vulnerability to report then please use the general security address below.

Please note that the security mailing lists should only be used for reporting undisclosed security vulnerabilities in Apache products and managing the process of fixing such vulnerabilities. We cannot accept regular bug reports or other security related queries at these addresses. All mail sent to these addresses that does not relate to an undisclosed security problem in an Apache product will be ignored.

Also note that the security team handles vulnerabilities in Apache products, not running ASF services. All reports of vulnerabilities in running ASF services should be sent to root@apache.org only.

The general security mailing list address is: security@apache.org. This is a private mailing list and only members of the Apache Security Team are subscribed.

HTTPS://WWW.APACHE.ORG/SECURITY/

GREAT EXAMPLE:

VULNERABILITY HANDLING

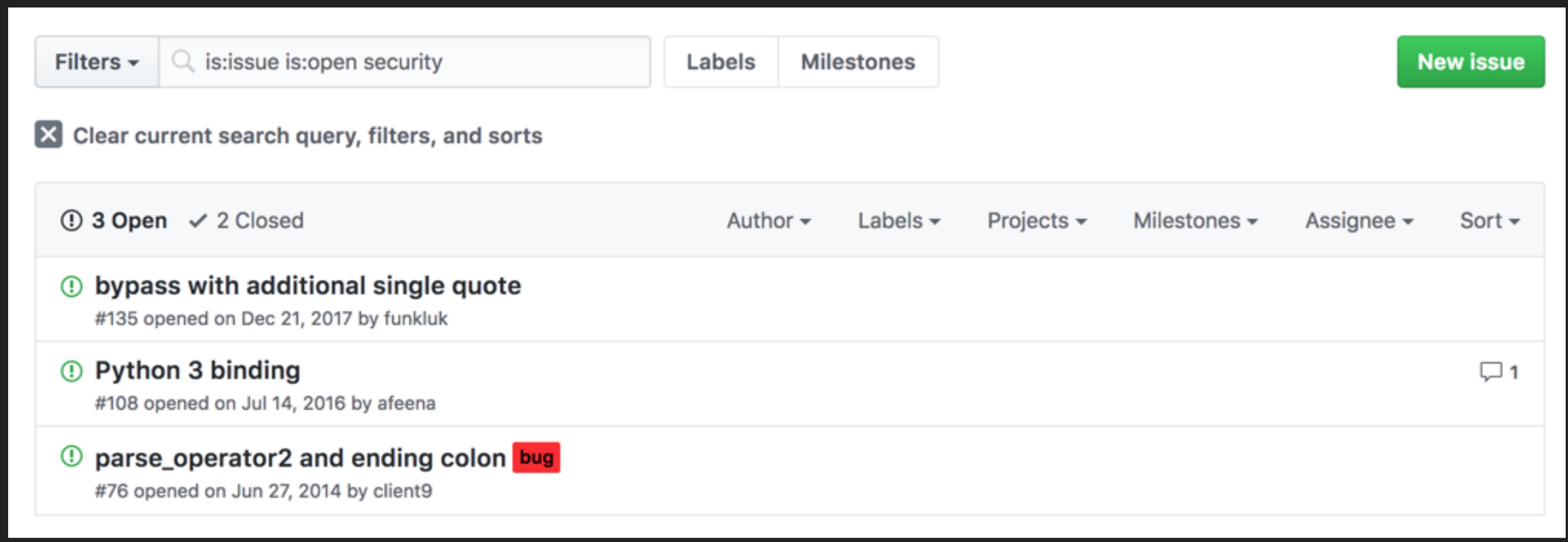
A typical process for handling a new security vulnerability is as follows. Projects that wish to use other processes MAY do so, but MUST clearly and publicly document their process and have security@ review it ahead of time.

Note: No information should be made public about the vulnerability until it is formally announced at the end of this process. That means, for example that a Jira issue must NOT be created to track the issue since that will make the issue public. Also the messages associated with any commits should not make ANY reference to the security nature of the commit.

1. The person discovering the issue, the reporter, reports the vulnerability privately to security@project.apache.org or to security@apache.org
2. Messages that do not relate to the reporting or managing of an undisclosed security vulnerability in Apache software are ignored and no further action is required.
3. If reported to security@apache.org, the security team will forward the report (without acknowledging it) to the project's security list or, of the project does not have a security list, to the project's private (PMC) mailing list.
4. The project team sends an e-mail to the original reporter to acknowledge the report.
5. The project team investigates report and either rejects it or accepts it.
6. If the report is rejected, the project team writes to the reporter to explain why.
7. If the report is accepted, the project team writes to report to let them know it is accepted and that they are working on a fix.
8. The project team requests a CVE number from security@apache.org by sending an e-mail with the subject "CVE request for ..." and providing a short (one line)

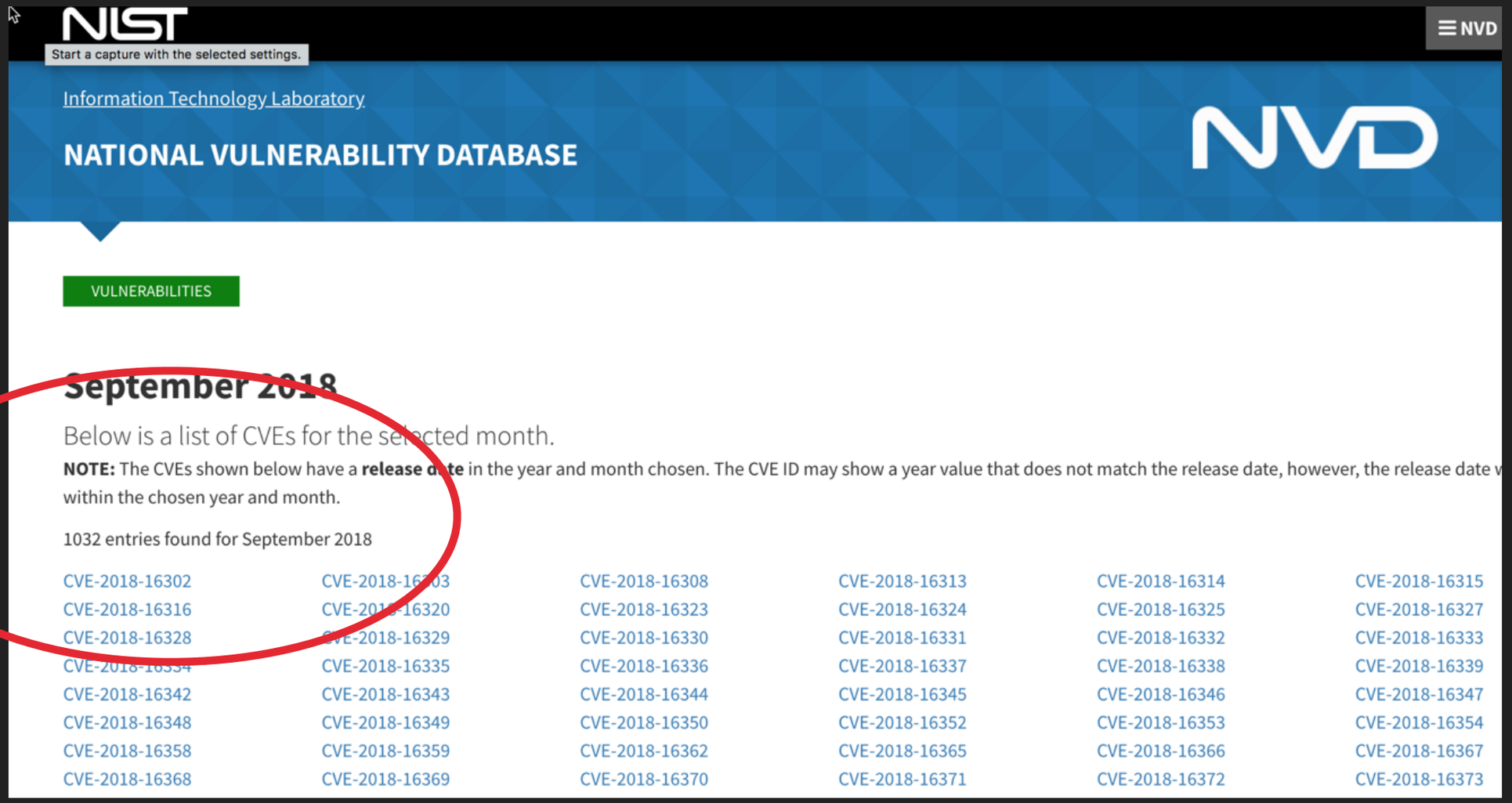
HTTPS://WWW.APACHE.ORG/SECURITY/COMMITTERS.HTML

WRITTEN POLICY



LOOK FOR UNFIXED SECURITY BUGS

No known vulnerabilities does not equal security



https://nvd.nist.gov/vuln/full-listing/2018/9

SEPTEMBER ALONE 1032 NEW VULNERABILITIES REPORTED SO FAR

- CVES (COMMON VULNERABILITIES AND EXPOSURES) AREN'T EASY TO FILE,
- THIS CAN BE A SIGN OF A LACK OF SECURITY EXPERTISE
- MANY TIMES NO ONE IS LOOKING
- SOMETIMES ISSUES ARE ACTIVELY REJECTED OR HIDDEN
- HOW PROJECT TEAMS REACT TO KNOWN VULNERABILITIES WILL HELP YOU TO EVALUATE THEIR SECURITY PROCESSES.

WHY?

Good security reporting
does not guarantee action

About
Screenshots
Downloads
Documentation
End-user docs
Technical docs
Contribute
Community

Ticket #15167 (closed task: wontfix)

Kernel Address Info Leak

Opened 4 months ago
Last modified 4 months ago

Reported by:	wcrobert	Owned by:	
Priority:	major	Component:	other
Version:	VirtualBox 5.0.14	Keywords:	info leak
Cc:		Guest type:	other
Host type:	Linux		

Description (last modified by frank) (diff)

I reported this via secalert_us@... and was told to resubmit here:

vbox kernel module seems to printk kernel addresses that get picked up by syslog. This information could be used by someone who has gained uid/gid syslog adm (On Ubuntu) to successfully chain an attack to kernel data structures (thus defeating ASLR). Information from /proc/modules is sanitized for non-root users.

The requested fix is to stop printing out kernel addresses.

Host $ lsb_release -a No LSB modules are available. Distributor ID: Ubuntu Description: Ubuntu 14.04.4 LTS Release: 14.04 Codename: trusty

$uname -a Linux wcrobert-MOBL1 3.19.0-18-generic ~~#18~~~14.04.1-Ubuntu SMP Wed May 20 09:38:33 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux

VBox Version: Version 5.0.14 r105127

What I found in syslog:

```
Feb 11 11:27:57 wcrobert-MOBL1 kernel: [    5.881847] vboxdrv: Found 4 processor cores
Feb 11 11:27:57 wcrobert-MOBL1 kernel: [    5.901307] vboxdrv: TSC mode is Invariant, t
Feb 11 11:27:57 wcrobert-MOBL1 kernel: [    5.901310] vboxdrv: Successfully loaded vers
Feb 11 11:27:57 wcrobert-MOBL1 kernel: [    6.112417] vboxpci: IOMMU not found (not req
Feb 11 12:16:23 wcrobert-MOBL1 kernel: [ 2913.482380] vboxdrv: ffffffffc0000020 VMMR0.r
Feb 11 12:16:23 wcrobert-MOBL1 kernel: [ 2913.571393] vboxdrv: ffffffffc00fa020 VBoxDDR
Feb 11 12:16:23 wcrobert-MOBL1 kernel: [ 2913.572892] vboxdrv: ffffffffc0119020 VBoxDD2
Feb 11 12:16:23 wcrobert-MOBL1 kernel: [ 2913.606759] vboxdrv: ffffffffc011de00 VBoxEhc
```

Change History

Changed 4 months ago by frank — comment:1

- **Description** modified (diff)

Changed 4 months ago by frank — comment:2

- **Status** changed from *new* to *closed*
- **Resolution** set to *wontfix*

Sorry, this is NOT a security problem and not a problem at all. Having these addresses is not a problem because special permissions are required to make use of them.

Note: See TracTickets for help on using tickets.

Changed 4 months ago by frank comment:2

- **Status** changed from *new* to *closed*
- **Resolution** set to *wontfix*

Sorry, this is NOT a security problem and not a problem at all. Having these addresses is not a problem because special permissions are required to make use of them.

Note: See TracTickets for help on using tickets.

This turned out to be untrue, and a CVE was actually issued

STEP 4:
IS THERE ANY TESTING

```
Test Plan:

Run tests to make sure we're not

crazy, and cross fingers.
```

- From https://github.com/Khan/react-components/commit/539b0568b983d534b5c186b588a47ed462da75db

---

IS THERE A TEST SUITE

TESTING ISN'T THE ONLY THING WE TAKE INTO ACCOUNT, BUT IT CAN BE USED AS A RULE OF THUMB IF YOU DON'T KNOW SECURITY AND WANT TO GUESS AT WHAT MIGHT BE A GOOD LIBRARY.

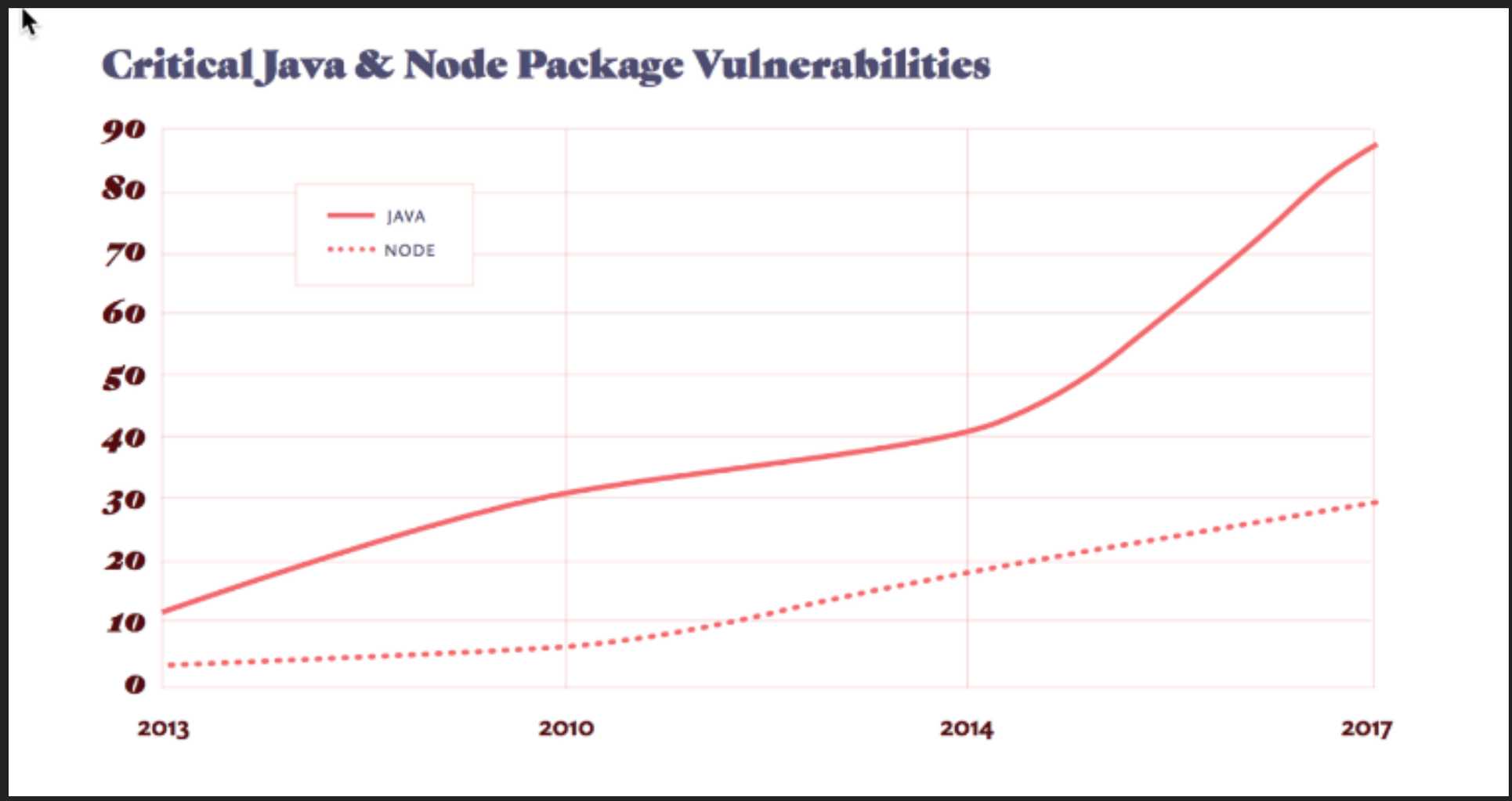
- DOES THIS TEST SUITE COVER BAD BEHAVIOUR?
- HOW COMPREHENSIVE IS THIS TEST SUITE?
- DO ALL TESTS PASS?
- IS THERE CONTINUOUS INTEGRATION FOR TESTS?

TESTING IS ESPECIALLY IMPORTANT FOR LIBRARIES THAT HANDLE USER INPUT: PARSERS, INPUT VALIDATION LIBRARIES, ETC. A POOR TEST SUITE DOESN'T GUARANTEE A BLACKLISTED COMPONENT, BUT A GOOD SUITE OFTEN IMPLIES A BETTER CHOICE.

KEY QUESTIONS ABOUT TEST SUITES

STEP 5:
BE AWARE

Popularity does not equal security



“90% of tested organizations use vulnerable dependencies."

- https://snyk.io/

Good packages do not guarantee good dependencies

- UPSTREAM PROJECTS DO NOT USE THE SAME CRITERIA FOR INCLUSION AS PRODUCTS SHOULD
- IT IS POSSIBLE FOR A PACKAGE TO DEPEND ON CODE THAT HAS VULNERABILITIES

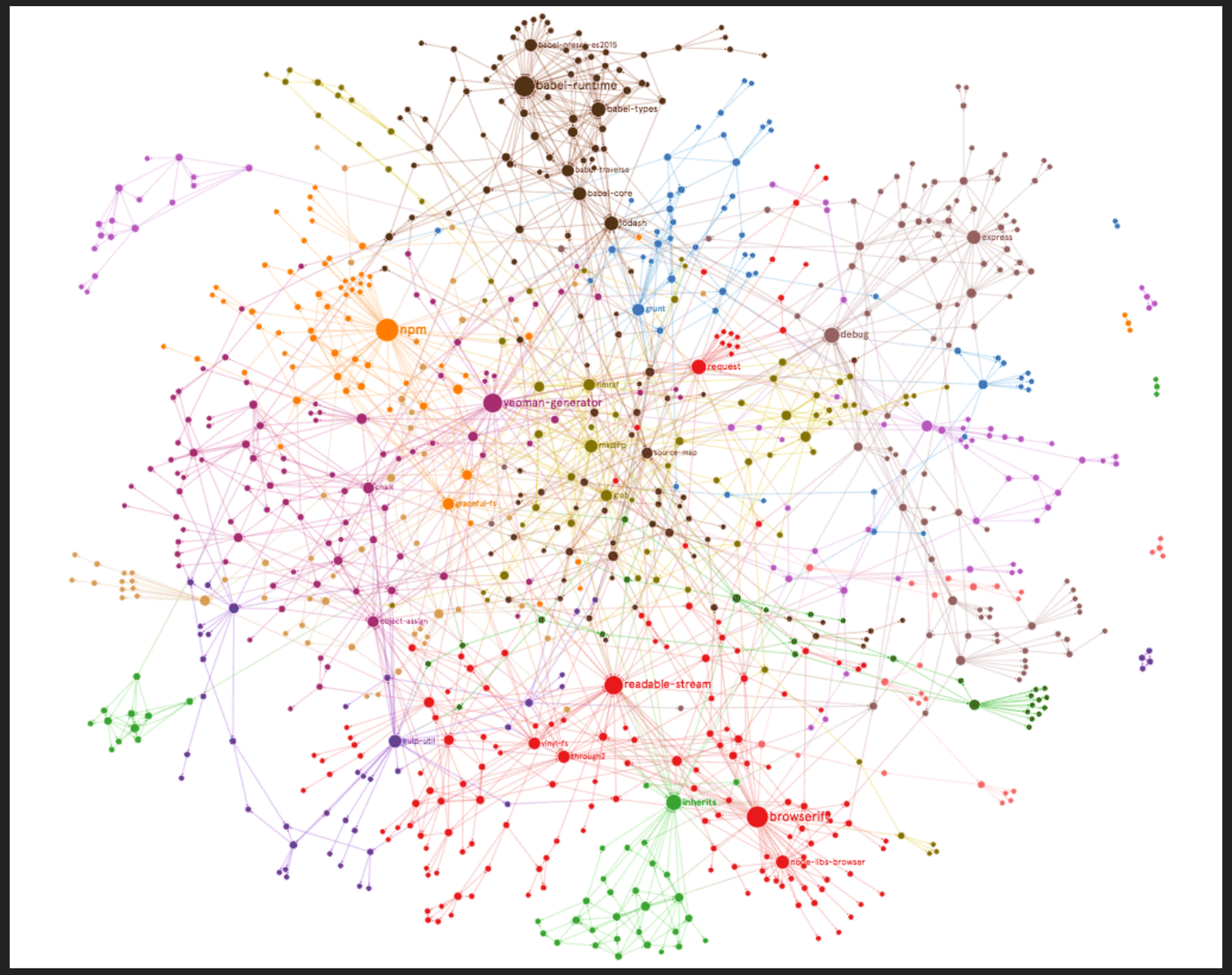
UNDERSTAND YOUR DEPENDENCIES AND THE ADDITIONAL RISK IT BRINGS. ALLOCATE TIME TO WORK ON THE ISSUES

WHY?

NPM (NODE.JS), PYPI (PYTHON), RUBY GEMS (RUBY), CPAN (PERL), ETC. ARE NOT CURATED

ALL REPOSITORIES ARE NOT CREATED EQUAL

I wonder what happens when you remove a module?



Oh wait!! that already happened

Package managers don’t always imply high quality

A programmer named Azer Koçulu, during a trademark dispute, refused to change the name of his software module. Lawyers were able to get npm to give them rights to the module. In protest Koçulu removed his other code from npm, including something called npm left-pad which was used by thousands of projects.

“Popular software projects like Babel, which helps Facebook, Netflix, and Spotify run code faster and React, which helps developers build better interfaces, were suddenly broken and no more work could be done with them.”

Npm re-published the code, giving it to a new owner

“HOW REMOVING 11 LINES OF CODE NEARLY BROKE THE INTERNET”

BRINGING IT ALL TOGETHER

TAKE A LOOK (ARE THERE RED FLAGS?) REALLY READ IT
CHECK FOR THE NUMBER OF CONTRIBUTORS & ACTIVITY
DOES THE PROJECT HANDLE SECURITY ISSUES
LOOK FOR A TEST SUITE (MAKE SURE IT DOES SOMETHING)
BE AWARE TRUST NOTHING

KEY TAKEAWAYS

Thank you

LinkedIn: **https://www.linkedin.com/in/mikide/**

Twitter: **@theDawgCr8**

Email: sec-princess@unroutable.me

Intel Credits: Terri Oda, Tiberius Heflin, Bill Roberts, John Anderson

https://github.com/sec-princess/WWCode-OSS-Study-Night-20180927

SECTION	GRADE	GRADING GUIDELINES
First look		A - Mentions security audit or other proactive security activity. B - No major warning signs, and code is used professionally. C - No major warning signs, but not widely used or not well-supported. D - Code has minor warning signs that need to be investigated in more detail. F - Code has known issues, major warning signs, or is abandoned
Contributors and activity		A - At least five significant, active contributors. B - More than two significant, active contributors. C - Only one major contributor who is active. D - Project has been inactive for nine months or less. F - Project has been inactive for more than one year.
Security issues		A - Project has had previous security issues and handled them quickly and well. Bonus if they also mention doing proactive security such as fuzz testing, static analysis, or security audits. B - Project has a plan for handling security issues but hasn't had to use it much yet. C - Project does not have a plan for security issues but at least has an active bug tracker and issues get resolved. D - Project does not seem to resolve many open bugs. F - Project has open security issues that are not in the process of being resolved.
Test suite		A - Project has test suite with good coverage of positive and negative test cases set up as part of continuous integration, and test results are published for each build. B - Project has test suite with good coverage but no continuous integration. C - Test suite mostly covers positive test cases; very few or no error cases. D - Test suite has very low coverage or is only a few examples. F - No test suite.