Open Source Approaches to Security for Applications and Services at Mozilla

Adam Muntner
amuntner@mozilla.com

Who is Adam?

- Security Engineer, Mozilla Enterprise Security team
- 4+ years Mozilla employee
  - Re-architecting Mozilla's Appsec program
  - Program owner, Mozilla Web Bug Bounty
- Working in Infosec since 1998: Pentester, Consulting Manager, Principal Consultant, CSO, Security Engineer
- FuzzDB (https://github.com/fuzzdb-project/fuzzdb)
- Based in NYC
- Reformed, former CISSP

Agenda

- Lessons learned from radical open sharing of design documentation
- Approaches to qualitative comparison of risk for an inventory of websites and services
- Using OpenSAMM in a DevOps organization
- Why your bug bounty program is one of the best sources of intelligence for driving the future direction of your application security program
- Maximizing the value gained from identified vulnerabilities
- Get non-security engineers help pentest by setting up a Red Team

Lessons learned from radical open sharing of design documentation

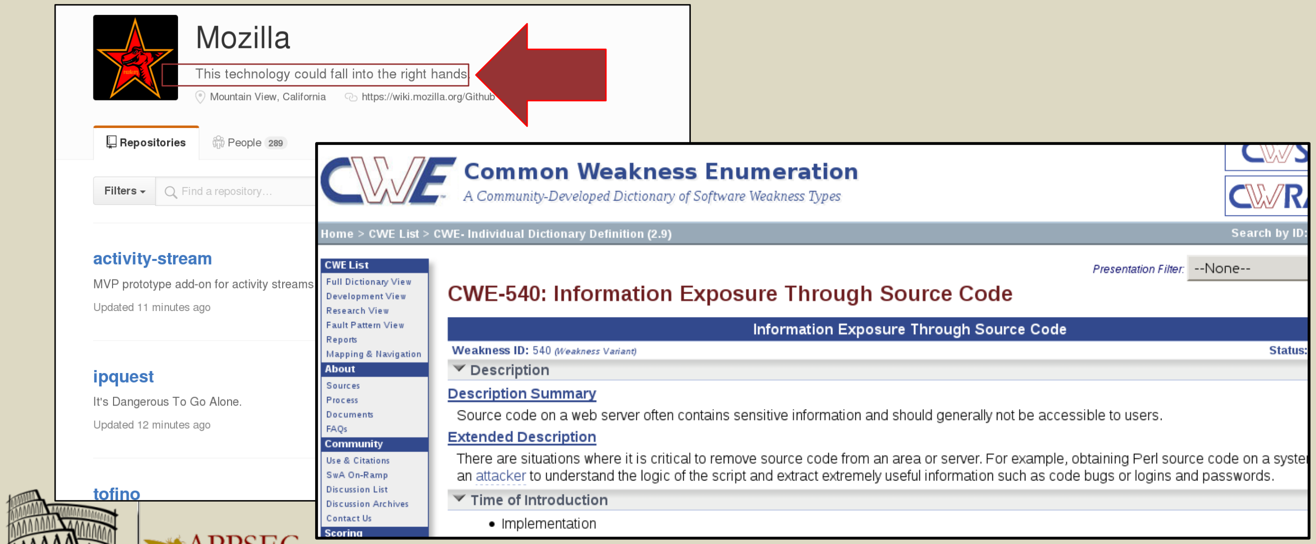
ROMA MMXVI | APPSEC EUROPE

Mozilla’s Appsec Threat Model

We make a web popular Open Source web browser

- Protecting users: our browser’s support is via the web
- Our security model expects our web services to be trustworthy
  - Installation, updates, crash reporting APIs, FxA, Addons, Hello, Sync, etc.
  - Each is attractive to spammers, criminals, state actors for different reasons
  - Our security model expects our web services to be trustworthy

Mozilla’s Appsec Threat Model



Mozilla

This technology could fall into the right hands

Mountain View, California https://wiki.mozilla.org/Github

activity-stream

MVP prototype add-on for activity streams

Updated 11 minutes ago

ipquest

It's Dangerous To Go Alone.

Updated 12 minutes ago

tofino

Common Weakness Enumeration

A Community-Developed Dictionary of Software Weakness Types

Presentation Filter: --None--

CWE-540: Information Exposure Through Source Code

Information Exposure Through Source Code

Weakness ID: 540 *(Weakness Variant)* **Status:**

▼ Description

Description Summary

Source code on a web server often contains sensitive information and should generally not be accessible to users.

Extended Description

There are situations where it is critical to remove source code from an area or server. For example, obtaining Perl source code on a syste an attacker to understand the logic of the script and extract extremely useful information such as code bugs or logins and passwords.

▼ Time of Introduction

- Implementation

https://cwe.mitre.org/

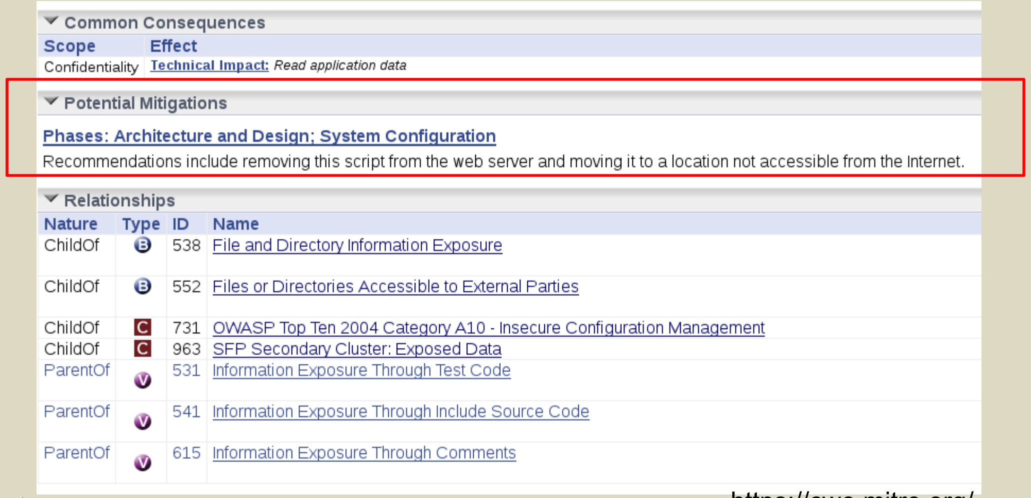
There’s a long debate over whether whether open source software is more security by virtue of its model.
The correct answer is, it depends on from which actor’s point of view and their subjective values and goals and available alternatives.
There are too many variables that would apply in any particular to generalize.

MITRE CWE - “Common Weakness Enumeration” is a categorical taxonomy of software weaknesses.
CWE-540 defines source code exposure as a security weakness.

That doesn’t mean “open source software is insecure,”
it means that the attacker can examine the code for flaws such as injection attacks or other execution paths that were never intended by the authors.

Mozilla’s Appsec Threat Model

*



Common Consequences

Scope	Effect
Confidentiality	Technical Impact: *Read application data*

Potential Mitigations

Phases: Architecture and Design; System Configuration

Recommendations include removing this script from the web server and moving it to a location not accessible from the Internet.

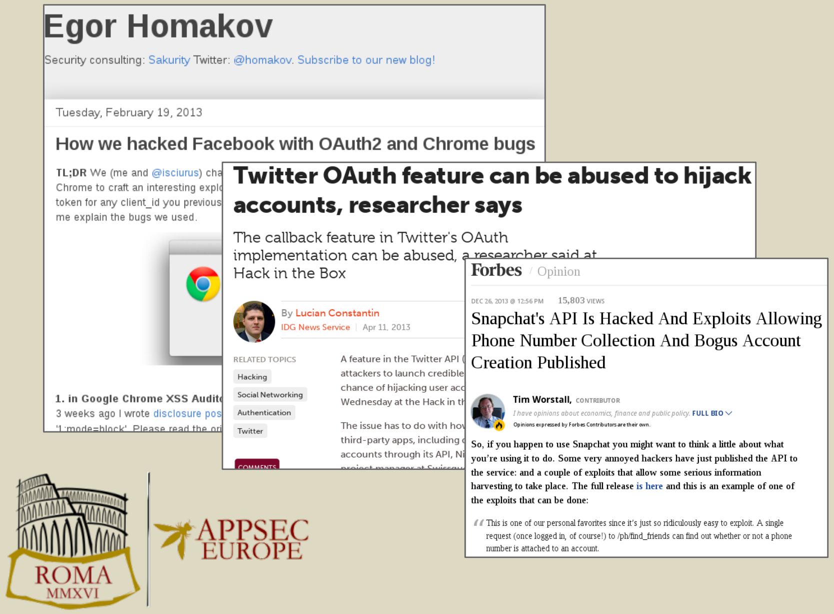
Relationships

Nature	Type	ID	Name
ChildOf	B	538	File and Directory Information Exposure
ChildOf	B	552	Files or Directories Accessible to External Parties
ChildOf	C	731	OWASP Top Ten 2004 Category A10 - Insecure Configuration Management
ChildOf	C	963	SFP Secondary Cluster: Exposed Data
ParentOf	V	531	Information Exposure Through Test Code
ParentOf	V	541	Information Exposure Through Include Source Code
ParentOf	V	615	Information Exposure Through Comments

https://cwe.mitre.org/

* *Not a potential mitigation for Mozilla.*

Mozilla's Appsec Threat Model



Egor Homakov

Security consulting: Sakurity Twitter: @homakov. Subscribe to our new blog!

Tuesday, February 19, 2013

How we hacked Facebook with OAuth2 and Chrome bugs

TL;DR We (me and @isciurus) cha
Chrome to craft an interesting explo
token for any client_id you previous
me explain the bugs we used.

1. in Google Chrome XSS Audit
3 weeks ago I wrote disclosure pos
'1;mode=block'. Please read the ori

Twitter OAuth feature can be abused to hijack accounts, researcher says

The callback feature in Twitter's OAuth implementation can be abused, a researcher said at Hack in the Box

By Lucian Constantin
IDG News Service | Apr 11, 2013

RELATED TOPICS
Hacking
Social Networking
Authentication
Twitter

COMMENTS

A feature in the Twitter API (
attackers to launch credible
chance of hijacking user acc
Wednesday at the Hack in th

The issue has to do with hov
third-party apps, including c
accounts through its API, Ni
project manager at Swissqu

Forbes / Opinion

DEC 26, 2013 @ 12:56 PM 15,803 VIEWS

Snapchat's API Is Hacked And Exploits Allowing Phone Number Collection And Bogus Account Creation Published

Tim Worstall, CONTRIBUTOR
I have opinions about economics, finance and public policy. FULL BIO
Opinions expressed by Forbes Contributors are their own.

So, if you happen to use Snapchat you might want to think a little about what you're using it to do. Some very annoyed hackers have just published the API to the service: and a couple of exploits that allow some serious information harvesting to take place. The full release is here and this is an example of one of the exploits that can be done:

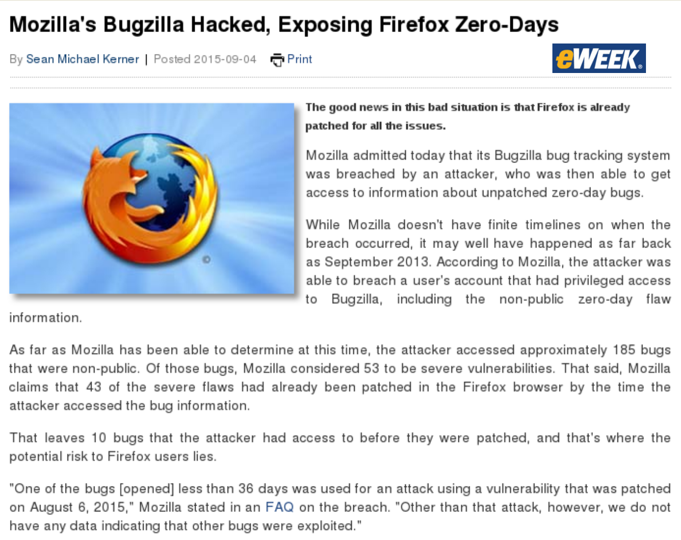
> This is one of our personal favorites since it's just so ridiculously easy to exploit. A single request (once logged in, of course!) to /ph/find_friends can find out whether or not a phone number is attached to an account.

Mozilla Back End Systems

Firefox Accounts
Sync
Addons
Plugincheck
Loop/Hello
Crashreports
Telemetry
Etc...
+ ~3000 Websites

It take lots of web services support a modern browser.

Mozilla’s Appsec Threat Model



Mozilla's Bugzilla Hacked, Exposing Firefox Zero-Days

By Sean Michael Kerner | Posted 2015-09-04 Print

The good news in this bad situation is that Firefox is already patched for all the issues.

Mozilla admitted today that its Bugzilla bug tracking system was breached by an attacker, who was then able to get access to information about unpatched zero-day bugs.

While Mozilla doesn't have finite timelines on when the breach occurred, it may well have happened as far back as September 2013. According to Mozilla, the attacker was able to breach a user's account that had privileged access to Bugzilla, including the non-public zero-day flaw information.

As far as Mozilla has been able to determine at this time, the attacker accessed approximately 185 bugs that were non-public. Of those bugs, Mozilla considered 53 to be severe vulnerabilities. That said, Mozilla claims that 43 of the severe flaws had already been patched in the Firefox browser by the time the attacker accessed the bug information.

That leaves 10 bugs that the attacker had access to before they were patched, and that's where the potential risk to Firefox users lies.

"One of the bugs [opened] less than 36 days was used for an attack using a vulnerability that was patched on August 6, 2015," Mozilla stated in an FAQ on the breach. "Other than that attack, however, we do not have any data indicating that other bugs were exploited."

Luckily, the attacker was not particularly ambitious.

ROMA MMXVI | APPSEC EUROPE

More details:
https://blog.mozilla.org/security/2015/08/06/firefox-exploit-found-in-the-wild/

Bugzilla is the site that worries me the most.

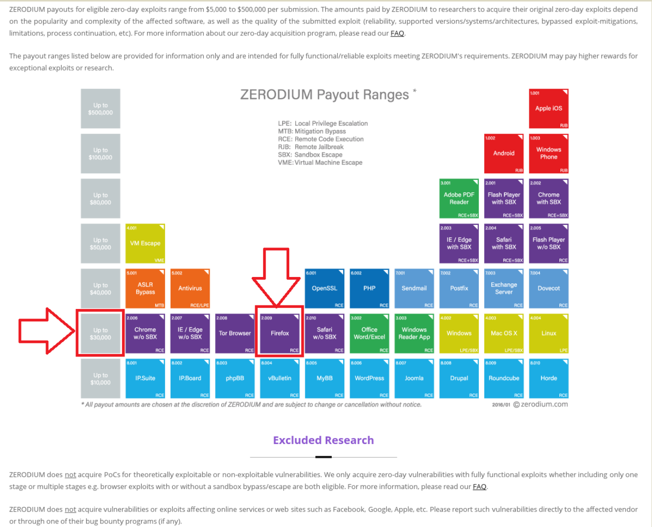
It’s where our open security bugs live.

Last year we discovered that an attack had gained access to security bugs through a legit account

A Firefox user informed us that an advertisement on a news site in Russia was serving a Firefox exploit in pdf.js that searched for sensitive files and uploaded them to a server that appears to be in Ukraine….

Mozilla’s Appsec Threat Model

How much is our most sensitive data worth?



ZERODIUM payouts for eligible zero-day exploits range from $5,000 to $500,000 per submission. The amounts paid by ZERODIUM to researchers to acquire their original zero-day exploits depend on the popularity and complexity of the affected software, as well as the quality of the submitted exploit (reliability, supported versions/systems/architectures, bypassed exploit-mitigations, limitations, process continuation, etc). For more information about our zero-day acquisition program, please read our FAQ.

The payout ranges listed below are provided for information only and are intended for fully functional/reliable exploits meeting ZERODIUM's requirements. ZERODIUM may pay higher rewards for exceptional exploits or research.

ZERODIUM Payout Ranges *

LPE: Local Privilege Escalation
MTB: Mitigation Bypass
RCE: Remote Code Execution
RJB: Remote Jailbreak
SBX: Sandbox Escape
VME: Virtual Machine Escape

Up to $500,000										1.001 Apple iOS RJB
Up to $100,000									1.002 Android RJB	1.003 Windows Phone RJB
Up to $80,000								3.001 Adobe PDF Reader RCE+SBX	2.001 Flash Player with SBX RCE+SBX	2.002 Chrome with SBX RCE+SBX
Up to $50,000	4.001 VM Escape VME							2.003 IE / Edge with SBX RCE+SBX	2.004 Safari with SBX RCE+SBX	2.005 Flash Player w/o SBX RCE
Up to $40,000	5.001 ASLR Bypass MTB	5.002 Antivirus RCE/LPE			6.001 OpenSSL RCE	6.002 PHP RCE	7.001 Sendmail RCE	7.002 Postfix RCE	7.003 Exchange Server RCE	7.004 Dovecot RCE
Up to $30,000	2.006 Chrome w/o SBX RCE	2.007 IE / Edge w/o SBX RCE	2.008 Tor Browser RCE	2.009 Firefox RCE	2.010 Safari w/o SBX RCE	3.002 Office Word/Excel RCE	3.003 Windows Reader App RCE	4.002 Windows LPE/SBX	4.003 Mac OS X LPE/SBX	4.004 Linux LPE
Up to $10,000	8.001 IP.Suite RCE	8.002 IP.Board RCE	8.003 phpBB RCE	8.004 vBulletin RCE	8.005 MyBB RCE	8.006 WordPress RCE	8.007 Joomla RCE	8.008 Drupal RCE	8.009 Roundcube RCE	8.010 Horde RCE

* All payout amounts are chosen at the discretion of ZERODIUM and are subject to change or cancellation without notice.

2016/01 © zerodium.com

Excluded Research

ZERODIUM does not acquire PoCs for theoretically exploitable or non-exploitable vulnerabilities. We only acquire zero-day vulnerabilities with fully functional exploits whether including only one stage or multiple stages e.g. browser exploits with or without a sandbox bypass/escape are both eligible. For more information, please read our FAQ.

ZERODIUM does not acquire vulnerabilities or exploits affecting online services or web sites such as Facebook, Google, Apple, etc. Please report such vulnerabilities directly to the affected vendor or through one of their bug bounty programs (if any).

There are companies that pay for security bugs in our products.

Goal of Zerodium and their competitors:

1. 0-day stays 0-day
2. Sell weaponized exploits
3. ???
4. Profit!

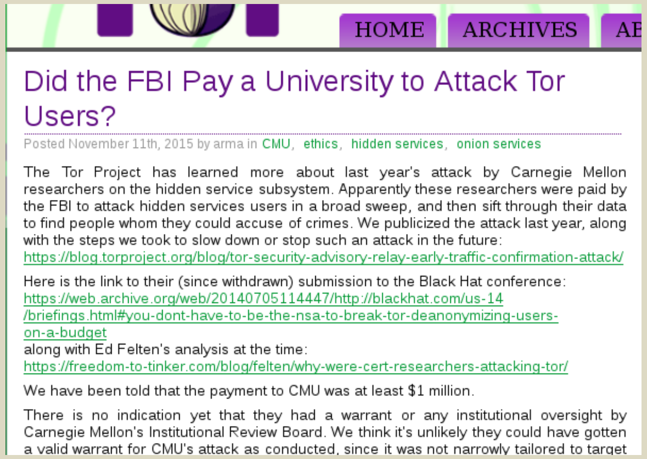
There are companies that aren’t us who pay for security bugs for some of our products, sometimes more than we do.

One of my biggest concerns for protecting our users is protecting Bugzilla.

Where do the bugs go?

Economics of the 0-Day

How much is our most sensitive data worth?



Did the FBI Pay a University to Attack Tor Users?

Posted November 11th, 2015 by arma in CMU, ethics, hidden services, onion services

The Tor Project has learned more about last year's attack by Carnegie Mellon researchers on the hidden service subsystem. Apparently these researchers were paid by the FBI to attack hidden services users in a broad sweep, and then sift through their data to find people whom they could accuse of crimes. We publicized the attack last year, along with the steps we took to slow down or stop such an attack in the future:
https://blog.torproject.org/blog/tor-security-advisory-relay-early-traffic-confirmation-attack/

Here is the link to their (since withdrawn) submission to the Black Hat conference:
https://web.archive.org/web/20140705114447/http://blackhat.com/us-14/briefings.html#you-dont-have-to-be-the-nsa-to-break-tor-deanonymizing-users-on-a-budget
along with Ed Felten's analysis at the time:
https://freedom-to-tinker.com/blog/felten/why-were-cert-researchers-attacking-tor/

We have been told that the payment to CMU was at least $1 million.

There is no indication yet that they had a warrant or any institutional oversight by Carnegie Mellon's Institutional Review Board. We think it's unlikely they could have gotten a valid warrant for CMU's attack as conducted, since it was not narrowly tailored to target

How much is it worth, to whom?

Compared to what?

Cost of next available substitute?

Polaris Privacy Initiative
https://wiki.mozilla.org/Polaris
https://blog.mozilla.org/netpolicy/2014/11/10/introducing-polaris-privacy-initiative-to-accelerate-user-focused-privacy-online/

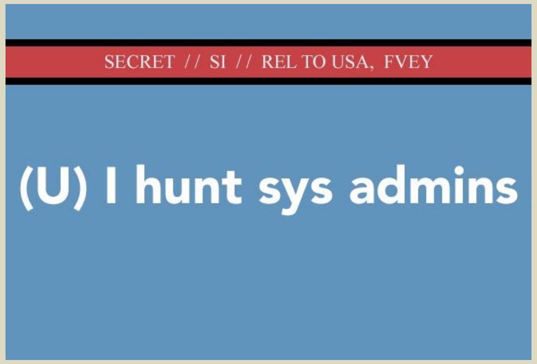
The economics of 0-days….

$1 million was a bargain for the fbi, compared to the cost of next available substitute:

If the FBI was willing to pay $1 million for the exploit, it’s only because the next available option to the FBI would have cost more than a million dollars, probably significantly more.

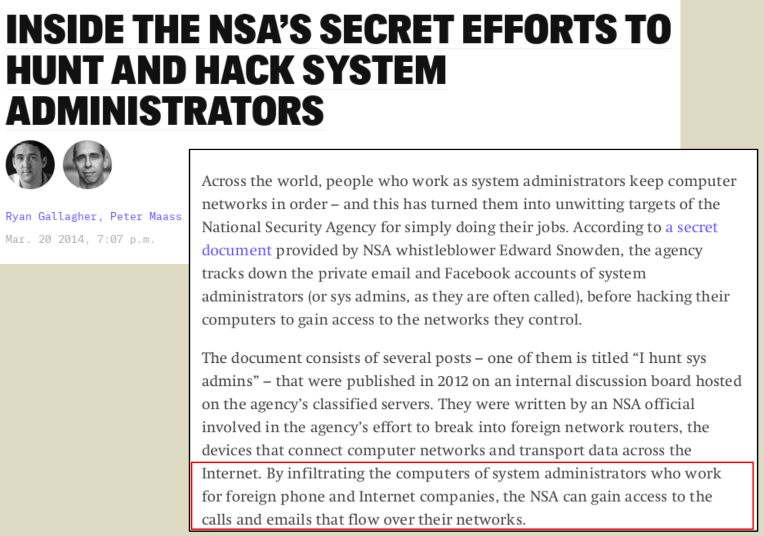
State actors have virtually unlimited budgets, they don’t face the usual resource constraints, if they need more money, they print it.

Mozilla's Appsec Threat Model



SECRET // SI // REL TO USA, FVEY

(U) I hunt sys admins



INSIDE THE NSA'S SECRET EFFORTS TO HUNT AND HACK SYSTEM ADMINISTRATORS

Ryan Gallagher, Peter Maass

Mar. 20 2014, 7:07 p.m.

Across the world, people who work as system administrators keep computer networks in order – and this has turned them into unwitting targets of the National Security Agency for simply doing their jobs. According to a secret document provided by NSA whistleblower Edward Snowden, the agency tracks down the private email and Facebook accounts of system administrators (or sys admins, as they are often called), before hacking their computers to gain access to the networks they control.

The document consists of several posts – one of them is titled "I hunt sys admins" – that were published in 2012 on an internal discussion board hosted on the agency's classified servers. They were written by an NSA official involved in the agency's effort to break into foreign network routers, the devices that connect computer networks and transport data across the Internet. By infiltrating the computers of system administrators who work for foreign phone and Internet companies, the NSA can gain access to the calls and emails that flow over their networks.

https://theintercept.com/2014/03/20/inside-nsa-secret-efforts-hunt-hack-system-administrators

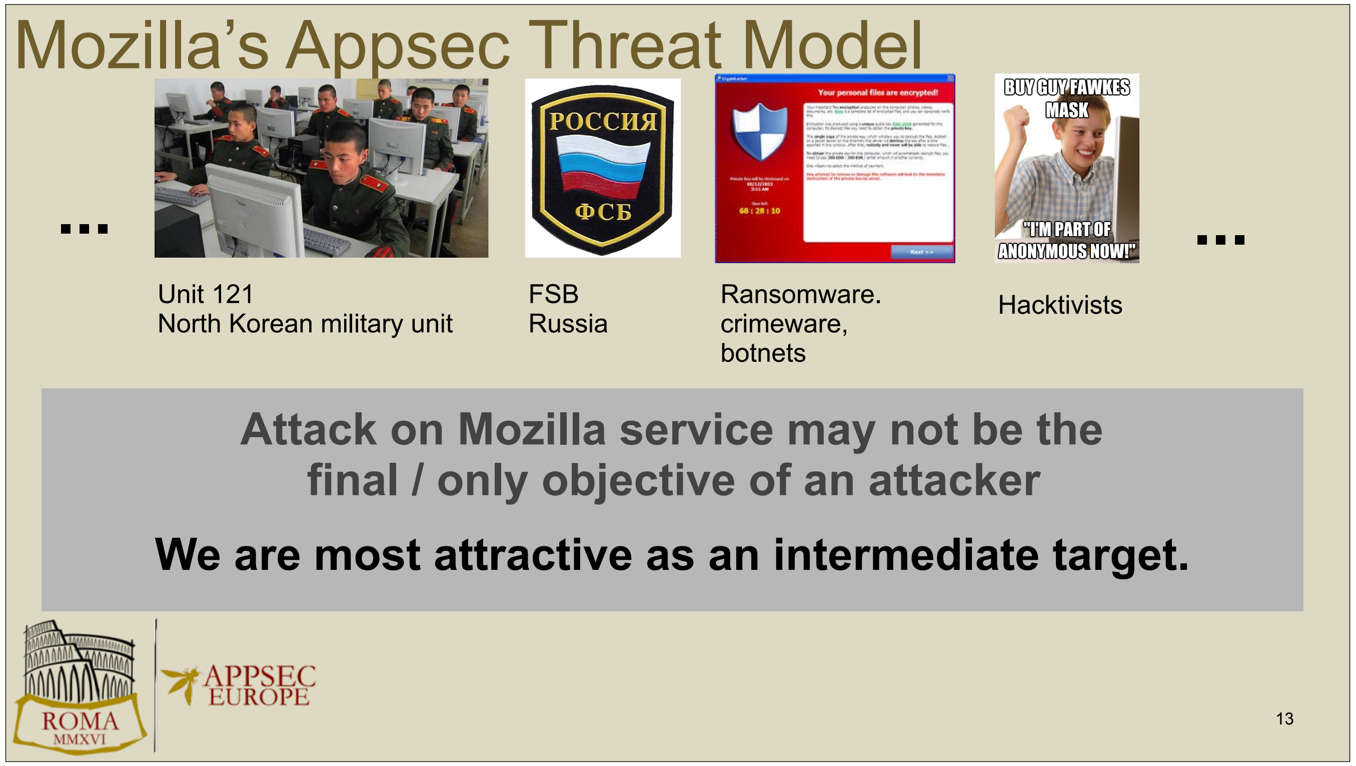
ROMA MMXVI

APPSEC EUROPE

The Intercept - Glenn Greenwald's investigative journalism website

The screenshot on the left is from a leaked NSA slide deck.

Take a look at the part I highlighted in red, on the right.


РОССИЯ
ФСБ
Your personal files are encrypted!
BUY GUY FAWKES MASK
"I'M PART OF ANONYMOUS NOW!"

Mozilla's Appsec Threat Model

...

Unit 121
North Korean military unit

FSB
Russia

Ransomware.
crimeware,
botnets

Hacktivists

...

Attack on Mozilla service may not be the final / only objective of an attacker

We are most attractive as an intermediate target.

Not picking on North Korea or Russia for any reason in particular, rather using them as exemplars that nearly every nation either has or is developing a cyberwar capability.

Because of Mozilla's addons, usage in the TOR browser and ironically popularity among people interested in privacy, our products make an attractive target to many organizations as a component of some larger goal.

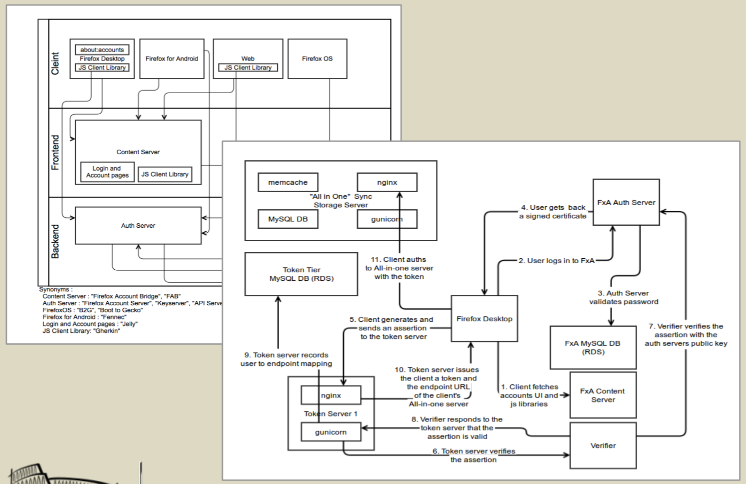
This is only one example of many actors with this kind of capability - not just state employed, sponsored, or tolerated,

I don't mean to scare you, but every single employee at Mozilla with any kind of access to internal resources is potentially a target, and not just by the NSA

There is another threat headline risk: that a news story story about a security issue will broadly and negatively affect our ability to pursue Mozilla's mission.

We share (almost) everything

Example: Firefox Accounts (FxA)


Cleint
about:accounts
Firefox Desktop
JS Client Library
Firefox for Android
Web
JS Client Library
Firefox OS
Frontend
Content Server
Login and Account pages
JS Client Library
Backend
Auth Server
memcache
nginx
"All in One" Sync Storage Server
MySQL DB
gunicorn
Token Tier MySQL DB (RDS)
11. Client auths to All-in-one server with the token
4. User gets back a signed certificate
FxA Auth Server
2. User logs in to FxA
3. Auth Server validates password
5. Client generates and sends an assertion to the token server
Firefox Desktop
FxA MySQL DB (RDS)
7. Verifier verifies the assertion with the auth servers public key
9. Token server records user to endpoint mapping
10. Token server issues the client a token and the endpoint URL of the client's All-in-one server
1. Client fetches accounts UI and js libraries
FxA Content Server
nginx
Token Server 1
gunicorn
8. Verifier responds to the token server that the assertion is valid
Verifier
6. Token server verifies the assertion

Synonyms :
Content Server : "Firefox Account Bridge", "FAB"
Auth Server : "Firefox Account Server", "Keyserver", "API Serve
FirefoxOS : "B2G", "Boot to Gecko"
Firefox for Android : "Fennec"
Login and Account pages : "Jelly"
JS Client Library: "Gherkin"

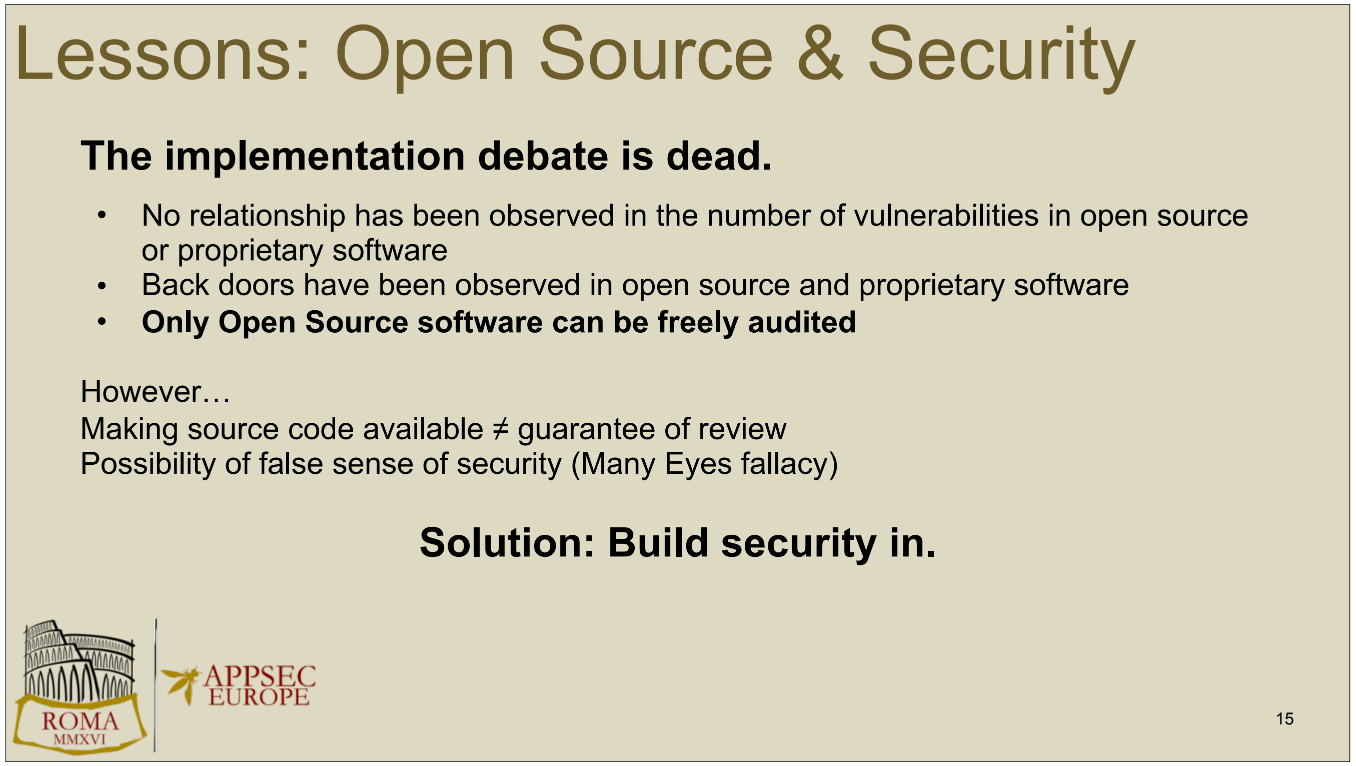
https://wiki.mozilla.org/Identity/Firefox_Accounts#Architecture

Public

- Source code
- Design docs
- Threat models
- Security tools
- Product mailing lists
- IRC
- Closed security bugs (Bugzilla)
- Incidents, post-incident

Not public

- Details of internal network
- MozDef configs
- Runbooks
- Open security bugs
- Live incidents



Lessons: Open Source & Security

The implementation debate is dead.

- No relationship has been observed in the number of vulnerabilities in open source or proprietary software
- Back doors have been observed in open source and proprietary software
- **Only Open Source software can be freely audited**

However…
Making source code available ≠ guarantee of review
Possibility of false sense of security (Many Eyes fallacy)

Solution: Build security in.

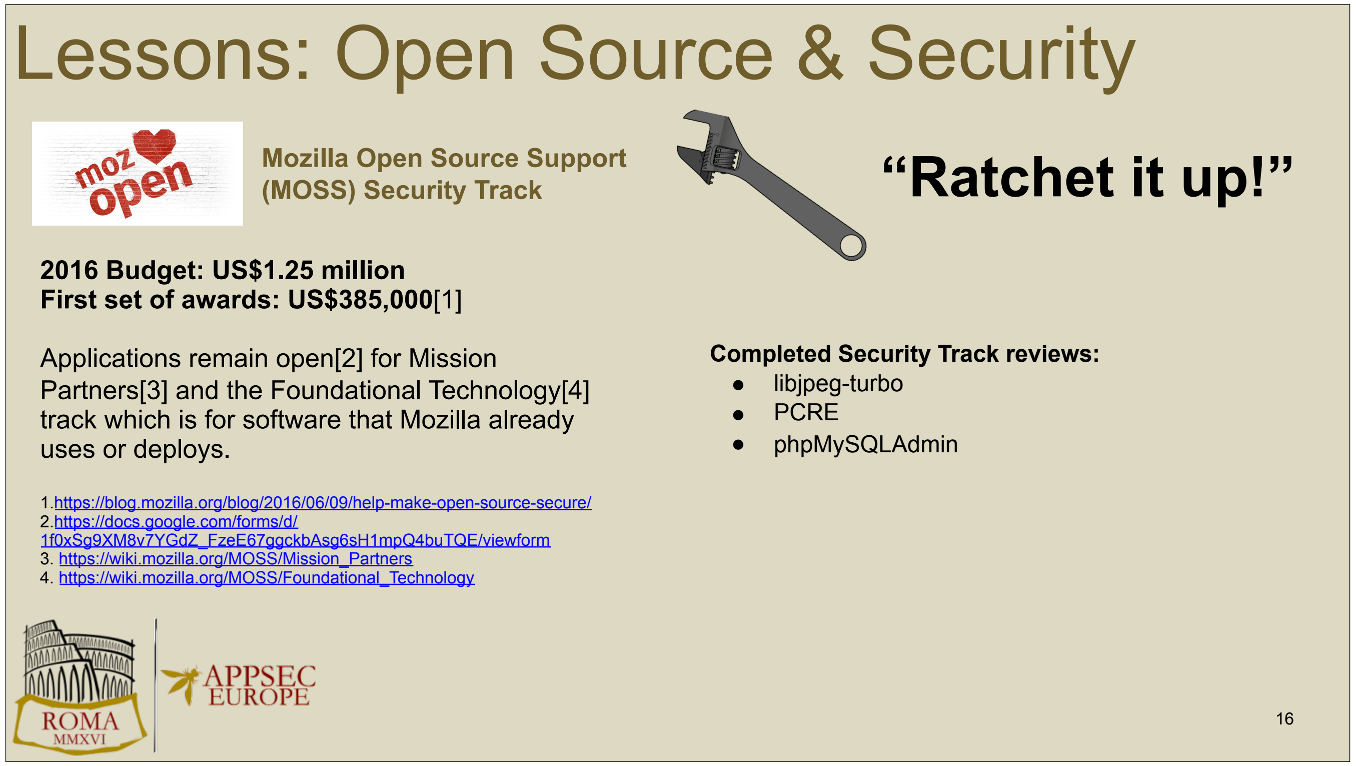
ROMA MMXVI | APPSEC EUROPE

Implementation debate: is oss or closed-source software better for security?
No relationship has been observed in the number of vulnerabilities in open source or proprietary software.
Only Open Source software can be freely audited

There are several dedicated Mozilla security teams,

- Triage browser bugs
- Fuzzing team
- Content security (web standards like CSP)
- Cloud Services (where I worked for three years, supports the back end of services used by the browser like Sync, Addons, etc)
  Enterprise Information Security, which I moved to in late 2015 to work on re-evaluating our overall appsec program and run the web bug bounty program

in addition to others that have security responsibilities of various kinds, not to mention a number of very dedicated community members



Lessons: Open Source & Security

Mozilla Open Source Support (MOSS) Security Track

"Ratchet it up!"

2016 Budget: US$1.25 million
First set of awards: US$385,000[1]

Applications remain open[2] for Mission Partners[3] and the Foundational Technology[4] track which is for software that Mozilla already uses or deploys.

Completed Security Track reviews:

- libjpeg-turbo
- PCRE
- phpMySQLAdmin

1.https://blog.mozilla.org/blog/2016/06/09/help-make-open-source-secure/
2.https://docs.google.com/forms/d/1f0xSg9XM8v7YGdZ_FzeE67ggckbAsg6sH1mpQ4buTQE/viewform
3. https://wiki.mozilla.org/MOSS/Mission_Partners
4. https://wiki.mozilla.org/MOSS/Foundational_Technology

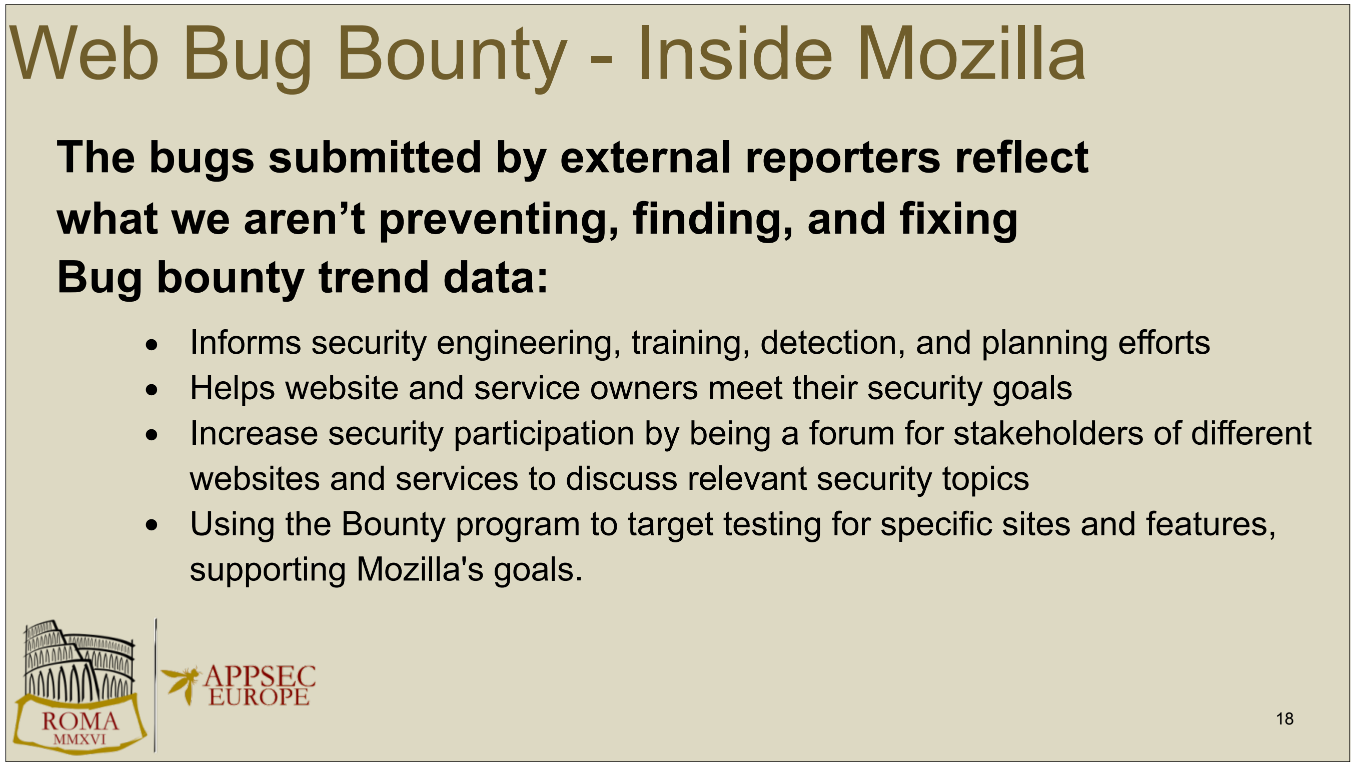
Security is a something you do, it's not a state.
Ratchet, as a verb, means to increase or tighten something in a series of small steps.
I got the phrase "ratchet it up" from my friend Perry Metzger, he runs the Cryptography mailing list which is the successor to the old Cypherpunks list.
To ratchet up security for software Mozilla depends on, we have funded a program to test the software and libraries we use. Mozilla Open Source Support (MOSS) is an awards program specifically focused on supporting the Open Source and Free Software movement, with a yearly budget of around $3 million.

With the security track, Mozilla will

- contract with and pay professional security firms to audit other projects' code
- work with project maintainers to support and implement fixes, and to manage disclosure and
- pay for the remediation work to be verified, to ensure any identified bugs have been fixed.

The other tracks just awarded grants to security and privacy related projects such as $152,500 to Tor for work on metrics to help make the network more stable, $77,000 to Tails,a secure-by-default live operating system that aims at preserving the user's privacy and anonymity, the money is for a method to verify that a Tails image was built from known-good sources
PeARS: $15,500. PeARS (Peer-to-peer Agent for Reciprocated Search) is a lightweight, distributed web search engine which runs in an individual's browser and indexes the pages they visit in a privacy-respecting way.
and others.
Which brings us to discussion about the web bug bounty program

Why your bug bounty program is one of the best sources of intelligence for driving the future direction of your Appsec program



Web Bug Bounty - Inside Mozilla

The bugs submitted by external reporters reflect what we aren’t preventing, finding, and fixing

Bug bounty trend data:

- Informs security engineering, training, detection, and planning efforts
- Helps website and service owners meet their security goals
- Increase security participation by being a forum for stakeholders of different websites and services to discuss relevant security topics
- Using the Bounty program to target testing for specific sites and features, supporting Mozilla's goals.

Bounty programs aren’t a substitute for good development practices, code review, pentesting

External reporters do help understand what the rest of your security program is missing

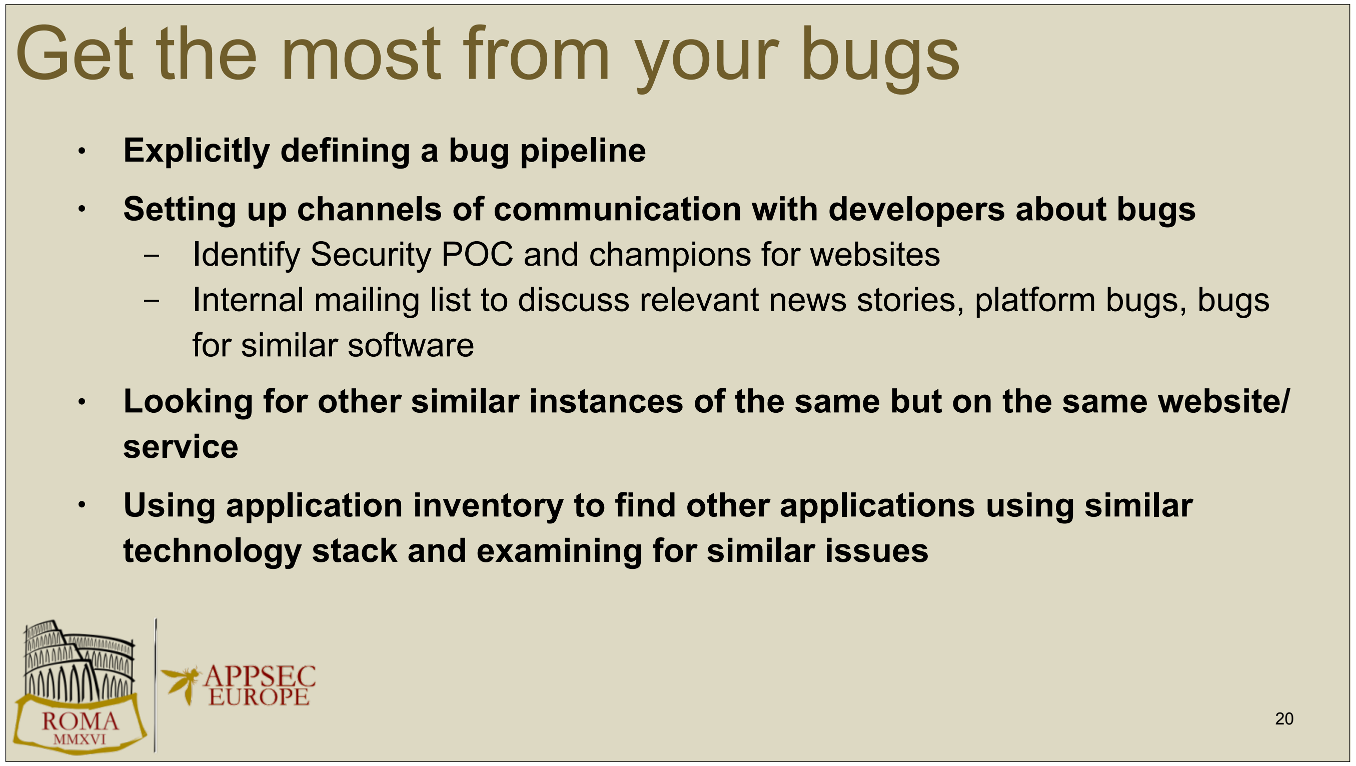
The program should be designed to encourage reporting for the kinds of bugs you’d like to hear about. You’re competing not just with other bounty programs, but with all other available options that bounty hunters and potential bounty hunters have for their time.

I mentioned an internal security mailing list - bounty bugs are excellent foil for discussion

Maximizing the value gained from identified vulnerabilities

ROMA
MMXVI

APPSEC
EUROPE



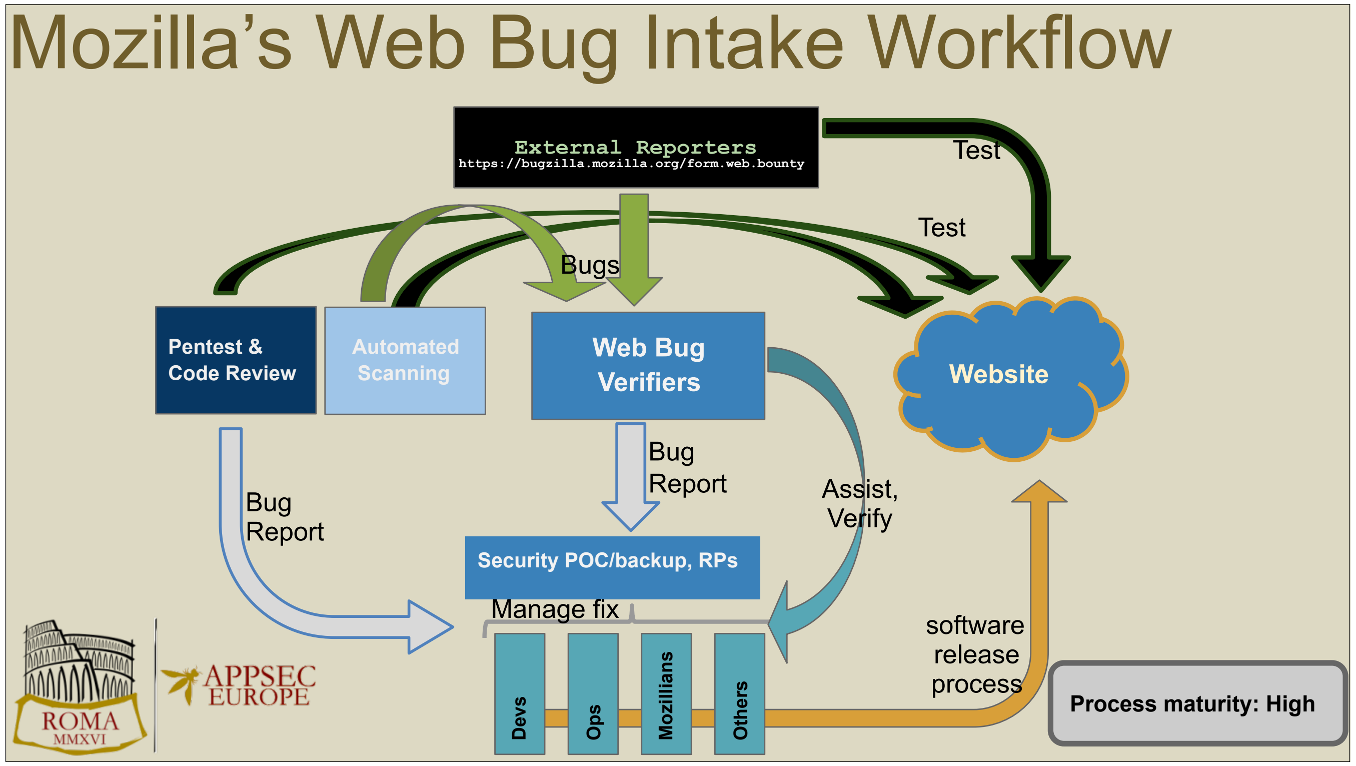
Get the most from your bugs

- **Explicitly defining a bug pipeline**
- **Setting up channels of communication with developers about bugs**
  - Identify Security POC and champions for websites
  - Internal mailing list to discuss relevant news stories, platform bugs, bugs for similar software
- **Looking for other similar instances of the same but on the same website/ service**
- **Using application inventory to find other applications using similar technology stack and examining for similar issues**

Explicitly define bug pipeline - if someone doesn’t own it, it probably isn’t going to happen consistently.

Internal comms channels - discussing bugs in similar websites and products and frameworks to those you use, not just the bugs that affect your software. Talking about security is fun, builds an internal security community, and the result will be fresh ideas & insights - learn from others problems, not just your own.

Next we’ll take a look at what Mozilla’s web bug intake workflow looks like.



Mozilla's Web Bug Intake Workflow

External Reporters
https://bugzilla.mozilla.org/form.web.bounty

Test

Test

Bugs

Pentest & Code Review

Automated Scanning

Web Bug Verifiers

Website

Bug Report

Bug Report

Assist, Verify

Security POC/backup, RPs

Manage fix

Devs

Ops

Mozillians

Others

software release process

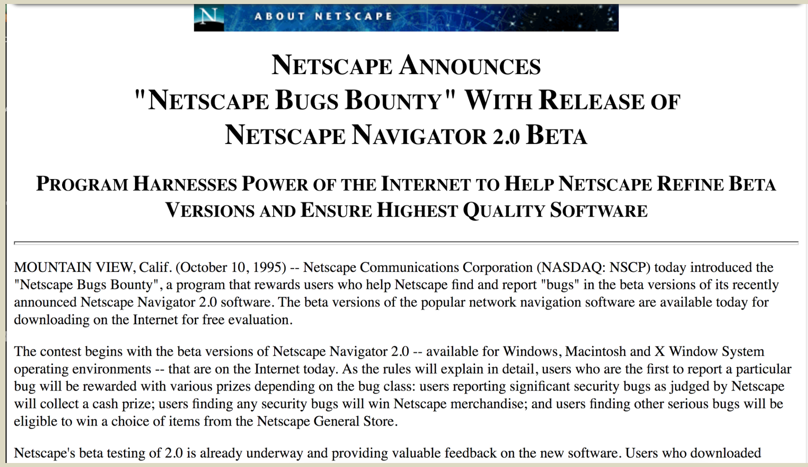
Process maturity: High

ROMA MMXVI

APPSEC EUROPE

One of my projects has been to reboot the process around our bug intake and workflow, this is what it looks like now

Mozilla Web Bug Bounty


ABOUT NETSCAPE

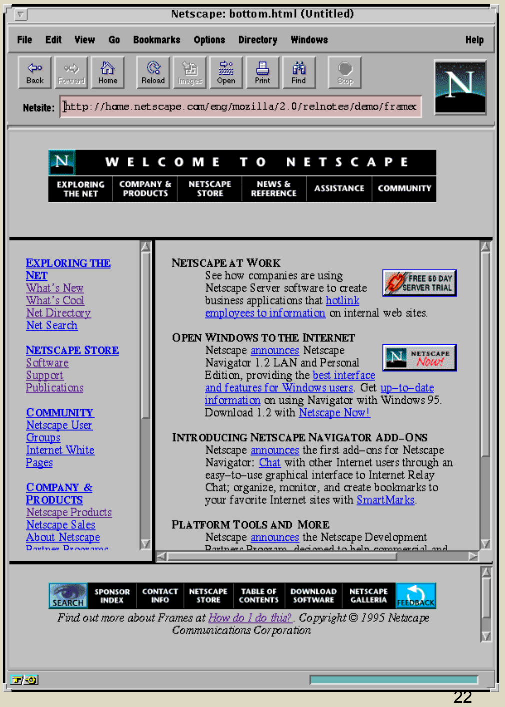
Netscape Announces "Netscape Bugs Bounty" With Release of Netscape Navigator 2.0 Beta

Program Harnesses Power of the Internet to Help Netscape Refine Beta Versions and Ensure Highest Quality Software

MOUNTAIN VIEW, Calif. (October 10, 1995) -- Netscape Communications Corporation (NASDAQ: NSCP) today introduced the "Netscape Bugs Bounty", a program that rewards users who help Netscape find and report "bugs" in the beta versions of its recently announced Netscape Navigator 2.0 software. The beta versions of the popular network navigation software are available today for downloading on the Internet for free evaluation.

The contest begins with the beta versions of Netscape Navigator 2.0 -- available for Windows, Macintosh and X Window System operating environments -- that are on the Internet today. As the rules will explain in detail, users who are the first to report a particular bug will be rewarded with various prizes depending on the bug class: users reporting significant security bugs as judged by Netscape will collect a cash prize; users finding any security bugs will win Netscape merchandise; and users finding other serious bugs will be eligible to win a choice of items from the Netscape General Store.

Netscape's beta testing of 2.0 is already underway and providing valuable feedback on the new software. Users who downloaded



http://home.netscape.com/eng/mozilla/2.0/relnotes/demo/framex

EXPLORING THE NET
What's New
What's Cool
Net Directory
Net Search

NETSCAPE STORE
Software
Support
Publications

COMMUNITY
Netscape User Groups
Internet White Pages

COMPANY & PRODUCTS
Netscape Products
Netscape Sales
About Netscape
Partner Programs

Netscape at Work
See how companies are using Netscape Server software to create business applications that hotlink employees to information on internal web sites.

Open Windows to the Internet
Netscape announces Netscape Navigator 1.2 LAN and Personal Edition, providing the best interface and features for Windows users. Get up-to-date information on using Navigator with Windows 95. Download 1.2 with Netscape Now!

Introducing Netscape Navigator Add-Ons
Netscape announces the first add-ons for Netscape Navigator: Chat with other Internet users through an easy-to-use graphical interface to Internet Relay Chat; organize, monitor, and create bookmarks to your favorite Internet sites with SmartMarks.

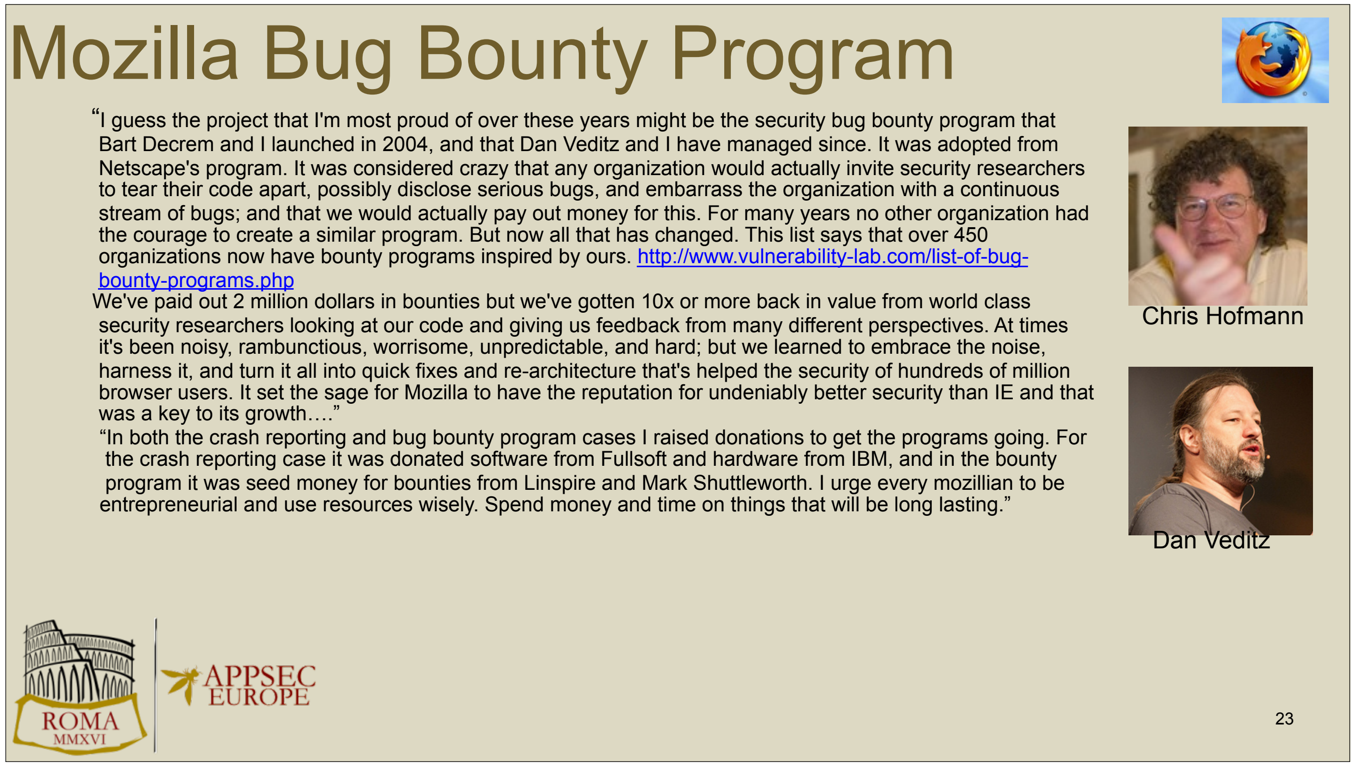
Platform Tools and More
Netscape announces the Netscape Development Partners Program, designed to help commercial and

Find out more about Frames at How do I do this?. Copyright © 1995 Netscape Communications Corporation

Bug bounty is one of the centerpieces of our web app sec program

Mozilla was born from Netscape.
Not going to retell the whole story of how Mozilla was born from Netscape Navigator, but it's an interesting tale you can look up yoursel.

The first bounty program was called the "Bugs Bounty," It was created by a technical support engineer named Jarrett Ridlinghafer in 1995 for the launch of Netscape Navigator 2.0 Beta. He also created the first community support forum for the product.



Mozilla Bug Bounty Program

"I guess the project that I'm most proud of over these years might be the security bug bounty program that Bart Decrem and I launched in 2004, and that Dan Veditz and I have managed since. It was adopted from Netscape's program. It was considered crazy that any organization would actually invite security researchers to tear their code apart, possibly disclose serious bugs, and embarrass the organization with a continuous stream of bugs; and that we would actually pay out money for this. For many years no other organization had the courage to create a similar program. But now all that has changed. This list says that over 450 organizations now have bounty programs inspired by ours. http://www.vulnerability-lab.com/list-of-bug-bounty-programs.php

We've paid out 2 million dollars in bounties but we've gotten 10x or more back in value from world class security researchers looking at our code and giving us feedback from many different perspectives. At times it's been noisy, rambunctious, worrisome, unpredictable, and hard; but we learned to embrace the noise, harness it, and turn it all into quick fixes and re-architecture that's helped the security of hundreds of million browser users. It set the sage for Mozilla to have the reputation for undeniably better security than IE and that was a key to its growth…."

"In both the crash reporting and bug bounty program cases I raised donations to get the programs going. For the crash reporting case it was donated software from Fullsoft and hardware from IBM, and in the bounty program it was seed money for bounties from Linspire and Mark Shuttleworth. I urge every mozillian to be entrepreneurial and use resources wisely. Spend money and time on things that will be long lasting."

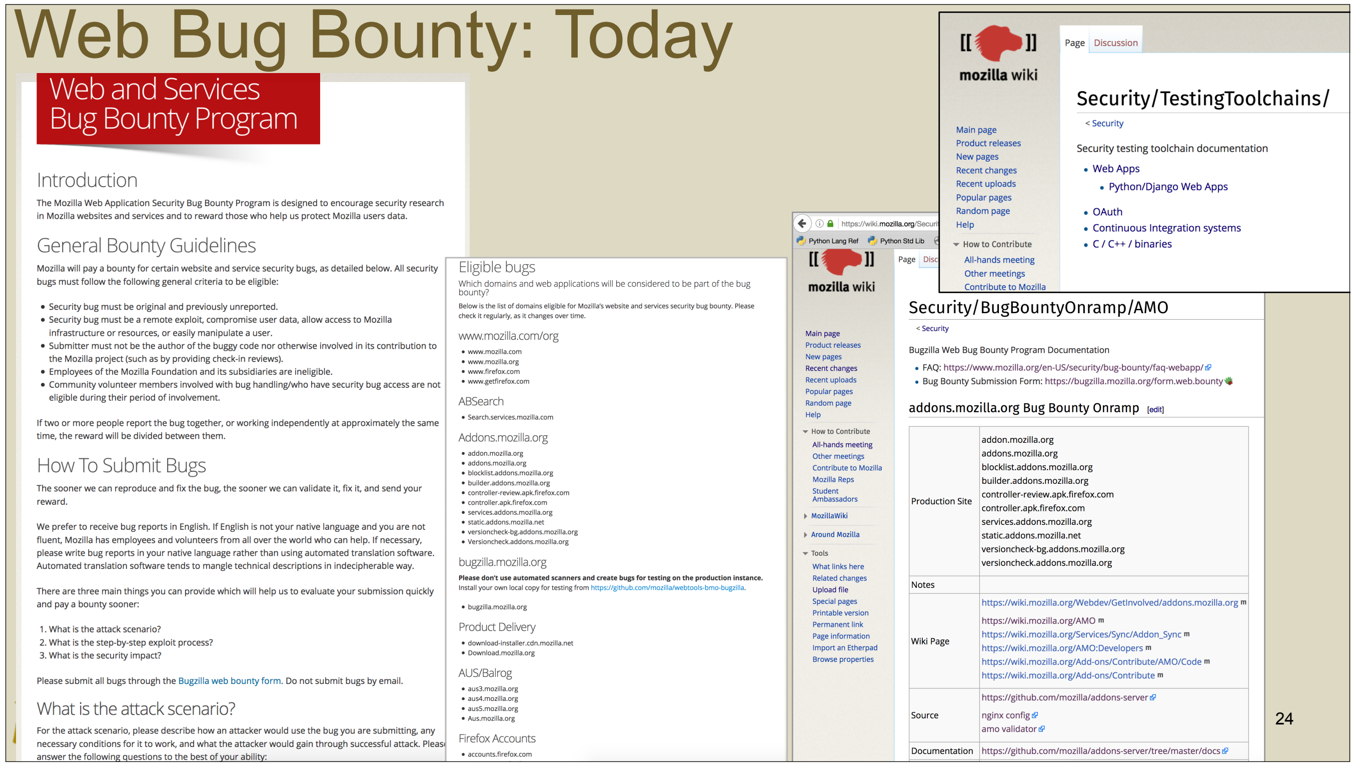
Chris Hofmann

Dan Veditz

Chris Hofmann and Dan Veditz started the modern Mozilla bug bounty program in 2004, the web bounty came a few years later.
Chris recently left Mozilla after 10 years so he could sail more often, though I wouldn't be surprised if he came back.
He said I could share his parting thoughts,Chris probably had more continuous years of working on a browser than any other living person, 20 years, he goes back to the beginnings of Netscape Navigator. He also started Mozilla's automated crash reporting system, mobile engineering, and ran the l10n internationalization effort - 60% of Mozilla Firefox users are not English speakers.

Dan Veditz does security engineering for the Firefox platform, he's a walking encyclopedia of browser security history and has contributed to many of the RFCs that define the security properties of HTTP and the web ecosystem

Web Bug Bounty: Today



Web and Services Bug Bounty Program

Introduction

The Mozilla Web Application Security Bug Bounty Program is designed to encourage security research in Mozilla websites and services and to reward those who help us protect Mozilla users data.

General Bounty Guidelines

Mozilla will pay a bounty for certain website and service security bugs, as detailed below. All security bugs must follow the following general criteria to be eligible:

- Security bug must be original and previously unreported.
- Security bug must be a remote exploit, compromise user data, allow access to Mozilla infrastructure or resources, or easily manipulate a user.
- Submitter must not be the author of the buggy code nor otherwise involved in its contribution to the Mozilla project (such as by providing check-in reviews).
- Employees of the Mozilla Foundation and its subsidiaries are ineligible.
- Community volunteer members involved with bug handling/who have security bug access are not eligible during their period of involvement.

If two or more people report the bug together, or working independently at approximately the same time, the reward will be divided between them.

How To Submit Bugs

The sooner we can reproduce and fix the bug, the sooner we can validate it, fix it, and send your reward.

We prefer to receive bug reports in English. If English is not your native language and you are not fluent, Mozilla has employees and volunteers from all over the world who can help. If necessary, please write bug reports in your native language rather than using automated translation software. Automated translation software tends to mangle technical descriptions in indecipherable way.

There are three main things you can provide which will help us to evaluate your submission quickly and pay a bounty sooner:

1. What is the attack scenario?
2. What is the step-by-step exploit process?
3. What is the security impact?

Please submit all bugs through the Bugzilla web bounty form. Do not submit bugs by email.

What is the attack scenario?

For the attack scenario, please describe how an attacker would use the bug you are submitting, any necessary conditions for it to work, and what the attacker would gain through successful attack. Please answer the following questions to the best of your ability:

Eligible bugs

Which domains and web applications will be considered to be part of the bug bounty?

Below is the list of domains eligible for Mozilla's website and services security bug bounty. Please check it regularly, as it changes over time.

www.mozilla.com/org

- www.mozilla.com
- www.mozilla.org
- www.firefox.com
- www.getfirefox.com

ABSearch

- Search.services.mozilla.com

Addons.mozilla.org

- addon.mozilla.org
- addons.mozilla.org
- blocklist.addons.mozilla.org
- builder.addons.mozilla.org
- controller-review.apk.firefox.com
- controller.apk.firefox.com
- services.addons.mozilla.org
- static.addons.mozilla.net
- versioncheck-bg.addons.mozilla.org
- Versioncheck.addons.mozilla.org

bugzilla.mozilla.org

Please don't use automated scanners and create bugs for testing on the production instance. Install your own local copy for testing from https://github.com/mozilla/webtools-bmo-bugzilla.

- bugzilla.mozilla.org

Product Delivery

- download-installer.cdn.mozilla.net
- Download.mozilla.org

AUS/Balrog

- aus3.mozilla.org
- aus4.mozilla.org
- aus5.mozilla.org
- Aus.mozilla.org

Firefox Accounts

- accounts.firefox.com

Security/BugBountyOnramp/AMO

< Security

Bugzilla Web Bug Bounty Program Documentation

- FAQ: https://www.mozilla.org/en-US/security/bug-bounty/faq-webapp/
- Bug Bounty Submission Form: https://bugzilla.mozilla.org/form.web.bounty

addons.mozilla.org Bug Bounty Onramp [edit]

Production Site	addon.mozilla.org addons.mozilla.org blocklist.addons.mozilla.org builder.addons.mozilla.org controller-review.apk.firefox.com controller.apk.firefox.com services.addons.mozilla.org static.addons.mozilla.net versioncheck-bg.addons.mozilla.org versioncheck.addons.mozilla.org
Notes	
Wiki Page	https://wiki.mozilla.org/Webdev/GetInvolved/addons.mozilla.org https://wiki.mozilla.org/AMO https://wiki.mozilla.org/Services/Sync/Addon_Sync https://wiki.mozilla.org/AMO:Developers https://wiki.mozilla.org/Add-ons/Contribute/AMO/Code https://wiki.mozilla.org/Add-ons/Contribute
Source	https://github.com/mozilla/addons-server nginx config amo validator
Documentation	https://github.com/mozilla/addons-server/tree/master/docs

Security/TestingToolchains/

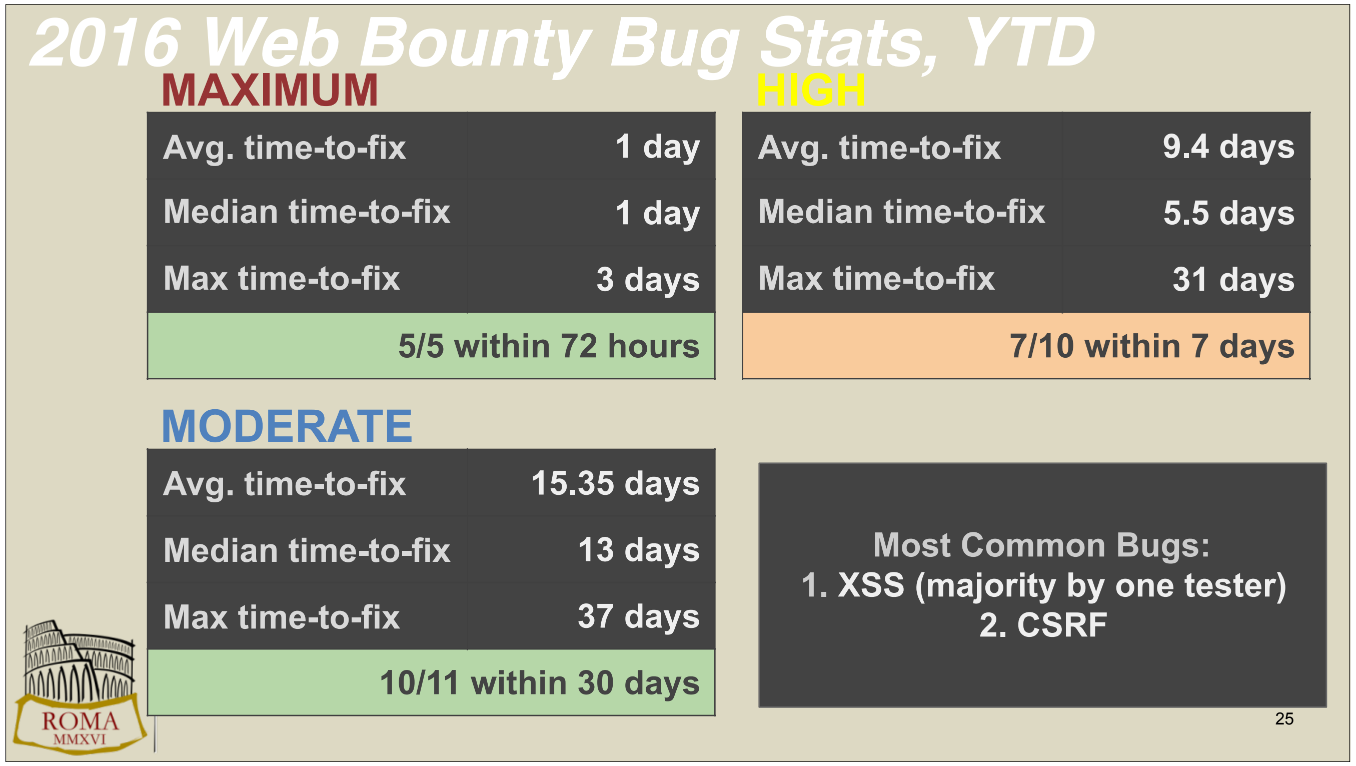
< Security

Security testing toolchain documentation

- Web Apps
  - Python/Django Web Apps
- OAuth
- Continuous Integration systems
- C / C++ / binaries

One of my goals with the program is to shift the curve to more high value, difficult bugs, and get close as we can to eliminating bugs we should never produce, such as XSS, SQLI, or remote OS command injection.

- Generate metrics about vulnerability trends in websites and vulnerability categories to inform the direction of security efforts <- information to help plan the overall appsec effort
- Provide an entry point for Mozilla Enterprise Information Security to help support security for all Mozilla web developers. Including community sites Mozilla the company doesn't run directly, there are about ~3000. I'm still working on a complete list. ←---
- Invite participation through detailed reporting instructions and information to bug bounty hunters, allowing bug hunters to do deeper reviews and reducing our time-to-fix ←-- community



2016 Web Bounty Bug Stats, YTD

MAXIMUM

Avg. time-to-fix	1 day
Median time-to-fix	1 day
Max time-to-fix	3 days
	5/5 within 72 hours

HIGH

Avg. time-to-fix	9.4 days
Median time-to-fix	5.5 days
Max time-to-fix	31 days
	7/10 within 7 days

MODERATE

Avg. time-to-fix	15.35 days
Median time-to-fix	13 days
Max time-to-fix	37 days
	10/11 within 30 days

Most Common Bugs:
1. XSS (majority by one tester)
2. CSRF

The 31 day HIGH bug had been patched and wasn't exploitable after a couple days, but there was additional follow-on work that kept the bug open a little longer.

I changed a few things about how we run the web bounty -

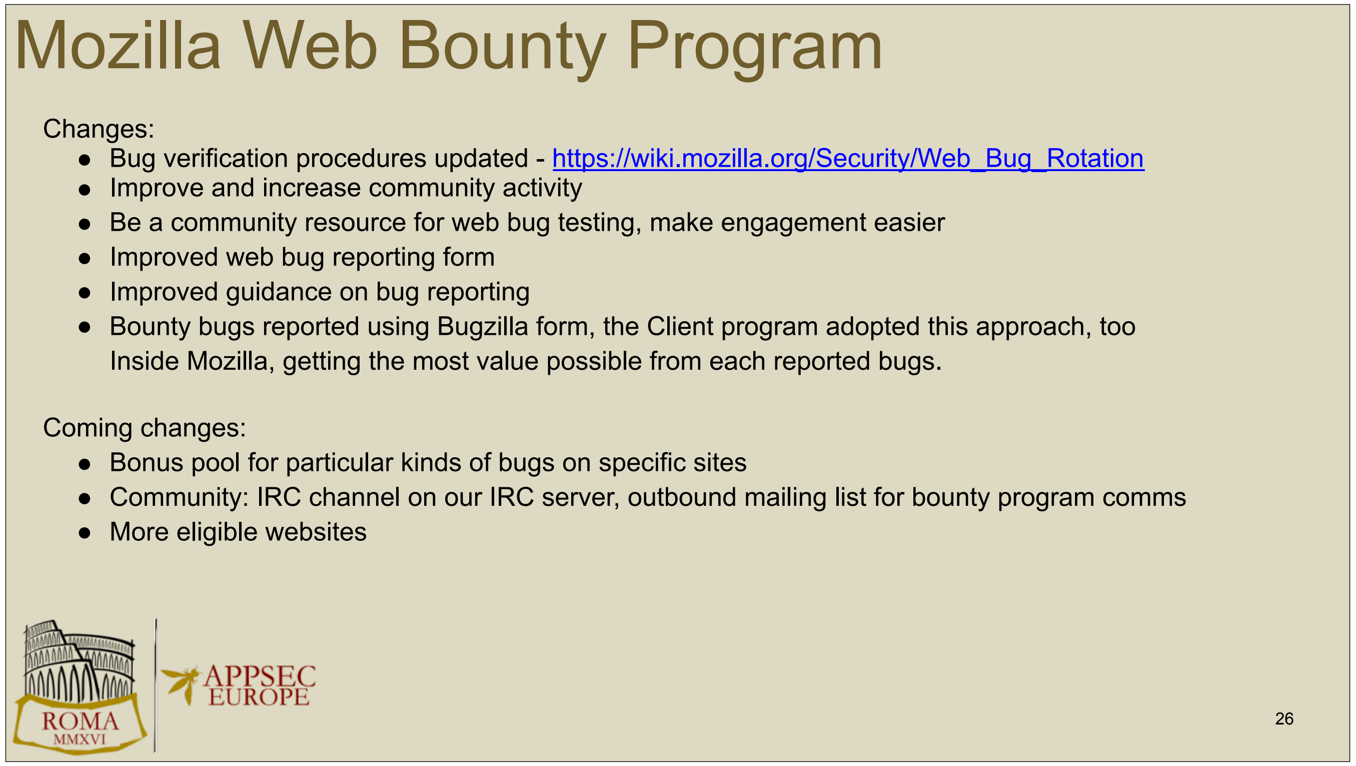
One of the biggest was to pay on verification, not on fix, even though the typical "High" vuln gets fixed within a few days of reporting, now. Slow response times and failure to fix make bug researchers unhappy, it increases the chances they will be told their bug is a dupe.

Another was to have all externally reported bugs come in through a form on Bugzilla, not by email as was the case in the past. The StartTLS flag-stripping bug is real. GPG is nifty but I wanted to make bug reporting as easy as possible. I also rewrote the bug submission directions to help guide bounty hunters report more efficiently - the better the report, the faster we can fix. BugCrowd's recent report says that they see about 45% invalid submissions and 36% duplicates.

We see few HIGH risk invalid bugs since the changes since the changes. The dupes are most often for things like text injection bugs that we've "wontfixed" because they don't look convincing enough to trick someone into doing something bad. We use some external vendors to host and manage a few services for us, we've gotten them to be a lot more responsive and in one case that had persistent XSS that were repeatedly reported but that the vendor ignored and we received many dupe reports, we fired them. It's not safe for our users and it's not respectful of bounty hunters time.

Web Bug Rotation

I picked Mondays because that means I get to work 3 days worth of bugs per week.



Mozilla Web Bounty Program

Changes:

- Bug verification procedures updated - https://wiki.mozilla.org/Security/Web_Bug_Rotation
- Improve and increase community activity
- Be a community resource for web bug testing, make engagement easier
- Improved web bug reporting form
- Improved guidance on bug reporting
- Bounty bugs reported using Bugzilla form, the Client program adopted this approach, too
  Inside Mozilla, getting the most value possible from each reported bugs.

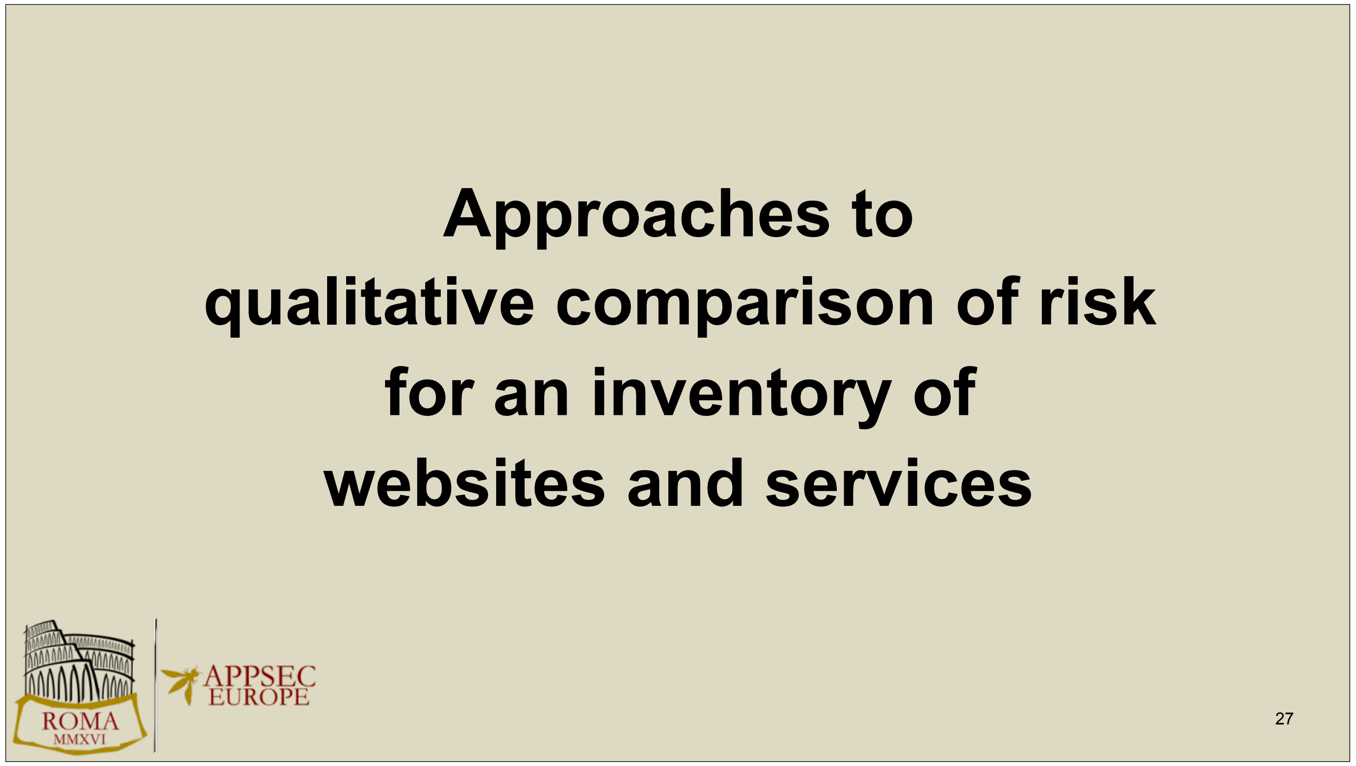
Coming changes:

- Bonus pool for particular kinds of bugs on specific sites
- Community: IRC channel on our IRC server, outbound mailing list for bounty program comms
- More eligible websites

Bounty hunters shouldn’t have to hunt for sources and tech information.

Mozilla and community developed testing workflow documentation on Wiki, detailed into on apps, I appreciate the time they spend, and want to make their use of time as efficient as possible.

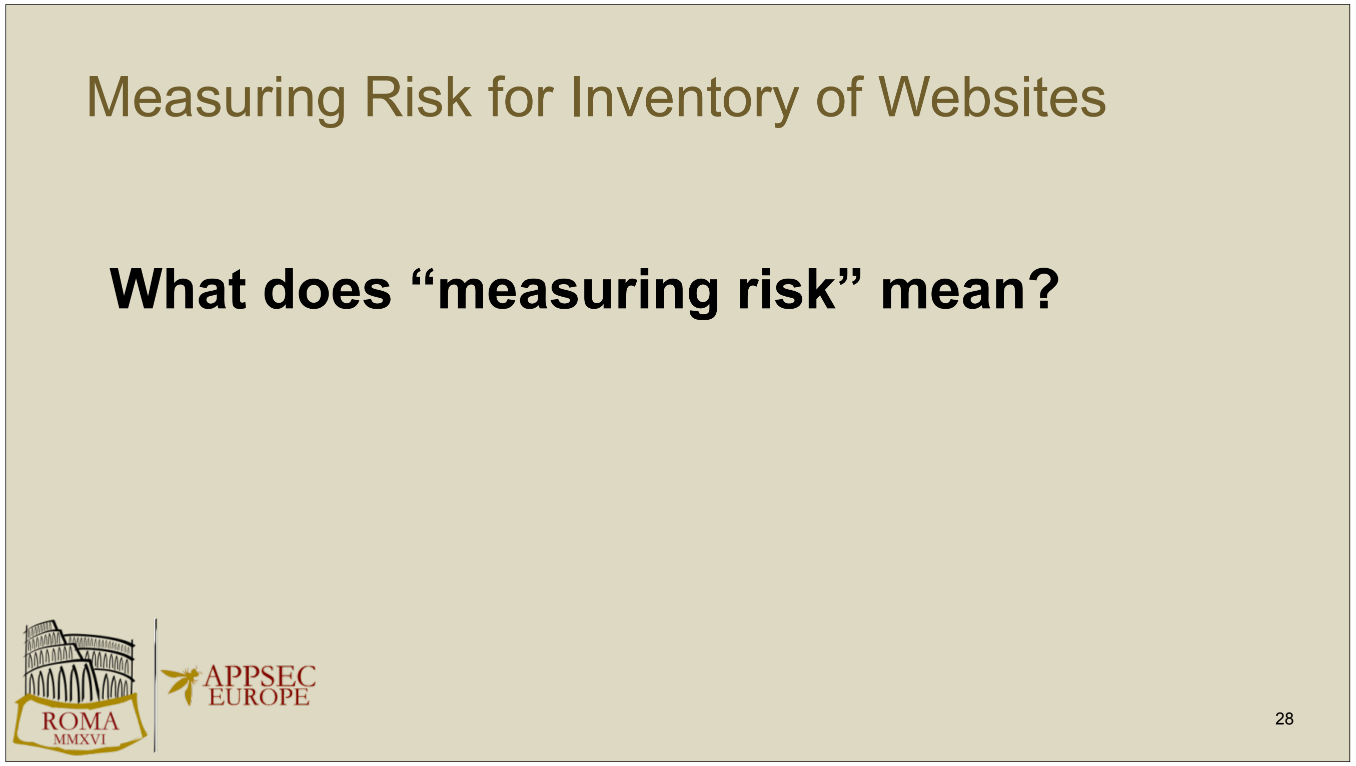
Upcoming MWOC project for college students - testing workflow for python web apps with known properties



Approaches to qualitative comparison of risk for an inventory of websites and services

Most of this part of the talk is about the knowledge problem that makes this so difficult, it’s full of pitfalls and roadblocks.

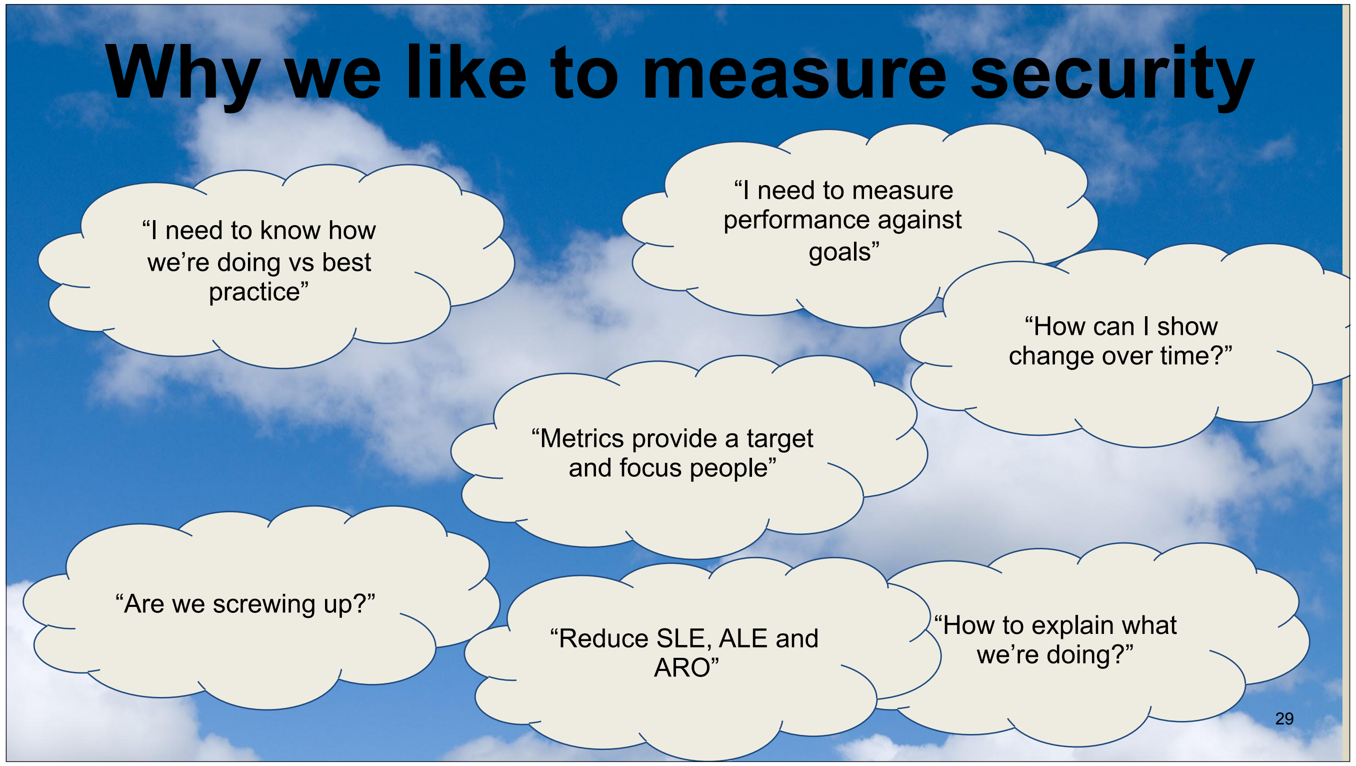
Then, we’ll talk unfortunately a lot more briefly about the possible.



Measuring Risk for Inventory of Websites

What does “measuring risk” mean?

Or, Reason 65537 why I let my CISSP expire after trying unsuccessfully to give it back



Why we like to measure security

"I need to know how we're doing vs best practice"

"I need to measure performance against goals"

"How can I show change over time?"

"Metrics provide a target and focus people"

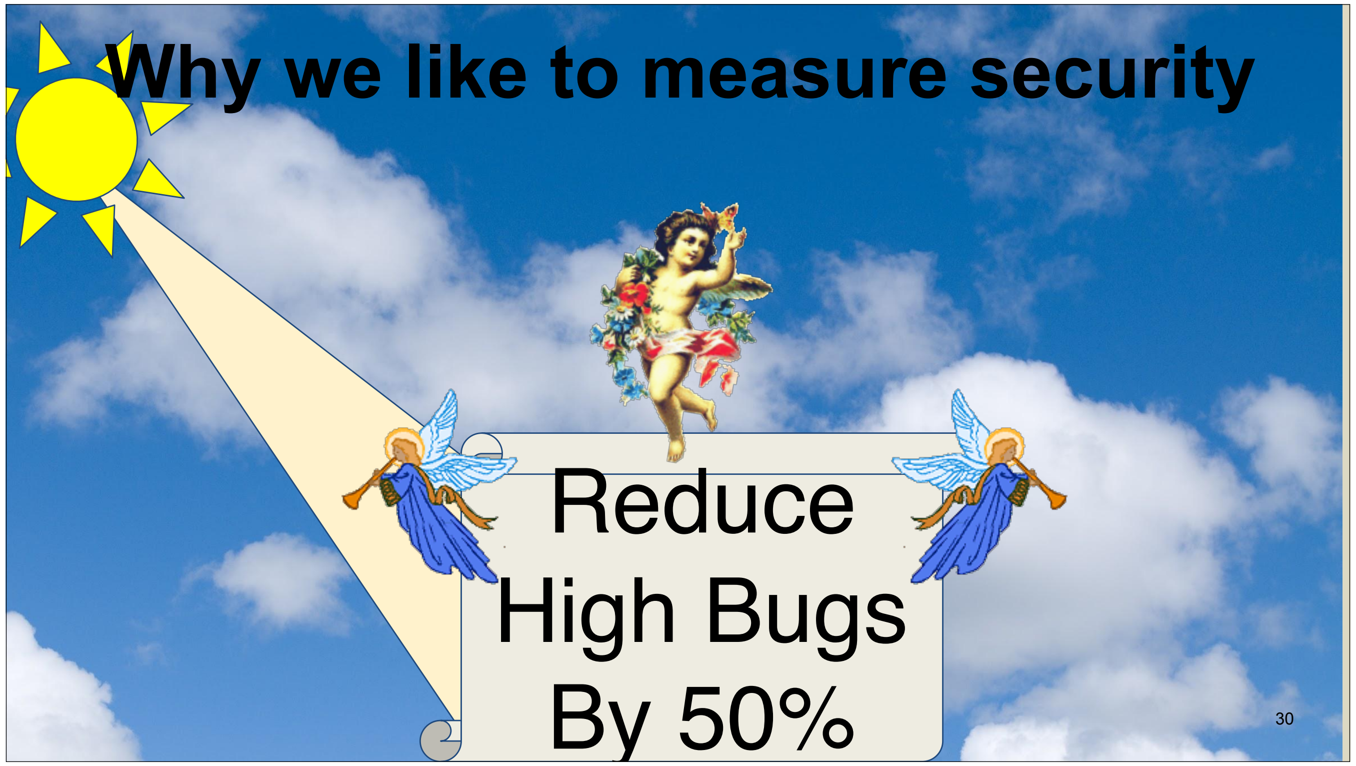
"Are we screwing up?"

"Reduce SLE, ALE and ARO"

"How to explain what we're doing?"

A focus on measures and metrics is often used to reach some desired state. This probably sounds familiar:

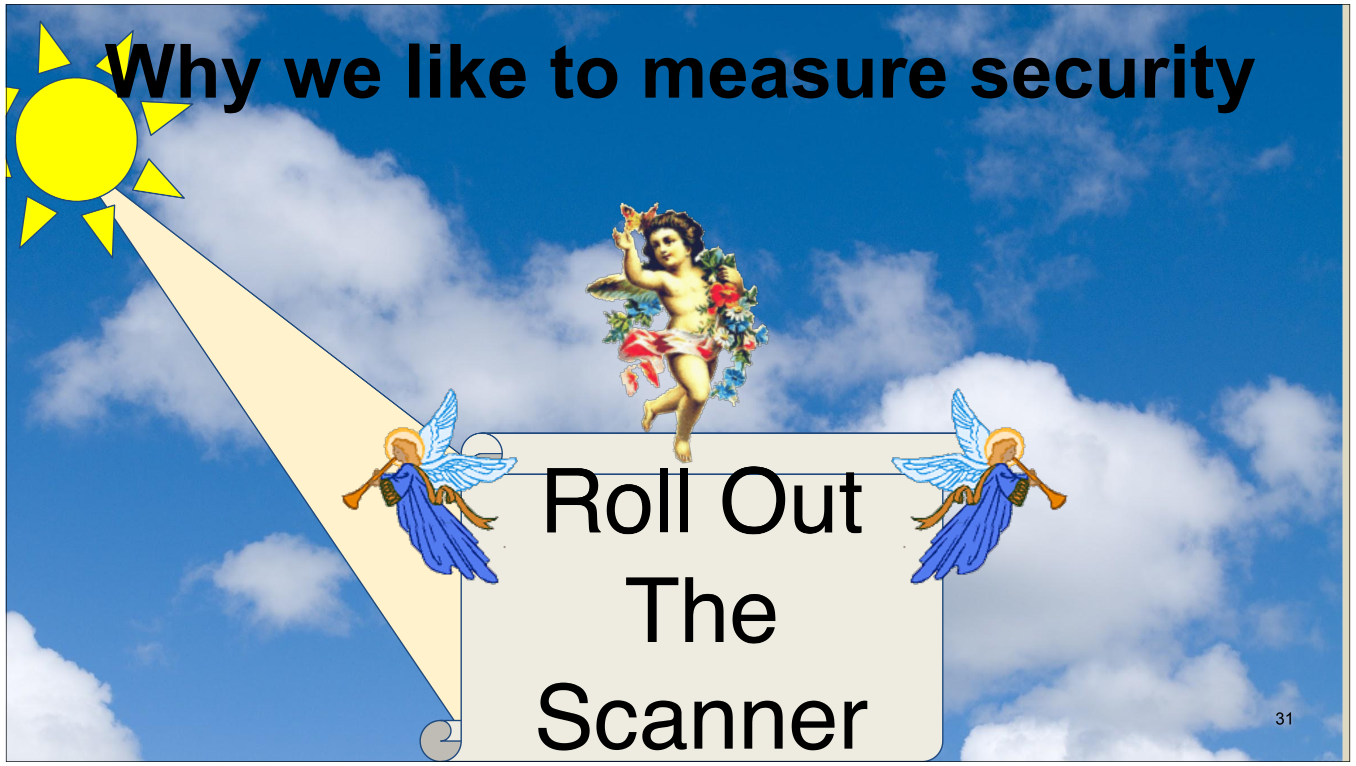
1. Management defines goal and comes up with a measure
2. Management establishes quarterly and annual targets
3. Management communicates the target, in terms of agreed measure
4. People do what they are being measured on



Why we like to measure security

Reduce
High Bugs
By 50%

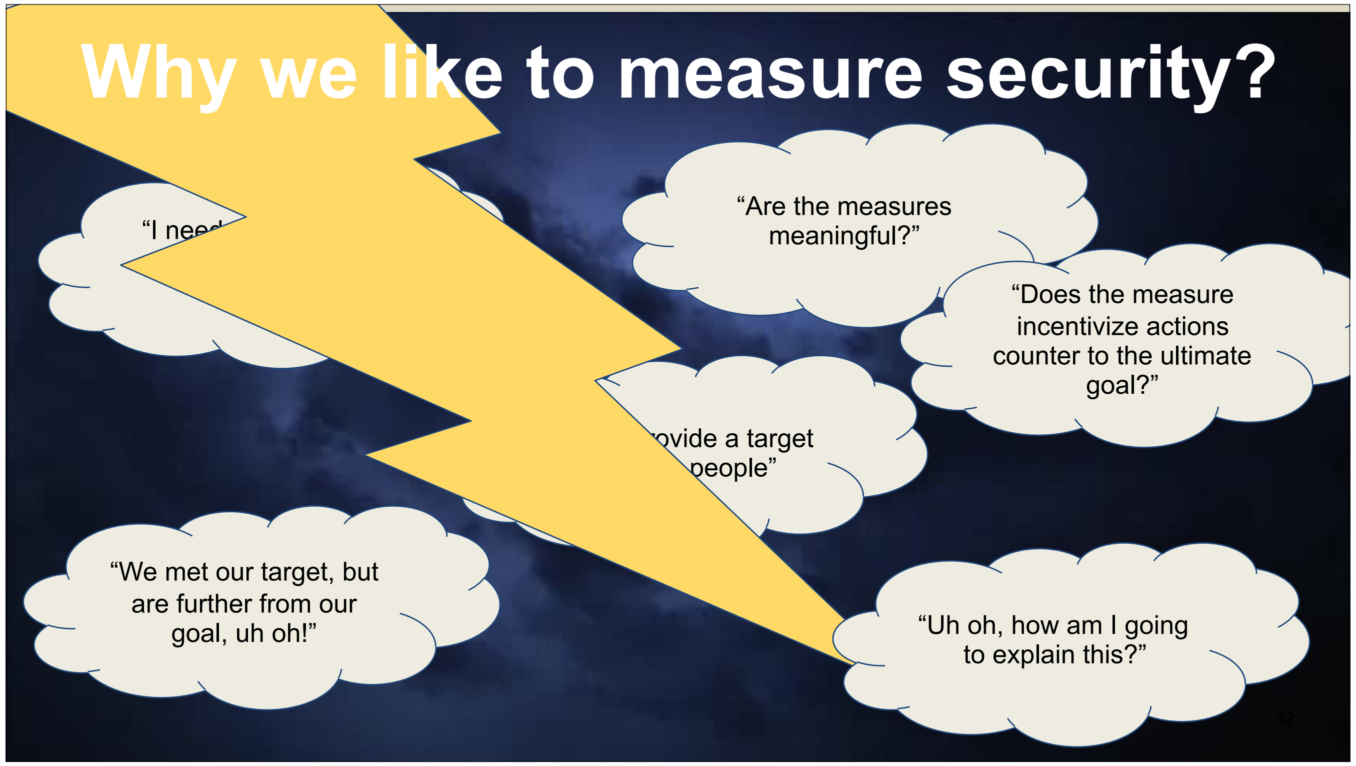
It usually looks something like this



Why we like to measure security

Roll Out The Scanner

or like this



Why we like to measure security?

“I need

“Are the measures meaningful?”

“Does the measure incentivize actions counter to the ultimate goal?”

ovide a target people”

“We met our target, but are further from our goal, uh oh!”

“Uh oh, how am I going to explain this?”

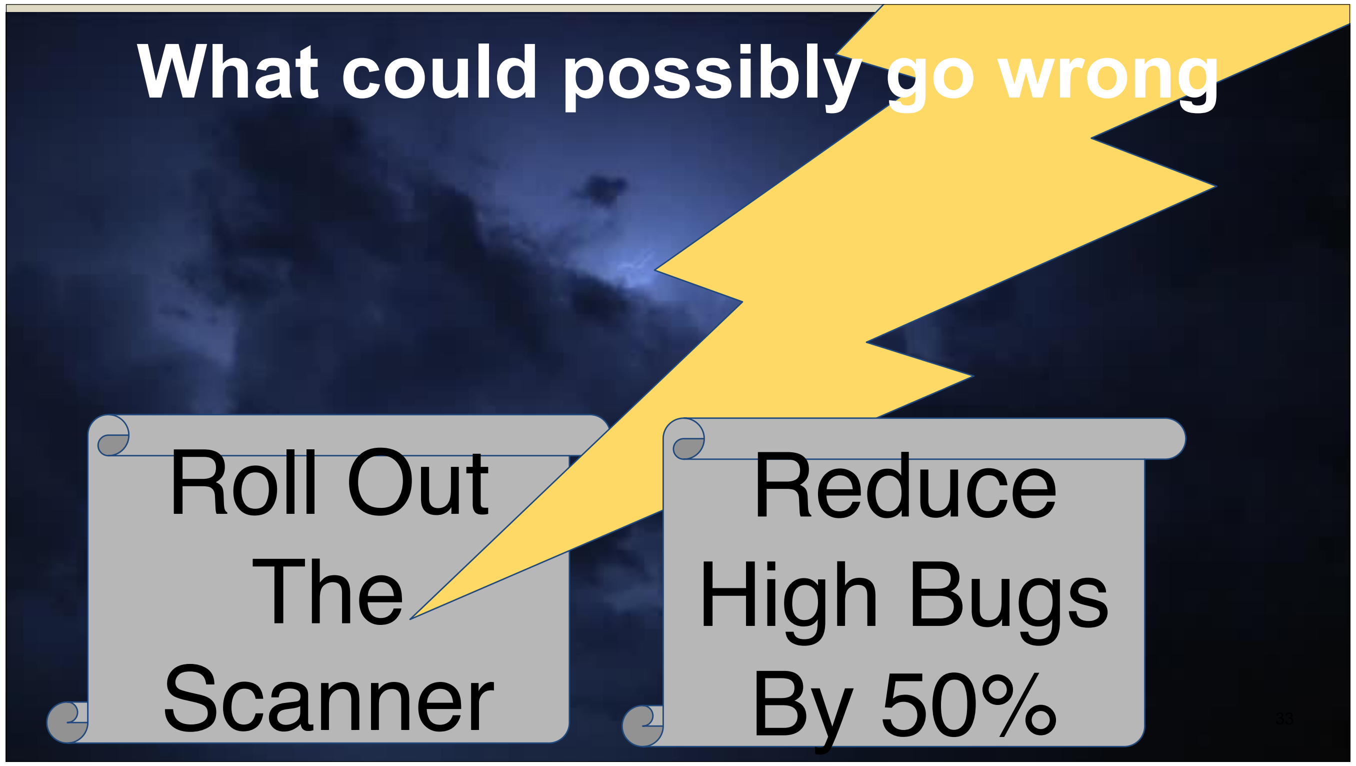
The process sounds logical but can really be counter-productive.

Overloading a metric for too many purposes leads to unintended consequences.

Performance metrics: Usually linked to performance targets, they substitute a number for a well-articulated goal. Unfortunately it’s only tracking progress towards the decided upon metric, often with unhappy consequences.

Best Practice measurements: I don’t like that phrase because of the word best. Best is subjectively objective.
 Rarely is there any objective measurement of various options, rarely is evidence collected and analyzed to demonstrate that a practice is better than another, to some end. This uses metrics as both a target and measure of performance. Implicitly this primes the brain to assume that “best practice” really is, not to think about whether it’s appropriate to the organization and its challenges.

Metrics target: The target is often the entire explanation of the goal. They’re easier to explain than complex goals. A metrics target like “reduce bugs by 50%” sounds clear but it’s ultimately an arbitrary number designed to appeal to the management of the definer of the metrics as much as it’s tied to a real organizational goal.



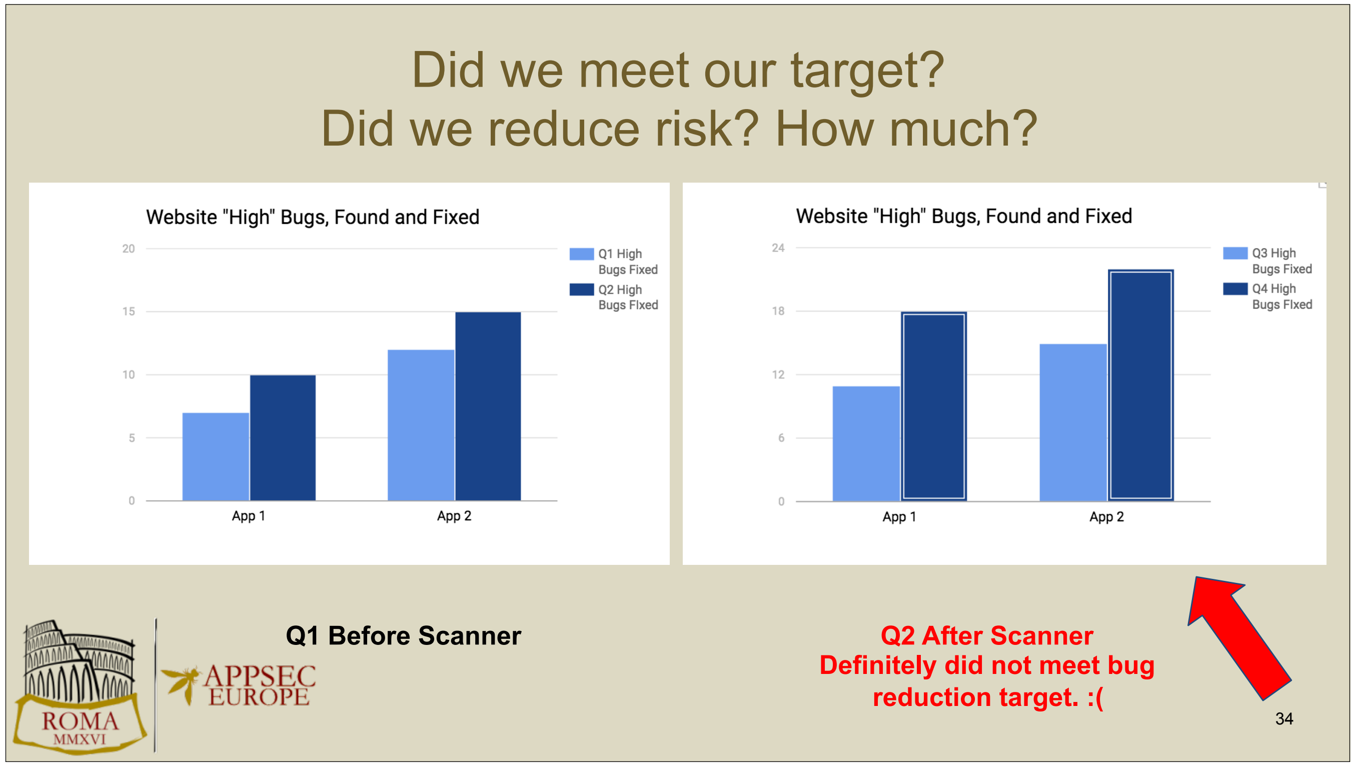
What could possibly go wrong

Roll Out The Scanner

Reduce High Bugs By 50%

If your goal was ‘Reduce bugs by 50%’ and you implement a web app scanner that increases open bugs by 60%, did you succeed or fail?

Did we meet our target?
Did we reduce risk? How much?



Q1 Before Scanner

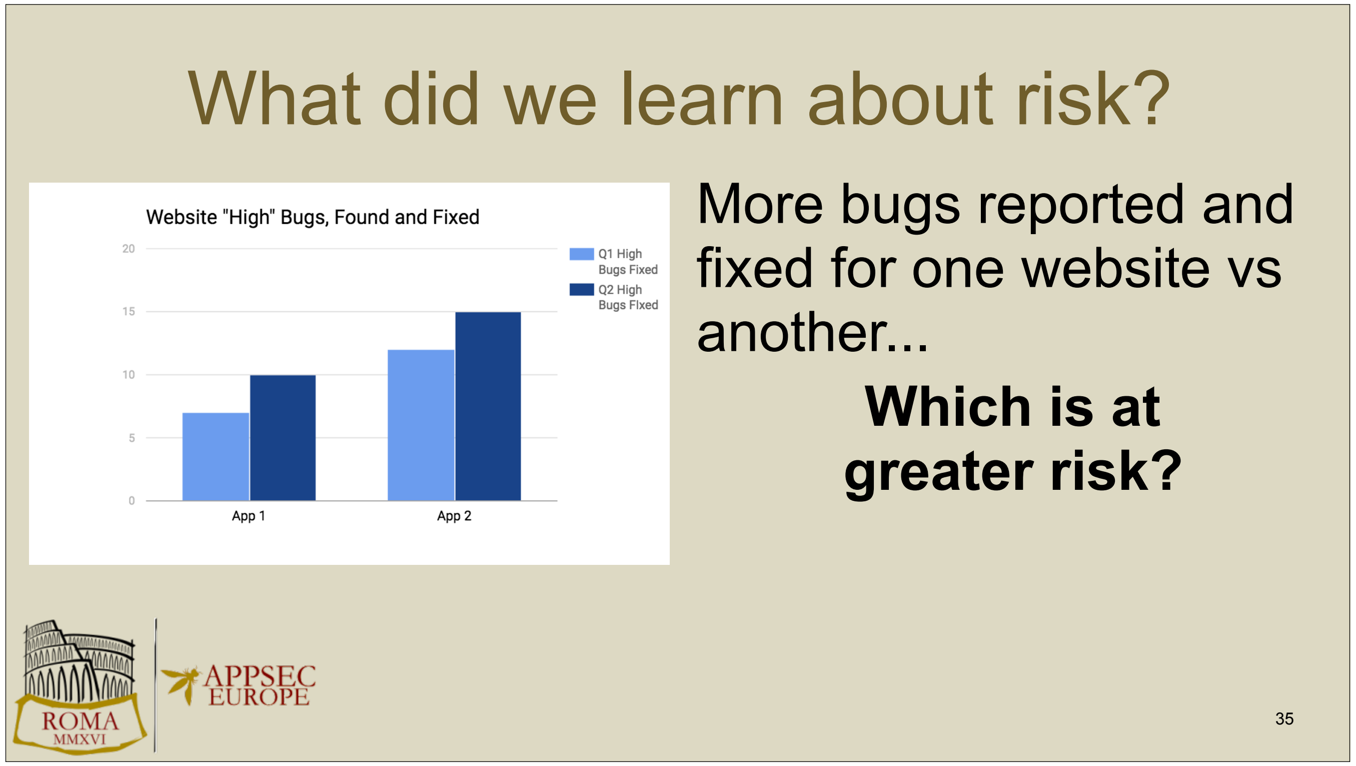
Q2 After Scanner
Definitely did not meet bug reduction target. :(

No way to know.

We can account for the addition of the scanner

There are too many variables that aren't and **can't** be accounted for.

- New code
- False negatives
- Attackers develop new tools and techniques
- Insider threat
- Platform issues
- 3rd party code
- etc etc

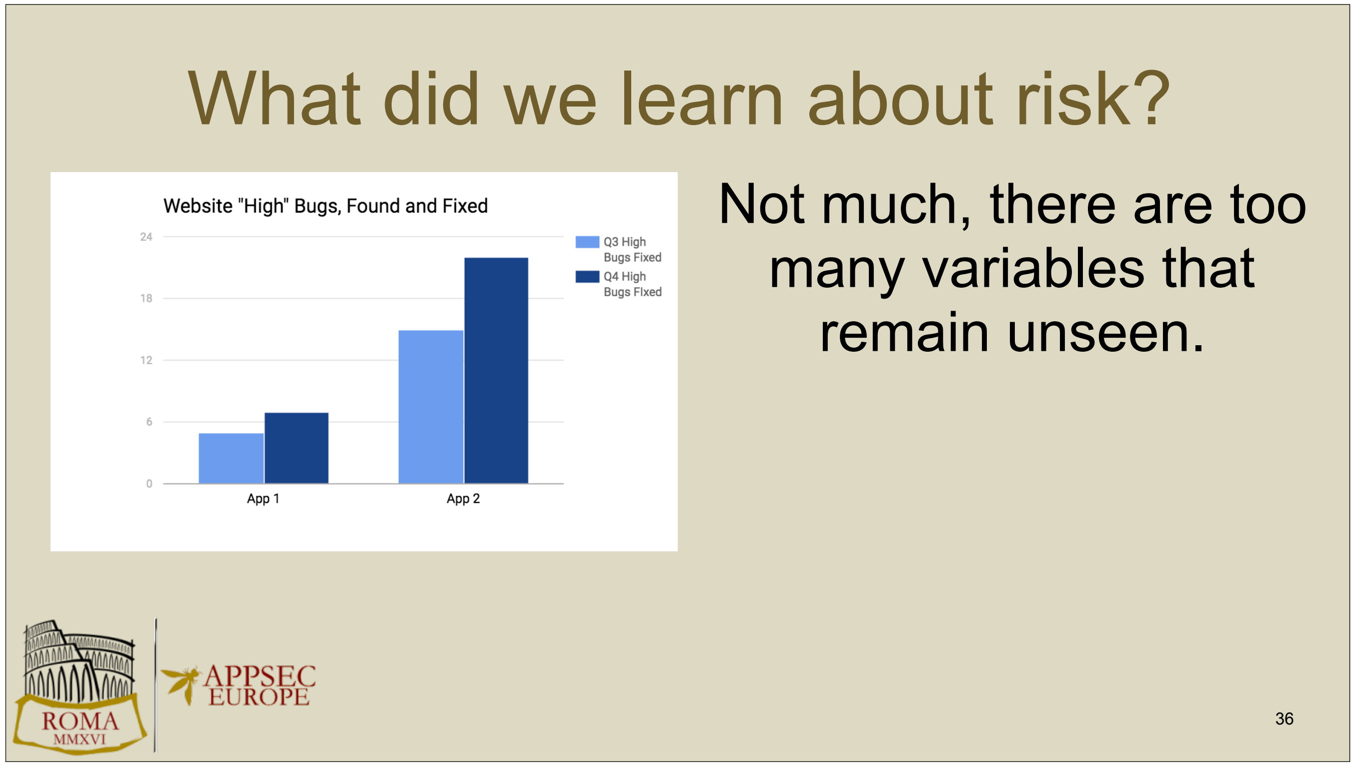


What did we learn about risk?

More bugs reported and fixed for one website vs another...

Which is at greater risk?

No way to know, there are too many variables that aren’t and **can’t** be accounted for
It’s too easy to assume that the first is safer, this is observational bias, also called the “streetlight effect.”



What did we learn about risk?

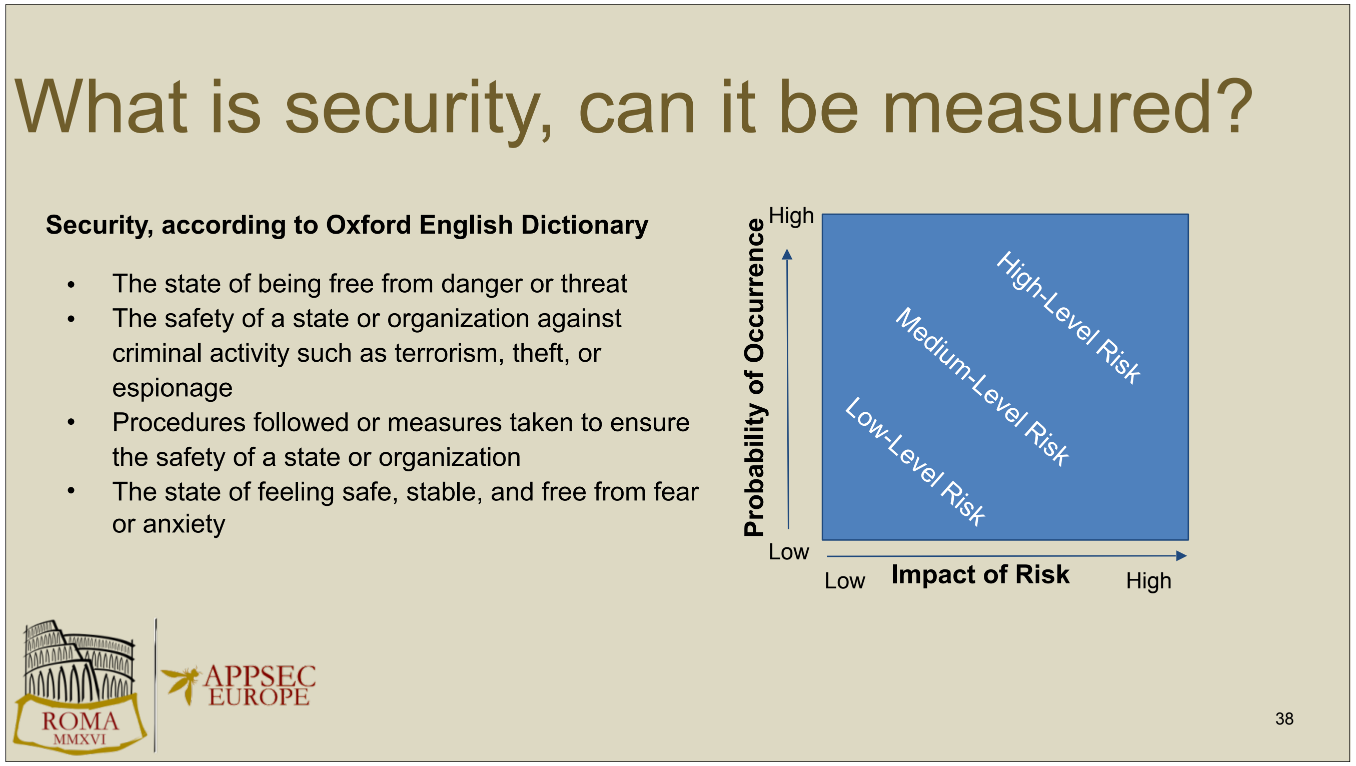
Not much, there are too many variables that remain unseen.

A policeman sees a dunk searching under a streetlight for something and asks what he lost.
The drunk man says that he lost his housekeys, the policeman helps him search.
After a while the policeman asks the drunk, “Are you sure you lost your keys here?”
“No,” says the drunk man, “I’m not really certain, I think I parked them in the lost, burp, lost them in the park.”
“So why are searching here??!” asks the Policeman?
The drunk response, “Because this is where the light is…”

What is security, can it be measured?

Security, according to Oxford English Dictionary

- The state of being free from danger or threat
- The safety of a state or organization against criminal activity such as terrorism, theft, or espionage
- Procedures followed or measures taken to ensure the safety of a state or organization
- The state of feeling safe, stable, and free from fear or anxiety



What is security, can it be measured?

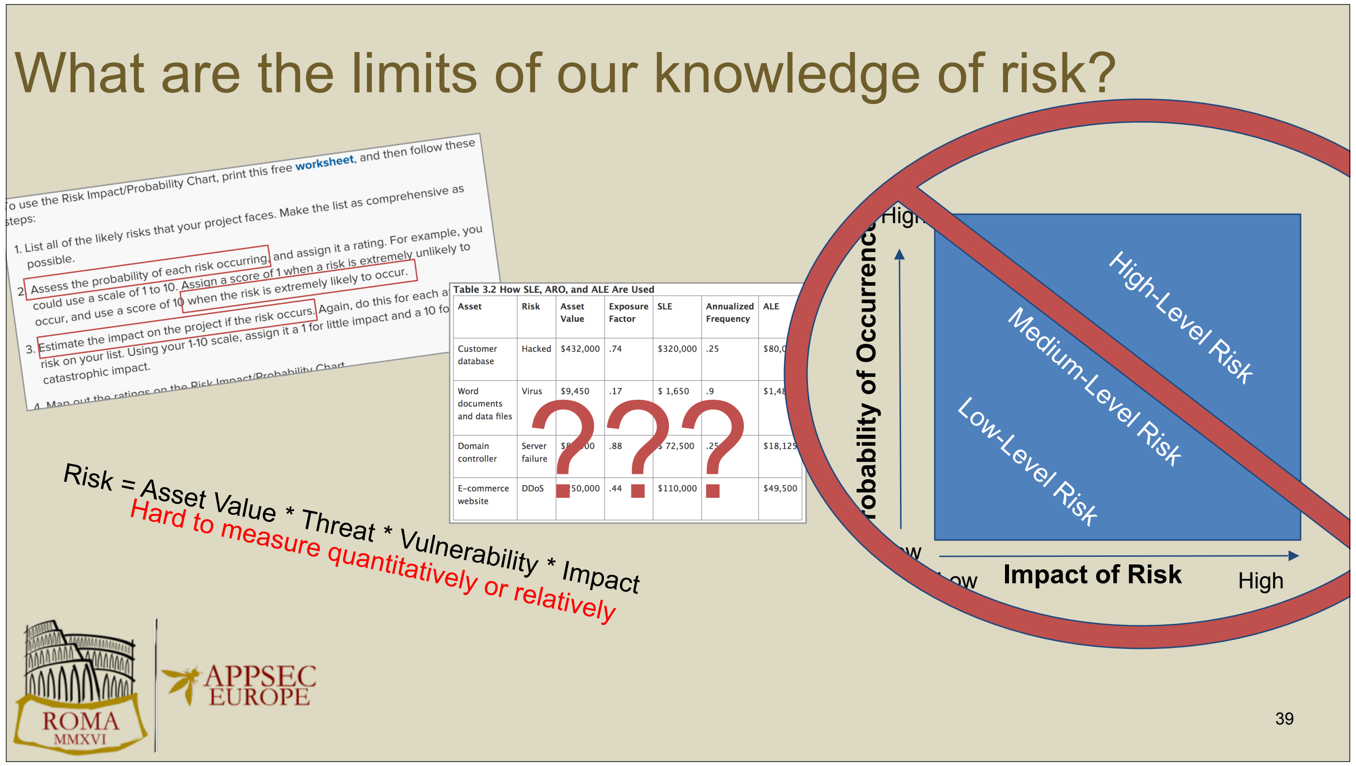
Security, according to Oxford English Dictionary

- The state of being free from danger or threat
- The safety of a state or organization against criminal activity such as terrorism, theft, or espionage
- Procedures followed or measures taken to ensure the safety of a state or organization
- The state of feeling safe, stable, and free from fear or anxiety

“Should you ignore a 49 percent probability risk, which will cause a 49 percent of maximum loss?
And why, in this example, should you pay maximum attention to a risk that has a 51 percent probability of occurring, with a loss of 51 percent of maximum loss?”

How much can we really know about any of this?
With what confidence interval?

What are the limits of our knowledge of risk?



o use the Risk Impact/Probability Chart, print this free **worksheet**, and then follow these steps:

1. List all of the likely risks that your project faces. Make the list as comprehensive as possible.
2. Assess the probability of each risk occurring, and assign it a rating. For example, you could use a scale of 1 to 10. Assign a score of 1 when a risk is extremely unlikely to occur, and use a score of 10 when the risk is extremely likely to occur.
3. Estimate the impact on the project if the risk occurs. Again, do this for each a risk on your list. Using your 1-10 scale, assign it a 1 for little impact and a 10 fo catastrophic impact.
4. Map out the ratings on the Risk Impact/Probability Chart

Table 3.2 How SLE, ARO, and ALE Are Used

Asset	Risk	Asset Value	Exposure Factor	SLE	Annualized Frequency	ALE
Customer database	Hacked	$432,000	.74	$320,000	.25	$80,0
Word documents and data files	Virus	$9,450	.17	$ 1,650	.9	$1,48
Domain controller	Server failure	[illegible]	.88	$ 72,500	.25	$18,125
E-commerce website	DDoS	[illegible]50,000	.44	$110,000	[illegible]	$49,500

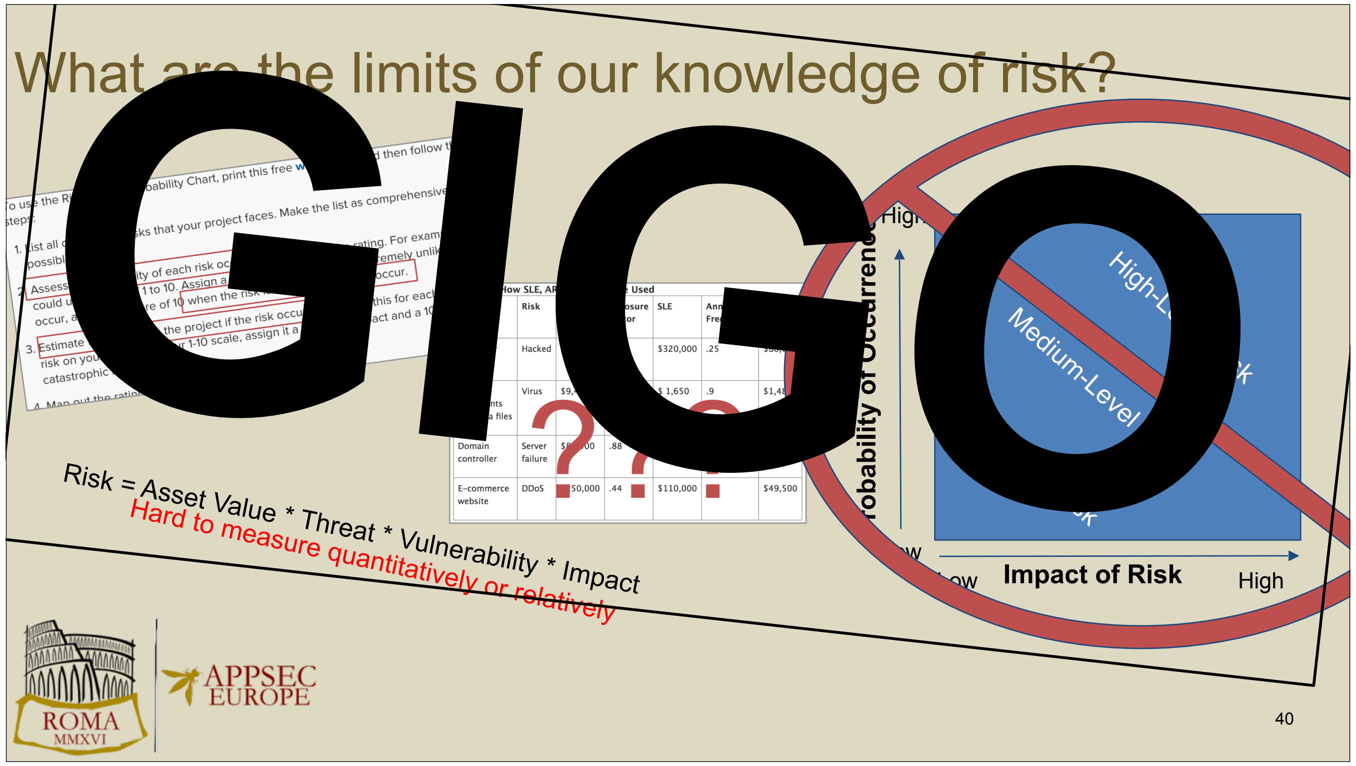
Risk = Asset Value * Threat * Vulnerability * Impact

Hard to measure quantitatively or relatively

How much can we really know about any of this?

Threat—A natural or man-made event that could have some type of negative impact on the organization.
Vulnerability—A flaw, loophole, oversight, or error that can be exploited to violate system security policy.

We can not possibly know enough to make use of tools like this in a meaningful way,



What are the limits of our knowledge of risk?

Risk = Asset Value * Threat * Vulnerability * Impact

Hard to measure quantitatively or relatively

How much can we really know about any of this?

Threat—A natural or man-made event that could have some type of negative impact on the organization.
Vulnerability—A flaw, loophole, oversight, or error that can be exploited to violate system security policy.

We can not possibly know enough to make use of tools like this.

We don’t really know what our scanner’s coverage are
Asset valuation, to whom, for what? Unless it’s a commodity item with a market price, GIGO
“Threat” is a singular word. How many real threats are single-factor? What if several threats coalesce in one event? What if several happen separately? GIGO
Vulnerability, Where do you draw the borders for your calculation? The app code? Framework? Libraries? OS? Platform? Other components? Firmware? CA certs? Rubber hose attack? Theoretical attack categories? How many of you are pentesters that have combined several low and moderate bugs in creative ways to hack something? More GIGO

The asset value of your customer database is “$432,000?” This is a totally meaningless statement, more GIGO

Just as true for Annualized Loss Expectancy, Single Loss Expectancy.

Qualitative Assessment?

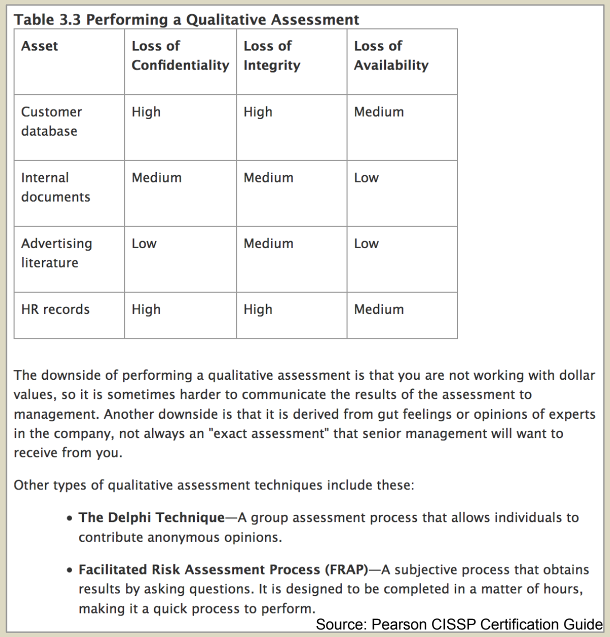


Table 3.3 Performing a Qualitative Assessment

Asset	Loss of Confidentiality	Loss of Integrity	Loss of Availability
Customer database	High	High	Medium
Internal documents	Medium	Medium	Low
Advertising literature	Low	Medium	Low
HR records	High	High	Medium

The downside of performing a qualitative assessment is that you are not working with dollar values, so it is sometimes harder to communicate the results of the assessment to management. Another downside is that it is derived from gut feelings or opinions of experts in the company, not always an "exact assessment" that senior management will want to receive from you.

Other types of qualitative assessment techniques include these:

- **The Delphi Technique**—A group assessment process that allows individuals to contribute anonymous opinions.
- **Facilitated Risk Assessment Process (FRAP)**—A subjective process that obtains results by asking questions. It is designed to be completed in a matter of hours, making it a quick process to perform.

Source: Pearson CISSP Certification Guide

Here's a screenshot from a CISSP certification guide that talks about qualitative assessment

Let's take a closer look:

Qualitative Assessment?

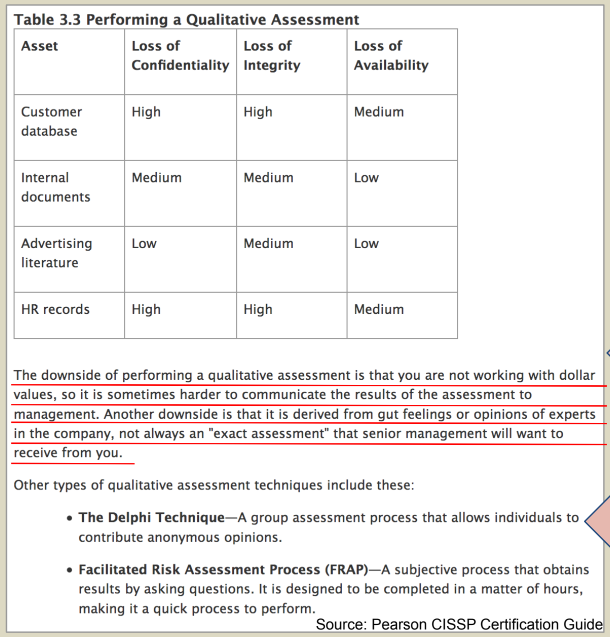


Table 3.3 Performing a Qualitative Assessment

Asset	Loss of Confidentiality	Loss of Integrity	Loss of Availability
Customer database	High	High	Medium
Internal documents	Medium	Medium	Low
Advertising literature	Low	Medium	Low
HR records	High	High	Medium

The downside of performing a qualitative assessment is that you are not working with dollar values, so it is sometimes harder to communicate the results of the assessment to management. Another downside is that it is derived from gut feelings or opinions of experts in the company, not always an "exact assessment" that senior management will want to receive from you.

Other types of qualitative assessment techniques include these:

- **The Delphi Technique**—A group assessment process that allows individuals to contribute anonymous opinions.
- **Facilitated Risk Assessment Process (FRAP)**—A subjective process that obtains results by asking questions. It is designed to be completed in a matter of hours, making it a quick process to perform.

Source: Pearson CISSP Certification Guide

Even though the quantitative values are just as if not more subjective...

Structurally aggregating subjective opinions until you reach consensus to measure risk as a source of truth? No.

The **downside** is that you’re not working with meaningless dollar values?

It’s derived from the opinions of experts so it’s not an “exact” assessment, but using dollar values that aren’t reflective of anything is, just because it’s a number?

The quantitative method is pure scientism, not pure science.

Delphi technique? Also based on the experiential knowledge of participants, a 1971 paper critical of Delphi described it as “dredging of half-formed ideas from the group memory.” It’s also been criticized as a way to shepherd a process to a pre-determined position. Not great if you are hoping for a source of truth.

Loss of integrity of internal documents, “Medium.” This isn’t really useful either. It doesn’t mean much and there are too many extraneous, unknown variables. “Loss of availability of internal documentation, “Low.” Totally ignores n-th order effects. Loss of availability of internal documentation in the table says Low, but what if it’s the docs are the runbook for the customer db, whose loss of availability is high? That’s different than if the internal docs that aren’t available are for the lunch-break table-tennis league.

Qualitative Assessment?

GIGO

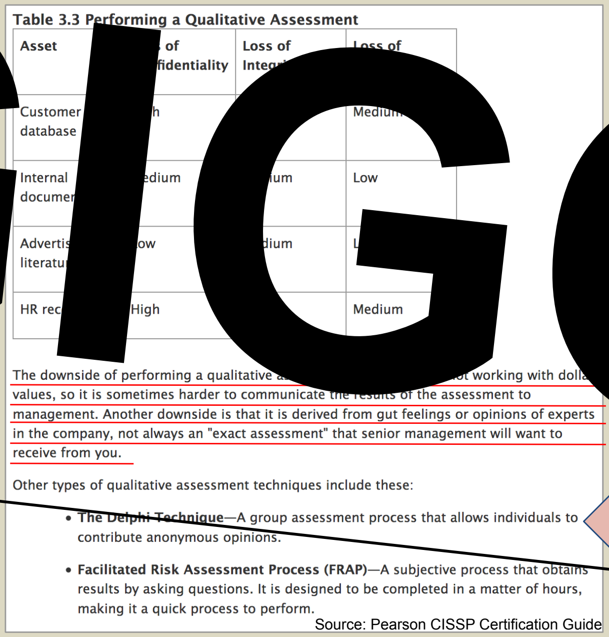


Table 3.3 Performing a Qualitative Assessment

Asset	of fidentiality	Loss of Integri	Loss of
Customer database	h		Medium
Internal documer	edium	ium	Low
Advertis literatur	ow	dium	L
HR rec	High		Medium

The downside of performing a qualitative a ot working with dolla values, so it is sometimes harder to communicate the results of the assessment to management. Another downside is that it is derived from gut feelings or opinions of experts in the company, not always an "exact assessment" that senior management will want to receive from you.

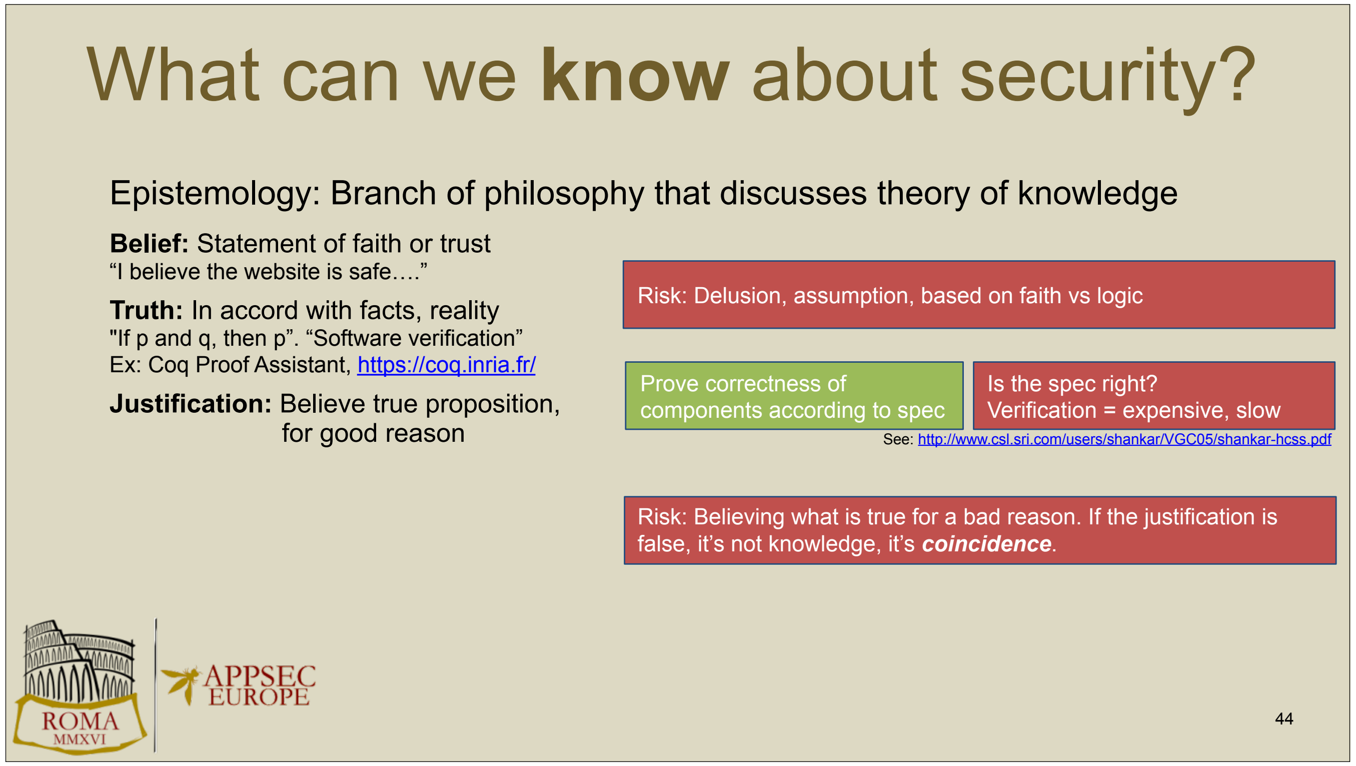
Other types of qualitative assessment techniques include these:

- **The Delphi Technique**—A group assessment process that allows individuals to contribute anonymous opinions.
- **Facilitated Risk Assessment Process (FRAP)**—A subjective process that obtains results by asking questions. It is designed to be completed in a matter of hours, making it a quick process to perform.

Source: Pearson CISSP Certification Guide

Even thoug quantitativ s are just ubjective...

Structurally aggregating subjective opinions until you reach consensus to measure risk as a source of truth? No.



What can we **know** about security?

Epistemology: Branch of philosophy that discusses theory of knowledge

Belief: Statement of faith or trust
"I believe the website is safe…."

Risk: Delusion, assumption, based on faith vs logic

Truth: In accord with facts, reality
"If p and q, then p". "Software verification"
Ex: Coq Proof Assistant, https://coq.inria.fr/

Prove correctness of components according to spec

Is the spec right?
Verification = expensive, slow

See: http://www.csl.sri.com/users/shankar/VGC05/shankar-hcss.pdf

Justification: Believe true proposition, for good reason

Risk: Believing what is true for a bad reason. If the justification is false, it's not knowledge, it's ***coincidence***.

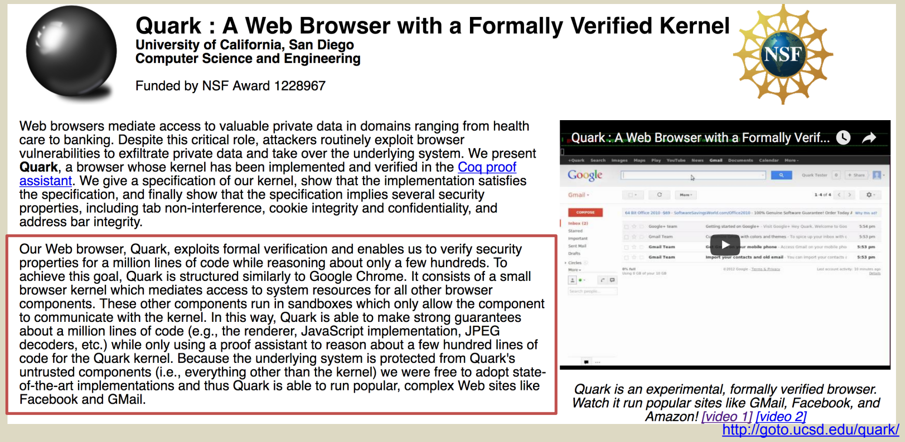
The main problem epistemology attempts to solve is understanding what the requirements for "knowledge" are.

It's really hard to make any true statement about security, other than that it's hard to make a true statement about it. We like to talk about "security verification" and "security assurance," but these terms are neither useful or accurate.
The tools we use most often do not verify anything at all. They find bugs, generally shallow ones.
Our scanners and pentesting don't prove code is safe if you find bugs and fix them, and they don't prove it's safe if you find no bugs.

How do you do you approach formal verification of a large software system like a browser or a web server? For starters, the specification has to be good. With HTML and HTTP for example the true specification is what servers and user agents support, the specs always trail the products. When a spec is introduced at the W3C, it's only meaningful if Google, Mozilla, Microsoft, and Apple implement it. But we could use it to ratchet up security for specific components that are most critical: the browser kernel and the TLS stack implementation for example.

Our programming languages were mostly not designed for security. I remember one of the members of our fuzzing team joking that browsers are a collection of use-after-free bugs that coincidentally happen to be able to render HTML.
That is why the Firefox product team is "Oxidizing" Firefox, that's our internal shorthand for "Replace components of Firefox with pieces of Servo, which is written in Rust, a programming language that started as the personal project of a Mozilla employee. Although its development is sponsored by Mozilla, it is an open community project.

Formally Verified Browser



Quark : A Web Browser with a Formally Verified Kernel

University of California, San Diego
Computer Science and Engineering

Funded by NSF Award 1228967

Web browsers mediate access to valuable private data in domains ranging from health care to banking. Despite this critical role, attackers routinely exploit browser vulnerabilities to exfiltrate private data and take over the underlying system. We present **Quark**, a browser whose kernel has been implemented and verified in the Coq proof assistant. We give a specification of our kernel, show that the implementation satisfies the specification, and finally show that the specification implies several security properties, including tab non-interference, cookie integrity and confidentiality, and address bar integrity.

Our Web browser, Quark, exploits formal verification and enables us to verify security properties for a million lines of code while reasoning about only a few hundreds. To achieve this goal, Quark is structured similarly to Google Chrome. It consists of a small browser kernel which mediates access to system resources for all other browser components. These other components run in sandboxes which only allow the component to communicate with the kernel. In this way, Quark is able to make strong guarantees about a million lines of code (e.g., the renderer, JavaScript implementation, JPEG decoders, etc.) while only using a proof assistant to reason about a few hundred lines of code for the Quark kernel. Because the underlying system is protected from Quark's untrusted components (i.e., everything other than the kernel) we were free to adopt state-of-the-art implementations and thus Quark is able to run popular, complex Web sites like Facebook and GMail.

Quark is an experimental, formally verified browser. Watch it run popular sites like GMail, Facebook, and Amazon! [video 1] [video 2]
http://goto.ucsd.edu/quark/

Protects the browser kernel

User still vulnerable to some of the most common types of attacks:

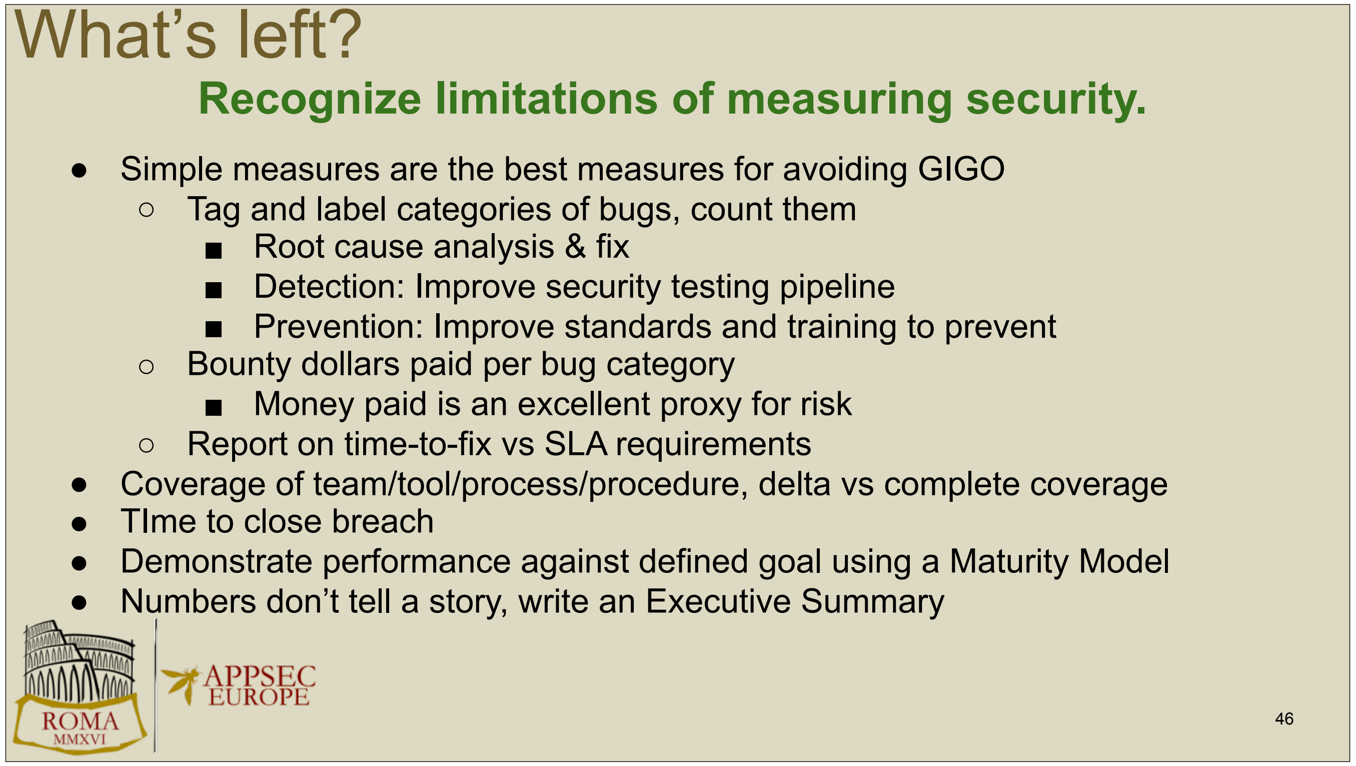
- XSS
- CSRF
- Session hijacking
- Redirects/forwards
- Server-side issues
- Network attacks
- Protocol spec problems

Quark is really cool, a formally verified browser using the Coq interactive theorem prover.

A million lines of code have verified security properties -renderer, jpeg decoders, javascript implementation.

The rest is sandboxed, and yet it’s still vulnerable to lots of protocol related issues.

Like Mozilla’s Rust/oxidation/Electrolysis efforts, it’s another way to to ratchet up security and reduce the attackable surface. But still hard to quantify using the kinds of metrics that people are often hoping to see.



What’s left?

Recognize limitations of measuring security.

- Simple measures are the best measures for avoiding GIGO
  - Tag and label categories of bugs, count them
    - Root cause analysis & fix
    - Detection: Improve security testing pipeline
    - Prevention: Improve standards and training to prevent
  - Bounty dollars paid per bug category
    - Money paid is an excellent proxy for risk
  - Report on time-to-fix vs SLA requirements
- Coverage of team/tool/process/procedure, delta vs complete coverage
- TIme to close breach
- Demonstrate performance against defined goal using a Maturity Model
- Numbers don’t tell a story, write an Executive Summary

ROMA MMXVI APPSEC EUROPE

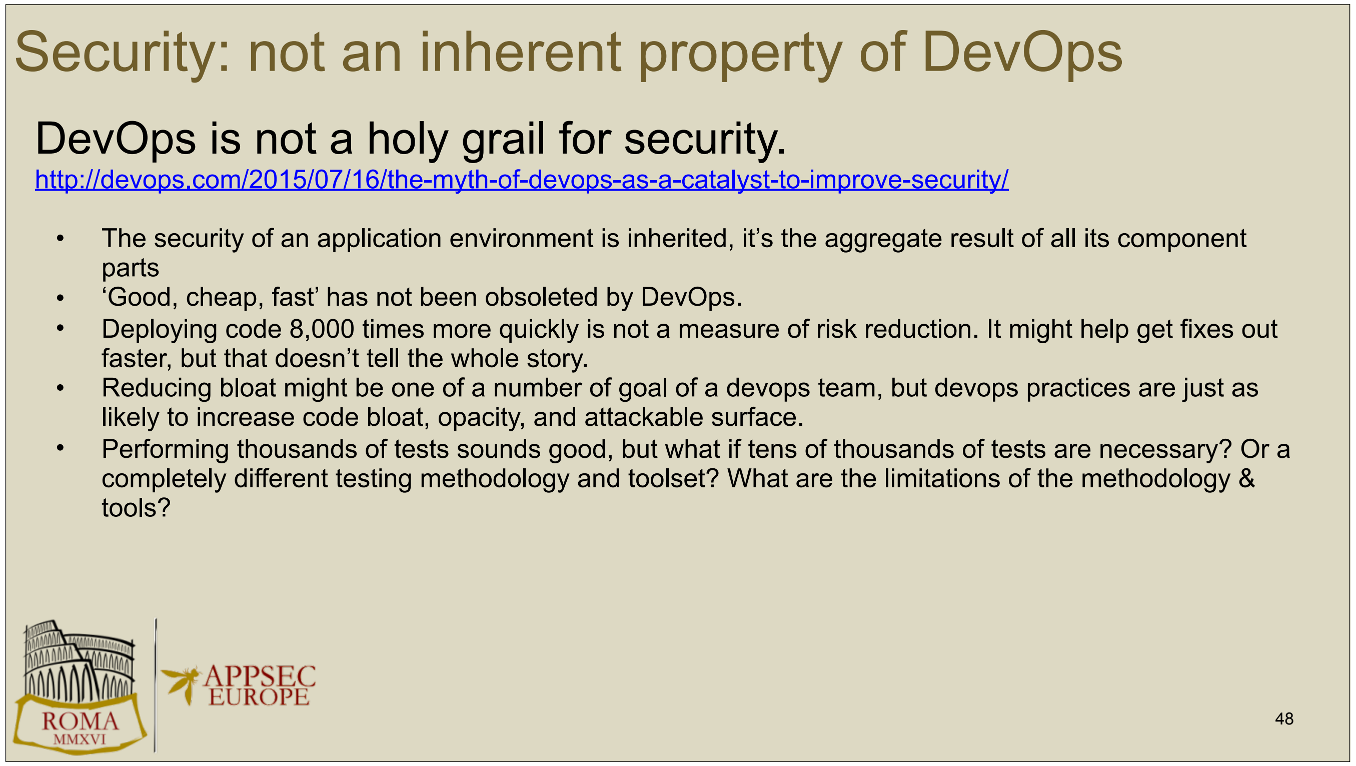
We discussed what’s hard to measure and quantify. What’s left?

Even these can get tricky - what is complete coverage?

Time to close breach - can you really have any knowledge about the amount of certainty of whether the breach is closed?

This is why the executive summary is so important - language is a much richer way of explaining these limitations than a chart or graph.

Using OpenSAMM in a DevOps organization



Security: not an inherent property of DevOps

DevOps is not a holy grail for security.

http://devops.com/2015/07/16/the-myth-of-devops-as-a-catalyst-to-improve-security/

- The security of an application environment is inherited, it’s the aggregate result of all its component parts
- ‘Good, cheap, fast’ has not been obsoleted by DevOps.
- Deploying code 8,000 times more quickly is not a measure of risk reduction. It might help get fixes out faster, but that doesn’t tell the whole story.
- Reducing bloat might be one of a number of goal of a devops team, but devops practices are just as likely to increase code bloat, opacity, and attackable surface.
- Performing thousands of tests sounds good, but what if tens of thousands of tests are necessary? Or a completely different testing methodology and toolset? What are the limitations of the methodology & tools?

There are many categories of attacks that automated tests are not able to identify:

Logic flaws are one.

Different components of a system understanding the same piece of (malicious or spurious) data to mean different things, another.

Sometimes the problem isn’t a bug, it’s architectural deficiency. If it’s deeply layered inside a component of your system that uses its own non standard build system, you probably won’t find it. Software tests are for known, expected code execution paths and interactions, it’s much harder to identify all possible execution paths and orders of operations.

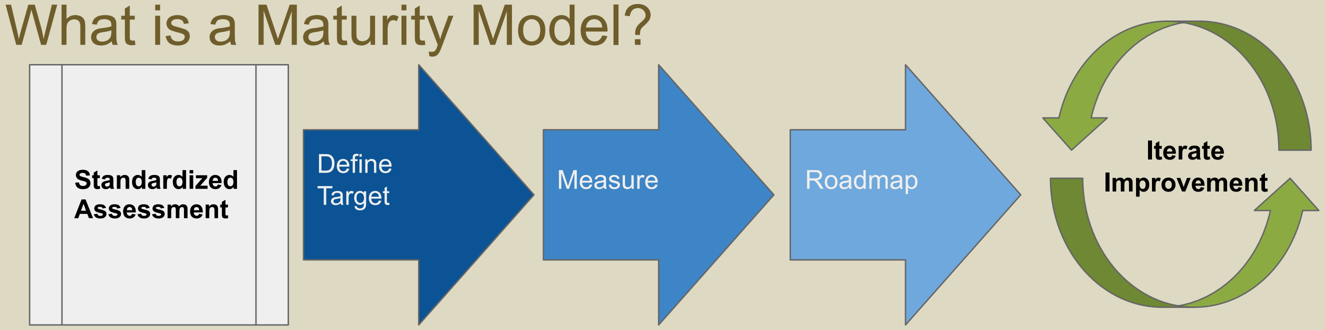
Thinking security testing through and automating as much as possible will yield results, but that can happen with or without devops. I’m not saying devops is invalid, rather that it alone is not responsible for good outcomes.

Thinking that an approach delivers more than it really does is only a false sense of security, arguably worse than awareness of insufficient security.

IT is a business support function. Security is a business risk analysis function. If you standardize and integrate things without understanding threat, risk and security posture, what have you done?

Ultimately the decisions are business decisions. Unfortunately, they are frequently made from the perspective of insufficient knowledge.

What is a Maturity Model?



Standardized Assessment

Define Target

Measure

Roadmap

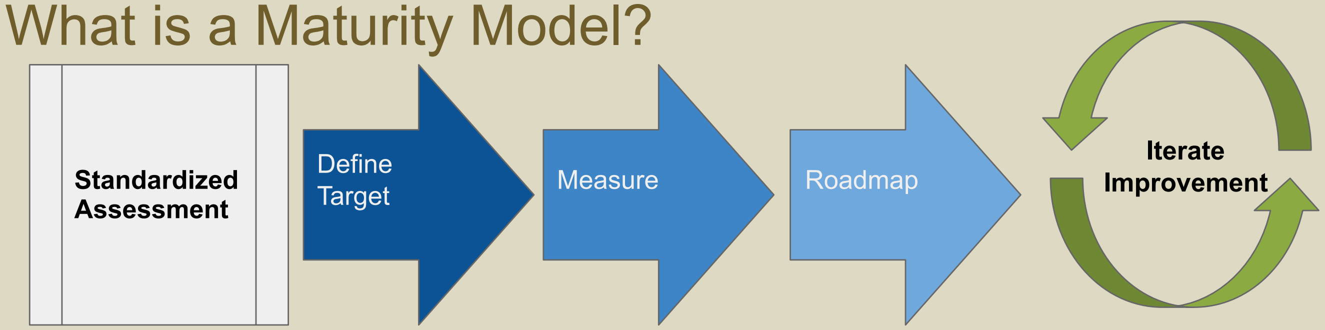
Iterate Improvement

Maturity: relates to the degree of formality and optimization of processes, ad-hoc practices, formally defined steps, and result metrics.

Used to reach active optimization of the processes being measured.

Standard, can be used to compare between organizations…

What is a Maturity Model?



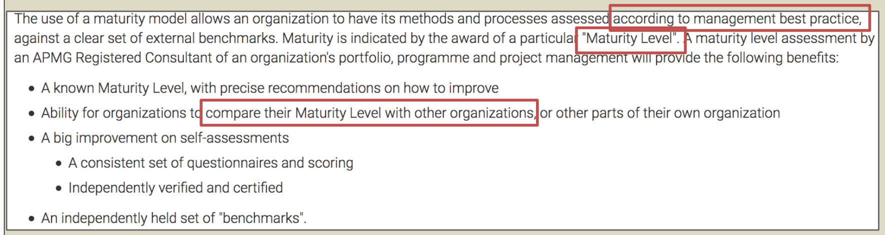
Maturity: relates to the degree of formality and optimization of processes, ad-hoc practices, formally defined steps, and result metrics.

Used to reach active optimization of the processes being measured.

Standard, can be used to compare between organizations… Or can it?

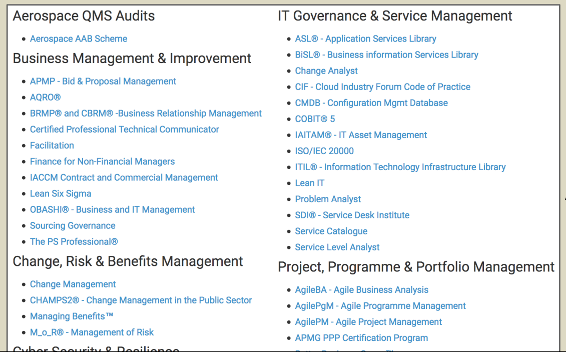
Maturity Models

APMG International
Accrediting Professionals



The use of a maturity model allows an organization to have its methods and processes assessed according to management best practice, against a clear set of external benchmarks. Maturity is indicated by the award of a particular "Maturity Level". A maturity level assessment by an APMG Registered Consultant of an organization's portfolio, programme and project management will provide the following benefits:

- A known Maturity Level, with precise recommendations on how to improve
- Ability for organizations to compare their Maturity Level with other organizations, or other parts of their own organization
- A big improvement on self-assessments
  - A consistent set of questionnaires and scoring
  - Independently verified and certified
- An independently held set of "benchmarks".



Aerospace QMS Audits

- Aerospace AAB Scheme

Business Management & Improvement

- APMP - Bid & Proposal Management
- AQRO®
- BRMP® and CBRM® -Business Relationship Management
- Certified Professional Technical Communicator
- Facilitation
- Finance for Non-Financial Managers
- IACCM Contract and Commercial Management
- Lean Six Sigma
- OBASHI® - Business and IT Management
- Sourcing Governance
- The PS Professional®

Change, Risk & Benefits Management

- Change Management
- CHAMPS2® - Change Management in the Public Sector
- Managing Benefits™
- M_o_R® - Management of Risk

Cyber Security & Resilience

IT Governance & Service Management

- ASL® - Application Services Library
- BiSL® - Business information Services Library
- Change Analyst
- CIF - Cloud Industry Forum Code of Practice
- CMDB - Configuration Mgmt Database
- COBIT® 5
- IAITAM® - IT Asset Management
- ISO/IEC 20000
- ITIL® - Information Technology Infrastructure Library
- Lean IT
- Problem Analyst
- SDI® - Service Desk Institute
- Service Catalogue
- Service Level Analyst

Project, Programme & Portfolio Management

- AgileBA - Agile Business Analysis
- AgilePgM - Agile Programme Management
- AgilePM - Agile Project Management
- APMG PPP Certification Program

ROMA
MMXVI
APPSEC
EUROPE

“Best” practice?

- Subjective, not objective.
- There is no “best practice” for all organizations and problem spaces.

Maturity level

- Just because something has a dependency on something else isn’t proof it provides more “maturity”

Cross-org comparison

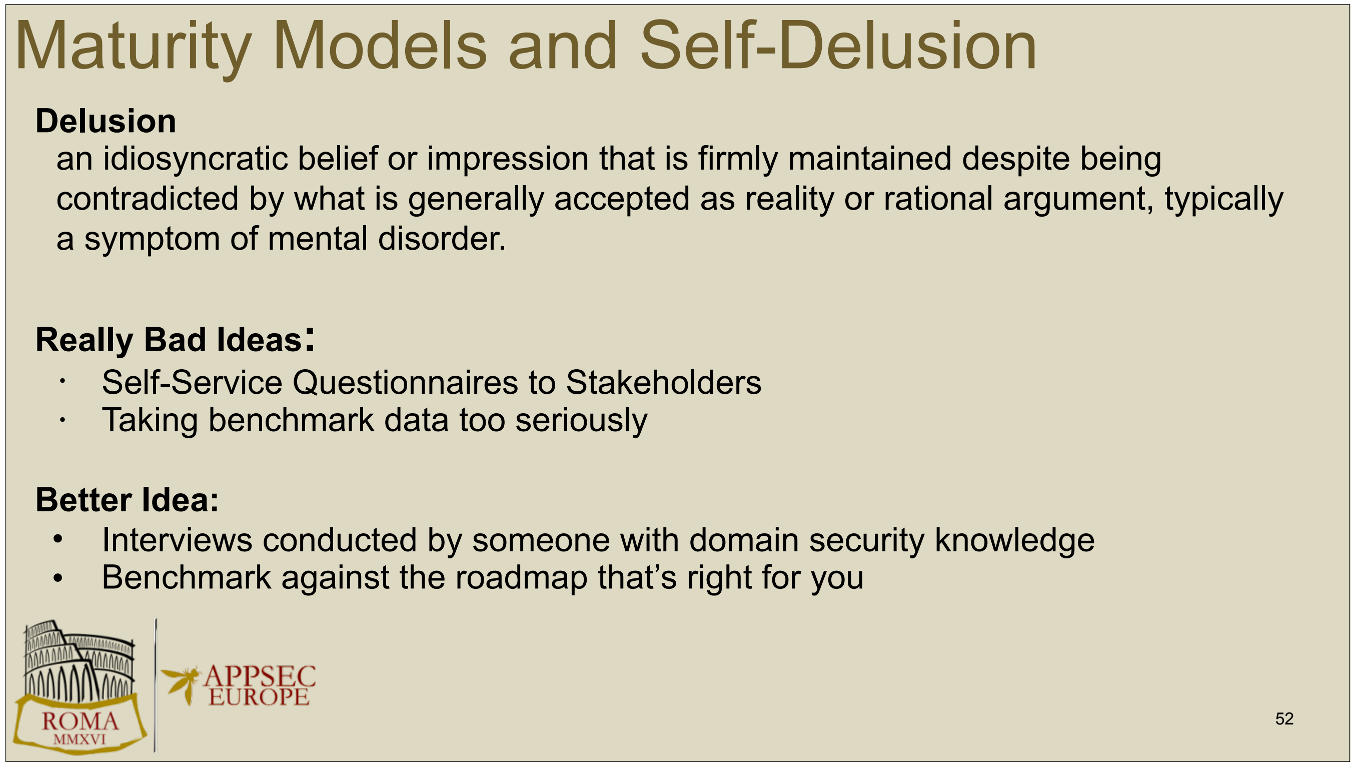
- Value is subjective, different orgs have different goals
- Emotional appeal, pop-psych
- Assumes other orgs being compared to have accurate self knowledge

Argument from Authority fallacy

- “Best practice” determination is not repeatable or falsifiable

Maturity models are another way that security is often measured.

This isn’t to say that Maturity Models are useless. I use and advocate OpenSAMM. You have to use your brain though and not take it as gospel.



Maturity Models and Self-Delusion

Delusion

an idiosyncratic belief or impression that is firmly maintained despite being contradicted by what is generally accepted as reality or rational argument, typically a symptom of mental disorder.

Really Bad Ideas:

- Self-Service Questionnaires to Stakeholders
- Taking benchmark data too seriously

Better Idea:

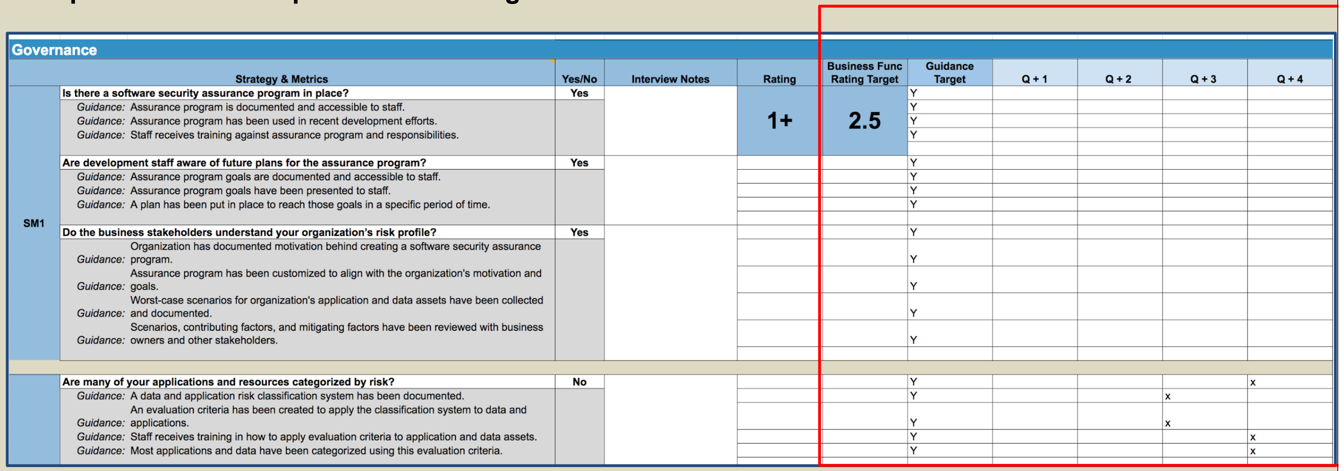
- Interviews conducted by someone with domain security knowledge
- Benchmark against the roadmap that’s right for you

Self service questionnaires - a terrible idea.
Interviews work the best, when done by a skeptical security curmudgeon they nearly always uncover things that the person being interviewed would not have considered

Benchmark data problems:
Aggregate information
Doesn’t necessarily apply to your organization or problems
Cognitive errors: Misperception (deficiency in knowledge of the present, overestimating competence), Misremembering (Knowledge of the present colors information remembered from the past), Impact bias (The tendency to overestimate expected future states)
Humans are not reliable observers, we tend to see what we want to see.
The quality and applicability of the measurements of other in different situations are a tempting comparison, but aren’t especially useful since the Margin of Error is unknowable.

OpenSAMM Interview Spreadsheet

Proposed Interview Spreadsheet Changes



Governance

	Strategy & Metrics	Yes/No	Interview Notes	Rating	Business Func Rating Target	Guidance Target	Q + 1	Q + 2	Q + 3	Q + 4
SM1	**Is there a software security assurance program in place?**	Yes		1+	2.5	Y				
	Guidance: Assurance program is documented and accessible to staff.					Y				
	Guidance: Assurance program has been used in recent development efforts.					Y				
	Guidance: Staff receives training against assurance program and responsibilities.					Y				
	Are development staff aware of future plans for the assurance program?	Yes				Y				
	Guidance: Assurance program goals are documented and accessible to staff.					Y				
	Guidance: Assurance program goals have been presented to staff.					Y				
	Guidance: A plan has been put in place to reach those goals in a specific period of time.					Y				
	Do the business stakeholders understand your organization's risk profile?	Yes				Y				
	Guidance: Organization has documented motivation behind creating a software security assurance program.					Y				
	Guidance: Assurance program has been customized to align with the organization's motivation and goals.					Y				
	Guidance: Worst-case scenarios for organization's application and data assets have been collected and documented.					Y				
	Guidance: Scenarios, contributing factors, and mitigating factors have been reviewed with business owners and other stakeholders.					Y				
	Are many of your applications and resources categorized by risk?	No				Y				x
	Guidance: A data and application risk classification system has been documented.					Y			x	
	Guidance: An evaluation criteria has been created to apply the classification system to data and applications.					Y			x	
	Guidance: Staff receives training in how to apply evaluation criteria to application and data assets.					Y				x
	Guidance: Most applications and data have been categorized using this evaluation criteria.					Y				x

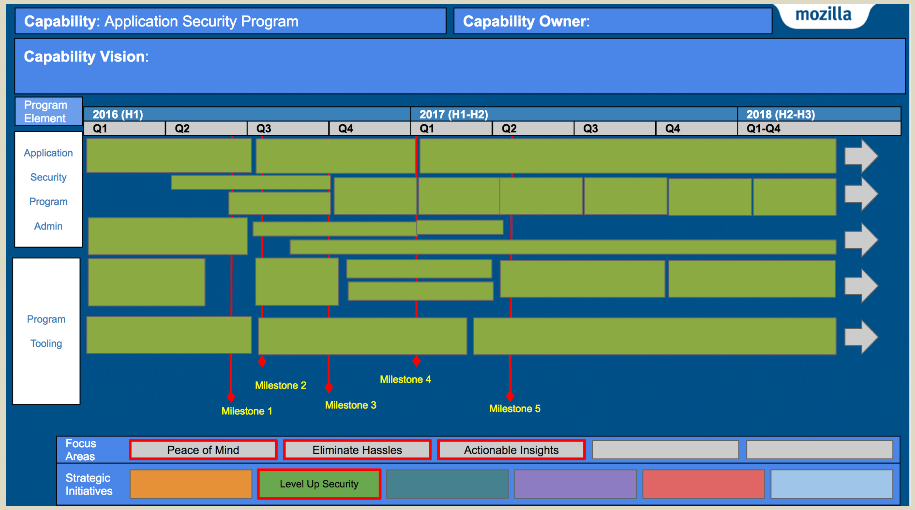
https://docs.google.com/spreadsheets/d/1mJ4XuDGKbT5brwqWzk31xS8NjniYQmo7R_flW1waNn4

ROMA MMXVI APPSEC EUROPE

One thing that OpenSAMM has lacked is a tool for developing your own roadmap.

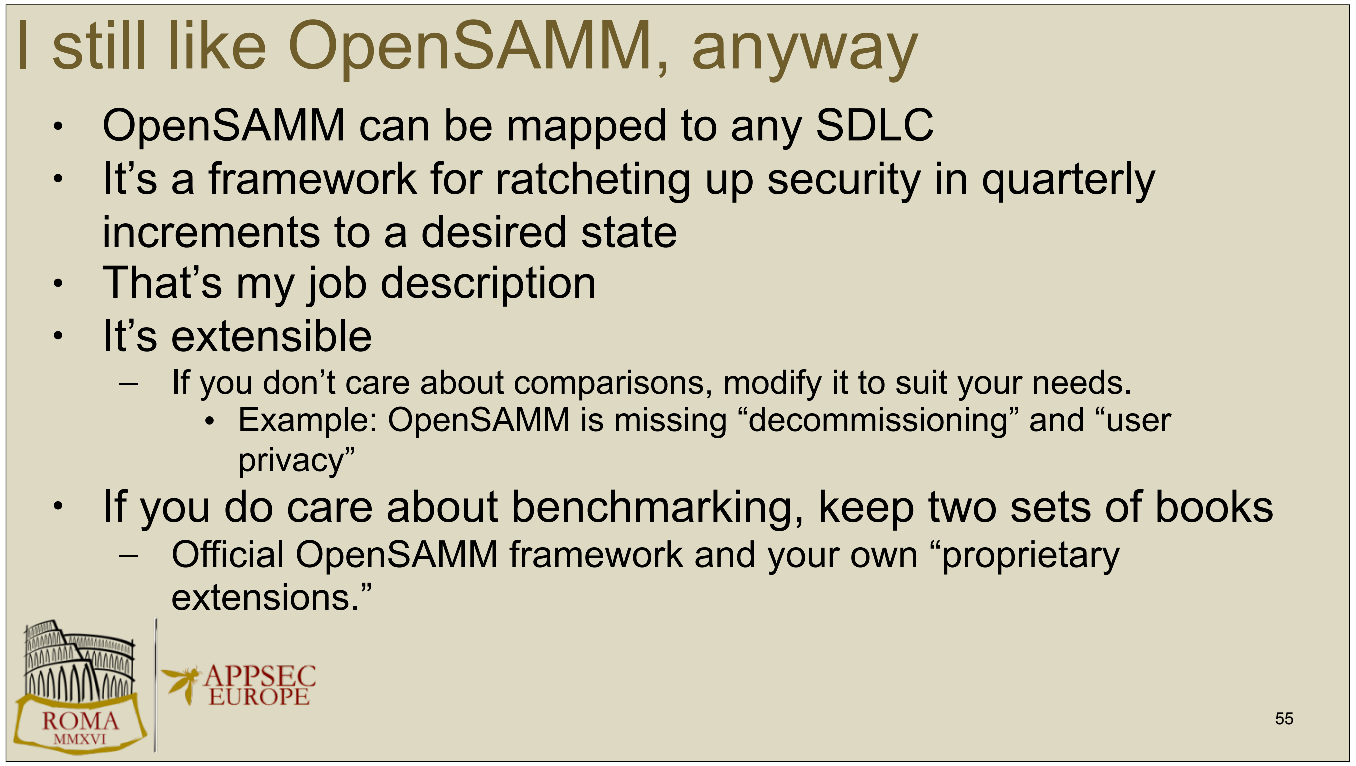
A few weeks ago I contributed this, it will hopefully make it into 1.2, either way you're welcome to use it

Security Program Roadmap



This is a sanitized sample of an alternative way to present an appsec program roadmap.

You could just was easily make the left column (white boxes) show “governance, construction, verification, operations” and use to tell a story for your OpenSAMM roadmap



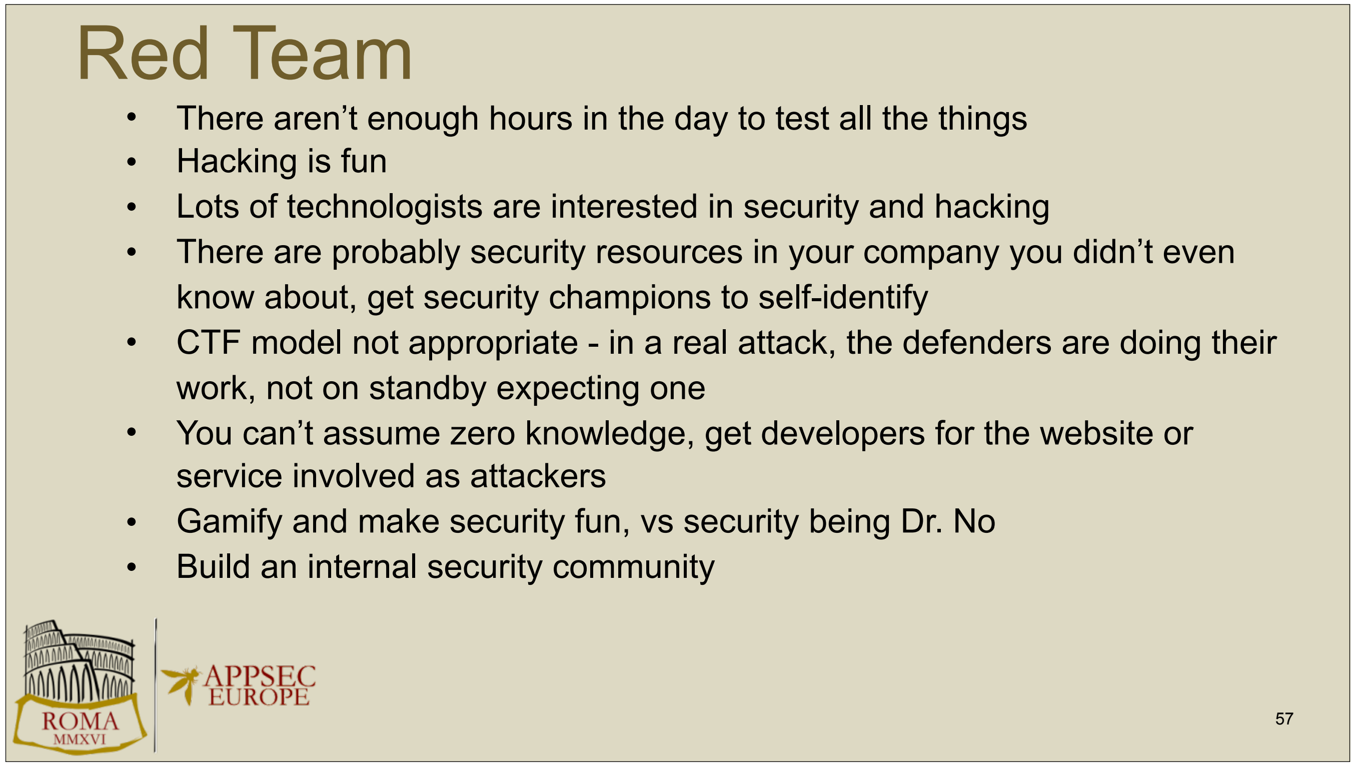
I still like OpenSAMM, anyway

- OpenSAMM can be mapped to any SDLC
- It’s a framework for ratcheting up security in quarterly increments to a desired state
- That’s my job description
- It’s extensible
  - If you don’t care about comparisons, modify it to suit your needs.
    - Example: OpenSAMM is missing “decommissioning” and “user privacy”
- If you do care about benchmarking, keep two sets of books
  - Official OpenSAMM framework and your own “proprietary extensions.”

Self service questionnaires - a terrible idea.
Interviews work the best, when done by a skeptical security curmudgeon they nearly always uncover things that the person being interviewed would not have considered

Benchmark data problems:
- Aggregate information
- Doesn’t necessarily apply to your organization or problems
- Cognitive errors: Misperception (deficiency in knowledge of the present, overestimating competence)
- Misremembering (Knowledge of the present colors information remembered from the past), Impact bias (The tendency to overestimate expected future states)
- Humans are not reliable observers, we tend to see what we want to see.
- The quality and applicability of the measurements of other in different situations are a tempting comparison, but aren’t especially useful since the Margin of Error is unknowable.

Get non-security engineers
to help pentest
by setting up a Red Team



Red Team

- There aren’t enough hours in the day to test all the things
- Hacking is fun
- Lots of technologists are interested in security and hacking
- There are probably security resources in your company you didn’t even know about, get security champions to self-identify
- CTF model not appropriate - in a real attack, the defenders are doing their work, not on standby expecting one
- You can’t assume zero knowledge, get developers for the website or service involved as attackers
- Gamify and make security fun, vs security being Dr. No
- Build an internal security community

ROMA MMXVI | APPSEC EUROPE

Get more stuff tested, more deeply

Get security champions to self-identify

Build a security community

best as monthly or bi-monthly activity for 3-4 hours, get exec approval and participants should have approval of their manager for their time.

Summary

Summary

- Radical open sharing of documentation: less scary than it sounds
- Security does not easily yield to quantitative measurement
  - It's easy to spend a lot of time generating metrics that don't inform, don't do that
  - Numbers don't tell a story, they are open to interpretation. So, tell a story.
- Bug bounty program + Maturity Model + organizational threat model to guide your Appsec program
- Create an internal cross-organizational Red Team to build an internal security community

ROMA MMXVI | APPSEC EUROPE

Thank you

Adam Muntner
amuntner@mozilla.com