Ca' Foscari
University
of Venice

Standard and metrics in software quality assurance

Pietro Ferrara

pietro.ferrara@unive.it

sw bug

WIRED

LILY HAY NEWMAN SECURITY 12.31.17 07:00 AM

THE WORST HACKS OF 2017

Equifax

This was really bad. The credit monitoring firm Equifax disclosed a massive breach at the beginning of September, which exposed personal information for 145.5 million people. The data included birth dates, addresses, some driver's license numbers, about 209,000 credit card numbers, and Social Security numbers—meaning that almost half the US population potentially had their crucial secret identifier exposed. Because the information Equifax coughed up was so sensitive, it's widely considered the worst corporate data breach ever. For now.

2012
Bug introduced in open source software Apache Struts

2012, 2015, 2016
Several versions released

Between
Several versions released

March 7, 2017
Vulnerability discovered

May 13, 2017
Hackers access Equifax files

Find during development

Find at release

Find at release

Find when acquiring third-party software

Ca' Foscari University of Venice

Nearly 157,000 had data breached in TalkTalk cyber-attack

BBC NEWS — Sign in

Friday 6 November 2015 08.32 GMT

BUSINESS INSIDER UK

The TalkTalk hack cost it £42 million

May 12, 2016, 8:33 AM

TalkTalk unveiled the full impact of the cyber attack it suffered last year — 95,000 customers left in the immediate aftermath and the cleanup cost hit £42 million ($60.53 million).

theguardian

TalkTalk hit with record £400k fine over cyber-attack

Wednesday 5 October 2016 14.00 BST

"The technique used by the attacker, called **SQL injection**, has been well known in security circles for almost 20 years. "SQL injection is well understood, defences exist and TalkTalk ought to have known it posed a risk to its data," the Information Commissioner's Office said."

What is SQL injection?

«permits the attacker to modify existing data, obtain system data, eliminate data or make it inaccessible»

Ca' Foscari University of Venice

eBay under pressure as hacks continue

BBC NEWS | Sign in

22 September 2014 | Technology

What is XSS?

«Cross-Site Scripting is a security vulnerability permitting the hacker to insert content on web pages, without the owner knowing it, and divert the user elsewhere»

BBC Sign in | News | Sport | Weather | Shop | Earth | Travel | Mc

NEWS

Home | Video | World | UK | Business | Tech | Science | Magazine | Entertainment & Arts

24 July 2015 | Technology

Fiat Chrysler recalls 1.4 million cars after Jeep hack

security affairs

Security vulnerabilities in the Hyundai Blue Link mobile apps allowed hackers to steal vehicles, the car maker fixed them.

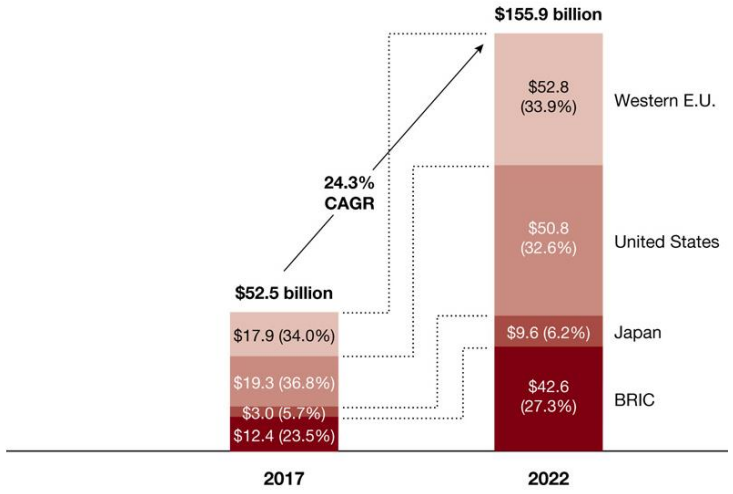
April 26, 2017 I

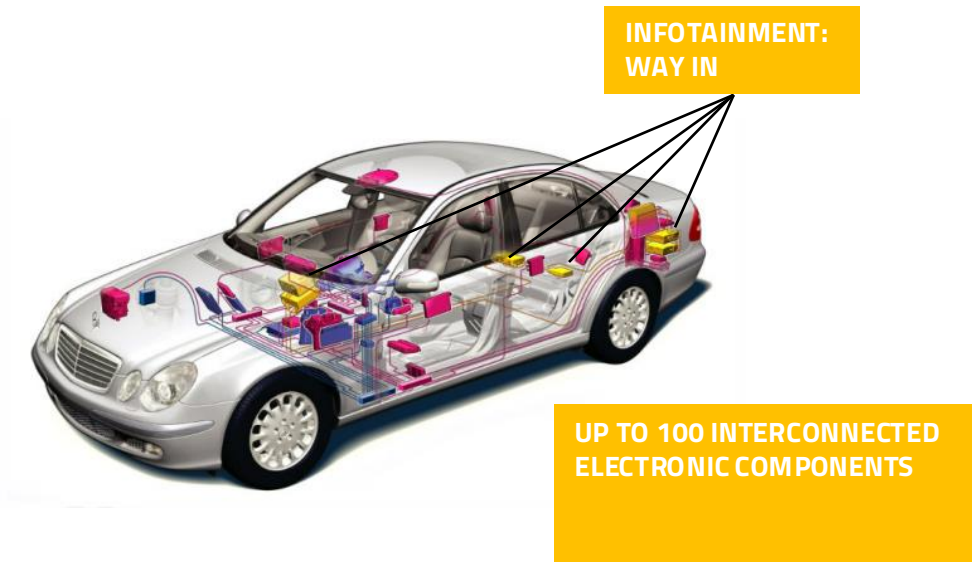
WIRED

ANDY GREENBERG SECURITY 02.16.17 5:30 PM

ANDROID PHONE HACKS COULD UNLOCK MILLIONS OF CARS

Exhibit 7
Connected car revenue potential, by region, 2017–22





Ca' Foscari University of Venice

Ariane 5 (June, 4th 1996)

https://www.youtube.com/watch?v=gp_D8r-2hwk

sw bug

cause a damage

Takeout:

1. Software contains bugs
2. Bugs might cause a damage

That means:

1. There is no shame on writing bugged software
2. There is shame in not taking all countermeasures to avoid bugs!

Program/Code analysis

We will focus on
- static
- automatic

program analyzers (SPA)

Main questions:
1. Who should use SPA and when?
2. Why using SPA?
3. Which SPA? How to evaluate?

Program analysis is the process of automatically analyzing the behavior of computer programs regarding a property such as correctness, robustness, safety and liveness

Program analysis can be performed without executing the program (static program analysis), during runtime (dynamic program analysis) or in a combination of both.

Ca' Foscari University of Venice

Static Program Analyzers (SPA) Input and output

Properties
- SQL Injection
- NullPointerExceptions
- Privacy leaks
- ...

Software

SPA

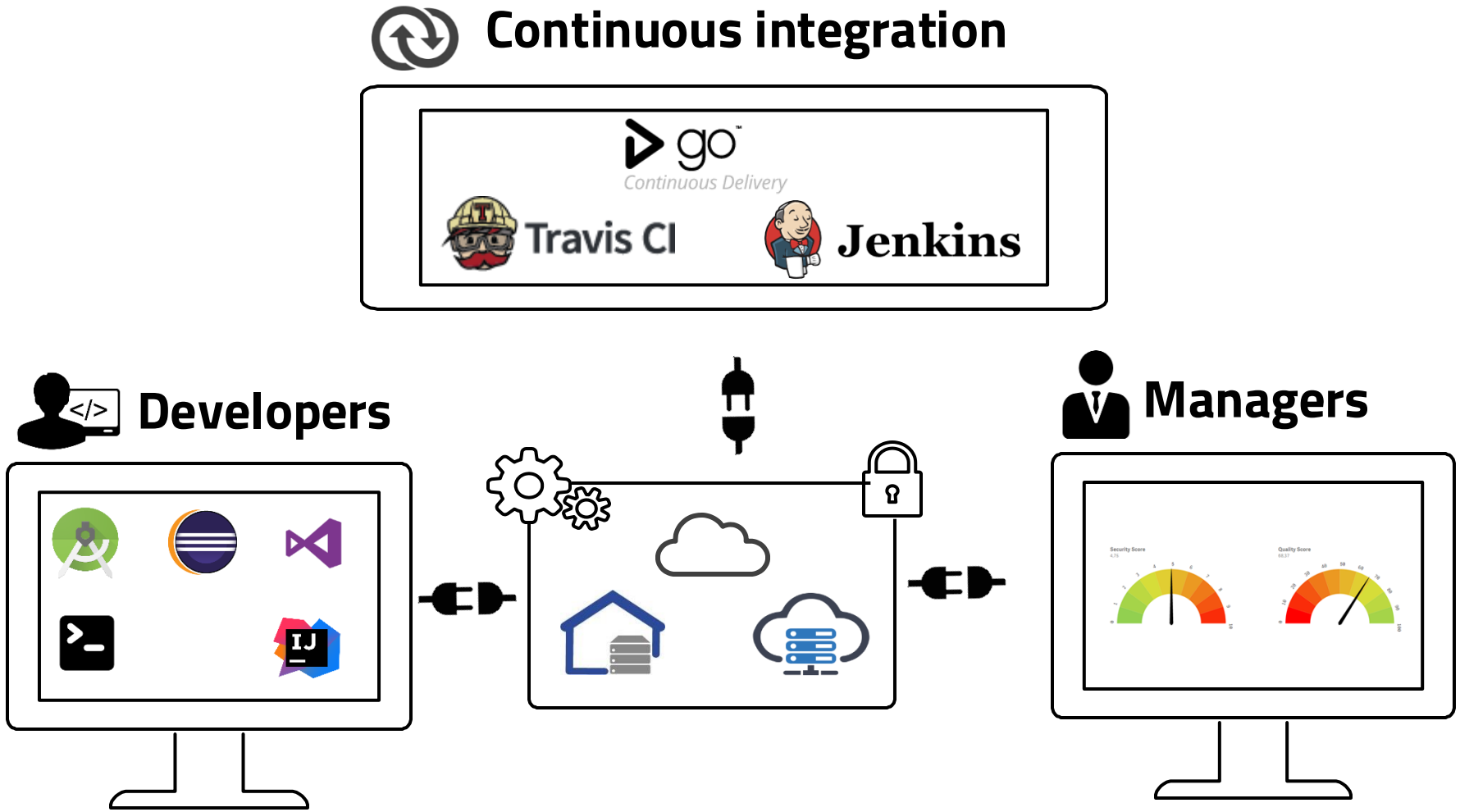
[Injection] possible SQL-injection

[Injection] possible XSS-injection

[BasicNullness] is the receiver of "getProperty" non-null?

Ca' Foscari University of Venice

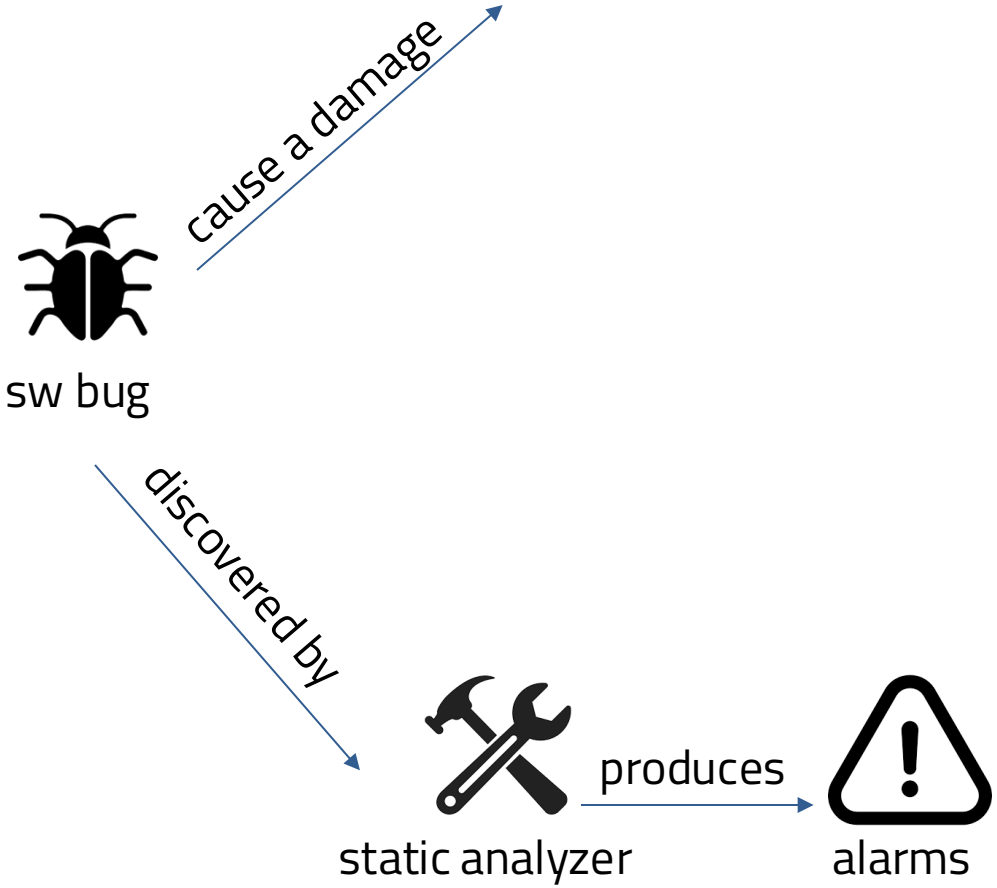
Who
Developers, project managers, CTOs



SPAs' features

- Syntactic vs. semantic
- Source code vs. bytecode
- Sound vs. unsound
- Evaluation
- Difficult to compare them

- Syntactic: efficient, rough
- Semantic: slow, precise
- Bytecode: analyze 3rd party software, need to compile the whole system
- Sound: identify all possible (and more) bugs, security vulnerabilities, privacy leaks
- Various proposals:
  - OWASP Benchmark
  - NIST SAMATE
  - Static Analysis Technologies Evaluation Criteria


cause a damage
sw bug
discovered by
produces
static analyzer
alarms

Ca' Foscari University of Venice

Mitre Corporation

https://www.mitre.org/

The MITRE Corporation's mission-driven team is dedicated to solving problems for a safer world. We are a ***not-for-profit company*** *that operates multiple* ***federally funded research and development centers****.*

- Born in 1958 as a private, not-for-profit company
- Operate federally funded research and development centers:
  - Scientific research and analysis
  - Development and acquisition
  - Systems engineering and integration
- Independent research program
  - explores new and expanded uses of technologies

Ca' Foscari University of Venice

Common Vulnerabilities and Exposures

http://cve.mitre.org/

- Dictionary of publicly disclosed cybersecurity vulnerabilities and exposures
  - free to search, use, and incorporate into products and services, per the terms of use.
- Feeds the U.S. National Vulnerability Database (NVD)
- CVE is:
  - One identifier for one vulnerability or exposure
  - One standardized description for each vulnerability or exposure
  - How disparate databases and tools can "speak" the same language
  - A basis for evaluation among services, tools, and databases

Vulnerability definition by CVE

"A **weakness** in the computational logic (e.g., code) found in **software and hardware** components that, when exploited, results in a **negative** impact to **confidentiality**, **integrity**, or **availability**. Mitigation of the vulnerabilities in this context typically involves **coding changes**, but could also include **specification changes** or even **specification deprecations** (e.g., removal of affected protocols or functionality in their entirety)."

Ca' Foscari University of Venice

National Institute of Standards and Technology

https://www.nist.gov/

*To **promote U.S. innovation and industrial competitiveness** by advancing **measurement** science, **standards**, and **technology** in ways that enhance economic security and improve our quality of life.*

- Measurement standards laboratory
- Non-regulatory agency
- United States Department of Commerce
- Formed on March, 3rd 1901 (!)
- 1901-1988: National Bureau of Standards
  - provide standard weights and measures
  - national physical laboratory for the United States

Ca' Foscari University of Venice

NIST National Vulnerability DB

https://nvd.nist.gov/

- U.S. government repository of standards based vulnerability management data

Time Period	New CVEs Received by NVD	New CVEs Analyzed by NVD	Modified CVEs Received by NVD	Modified CVEs Re-analyzed by NVD
Today	63	1	0	0
This Week	167	98	52	1
This Month	238	225	158	1
Last Month	1371	1065	1030	6
This Year	2874	2363	4173	35

CVSS V3 Score Distribution

Severity	Number of Vulns
CRITICAL	3550
HIGH	10200
MEDIUM	8764
LOW	445

CVSS

https://www.first.org/cvss/

The Common Vulnerability Scoring System (CVSS) provides a way to capture the principal characteristics of a vulnerability and produce a **numerical score reflecting its severity**. The numerical score can then be translated into a **qualitative representation** (such as low, medium, high, and critical) to help organizations properly **assess and prioritize their vulnerability management processes**.

- Free and open industry standard for assessing the severity of computer system security vulnerabilities
- Maintained by FIRST

FIRST is an **international confederation** of trusted **computer incident response teams** who cooperatively handle computer security incidents and promote incident prevention programs.

Ca' Foscari University of Venice

Common Weakness Enumeration

https://cwe.mitre.org/

- The challenge:
  - Software acquirers want assurance that software products are reviewed for known types of security flaws
  - There are no nomenclature, taxonomies, or standards to define the capabilities and coverage of these tools
- The solution
  - A list of common software security weaknesses
  - It encompasses a large portion of the CVE List
- MITRE began working on the issue of categorizing software weaknesses as early 1999 when it launched the CVE List

Ca' Foscari University of Venice

Common Weakness Enumeration
The lingua franca

CWE is a:

- list of software weakness types
- common language for describing software security weaknesses
- measuring stick for software security tools
- common baseline standard for weakness identification

Each weakness type is identified by a unique number

CWE 89

Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')

CWE 79

Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')

[Almost] all existing SPAs are CWE compatible
http://cwe.mitre.org/compatible

Log4J vulnerability

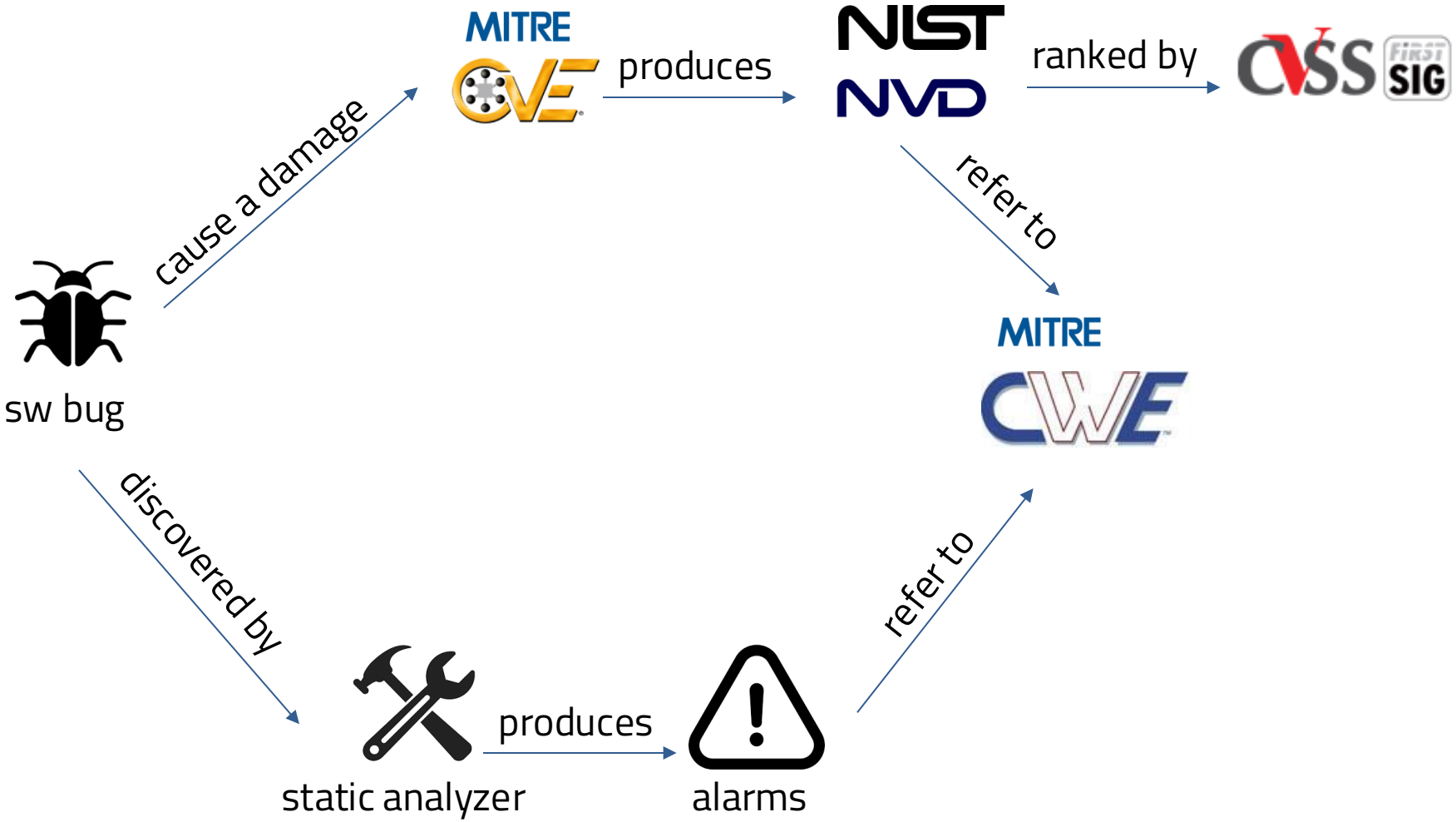
- https://logging.apache.org/log4j/2.x/
  - Any software MUST do logging!
- But then a security vulnerability there might have some impact
  - https://en.wikipedia.org/wiki/Log4Shell
  - https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2021-44228
  - https://nvd.nist.gov/vuln/detail/CVE-2021-44228

*The exploit allows hackers to gain **control of vulnerable devices** using Java (…) In the days following the vulnerability's disclosure, Check Point observed millions of attacks being initiated by hackers, with some researchers observing a rate of over one hundred attacks per minute that ultimately resulted with attempted attacks on over 40% of business networks internationally.*

Ca' Foscari University of Venice

Heartbleed bug
A notable example

- http://heartbleed.com/
- Discovered on April 1st/3rd 2014 by Google and Codenomicon
  - https://en.wikipedia.org/wiki/Codenomicon
- Fixed on April 7th
- Quite some noise in the news
- Few exploits:
  - April 8th: steal of SSN of 900 Canadian tax payers
  - Hijacking of account of https://www.mumsnet.com/
  - Steal of 4.5 million patient records of U.S. hospitals
- https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2014-0160


MITRE
CVE
produces
NIST
NVD
ranked by
CVSS
FIRST
SIG
cause a damage
refer to
MITRE
CWE
sw bug
discovered by
produces
refer to
static analyzer
alarms

Ca' Foscari University of Venice

CWE/SANS TOP 25 Most Dangerous SW Errors

http://cwe.mitre.org/top25

Rank	ID	Name	Score	CVEs in KEV	Rank Change vs. 2022
1	CWE-787	Out-of-bounds Write	63.72	70	0
2	CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	45.54	4	0
3	CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	34.27	6	0
4	CWE-416	Use After Free	16.71	44	+3
5	CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')	15.65	23	+1
6	CWE-20	Improper Input Validation	15.50	35	-2
7	CWE-125	Out-of-bounds Read	14.60	2	-2
8	CWE-22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')	14.11	16	0
9	CWE-352	Cross-Site Request Forgery (CSRF)	11.73	0	0
10	CWE-434	Unrestricted Upload of File with Dangerous Type	10.41	5	0
11	CWE-862	Missing Authorization	6.90	0	+5

Ca' Foscari University of Venice

Open Web Application Security Project

https://www.owasp.org/

- An online community that produces freely-available articles, methodologies, documentation, tools, and technologies in the field of web application security

- History:
  - Born on September 9th, 2001
  - OWASP foundation established in 2004
  - OWASP Europe VZW registered in Belgium in 2011
- 368 projects
  - https://www.owasp.org/index.php/Category:OWASP_Project#tab=Project_Inventory

Ca' Foscari University of Venice

OWASP Top 10 2017

https://www.owasp.org/index.php/Top_10-2017_Top_10

"The OWASP Top 10 provides a list of the 10 Most Critical Web Application Security Risks"

De-facto standard about most dangerous SW vulnerabilities

ID	Description
1	Injection
2	Broken Authentication
3	Sensitive Data Exposure
4	XML External Entities (XXE)
5	Broken Access Control
6	Security Misconfiguration
7	Cross-Site Scripting (XSS)
8	Insecure Deserialization
9	Using Components with Known Vulnerabilities
10	Insufficient Logging&Monitoring

A1 :2017 Injection

App. Specific	Exploitability: 3	Prevalence: 2	Detectability: 3	Technical: 3	Business ?
Almost any source of data can be an injection vector, environment variables, parameters, external and internal web services, and all types of users. Injection flaws occur when an attacker can send hostile data to an interpreter.		Injection flaws are very prevalent, particularly in legacy code. Injection vulnerabilities are often found in SQL, LDAP, XPath, or NoSQL queries, OS commands, XML parsers, SMTP headers, expression languages, and ORM queries. Injection flaws are easy to discover when examining code. Scanners and fuzzers can help attackers find injection flaws.		Injection can result in data loss, corruption, or disclosure to unauthorized parties, loss of accountability, or denial of access. Injection can sometimes lead to complete host takeover. The business impact depends on the needs of the application and data.	

Example Attack Scenarios

Scenario #1: An application uses untrusted data in the construction of the following **vulnerable** SQL call:

```
String query = "SELECT * FROM accounts WHERE
custID='" + request.getParameter("id") + "'";
```

Scenario #2: Similarly, an application's blind trust in frameworks may result in queries that are still vulnerable, (e.g. Hibernate Query Language (HQL)):

```
Query HQLQuery = session.createQuery("FROM accounts
WHERE custID='" + request.getParameter("id") + "'");
```

In both cases, the attacker modifies the 'id' parameter value in their browser to send: **' or '1'='1**. For example:

```
http://example.com/app/accountView?id=' or '1'='1
```

This changes the meaning of both queries to return all the records from the accounts table. More dangerous attacks could modify or delete data, or even invoke stored procedures.

References

OWASP

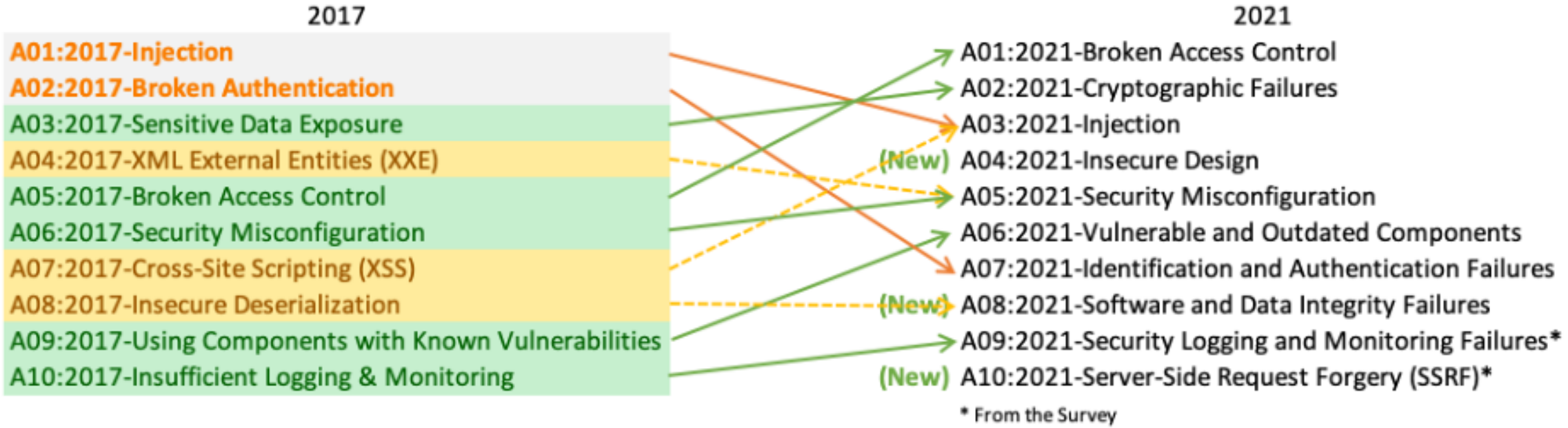
- OWASP Proactive Controls: Parameterize Queries
- OWASP ASVS: V5 Input Validation and Encoding
- OWASP Testing Guide: SQL Injection, Command Injection, ORM injection
- OWASP Cheat Sheet: Injection Prevention
- OWASP Cheat Sheet: SQL Injection Prevention
- OWASP Cheat Sheet: Injection Prevention in Java
- OWASP Cheat Sheet: Query Parameterization
- OWASP Automated Threats to Web Applications – OAT-014

External

- CWE-77: Command Injection
- CWE-89: SQL Injection
- CWE-564: Hibernate Injection
- CWE-917: Expression Language Injection
- PortSwigger: Server-side template injection

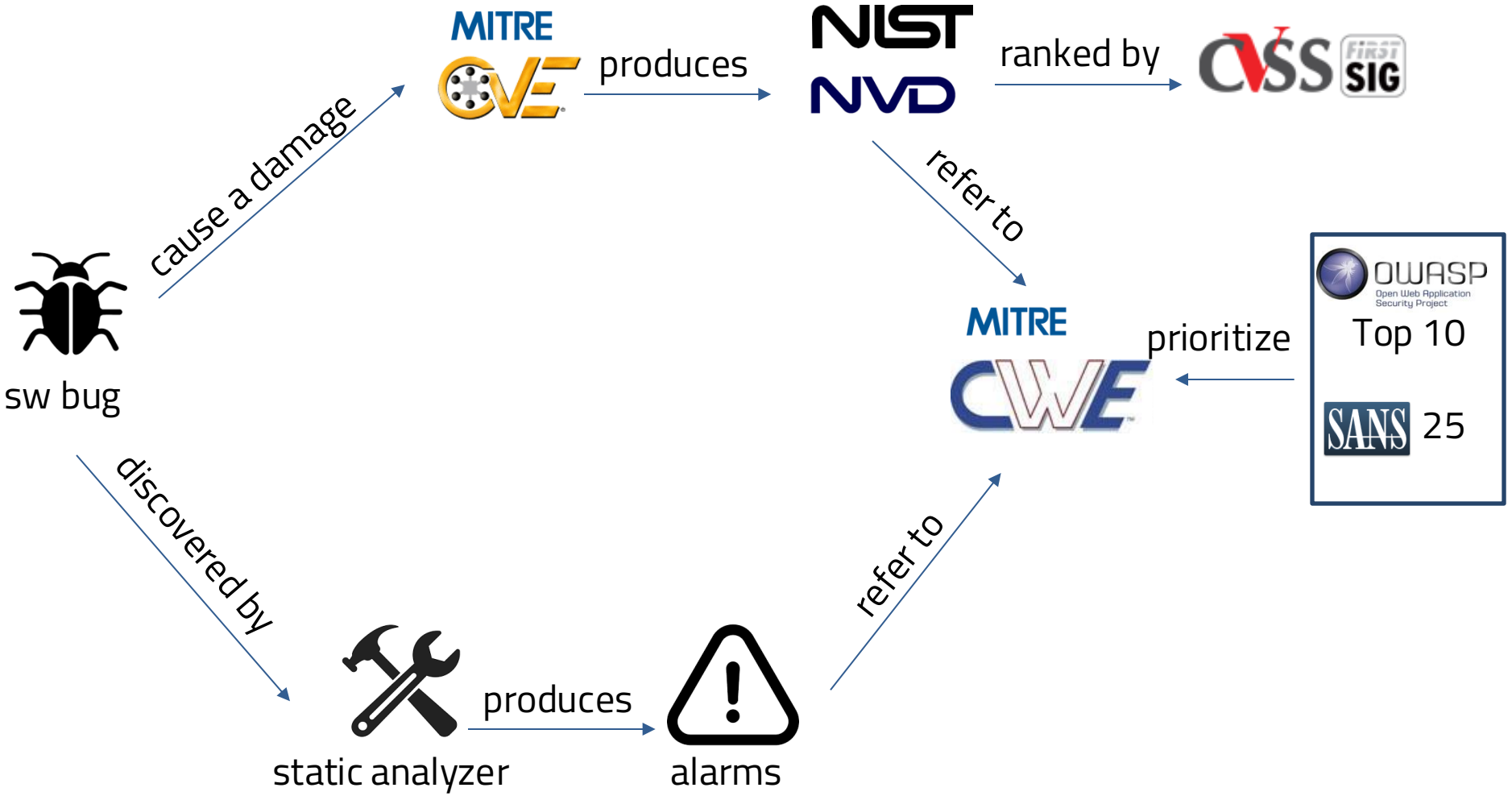
From 2017 to 2021

https://owasp.org/www-project-top-ten/



- **A03:2021-Injection** slides down to the third position. 94% of the applications were tested for some form of injection, and the 33 CWEs mapped into this category have the second most occurrences in applications. Cross-site Scripting is now part of this category in this edition.

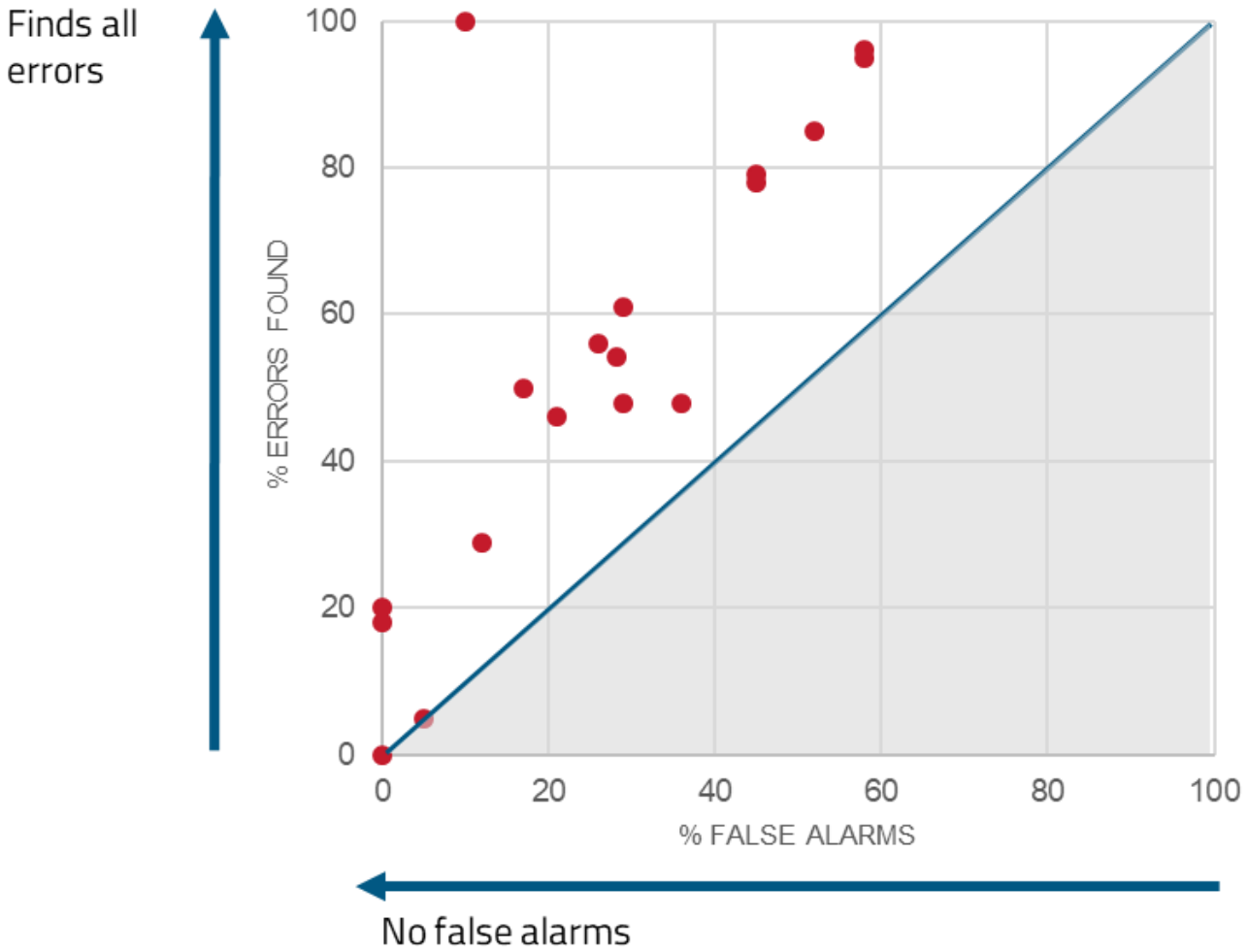
https://owasp.org/Top10/A03_2021-Injection/


MITRE
CVE
produces
NIST
NVD
ranked by
CVSS
FIRST SIG
cause a damage
refer to
sw bug
MITRE
CWE
prioritize
OWASP
Open Web Application Security Project
Top 10
SANS
25
discovered by
produces
refer to
static analyzer
alarms

The OWASP Benchmark

https://www.owasp.org/index.php/Benchmark

- Free and open test suite
  - Evaluate the speed, coverage, and accuracy of static analyzers
  - Fully runnable and exploitable
  - Security issues like injection, XSS, weak cryptographic algorithms, …
- About 2700 Java test cases
  - evaluate program analyzers on OWASP Top 10 vulnerabilities
- Mapped to CWE vulnerability ids
  - True positive: vulnerabilities detected by the tool (coverage)
  - False positive: warning that is not a real vulnerability (precision)


Finds all errors
% ERRORS FOUND
100
80
60
40
20
0
0
20
40
60
80
100
% FALSE ALARMS
No false alarms

Ca' Foscari University of Venice

NIST Software Assurance Metrics And Tool Evaluation

The SAMATE project is dedicated to improving software assurance by developing methods to enable **software tool evaluations**, **measuring the effectiveness** of tools and techniques, and **identifying gaps** in tools and methods. The scope of the SAMATE project is **broad**: ranging from operating systems to firewalls, SCADA to web applications, **source code security analyzers** to correct-by-construction methods.

- Define the Bug Framework
  - https://samate.nist.gov/BF/
- Juliet test suite for C/C++ and Java
  - https://samate.nist.gov/SARD/testsuite.php
- Software Assurance Reference Dataset Project
  - https://samate.nist.gov/SARD/

Ca' Foscari University of Venice

Web Application Security Consortium (WASC)

http://www.webappsec.org/

*The Web Application Security Consortium (WASC) is 501c3 non profit made up of an **international** group of experts, industry practitioners, and organizational representatives who produce **open source** and widely agreed upon **best-practice security standards for the World Wide Web**.*

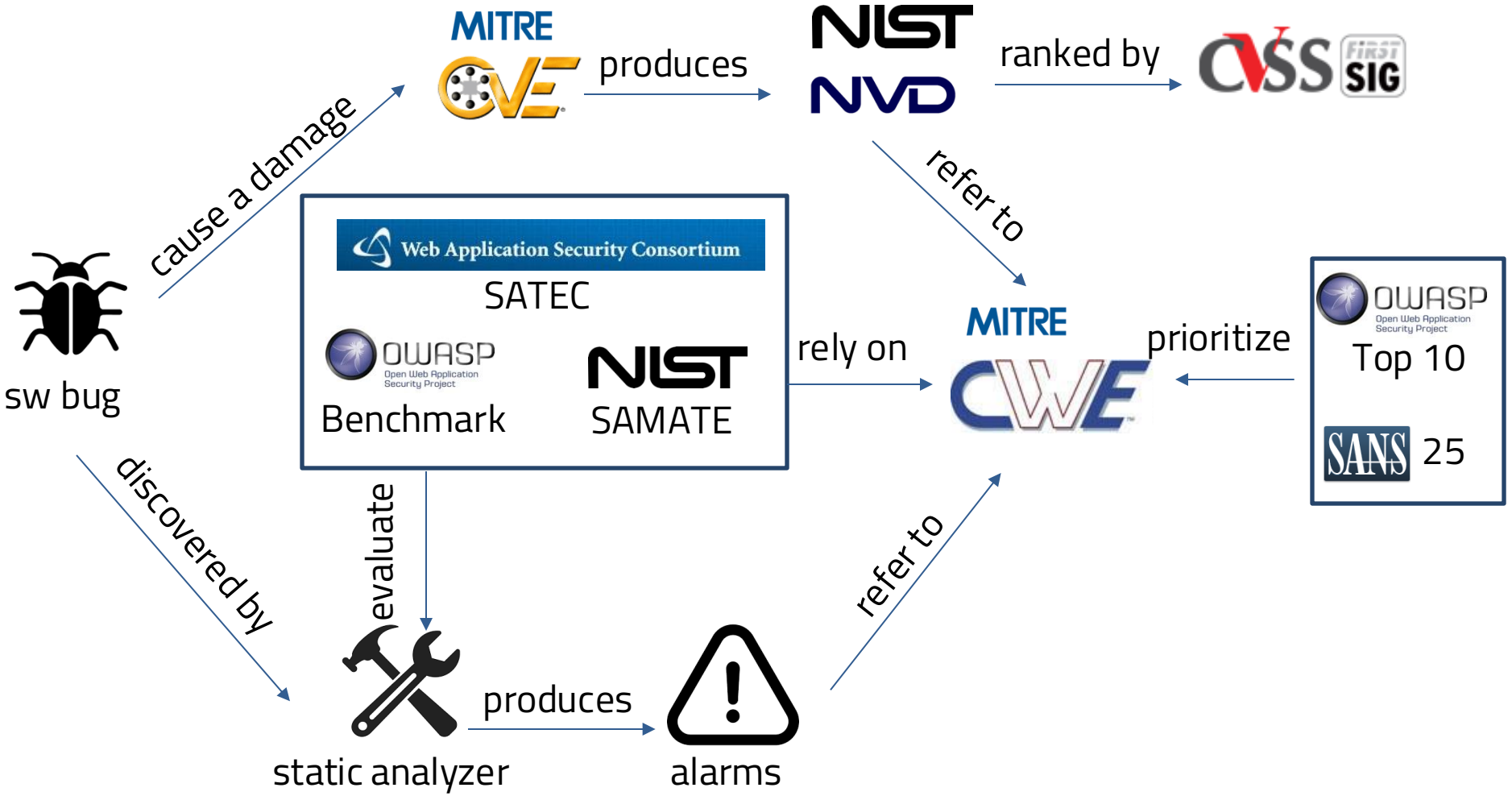
- Exchange of ideas, several industry projects
- Releases technical information, contributed articles, security guidelines, etc..
- Focused to: businesses, educational institutions, governments, application developers, security professionals, and software vendors

Ca' Foscari University of Venice

WASC Static Analysis Technologies Evaluation Criteria

The goal of the SATEC project is to create a **vendor-neutral** set of criteria to help guide application security professionals during the process of **acquiring a static code analysis technology** that is intended to be used during **source-code driven security programs**.

- Guidelines to evaluate commercial static analyzers
  - Based on "functionalities" and not on test cases
  - More about coverage than precision
- Spreadsheet to evaluate different tools


MITRE
CVE
produces
NIST
NVD
ranked by
CVSS
FIRST SIG
cause a damage
refer to
sw bug
Web Application Security Consortium
SATEC
OWASP
Open Web Application Security Project
Benchmark
NIST
SAMATE
rely on
MITRE
CWE
prioritize
OWASP
Open Web Application Security Project
Top 10
SANS
25
discovered by
evaluate
produces
refer to
static analyzer
alarms

Ca' Foscari University of Venice

Standards

International standards and regulations require compliance for assuring software quality for several industries.

Recommended techniques include static analysis.

ISO DIS 26262-6
Road vehicles – functional safety

RTCA DO-178C
Airborne systems and equipment certification

IEC 62304:2006
Medical device software

Ca' Foscari University of Venice

Privacy

GDPR: European General Data Protection Regulation

1

Implement appropriate technical and organisational measures in order to ensure data protection

2

Report data breach no later than **72 hours**

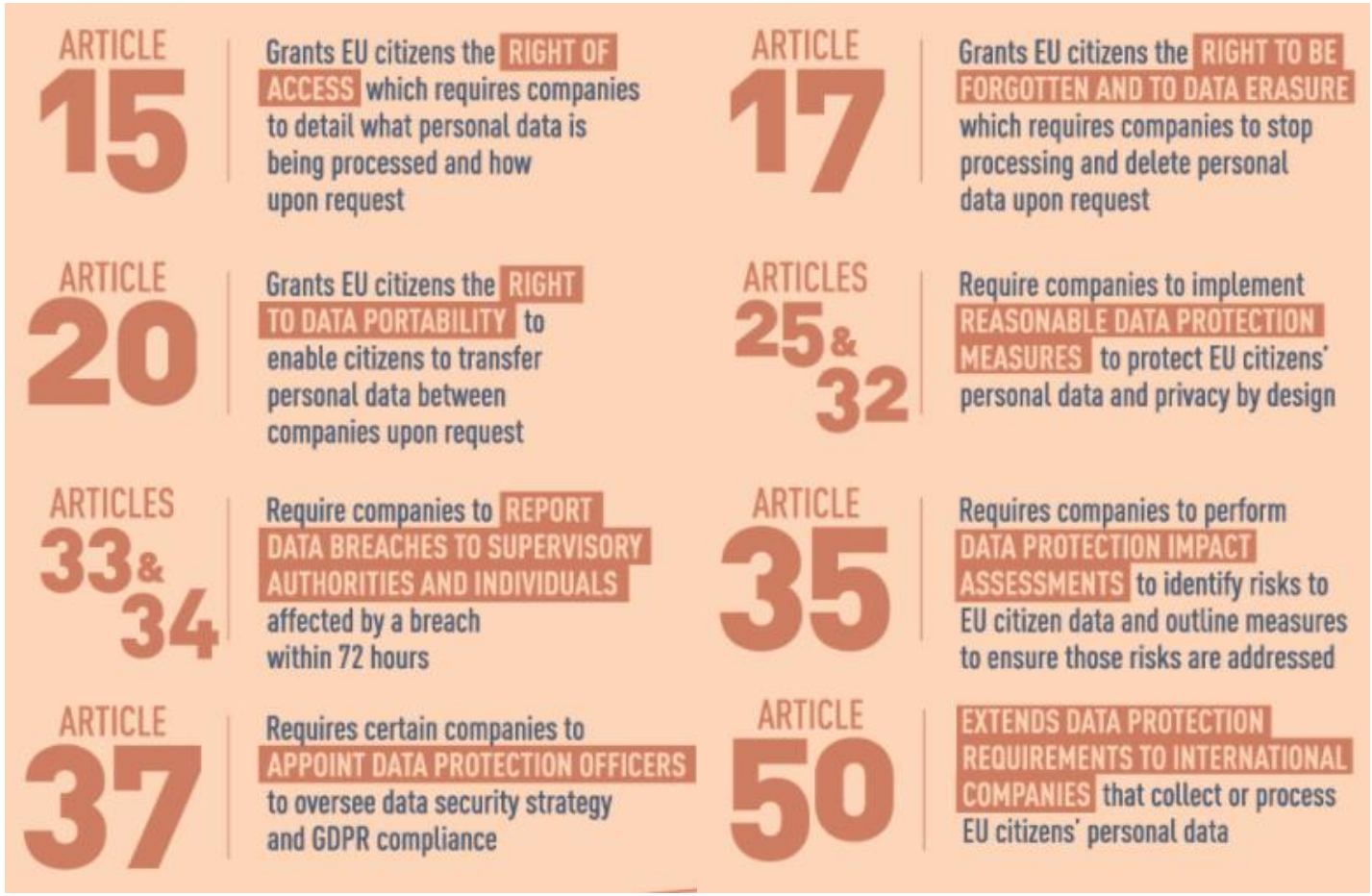
3

Fines up to **4%** of global revenue or **€ 20 Million**

COMPLIANCE BY MAY 2018

Ca' Foscari University of Venice

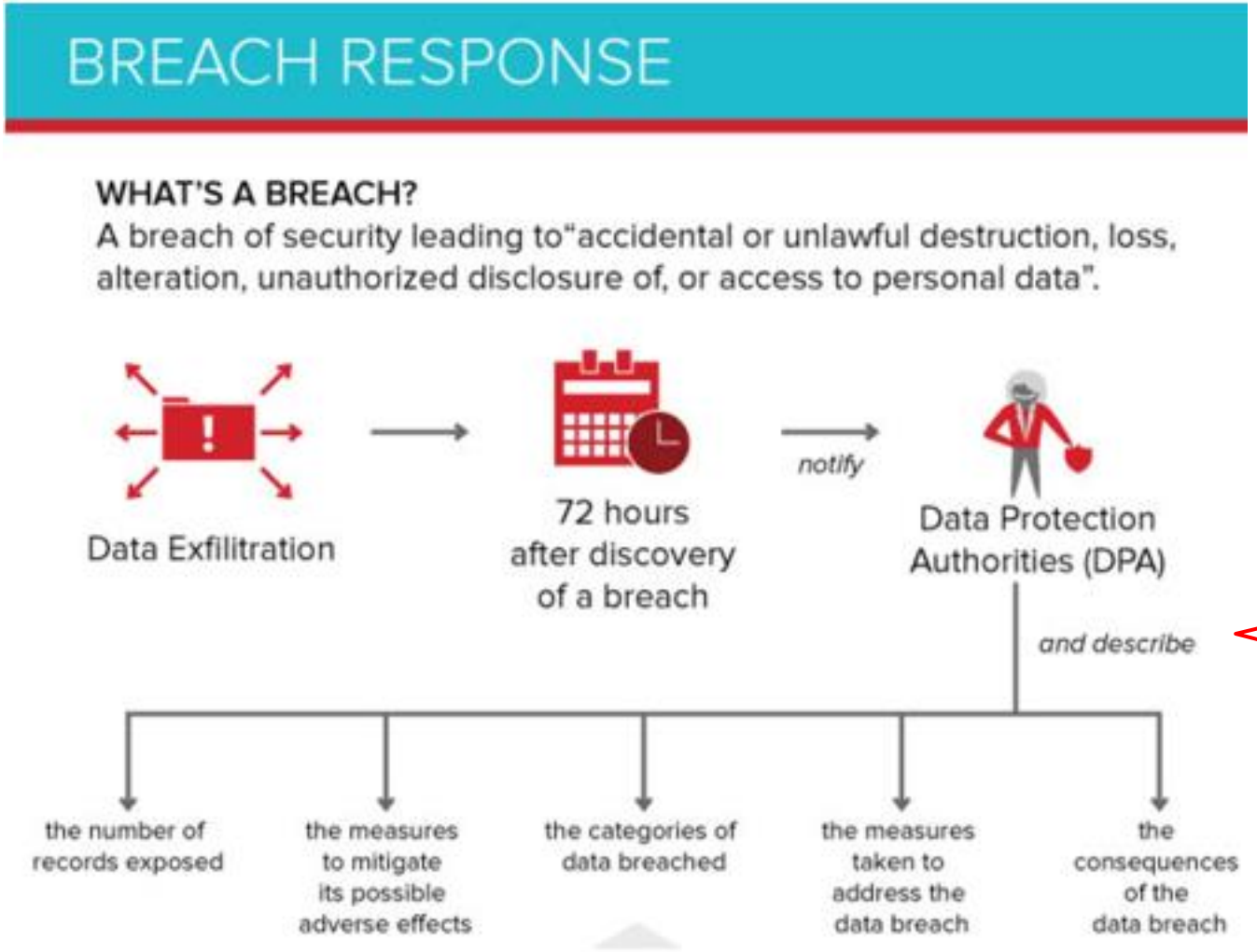
Focal points



GDPR

Ca' Foscari University of Venice

In case of Data Breach: support forensics



Sanctions up to **4%** of global revenue, or **€ 20 Mn**

In case of incident use Julia to trace data, identify vulnerable points and **gain awareness** of all the data possibly compromized.

Ca' Foscari University of Venice

OWASP Top 10 from 2013 to 2017

ID	2013
1	Injection
2	Broken Authentication and Session Management
3	Cross-Site Scripting (XSS)
4	Insecure Direct Object References
5	Security Misconfiguration
6	Sensitive Data Exposure
7	Missing Function Level Access Control
8	Cross-Site Request Forgery (CSRF)
9	Using Components with Known Vulnerabilities
10	Unvalidated Redirects and Forwards

ID	2017
1	Injection
2	Broken Authentication
3	Sensitive Data Exposure
4	XML External Entities (XXE)
5	Broken Access Control
6	Security Misconfiguration
7	Cross-Site Scripting (XSS)
8	Insecure Deserialization
9	Using Components with Known Vulnerabilities
10	Insufficient Logging&Monitoring

Ca' Foscari University of Venice

Sensitive Data Exposure From 2013 to 2017

A6 2013

Am I Vulnerable to Data Exposure?

The first thing you have to determine is which data is sensitive enough to require extra protection. For example, passwords, credit card numbers, health records, and personal information should be protected. For all such data:

A3 :2017

Is the Application Vulnerable?

The first thing is to determine the protection needs of data in transit and at rest. For example, passwords, credit card numbers, health records, personal information and business secrets require extra protection, particularly if that data falls under privacy laws, e.g. EU's General Data Protection Regulation (GDPR), or regulations, e.g. financial data protection such as PCI Data Security Standard (PCI DSS). For all such data:

Ca' Foscari University of Venice

Sensitive Data Exposure

Am I Vulnerable?

- Determine protection needs
- Consider if falls under regulation
- Any data transmitted in clear text?
- Stored in clear text?
- Weak cryptographic algorithms?

How do I Prevent This?

- Classify data
- Review privacy laws
- Encrypt all sensitive data
- Encrypt all data in transit
- Ensure strong cryptographic algorithms

Static program analyzers can help you checking that

- **All sensitive data is encrypted**
- **All data in transit is encrypted**
- **Weak cryptographic algorithms are not used**

Ca' Foscari University of Venice

Sensitive Data Exposure Current coverage

OWASP Open Web Application Security Project

A6 2013

- CWE Entry 310 on Cryptographic Issues
- CWE Entry 312 on Cleartext Storage of Sensitive Information
- CWE Entry 319 on Cleartext Transmission of Sensitive Information
- CWE Entry 326 on Weak Encryption

Julia

- CWE-319: Injection
- CWE-327, 328: Cryptography
  - Sons of CWE-326

SonarQube

- CWE-326, 327, 328:
  - Rules S2089, S2278, S2245, S2277, S2258, S2257, S2070

Sensitive Data Exposure From 2013 to 2017

A6 2013

- CWE Entry 310 on Cryptographic Issues
- CWE Entry 312 on Cleartext Storage of Sensitive Information
- CWE Entry 319 on Cleartext Transmission of Sensitive Information
- CWE Entry 326 on Weak Encryption

A3 :2017

CWE-359 Exposure of Private Information (Privacy Violation)
CWE-220 Exposure of sens. information through data queries
CWE-310: Cryptographic Issues; CWE-326: Weak Encryption
CWE-312: Cleartext Storage of Sensitive Information
CWE-319: Cleartext Transmission of Sensitive Information

Ca' Foscari University of Venice

CWE 359
Exposure of Private Information

The software does not properly prevent private data (such as credit card numbers) from being accessed by actors who either (1) are not explicitly authorized to access the data or (2) do not have the implicit consent of the people to which the data is related.

SPAs might be applied to

1. Comprehensively identify all accesses to personal data
2. Track its flow in the program
3. Detect where such data could be leaked

Examples

Leakage

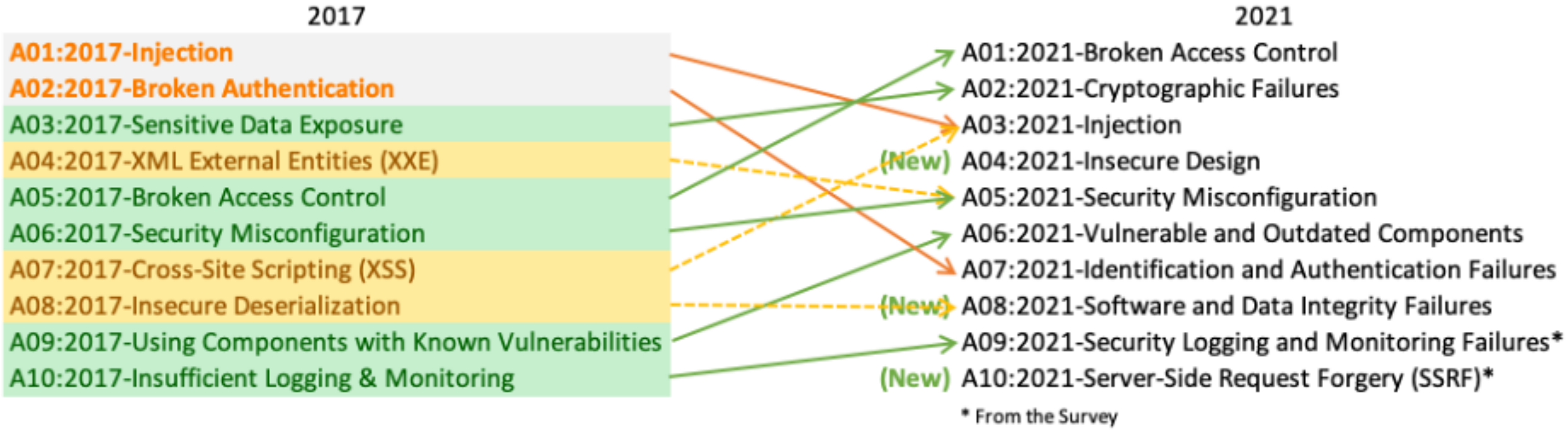
```
pass = getPassword();
log.write(pass);
```

Sensible data not needed

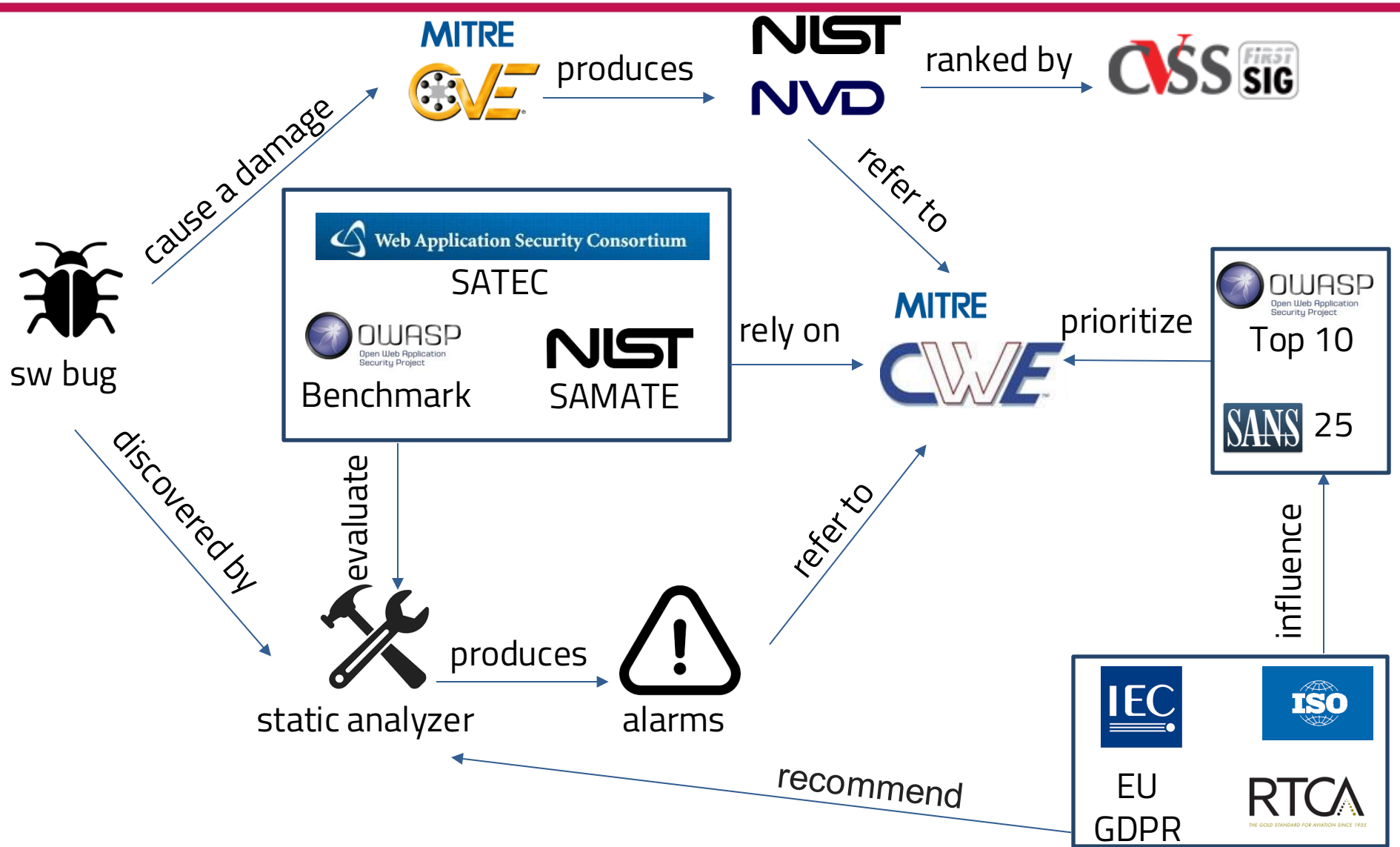
```
Permission ACCESS_FINE_LOCATION
Location loc = getLastLocation();
deriveStateFromCoords(loc);
ACCESS_COARSE_LOCATION enough!
```

Ca' Foscari University of Venice

From 2017 to 2021



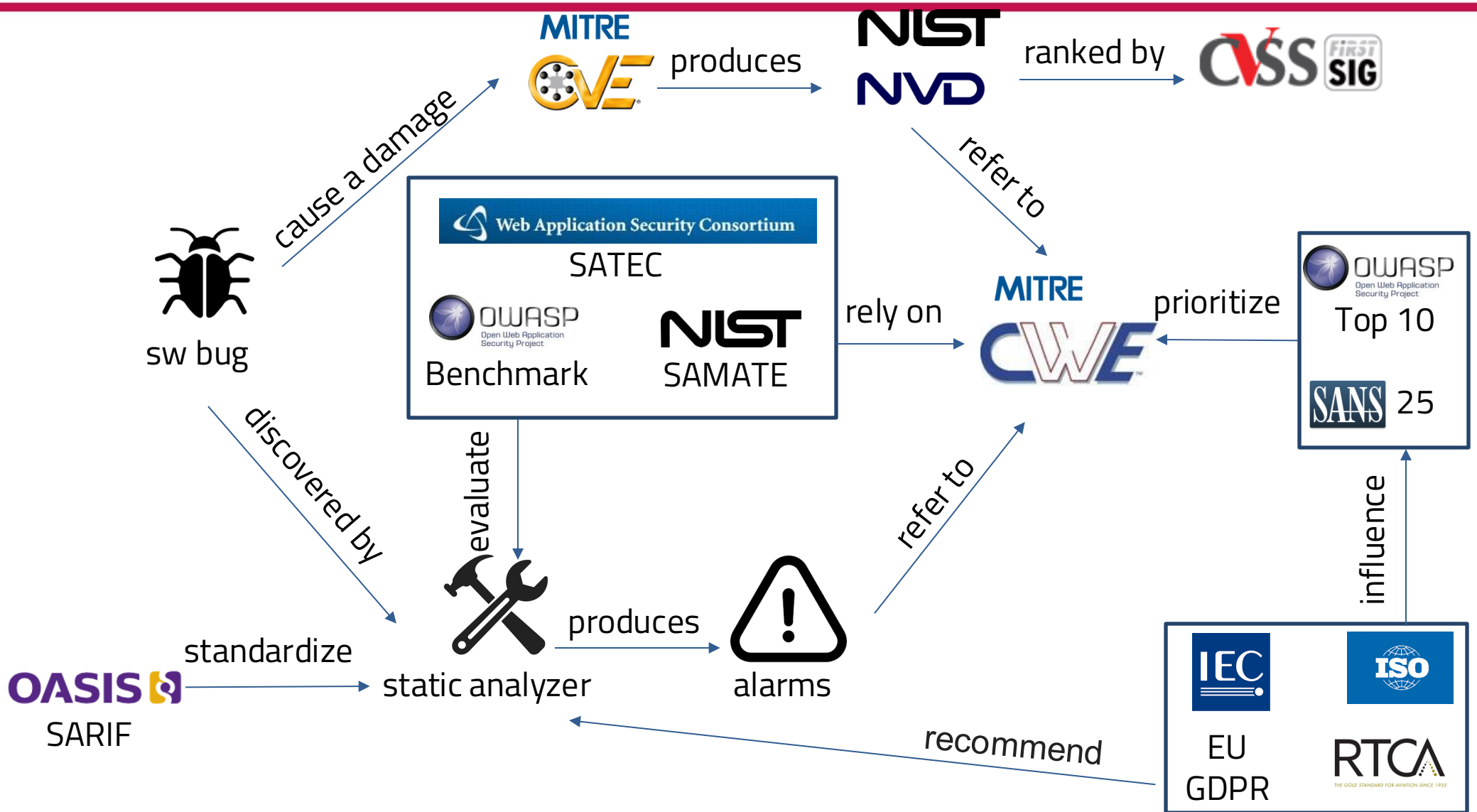
Huge effort to collect data: https://owasp.org/www-project-top-ten/#div-data_2020
Many more CWE ids… but 359 disappeared!


MITRE
CVE
produces
NIST
NVD
ranked by
CVSS
FIRST SIG
cause a damage
refer to
sw bug
Web Application Security Consortium
SATEC
OWASP
Open Web Application Security Project
Benchmark
NIST
SAMATE
rely on
MITRE
CWE
prioritize
OWASP
Open Web Application Security Project
Top 10
SANS
25
discovered by
evaluate
refer to
influence
produces
static analyzer
alarms
IEC
ISO
EU
GDPR
RTCA
recommend

Ca' Foscari University of Venice

OASIS Static Analysis Results Interchange Format (SARIF)

- https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=sarif
- Standard output format for static analysis tools
- Allow developers and teams to:
  - view, understand, interact with, and manage
  - the results produced by a variety all the tools that they use
- Support aggregation of the results of many tools
- Comprehensively capture the range of data produced by commonly used static analysis tools
- Capture information useful for assessing a project's compliance with corporate policy or conformance to certification standards.
- Current standard


MITRE
CVE
produces
NIST
NVD
ranked by
CVSS
FIRST SIG
cause a damage
refer to
sw bug
Web Application Security Consortium
SATEC
OWASP
Open Web Application Security Project
Benchmark
NIST
SAMATE
rely on
MITRE
CWE
prioritize
OWASP
Open Web Application Security Project
Top 10
SANS
25
discovered by
evaluate
refer to
influence
produces
standardize
OASIS
SARIF
static analyzer
alarms
IEC
ISO
EU
GDPR
RTCA
recommend