BBVA

Creating Opportunities

SUSTO: Systematic Universal Security Testing Orchestration

Collaboration in Security and Compliance

May 2021

BBVA
Creating Opportunities

Index

BBVA

Creando Oportunidades

MM0003CA:~ whoami luissaiz

Luis Saiz Gimeno

@lsaiz

Telecomm. Eng. - Cryptography - Sys.Sec - Info.Sec - Tech. Fraud Prevention - Fraud Prevention Tech. - Global Security Center - Innovation in Security @BBVA

Madrid

bbva.com/en/featured/bb…

Se unió en julio de 2010

BBVA
Creating Opportunities

01

Intro

Who we are?

BBVA in numbers

€736
miles de millones de activo total

80,7
millones de clientes

>30
países

7.432
oficinas

31.000
cajeros

123.174
empleados

Información a cierre de diciembre de 2020. El mapa excluye aquellos países en los que BBVA no tiene sociedad o el nivel de actividad es reducido

BBVA
Creating Opportunities

Innovation Labs

- Innovation is about **new questions**, not new answers
- **Hyperscale, AI, Cybersecurity**
- Multidisciplinar cross pollination
- Learning by doing
- Doing to learn

BBVA

Creating Opportunities

Our way of doing

- Explorations vs projects
- Team self prioritization of their own agenda
- Coolness is not an acceptance criteria
- Pre approved exploration
- From Fail Fast to Drop Fast
- Collateral benefits… always

BBVA

BBVA

Madrid http://bbva.com opensource@bbva.com

Repositories 97 Packages People 36 Teams 6 Projects 1

Pinned repositories

kapow
Kapow! If you can script it, you can HTTP it.
Go ☆ 505 20

apicheck
The DevSecOps toolset for REST APIs
Python ☆ 118 30

qed
The scalable, auditable and high-performance tamper-evident log project
Go ☆ 73 13

patton
The clever vulnerability dependency finder
Gherkin ☆ 79 11

timecop
Time series based anomaly detector
Python ☆ 56 13

susto
Systematic Universal Security Testing Orchestration
☆ 10

BBVA
Creating Opportunities

02

The problem

Security and Compliance in Agile times


NEED FOR SPEED
PAYBACK
POLICE

Goals Alignment

- Adjust development lifecycle security to **rapid** evolving risks in **design time**
  - **Fast** changing threats
  - **Agile** product development
- **Accelerate** detection and reaction to abnormal behaviours **in runtime**
- Achieve **constant** risk exposure **monitoring & assurance**

Help Net Security
July 31, 2019

53% of enterprises have no idea if their security tools are working

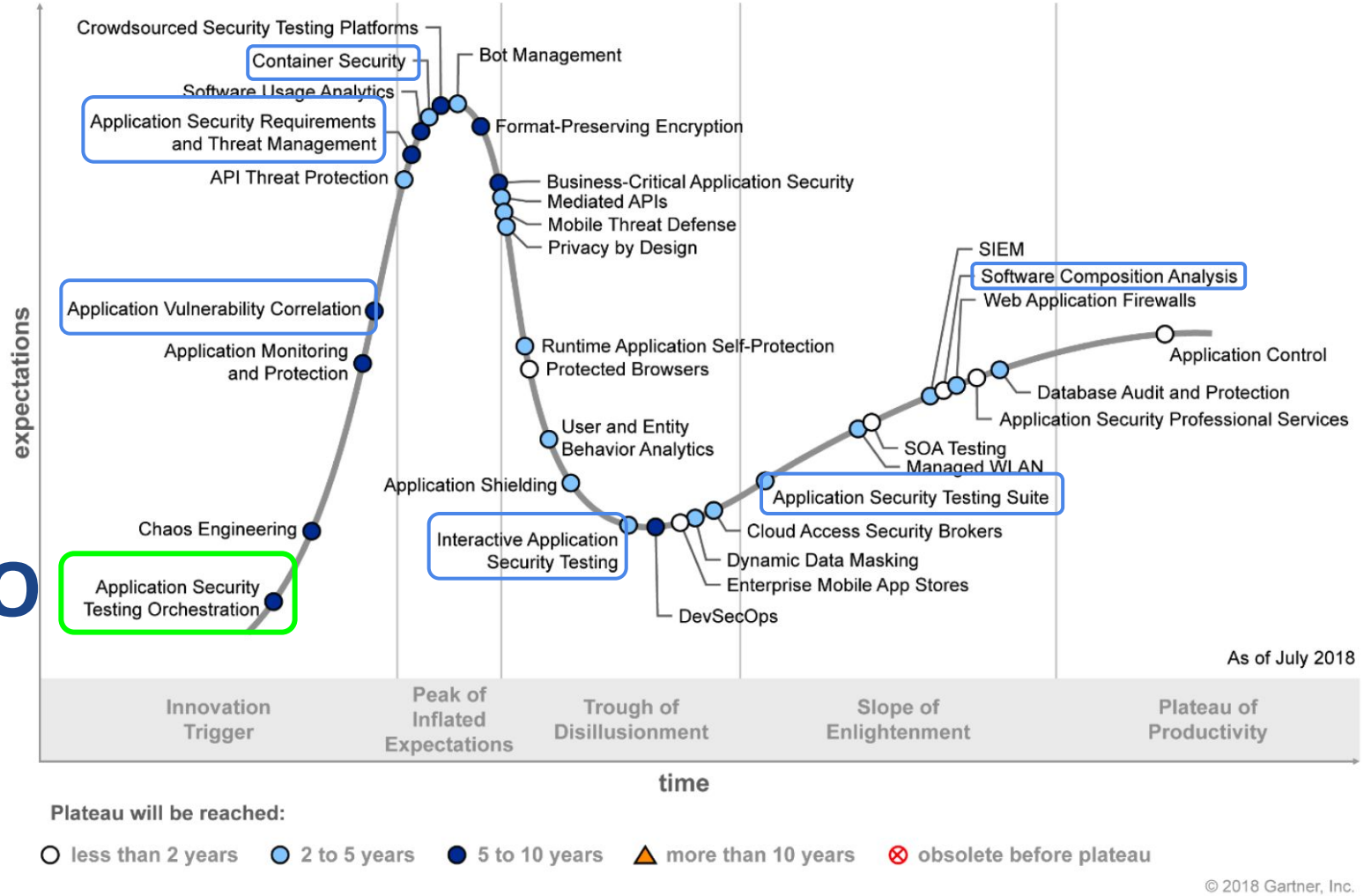
The majority of organizations don't know if the security tools they deploy are working, and are not confident they can avoid data breaches, according to AttackIQ.

Problems created by not determining the efficacy of security technologies

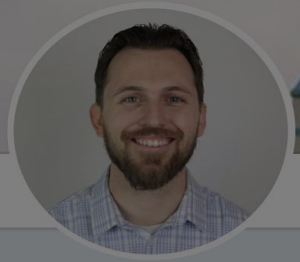

RISK MANAGEMENT


COMPLIANCE

Hype Cycle for Application Security, 2018



ASTO



Justin Garrison
@rothgar

Here to spread love and learn from others. Co-author of cnibook.info Tweets about tech and K8s. ❤ community & OSS. @DisneyAnimation - DMs open

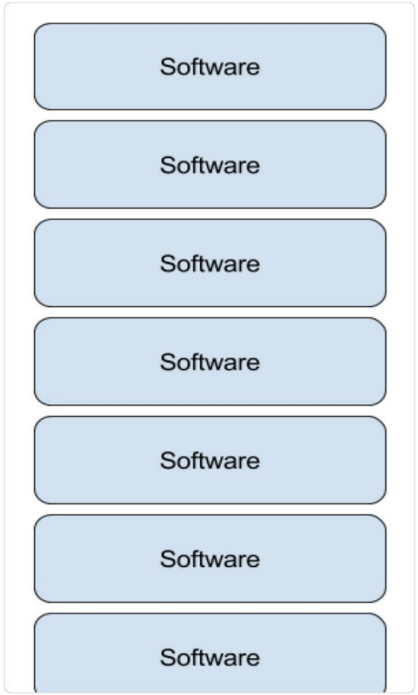
Clever phrase
medium.com/@rothgar
Se unió en abril de 2008

Justin Garrison
@rothgar

Siguiendo

The new OSI model is much easier to understand

Traducir Tweet



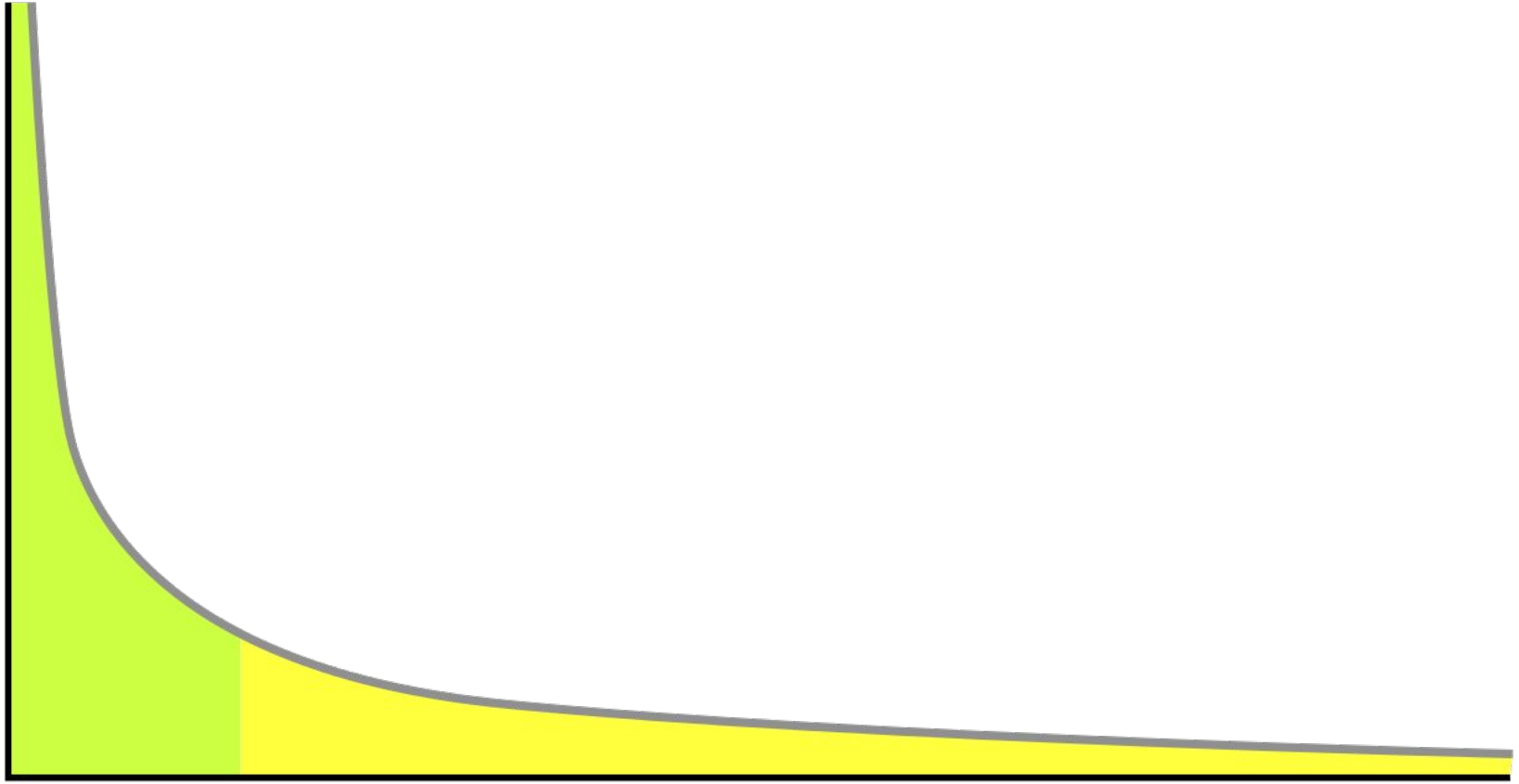
18:22 - 18 jul. 2017

2.820 Retweets **3.930** Me gusta

93 2,8K 3,9K

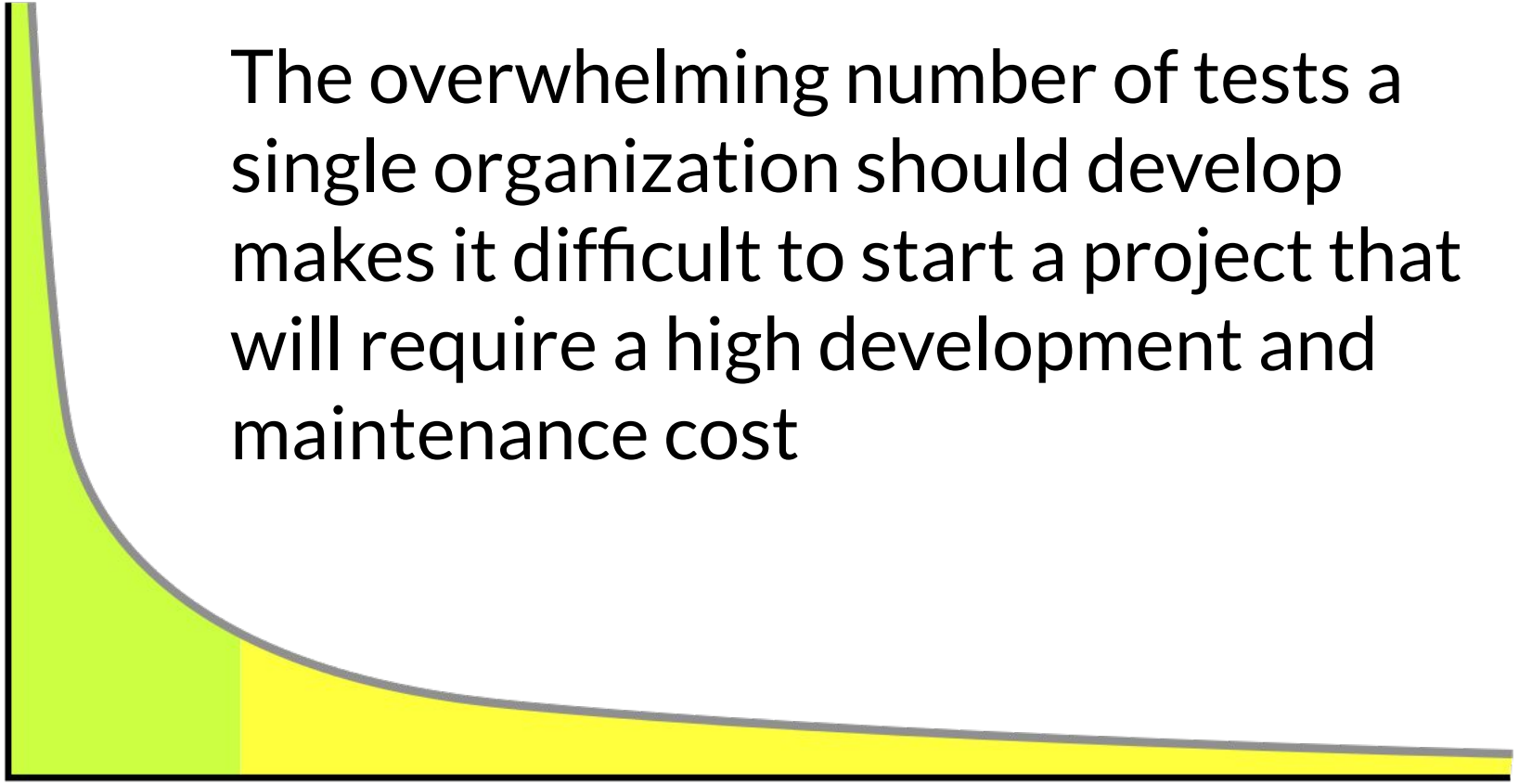
SdE

The long tail of security control checks



Picture by Hay Kranen

The long tail of security control checks



The overwhelming number of tests a single organization should develop makes it difficult to start a project that will require a high development and maintenance cost

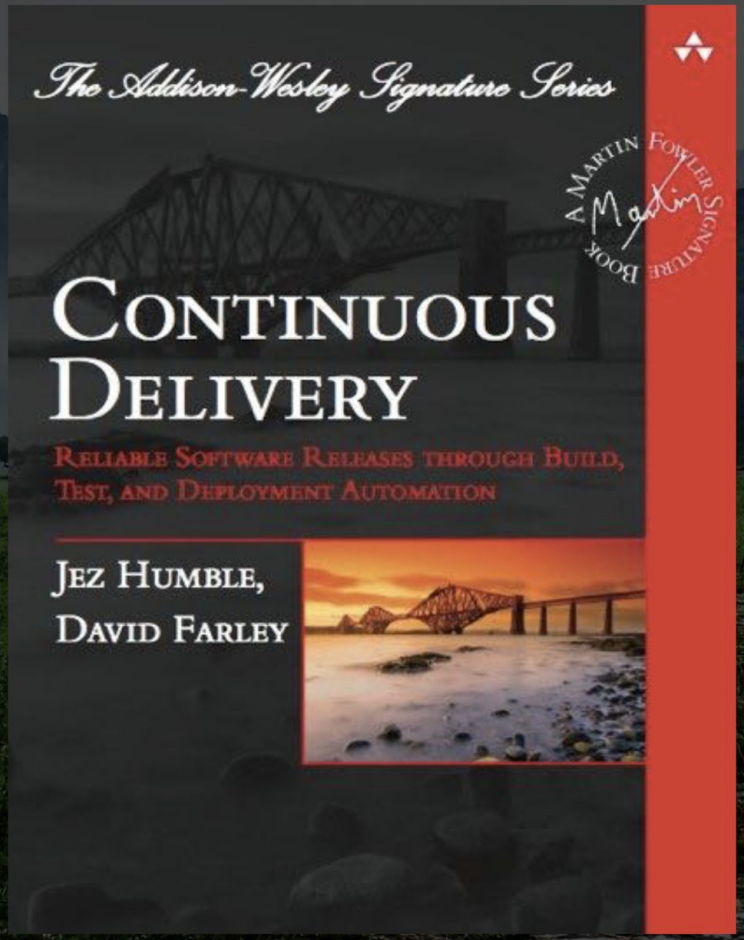
Picture by Hay Kranen

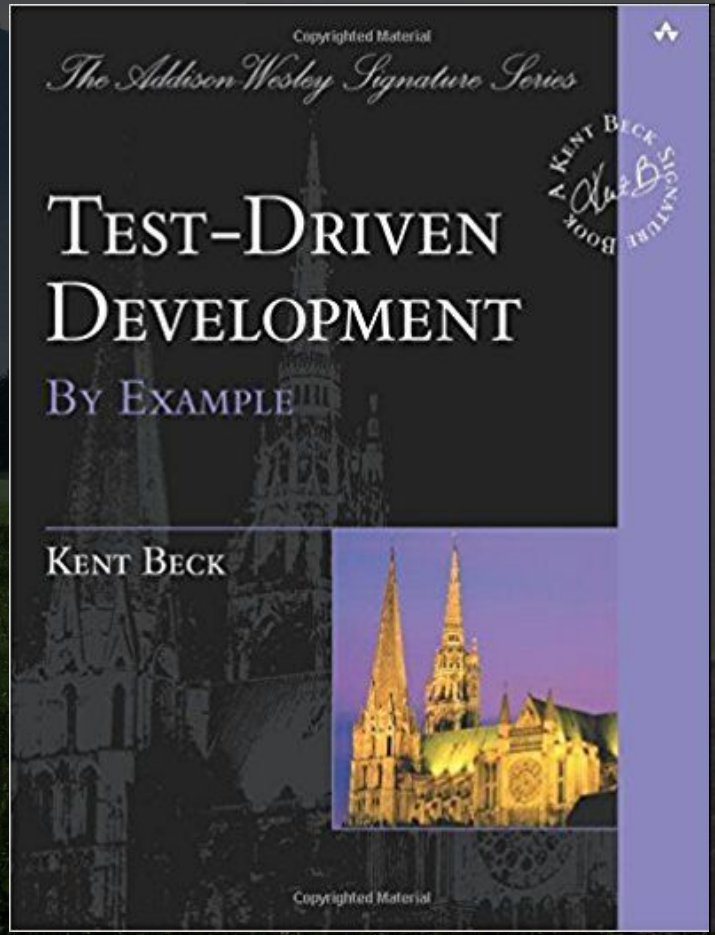
BBVA
Creating Opportunities

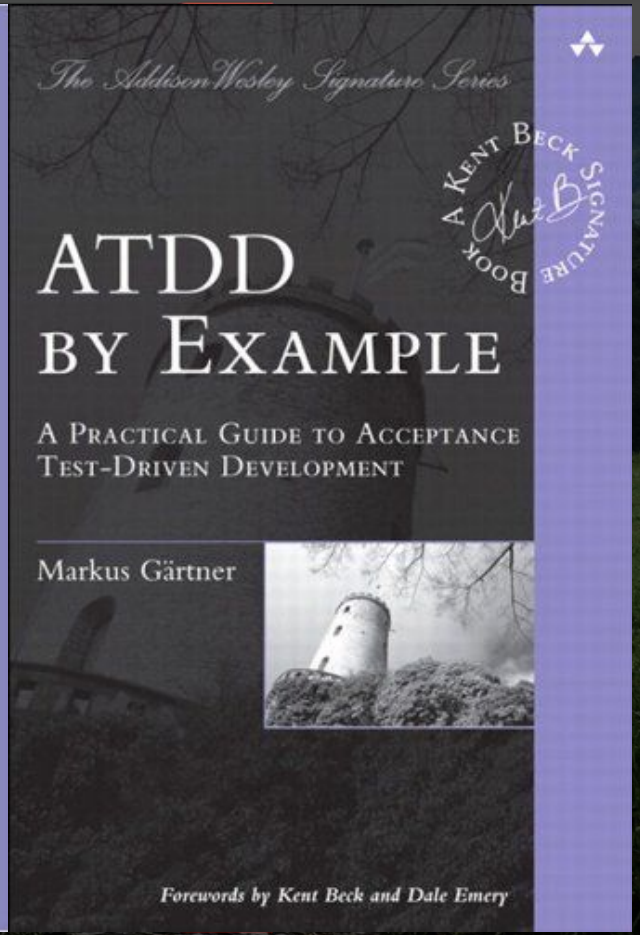
03

Looking for other successful patterns

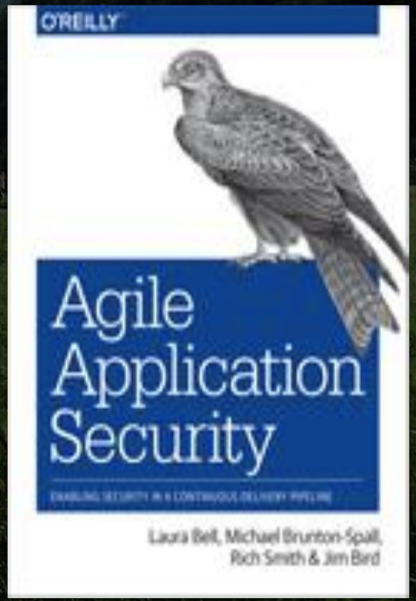
Testing and security


The Addison-Wesley Signature Series
A Martin Fowler Signature Book
Martin
Continuous Delivery
Reliable Software Releases through Build, Test, and Deployment Automation
Jez Humble,
David Farley


Copyrighted Material
The Addison Wesley Signature Series
A Kent Beck Book Signature
Test-Driven Development
By Example
Kent Beck
Copyrighted Material


The Addison Wesley Signature Series
A Kent Beck Book Signature
ATDD by Example
A Practical Guide to Acceptance Test-Driven Development
Markus Gärtner
Forewords by Kent Beck and Dale Emery

[Deploys] can be treated as standard or routine changes that have been pre-approved by management, and that don't require a heavyweight change review meeting.



Feature testing

```
Feature: Record a ledger entry
  As an Accountant
  I want to account according COA selected by accountant
  In order to account all movements and comply with legislation and accounting rules

  Scenario: Record a ledger entry
    Given a bank named "Gringotts"
    And "Gringotts" bank has the default Accounting Periods, Chart of Accounts and Statements

    When the system registers an financial event with date "2016-05-25"
      | description                       | accountCode | Debit    | Credit   |
      | Cash and Balance with central Banks | 11000       | 1000 EUR |          |
      | Share Capital                     | 31000       |          | 1000 EUR |

    Then the following accounting entries in the Journal are added
      | description                | entryDate  | requestDate | accountingPeriod | periodCodes              |
      | society (bank) constitution | 2016-05-25 | 2016-05-25  | FY2016           | 2016M05, 2016Q2, FY2016 |

    And all accounting movements are recorded
      | description                       | accountCode | Debit    | Credit   |
      | Cash and Balance with central Banks | 11000       | 1000 EUR |          |
      | Share Capital                     | 31000       |          | 1000 EUR |
```

Simple tests can be safely extrapolated

Business dependent: Difficult to reuse

A lesson about intentional risk

https://pixabay.com/photos/window-window-grille-lattice-window-243074/

Security Controls testing complexity

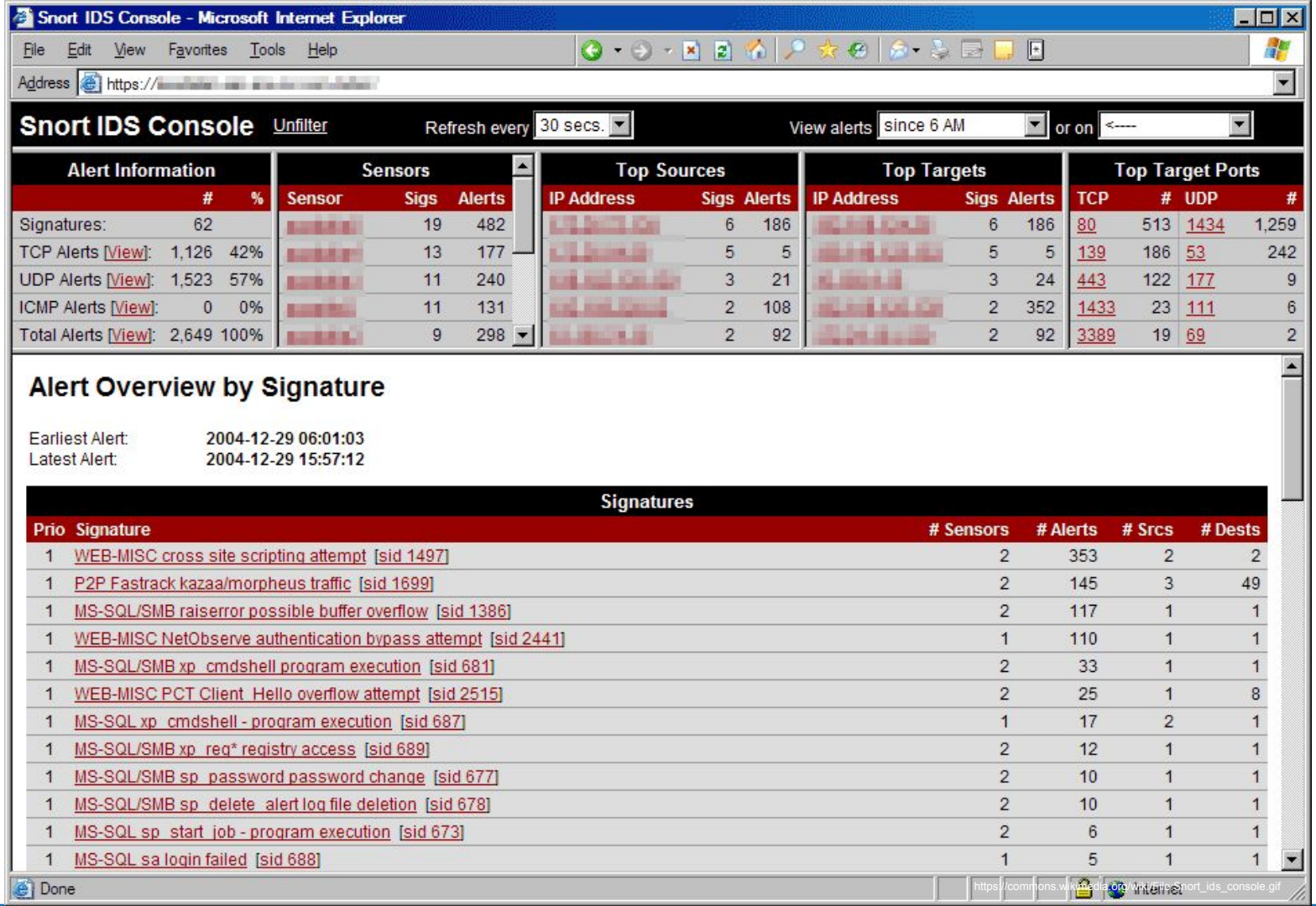
All or nothing + Orchestration

In security we need to test for **"all of" or "none of"** conditions, making it necessary to **pipeline and orchestrate outputs** of tools as inputs of other tools

Standardizing Security Controls Testing

Unlike Functional Features, Security Controls are similar among different projects (Common Controls) and across organizations

We even have industry standards


Snort IDS Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address https://
Snort IDS Console Unfilter Refresh every 30 secs. View alerts since 6 AM or on <----
Alert Information
%
Signatures: 62
TCP Alerts [View]: 1,126 42%
UDP Alerts [View]: 1,523 57%
ICMP Alerts [View]: 0 0%
Total Alerts [View]: 2,649 100%
Sensors
Sensor Sigs Alerts
19 482
13 177
11 240
11 131
9 298
Top Sources
IP Address Sigs Alerts
6 186
5 5
3 21
2 108
2 92
Top Targets
IP Address Sigs Alerts
6 186
5 5
3 24
2 352
2 92
Top Target Ports
TCP # UDP #
80 513 1434 1,259
139 186 53 242
443 122 177 9
1433 23 111 6
3389 19 69 2
Alert Overview by Signature
Earliest Alert: 2004-12-29 06:01:03
Latest Alert: 2004-12-29 15:57:12
Signatures
Prio Signature # Sensors # Alerts # Srcs # Dests
1 WEB-MISC cross site scripting attempt [sid 1497] 2 353 2 2
1 P2P Fastrack kazaa/morpheus traffic [sid 1699] 2 145 3 49
1 MS-SQL/SMB raiserror possible buffer overflow [sid 1386] 2 117 1 1
1 WEB-MISC NetObserve authentication bypass attempt [sid 2441] 1 110 1 1
1 MS-SQL/SMB xp_cmdshell program execution [sid 681] 2 33 1 1
1 WEB-MISC PCT Client_Hello overflow attempt [sid 2515] 2 25 1 8
1 MS-SQL xp_cmdshell - program execution [sid 687] 1 17 2 1
1 MS-SQL/SMB xp_reg* registry access [sid 689] 2 12 1 1
1 MS-SQL/SMB sp_password password change [sid 677] 2 10 1 1
1 MS-SQL/SMB sp_delete_alert log file deletion [sid 678] 2 10 1 1
1 MS-SQL sp_start_job - program execution [sid 673] 2 6 1 1
1 MS-SQL sa login failed [sid 688] 1 5 1 1
Done
Internet
https://commons.wikimedia.org/wiki/File:snort_ids_console.gif

Awesome YARA

A curated list of awesome YARA rules, tools, and resources. Inspired by awesome-python and awesome-php.

> YARA is an ancronym for: YARA: Another Recursive Ancronym, or Yet Another Ridiculous Acronym. Pick your choice.
>
> -- *Victor M. Alvarez (@plusvic)*

YARA, the "pattern matching swiss knife for malware researchers (and everyone else)" is developed by @plusvic and @VirusTotal. View it on GitHub.

build passing

Sigma

Generic Signature Format for SIEM Systems

What is Sigma

Sigma is a generic and open signature format that allows you to describe relevant log events in a straight forward manner. The rule format is very flexible, easy to write and applicable to any type of log file. The main purpose of this project is to provide a structured form in which researchers or analysts can describe their once developed detection methods and make them shareable with others.

Sigma is for log files what Snort is for network traffic and YARA is for files.

build passing

Sigma

Generic Signature Format for SIEM Systems

What is Sigma

Sigma is a generic and open signature format that allows you to describe relevant log events in a straight forward manner. The rule format is very flexible, easy to write and applicable to any type of log file. The main purpose of this project is to provide a structured form in which researchers or analysts can describe their once developed detection methods and make them shareable with others.

Sigma is for log files what Snort is for network traffic and YARA is for files.

Pattern

SUSTO would be for security control testing what

- Sigma is for log files
- Snort is for network traffic
- YARA is for files

BBVA

Creating Opportunities

04

SUSTO: Systematic Universal Security Testing Orchestration

Open Collaboration

Proposing SUSTO

Extending the concept of ASTO

Systematic: Long tail coverage

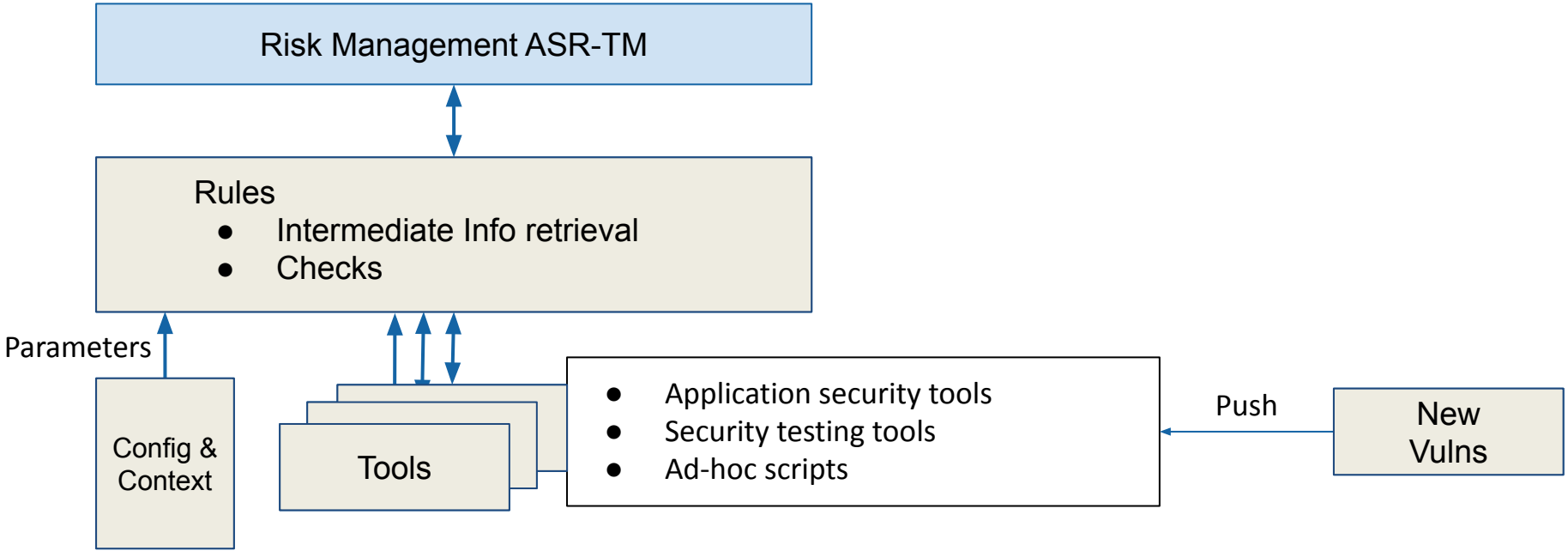
Universal: Full stack testing, not only Application

SUSTO: Also Pun-intended

SUSTO implementation main features

- Tests should be generic in their objectives: WHAT?
- Tests should be configurable in their instantiation
  - Target Instance
  - Desired Goal (Boolean, Score, …)
- It should facilitate the integration of existing tools to
  - Obtain intermediate information
  - Launch the execution of more specific tests
- Manual intervention should be minimized
- Test execution should not introduce new threats

Overlord Architecture



PCI DSS controls

PDF | Kindle | RSS

The PCI DSS security standard in Security Hub supports the following controls. For each control, the information includes the severity, the resource type, the AWS Config rule, and the remediation steps.

[PCI.AutoScaling.1] Auto Scaling groups associated with a load balancer should use health checks

Severity: Low

Resource: Auto Scaling group

AWS Config rule: `autoscaling-group-elb-healthcheck-required`

Parameters: None

This control checks whether your Auto Scaling groups that are associated with a load balancer are using Elastic Load Balancing health checks.

PCI DSS does not require load balancing or highly available configurations. However, this check aligns with AWS best practices.

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 2.2: Develop configuration standards for all system components. Assure that these standards address all known security vulnerabilities and are consistent with industry-accepted system hardening standards.

Replicating systems using load balancing provides high availability and is a means to mitigate the effects of a DDoS event.

This is one method used to implement system hardening configurations.

Remediation

To enable Elastic Load Balancing health checks

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the navigation pane, under **Auto Scaling**, choose **Auto Scaling Groups**.
3. To select the group from the list, choose the right box.
4. From **Actions**, choose **Edit**
5. For **Health Check Type**, choose **ELB**.
6. For **Health Check Grace Period**, enter `300`.
7. Choose **Save**.

For more information on using a load balancer with an Auto Scaling group, see the *Amazon EC2 Auto Scaling User Guide*.

Examples

```
Rule:
    ALL ?users
    WITH ?old_passwords
    MUST BE ?a_disabled_user
```

```
Rule:
        ?vulnerabilities
        OF ?repositories
        CAN'T BE ?critical
```

```
Rule:
        ALL ?possible_connections_to(?server, ?port)
        MUST BE ?from_legitimate_host
```

Illustrative

Examples

Example1.BaseRule

Safe Haskell	None
Language	Haskell2010

Documentation

```
ruleOldPasswords :: (Member Rule effs, _) => Eff effs Bool                    # Source
```

This function is a transcription of the high-level rule that can be found in `src/Example1/BaseRule.rule`:

```
Rule:
    ALL ?users
    WITH ?old_passwords
    MUST BE ?a_disabled_user
```

The signature has a type hole to be filled by the compiler with the found implicit parameters. This is the result:

```
>>> :t ruleOldPasswords
ruleOldPasswords
  :: (Data.OpenUnion.Internal.FindElem Rule effs,
      ?users::Eff effs [t]
      ?old_password::t -> Eff effs Bool,
      ?a_disabled_user::t -> Eff effs Bool,
     ) =>
     Eff effs Bool
```

By the type of the found implicit parameters we know that:

- `?users`: Is a list of elements of some type `t`.
- `?old_password`: Is a function that given an element of type `t` returns a `Bool`.
- `?a_disabled_user`: Is a function that given an element of type `t` returns a `Bool`.

For a concrete implementation of this rule see `ruleLDAPOldPassword`.

Illustrative

Examples

Example1.LDAPOldPasswords

Safe Haskell	None
Language	Haskell2010

This module contains some functions to illustrate a concrete rule (`ruleLDAPOldPassword`) over a general one (`ruleOldPasswords`).

The main purpose is to check if the users of a given LDAP database that haven't changed their password in a while, have been disabled.

Documentation

```
ruleLDAPOldPassword :: (Member Rule effs, _) => Eff effs _
```

Source

This function is a transcription of the high-level rule that can be found in `src/Example1/LDAPOldPasswords.rule`:

```
Rule:
        FROM /src/Example1/BaseRule.rule
Given:
        ?users = ldapUsers
        ?old_password = userOldDays > ?max_user_days
        ?a_disabled_user = disabled
```

The signature has a type hole to be filled by the compiler with the found implicit parameters. This is the result:

```
>>> :t ruleLDAPOldPassword
ruleLDAPOldPassword
  :: (Data.OpenUnion.Internal.FindElem Rule effs,
      ?max_user_days::Integer
     ) =>
     Eff effs Bool
```

This rule simply extends `ruleOldPasswords` filling its holes (`?users`, `?old_password` and `?a_disabled_user`) and generating a new smaller hole (`?max_user_days`).

```
runRuleLDAPOldPassword
```

Source

`:: Integer`	The maximum number of days allowed for an LDAP password to be unchanged before the user becomes disabled
`-> IO Bool`	The result of running the rule over the (fake) LDAP database

Run `ruleLDAPOldPassword` filling the remaining hole (`?max_user_days`) with the given parameter

Illustrative

Stakeholders

CSPs/XaaS (AWS/GCP/Azure/Salesforce/RedHat/...)

Security Vendors

- Application Security Requirements & Threat Management
- Governance, Risk&Compliance
- SAST/DAST/IAST/SCA/AVC (Build-time checking tools)
- SOAR (Run-time response automation tools)

Start-ups opportunity

And anyway: **Your organization**

Next steps: Feedback&Contribution

Roadmap & Features

“Overlord” as the OWASP SUSTO tool

Plug-ins to existing tools

Map standard control sets to Rules

Contribute Community Rules

Contribute CSP/Vendor Rules


Don't fear the long tail
You better choose
SUSTO
THANKS
@lsaiz
You
And you
https://bbvalabs.gitbook.io/oss/bbva_labs_security
CC0 https://pxhere.com/es/photo/792049