ThoughtWorks®

TECHNOLOGY RADAR *SUMMIT*

2017技术雷达峰会

洞察构建未来的技术和趋势

5.13@Beijing

ThoughtWorks®

DEV SECURITY

Speaker: 马伟

一个漏洞引发的反思

如有雷同，说明你也踩过同样的坑...

遭遇安全事故

发现异常

订单总价和实际商品不符合

调查结果

某个影响价格计算的参数，
在等待订单支付期间可被强行修改

疑问

- 为何防火墙没有拦截或者报警?

  该漏洞和业务强相关，防火墙无法提供有力支持

- 为何渗透测试没有报告这个问题?

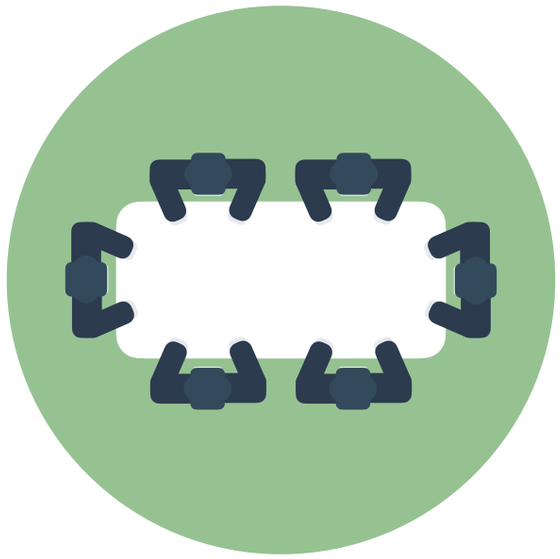
  该漏洞和业务强相关，躲过了渗透测试的检查

多么痛的领悟

严重依赖于渗透测试、防火墙，

然而它们也**有解决不了的问题。**

反思回顾(tù cáo)会议



反思&抱怨

#1 太晚才做渗透测试，

报告的安全问题来不及全部修复

耗时长

数量多

时间紧

反思&抱怨

#2 产品功能在**持续**迭代改进，

然而渗透测试却**没办法每次发布都做**

反思&抱怨

#3 安全活动流于形式

- 威胁建模，拖了很久才做
- 有安全编码规范，但是太过于冗长，没人愿意用
- 理论上团队应该对应用进行安全自查，但很难落地

反思&抱怨

#4 排查第三方组件安全漏洞工作量大，过程太痛苦

你以为就这几个依赖?

```
dependencies {
    compile 'org.springframework.boot:spring-boot-starter-web'
    compile 'org.springframework.boot:spring-boot-starter-data-jpa'
    compile 'org.postgresql:postgresql:9.4.1212'
    compile 'org.apache.poi:poi:3.15'
    compile 'org.apache.poi:poi-ooxml:3.15'
    compile 'io.springfox:springfox-swagger2:2.6.1'
    compile 'io.springfox:springfox-swagger-ui:2.6.1'
    compile 'commons-io:commons-io:2.4'
    compile 'org.apache.commons:commons-lang3:3.5'
    compile 'com.yunpian.sdk:yunpian-java-sdk:1.2.2'
    compile 'org.springframework.boot:spring-boot-starter-security:1.3.0.RC1'
    compile 'io.jsonwebtoken:jjwt:0.7.0'
    compile 'org.json:json:20160810'
    compile 'org.quartz-scheduler:quartz:2.2.3'
    compile 'org.springframework:spring-context-support:4.1.6.RELEASE'

    testCompile 'org.springframework.boot:spring-boot-starter-test'
    testCompile 'com.h2database:h2'
    testCompile 'org.springframework.security:spring-security-test'
    testCompile 'org.powermock:powermock-module-junit4:1.6.5'
    testCompile 'org.powermock:powermock-api-mockito:1.6.5'

}
```

依赖是树形结构

反思&抱怨

#4 排查第三方组件安全漏洞工作量大，过程太痛苦

你以为就这几个依赖?

```
dependencies {
    compile 'org.springframework.boot:spring-boot-starter-web'
    compile 'org.springframework.boot:spring-boot-starter-data-jpa'
    compile 'org.postgresql:postgresql:9.4.1212'
    compile 'org.apache.poi:poi:3.15'
    compile 'org.apache.poi:poi-ooxml:3.15'
    compile 'io.springfox:springfox-swagger2:2.6.1'
    compile 'io.springfox:springfox-swagger-ui:2.6.1'
    compile 'commons-io:commons-io:2.4'
    compile 'org.apache.commons:commons-lang3:3.5'
    compile 'com.yunpian.sdk:yunpian-java-sdk:1.2.2'
    compile 'org.springframework.boot:spring-boot-starter-security:1.3.0.RC1'
    compile 'io.jsonwebtoken:jjwt:0.7.0'
    compile 'org.json:json:20160810'
    compile 'org.quartz-scheduler:quartz:2.2.3'
    compile 'org.springframework:spring-context-support:4.1.6.RELEASE'

    testCompile 'org.springframework.boot:spring-boot-starter-test'
    testCompile 'com.h2database:h2'
    testCompile 'org.springframework.security:spring-security-test'
    testCompile 'org.powermock:powermock-module-junit4:1.6.5'
    testCompile 'org.powermock:powermock-api-mockito:1.6.5'

}
```

依赖是树形结构

三大安全挑战

#1 一次性的安全检查无法匹配持续性的交付

#2 缺乏自动化、自助化的安全服务

#3 部门墙阻碍开发和安全团队高效协作

挑战1： 一次性的安全检查无法匹配持续性的交付

传统安全控制措施

挑战1： 一次性的安全检查无法匹配持续性的交付

传统安全控制措施

持续交付模式下，开发团队尽可能的降低Cycle Time

挑战1： 一次性的安全检查无法匹配持续性的交付

传统安全控制措施

在这里做渗透测试?

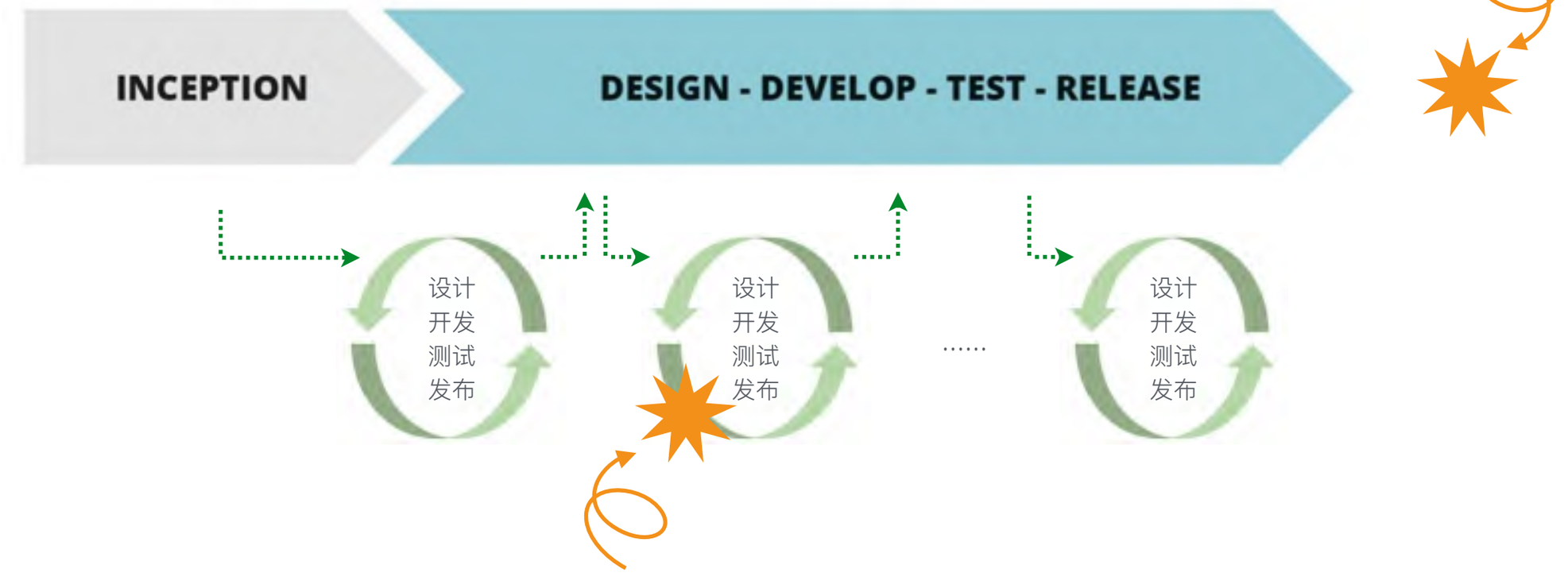
持续交付模式下，开发团队尽可能的降低Cycle Time

挑战1： 一次性的安全检查无法匹配持续性的交付

传统安全控制措施

在这里做渗透测试?

持续交付模式下，开发团队尽可能的降低Cycle Time



在这里做渗透测试?

我们做出的改变

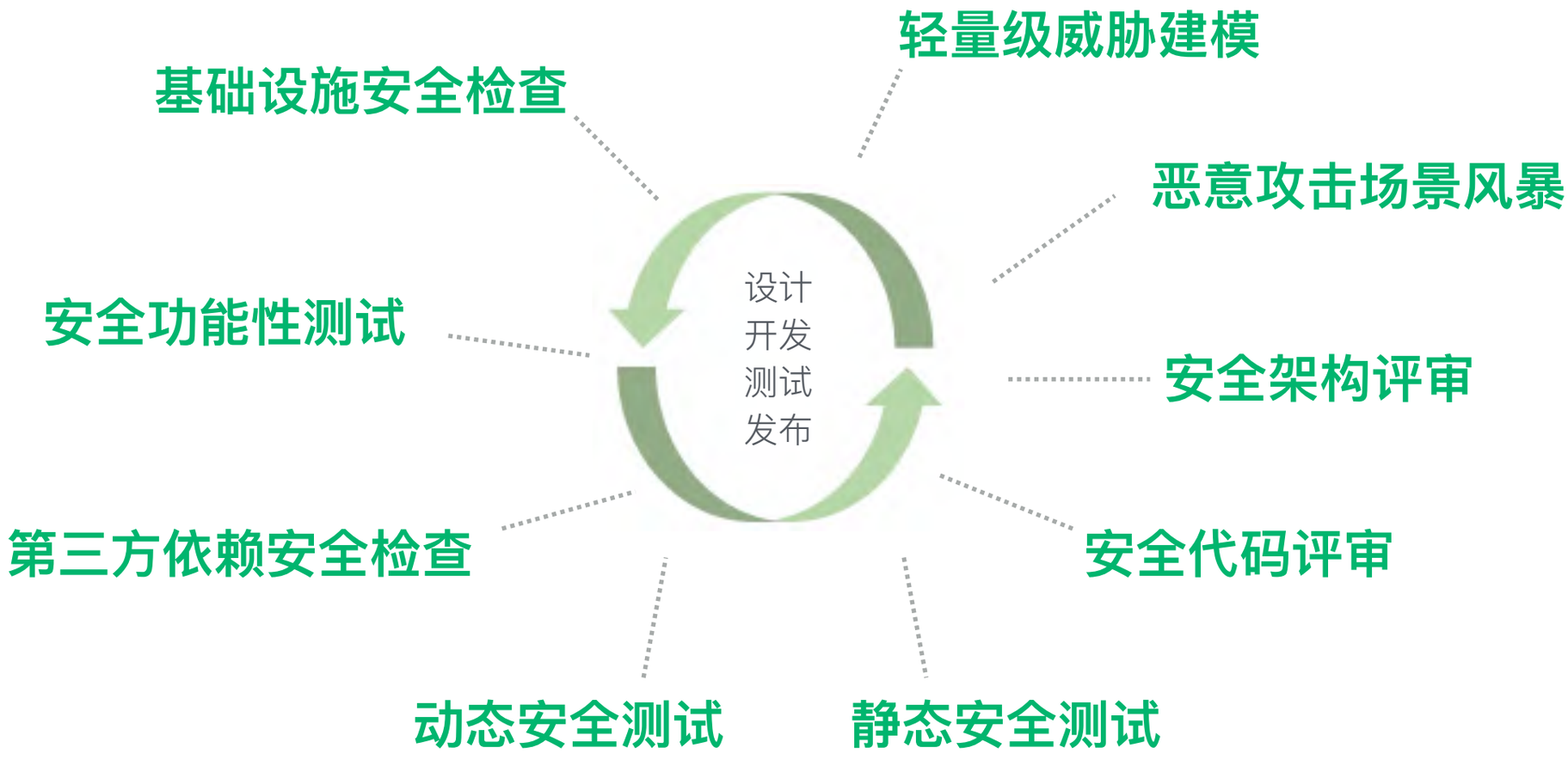
核心原则

不再依赖一次性安全检查，

而是将安全实践融入持续交付流程，

通过持续性的收集安全质量反馈，及时对应用做出调整。

我们做出的改变 - 实践


基础设施安全检查
轻量级威胁建模
恶意攻击场景风暴
安全功能性测试
设计
开发
测试
发布
安全架构评审
第三方依赖安全检查
安全代码评审
动态安全测试
静态安全测试

威胁建模

Threat Modeling

Threat modeling is an approach for analysing the security of an application. It is a structured approach that enables you to identify, quantify, and address the security risks associated with an application.

我们做出的改变 - 威胁建模

威胁建模产出物转化为安全故事卡

[Security] Prevent the signed in session from being used by unauthorized users #473

(v23 - Latest version, last modified 9 months ago)

Security + |

Checklist +

Description

攻击场景

Story Phrase

As a hacker

I can steal the the session tokens of signed in user somehow, then from my own network environment I can access data and functions of [redacted] which I'm not authorized

Below is one of the technical solutions can be adopted as a **mitigation**,This is also the **mitigation** for the flaw that we are unable to immediately invalidate the session tokens on server side when remote client signs out

对应技术方案

Bind the remote client specific information(ip address and User-Agent of browser) to session tokens(cook[redacted]m to access [redacted] unless the hacker

(1) uses the same IP address and

(2) uses the same browser and

(3) the tokens doesn't expire.

Assumptions

It is acceptable that after signing into [redacted]nd before session expires if remot[redacted]uch as disconnect and reconnect to network so ip is changed by DHCP) the re-sign in is required to go on with the operations in [redacted]

验收标准

Acceptance Criteria

AC01

(1) if remote client's ip address is changed for existing valid session [redacted]hould deny the access request to protected resource till re-sign in.

(2) if remote client's browser is changed for existing valid session(obviously it is abnormal behavior,90% hacker attack) [redacted]hould deny the access request to protected resource till re-sign in

我们做出的改变 - 威胁建模的注意事项

轻量级 威胁建模

Lightweight Threat Modeling

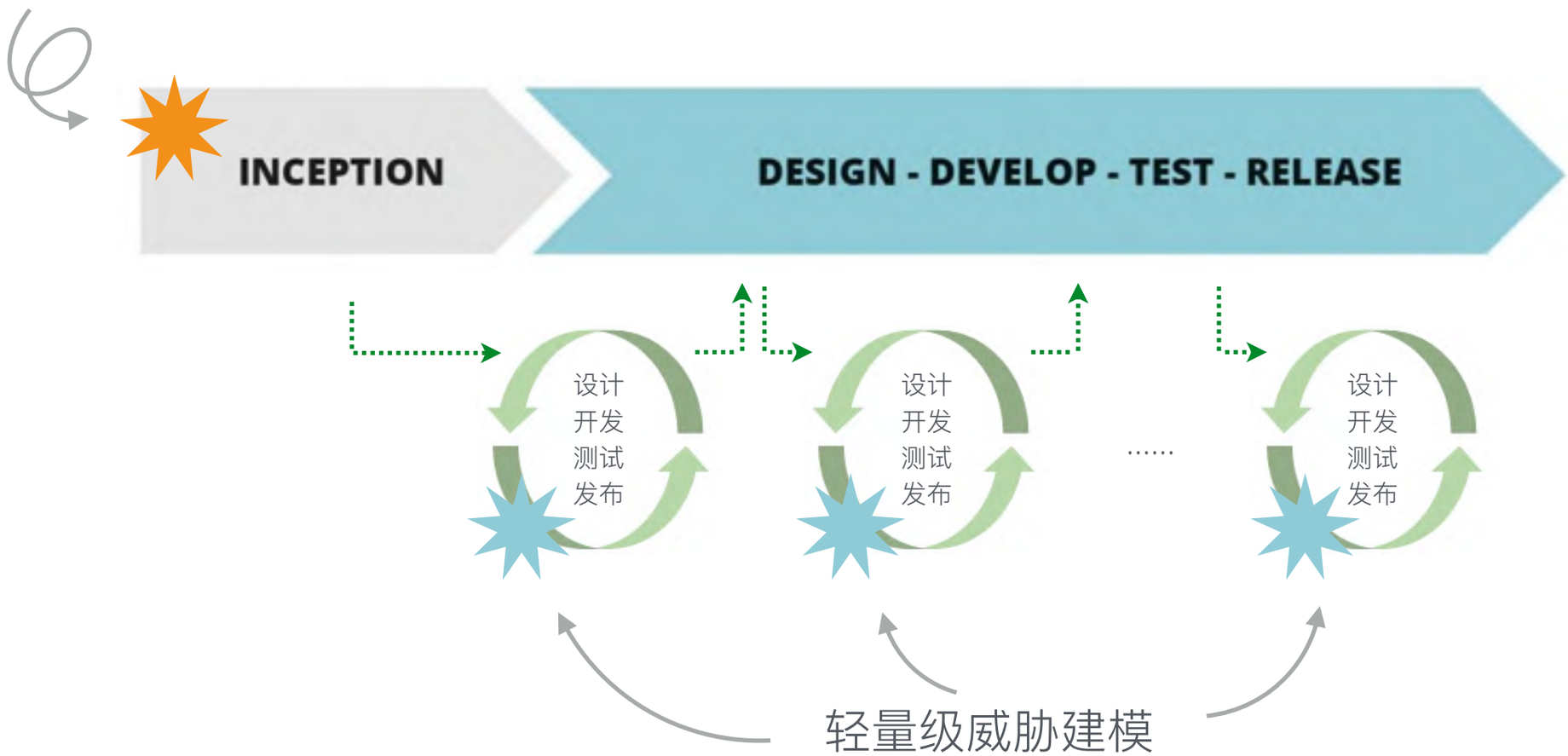
一次性威胁建模

我们做出的改变 - 威胁建模的注意事项

轻量级 威胁建模

Lightweight Threat Modeling

一次性威胁建模



我们做出的改变 - 安全功能性测试

通过自动化测试用例，对应用的安全功能进行测试

API	期待的安全行为
/customers	- 必须经过身份认证后才允许访问 - 只允许具有CustomerManager角色的用户访问 - 只应返回该CustomerManager所管辖区域的客户数据 ……

我们做出的改变 - 安全功能性测试

通过自动化测试用例，对应用的安全功能进行测试

API	期待的安全行为
/customers	- 必须经过身份认证后才允许访问 - 只允许具有CustomerManager角色的用户访问 - 数据

```java
@Test
public void should_not_return_all_the_customers() throws Exception {
    String token = loginAsDefaultCustomer();

    given().header("Authorization", token)
            .when().get("/customers")
            .then().body(empty())
            .statusCode(SC_NOT_FOUND);
}

@Test
public void should_return_information_of_current_customer() throws Exception {
    String tokenOfCustomer1 = loginAsCustomer(authData.customer1, authData.pwdOfCustomer1);
    String customerId = given().header("Authorization", tokenOfCustomer1)
            .when().get(format("/customers/%s", authData.customer1))
            .then().statusCode(SC_OK).extract().path("customerId");
    assertThat(customerId, is(authData.customer1));
}

@Test
public void should_not_return_information_of_other_customer() throws Exception {
    String tokenOfCustomer1 = loginAsCustomer(authData.customer1, authData.pwdOfCustomer1);
    given().header("Authorization", tokenOfCustomer1)
            .when().get(format("/customers/%s", authData.customer2))
            .then().statusCode(SC_UNAUTHORIZED).body("errorKey", is("authorize_failed"));
}
```

挑战2：缺乏自动化、自助化的安全服务

我们知道安全很重要，但是安全实践落地难

核心原则

#1 凡是能自动化的，统统自动化

#2 凡是必须人工参与的，统统进行自助化改造

我们做出的改变


自动化 基础设施安全检查
自助式 轻量级威胁建模
自助式 恶意攻击场景风暴
自动化 安全功能性测试
设计
开发
测试
发布
自助式 安全架构评审
自动化 第三方依赖安全检查
自助式 安全检查
自动化
动态安全测试
自动化
静态安全测试

我们做出的改变 - 自动化第三方依赖安全扫描

以前

1. 通过新闻报道获知组件安全漏洞

2. 手动进行排查

大量人工成本

&

一次性的检查

```
:dependencies

------------------------------------------------------------
Root project
------------------------------------------------------------

archives - Configuration for archive artifacts.
No dependencies

checkstyle - The Checkstyle libraries to be used for this project.
\--- com.puppycrawl.tools:checkstyle:7.6
     +--- antlr:antlr:2.7.7
     +--- org.antlr:antlr4-runtime:4.6
     +--- commons-beanutils:commons-beanutils:1.9.3
     |    \--- commons-collections:commons-collections:3.2.2
     +--- commons-cli:commons-cli:1.3.1
     \--- com.google.guava:guava:19.0

compile - Dependencies for source set 'main'.
+--- org.springframework.boot:spring-boot-starter-web: -> 1.5.1.RELEASE
|    +--- org.springframework.boot:spring-boot-starter:1.5.1.RELEASE
|    |    +--- org.springframework.boot:spring-boot:1.5.1.RELEASE
|    |    |    +--- org.springframework:spring-core:4.3.6.RELEASE
|    |    |    |    \--- commons-logging:commons-logging:1.2
|    |    |    \--- org.springframework:spring-context:4.3.6.RELEASE
|    |    |         +--- org.springframework:spring-aop:4.3.6.RELEASE
|    |    |         |    +--- org.springframework:spring-beans:4.3.6.RELEASE
|    |    |         |    |    \--- org.springframework:spring-core:4.3.6.RELEASE (*)
|    |    |         |    \--- org.springframework:spring-core:4.3.6.RELEASE (*)
|    |    |         +--- org.springframework:spring-beans:4.3.6.RELEASE (*)
|    |    |         +--- org.springframework:spring-core:4.3.6.RELEASE (*)
|    |    |         \--- org.springframework:spring-expression:4.3.6.RELEASE
|    |    |              \--- org.springframework:spring-core:4.3.6.RELEASE (*)
|    |    +--- org.springframework.boot:spring-boot-autoconfigure:1.5.1.RELEASE
|    |    |    \--- org.springframework.boot:spring-boot:1.5.1.RELEASE (*)
|    |    +--- org.springframework.boot:spring-boot-starter-logging:1.5.1.RELEASE
|    |    |    +--- ch.qos.logback:logback-classic:1.1.9
|    |    |    |    +--- ch.qos.logback:logback-core:1.1.9
|    |    |    |    \--- org.slf4j:slf4j-api:1.7.22
|    |    |    +--- org.slf4j:jcl-over-slf4j:1.7.22
|    |    |    |    \--- org.slf4j:slf4j-api:1.7.22
|    |    |    +--- org.slf4j:jul-to-slf4j:1.7.22
|    |    |    |    \--- org.slf4j:slf4j-api:1.7.22
|    |    |    \--- org.slf4j:log4j-over-slf4j:1.7.22
|    |    |         \--- org.slf4j:slf4j-api:1.7.22
|    |    +--- org.springframework:spring-core:4.3.6.RELEASE (*)
|    |    \--- org.yaml:snakeyaml:1.17
|    +--- org.springframework.boot:spring-boot-starter-tomcat:1.5.1.RELEASE
|    |    +--- org.apache.tomcat.embed:tomcat-embed-core:8.5.11
|    |    +--- org.apache.tomcat.embed:tomcat-embed-el:8.5.11
|    |    \--- org.apache.tomcat.embed:tomcat-embed-websocket:8.5.11
|    |         \--- org.apache.tomcat.embed:tomcat-embed-core:8.5.11
|    +--- org.hibernate:hibernate-validator:5.3.4.Final
|    |    +--- javax.validation:validation-api:1.1.0.Final
|    |    +--- org.jboss.logging:jboss-logging:3.3.0.Final
|    |    \--- com.fasterxml:classmate:1.3.1 -> 1.3.3
|    +--- com.fasterxml.jackson.core:jackson-databind:2.8.6
|    |    +--- com.fasterxml.jackson.core:jackson-annotations:2.8.0
|    |    \--- com.fasterxml.jackson.core:jackson-core:2.8.6
|    +--- org.springframework:spring-web:4.3.6.RELEASE
```

我们做出的改变 - 自动化第三方依赖安全扫描

以前

1. 通过新闻报道获知组件安全漏洞
2. 手动进行排查

大量人工成本

&

一次性的检查

现在

全程自动化

OWASP DependencyCheck

Node Security Platform ...

分分钟获得扫描结果

&

持续监控

我们做出的改变 - 自动化第三方依赖安全扫描

OWASP DependencyCheck Report

DEPENDENCY-CHECK

Dependency-Check is an open source tool performing a best effort analysis of 3rd party dependencies; false positives and false negatives may exist in the analysis performed by the tool. Use of the tool and the reporting provided constitutes acceptance for use in an AS IS condition, and there are NO warranties, implied or otherwise, with regard to the analysis or its use. Any use of the tool and the reporting provided is at the user's risk. In no event shall the copyright holder or OWASP be held liable for any damages whatsoever arising out of or in connection with the use of this tool, the analysis performed, or the resulting report.

How to read the report | Suppressing false positives | Getting Help: google group | github issues

Project: xxxxxxxx

Scan Information (show all):

- *dependency-check version*: 1.4.5
- *Report Generated On*: May 9, 2017 at 11:35:22 +08:00
- *Dependencies Scanned*: 100 (100 unique)
- *Vulnerable Dependencies*: 9
- *Vulnerabilities Found*: 21
- *Vulnerabilities Suppressed*: 0
- ...

Display: Showing Vulnerable Dependencies (click to show all)

Dependency	CPE	GAV	Highest Severity	CVE Count	CPE Confidence	Evidence Count
logback-classic-1.1.9.jar	cpe:/a:logback:logback:1.1.9	ch.qos.logback:logback-classic:1.1.9	High	1	LOW	18
logback-core-1.1.9.jar	cpe:/a:logback:logback:1.1.9	ch.qos.logback:logback-core:1.1.9	High	1	LOW	18
jackson-annotations-2.8.0.jar	cpe:/a:fasterxml:jackson:2.8.0	com.fasterxml.jackson.core:jackson-annotations:2.8.0	Medium	1	LOW	25
jackson-core-2.8.6.jar	cpe:/a:fasterxml:jackson:2.8.6	com.fasterxml.jackson.core:jackson-core:2.8.6	Medium	1	LOW	25
jackson-databind-2.8.6.jar	cpe:/a:fasterxml:jackson:2.8.6	com.fasterxml.jackson.core:jackson-databind:2.8.6	Medium	1	LOW	25
jjwt-0.7.0.jar	cpe:/a:sonatype:nexus:0.7.0	io.jsonwebtoken:jjwt:0.7.0	High	1	LOW	18
tomcat-embed-core-8.5.11.jar	cpe:/a:apache:tomcat:8.5.11	org.apache.tomcat.embed:tomcat-embed-core:8.5.11	High	7	HIGHEST	16
tomcat-embed-websocket-8.5.11.jar	cpe:/a:apache:tomcat:8.5.11	org.apache.tomcat.embed:tomcat-embed-websocket:8.5.11	High	7	HIGHEST	18
spring-boot-starter-data-jpa-1.5.1.RELEASE.jar	cpe:/a:pivotal_software:spring_data_jpa:1.5.1	org.springframework.boot:spring-boot-starter-data-jpa:1.5.1.RELEASE	Medium	1	LOW	20

我们做出的改变 - 自动化基础设施安全检查

Docker 容器安全最佳实践

Host Configuration

- ☐ Keep Docker up to date
- ☐ Audit Docker files and directoires - /var/lib/docker
- ☐ Audit Docker files and directoires - /etc/docker
- ☐ Audit Docker files and directoires - docker.service

……

Docker daemon configuration files

Container Images and Build File

Container Runtime

……

我们做出的改变 - 自动化基础设施安全扫描

Docker bench for security

```
docker run -it --net host --pid host --cap-add audit_control \
    -e DOCKER_CONTENT_TRUST=$DOCKER_CONTENT_TRUST \
    -v /var/lib:/var/lib \
    -v /var/run/docker.sock:/var/run/docker.sock \
    -v /usr/lib/systemd:/usr/lib/systemd \
    -v /etc:/etc --label docker_bench_security \
    docker/docker-bench-security
```

我们做出的改变 - 自动化基础设施安全扫描

Docker bench for security

```
# ------------------------------------------------------------------------------
# Docker Bench for Security v1.3.0
#
# Docker, Inc. (c) 2015-
#
# Checks for dozens of common best-practices around deploying Docker containers in production.
# Inspired by the CIS Docker 1.13 Benchmark.
# ------------------------------------------------------------------------------

Initializing Thu Jan 26 08:58:33 UTC 2017


[INFO] 1 - Host Configuration
[WARN] 1.1  - Create a separate partition for containers
[INFO] 1.2  - Harden the container host
[PASS] 1.3  - Keep Docker up to date
[INFO]      * Using 1.13.0 which is current as of 2017-01-18
[INFO]      * Check with your operating system vendor for support and security maintenance for Docker
[INFO] 1.4  - Only allow trusted users to control Docker daemon
[INFO]      * docker:x:998:ubuntu
[WARN] 1.5  - Audit docker daemon - /usr/bin/docker
[WARN] 1.6  - Audit Docker files and directories - /var/lib/docker
[WARN] 1.7  - Audit Docker files and directories - /etc/docker
[WARN] 1.8  - Audit Docker files and directories - docker.service
[WARN] 1.9  - Audit Docker files and directories - docker.socket
[WARN] 1.10 - Audit Docker files and directories - /etc/default/docker
[INFO] 1.11 - Audit Docker files and directories - /etc/docker/daemon.json
[INFO]      * File not found
[WARN] 1.12 - Audit Docker files and directories - /usr/bin/docker-containerd
[WARN] 1.13 - Audit Docker files and directories - /usr/bin/docker-runc


[INFO] 2 - Docker Daemon Configuration
[WARN] 2.1  - Restrict network traffic between containers
[WARN] 2.2  - Set the logging level
[PASS] 2.3  - Allow Docker to make changes to iptables
[PASS] 2.4  - Do not use insecure registries
[WARN] 2.5  - Do not use the aufs storage driver
[WARN] 2.6  - Configure TLS authentication for Docker daemon
[WARN]      * Docker daemon currently listening on TCP with TLS, but no verification
[INFO] 2.7  - Set default ulimit as appropriate
[INFO]      * Default ulimit doesn't appear to be set
[WARN] 2.8  - Enable user namespace support
[PASS] 2.9  - Confirm default cgroup usage
[PASS] 2.10 - Do not change base device size until needed
[WARN] 2.11 - Use authorization plugin
[WARN] 2.12 - Configure centralized and remote logging
[WARN] 2.13 - Disable operations on legacy registry (v1)
[WARN] 2.14 - Enable live restore
[PASS] 2.15 - Do not enable swarm mode, if not needed
[PASS] 2.16 - Control the number of manager nodes in a swarm (Swarm mode not enabled)
[PASS] 2.17 - Bind swarm services to a specific host interface
[WARN] 2.18 - Disable Userland Proxy
[PASS] 2.19 - Encrypt data exchanged between containers on different nodes on the overlay network
[PASS] 2.20 - Apply a daemon-wide custom seccomp profile, if needed
[PASS] 2.21 - Avoid experimental features in production
```

我们做出的改变 - 持续监控安全质量

把安全测试加入到CI Pipeline!

我们做出的改变 - 自助式安全检查

为团队提供：自动化的工具 / 实用的检查清单

- **Authentication verification**
  - Verify all pages and resources by default require authentication except those specifically intended to be public
  - Verify all authentication controls are enforced on the server side.
  - Verify that account passwords are one way hashed with a salt
  - ……
- **Session management**
- **Access control**
- **Malicious input handling**
- **Output encoding/escaping**
- **Cryptography as rest**
- **Error handling**

……

自助式安全服务的未来

构建包含**安全服务**在内的**数字化交付平台**

我们做出的改变 - 注意有坑

常见误区

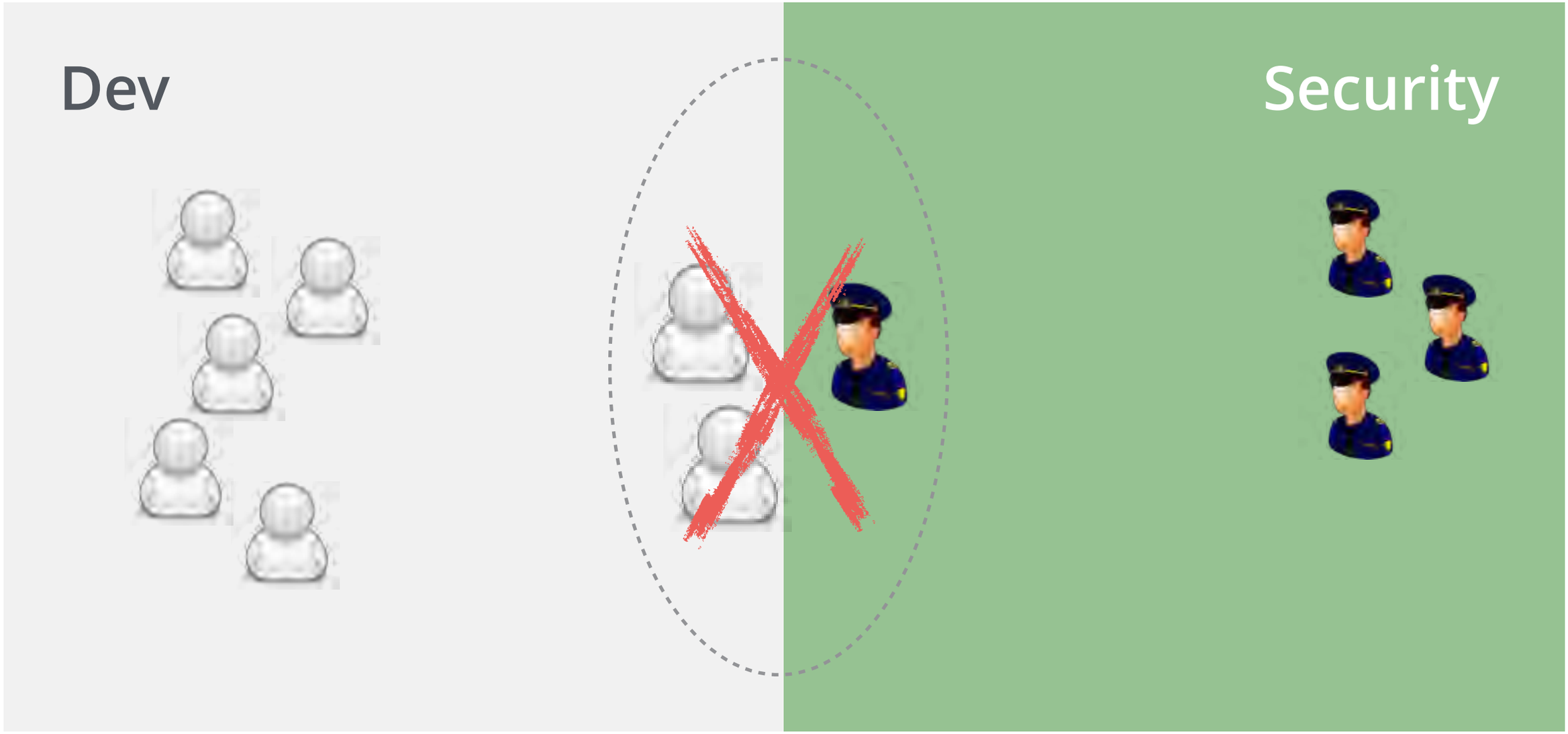
#1 自动化工具无所不能

#2 只要漏扫没报告问题，就可以高枕无忧

我们做出的改变 - 注意有坑

制定安全测试策略

挑战3：部门墙阻碍开发和安全团队高效协作



貌合神离的协作

我们做出的改变

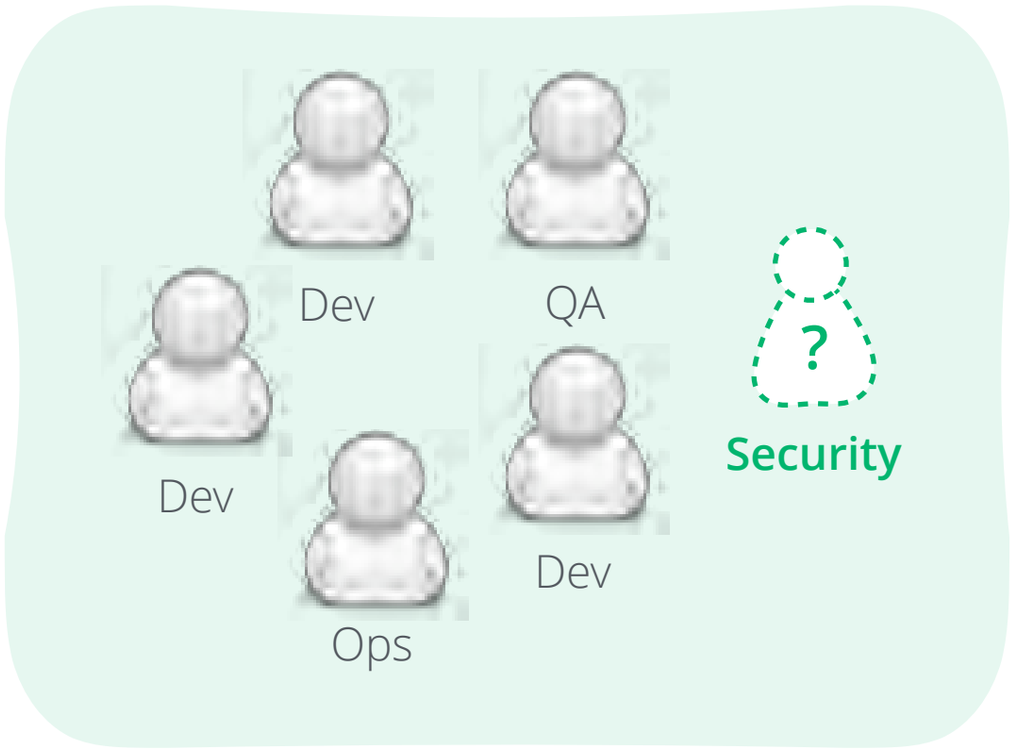
核心原则

真正**高效的沟通协作**，而不是靠**强硬的流程管控**来保证应用安全质量

我们做出的改变

- 建立**包括安全专家在内**的全功能开发团队
- 充分的沟通，设置共同的目标
- 建立“改革特区”

我们做出的改变 - 注意有坑



#1 安全专家是稀缺的

无法做到每个全功能团队均配备一名安全专家

#2 资源浪费

如果安全专家全程跟着项目，会出现工作量不饱和的情况，
安全专家的价值没有被最大程度的发挥出来

我们做出的改变 - 注意有坑

解决办法：安全专家定期Rotation

我们做出的改变 - 注意有坑

不是团队有了安全专家，一切问题都迎刃而解，

关键在于**建立开发团队自己的安全能力。**

小结

三大挑战：

- 一次性的安全检查无法匹配持续性的交付
- 缺乏自动化、自助化的安全服务，安全实践落地难
- 部门墙阻碍开发和安全团队高效协作

小结

解决方案

持续收集安全反馈

自动化、自助化安全服务

破除部门墙高效协作

总结

DevSecurity

通过**转变理念**，运用**安全最佳实践**以及**工具**，

在快速发展和改进产品的同时，

交付具有更高安全质量的应用程序和服务。

THANKS

Q & A

马伟
wma@thoughtworks.com

ThoughtWorks®