自动化运维安全机制的设计原则与实践分享

刘淼

目录

CONTENTS

ITEC 准了么技术社区

IT大咖说 知识分享平台

企业运维的安全现状调查

1
用户账单地址/快递地址/电话/…
账户密码或者Token
账户信息/用户习惯…

2
缓冲区溢出
跨站点脚本攻击
注入式攻击…

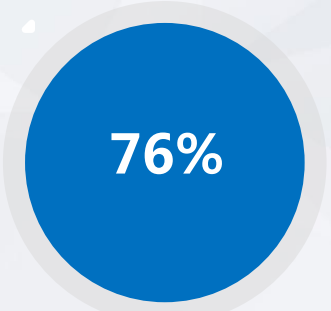
3
拒绝服务Dos
防火墙/病毒库
身份欺骗/撰改

4
系统漏洞/补丁
第三方/开源工具隐患
运维流程漏洞…

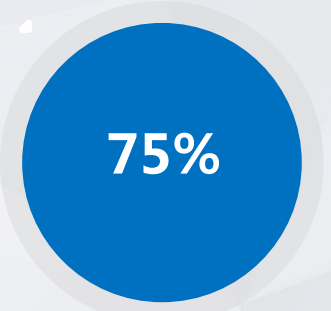
企业运维的安全现状调查

数据丢失的威胁来源

- **恶意软件**
  可以被木马等方式利用进行网络关键信息获取的漏洞
- **钓鱼攻击**
  整合第三方业务服务引入的安全漏洞
- **内部泄露**
  网络设定相关或者容易被黑客进行攻击的安全漏洞
- **人为失误**
  网络设定相关或者容易被黑客进行攻击的安全漏洞



的首席信息安全官将设备失窃或设备上的敏感数据丢失视为严重问题



的首席信息安全官表示云安全预算在不断增加

企业运维的安全现状调查

安全问题现状

- **安全事故**

90%的业务曾发生过安全事故，而且，高达46%的业务由于内部或者外部的安全问题丢失过敏感的数据

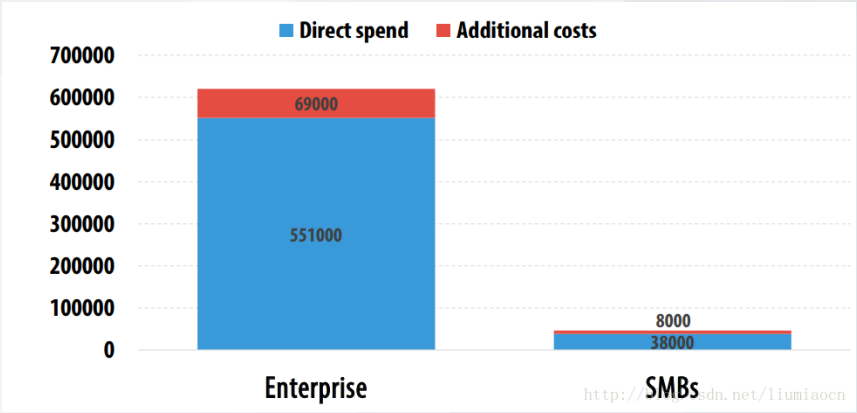
- **直接成本**

大型企业平均每个安全漏洞要付出551,000$的直接成本，而对于中小型企业这个数字是38,000$

- **间接成本**

大型企业平均每个安全漏洞要付出额外的69,000$ 的间接成本, 而对于中小型企业这个数字是8,000$

Kaspersky Lab对26个国家超过5500公司进行了安全相关的调查，结果发现，安全风险无处不在，付出成本相当昂贵。



企业运维的安全现状调查

安全漏洞影响(Top 3)

- **信用危机**

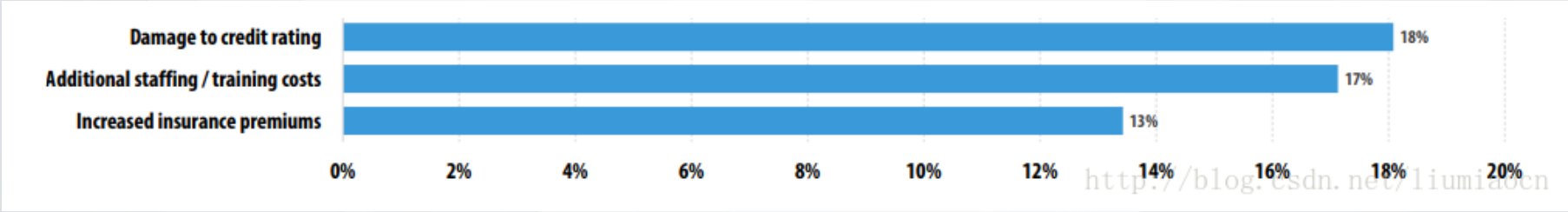
  安全漏洞直接会对对公司信用产生不良影响

- **额外费用**

  安全漏洞产生的额外的人员以及培训等费用

- **业务影响**

  关键业务不能服务或者错误服务导致的额外保险等费用



http://blog.csdn.net/liumiaocn

企业运维的安全现状调查

安全漏洞类型(Top 3)

- **木马漏洞**

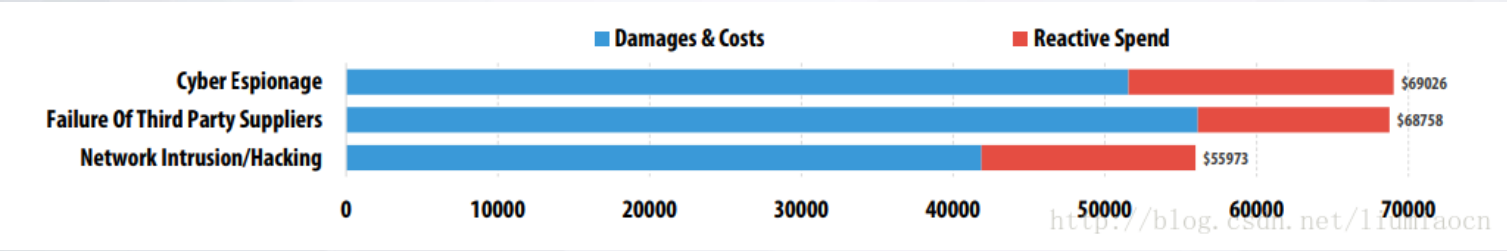
  可以被木马等方式利用进行网络关键信息获取的漏洞

- **应用漏洞**

  整合第三方业务服务引入的安全漏洞

- **网络漏洞**

  网络设定相关或者容易被黑客进行攻击的安全漏洞



目录

CONTENTS

TECH 开了么技术社区

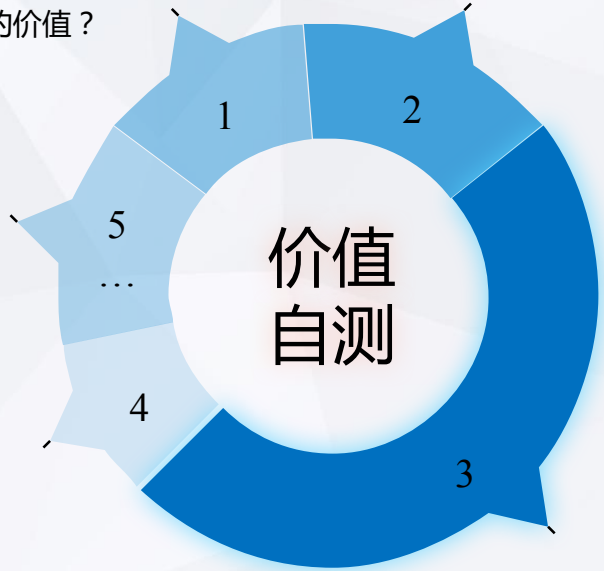
IT大咖说 知识分享平台

安全机制设计的整体策略

策略一：以终为始，分析被攻击的价值所在，制定对应措施

通过你的系统是否有可能接触到大量的用户私密数据，而这些数据在现在的时代具有重要的价值？

通过你的系统是否能接触到用户的信用卡号码和账单地址等？



通过你的系统是否能够接触到很多用户名/密码，这些具有不同权限的用户名和密码是否能给攻击者带来很多价值，比如身份盗用？

通过你的系统是否能接触到用户行为习惯这些隐私性的数据，而这些数据能够使得算法更加聪明？

通过你的系统是否能够接触到进行转账相关的关键性数据？

安全机制设计的整体策略

传统方式：团队分散，安全问题一般有充足时间对应：常见流程

开发团队

- 应用软件的开发和价值的交付

安全团队

- 负责安全保障

运维团队

- 保证服务的可用性和连续性

4-5x

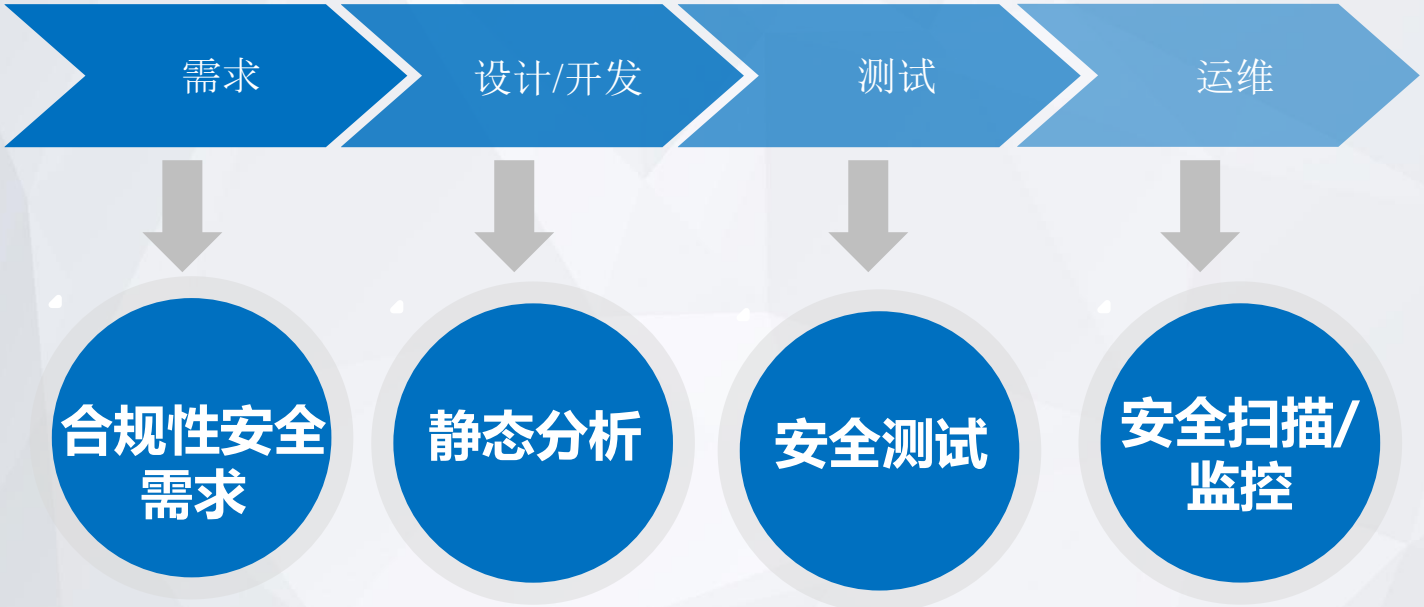
产品发布以后修复安全问题的成本是设计阶段解决成本的4到5倍

>100x

在运维阶段修复安全问题的成本可能达到甚至超出设计阶段解决成本的100倍

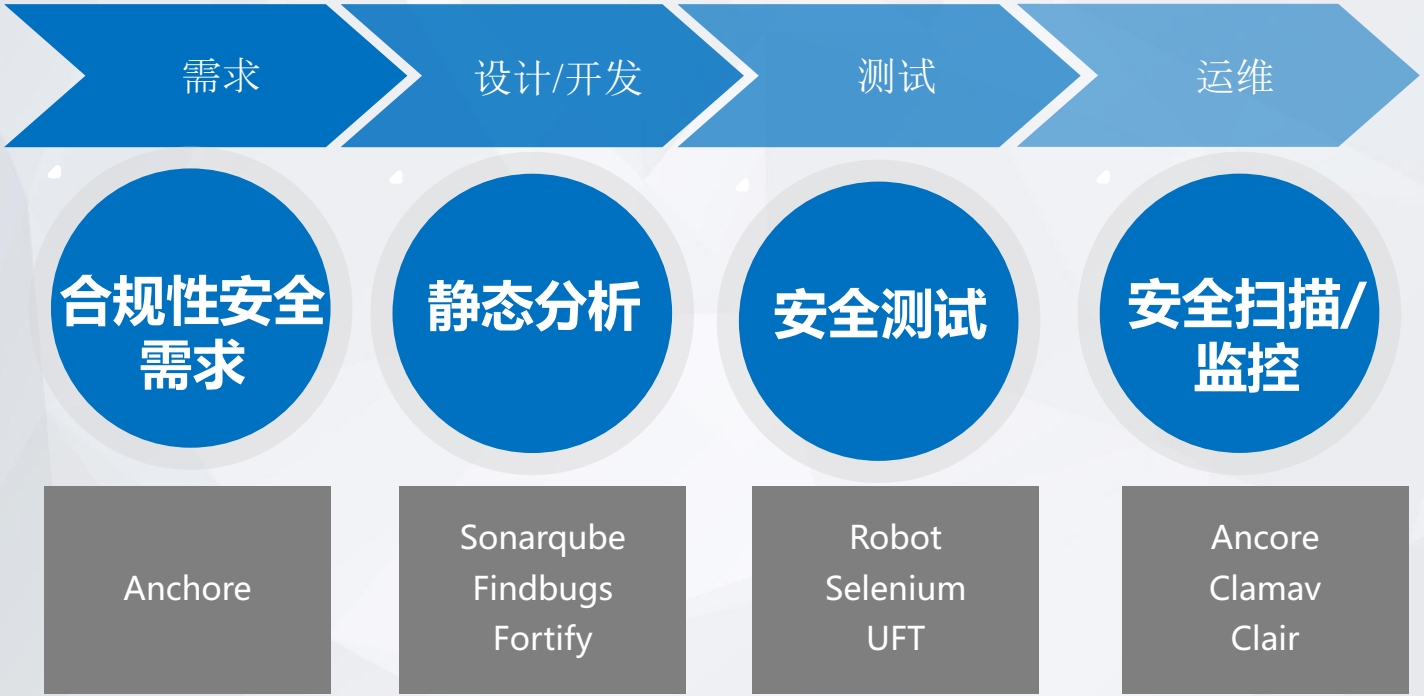
安全机制设计的整体策略

策略二：安全策略的左移，将安全对应融入全生命期



安全机制设计的整体策略

策略三：适应DevOps实践节奏，融合安全工具



安全机制设计的整体策略

- 策略四：平衡业务价值和资产安全，建立以客户为中心的安全策略
- 策略五：持续评估CI/CD的安全状况
- 策略六：创建分阶段全生命周期的安全验收标准
- 策略七：主动监控而不是被动应对
- 策略八：自我攻击和验证
- 策略九：安全的持续评估&安全规则固化

目录

CONTENTS

全阶段安全标准的设定

合规性安全需求

侧重于整体性的业务资源相关以及企业资产安全相关的内容

静态分析

侧重于代码的漏洞或者缺陷

安全测试

系统功能性的正确性和安全相关的测试内容

安全扫描/监控

基础设施和配置方面存在的缺陷和漏洞

建立明确可衡量的安全标准，能够与自动化进行结合自动判定是否符合安全标准

目录

CONTENTS

安全与自动化工具的融合

需求 → 设计/开发 → 测试 → 运维

安全策略设定
黑白名单设定
安全漏洞扫描

合规性 & 安全扫描

Anchore

```
[root@liumiaocn ~]# anchore gate --image docker.io/postgres:latest
33b13ed6b80a: evaluating policies ...
```

Image Id	Repo Tag	Gate	Trigger	Check Output	Gate Action
33b13ed6b80a	docker.io/postgres:latest	DOCKERFILECHECK	FROMSCRATCH	'FROM' container is 'scratch' - (scratch)	GO
33b13ed6b80a	docker.io/postgres:latest	ANCHORESEC	VULNLOW	Low Vulnerability found in package - coreutils (CVE-2016-2781 - https://security-tracker.debian.org/tracker/CVE-2016-2781)	GO
33b13ed6b80a	docker.io/postgres:latest	ANCHORESEC	VULNUNKNOWN	Negligible Vulnerability found in package - login (CVE-2007-5686 - https://security-tracker.debian.org/tracker/CVE-2007-5686)	GO
33b13ed6b80a	docker.io/postgres:latest	ANCHORESEC	VULNUNKNOWN	Negligible Vulnerability found in package - passwd (CVE-2007-5686 - https://security-tracker.debian.org/tracker/CVE-2007-5686)	GO
33b13ed6b80a	docker.io/postgres:latest	ANCHORESEC	VULNMEDIUM	Medium Vulnerability found in package - libxml2 (CVE-2017-9048 - https://security-tracker.debian.org/tracker/CVE-2017-9048)	WARN
33b13ed6b80a	docker.io/postgres:latest	ANCHORESEC	VULNMEDIUM	Medium Vulnerability found in package - libxml2 (CVE-2017-9049 - https://security-tracker.debian.org/tracker/CVE-2017-9049)	WARN
33b13ed6b80a	docker.io/postgres:latest	ANCHORESEC	VULNUNKNOWN	Negligible Vulnerability found in package - python2.7 (CVE-2013-7040 - https://security-tracker.debian.org/tracker/CVE-2013-7040)	GO
33b13ed6b80a	docker.io/postgres:latest	ANCHORESEC	VULNHIGH	High Vulnerability found in package - libsqlite3-0 (CVE-2017-10989 - https://security-tracker.debian.org/tracker/CVE-2017-10989)	STOP

安全与自动化工具的融合

需求 | 设计/开发 | 测试 | 运维

脆弱性扫描
代码质量

静态分析

Sonarqube

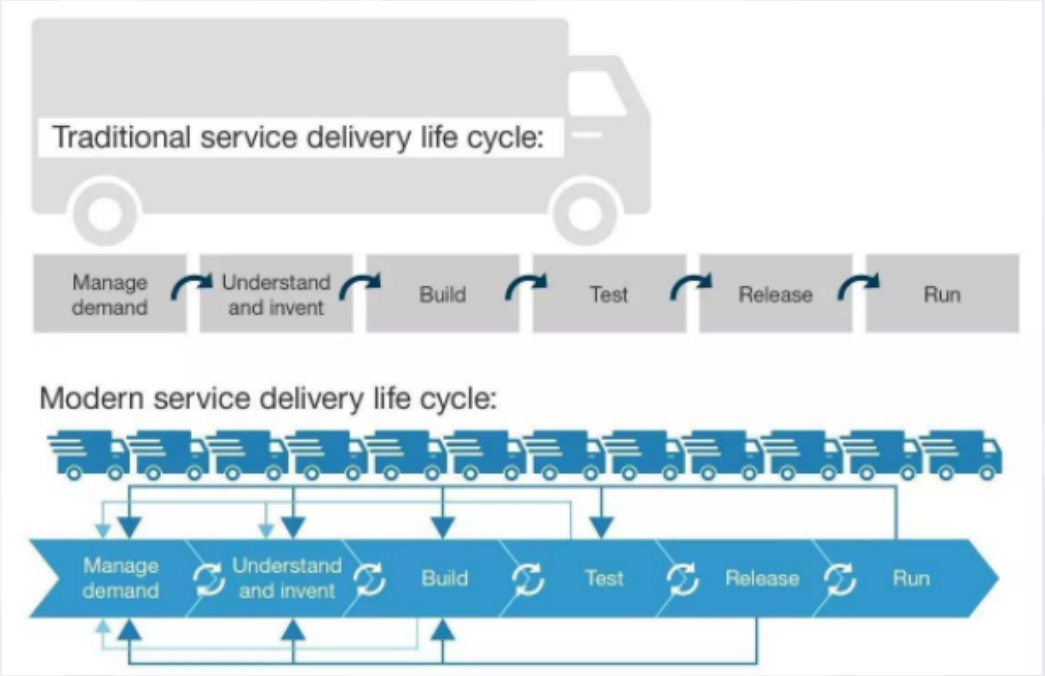
安全与自动化工具的融合

需求 | 设计/开发 | 测试 | 运维

系统功能测试
自动化&接口测试

安全测试

Robot
Selenium
UFT



安全与自动化工具的融合

需求 → 设计/开发 → 测试 → 运维

安全漏洞扫描
镜像安全保证

安全扫描

Clair

CLAIR CONTROL REPORT

Image: nginx

Total : 52 vulnerabilities

Unknown : 4 | Negligible : 25 | Low : 3 | Medium : 16 | High : 4

aced44773d50d1d0bd1d9ae38882e1194cb7e96e2f1a69b42385bcc79c6f9fb7

expat 2.2.0-2+deb9u1 - ⚠

- CVE-2013-0340
  expat 2.1.0 and earlier does not properly handle entities expansion unless an application developer uses the XML_SetEntityDeclHandler function, which allows remote attackers to cause a denial of service (resource consumption), send HTTP requests to intranet servers, or read arbitrary files via a crafted XML document, aka an XML External Entity (XXE) issue. NOTE: it could be argued that because expat already provides the ability to disable external entity expansion, the responsibility for resolving this issue lies with application developers; according to this argument, this entry should be REJECTed, and each affected application would need its own CVE.
  Link

http://blog.csdn.net/liumiaocn

安全与自动化工具的融合

需求 | 设计/开发 | 测试 | 运维

恶意软件扫描
木马病毒监视

安全扫描

Clamav

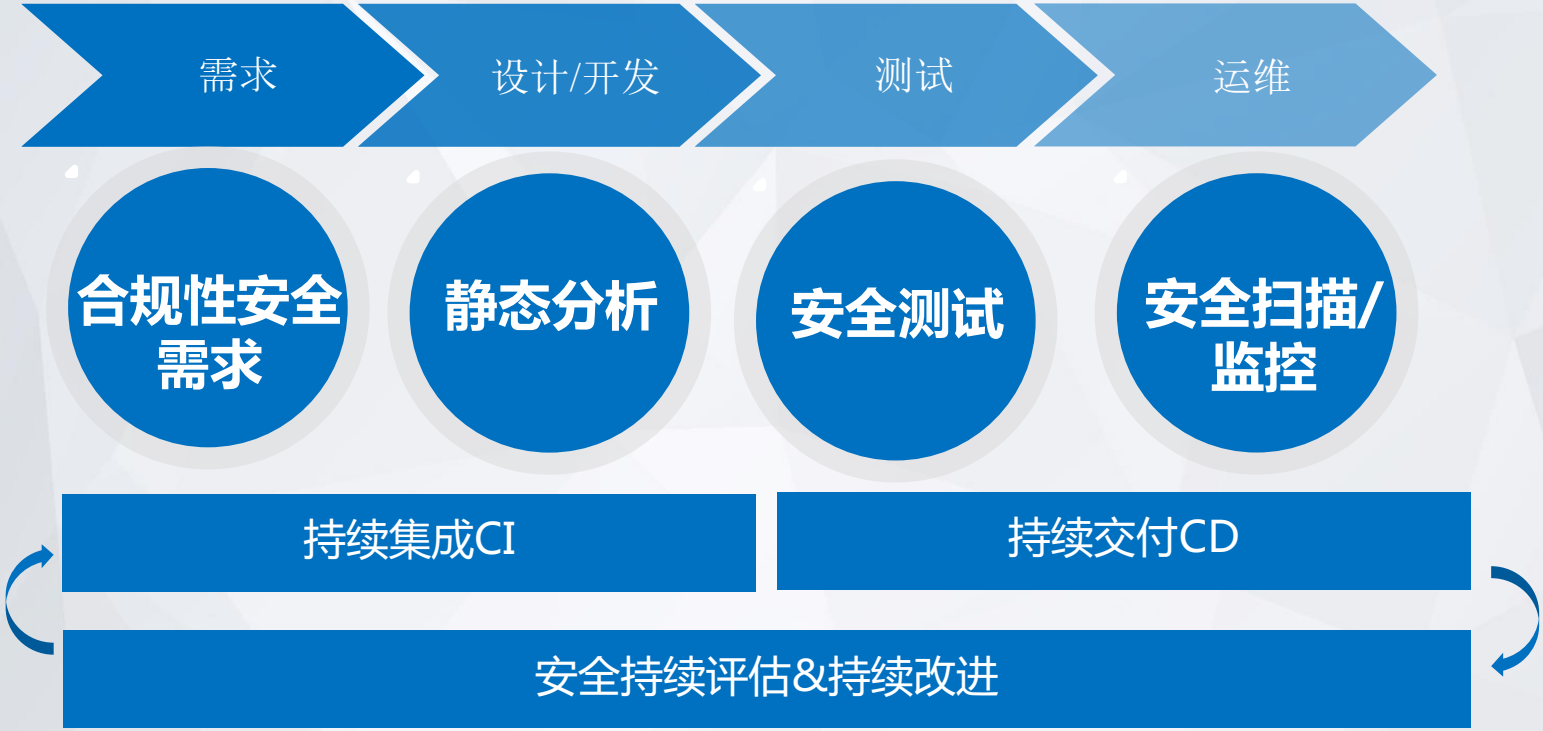
```
[root@liumiaocn ~]# /usr/local/clamav/bin/clamscan --remove /root
/root/.bash_logout: OK
/root/.bash_profile: OK
/root/.bashrc: OK
/root/.cshrc: OK
/root/.tcshrc: OK
/root/anaconda-ks.cfg: OK
/root/.bash_history: OK
/root/eicar.com: Eicar-Test-Signature FOUND
/root/eicar.com: Removed.

----------- SCAN SUMMARY -----------
Known viruses: 6302548
Engine version: 0.99.2
Scanned directories: 1
Scanned files: 8
Infected files: 1
Data scanned: 0.01 MB
Data read: 0.00 MB (ratio 2.00:1)
Time: 22.310 sec (0 m 22 s)
[root@liumiaocn ~]# ls
anaconda-ks.cfg
[root@liumiaocn ~]#
```

目录

CONTENTS

安全机制的持续评估和改善



安全机制的持续评估和改善

- 自测问题1：开发者能够看到其他项目的敏感信息么？
- 自测问题2：各种用户的权限是否清晰，是否不存在越权的风险？
- 自测问题3：密码等敏感信息是否是用明文的形式进行存储？
- 自测问题4：匿名用户会取得过于宽松权限比如可以执行所有项目的脚本？
- 自测问题5：构建的机制是否容易或者可能被攻击?

安全机制的持续评估和改善

- 自测问题6：开发者是否可能轻易地删除其他项目的一些信息？
- 自测问题7：是否使用了一些其他的不安全的服务或者机制进行CI/CD？
- 自测问题8：是否存在上传脚本等自定义机制引入不安全的因素？
- 自测问题9：安全相关的基线管理是否融入和CI/CD之中？
- ……

目录

CONTENTS

案例分享:银行系统运维安全

需求 → 设计/开发 → 测试 → 运维

系统安全	系统资源监控	机器死活监控
	集群状态监控	灾备中心监控
	异常登陆监控	用户操作监控
业务安全	业务进程监控	外接进程监控
	业务峰值监控	迟延预防监控
	…	…

安全监控

案例分享:银行系统运维安全

需求 | 设计/开发 | 测试 | 运维

1	警告信息分级，根据运维手册保证所有提示信息都有对应流程和时限
2	生产环境操作需要保证在准生产环境中验证通过，紧急对应需要结对对应
3	用户统一管理，需要在生产环境手工作业的用户账号都是事前申请，临时生成，操作记录

监控对应

案例分享:银行系统运维安全

需求 | 设计/开发 | 测试 | 运维

1	数据使用不同策略保存在不同媒介上，多种冗余方式保证硬件损害不至于数据丢失
2	定期全量数据备份结合时时查分数据备份保证系统性能
3	建立生产环境和测试环境数据自动脱敏机制，保障安全和效率

数据安全

案例分享:银行系统运维安全

需求 → 设计/开发 → 测试 → 运维

1	定期更新系统补丁和安全漏洞
2	部署上线之前必须多层审核以及结果确认，并列出所有可能出现的问题及对策
3	定期培训以及灾害训练实施

日常操作

目录

CONTENTS

DevOps标准化提议

是可以可能
标准化的麼

目录

CONTENTS

ITECH 淮了么技术社区

IT大咖说 知识分享平台