SDL探索之路

晴天安好2997 / 2019-10-29 09:14:21 / 浏览数 5267 技术文章 企业安全 顶(1) 踩(0)

SDL探索之路
SDL路阻且艰，但是却是一条必走之路。
目前就职于一家甲方公司，平时的主要工作是做安全渗透，偶尔也会做一下代码审计。总的来说流程很固定，每天重复差不多同样的工作。时间长了，也慢慢在思考一些问是
自2004年起，微软将SDL作为全公司的计划和强制政策，SDL的核心理念就是将安全考虑集成在软件开发的每一个阶段:需求分析、设计、编码、测试和维护。从需求、设计
安全开发生命周期。

通过上图可以看到，微软大致将SDL分为以下几个阶段：培训阶段；需求阶段；设计阶段；实施阶段；验证阶段；发布阶段；相应阶段。可能各个公司根据情况不同会重新定

1. 培训阶段

■■■■■■■■■■■■■■■■■■■■■■■■■(■■C#/C++/JAVA■■■■■■■■■■PHP)■■■■■■■■■■■■■■■
■■
■■■

2. 需求和设计阶段

■■■■SDL■■
■■■
■■“■■”■■■■■■■■■■■■■■■■■■■■
■■■■■■■■■■■■■■■

3. 代码质量扫描阶段

■■■
Jenkins■■■■■■■■■■■■■Java■■
Sonar■SonarQube■■■■■■■■,■■■Java■PHP■C#■C■■
Pclint■■■C++/C■■■
Findsecbug■■■java■■■■■■■■■Dependency-check■■■■■■■■■■■■■
Sonar■■■■pclint■findbug■dindsecbug■dependency-check■■■■Jenkins■■■■sonar■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■ denpenc

3.1 工具集成

3.1.1 Jenkins集成sonar

(1) Jenkins安装
有两种方式可以实现Jenkins环境搭建。一种是直接去官网下载windows版安装包然后运行，输入密码进入。密码通常在以下目录存储:C:\Program Files (x86)\Jenkins\secrets\initialAdminPassword。
下载地址：http://mirrors.jenkins.io/war-stable/

除此之外还有一种方式，直接运行war包。war下载以后直接运行以下命令即可：

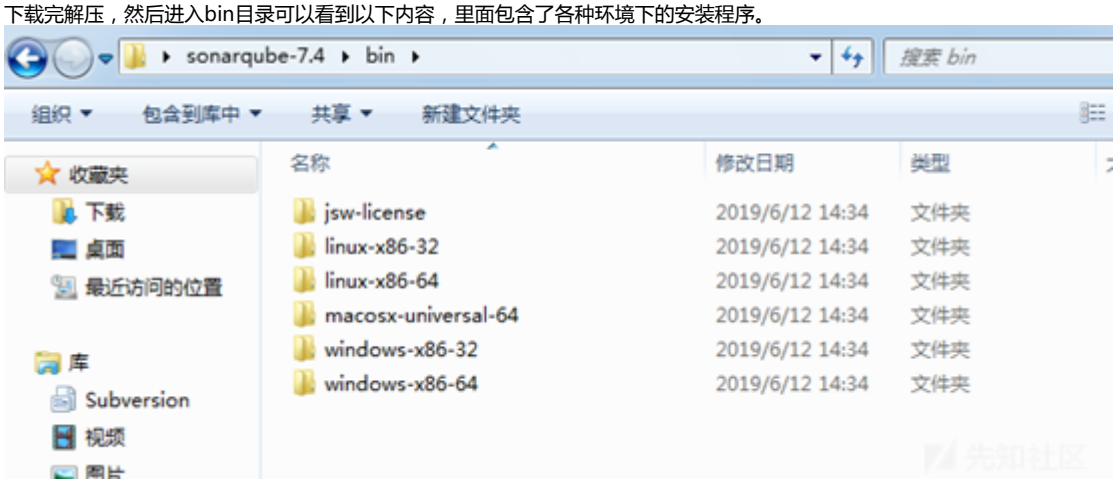
Java -jar jenkins.war --httpPort=8080 。

war包下载地址：http://mirrors.jenkins-ci.org/war/

(2) Sonar安装

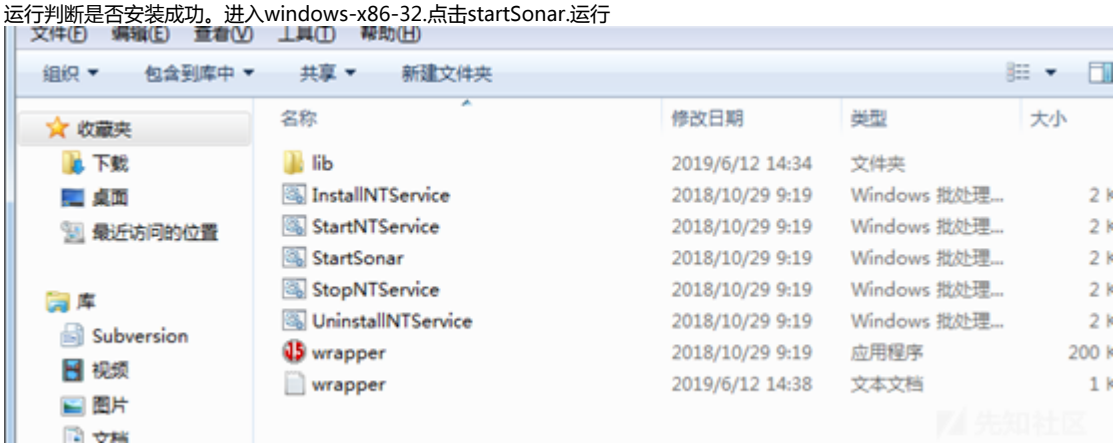
通过sonar官网下载安装包。https://www.sonarqube.org/

下载完解压，然后进入bin目录可以看到以下内容，里面包含了各种环境下的安装程序。



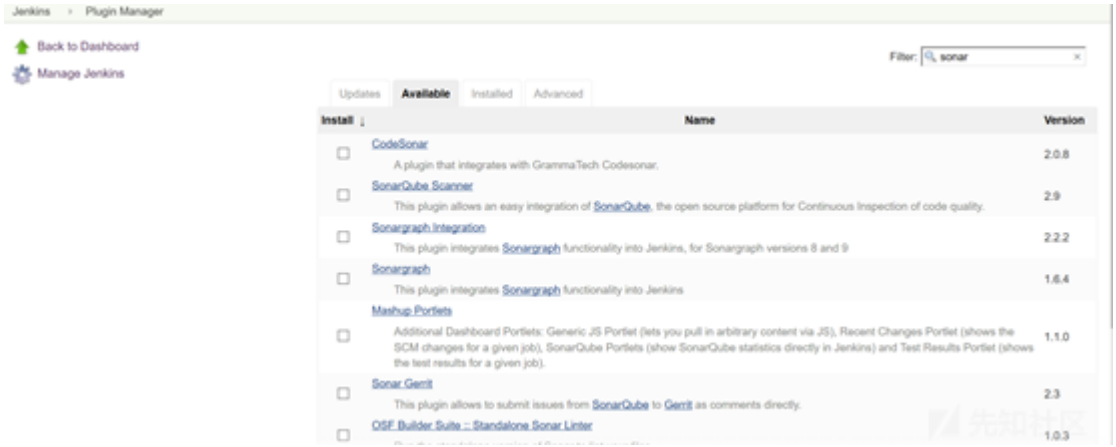
配置数据库，在Conf目录下，sonar.properties文件，配置数据库信息。

运行判断是否安装成功。进入windows-x86-32.点击startSonar.运行

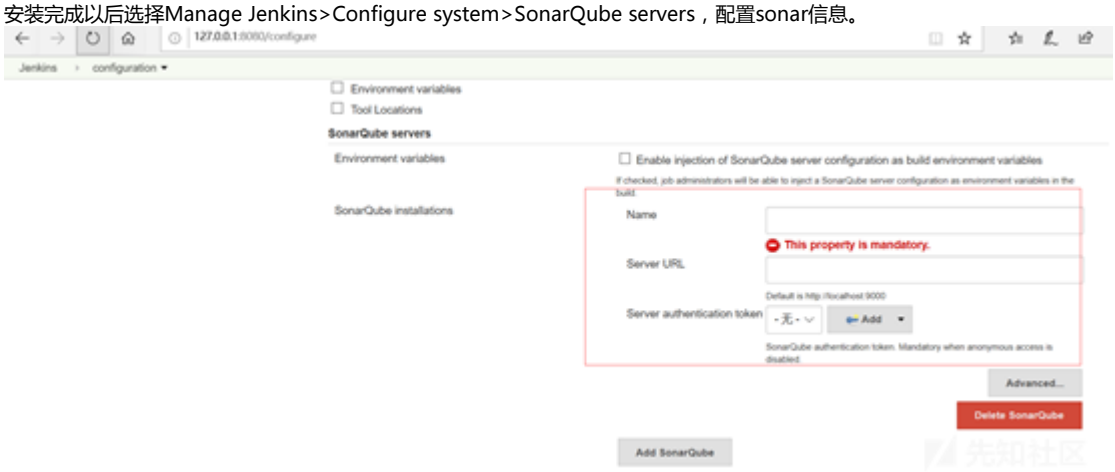


(3)Jenkins集成sonar

首先，需要在Jenkins中安装Sonar插件。进入Manage Jenkins>Plugin Manager，搜索sonar，然后选择SonarQube Scanner插件。



安装完成以后选择Manage Jenkins>Configure system>SonarQube servers，配置sonar信息。



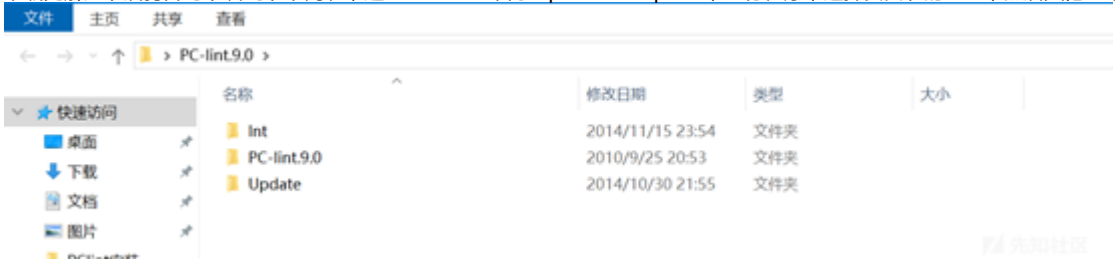
进入Manage Jenkins>Global Tool Configuration配置sonarscanner信息。

至此，已完整将sonar集成到Jenkins。

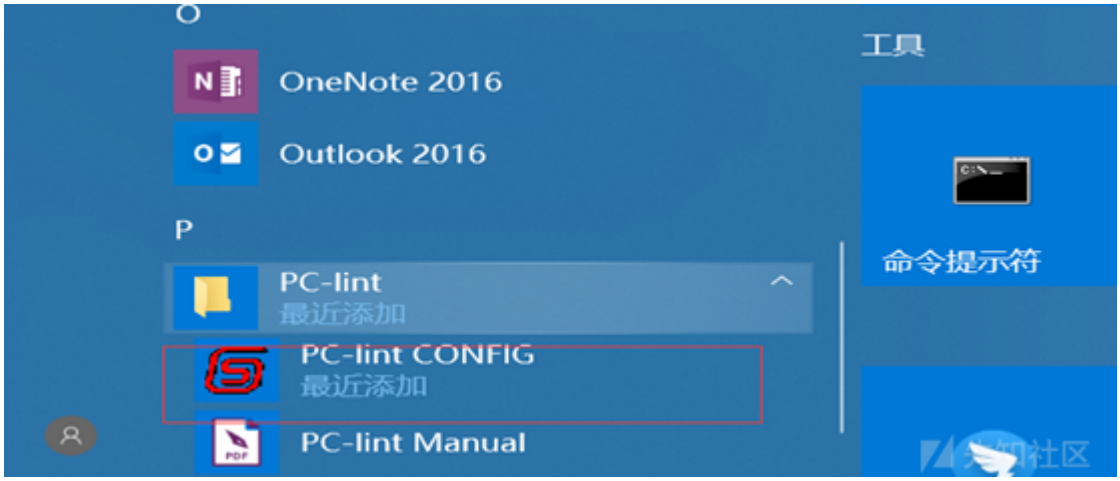
3.1.2 Sonar集成pclint

(1)pclint安装

下载完解压以后打开可以看到以下列表，进入Pc-lint9.0目录- pclint9setup.exe，运行程序，选择要安装的地址，然后其他一路默认即可。



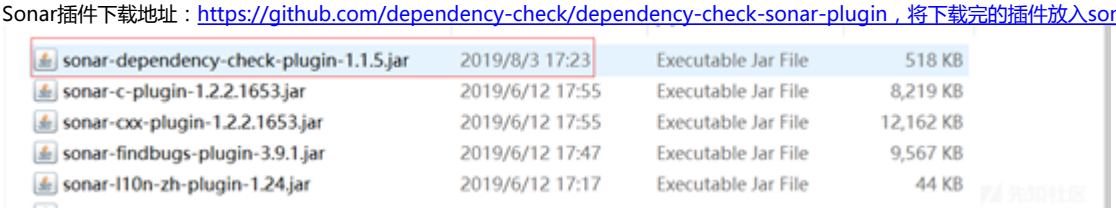
安装完以后，自动进入pclint配置目录。具体配置升级请参考以下链接，按照步骤操作即可：http://www.opdown.com/soft/69530.html。



需要说明的是pclint配置其实就是配置std.lnt，里面主要配置系统函数库地址，要忽略的安全点等。提到函数库就必然会想到windows和linux。针对环境下开发的程序需要配

3.1.3 Sonar集成dependency-check

(1)dependency-check环境搭建
在OWASP官网下载dependency-check程序以及对应的sonar插件。Denpendency-check工具下载地址：https://www.owasp.org/index.php/OWASP_Dependency_Ch
(2) jenkins和sonar集成
集成还需要下载一个插件sonar-dependency-check-plugin。Dependecny负责扫描第三方jar包，而对应的sonar-dependency-check-plugin插件用来进行扫描结果展示。
Sonar插件下载地址：https://github.com/dependency-check/dependency-check-sonar-plugin，将下载完的插件放入sonar的extensions->plugins目录。



至此已经成功的完成了dependency-check的集成搭建工作。

3.1.4 Sonar集成findbugs

Findbugs■■■■■java■■■■■■■■■■■■■■■■findbugs■■■■■■■■sonar■extensions->plugins■■■

3.2 工具扫描

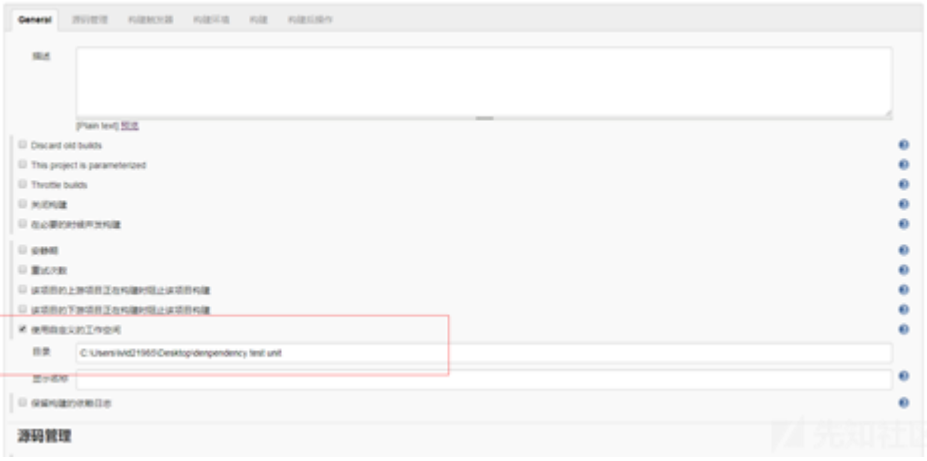
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■

3.2.1 pclint扫描

(1).新建item
新建item-配置,并设置工作目录
(2).构建windows批处理命令
点击构建-execute windows batch
command，执行批处理命令。这里的目的是为了使用调用pclint工具进行项目源代码扫描。如果代码层次较多请参考PClint扫描指南。
Lint-nt -i"e:\pclint\lint" –u std.lnt env-vc10.lnt -v -width(0,0) +xml(results) -"format=<issue file =\q%%f\q line = \q%%l\q number = \q%%n\q desc = \q%%m\q/>" -"format_specific= " -hFs1 -e900 .cpp>Pclint_report.xml
如果是写批处理文件，需要使用如下方式：
Lint-nt -i"e:\pclint\lint" –u std.lnt env-vc10.lnt -v -width(0,0^) +xml(results^) -"format=<issue file =\q%%f\q line = \q%%l\q number = \q%%n\q desc = \q%%m\q/>" -"format_specific= " -hFs1 -e900 .cpp>Pclint_report.xml
(3).构建sonar扫描
配置analysis properties
sonar.projectKey=sectest //该参数获取参考2.1
sonar.projectName=sectest //该参数获取参考2.1
sonar.projectVersion=2.0
sonar.sources=.
sonar.language=c++
sonar.cxx.pclint.reportPath= Pclint_report.xml
(4).运行(build now)
3.2.2 dependency-check扫描
(1).新建item
创建新的项目。

图2 jenkins新建项目

(2).点击项目-进入配置页面：设置工作目录



(3).构建

在构建中选择dependency-check扫描，参数里面添加以下选项，如果没有设置系统将采用默认参数配置。

需要说明：这里面需要设置导出为xml，因为下一步sonar扫描是基于xml进行的。

(4).添加sonar扫描构建

在analysis properties中填写以下信息

sonar.projectKey=denpendencyy //sonar中创建的项目

sonar.projectName=denpendencyy //sonar中创建的项目

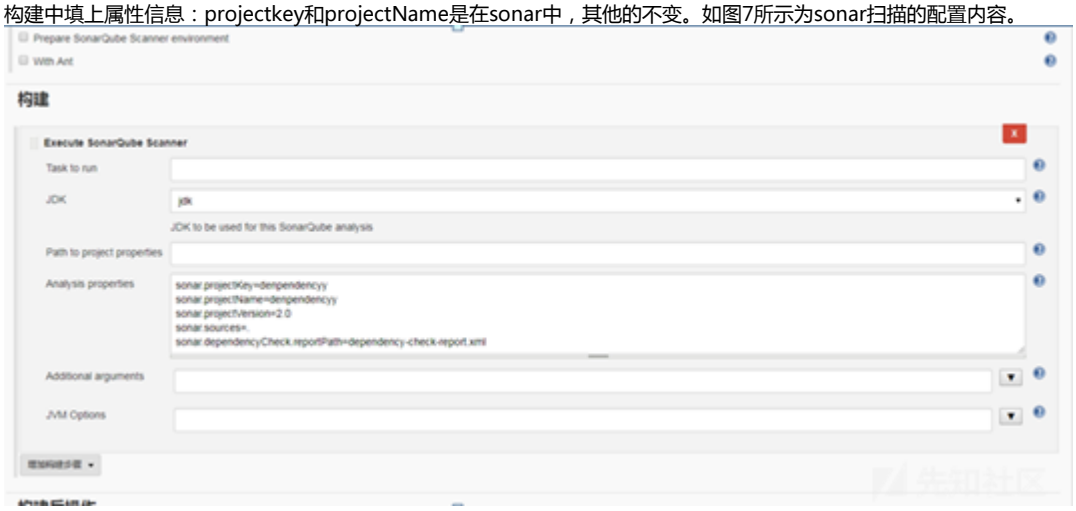
sonar.projectVersion=2.0

sonar.sources=.

sonar.dependencyCheck.reportPath=dependency-check-report.xml

Projectkey和projectname获取参考下图：当创建完项目以后，点击配置-项目找到自己创建的项目信息。

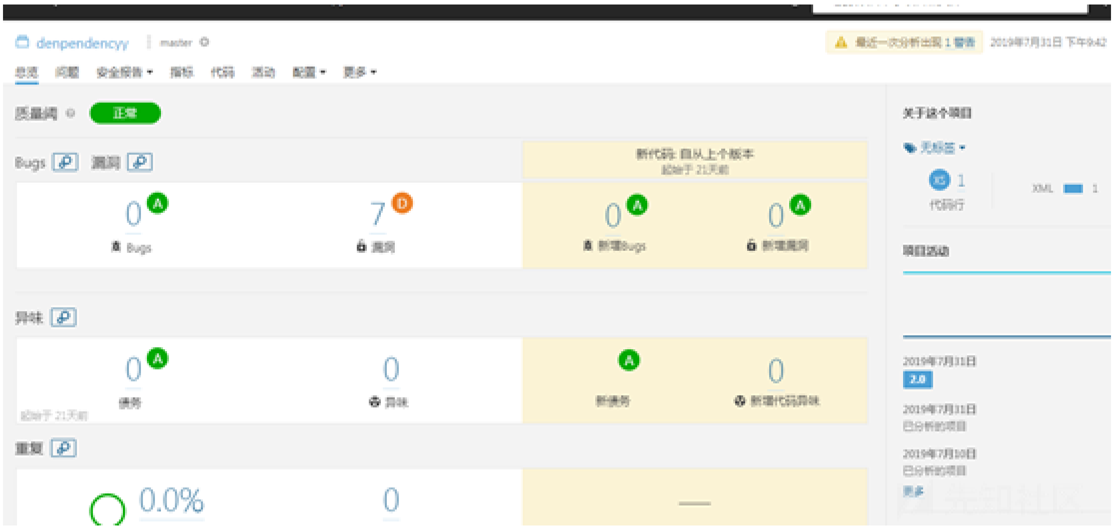
构建中填上属性信息：projectkey和projectName是在sonar中，其他的不变。如图7所示为sonar扫描的配置内容。



(5).build

点击build now运行项目，如图8所示。

运行完以后点击sonarqube，直接进入sonar看到扫描结果，如图9所示。



3.2.3 findbugs扫描

Findbus[redacted]sonar[redacted]

4.安全扫描阶段

[redacted]1[redacted]

4.1 源代码动态扫描

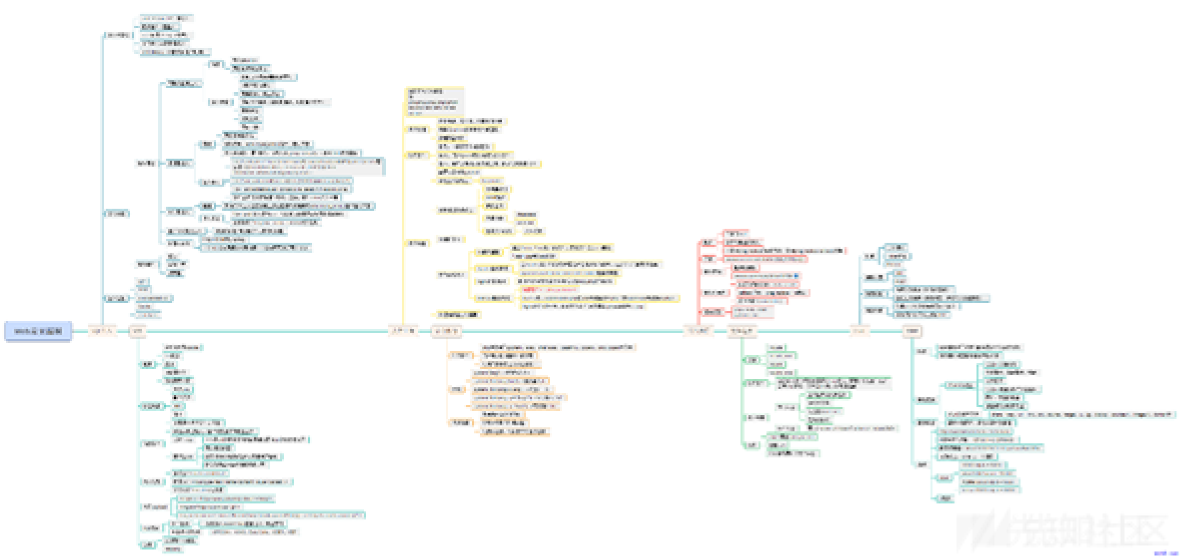
[redacted]fortify[redacted]Fortify SCA[redacted]CoBOT[redacted]PHP[redacted]java[redacted]

4.2 黑盒扫描

[redacted]web[redacted]Android[redacted]PC[redacted]
[redacted]web[redacted]awvs[redacted]appscan[redacted]WVSS[redacted]
[redacted]Android[redacted]360app[redacted]android[redacted]MOBSF[redacted]Django[redacted]web[redacted]
[redacted]PC[redacted]PC[redacted]PC[redacted]
Web[redacted]Github[redacted]

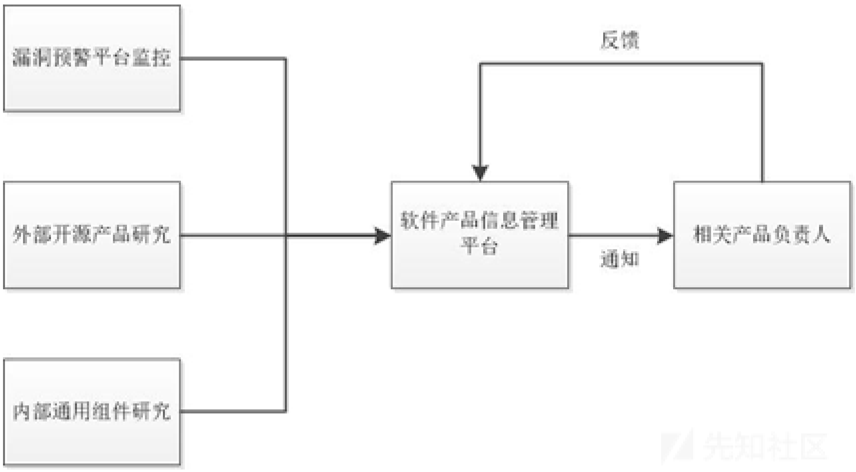
4.3 渗透测试

[redacted]



5.后监控阶段

[redacted]



除此之外，还要及时监控敏感信息泄露的问题。Github是一个开发者广泛使用的平台，众多人员可能不经意间将公司敏感数据上传，因此有必要使用工具及时监控Github。

6 总结

[redacted]

点击收藏 | 1 关注 | 1

[上一篇：BadUsb从协议分析到实战渗透](#) [下一篇：从一道题深入HTTP协议与HTTP...](#)

1. 3 条回复


stay 2019-10-29 11:17:02

老哥能否发一张高清的渗透测试的图？或者给个链接？

0 回复Ta

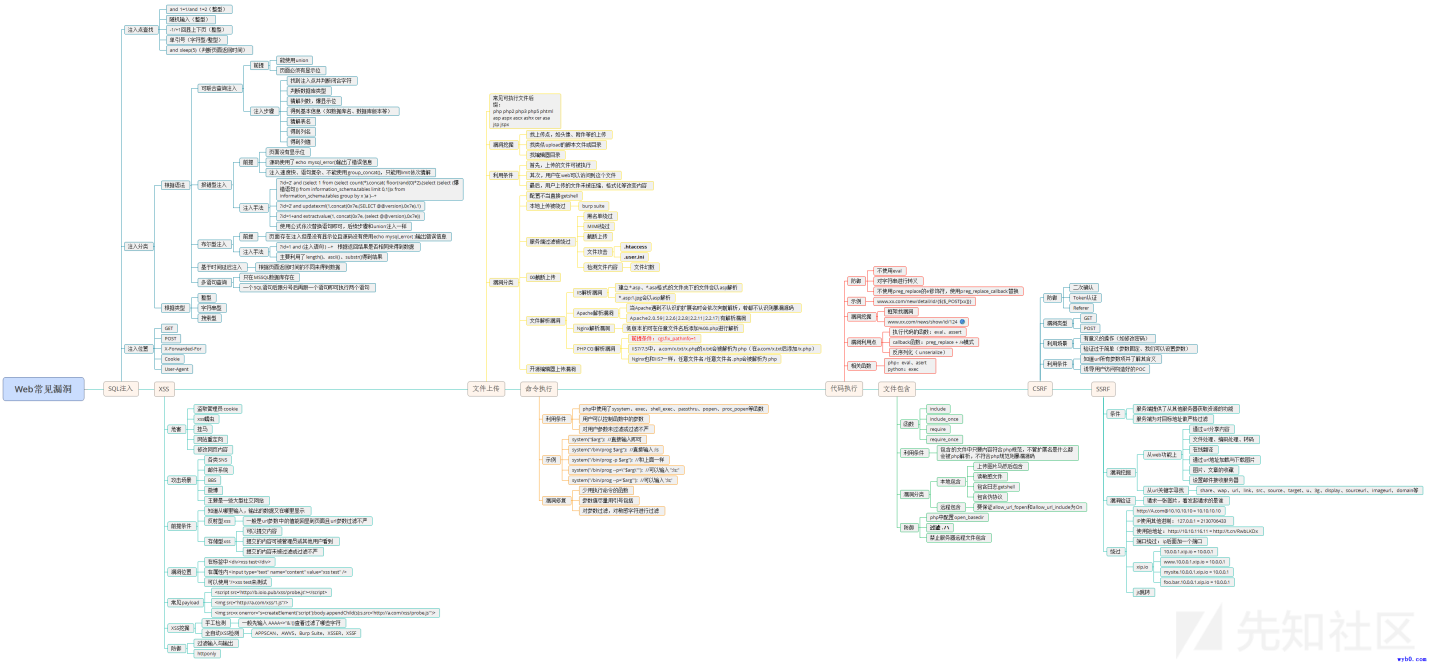

twosmi1e 2019-10-29 16:41:05

这图片也太糊了

1 回复Ta

晴天安好2997 2019-11-11 09:38:53

@stay



0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板