

代码安全扫描服务交付指南（试行）

学校 2017 年起部署了 Fortify 代码安全扫描工具，代码安全是信息安全的基石，除商业成品软件外，代码安全扫描是学校信息系统上线前安全评估的重要内容，但如果只在系统上线前提交代码扫描，有可能导致返工，耽误项目进展，而且线下交付代码增大泄露风险。为此，2019 年起试行自动化扫描，开发人员可以在开发生命周期自主提交代码扫描，跟踪代码安全漏洞和修复结果。

自动化扫描环境简介

自动化扫描环境构成：Fortify SCA + Fortify SSC + Jenkins + Gitlab + Git，除了 Fortify SCA 由网络中心提供，其他均可自主部署。针对不同用户情况，网络中心安全组也部署了完整的自动化扫描测试环境。建议用户自行部署 Jenkins + Gitlab + Git，减少代码交付，降低泄露风险。本文附录提供了自动化构建指南、Fortify 与 Jenkins 集成指南、Jenkins 和 Gitlab 的安装指南，供参考。

环境构成说明：

Fortify SCA：Fortify Source Code Analysis，代码扫描引擎。

Fortify SSC：Fortify Software Security Center，Web 管理平台（<http://10.8.8.50:18080/ssc>），用户可登陆进行代码安全审计和报告下载。

Gitlab：类似 Github，测试环境 URL 为 <http://172.22.2.56>（IP 可能变化），测试环境不保证可用性。

服务交付方式

方式一：用户部署 Jenkins + Gitlab + Git

用户已部署 Jenkins + Gitlab + Git，参考以下步骤：

步骤 1：参考附录《Fortify 与 Jenkins 集成指南》，用户自行完成 Fortify 与 Jenkins 集成。

步骤 2：参考附录《自动化构建指南》，提交扫描任务。

步骤 3：通知安全组扫描的项目名称，安全组为用户分配 Fortify SSC 账号和权限。

方式二：用户部署 Gitlab + Git

用户只部署 Gitlab + Git，参考以下步骤：

步骤 1：用户提供 Repository URL 和 Credentials（建议用户名和密码形式）。

样式：

Repository URL - `http://YourGitlabIP/Username/YourProject.git`
Credentials - `Username/Password`

步骤 2：安全组为用户分配 Fortify SSC 账号和权限。

方式三：用户部署 Git

用户只安装了 Git，参考以下步骤：

步骤 1：Git 官方下载安装：<https://git-scm.com/download/gui/win>。

步骤 2：安装完成以后配置 Git 生成 SSHKEY 密钥。
参考：<https://blog.csdn.net/jingtingfengguo/article/details/51892864>

步骤 3：将 Git 生成的 SSHKEY 密钥发送给我们。

步骤 4：我们分配 GitLab 代码上传地址，上传代码至 GitLab。

代码安全漏洞修复要求

信息系统上线前至少需修复 Critical 和 High 级别安全漏洞，如有误报请予说明，反馈至网络中心安全组。

附录

以下指南由个人按照官网和网上的文档测试，重点记录关键步骤和提醒容易出错的解决办法。

Fortify 与 Jenkins 集成指南

登录 Jenkins，点击系统管理>全局安全配置> Security Realm> Jenkins 专用户户数据库，选择“Allow users to sign up”。

下载 “ HP_Fortify_Jenkins_Plugin_4.40.hpi ” （ 链接: https://pan.baidu.com/s/1eO_HJEC3u8-cfAu_gMheBA 密码:qth3），点击系统管理 > 插件管理 > Advanced> Upload Plugin，选择刚下载的文件，点击“Upload”，然后关闭浏览器，重启 Jenkins。

重新登录 Jenkins，点击系统管理>系统设置>HP Fortify Assessment，URL 栏输入“http://10.8.8.50:18080/ssc/”， Authentication Token（请与网络中心安全组联系），点击下面的“Advanced settings”，再点击“Test Connection”，旁边显示“Connection successful!”，连接成功。（注意：如果连接不成功，关掉浏览器重新登录，或者换台电脑试试。）

点击系统管理>全局安全配置，配置如下图。

Configure Global Security

☒ 启用安全

TCP port for JNLP agents ☐ 指定端口: ☒ 随机选取 ☐ 禁用

把禁用改为随机选取或指定端口即可

Agent protocols...

Disable remember me ☐

<http://blog.csdn.net/ouyanggengcheng>

访问控制

安全域

☒ Jenkins专用户数据库

☒ 允许用户注册

☐ LDAP

☐ Servlet容器代理

点击系统管理>管理节点，新建节点（New Node），填写节点名（Node name，

如 Fortify），选择固定节点，点击 OK 进入下一步，配置如下图。注意：远程工作目录（Remote root directory）填写“C:\nodefiel_floor3”（此为示例，请咨询网络中心安全组）。

Name	fortify
描述	fortify
# of executors	10
远程工作目录	c:\nodefile_floor3
标签	fortify
用法	尽可能的使用这个节点
启动方法	Launch agent via Java Web Start
Availability	Keep this agent online as much as possible

Node Properties

- ☐ Environment variables
- ☐ Prepare jobs environment
- ☒ Tool Locations

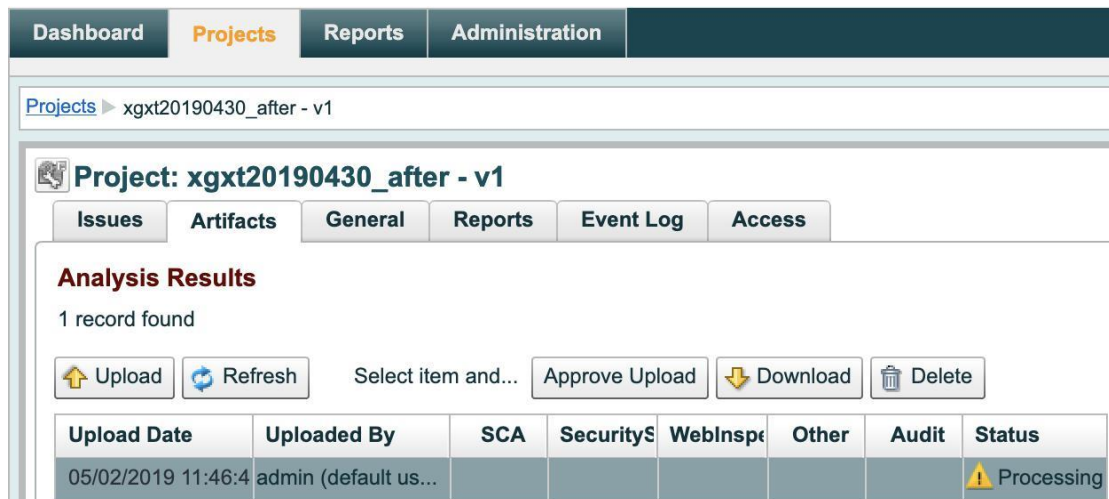
工具位置列表	名称	(Git) Default
	目录	c:\program files\git\cmd\git.exe

Fortify 管理员操作

每个 Jenkins 新建节点，需要在 Fortify 新建文件夹。



出现上图，需要在 fortify 机器上打开要连接的 jenkins web，点击“launch”，出现下图，节点连接成功。



完成构建后，需要在 ssc 对相应项目 approve upload。

为项目开发人员分配 ssc 登录账号和项目权限，点击 preview new dashboard。

UsersLocalLDAPRolesConfiguration

First Name

xia

Last Name

biyu

Roles *

Developer

Email

xiabiyu@mail.sysu.edu.cn

☒ User must change password at next login

☐ Password never expired

☐ Suspended

Access

xgxt20190430_after v1

Delete

Edit

自动化构建指南

登录 Jenkins，新建任务，选择“构建一个自由风格的软件项目”，点击 OK 进入项目构建页面。

标签栏点击“General”，配置运行节点，输入节点名或标签名，如下图。

Jenkins + test0321 +

GeneralSource Code ManagementBuild TriggersBuild EnvironmentBuildPost-build Actions

Description

[纯文本] Preview

☐ GitHub 项目

☐ This build requires lockable resources

☐ Throttle builds

☐ 丢弃旧的构建

☐ 参数化构建过程

☐ Disable this project

☐ Execute concurrent builds if necessary

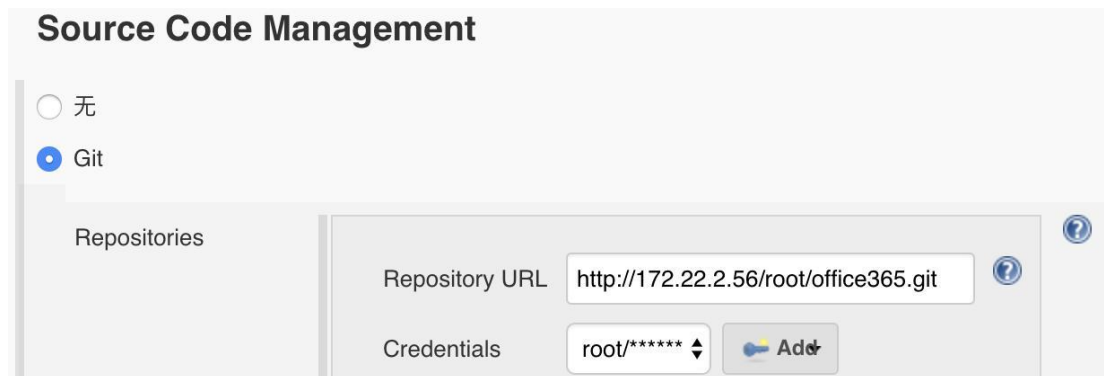
☒ Restrict where this project can be run

Label Expression

Fortify

Label Fortify is serviced by 1 node. Permissions or other restrictions provided by plugins may prevent this job from running on those nodes.

标签栏点击“Source Code Management”，选择“Git”，填写 Repository URL 和 Credentials，以下图为例。注意：Credentials 不需要 root 权限。



Source Code Management

☐ 无

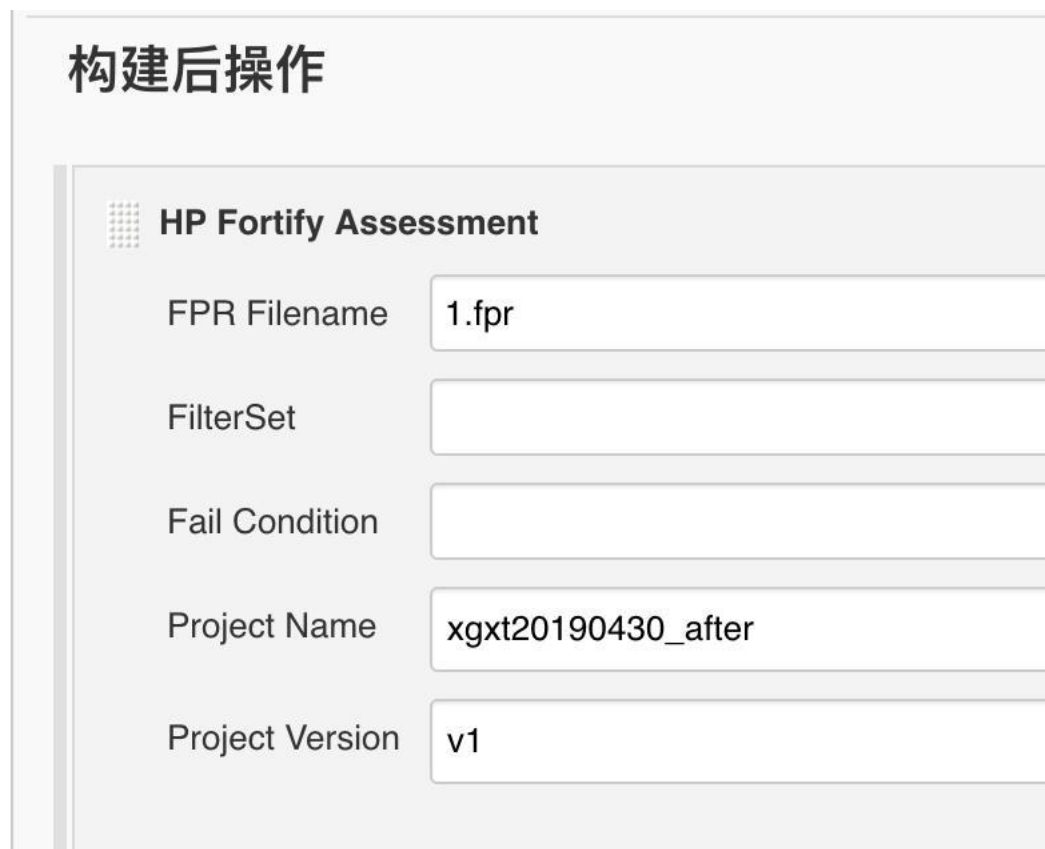
☒ Git

Repositories


Repository URL

Credentials

标签栏点击“Build”，下拉菜单选择“执行 Windows 批处理命令”，Command 栏输入“sourceanalyzer -scan %WORKSPACE% -f 1.fpr”，以调用 Fortify SCA。（1.fpr 是 fortify 扫描产生的文件，最好每次构建修改文件名，避免覆盖）



构建后操作

 **HP Fortify Assessment**

FPR Filename

FilterSet

Fail Condition

Project Name

Project Version

继续按上图配置构建后操作。(fpr 文件名与上面构建填的文件名相同, project name 自行命名)

保存后点击立即构建就可以完成第一次构建。

Jenkins 安装和配置指南

Linux 平台

以测试环境 Unbuntu 虚拟机为例, 步骤如下:

安装和配置 JAVA

注意: JDK 有 OpenJDK 和 Oracle 官方 JDK, 最新版本的 Jenkins 不支持 OpenJDK, 否则在安装 Jenkins 的时候会报错“Found an incorrect Java version”。

OracleJDK 安装方法如下:

```
#添加 ppa
sudo add-apt-repository ppa:webupd8team/java
sudo apt-get update

#安装 oracle-java-installer
sudo apt-get install oracle-java8-installer

#设置系统默认 jdk
sudo update-java-alternatives -s java-8-oracle

#java 安装测试
java -version
javac -version
```

安装 Jenkins

安装方法见 Jenkins 官网 (<https://pkg.jenkins.io/debian/>) 。

由于 Unbuntu 虚拟机同时装了 Gitlab, 8080 端口有冲突, 需要把 Jenkins 端口改为 8081, 修改 2 个文件 (“/etc/init.d/Jenkins”、“/etc/default/Jenkins”), 把 8080 端口改为 8081 端口。


其他安装方法也可以参考以下链接:

<https://www.jianshu.com/p/845f267aec52>

<https://blog.csdn.net/jb19900111/article/details/18552913>

Jenkins 与 Gitlab 配置

登录 Jenkins, 新建任务, 点击“构建一个自由风格的软件项目”, 标签栏点击“Source Code Management”, 选择“Git”, 填写 Repository URL 和 Credentials, 以下图为例。



Source Code Management

☐ 无
☒ Git

Repositories

Repository URL

Credentials

执行立即构建，测试能否 check out 代码，以下图为例。

Jenkins ▶ test0321 ▶

Back to Dashboard

Status

Changes

Workspace

立即构建

Delete 工程

Configure

Rename

Build History

find

x

#10

2019-3-22 下午3:20

#9

2019-3-22 下午2:54

工程 test0321

Workspace

Recent Changes

Permalinks

- [最近一次构建 \(#10\), 2 天 19 小时 ago](#)
- [最近稳定构建 \(#10\), 2 天 19 小时 ago](#)
- [最近成功的构建 \(#10\), 2 天 19 小时 ago](#)
- [最近失败的构建 \(#9\), 2 天 19 小时 ago](#)
- [最近未成功的构建 \(#9\), 2 天 19 小时 ago](#)
- [最近完成的构建 \(#10\), 2 天 19 小时 ago](#)

Windows 平台

安装和配置 JAVA

安装完 JDK 后需要配置环境变量，运 cmd 输入“java -version”，验证变量是否配置成功。过程略。

安装 Jenkins

安装过程请参考 Jenkins 官网。注意：安装 Jenkins 尽量新建一个文件夹，文件夹名字不能有空格。

Jenkins 与 Gitlab 配置见上。

Gitlab 安装和配置指南

Gitlab 官网安装指南 (<https://about.gitlab.com/install/>) , 选择相应的 OS。

CentOS 7 安装 Gitlab

注意事项:

Windows 平台安装 Gitlab

Gitlab 服务端不支持 Windows 平台, 因此本文通过 Windows 10 自带的 Hyper-v 安装 Unbuntu 虚拟机, 在虚拟机上安装 Gitlab 服务端。

安装 Unbuntu 虚拟机

虚拟机安装过程略过, 提醒一下, Hyper-v 有第一代虚拟机和第二代虚拟机, 如果选第二代虚拟机安装过程出错, 建议选第一代虚拟机试试。

安装 Gitlab 服务端

安装过程参考 Gitlab 官网 (<https://about.gitlab.com/install/#ubuntu>) , 提醒一下, Gitlab 默认管理员账号为 root。

Git 安装配置

以 OS X 为例。

1. 通过 xcode 安装 git

2. 生成 ssh key

#检查是否已经存在 SSH

Key ls -al ~/.ssh

```
-rw----- 1 lb staff 1675 4 7 2018 id_rsa
-rw-r--r-- 1 lb staff 394 4 7 2018 id_rsa.pub
```

#已存在 ssh key（不存在请先生成 ssh key），将公钥放到剪切板

pbcopy < ~/.ssh/id_rsa.pub

打开 GitLab, 登录, 点击账号头像, 找到左边栏有一个的按钮, 点击“ADD SSH KEY”

按钮添加, 将已经获得的 SSH Key 粘贴到“Key”, title 随便。

源代码上传到 Gitlab

Windows 平台

1、 测试环境

源代码本地环境: Windows 10, Git

Gitlab: Ubuntu

2、 步骤

步骤 1: Windows 10 安装 Git, 生成公钥。过程略。

步骤 2: Web 登录 Gitlab, 导入步骤 1 生成的公钥。过程略。

步骤 3: Windows 10 上操作, 右键要上传的项目, 选择 Git Bash Here

```
# 把下面标红部分改为你的实际情况
git config --global user.name "linb23"
git config --global user.email
"linb23@mail.sysu.edu.cn" git init
git remote add origin git@172.22.2.56:root/office365.git
git add .
git commit -m "office365"
git push -u origin master
```

“git remote add origin…”如果出现“fatal: remote origin already exists.”错误提示，输入“git remote rm origin”,再重新输入“git remote add origin…” 步骤 4: 登录 Gitlab 即可看到刚才上传的项目。

OS X 平台

打 开 终 端 ,cd 到 需 要 上 传 的 项 目 中 (如 :
/users/lb/sourcecode/xgxt_201904_before)

其他同 Windows 平台。

注意设置 git 的 user.name 和 user.email 与 ssh key 一致，我的 mac 上的 ssh key 在 github 使用的是 qq89205/89205@qq.com，所以 user.name 和 user.email 相应修改。