源码泄露

Git 源码泄露

介绍

Git是一个开源的分布式版本控制系统，在执行git init初始化目录的时候，会在当前目录下自动创建一个.git目录，用来记录代码的变更记录等。发布代码的时候，如果没有把.git这个目录删除，就直接发布到了服务器上，攻击者就可以通过它来恢复源代码。

使用dirsearch工具扫描后台，如果存在则会扫描出 .git 目录

```
HEAD：这个git项目当前处在哪个分支里；

config：文件包含项目特有的配置选项，git config命令会改动它；

description：项目的描述信息

hooks/：系统默认钩子脚本目录

info/：目录包含一个全局性排除（global exclude）文件，用以放置不希望被记录在 .gitignore 文件中的忽略模式（ignored patterns）

objects/：目录存储所有数据内容 (commits, trees, blobs, tags)

refs/：标识你项目里的每个分支指向了哪个提交(commit)。

index:文件保存暂存区信息
```

漏洞利用工具

GitHack

GitHack 是一个 .git 泄露利用脚本，通过泄露的 .git 文件夹下的文件重建还原工程源代码

用法示例：

```
GitHack.py http://www.openssl.org/.git/
```

GitHack 是一个 .git 泄露利用脚本，通过泄露的 .git 文件夹下的文件重建还原工程源代码

GitHacker

dvcs-ripper

用法示例：

```
rip-git.pl -v -u http://www.example.com/.git/
```

此工具集合了多种漏洞利用工具，包括bzr、cvs、git、hg、svn，比较推荐使用

Dumpall

适用于.git/.svn/.DS_Store泄漏和目录列出

SVN 源码泄露

介绍

SVN是一个开放源代码的版本控制系统。在使用SVN管理本地代码过程中，会自动生成一个名为.svn的隐藏文件夹，其中包含重要的源代码信息。网站管理员在发布代码时，没有使用'导出'功能，而是直接复制代码文件夹到WEB服务器上，这就使.svn隐藏文件夹被暴露于外网环境，可以利用.svn/entries文件，获取到服务器源码。

使用工具AWVS，可以探测出来

漏洞利用工具

SvnHack

SvnExploit

Seay SVN

Dumpall

dvcs-ripper

用法示例：

```
rip-svn.pl -v -u http://www.example.com/.svn/
```

HG 源码泄漏

介绍

Mercurial 是一种轻量级分布式版本控制系统，使用 `hg init` 的时候会生成.hg。当开发人员使用 Mercurial 进行版本控制，对站点自动部署。如果配置不当,可能会将.hg 文件夹直接部署到线上环境。这就引起了 hg 泄露漏洞。

漏洞利用工具

dvcs-ripper

用法示例：

```
rip-hg.pl -v -u http://www.example.com/.hg/
```

CVS 泄露

介绍

CVS是一个C/S系统，多个开发人员通过一个中心版本控制系统来记录文件版本，从而达到保证文件同步的目的。主要是针对 CVS/Root以及CVS/Entries目录，直接就可以看到泄露的信息。

cvs项目在初始化(cvs checkout project)的时候, 会在project目录下创建一个名为CVS的目录,其中保存了各个文件的修改和commit记录. 通过此目录可以获取代码的历史版本. 其中两个关键文件为: CVS/Root和CVS/Entries, 分别记录了项目的根信息和所有文件的结构

漏洞利用工具

dvcs-ripper

运行示例:

```
rip-cvs.pl -v -u http://www.example.com/CVS/
```

Bazaar/bzr 泄露

介绍

Bazaar是一种分布式版本控制系统，类似于Git和Mercurial，Bzr泄露是指在使用Bazaar版本控制系统时，由于配置不当或操作失误，导致敏感信息（如源代码、密码等）被泄露到公共仓库或其他不安全的地方。这种泄露可能会导致项目的知识产权受损，甚至影响到项目的安全性。

漏洞利用工具

dvcs-ripper

运行示例:

```
rip-bzr.pl -v -u http://www.example.com/.bzr/
```

DS_Store 泄露

介绍

.DS_Store是Mac OS保存文件夹的自定义属性的隐藏文件，如文件的图标位置或背景色，相当于Windows的desktop.ini。
由于开发人员发布代码时未删除文件夹中隐藏的.DS_store，可能造成文件目录结构泄漏、源代码文件等敏感信息的泄露。

漏洞利用工具

[ds_store_exp](

https://github.com/lijiejie/ds_store_exp
)

使用教程:

```
python ds_store_exp.py http://www.example.com/.DS_Store
```

[Dumpall]

备份文件 泄露

介绍

管理员将网站源代码备份在Web目录下，攻击者通过猜解文件路径，下载备份文件，导致源代码泄露。

常见的备份文件后缀：

```
.rar

.zip

.7z

.tar.gz

.bak

.txt

.old

.temp
```

漏洞利用工具

[御剑]

[dirsearch]

WEB-INF/web.xml 泄露

介绍

WEB-INF是Java的WEB应用的安全目录。如果想在页面中直接访问其中的文件，必须通过web.xml文件对要访问的文件进行相应映射才能访问。
WEB-INF主要包含一下文件或目录：

```
/WEB-INF/web.xml：
Web应用程序配置文件，描述了 servlet 和其他的应用组件配置及命名规则

/WEB-INF/classes/：
含了站点所有用的 class 文件，包括 servlet class 和非servlet class，他们不能包含在 .jar文件中

/WEB-INF/lib/：
存放web应用需要的各种JAR文件，放置仅在这个应用中要求使用的jar文件,如数据库驱动jar文件

/WEB-INF/src/：
源码目录，按照包名结构放置各个java文件。
```

```
/WEB-INF/database.properties：
数据库配置文件
```

WEB-INF/web.xml泄露的起因就是我们在使用网络架构的时候，对静态资源的目录或文件的映射配置不当，可能会引发一些的安全问题，导致web.xml等文件能够被读取。

漏洞利用

通过找到 web.xml 文件，推断 class 文件的路径，最后直接 class 文件，再通过反编译 class 文件，得到网站源码。

---

vim文件 泄露

介绍

vim程序在编辑文件时产生的临时文件，它是隐藏文件，如果正常退出临时文件自动删除，如果意外退出就会保留，文件名为 `.filename.swp`。

第一次意外退出会产生`.filename.swp`，第二次会产生`.filename.swo`，第三次会产生`.filename.swn`、

漏洞利用

可通过直接访问临时文件，下载回来后删掉末尾的.sw*后缀即可，获得源码文件。