【转】ctf——web源码泄露及利用办法

转载

ch3nwr1d 于 2019-06-04 14:55:12 发布 398 收藏 1

文章标签： ctf 源码泄露

转载： https://blog.csdn.net/wy_97/article/details/78165051

.hg源码泄漏

漏洞成因：

hg init的时候会生成.hg

e.g.http://www.am0s.com/.hg/

漏洞利用：工具：dvcs-ripper

rip-hg.pl -v -u http://www.am0s.com/.hg/

.git源码泄漏

漏洞成因：在运行git init初始化代码库的时候，会在当前目录下面产生一个.git的隐藏文件，用来记录代码的变更记录等等。在发布代码的时候，把.git这个目录没有删除，直接发布了。使用这个文件，可以用来恢复源代码。

1.漏洞利用：工具：GitHack #注：遇到的题这个工具好像都不行

2.上海市网络安全大赛

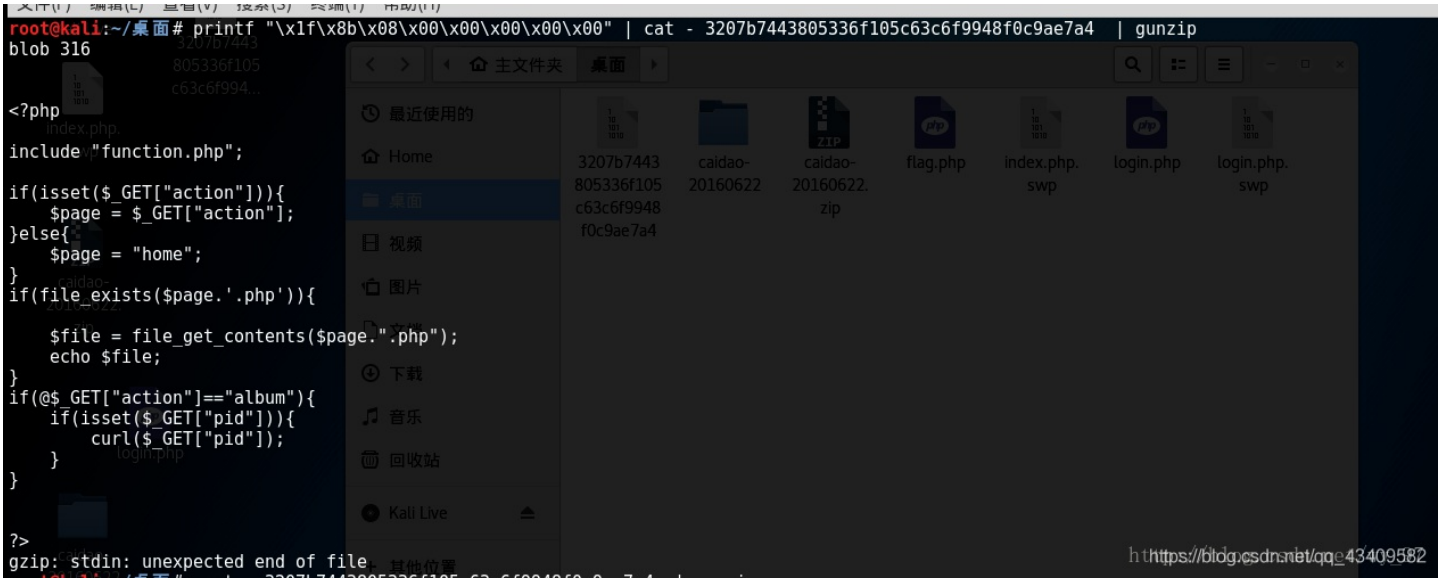
官方wp给的利用方法：

/.git可以列目录，下载3207b7443805336f105c63c6f9948f0c9ae7a4

printf "\x1f\x8b\x08\x00\x00\x00\x00\x00" | cat -

3207b7443805336f105c63c6f9948f0c9ae7a4 | gunzip

本地kali测试：



ummmm，这个命令查了很久，linux不好的人真的受伤

大体意思就是把 printf后面的字符串和 cat后面文件的内容拼接起来交给gunzip解压.
关于\x1f\x8b\x08\x00\x00\x00\x00\x00，本地我也测试了下，是乱码，转gbk，utf-8都不行

```
# -*- coding:utf-8 -*-
'''
 @Author:       GETF
 @Email:        GETF_own@163.com
 @DateTime:     2017-11-07 21:01:26
 @Description: Description
'''
import chardet



data = b"\x1f\x8b\x08\x00\x00\x00\x00\x00"
print(chardet.detect(data))


str2 = data.decode('windows-1252')
str2.encode('utf-8')
print(str2)
```

查询了百度，
![在这里插入图片描述](https://img-blog.csdnimg.cn/20190604145018546.png?x-oss-process=image/watermark,type_ZmFuZ3poZW5naGVpdGk,shadow_10,text_aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3FxXzQzNDA5NTgy,size_16,color_FFFFFF,t_70

猜测应该是php文件解释的前几个字符标志，这里带上这个才能让gunzip解压出来（如有不对，欢迎大佬指点~）

.DS_Store文件泄漏

漏洞成因:在发布代码时未删除文件夹中隐藏的.DS_store，被发现后，获取了敏感的文件名等信息。漏洞利用:

http://www.am0s.com/.ds_store

注意路径检查工具：dsstoreexp

python ds_store_exp.py http://www.am0s.com/.DS_Store

网站备份压缩文件

在网站的使用过程中，往往需要对网站中的文件进行修改、升级。此时就需要对网站整站或者其中某一页面进行备份。当备份文件或者修改过程中的缓存文件因为各种原因而被留在网站web目录下，而该目录又没有设置访问权限时，便有可能导致备份文件或者编辑器的缓存文件被下载，导致敏感信息泄露，给服务器的安全埋下隐患。漏洞成因及危害:该漏洞的成因主要有以下两种：

```
服务器管理员错误地将网站或者网页的备份文件放置到服务器web目录下。



编辑器在使用过程中自动保存的备份文件或者临时文件因为各种原因没有被删除而保存在web目录下。
```

漏洞检测:该漏洞往往会导致服务器整站源代码或者部分页面的源代码被下载，利用。源代码中所包含的各类敏感信息，如服务器数据库连接信息，服务器配置信息等会因此而泄露，造成巨大的损失。被泄露的源代码还可能会被用于代码审计，进一步利用而对整个系统的安全埋下隐患。

.rar

.zip

.7z

.tar.gz

.bak

.swp

.txt

.html

SVN导致文件泄露

Subversion，简称SVN，是一个开放源代码的版本控制系统，相对于的RCS、CVS，采用了分支管理系统，它的设计目标就是取代CVS。互联网上越来越多的控制服务从CVS转移到Subversion。Subversion使用服务端—客户端的结构，当然服务端与客户端可以都运行在同一台服务器上。在服务端是存放着所有受控制数据的Subversion仓库，另一端是Subversion的客户端程序，管理着受控数据的一部分在本地的映射（称为“工作副本”）。在这两端之间，是通过各种仓库存取层（Repository Access，简称RA）的多条通道进行访问的。这些通道中，可以通过不同的网络协议，例如HTTP、SSH等，或本地文件的方式来对仓库进行操作。

e.g.http://www.am0s.com/admin/scripts/fckeditor.266/editor/.svn/entries

漏洞利用:工具：dvcs-ripper

rip-svn.pl -v -u http://www.am0s.com/.svn/

Seay-Svn

WEB-INF/web.xml泄露

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

漏洞成因：通常一些web应用我们会使用多个web服务器搭配使用，解决其中的一个web服务器的性能缺陷以及做均衡负载的优点和完成一些分层结构的安全策略等。在使用这种架构的时候，由于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。漏洞检测以及利用方法：通过找到web.xml文件，推断class文件的路径，最后直接class文件，在通过反编译class文件，得到网站源码。一般情况，jsp引擎默认都是禁止访问WEB-INF目录的，Nginx 配合Tomcat做均衡负载或集群等情况时，问题原因其实很简单，Nginx不会去考虑配置其他类型引擎（Nginx不是jsp引擎）导致的安全问题而引入到自身的安全规范中来（这样耦合性太高了），修改Nginx配置文件禁止访问WEB-INF目录就好了： location ~ ^/WEB-INF/* { deny all; } 或者return 404; 或者其他！
CVS泄漏

漏洞利用测试的目录

http://www.am0s.com/CVS/Root 返回根信息

http://www.am0s.com/CVS/Entries 返回所有文件的结构

取回源码的命令

bk clone http://www.am0s.com/name dir

这个命令的意思就是把远端一个名为name的repo clone到本地名为dir的目录下。查看所有的改变的命令，转到download的目录

bk changes

Bazaar/bzr

工具：dvcs-ripper

rip-bzr.pl -v -u http://www.am0s.com/.bzr/