网军利用谷歌Chrome浏览器漏洞,针对朝鲜目标发起水坑攻击



前言

今天要说的这个故事主角,疑似来自于XX国家情报院的Darkhotel同志,利用0day漏洞为Chrome音频组件的UAF漏洞,其于最后还试图在下载的恶意代码中混入朝鲜Lazarus网军代码作为假旗试图混淆视听。(声明:该疑似出自卡巴斯基报告,与黑鸟无关)

正文

近日,卡巴斯基曝光了一起利用Chrome漏洞(CVE-2019-13720)进行水坑攻击的活动,该攻击行动被命名为WizardOpium,直译为巫师**。

我们称这些攻击为Operation WizardOpium。到目前为止,我们还无法与任何已知的威胁参与者建立明确的联系。与Lazarus攻击有某些非常弱的代码相似性,尽管这很可能是错误的标记。目标网站的配置与最近部署了类似虚假标记攻击的反对。即与DarkHotel攻击更加一致。

而该攻击行动,疑似来自Darkhotel的网军入侵了一个朝鲜的网站,并挂上了一个js脚本,疑似目的为针对访问朝鲜网站的某些目标进行攻击,若选中目标,则会下发Chrome音频组件漏洞利用代码,触发漏洞后,通过探测泄露地址和堆喷等操作下发最终的恶意软件。



你要问我怎么看出来是朝鲜的网站,随便点开一个网址便是。

十月24, 2019

朝鲜民主主义人民共和国

可见,此前网站上被挂上并加载了一个javascript编写的js文件

hxxp://http://code.jquery.cdn.behindcorona.com/jquery-validates.js

```
vid="footer-mark" class="col-md-3 mgtop20">
tja href="index.php"><mg src="/asset/images/footer_mark.png" /&gt;&lt;/a&gt;&lt;/a&gt;
tja footer-content" class="col-md-9 mgtop20"&gt;
tja vid="footer-naw"&gt;
tja vi
```

然后,该脚本将加载另一个名为.charlie.XXXXXXXX.js的脚本。

脚本功能是为了判断目标是否为:

- 1、windows 64位版本而不是WOW64进程
- 2、Chrome浏览器版本是否大于或等于65

```
if (navigator.userAgent.indexOf("Win64") == -1 || navigator.userAgent.indexOf("WOW64") != -1)
return;

if (navigator.userAgent.indexOf("Windows NT 6.1") == -1)
return;

let r = navigator.userAgent.indexOf("Chrome/");

if (r == -1)
return;

if (parseInt(navigator.userAgent.substr(r + "Chrome/".length, 3)) < 65)
```

若符合条件,该脚本就会开始向http://behindcorona.com执行AJAX请求,并将其中路径名指向传递给脚本的参数xxx.php,其中含有一些十六进制编码的字符串,而服务器会根据这些字符串返回指定的漏洞利用代码,而URL中还带有最终Payload的密钥和R**密钥。



下载完所有代码块后,R**脚本会将所有部分解密,并拼接在一起,这为攻击者提供了一个包含完整浏览器漏洞的新JavaScript代码,代码入口还会检查一次浏览器版本。

漏洞描述

```
Edit As: Code /wrap Y Run Script ✓
                                      Syntax >
                      20 30
                                               40
                                                    50 60
1 Function x(){var
   e=[144,144,144,144,101,72,139,4,37,48,0,0,0,72,139,96,8,72,129,236,8,16,0,0,144,144,144,14
  4,64,87,72,129,236,48,3,0,0,199,68,36,32,0,0,0,0,198,68,36,36,78,198,68,36,37,84,198,68,36
   ,38,68,198,68,36,39,76,198,68,36,40,76,198,68,36,41,46,198,68,36,42,68,198,68,36,43,76,198
   ,68,36,44,76,198,68,36,45,0,72,141,68,36,46,72,139,248,51,192,185,2,0,0,0,243,170,198,68,3
   6,48,76,198,68,36,49,100,198,68,36,50,114,198,68,36,51,76,198,68,36,52,111,198,68,36,53,97
   ,198,68,36,54,100,198,68,36,55,68,198,68,36,56,108,198,68,36,57,108,198,68,36,58,0,72,141,
  68,36,59,72,139,248,51,192,185,17,0,0,0,243,170,198,68,36,76,76,198,68,36,77,100,198,68,36,78,114,198,68,36,79,85,198,68,36,80,110,198,68,36,81,108,198,68,36,82,111,198,68,36,83,97
    198,68,36,84,100,198,68,36,85,68,198,68,36,86,108,198,68,36,87,108,198,68,36,88,0,72,141,
  68,36,89,72,139,248,51,192,185,3,0,0,0,243,170,198,68,36,92,82,198,68,36,93,116,198,68,36,94,108,198,68,36,95,65,198,68,36,96,108,198,68,36,97,108,198,68,36,98,111,198,68,36,99,99,
   198,68,36,100,97,198,68,36,101,116,198,68,36,102,101,198,68,36,103,72,198,68,36,10
  8,68,36,105,97,198,68,36,106,112,198,68,36,107,0,198,68,36,108,82,198,68,36,109,116,198,68
```

然后我手打了一下, 没啥解混淆的思路。

```
44,64,87,72,129,236,48,3,0,0,199,68,36,32,0,0,0,0,198,68,36,36,78,198,68,36,37,84,198
,68,36,38,68,198,68,36,39,76,198,68,36,40,76,198,68,36,4146,198,68,36,42,68,198,68,36
,43,76,198,68,36,44,76,198,68,36,45,0,72,141,68,36,4672,139,248,51,192,185,2,0,0,0,24
3,170,198,68,36,48,76,198,68,36,49,100,198,68,36,50,114,198,68,36,51,76,198,68,36,52,
111,198,68,36,53,97198,68,36,54,100,198,68,36,55,68,198,68,36,56,108,198,68,36,57,108
,198,68,36,58,0,72,141,68,36,59,72,139,248,51,192,185,17,0,0,0,243,170,198,68,36,76,7
6, 198, 68, 36, 77, 100, 198, 68, 36, 78, 114, 198, 68, 36, 79, 85, 198, 68, 36, 80, 110, 198, 68, 36, 81, 108
,198,68,36,82,111,198,68,36,83,97,198,68,36,84,109,198,68,36,85,68,198,68,36,86,108,1
98,68,36,87,108,198,68,36,88,0,72,141,68,36,89,72,139,248,51,192,185,3,0,0,e,243,170,
198,68,36,92,82,198,68,36,93,116,198,68,36,94,18,198,68,36,95,65,198,68,36,96,108,198
,68,36,97,108,198,68,3698,111,198,68,36,99,99,198,68,36,100,97,198,68,36,101,116,198,
68,36,102,101,198,68,36,103,72,198,68,36,104,101,198,68,36,105,97,198,68,36,106,112,1
                                                    time: 1ms
length: 403
lines: 1
Output
.AD.AA.r..7H...19...1)#h...AD.ADd.r..6H0..h62.....cg..h67....c...h69v...d.a.h6AF...d&..h
6Cv...dGa.h6E.!Ah6Fr..HQ.!.
.$1p...d.a.h6I...h6P.A.h6Qv...e!...e9q.h6T...h6Uh...ea....eq....e.r...e.!9$....Qp.$1p
...gga.h6w...h6x.A.h6y...h...h...h!...h9q.h6...h6.h..ha...hq...h.r...h.!9$...
h6.Yq.h6.a...a...6..!.h6.....
```

更详细的分析可见卡巴斯基的分析,分析中称漏洞利用代码中还含有很多的调试代码。

该0day漏洞利用了两个线程之间的竞争条件错误,原因是它们之间缺少适当的同步。它使攻击者处于UaF的状态,从而可能导致代码执行。

如果一个被释放的内存没有被正确地管理,就可能发生信息泄露,甚至是任意代码执行,而该漏洞利用程序首先尝试触发UaF对64位地址(作为指针)尝试获取泄露的地址。(信息泄露)

理想情况为下面的顺序。

- 1、如果地址成功泄漏,则表明漏洞利用正常。
- 2、泄漏的地址用于探测堆/堆栈的位置,这可使地址空间布局随机化 (ASLR) 技术无效;
- 3、通过在该地址附近进行进一步探测,便可以找到其他一些有用的指针,以供进一步利用。

之后,它尝试使用递归函数创建大量对象,从而开始堆喷寻找在内存未释放的指针进行利用,这也是教科书级的方式。并在最终得以执行Shellcode代码,并运行嵌入的Payload。

```
000000000: <mark>90</mark>909090
00000004: 6548
                                                  dec
                                                                   eax
                                                                   eax,[000000030]
00000006: 8B042530000000
                                                  mov
0000000D: 48
                                                  dec
                                                                   eax
0000000E: 8B6008
                                                  mov
                                                                   esp,[eax][8]
00000011: 48
                                                  dec
                                                                   eax
00000012: 81EC08100000
                                                                  esp,000001008
00000018: 90909090
                                                  nop
0000001C: 40
                                                                  eax
0000001D: 57
                                                  push
0000001E: 48
0000001F: 81EC30030000
                                                                  esp,000000330;
d,[esp][020],0
b,[esp][024],04E
b,[esp][025],054
b,[esp][027],04C
b,[esp][028],04C
00000025: C744242000000000
0000002D: C64424244E
                                                                                        ;'N'
00000032: C644242554
                                                                                        ;'T'
;'D'
00000037: C644242644
                                                                                        ;'L'
;'L'
0000003C: C64424274C
                                                   mov
00000041: C64424284C
00000046: C64424292E
                                                   mov
0000004B: C644242A44
                                                                                        ; 'D'
                                                                  b,[esp][02A],
                                                                                        ;'L'
                                                                  b,[esp][02B],04C
00000050: C644242B4C
                                                   mov
00000055: C644242C4C
                                                                  b,[esp][02C],040
b,[esp][02D],0
                                                   mov
                                                                                       知平地黑鸟
0000005A: C644242D00
                                                   mov
 999995F: 48
```

恶意软件

最终的Payload将下载加密的二进制文件(worst.jpg),并由shellcode进行解密。

解密后,恶意软件会将updata.exe释放到磁盘上并执行。 为了达到持久化,该恶意软件会在Windows任务计划列表中安装任务。

而该安装程序实际上是RAR格式的压缩文件,两个文件同时在2019年10月8号星期二01:49:31编译。

```
;The comment below contains SFX script commands
Path=%programdata%
Setup=iohelper.exe
Silent=1
Overwrite=1
Testing msdisp64.exe
Testing iohelper.exe
All OK

CK

TESTING

TEST
```

其中msdisp64.exe会尝试从硬编码的C2服务器中下载下一阶段的木马。下一阶段位于C2服务器上具有受害者计算机名称的文件夹中

类似 User-PC

因此攻击者可以了解哪些计算机被感染,从而筛选目标,并将下一阶段模块放置在C2服务器上的特定文件夹中,从而可以直接获取下一阶段Payload并运行。

解决方案

Google已针对Windows,Mac和Linux发布了Chrome版本78.0.3904.87,主要针对两个高危漏洞,一个是CVE-2019-13721,另一个就是上面提到的CVE-2019-13720,里面提到了是Chrome音频组件存在Use-after-free漏洞,请及时更新

Security Fixes and Rewards

Note: Access to bug details and links may be kept restricted until a majority of users are updated with a fix. We will also retain restrictions if the bug exists in a third party library that other projects similarly depend on, but haven't yet fixed.

This update includes <u>2</u> security fixes. Below, we highlight fixes that were contributed by external researchers. Please see the <u>Chrome Security Page</u> for more information.

[\$7500][1013868] **High** CVE-2019-13721: Use-after-free in PDFium. *Reported by bananapenguin on* 2019-10-12

[\$TBD][1019226] **High** CVE-2019-13720: Use-after-free in audio. *Reported by Anton Ivanov and Alexey Kulaev at Kaspersky Labs on 2019-10-29*

Google is aware of reports that an exploit for CVE-2019-13720 exists in the wild.

还有就是将下面这些IOC置黑

http://behindcorona.com

code.jquery.cdn.behindcorona [。] com 8f3cd9299b2f241daf1f5057ba0b9054 35373d07c2e408838812ff210aa28d90e97e38f2d0132a86085b0d54256cc1cd 27e941683d09a7405a9e806cc7d156c9 8fb2558765cf648305493e1dfea7a2b26f4fc8f44ff72c95e9165a904a9a6a48 f614909fbd57ece81d00b01958338ec2 cafe8f704095b1f5e0a885f75b1b41a7395a1c62fd893ef44348f9702b3a0deb kennethosborne@protonmail.com

当然最重要的一句话还是: Darkhotel这次攻击朝鲜目标, 也印证了朝鲜半岛的局势依旧紧张。



那对于我们而言,如何像卡巴那样捡到他们的0day攻击武器才是至关重要。

参考链接:

https://securelist.com/chrome-0-day-exploit-cve-2019-13720-used-in-operation-wizardopium/94866/

更多情报,请公众号关注后点击菜单栏的知识星球(打折ing) 扫二维码加入每日更新的知识星球,打开威胁情报&安全分析&红队攻防等等世界大门 黑鸟威胁情报中心 BlackorBird Threat Intelligence Center 可能,于您黑马

本文作者: 黑鸟, 转载请注明来自FreeBuf.COM

0Day漏洞

• # Google Chrome

o # apt