FireEye

Reversing the Gophe Spambot

Confronting COM Code and Surmounting STL Snags

Michael Bailey (@mykill)
BSides 2020
8 Feb 2020

FLARE

1

Michael Bailey (@mykill)

- Husband, Daddy, Bon Vivant
- Remote from Huntsville, AL
  - FLARE Reverse Engineer
  - Manager
  - Offensive Task Force
- Previously
  - Mandiant Red Team
  - Linux/Windows kernel @ Pikewerks
    - Bought by Raytheon
  - Windows server admin
- For fun
  - Babies, Foreign language, spinning DNB

2

What have I got for you?

- Something for reversers and learners
- Gophe from an RE perspective – this malware's got it all!
  - **COM** – so hot right now
  - **MAPI**, an API that will make you want to throw furniture
  - **VC STL**, under-represented in C++ reversing literature IMO
- Gophe summary

Avoiding

- Tool survey
- Full course on COM or STL

3

FireEye

Gophe

Aka MailClient, spambot extraordinaire

4

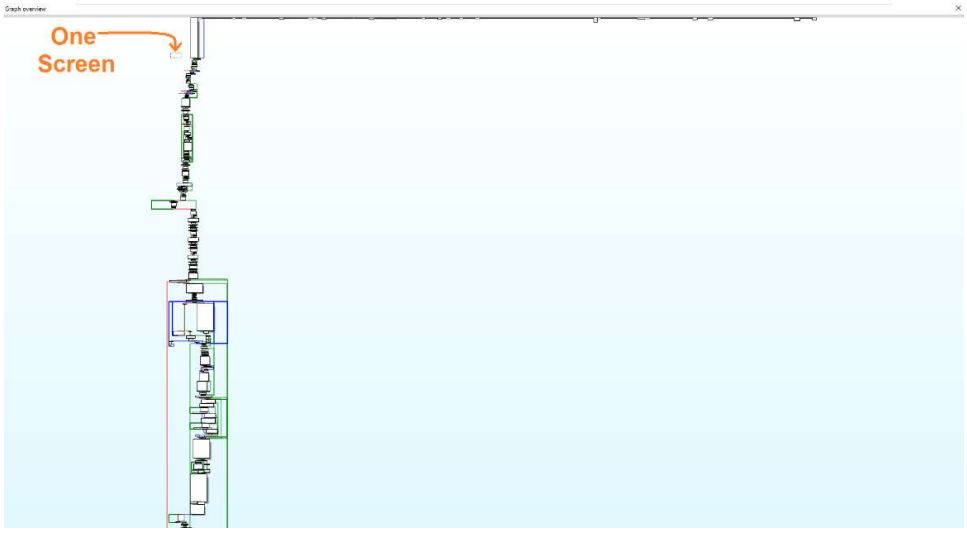
Harper's Index of Gophe

(MD5: `D9630C174B8FF5C0AA26168DF523E63E`)

Statistic	Figure
Size in bytes	2,783,232
Seconds to parse and auto-analyze in IDA 7.1	92
Size of `WinMain` in bytes	11,139
Number of monitors that would be needed to stack vertically to display all of `WinMain` at once	~90
`WinMain` function ranked by size	#2
Embedded PE-COFF binaries	3
Number of bytes in all `.text` sections combined	1,910,507
Number of functions in all binaries combined	9,792

5

Graph View of `WinMain`, Maximized



6

FireEye

Component Object Model

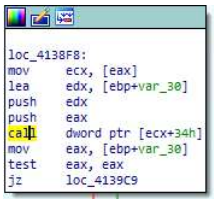
^ old font because 1993 technology

7

COM

Facts

- Out of 250,000 PE-COFF samples from a private corpus of mostly malware, **22%** call `CoCreateInstance`
- COM can abet persistence (`IShellLink` or `ITaskService`), FireWall modification (`HNetCfg.FwMgr`) and more
- Indirect calls hide code semantics

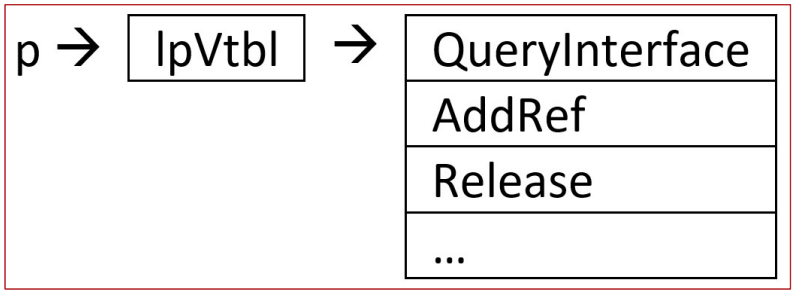


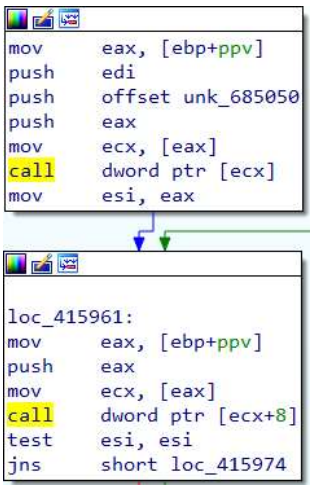
Primer

- A **CoClass** (COM class) can be identified by:
  - **`ProgID`** (e.g. `WScript.Shell`)
  - **`CLSID`** (a GUID)
- A CoClass implements one or more interfaces
- `Interfaces` define a virtual function table, identified uniquely by an:
  - **`IID`** – Interface ID (also a GUID)
- **`CoCreateInstance`** takes a `CLSID` and an `IID` and returns an interface pointer (called `ppv`)

8

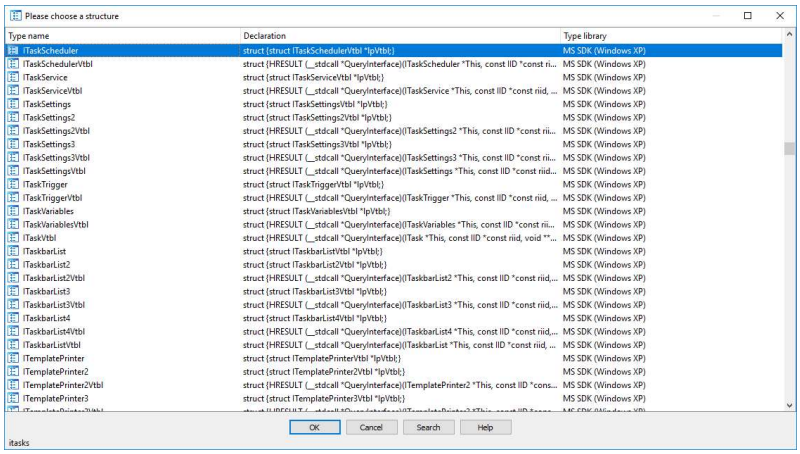
Pointer to Pointer to Vtable (ppv) and Indirect Calls





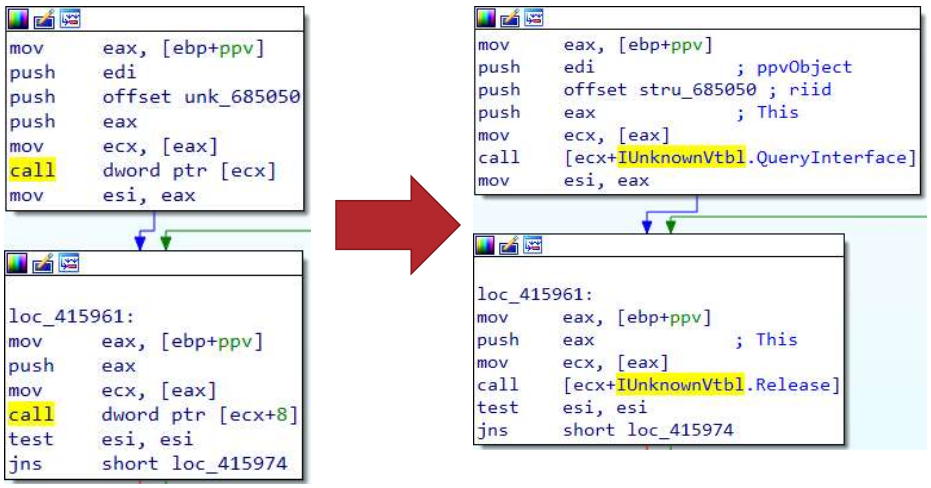
9

IDA Pro Structures

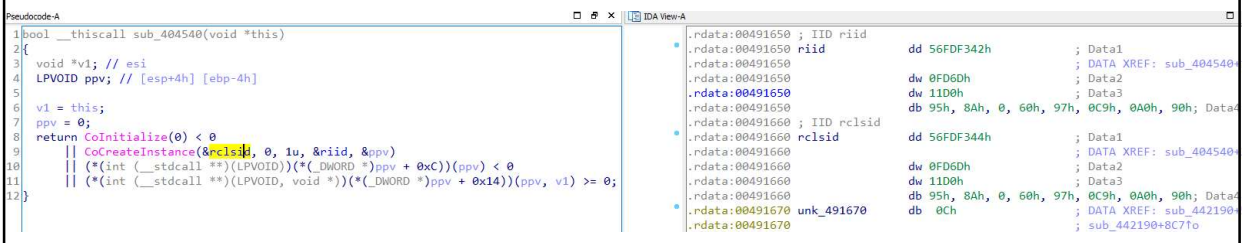


10

COM is a Game of Types

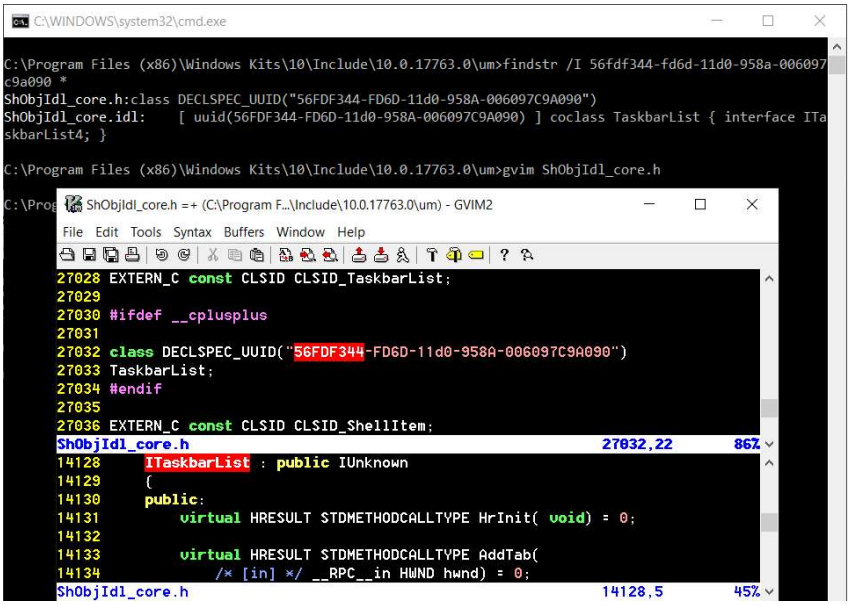


CoCreateInstance Xref 1/2 – Vanilla



SDK Headers

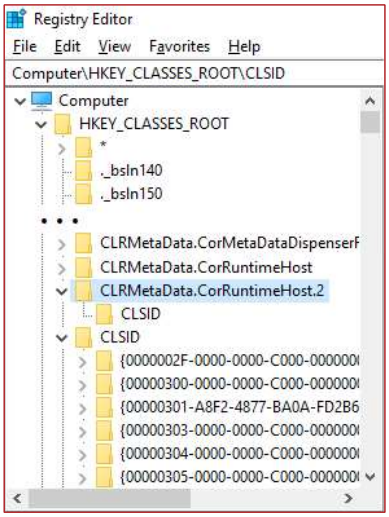
Before I knew about a much slicker trick



13

Searching COM Registrations

- Registry = **Authoritative**, **offline** CLSID/ProgID lookup
- Under `HKCR`*
  - `HKCR\` ← All registered **ProgIDs**
  - `HKCR\CLSID\` ← All registered **CLSIDs**
- To find a DLL given a ProgID:
  - **`HKCR\<YourProgID>\CLSID`** (Default) → GUID
  - **`HKCR\CLSID\{that-GUID}\`**
    - **`InprocServer32`** or **`LocalServer32`** (Default) → DLL/EXE



*`HKCR` is really a virtual hive composed from superimposing `HKCU\Classes` over `HKLM\Classes`

14

This One Weird Trick

15

Can't Argue with Success

Pseudocode-A

```
1  bool __thiscall sub_404540(void *this)
2  {
3    LPVOID ppv; // [esp+4h] [ebp-4h]
4
5    ppv = 0;
6    return CoInitialize(0) < 0
7        || CoCreateInstance(&rclsid, 0, 1u, &CLSID_ITaskbarList, &ppv)
8        || (*(int (__stdcall **)(LPVOID))(*(_DWORD *)ppv + 12))(ppv) < 0
9        || (*(int (__stdcall **)(LPVOID, void *))(*(_DWORD *)ppv + 20))(ppv, this) >= 0;
10 }
```

IDA View-A

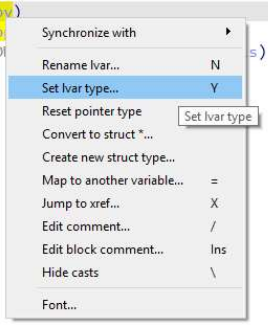
```
.rdata:0049164D                 db    0
.rdata:0049164E                 db    0
.rdata:0049164F                 db    0
.rdata:00491650 ; CLSID CLSID_ITaskbarList
.rdata:00491650 CLSID_ITaskbarList dd 56FDF342h
.rdata:00491650
.rdata:00491650                 dw 0FD6Dh
.rdata:00491650                 dw 11D0h
.rdata:00491650                 db 95h, 8Ah, 0,
.rdata:00491660 ; IID rclsid
```


16

Finally: Interface Pointer Illumination

```
bool __thiscall sub_404540(void *this)
{
  LPVOID ppv; // [esp+4h] [ebp-4h]

  ppv = 0;
  return CoInitialize(0) < 0
      || CoCreateInstance(&rclsid, 0, 1u, &riid, &ppv)
      || (*(int (__stdcall **)(LPVOID))(*(_DWORD *)p
      || (*(int (__stdcall **)(LPVOID, void *))(*(_D                s) >= 0;
}
```



17

Finally: Interface Pointer Illumination

```
bool __thiscall taskbar_deletetab(HWND hwnd)
{
  HWND v1; // esi
  ITaskbarList *ppv; // [esp+4h] [ebp-4h]

  v1 = hwnd;
  ppv = 0;
  return CoInitialize(0) < 0
      || CoCreateInstance(&clsid_00491660_TaskbarList, 0, 1u, &iid_00491650_ITaskbarList, (LPVOID *)&ppv)
      || ppv->lpVtbl->HrInit(ppv) < 0
      || ppv->lpVtbl->DeleteTab(ppv, v1) >= 0;
}
```

18

CoCreateInstance Xref 2/2 – Peculiar

- IID → CLSID trick is not even applicable!

Pseudocode-A

```
1 HRESULT __thiscall sub_415910(_DWORD *this, int a2, int a3, int a4)
2 {
3   _DWORD *v4; // edi
4   HRESULT v5; // esi
5   LPVOID ppv; // [esp+8h] [ebp-4h]
6
7   v4 = this;
8   if ( *this )
9     (*(void (__stdcall **)(_DWORD))(*(_DWORD *)*this + 8))(*this);
10  v5 = CoCreateInstance(&stru_685070, 0, 0x17u, &IID_IUnknown, &ppv);
11  if ( v5 < 0 )
12    goto LABEL_11;
13  v5 = OleRun((LPUNKNOWN)ppv);
14  if ( v5 >= 0 )
15    v5 = (**(int (__stdcall ***)(LPVOID, void *, _DWORD *))ppv)(ppv, &unk_685050, v4);
16  (*(void (__stdcall **)(LPVOID))(*(_DWORD *)ppv + 8))(ppv);
17  if ( v5 < 0 )
18LABEL_11:
19    *v4 = 0;
20  return v5;
21}
```

IDA View-A

```
.rdata:0068505B                 db    0
.rdata:0068505C                 db    0
.rdata:0068505D                 db    0
.rdata:0068505E                 db    0
.rdata:0068505F                 db  46h ; F
.rdata:00685060 ; IID IID_IUnknown
.rdata:00685060 IID_IUnknown    dd 0
.rdata:00685060
.rdata:00685060                 dw 0
.rdata:00685060                 dw 0
.rdata:00685060                 db 0C0h, 6 dup(0), 46h
.rdata:00685070 ; CLSID stru_685070
.rdata:00685070 stru_685070     dd 6F03Ah
.rdata:00685070
.rdata:00685070                 dw 0
.rdata:00685070                 dw 0
.rdata:00685070                 db 0C0h, 6 dup(0), 46h
.rdata:00685080 aBadCast        db 'bad cast',0
.rdata:00685080
.rdata:00685089                 align 4
.rdata:0068508C asc_68508C      db ': ',0
```


19

GUID Easy Mode

```
.rdata:00685070 stru_685070     dd 6F03Ah               ; Data1
.rdata:00685070                                         ; DATA XREF: s
.rdata:00685070                 dw 0                    ; Data2
.rdata:00685070                 dw 0                    ; Data3
.rdata:00685070                 db 0C0h, 6 dup(0), 46h  ; Data4
.rdata:006850                                                    s
.rdata:006850                                                   +E
.rdata:006850                                                    s
.rdata:00685090 aIostream       db 'iostream',0         ; DATA XREF: s
```

Output window

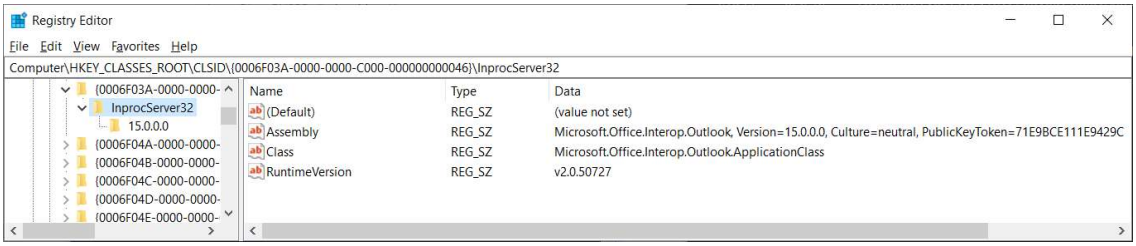
```
Python>uuid.UUID(bytes_le=idc.GetManyBytes(idc.here(), 16))
0006f03a-0000-0000-c000-000000000046
```

Python

20

CLSID Lookup

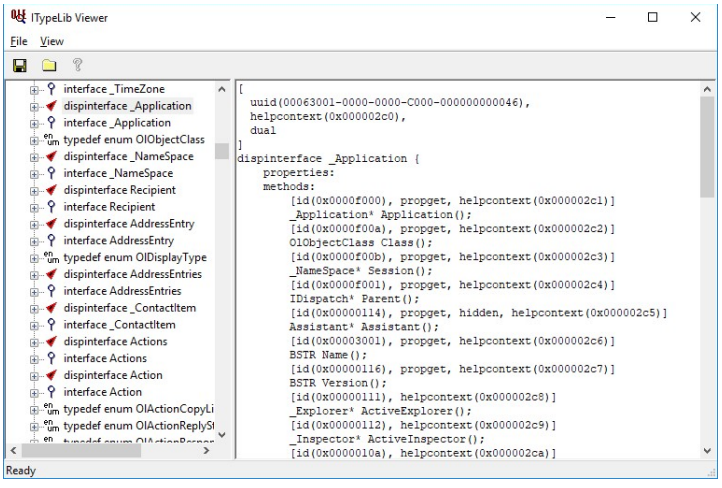
- Under `HKCR\CLSID`



21

Hunting TypeLibs

- `oleview.exe` : File->View TypeLib:
  - The COM EXE or DLL itself
  - *.tlb, *.olb, even *.dll
- Found via MSDN: `msoutl.olb`
- Had to hand-create vtbl
- **BUT REMEMBER!**
- Dispinterfaces inherit from `IDispatch`
- Meaning?



22

COM Interfaces Galaxy Brain Meme

IUnknown

1. **QueryInterface**
2. **AddRef**
3. **Release**

All COM objects

IDispatch : IUnknown

1. QueryInterface
2. AddRef
3. Release
4. **GetTypeInfoCount**
5. **GetTypeInfo**
6. **GetIDsOfNames**
7. **Invoke**

"Automation" objects

IWhatevs : IDispatch

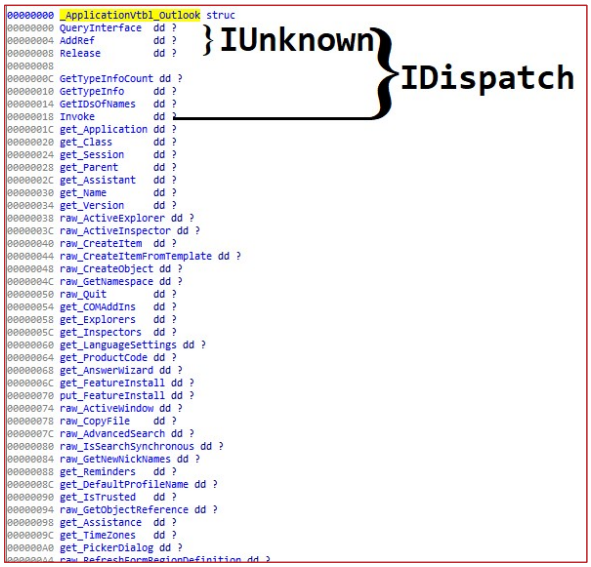
1. QueryInterface
2. AddRef
3. Release
4. GetTypeInfoCount
5. GetTypeInfo
6. GetIDsOfNames
7. Invoke
8. **DoWhatever**
9. **DoSomethingElse**

Arbitrary object

23

Outlook.Application (**implements IDispatch**)

- Implementing Vtbl by hand
- **Important**
  - OleView said "**dispinterface**"
  - Inherits from **IDispatch**
  - Which inherits from **IUnknown**
  - Must include those 7 methods FIRST



24

So, What was it Doing?

```
int __thiscall get_Outlook_Application(_Application_Outlook **ppvObject, int a2, int a3, int a4)
{
  void **ppvObject_alias; // edi
  int v5; // esi
  _Application_Outlook *ppv; // [esp+8h] [ebp-4h]

  ppvObject_alias = (void **)ppvObject;
  if ( *ppvObject )
    ((void (__stdcall *)(_Application_Outlook *))(*ppvObject)->lpVtbl->Release)(*ppvObject);
  v5 = CoCreateInstance(&CLSID_Outlook_dot_Application, 0, 0x17u, &iid_00685060_IUnknown, (LPVOID *)&ppv);
  if ( v5 < 0 )
    goto LABEL_11;
  v5 = OleRun((LPUNKNOWN)ppv);
  if ( v5 >= 0 )
    v5 = ((HRESULT (__stdcall *)(IUnknown *, const IID *const , void **))ppv->lpVtbl->QueryInterface)(
           (IUnknown *)ppv,
           &IID__Application,
           ppvObject_alias);
  ((void (__stdcall *)(IUnknown *))ppv->lpVtbl->Release)((IUnknown *)ppv);
  if ( v5 < 0 )
LABEL_11:
    *ppvObject_alias = 0;
  return v5;
}
```


25

And thennnnn?

- Getting MS Outlook version

```
CoInitializeEx(0, 2u);
ppv_Outlook__Application = 0;
v19 = 0;
if ( get_Outlook_Application((void **)&ppv_Outlook__Application, v12, v3, v4) < 0 )
  goto LABEL_26;
outlook_version = 0;
if ( !ppv_Outlook__Application )
  w_guard_check_bail(0x80004003);
((void (__stdcall *)(struct _Application_Outlook *, int *))ppv_Outlook__Application->Vtbl->get_Version)(
  ppv_Outlook__Application,
  &outlook_version);
```


26

And thennnnn?

- Quitting Outlook

```
if ( !ppv_Outlook_Application )
  w_guard_check_bail(0x80004003);
((void (__stdcall *)(struct _Application_Outlook *))ppv_Outlook_Application->Vtbl->raw_Quit)(ppv_Outlook_Application);
v40 = 7;
v39 = 0;
```


27

FireEye

MAPI

Messaging API
aka
COM jungle of perpetual MSDN searches

28

Three Facts about Messaging API (MAPI)

(#3 may surprise you)

1. MAPI is based on COM

29

Three Facts about Messaging API (MAPI)

(#3 may surprise you)

1. MAPI is based on COM
2. MAPI clients must match the processor architecture of the installed MAPI app

30

Three Facts about Messaging API (MAPI)

(#3 may surprise you)

1. MAPI is based on COM
2. MAPI clients must match the processor architecture of the installed MAPI app
3. MAPI is supported by Thunderbird

31

Meet MAPI

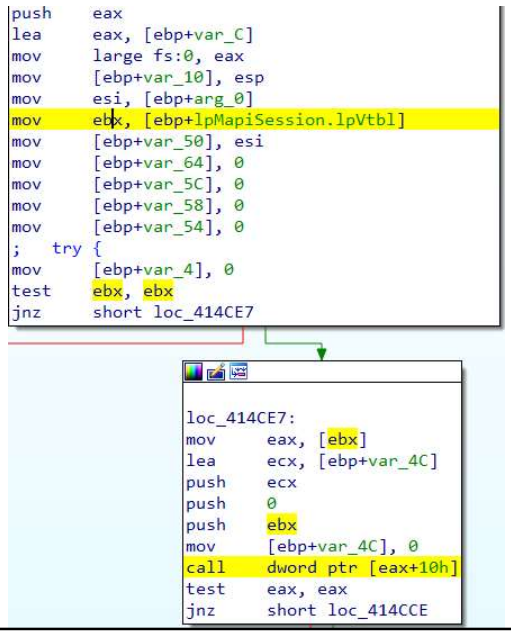
```
loc_41301F:
lea     eax, [ebp+MapiInit]
mov     [ebp+MapiInit], 0
push    eax             ; lpMapiInit
mov     [ebp+var_9C], 0
call    ds:MAPIInitialize
lea     eax, [ebp+pSession]
mov     [ebp+pSession], 0
push    eax             ; lppSession
push    62h ; 'b'       ; ulFlags
push    0               ; lpszPassword
push    0               ; lpszProfileNam
push    0               ; ulUIParam
call    ds:MAPILogonEx
mov     eax, [ebp+pSession]
test    eax, eax
jnz     short loc_413079
```

```
loc_414D1B:
push    0
mov     [ebp+var_24], 3
lea     edx, [ebp+var_24]
mov     [ebp+var_20], 3001001Fh
mov     [ebp+var_1C], 0FFF0102h
mov     [ebp+var_18], 3400000Bh
mov     eax, [ecx]
push    edx
push    ecx
call    dword ptr [eax+1Ch]
test    eax, eax
mov     eax, [ebp+var_4C]
jnz     loc_414F19
```


32

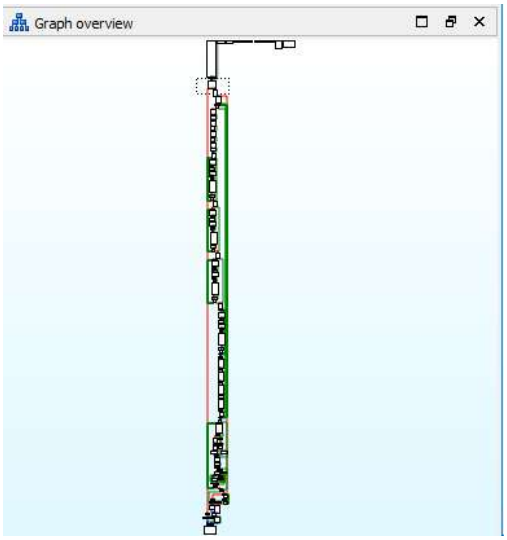
Handling MAPI

- Cascading type deps force top-down analysis
- Very documentation-driven
- Hex-Rays helps manage type propagation
  - Otherwise, more work applying `XxxVtbl` type to calls – see illustration →
- Ideally, hit `F5` or `Tab` to go to Hex-Rays
- Start at `MAPILogonEx`
- Then…

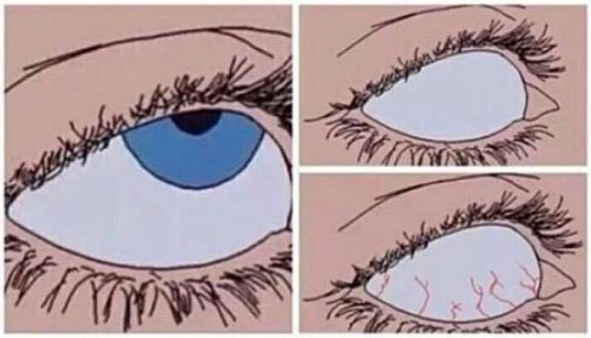


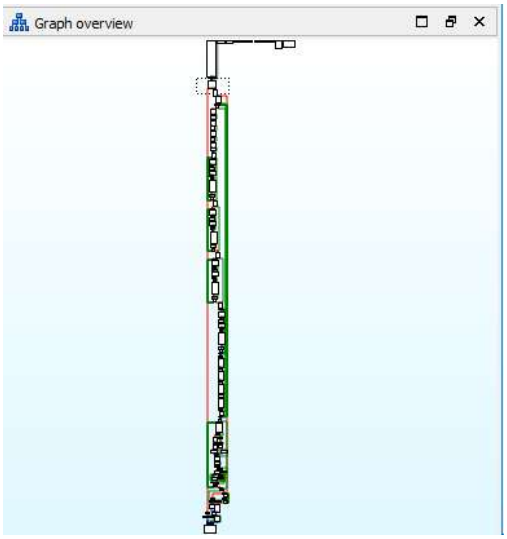
Algorithm for MAPI

- For each MAPI function call, may need to:
  - Check MSDN* for output value's COM type
  - Add that struct type to structures tab
  - Apply struct type to relevant variables
  - Check MSDN* to apply symbolic names for constants/enums
- Lather, rinse, repeat



Lather, Rinse, Repeat

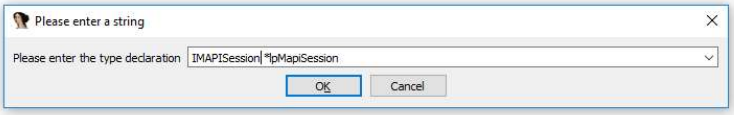




35

Example

```
if ( !lpMapiSession
  || (v4 = *lpMapiSession, v20 = 0, (*(int (__stdcall **)(int *, _DWORD, int *))(v4 + 0x10))(lpMapiSession, 0, &v20)) )
{
  *a1 = 0;
  a1[1] = 0;
  a1[2] = 0;
}
else
{
  v5 = v20;
  if ( v20 )
  {
    v27 = 3;
    v28 = 0x3001001F;
    v29 = 0xFFF0102;
    v30 = 0x3400000B;
    v6 = (*(int (__stdcall **)(int, int *, _DWORD))(*(_DWORD *)v20 + 0x1C))(v20, &v27, 0) == 0;
```



36

Example

```
if ( !lpMapiSession
  || (v4 = lpMapiSession->lpVtbl, v20 = 0, v4->GetMsgStoresTable(lpMapiSession, 0, (LPMAPITABLE *)&v20)) )
{
  *a1 = 0;
  a1[1] = 0;
  a1[2] = 0;
}
else
{
  v5 = v20;
  if ( v20 )
  {
    v27 = 3;
    v28 = 0x3001001F;
    v29 = 0xFFF0102;
    v30 = 0x34000008;
    v6 = (*(int (__stdcall **)(int, int *, ULONG))(*(_DWORD *)v20 + 0x1C))(v20, &v27, 0) == 0;
```


37

Beware When Using `IMAPIXxx` Interface Names…

- From `MAPIDefs.h`
- If you try `IMAPITABLE*`…
  - Hex-Rays says **Bad Declaration**
  - You **throw furniture**
  - All because capital letters

```
DECLARE_MAPI_INTERFACE_PTR(IMsgStore,        LPMDB);
DECLARE_MAPI_INTERFACE_PTR(IMAPIFolder,      LPMAPIFOLDER);
DECLARE_MAPI_INTERFACE_PTR(IMessage,         LPMESSAGE);
DECLARE_MAPI_INTERFACE_PTR(IAttach,          LPATTACH);
DECLARE_MAPI_INTERFACE_PTR(IAddrBook,        LPADRBOOK);
DECLARE_MAPI_INTERFACE_PTR(IABContainer,     LPABCONT);
DECLARE_MAPI_INTERFACE_PTR(IMailUser,        LPMAILUSER);
DECLARE_MAPI_INTERFACE_PTR(IDistList,        LPDISTLIST);
DECLARE_MAPI_INTERFACE_PTR(IMAPIStatus,      LPMAPISTATUS);
DECLARE_MAPI_INTERFACE_PTR(IMAPITable,       LPMAPITABLE);
DECLARE_MAPI_INTERFACE_PTR(IProfSect,        LPPROFSECT);
DECLARE_MAPI_INTERFACE_PTR(IMAPIProp,        LPMAPIPROP);
DECLARE_MAPI_INTERFACE_PTR(IMAPIContainer,   LPMAPICONTAINER);
DECLARE_MAPI_INTERFACE_PTR(IMAPIAdviseSink,  LPMAPIADVISESINK);
DECLARE_MAPI_INTERFACE_PTR(IMAPIProgress,    LPMAPIPROGRESS);
DECLARE_MAPI_INTERFACE_PTR(IProviderAdmin,   LPPROVIDERADMIN);
```


38

Meanwhile, at MSDN… Same Pitfall

IMAPISession::GetMsgStoresTable

03/08/2015 • 2 minutes to read •

Applies to: Outlook 2013 | Outlook 2016

Provides access to the message store table that contains information about all the message stores in the session profile.

C++ Copy

```
HRESULT GetMsgStoresTable(
  ULONG ulFlags,
  LPMAPITABLE FAR * lppTable
);
```


39

Not Helpful

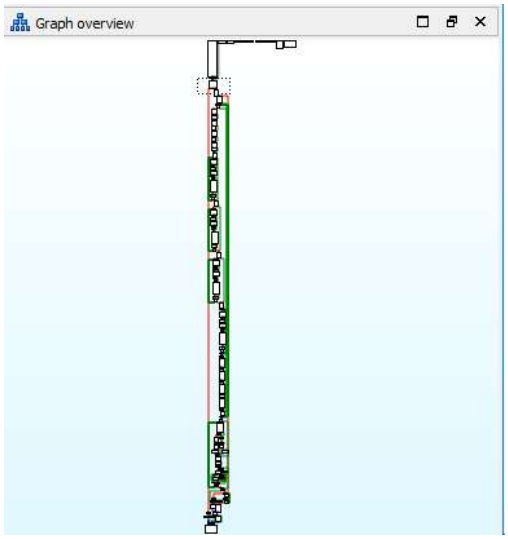
lppTable

> [out] A pointer to a pointer to the message store table.

40

Gophe MAPI Usage

- Hundreds of calls
- Three of the four binaries use it
- Often in conjunction with STL
  - Strings
  - Vectors
  - Maps
  - Sets
  - Vectors of Maps (yes, really)



41

FireEye

STL

Reversing Standard Template Library Code

42

Template Caveats

- *Very* slippery
  - Inheritance
  - Allocators
  - Template type (e.g. <class _**Ty**> varies generated code/structs
  - Mem layout fluctuation due to
    - Debug on/off
    - Security features
    - VC version
  - Inlining
  - Optimization

- Difficult to say anything absolute
  - Extra >> 2 (division by sizeof(x)=4)
  - Random constants like 0x3fffffff
  - Missing/added struct members
  - Etc.

43

FireEye

STL `basic_string` in VC

Level 1, go!

44

basic_string layout on x86 (featuring union _Bxty)

High

Offset (Size 0x18)	Field	
+0x14	_Myres	
+0x10	_Mysize	
+0x0c		
+8		
+4		
0	_Buf	_Ptr

Low

45

basic_string as a Structure

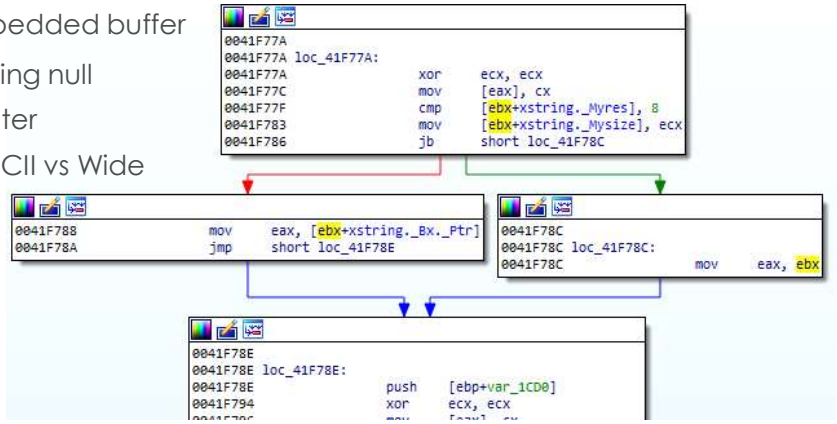
```
/* basic_string struct definition for IDA */
struct basic_string {
        union {
                char _Buf[16];
                wchar_t _BufW[8];
                char * _Ptr;
                wchar_t *_PtrW;
        } _Bx;
        size_t _Mysize;
        size_t _Myres;
};
```

```
00000000 basic_string    struc ; (size
00000000 _Bx             _Bxty ?
00000010 _Mysize         dd ?
00000014 _Myres          dd ?
00000018 basic_string    ends
00000018
00000000 ; ---------------------------
00000000
00000000 _Bxty           union ; (size
00000000
00000000 _Buf            db 16 dup(?)
00000000 _BufW           dw 8 dup(?)
00000000 _Ptr            dd ?
00000000 _PtrW           dd ?
00000000 _Bxty           ends
```


46

basic_string recognition

- Evalutes size versus embedded buffer
  - Accounts for terminating null
  - Returns pointer if greater
  - Size used indicates ASCII vs Wide
- ASCII vs. Wide strings:
  - 0x10: ASCII
  - 8: Unicode →



47

FireEye

STL vector in VC

Simple as 1 2 3

48

vector recognition – Before/After

```
v31 = &v13;
v19 = a1;
v14 = 0;
v16 = 0;
v17 = 0;
v18 = 0;
v32 = 0;
if ( !a2 || (v4 = *a2, v20 = 0, (*(int (__stdcall **)(int *, _DWORD, int *))(v4 + 0x10))(a2,
{
  *a1 = 0;
  a1[1] = 0;
  a1[2] = 0;
}
else
{
  v5 = v20;
  if ( v20 )
  {
    v27 = 3;
    v28 = 0x3001001F;
    v29 = 0xFFF0102;
    v30 = 0x3400000B;
    v6 = (*(int (__stdcall **)(int, int *, _DWORD))(*(_DWORD *)v20 + 0x1C))(v20, &v27, 0) ==
    v7 = v20;
    if ( v6 )
    {
      v15 = 0;
      (*(void (__stdcall **)(int, _DWORD, unsigned int *))(*(_DWORD *)v20 + 0x24))(v20, 0, &v
      if ( v15 )
      {
        LOBYTE(v32) = 1;
        if ( v15 > 0x7FFFFFF )
          sub_4AEA18("vector<T> too long");
        sub_415B70(&v16, v15);
        v32 = 0;
        lpRows = 0;
        while ( !(*(int (__stdcall **)(int, signed int, _DWORD, LPSRowSet *))(*(_DWORD *)v20
                  v20,
                  1,
                  0,
```

```
v23 = &ulInterfaceOptions;
pIMsgStore = (struct IMsgStore *)vect_CMessageStores;
v13 = 0;
vect_docparsers_andor_wstrs._Myfirst = 0;
vect_docparsers_andor_wstrs._Mylast = 0;
vect_docparsers_andor_wstrs._Myend = 0;
v24 = 0;
if ( !lpMapiSession
  || (MapiSessionVtbl = lpMapiSession->lpVtbl,
      pMapiTable = 0,
      MapiSessionVtbl->GetMsgStoresTable(lpMapiSession, 0, (LPMAPITABLE *)&pMapiTable)) )
{
  vect_CMessageStores->_Myfirst = 0;
  vect_CMessageStores->_Mylast = 0;
  vect_CMessageStores->_Myend = 0;
}
else
{
  if ( pMapiTable )
  {
    PropTagArray.cValues = 3;
    PropTagArray.aulPropTag[0] = PR_DISPLAY_NAME_W;
    aulPropTag1 = PR_ENTRYID;
    aulPropTag2 = PR_DEFAULT_STORE;
    is_S_OK = pMapiTable->lpVtbl->SetColumns((#301 *)pMapiTable, &PropTagArray, 0) == S_OK;
    pMapiTable_alias2 = pMapiTable;
    if ( is_S_OK )
    {
      lpulCount = 0;
      pMapiTable->lpVtbl->GetRowCount((#301 *)pMapiTable, 0, (ULONG *)&lpulCount);
      if ( lpulCount )
      {
        LOBYTE(v24) = 1;
        if ( (unsigned int)lpulCount > 0x7FFFFFF )
          XLength((int)"vector<T> too long");
        vector_parser_doc_path_data_alloc_realloc(&vect_docparsers_andor_wstrs._Myfirst, (
        v24 = 0;
        lppRows = 0;
        while ( !pMapiTable->lpVtbl->QueryRows((#301 *)pMapiTable, 1, 0, &lppRows) )
        {
```


49

vector recognition – Adjacent Member Initialization

```
v31 = &v13;
v19 = a1;
v14 = 0;
v16 = 0;
v17 = 0;
v18 = 0;
v32 = 0;
if ( !a2 || (v4 = *a2, v20 = 0, (*(int (__stdcall **)(int *, _DWORD, int *))(v4 + 0x10))(a2,
{
  *a1 = 0;
  a1[1] = 0;
  a1[2] = 0;
}
else
{
  v5 = v20;
  if ( v20 )
  {
    v27 = 3;
    v28 = 0x3001001F;
    v29 = 0xFFF0102;
    v30 = 0x3400000B;
    v6 = (*(int (__stdcall **)(int, int *, _DWORD))(*(_DWORD *)v20 + 0x1C))(v20, &v27, 0) ==
    v7 = v20;
    if ( v6 )
    {
      v15 = 0;
      (*(void (__stdcall **)(int, _DWORD, unsigned int *))(*(_DWORD *)v20 + 0x24))(v20, 0, &v
      if ( v15 )
      {
        LOBYTE(v32) = 1;
        if ( v15 > 0x7FFFFFF )
          sub_4AEA18("vector<T> too long");
        sub_415B70(&v16, v15);
        v32 = 0;
        lpRows = 0;
        while ( !(*(int (__stdcall **)(int, signed int, _DWORD, LPSRowSet *))(*(_DWORD *)v20
                  v20,
                  1,
                  0,
```

```
v23 = &ulInterfaceOptions;
pIMsgStore = (struct IMsgStore *)vect_CMessageStores;
v13 = 0;
vect_docparsers_andor_wstrs._Myfirst = 0;
vect_docparsers_andor_wstrs._Mylast = 0;
vect_docparsers_andor_wstrs._Myend = 0;
v24 = 0;
if ( !lpMapiSession
  || (MapiSessionVtbl = lpMapiSession->lpVtbl,
      pMapiTable = 0,
      MapiSessionVtbl->GetMsgStoresTable(lpMapiSession, 0, (LPMAPITABLE *)&pMapiTable)) )
{
  vect_CMessageStores->_Myfirst = 0;
  vect_CMessageStores->_Mylast = 0;
  vect_CMessageStores->_Myend = 0;
}
else
{
  if ( pMapiTable )
  {
    PropTagArray.cValues = 3;
    PropTagArray.aulPropTag[0] = PR_DISPLAY_NAME_W;
    aulPropTag1 = PR_ENTRYID;
    aulPropTag2 = PR_DEFAULT_STORE;
    is_S_OK = pMapiTable->lpVtbl->SetColumns((#301 *)pMapiTable, &PropTagArray, 0) == S_OK;
    pMapiTable_alias2 = pMapiTable;
    if ( is_S_OK )
    {
      lpulCount = 0;
      pMapiTable->lpVtbl->GetRowCount((#301 *)pMapiTable, 0, (ULONG *)&lpulCount);
      if ( lpulCount )
      {
        LOBYTE(v24) = 1;
        if ( (unsigned int)lpulCount > 0x7FFFFFF )
          XLength((int)"vector<T> too long");
        vector_parser_doc_path_data_alloc_realloc(&vect_docparsers_andor_wstrs._Myfirst, (
        v24 = 0;
        lppRows = 0;
        while ( !pMapiTable->lpVtbl->QueryRows((#301 *)pMapiTable, 1, 0, &lppRows) )
        {
```


50

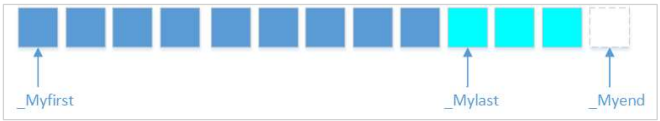
vector recognition – Indirect Member Initialization

```
v31 = &v13;
v19 = a1;
v14 = 0;
v16 = 0;
v17 = 0;
v18 = 0;
v32 = 0;
if ( !a2 || (v4 = *a2, v20 = 0, (*(int (__stdcall **)(int *, _DWORD, int *))(v4 + 0x10))(a2,
{
  *a1 = 0;
  a1[1] = 0;
  a1[2] = 0;
}
else
{
  v5 = v20;
  if ( v20 )
  {
    v27 = 3;
    v28 = 0x3001001F;
    v29 = 0xFFF0102;
    v30 = 0x3400000B;
    v6 = (*(int (__stdcall **)(int, int *, _DWORD))(*(_DWORD *)v20 + 0x1C))(v20, &v27, 0) ==
    v7 = v20;
    if ( v6 )
    {
      v15 = 0;
      (*(void (__stdcall **)(int, _DWORD, unsigned int *))(*(_DWORD *)v20 + 0x24))(v20, 0, &v
      if ( v15 )
      {
        LOBYTE(v32) = 1;
        if ( v15 > 0x7FFFFFF )
          sub_44E418("vector<T> too long");
        sub_415B70(&v16, v15);
        v32 = 0;
        lpRows = 0;
        while ( !(*(int (__stdcall **)(int, signed int, _DWORD, LPSRowSet *))(*(_DWORD *)v20
                  v20,
                  1,
                  0,
```

```
v23 = &ulInterfaceOptions;
pIMsgStore = (struct IMsgStore *)vect_CMessageStores;
v13 = 0;
vect_docparsers_andor_wstrs._Myfirst = 0;
vect_docparsers_andor_wstrs._Mylast = 0;
vect_docparsers_andor_wstrs._Myend = 0;
v24 = 0;
if ( !lpMapiSession
  || (MapiSessionUtbl = lpMapiSession->lpVtbl,
      pMapiTable = 0,
      MapiSessionUtbl->GetMsgStoresTable(lpMapiSession, 0, (LPMAPITABLE *)&pMapiTable)) )
{
  vect_CMessageStores->_Myfirst = 0;
  vect_CMessageStores->_Mylast = 0;
  vect_CMessageStores->_Myend = 0;
}
else
{
  if ( pMapiTable )
  {
    PropTagArray.cValues = 3;
    PropTagArray.aulPropTag[0] = PR_DISPLAY_NAME_W;
    aulPropTag1 = PR_ENTRYID;
    aulPropTag2 = PR_DEFAULT_STORE;
    is_S_OK = pMapiTable->lpVtbl->SetColumns((#301 *)pMapiTable, &PropTagArray, 0) == S_OK;
    pMapiTable_alias2 = pMapiTable;
    if ( is_S_OK )
    {
      lpulCount = 0;
      pMapiTable->lpVtbl->GetRowCount((#301 *)pMapiTable, 0, (ULONG *)&lpulCount);
      if ( lpulCount )
      {
        LOBYTE(v24) = 1;
        if ( (unsigned int)lpulCount > 0x7FFFFFF )
          XLength((int)"vector<T> too long");
        vector_parser_doc_path_data_alloc_realloc(&vect_docparsers_andor_wstrs._Myfirst, (
        v24 = 0;
        lppRows = 0;
        while ( !pMapiTable->lpVtbl->QueryRows((#301 *)pMapiTable, 1, 0, &lppRows) )
        {
```


51

vector – Three Members




```
/* vector struct definition for IDA */
struct vector {
        void *_Myfirst;          // pointer to beginning of array
        void *_Mylast;           // pointer to current end of sequence
        void *_Myend;            // pointer to end of array
};
```

```
00000000 vector          struc ; (sizeof=0xC, mappedto_1807)
00000000 _Myfirst        dd ?
00000004 _Mylast         dd ?
00000008 _Myend          dd ?
0000000C vector          ends
```


52

Hints of Vector Usage

- Adjacent variables subtracted from each other
  - Size: _Mylast - _Myfirst
  - Unused Capacity: _MyEnd - _Mylast
- Comparison of pointer difference with potential `sizeof(value type)`

```
v5 = v90;
v6 = v91 - (_DWORD)v90;
if ( (unsigned int)(v91 - (_DWORD)v90) >= 0x20 )
{
  lpMem = 0;
  v114 = 0;
```


53

More Hints of Vector Usage

- Address of variable passed to function containing:
  - `"vector<T> too long"` ☺
  - Subtraction among `*var`, `var[1]`, and/or `var[2]`
  - Division or Shift operations to scale pointer differences by value type size
  - Large constants that divide (almost) evenly into 0xffffffff

```
v2 = this;
v3 = this[1];
v4 = a2;
if ( a2 >= v3 || (v5 = *this, *this > a2) )
{
  v15 = this[2];
  if ( v3 == v15 && (unsigned int)((int)(v15 - v3) >> 5) < 1 )
  {
    v16 = (int)(v3 - *this) >> 5;
    if ( (unsigned int)(0x7FFFFFF - v16) <
      sub_44E418("vector<T> too long");
    v17 = v16 + 1;
    v18 = (int)(v15 - *this) >> 5;
```

Output window

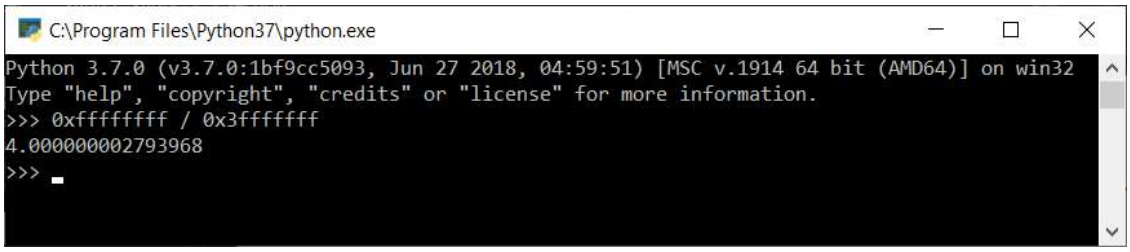
Python>0xffffffff/0x7ffffff
0x20

Python

54

Inferring `vector` Value Type Size – `max_size()` method

- `allocator<blah>::max_size() { return 0x3FFFFFFF; }`
- This is `0xffffffff / sizeof(_Ty)`
- In this case, 4



55

Inferring `vector` Value Type Size

- Problem: how big is struct?

```
if ( v18 )
{
  struct_instack_unknown_size = v18;
  v22 = 0;
  v9 = lpRows->aRow[0].lpProps;
  if ( LOWORD(v9[2].ulPropTag) == 11 )
    v22 = v9[2].Value.i != 0;
  v10 = lpRows->aRow[0].lpProps;
  if ( LOWORD(v10->ulPropTag) == 31 )
    sub_40D9F0(v10->Value.lpszA);
  vect_func(&vec_unknown_elmt_size, (unsigned int)&struct_instack_unknown_size);
}
FreeProws(lpRows);
LOBYTE(v31) = 0;
lpRows = 0;
if ( v25 >= 8 )
  sub_40F3C0(lpMem, v25 + 1);
```


56

Inferring `vector` Value Type Size

- Solution: 32 bytes

```
void __thiscall vect_func(vector *this, unsigned int struct_instack_unknown_size)
{
  // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL-"+" TO EXPAND]

  v2 = this;
  _Mylast = this->_Mylast;
  struct_alias = struct_instack_unknown_size;
  if ( struct_instack_unknown_size >= _Mylast || (v5 = this->_Myfirst, this->_Myfirst > struct_instack_unknown_size) )
  {
    _Myend = this->_Myend;
    if ( _Mylast == _Myend && (unsigned int)((int)(_Myend - _Mylast) >> 5) < 1 )
    {
      elements = (int)(_Mylast - this->_Myfirst) >> 5;
      if ( (unsigned int)(0x7FFFFFF - elements) < 1 )
        sub_44E418("vector<T> too long");
```


57

Inferring `vector` Value Type Size

- Use inferred value type size info in the calling function to

```
-0000004C var_4C          dd ?
-00000048 lpRows          dd ?                    ; offset
-00000044 struct_instack_unknown_size dd ?
-00000040 var_40
-0000003F
-0000003E
-0000003D
-0000003C lpMem
-00000038
-00000037
-00000036
-00000035
-00000034
-00000033
-00000032
-00000031
-00000030
```

Copy Ctrl+C
Undo Struct var... Ctrl+Z
Redo Ctrl+Y
Data D
Struct var... Alt+Q
String
Array... Numpad+*
Undefine U
Rename N
Set type... Y
Create struct from selection

Declare a structure variable

Structures

```
00000000 st_20h           struc ; (sizeof=0x20, r
00000000 field_0          dd ?
00000004 field_4          dd ?
00000008 field_8          dd ?
0000000C field_C          dd ?
00000010 field_10         dd ?
00000014 field_14         dd ?
00000018 field_18         dd ?
0000001C field_1C         dd ?
00000020 st_20h           ends
```

3. st_20h:000C

58

Inferring `vector` Value Type Size

- Xrefs crawling out of the woodwork:

```
if ( v16 )
{
  struct_instack_unknown_size.field_0 = (int)v16;
  LOBYTE(struct_instack_unknown_size.field_4) = 0;
  v9 = lpRows->aRow[0].lpProps;
  if ( LOWORD(v9[2].ulPropTag) == 11 )
    LOBYTE(struct_instack_unknown_size.field_4) = v9[2].Value.i != 0;
  v10 = lpRows->aRow[0].lpProps;
  if ( LOWORD(v10->ulPropTag) == 31 )
    sub_40D9F0(v10->Value.lpszA);
  vect_func(&vec_unknown_elmt_size, (unsigned int)&struct_instack_unknown_size);
}
FreeProws(lpRows);
LOBYTE(v25) = 0;
lpRows = 0;
if ( struct_instack_unknown_size.field_1C >= 8u )
  sub_40F3C0((LPVOID)struct_instack_unknown_size.field_8, struct_instack_unknown_size.field_1C + 1);
```


59

Inferring `vector` Value Type Size (another example)

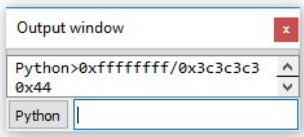
```
unsigned int __thiscall vect18h_func0(vector *this, int a2)
{
  vector *v2; // ebx
  int myend; // ecx
  int mylast; // esi
  unsigned int result; // eax
  int v6; // edi
  unsigned int v7; // edx
  unsigned int v8; // ecx
  unsigned int v9; // edi
  unsigned int v10; // esi

  v2 = this;
  myend = this->_Myend;
  mylast = v2->_Mylast;
  result = (myend - mylast) / 0x18;
```


60

Inferring `vector` Value Type Size (another example)

```
1 unsigned int __thiscall vect44h_func(vector *this, int a2)
2 {
3   vector *this_alias; // ebx
4   int myend; // ecx
5   int mylast; // esi
6   unsigned int _Unused_capacity; // eax
7   int oldsize; // edi
8   unsigned int v7; // edx
9   unsigned int v8; // ecx
10  unsigned int v9; // edi
11  unsigned int v10; // esi
12
13  this_alias = this;
14  myend = this->_Myend;
15  mylast = this_alias->_Mylast;
16  _Unused_capacity = (myend - mylast) / 0x44;
17  if ( _Unused_capacity < 1 )
18  {
19    oldsize = (mylast - this_alias->_Myfirst) / 0x44;
20    if ( (unsigned int)(0x3C3C3C3 - oldsize) < 1 )
21      XLength((int)"vector<T> too long");
```



61

Inferring `vector` Value Type Size from `size()` Method

```
int __thiscall vector_size_array44h(vector *this)
{
  return (this->_Mylast - this->_Myfirst) / 0x44;
}
```


62

Inferring `vector` Value Type Size From `size()` method

```
int __thiscall std::vector<std::basic_string<char,std::char_traits<char>
{
  return (this->_Mylast - this->_Myfirst) / (signed int)sizeof(string);
}
```


63

FireEye

STL `map` and `set` in VC

Extra credit

64

Node Construction

```
_DWORD *sub_407880()
{
  _DWORD *result; // eax

  result = operator new(0x34u);
  if ( result )
    *result = result;
  if ( result != (_DWORD *)-4 )
    result[1] = result;
  if ( result != (_DWORD *)-8 )
    result[2] = result;
  *((_WORD *)result + 6) = 257;
  return result;
}
```

```
_Node34h *new_Node_size34h()
{
  _Node34h *result; // eax

  result = (_Node34h *)operator new(0x34u);
  if ( result )
    result->_NodeHdr._Left = (int)result;
  if ( result != (_Node34h *)0xFFFFFFFC )
    result->_NodeHdr._Parent = (int)result;
  if ( result != (_Node34h *)0xFFFFFFF8 )
    result->_NodeHdr._Right = (int)result;
  *(_WORD *)&result->_NodeHdr._Color = 0x101;
  return result;
}
```


65

Composing `struct _Node`

- Node standard fields

```
00000000 _Node          struc ; (sizeof=0x10,
00000000 _Left          dd ?
00000004 _Right         dd ?
00000008 _Parent        dd ?
0000000C _Color         db ?
0000000D _Isnil         db ?
0000000E padding        dw ?
00000010 _Node          ends
```

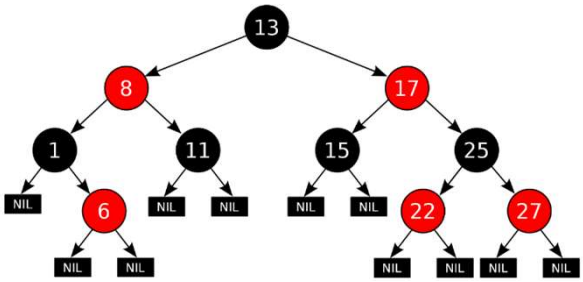
- Pair is just two adjacent objects
- Ex: `pair<basic_string, int>`

```
00000000 pair_xstring_int struc ; (sizeof=0x1C,
00000000
00000000 xstr           xstring ?
00000018 n              dd ?
0000001C pair_xstring_int ends
```

66

Node Structure for STL `map` and `set`

```
/* node struct definition for IDA */
struct _Node {
    void *_Left;
    void *_Parent;
    void *_Right;
    char _Color;
    char _Isnil;
    short padding;
};
```



67

Defining Structs for STL `map`

Here's what this looks like in IDA

- Define **value types** (e.g. string)
- Define **node struct** separately
- Define **key/value pair type**(s)
- Define **composite node type**(s)
  - Disambiguate at least by size

```
00000000 basic_string    struc ; (sizeof=0x18,
00000000
00000000 _Bx             basic_string::$DC411C0
00000010 _Mysize         dd ?
00000014 _Myres          dd ?
00000018 basic_string    ends
00000018
00000000 ; ------------------------------------
00000000
00000000 _Node           struc ; (sizeof=0x10,
00000000
00000000 _Left           dd ?
00000004 _Parent         dd ?
00000008 _Right          dd ?
0000000C _Color          db ?
0000000D _Isnil          db ?
0000000E padding         dw ?
00000010 _Node           ends
00000010
00000000 ; ------------------------------------
00000000
00000000 pair_xstr_n     struc ; (sizeof=0x1C,
00000000
00000000 str             basic_string ?
00000018 n               dd ?
0000001C pair_xstr_n     ends
0000001C
00000000 ; ------------------------------------
00000000
00000000 _Node_size2Ch   struc ; (sizeof=0x2C,
00000000 node            _Node ?
00000010 pair_wstr_n     pair_xstr_n ?
0000002C _Node_size2Ch   ends
```


68

FireEye

Tips for Analyzing Large STL Samples

69

Hit F5

- Decomp is best for C++
- Use it if ya got it
- Also, try pressing Enter…

```
768  // ////////////////////////////////////////////////////////////////
769  // Self-delete 1/4 in WinMain (many more elsewhere)
770  // ////////////////////////////////////////////////////////////////
771  selfdel(0);
772  _loaddll(0);
773proceed:
774  // ////////////////////////////////////////////////////////
775  // Proceed (no self-del yet)
776  // ////////////////////////////////////////////////////////
777  std::vector<char,std::allocator<char>>::_Tidy(&vectorx);
778  if ( v398 >= 8 )
779    std::_Wrap_alloc<std::allocator<wchar_t>>::deallocate(pwszObjectName[0], v398 + 1);
780  v398 = 7;
781  v397 = 0;
782  LOWORD(pwszObjectName[0]) = 0;
783  connection_cleanup(HttpServerConnectionCtx2);
784
785
786  // ////////////////////////////////////////////////////////
787  // Parse HTTP response
788  // Parse collect_params:
789  //  - type
790  //  - from_ab
791  //  - from_in_box
792  //  - from_out_box
793  //  - from_other
794  // ////////////////////////////////////////////////////////
795  vector_from_response._Myfirst = 0;
796  vector_from_response._Mylast = 0;
797  vector_from_response._Myend = 0;
798  LOBYTE(phase) = 0x13;
799  if ( parse_http_response(&vector_from_response, &outbuf_http_response_html) != 1
800    || parse_collect_params_X_from_ab_X_in_box_X_other_X_out_box(&vector_from_response, &flags_01010101h) != 1 )
801  {
802    flags_01010101h = 0x1010101;
803  }
804  outlook_version._Myres = 7;
805  outlook_version._Mysize = 0;
806  LOWORD(outlook_version._Bx._Ptr) = 0;
```


70

Hex-Rays Hot Tip

- `__thiscall` and `__fastcall` blind spots:
  - Hex-Rays doesn't always notice implicit `this` argument (or GPR usage)
  - Deceptive when using xrefs
  - Quick fix:
    - Go to disassembly
    - Compare xrefs + identify function call sites
    - Return to decompilation
    - Enter and exit missing functions
    - Other times, have to confirm and force calling convention for various functions

71

STL Nomenclature Suggestions

- Large samples → same containers, multiple value types
- Name structs, variables, methods after `sizeof(value_type)`/key/value
- **`vector`**: `vect24h_emails`, `vect20h_message_stores`
- **`map`**: `_Node34h_email_srctype // Node, size 34h: unique email → src type`

72

FireEye

Gophe Summary

73

Gophe Summary (MD5: D9630C174B8FF5C0AA26168DF523E63E)

Overall

- Beacon (`collect_params` phase)
- Scrapes emails (MAPI):
  - Contacts list
  - Received folder
  - Sent Items
  - Folder Hierarchy
- Beacon (`data` phase)
- Parses config further
- Sends spam emails (MAPI)
- Beacon (`work_report` phase)

Binaries

- `MailClient5.exe` – main sample
  - 64-bit MAPI email scraper
  - 64-bit MAPI SPAM email sender
  - `nprt_copy.dll` (Mozilla NPAPI plug-in)

Host Behaviors

- Obtains and uses user's email signature
- `PR_DELETE_AFTER_SUBMIT` deletes spam from sent items
- Can disable email alerts, hide (taskbar)
- ThunderBird XUL/XPCOM and NPAPI plug-ins

74

Gophe C2 (MD5: D9630C174B8FF5C0AA26168DF523E63E)

Host/Protocol

- HTTPS, 195.2.252[.]67:2050:
  - GET /<RandomA>
  - POST /<RandomA>.com=<RandomB>
  - User-Agent: Mozilla/5.0 (Windows NT 6.1; Trident/7.0; rv:11.0) like Gecko

Phases

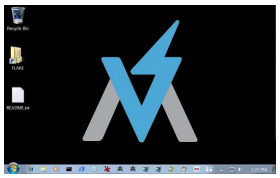
- collect_parms (solicits config)
- data (emails, semicolon delimited)
- work_report (survey/report)
  - Client-ConnectionId
  - Created
  - Total
  - Thunderbird-version
  - Thunderbird-plugins-installed: true|false

75

Questions?

Twitter: @mykill

FLARE-VM
Windows security distribution



FakeNet-NG
Internet Simulation for malware

FLOSS – Obfuscated string decoder

github.com/fireeye/

The F L A R E On Challenge

Reverse Engineering CTF 6 years running

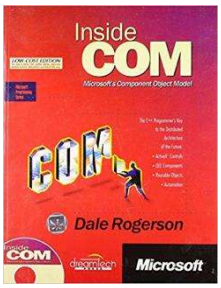
2019: 5,790 registered, 308 finished

www.flare-on.com

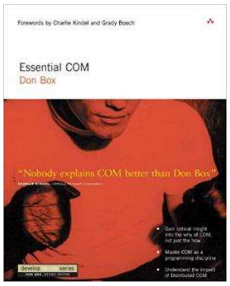
76

References

- **Dyre Malware Campaigners Innovate with Distribution Techniques**: https://www.proofpoint.com/us/threat-insight/post/dyre-malware-campaigners-innovate-distribution-techniques
- **Layout of a COM Object**: https://devblogs.microsoft.com/oldnewthing/20040205-00/?p=40733



Rogerson, D., **Inside COM**, MS Press, 1997



Box, D., **Essential COM**, Addison-Wesley, 1998

77

FireEye

Extra

78

STL Structure Round-Up

Strings

```
struct basic_string {
  union {
    char _Buf[16];
    wchar_t _BufW[8];
    char * _Ptr;
    wchar_t *_PtrW;
  } _Bx;
  size_t _Mysize;
  size_t _Myres;
};
```

Vectors

```
struct vector {
  void *_Myfirst;
  void *_Mylast;
  void *_Myend;
};
```

Maps/Sets

```
// Struct header only
struct _Node {
    void *_Left;
    void *_Parent;
    void *_Right;
    char _Color;
    char _Isnil;
    short padding;
};

// Your Mileage May Vary!
```


79

Other References

- **Registering Assemblies with COM:** https://docs.microsoft.com/en-us/dotnet/framework/interop/registering-assemblies-with-com
- **Programming Outlook.Application**: https://docs.microsoft.com/en-us/office/troubleshoot/office-developer/automate-outlook-2010-using-c-with-mfc
- **Building MAPI applications on 32-bit and 64-bit platforms**: https://docs.microsoft.com/en-us/office/client-developer/outlook/mapi/building-mapi-applications-on-32-bit-and-64-bit-platforms
- **MAPI and Thunderbird**: http://kb.mozillazine.org/MAPI_Support

80

Composing `struct _Tree`

- For completeness, here's `_Tree`:

```
00000000 _Tree          struc ; (sizeof=0x8,
00000000 _Myhead        dd ?
00000004 _Mysize        dd ?
00000008 _Tree          ends
```


81

`vector` struct (YMMV)

VS17 release

- Aux and allocator present

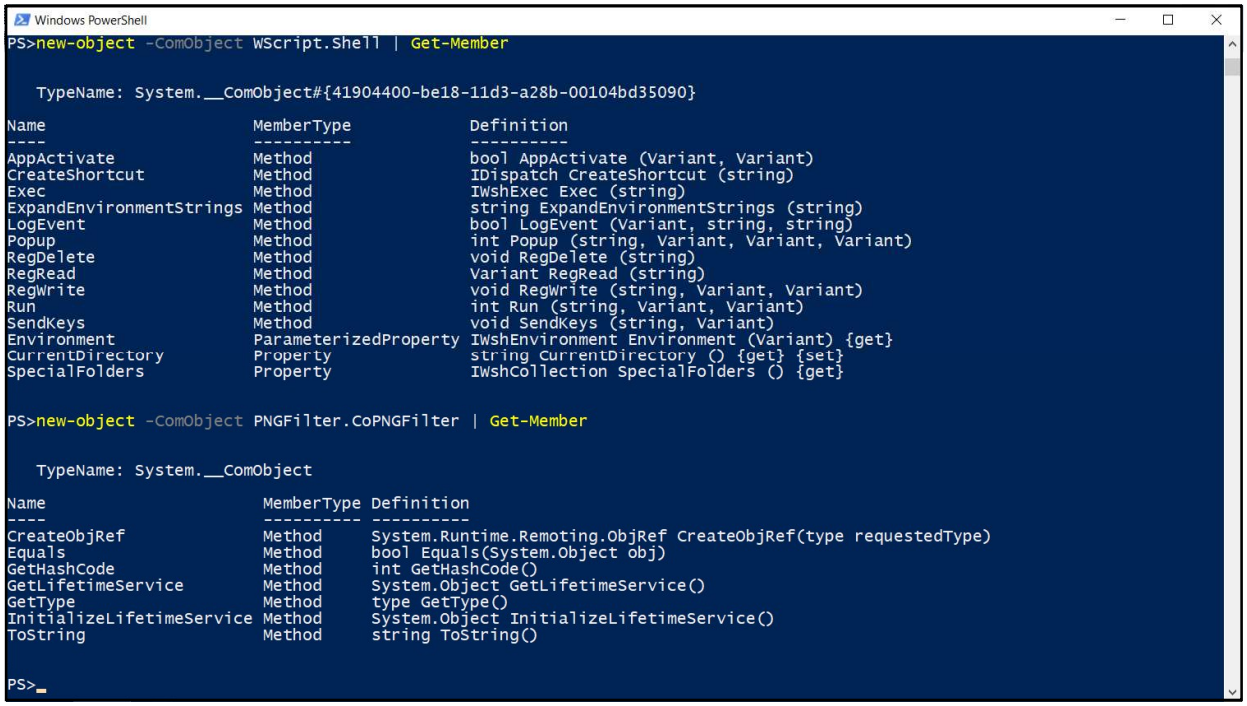
```
00000000 vector          struc ; (sizeof=0x18, mappedto_59)
00000000                                       ; XREF: _main/r
00000000 _Myownedaux     dd ?
00000004 _Aux2unknown    dd ?
00000008 _Alval          dd ?
0000000C _Myfirst        dd ?
00000010 _Mylast         dd ?
00000014 _Myend          dd ?
00000018 vector          ends
```

Unknown 2017 binary with STL+Boost

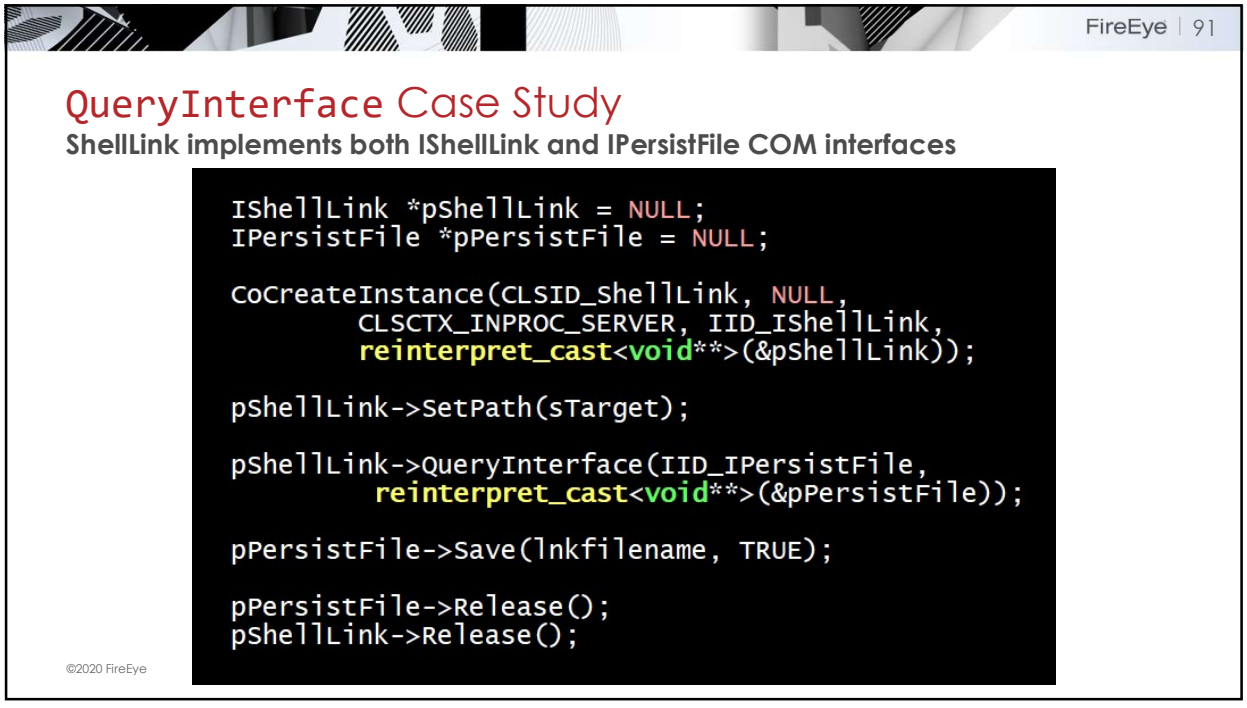
- Naked
  - No iterator debug
  - No _SECURE_SCL
  - No allocator fields

```
00000000 vector          struc ; (sizeof=0xC, mappedto_1807)
00000000 _Myfirst        dd ?
00000004 _Mylast         dd ?
00000008 _Myend          dd ?
0000000C vector          ends
```


82


Windows PowerShell
PS>new-object -ComObject WScript.Shell | Get-Member
TypeName: System.__ComObject#{41904400-be18-11d3-a28b-00104bd35090}
Name MemberType Definition
AppActivate Method bool AppActivate (Variant, Variant)
CreateShortcut Method IDispatch CreateShortcut (string)
Exec Method IWshExec Exec (string)
ExpandEnvironmentStrings Method string ExpandEnvironmentStrings (string)
LogEvent Method bool LogEvent (Variant, string, string)
Popup Method int Popup (string, Variant, Variant, Variant)
RegDelete Method void RegDelete (string)
RegRead Method Variant RegRead (string)
RegWrite Method void RegWrite (string, Variant, Variant)
Run Method int Run (string, Variant, Variant)
SendKeys Method void SendKeys (string, Variant)
Environment ParameterizedProperty IWshEnvironment Environment (Variant) {get}
CurrentDirectory Property string CurrentDirectory () {get} {set}
SpecialFolders Property IWshCollection SpecialFolders () {get}
PS>new-object -ComObject PNGFilter.CoPNGFilter | Get-Member
TypeName: System.__ComObject
Name MemberType Definition
CreateObjRef Method System.Runtime.Remoting.ObjRef CreateObjRef(type requestedType)
Equals Method bool Equals(System.Object obj)
GetHashCode Method int GetHashCode()
GetLifetimeService Method System.Object GetLifetimeService()
GetType Method type GetType()
InitializeLifetimeService Method System.Object InitializeLifetimeService()
ToString Method string ToString()
PS>

90


FireEye | 91
QueryInterface Case Study
ShellLink implements both IShellLink and IPersistFile COM interfaces
IShellLink *pShellLink = NULL;
IPersistFile *pPersistFile = NULL;
CoCreateInstance(CLSID_ShellLink, NULL,
CLSCTX_INPROC_SERVER, IID_IShellLink,
reinterpret_cast<void**>(&pShellLink));
pShellLink->SetPath(sTarget);
pShellLink->QueryInterface(IID_IPersistFile,
reinterpret_cast<void**>(&pPersistFile));
pPersistFile->Save(lnkfilename, TRUE);
pPersistFile->Release();
pShellLink->Release();
©2020 FireEye

91

vector recognition – Not All Initializations Adjacent

```
if ( vect_iterator_pMessageStore1->ptr_IMsgStore )
{
  contacts_iter = 0;
  zero = 0;
  vect_contact_emails._Myfirst = 0;
  vect_contact_emails._Mylast = 0;
  contacts_end = 0;
  vect_contact_emails._Myend = 0;
  LOBYTE(phase) = 2;

  if ( (_BYTE)flags == 1 )
```

```
0041BD41              xor     esi, esi
0041BD43              xor     edi, edi
0041BD45              mov     [ebp+vector_xxx._Myfirst], esi
0041BD4B              mov     [ebp+vector_xxx._Mylast], edi
0041BD51              mov     [ebp+vector_xxx._Myend], esi
0041BD57 ;   try {
0041BD57              mov     [ebp+phase], edi ; Not adjacent
0041BD5A              mov     [ebp+var_9F54], edi ; Not adjacent
0041BD60              test    eax, eax
0041BD62              jle     short loc_41BDA3
```


107

vector Recognition – Size of Value Type

- Where can you infer **sizeof(_Ty)** from below, and why?

```
131  size_unscaled = vect_MessageStores._Mylast - vect_MessageStores._Myfirst;
132  if ( (unsigned int)(vect_MessageStores._Mylast - vect_MessageStores._Myfirst) >= 0x20 )
133  {
134    map_or_set_2Ch_root? = 0;
135    v104 = 0;
136    node = new_2Ch_treenode_for_map_or_set_();
137    map_or_set_2Ch_root? = node;
138    v_size = size_unscaled >> 5;
```

- Recall:
  - `pointer _Myfirst; // pointer to beginning of array`
  - `pointer _Mylast;  // pointer to current end of sequence`
  - `pointer _Myend;   // pointer to end of array`
- Also: `size_type size()  { return _Mylast - _Myfirst; }`

108