BSides 2020
February 8, 2020
Michael Bailey

FLARE

Standard Template Library (STL)

Structure definitions

For `std::string`

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

For `std::vector`

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

For `std::set` and `std::map`

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

String

Comparison against 0x10: ASCII

Comparison against 8: Wide

Inferring Vector Value Type Size

If you see…	→	You should think:
`(this[1] - *this) / 0x44`	→	0x44
`(v2->_Mylast – v2->_Myfirst) / 0x44`	→	0x44
`(_Mylast – this->_Myfirst) >> 5`	→	2^5 = 0x20
`(v91 – v90) >= 0x20`	→	0x20
`return 0x3fffffff;`	→	0xffffffff / 0x3fffffff = 4

Map Data Type Example (Example: int -> string mapping)

Node Composition for `std::map`

```
struct pair_n_str {
    int n;
    string s;
}; // Size 0x18 on x86

struct Node2Ch_n_str {
    struct NodeHdr node;
    struct pair_n_str n_s;
}; // Size 0x2C on x86
```

Recognizing Tree Node Construction

```
Node2Ch *new_Node_size2Ch()
{
  Node2Ch *result;

  result = (Node2Ch *)operator new(0x2Cu);
  if ( result )
    result->NodeHdr._Left = (int)result;
  if ( result != (Node2Ch *)0xFFFFFFFC )
    result->NodeHdr._Parent = (int)result;
  if ( result != (Node2Ch *)0xFFFFFFF8 )
    result->NodeHdr._Right = (int)result;
  *(_WORD *)&result->NodeHdr._Color = 0x101;
  return result;
}
```

BSides 2020
February 8, 2020
Michael Bailey

FLARE

Component Object Model (COM)

Terminology and Identification Tactics

CoClass	COM class aka COM object aka COM server
Interface	Definition of how a Vtable of functions will be laid out
Vtable	Set of function pointers
ProgID	Friendly CoClass name, e.g. `SAPI.SpVoice` or `WScript.Dictionary`
CLSID	Class ID, a GUID identifying one CoClass
IID	Interface ID, a GUID identifying an interface

Identification Tactics:

1. IDA: set IID type to `CLSID`
2. Search HKCR
3. Search WinSDK headers

Hunting Typelibs:

- DLL/EXE itself, `*.tlb`, `*.olb`, `*.dll`
- `oleview`: File -> View TypeLib

Interfaces and Layout

IUnknown

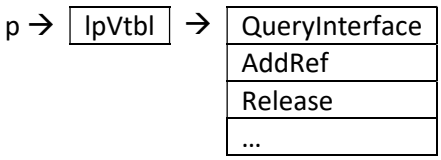
1. **QueryInterface**
2. **AddRef**
3. **Release**

(All COM objects)

IDispatch : IUnknown

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

(Automation objects)



Obtaining an Interface Pointer

```
CoCreateInstance(CLSID_Something, ..., IID_ISomething, ppv)

ppv->QueryInterface(IID_SomethingElse, ppv2)
```

COM Registration

Example: SAPI.SpVoice (InprocServer32)

ProgID to CLSID	[HKCR**SAPI.SpVoice**\CLSID] (Default) = **{96749377-3391-11D2-9EE3-00C04F797396}**
CLSID to DLL	[HKCR\clsid**{96749377-3391-11D2-9EE3-00C04F797396}**\InprocServer32] (Default) = **%SystemRoot%\System32\Speech\Common\sapi.dll**

Example: InternetExplorer.Application (LocalServer32)

ProgID to CLSID	[HKCR**InternetExplorer.Application**\CLSID] (Default) = **{0002DF01-0000-0000-C000-000000000046}**
CLSID to DLL	[HKCR\clsid**{0002DF01-0000-0000-C000-000000000046}**\LocalServer32] (Default) = **"C:\Program Files\Internet Explorer\IEXPLORE.EXE"**