TippingPoint

Reversing Microsoft DirectShow

And 3rd Party Codecs

Aaron Portnoy, TippingPoint DVLabs

Obligatory Introduction

TippingPoint
IPS-SECURED NETWORKS

- **TippingPoint DVLabs**
  - Security Research
  - Vulnerability Discovery
  - Vulnerability Exploitation
  - http://dvlabs.tippingpoint.com
    - Published and Upcoming Advisories
    - Blog
- **Zero Day Initiative**
  - Vulnerability Purchasing Program

Overview

TippingPoint
IPS-SECURED NETWORKS

- **Video For Windows**
  - Obsolete, will touch briefly on this
- **DirectShow**
  - Current, please follow along with your favorite disassembler…
- **Media Foundation**
  - The new hotness, will discuss if time permits

Why You Should Care

TippingPoint
IPS-SECURED NETWORKS

- **Bugs bugs bugs bugs bugs bugs bugs bugs**
  - Recent unpatched examples include…
    - MS DirectShow QuickTime 0day (MS Security Advisory 971778)
    - A Microsoft DirectShow 0day I discovered last week
  - Recent patched examples include…
    - My VMware Codec Vulns
      - Will demo a working exploit if we have time
    - ISS' Xvid Vulns (see Dowd/McDonald presentation at CSW08)
- **Extensible Framework**
  - Same mistakes are made in new codecs (OFTEN)
- **Client-side vulns are ubiquitous and fun to exploit**

DirectShow and AVI Primer

From The Top – RIFF-based Media

TippingPoint
IPS-SECURED NETWORKS

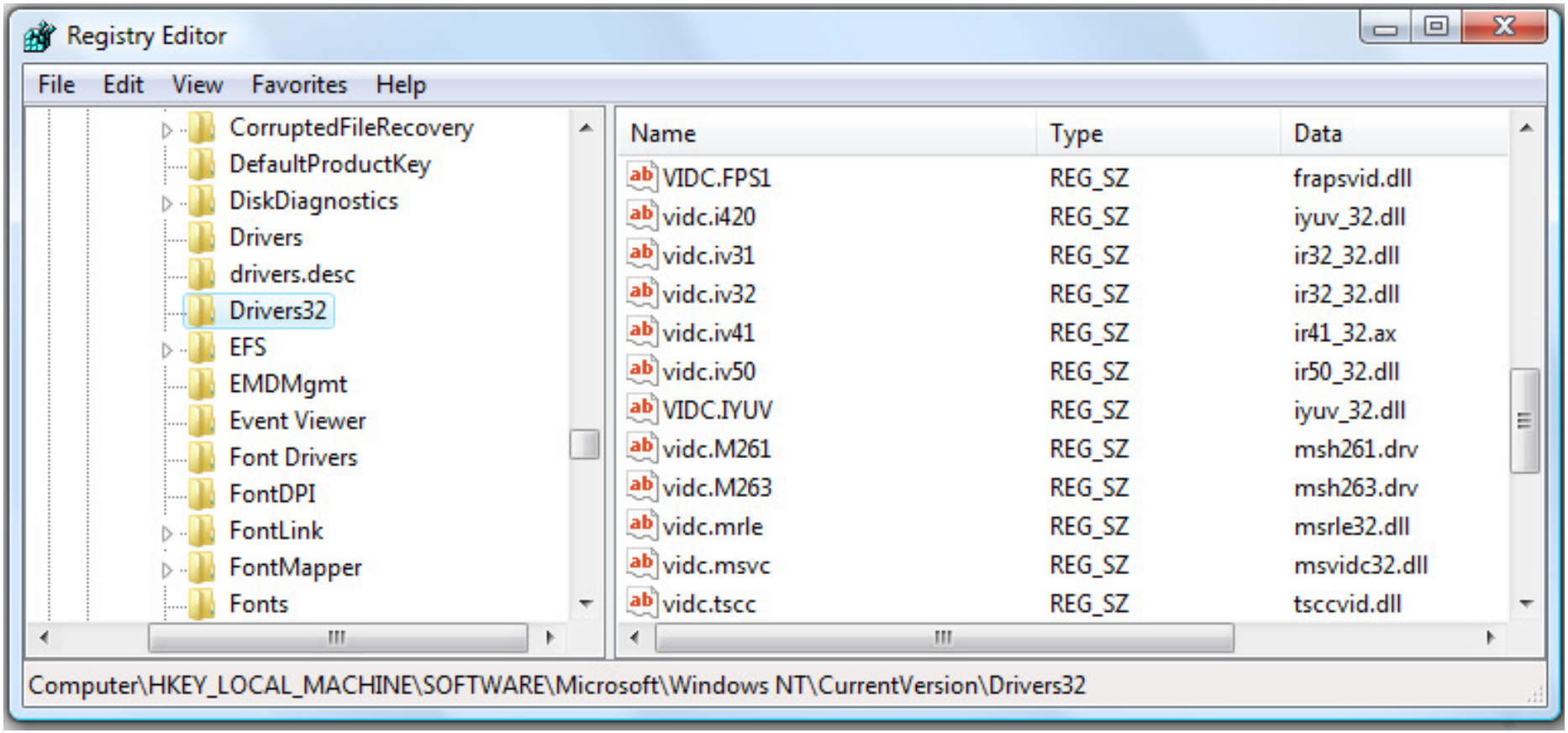
- **AVI for Video, WAVE for Audio**
  - Container format (RIFF)
  - Streams
  - Compression Formats (FOURCCs)

- **Compressors/Decompressors**
  - Third party code ftw
  - Audio and Video
  - Registered in…. the registry
    - HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Drivers32
  - Divx, Xvid, Cinepak, Indeo, …

Registry

TippingPoint
IPS-SECURED NETWORKS

- **Entries for Installable Compressors/Decompressors**



DirectShow Pins, Filters, and Graphs

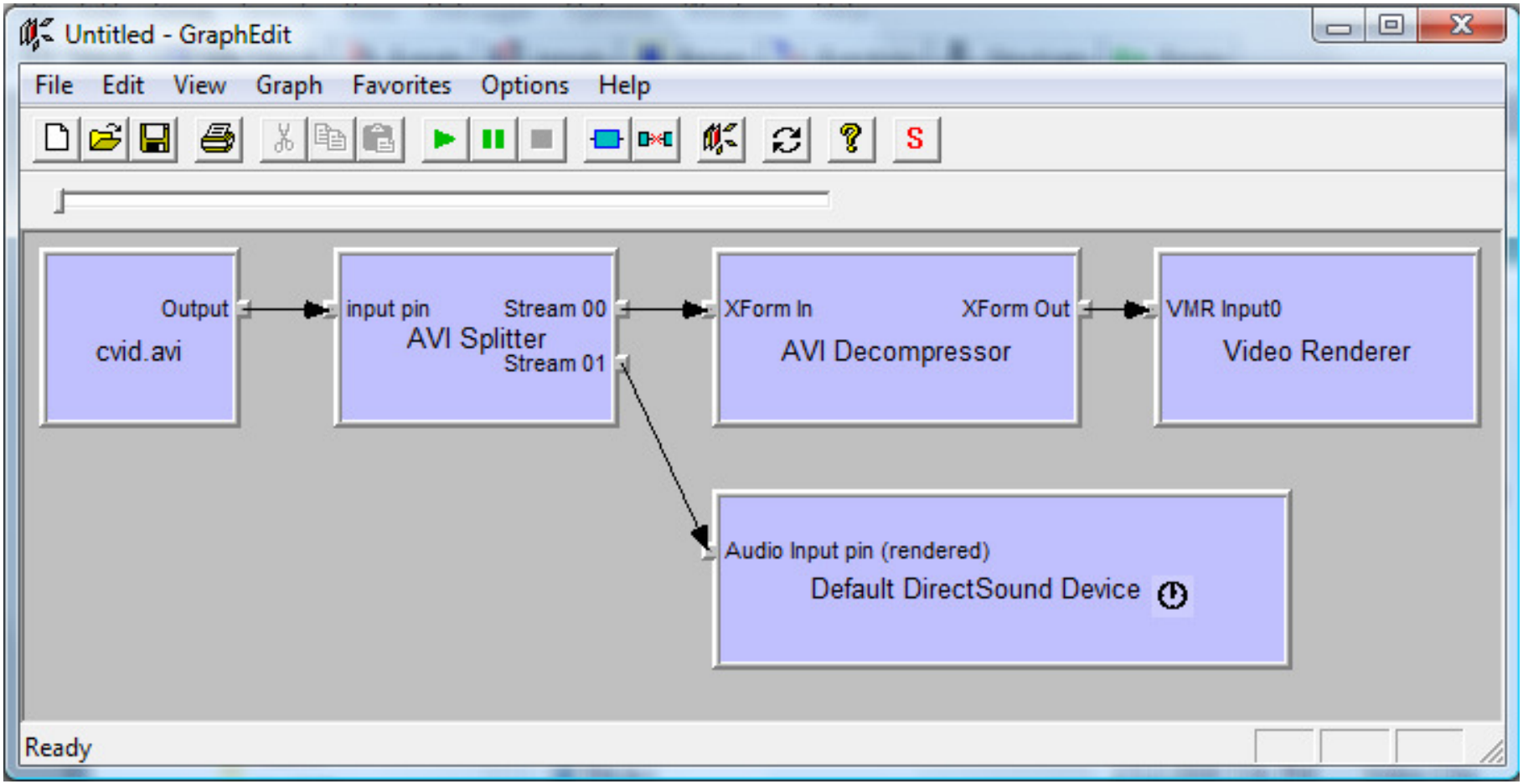
TippingPoint
IPS-SECURED NETWORKS

- **Filters**
  - A node on the filter graph
    - Responsible for one phase of processing
    - Source, Transform, Renderer
- **Pins**
  - Each filter has at least one Pin
    - Specifies which type of media the Filter can accept or produce
  - Can be either Input or Output
  - Codecs register Media Types they can handle
- **Filter Graphs**
  - Putting it all together, contains Filters and Pins

Filter Graph Example

TippingPoint
IPS-SECURED NETWORKS

- **Filter graph for rendering an AVI with a CVID stream**



So, How's it Really Work?

TippingPoint
IPS-SECURED NETWORKS

- **You open a media file and DirectShow…**
  - Creates a filter graph:

```
ole32!CoCreateInstance
quartz!CoCreateFilter+0x1a
quartz!CFilterGraph::CreateFilter+0x4c
quartz!CFilterGraph::AddSourceFilterInternal+0x199
quartz!CFilterGraph::RenderFile+0x77
```

  - Parses the RIFF chunks:

```
quartz!CAviMSRFilter::ParseHeaderCreatePins
quartz!CAviMSRFilter::LoadHeaderParseHeaderCreatePins+0xa5
quartz!CBaseMSRFilter::NotifyInputConnected+0x50
quartz!CBaseMSRInPin::CompleteConnect+0x3a
quartz!CBasePin::ReceiveConnection+0xc2
quartz!CBasePin::AttemptConnection+0x54
quartz!CBasePin::TryMediaTypes+0x64
quartz!CBasePin::AgreeMediaType+0x73
quartz!CBasePin::Connect+0x55
…
```

So, How's it Really Work? (cont.)

- Determines which codec can accept the media type
  - Specified as a GUID
  - AM_MEDIA_TYPE structure

```
quartz!CBaseMSRFilter::NotifyInputConnected+0x50
quartz!CBaseMSRInPin::CompleteConnect+0x3a
quartz!CBasePin::ReceiveConnection+0xc2
quartz!CBasePin::AttemptConnection+0x54
[loop here until a successful connection]
quartz!CBasePin::TryMediaTypes+0x64
quartz!CBasePin::AgreeMediaType+0x73
quartz!CBasePin::Connect+0x55
```

- Creates BITMAPINFOHEADER or WAVEFORMATEX
  - Passes this with a driver message, to the appropriate codec

- **Creation of filters can be tracked here**
  - bp quartz!CFilterGraph::CreateFilterAndNotify+0x32 "dd @esi ; .echo ; g"
  - The above breakpoint will dump the GUID shown in memory for that filter

Auditing Codecs

Codec Structure

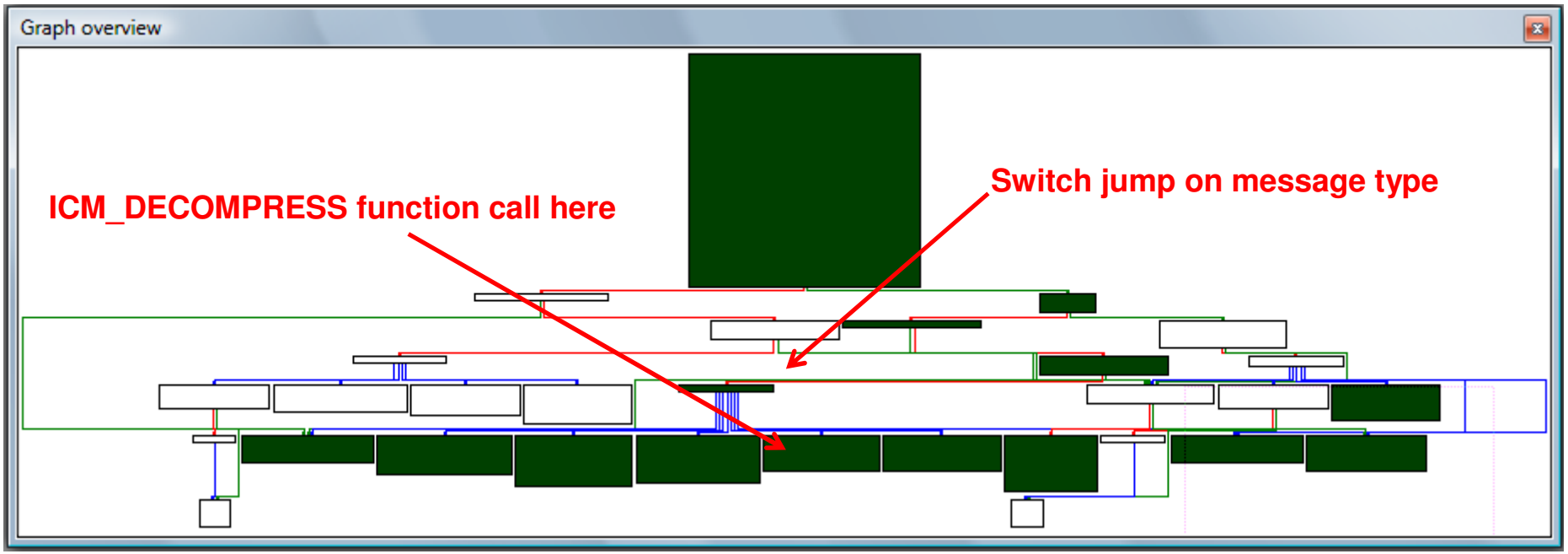
- **Implemented as Drivers**
  - Generally in \Windows\System32
  - DriverProc function exported
    - Called by quartz, receives driver messages
      - ICM_DECOMPRESS, ICM_GET_FORMAT, ICM_GET_INFO,…
    - Starting point to look for fun trust-boundary bugs
      - Track allocations here! (ICM_DECOMPRESS_BEGIN)
  - Accept data generically
    - BITMAPINFOHEADER and WAVEFORMATEX

- **Decompresses or compresses media data**
  - ICM_DECOMPRESS is where custom compressions are used
  - Useful debugging breakpoint for grabbing driver message #
    - bp codec!DriverProc "dd @esp+C L1;g"

DriverProc Graph

TippingPoint
IPS-SECURED NETWORKS

- Function graph for a generic codec's DriverProc



Writing a Fuzzing Codec

TippingPoint
IPS-SECURED NETWORKS

- **We can write our own proxy codec**
  - Implement DriverProc
  - Retrieve handle to the codec we want to fuzz

```
typedef LRESULT (CALLBACK *REMOTEDRIVERPROC)(DWORD, HDRVR, UINT, LPARAM, LPARAM);
REMOTEDRIVERPROC hRemoteDriverProc = NULL;

LRESULT CALLBACK DriverProc(DWORD dwID, HDRVR hDriver, UINT uiMessage, LPARAM
lParam1, LPARAM lParam2)
{

if (hRemoteDriverProc == NULL)
{
   HMODULE hRemoteModule;

   hRemoteModule = LoadLibrary("C:\\windows\\system32\\cvid.dll");

   hRemoteDriverProc = (REMOTEDRIVERPROC)GetProcAddress(hRemoteModule,
"DriverProc");

   assert(hRemoteDriverProc != NULL);
}
```

Writing a Fuzzing Codec (cont.)

TippingPoint
IPS-SECURED NETWORKS

- **Now we can fuzz data frame-by-frame**
  - Very fast, cool Screensaver
  - OutputDebugString is helpful, yay WinDBG

```
switch(uiMessage)
{

case ICM_DECOMPRESS:

   ICDECOMPRESS * icdecomp;
   BITMAPINFOHEADER *bmp;

   icdecomp = (ICDECOMPRESS *)lParam1;
   bmp = (BITMAPINFOHEADER *)icdecomp->lpbiInput;

   char foo[818];
   sprintf(foo, "old bitmap width is 0x%08x, new is 0x%08x\n", bmp->biWidth, -1);
   OutputDebugString(foo);
   bmp->biWidth = -1;

   // Proxy the modified data to the codec we want to fuzz via the handle
   return hRemoteDriverProc(dwID, hDriver, uiMessage, lParam1, lParam2);
}
```

Writing a Fuzzing Codec (cont.)

TippingPoint
IPS-SECURED NETWORKS

- **Problems**
  - False positives
    - DirectShow verifies certain data, thus a bug you find might not be reachable
- **DirectShow verifications**
  - quartz!ValidateBitmapInfoHeader
  - quartz!AMValidateAndFixWaveFormatEx
  - quartz!CImplStdAviIndex::ValidateStdIndex
  - quartz!CImplStdAviIndex::ValidateSuperIndex
- **Don't waste CPU cycles…**

```
quartz!ValidateBitmapInfoHeader+0x8b
   .text:6C239F38   call    MultiplyCheckOverflow(x,x,x) ; if biWidth
* biHeight * 200 wraps a DWORD you're busted here
```

redacted

redacted

TippingPoint

IPS-SECURED NETWORKS

redacted

TippingPoint

IPS-SECURED NETWORKS

redacted

TippingPoint

IPS-SECURED NETWORKS

redacted

TippingPoint

IPS-SECURED NETWORKS

redacted

TippingPoint

IPS-SECURED NETWORKS

redacted

TippingPoint

IPS-SECURED NETWORKS

redacted

TippingPoint

IPS-SECURED NETWORKS

redacted

TippingPoint

IPS-SECURED NETWORKS

redacted

redacted

TippingPoint

IPS-SECURED NETWORKS

redacted

TippingPoint

IPS-SECURED NETWORKS

redacted

TippingPoint

IPS-SECURED NETWORKS

redacted

TippingPoint

IPS-SECURED NETWORKS

redacted

TippingPoint

IPS-SECURED NETWORKS

Demo!

TippingPoint®

Thank You – Questions…?

dvlabs.tippingpoint.com
aportnoy [.at.] tippingpoint.com