ENISA Trainings
Windows malware analysis
Investigation - Infection #1

Section objectives

1. Analysis of network flows
2. Analysis of system (sysmon) logs

Is good to know

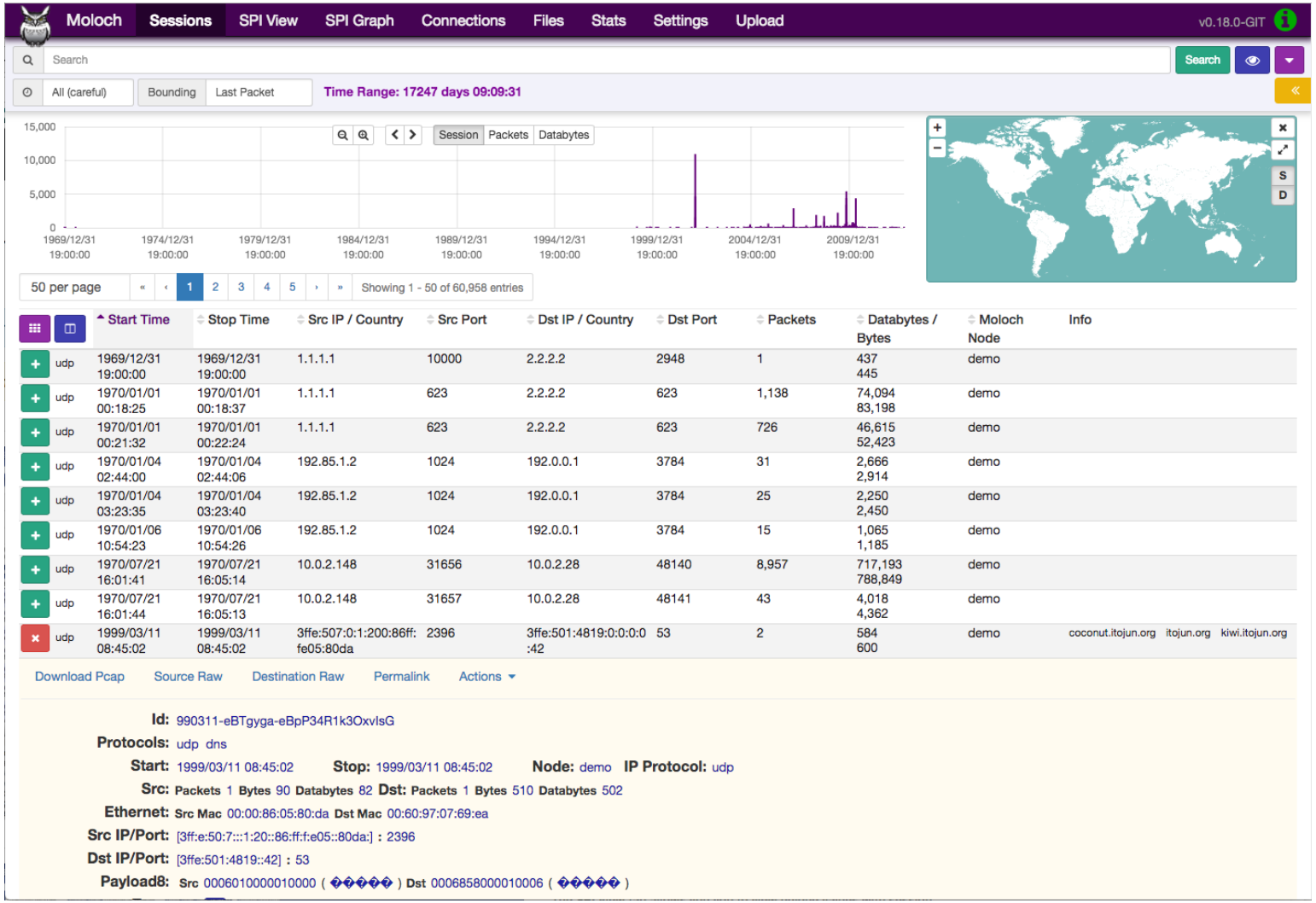
- Add all necessary information to Hive
- Most of operating system events are noise (in this case):
  - Performance data
  - OS start / shutdown events
  - WMI events
- Most of network traffic is a noise:
  - Service & routing protocols
  - Auto configuration protocols
  - Telemetry
  - Advertisement networks & external service providers (in Web traffic analysis case)

Context

1. Spear-phishing with malicious attachment: *Invoice_no_89685958466.pdf.exe*
2. We have only system logs & network traffic from infected workstation

Exercise #1
Network flow analysis

Moloch


Moloch
Sessions
SPI View
SPI Graph
Connections
Files
Stats
Settings
Upload
v0.18.0-GIT
Search
Search
All (careful)
Bounding
Last Packet
Time Range: 17247 days 09:09:31
15,000
10,000
5,000
0
Session
Packets
Databytes
1969/12/31 19:00:00
1974/12/31 19:00:00
1979/12/31 19:00:00
1984/12/31 19:00:00
1989/12/31 19:00:00
1994/12/31 19:00:00
1999/12/31 19:00:00
2004/12/31 19:00:00
2009/12/31 19:00:00
50 per page
Showing 1 - 50 of 60,958 entries
Start Time
Stop Time
Src IP / Country
Src Port
Dst IP / Country
Dst Port
Packets
Databytes / Bytes
Moloch Node
Info
udp 1969/12/31 19:00:00 1969/12/31 19:00:00 1.1.1.1 10000 2.2.2.2 2948 1 437 445 demo
udp 1970/01/01 00:18:25 1970/01/01 00:18:37 1.1.1.1 623 2.2.2.2 623 1,138 74,094 83,198 demo
udp 1970/01/01 00:21:32 1970/01/01 00:22:24 1.1.1.1 623 2.2.2.2 623 726 46,615 52,423 demo
udp 1970/01/04 02:44:00 1970/01/04 02:44:06 192.85.1.2 1024 192.0.0.1 3784 31 2,666 2,914 demo
udp 1970/01/04 03:23:35 1970/01/04 03:23:40 192.85.1.2 1024 192.0.0.1 3784 25 2,250 2,450 demo
udp 1970/01/06 10:54:23 1970/01/06 10:54:26 192.85.1.2 1024 192.0.0.1 3784 15 1,065 1,185 demo
udp 1970/07/21 16:01:41 1970/07/21 16:05:14 10.0.2.148 31656 10.0.2.28 48140 8,957 717,193 788,849 demo
udp 1970/07/21 16:01:44 1970/07/21 16:05:13 10.0.2.148 31657 10.0.2.28 48141 43 4,018 4,362 demo
udp 1999/03/11 08:45:02 1999/03/11 08:45:02 3ffe:507:0:1:200:86ff:fe05:80da 2396 3ffe:501:4819:0:0:0:0:42 53 2 584 600 demo coconut.itojun.org itojun.org kiwi.itojun.org
Download Pcap
Source Raw
Destination Raw
Permalink
Actions
Id: 990311-eBTgyga-eBpP34R1k3OxvIsG
Protocols: udp dns
Start: 1999/03/11 08:45:02 Stop: 1999/03/11 08:45:02 Node: demo IP Protocol: udp
Src: Packets 1 Bytes 90 Databytes 82 Dst: Packets 1 Bytes 510 Databytes 502
Ethernet: Src Mac 00:00:86:05:80:da Dst Mac 00:60:97:07:69:ea
Src IP/Port: [3ff:e:50:7:::1:20::86:ff:fe05::80da:] : 2396
Dst IP/Port: [3ffe:501:4819::42] : 53
Payload8: Src 0006010000010000 ( ) Dst 0006858000010006 ( )

Moloch

Moloch is an open source, large scale, full packet capturing, indexing, and database system.

Features:

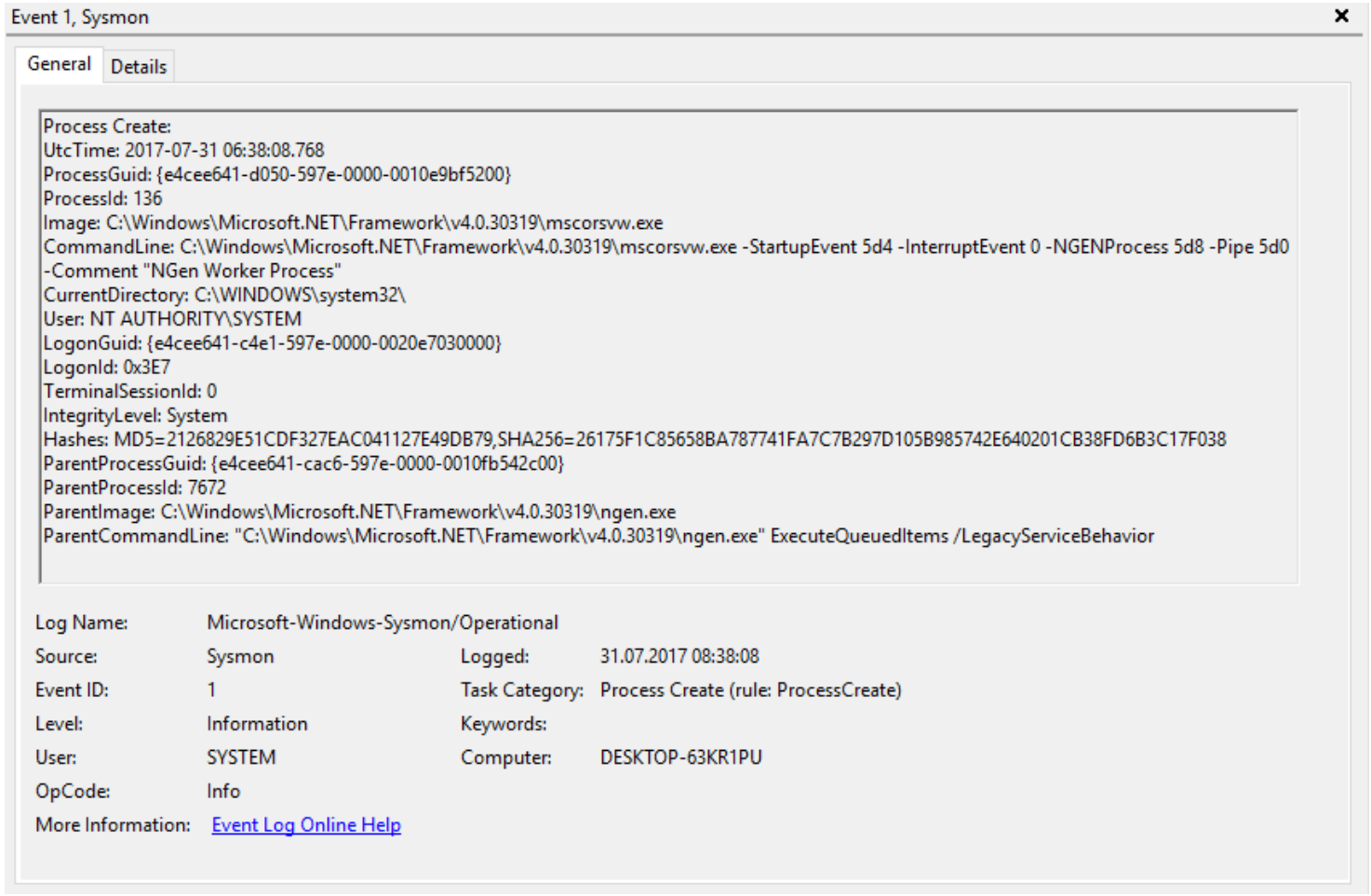
- Indexing all network traffic
- Displaying TCP & UDP sessions
- Stateful packet inspection (SPI)
- API for third-party apps

Exercise #1.1 – Network flow analysis

- Open **moloch.enisa.ex** in web browser
  - Log to Moloch with credentials: *admin/MOLOCH*
  - Navigate to *Sessions* tab
    - In Search textbox type: **`dns.query.type == A`** (All DNS requests)
    - Adding next condition (with AND) like **`host.dns.cnt == 1`** (number of server's IP from request) could help with manual packet filtering
  - Any unusual or evil domains?
    - A lot of network traffic!

Exercise #2
Windows (Sysmon) log analysis

Sysmon


Event 1, Sysmon
General
Details
Process Create:
UtcTime: 2017-07-31 06:38:08.768
ProcessGuid: {e4cee641-d050-597e-0000-0010e9bf5200}
ProcessId: 136
Image: C:\Windows\Microsoft.NET\Framework\v4.0.30319\mscorsvw.exe
CommandLine: C:\Windows\Microsoft.NET\Framework\v4.0.30319\mscorsvw.exe -StartupEvent 5d4 -InterruptEvent 0 -NGENProcess 5d8 -Pipe 5d0
-Comment "NGen Worker Process"
CurrentDirectory: C:\WINDOWS\system32\
User: NT AUTHORITY\SYSTEM
LogonGuid: {e4cee641-c4e1-597e-0000-0020e7030000}
LogonId: 0x3E7
TerminalSessionId: 0
IntegrityLevel: System
Hashes: MD5=2126829E51CDF327EAC041127E49DB79,SHA256=26175F1C85658BA787741FA7C7B297D105B985742E640201CB38FD6B3C17F038
ParentProcessGuid: {e4cee641-cac6-597e-0000-0010fb542c00}
ParentProcessId: 7672
ParentImage: C:\Windows\Microsoft.NET\Framework\v4.0.30319\ngen.exe
ParentCommandLine: "C:\Windows\Microsoft.NET\Framework\v4.0.30319\ngen.exe" ExecuteQueuedItems /LegacyServiceBehavior
Log Name: Microsoft-Windows-Sysmon/Operational
Source: Sysmon
Logged: 31.07.2017 08:38:08
Event ID: 1
Task Category: Process Create (rule: ProcessCreate)
Level: Information
Keywords:
User: SYSTEM
Computer: DESKTOP-63KR1PU
OpCode: Info
More Information: Event Log Online Help

Sysmon

“It provides detailed information about process creations, network connections, and changes to file creation time.” – from official site

- Works as a Windows system service and device driver
- Once installed on a system, remains resident across system reboots to monitor and log system activity to the Windows event log
- All logged events are stored in **Applications and Services Logs/Microsoft/Windows/Sysmon/Operational**

Sysmon

To monitor network connections and hashes (MD5, SHA1) for created processes:

- `sysmon -accepteula -i -n -h md5,sha1`

To monitor lsass.exe process with all hashes:

- `sysmon -accepteula -i -l lsass.exe -h *`

Sysmon

Our provided log contains events from infected workstation about:

- Process creation & termination (with file / image hashes)
- Network connections

In exercise context the most interesting things for us are process creation events - **potential malicious code execution**

Exercise #2.0 – Sysmon log analysis

Quick tips before start:

- Check **MISP** for possible IoCs and threat actor information
- Open **HIVE** in new tab
- Take notes for them before start of your analysis ☺

Exercise #2.1 – Sysmon log analysis

- When we have some IoCs – time to check logs for them
  - Process (image) names
  - Hashes
  - Domains
- Quick and dirty check for confirmed malicious hashes
  - On Linux machine in Terminal: `cat sysmon_logs.txt | grep <hash_from_MISP>`
  - If exists – analyzed system is infected

ENISA Trainings
Windows malware analysis
Investigation - Infection #2

Section objectives

1. Static malware analysis
2. Dynamic malware analysis
3. Malware decompilation

Before we start…

Is good to know:

- Add all necessary information to Hive
- Malware very often try to hide with names of system's executables or random names in %APPDATA% folders'
- At the beginning of analysis is worth to check autostart entries with, for example, Autoruns or Task Manager and run:
  1. Process Monitor
  2. Wireshark
  3. Process Explorer
  4. (If malicious binary found) pestudio

All will be introduced later

Context

Little reminder:

1. Spear-phishing with malicious invoice
    1. Malware #1 – mostly log analysis
2. Infected Wordpress instance
    1. Malware #2 (**We are here!**)

Must-Have Tools

- Windows Sysinternals suite
- Pestudio
- Wireshark
- Loki

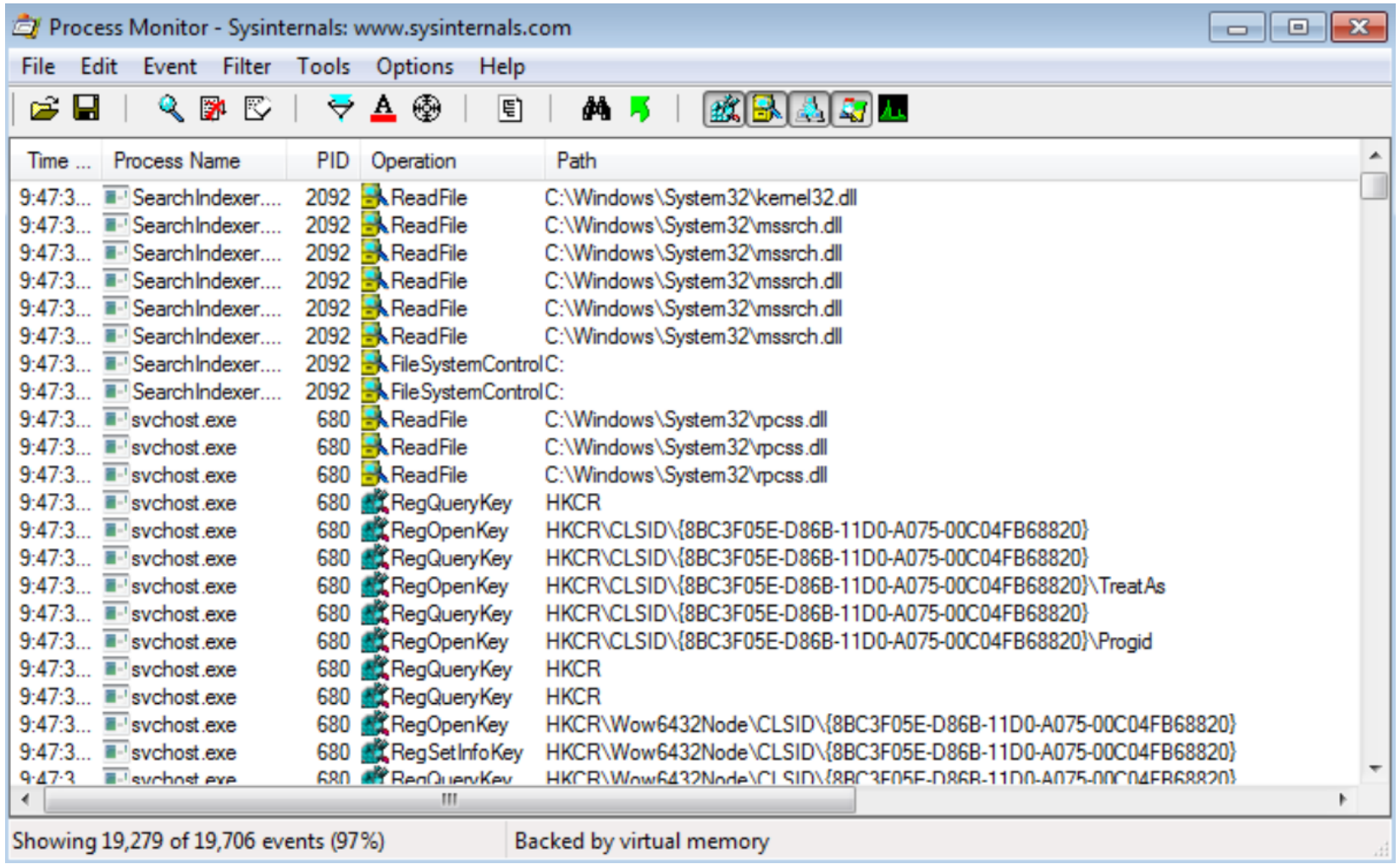
Windows Sysinternals

“Whether you’re an IT Pro or a developer, you’ll find Sysinternals utilities to help you manage, troubleshoot and diagnose your Windows systems and applications” – from official site.

Most useful tools for initial Windows’ malware research:

- Process Monitor
- Process Explorer
- Autoruns

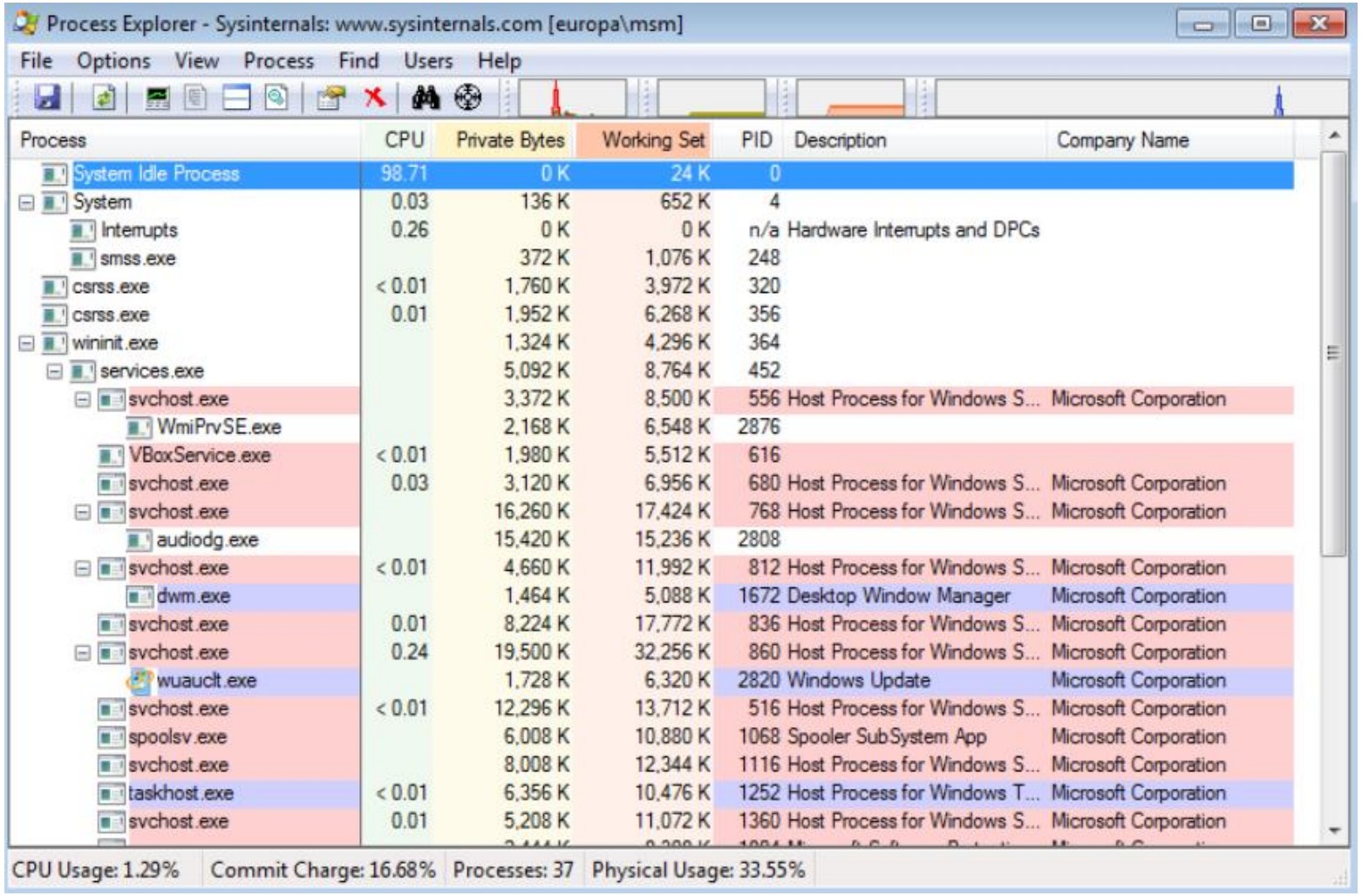
Process Monitor



Process Monitor

- **Advanced monitoring tool for Windows that shows real-time file system, registry, network and process / thread activity**
- Output: lots of Windows internal mechanism information
  - Loaded libraries
  - Process and thread stacks
  - Operation status
- Unfortunately is difficult to use for beginners

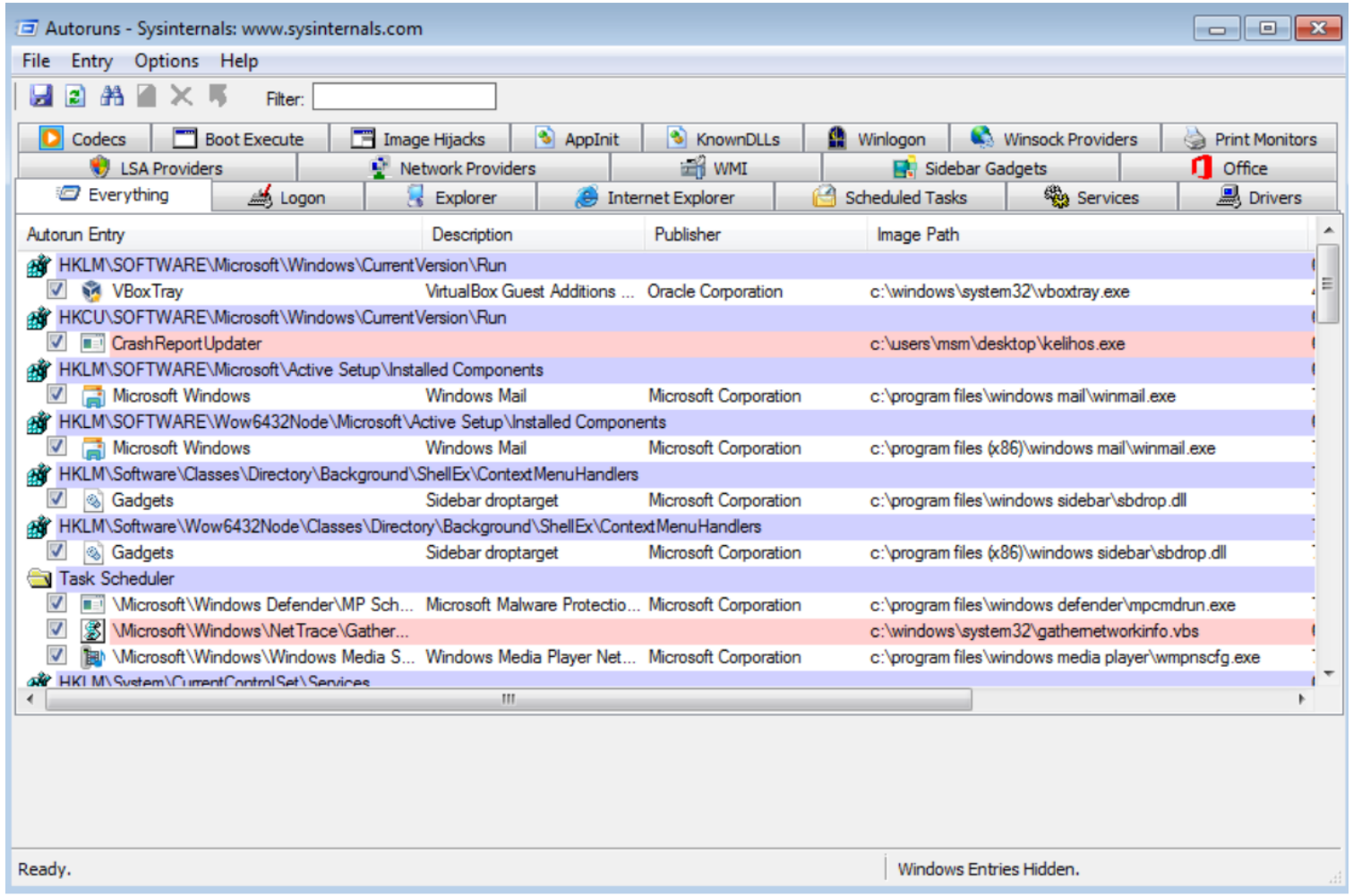
Process Explorer



Process Explorer

- **“Task manager on steroids” ☺**
- Output:
  - System resources statistics (CPU, memory, etc…)
  - Displays process's handles which includes:
    - Named mutants
    - Events
    - Sockets
    - Files
    - Registry keys
  - Creates dump of process memory

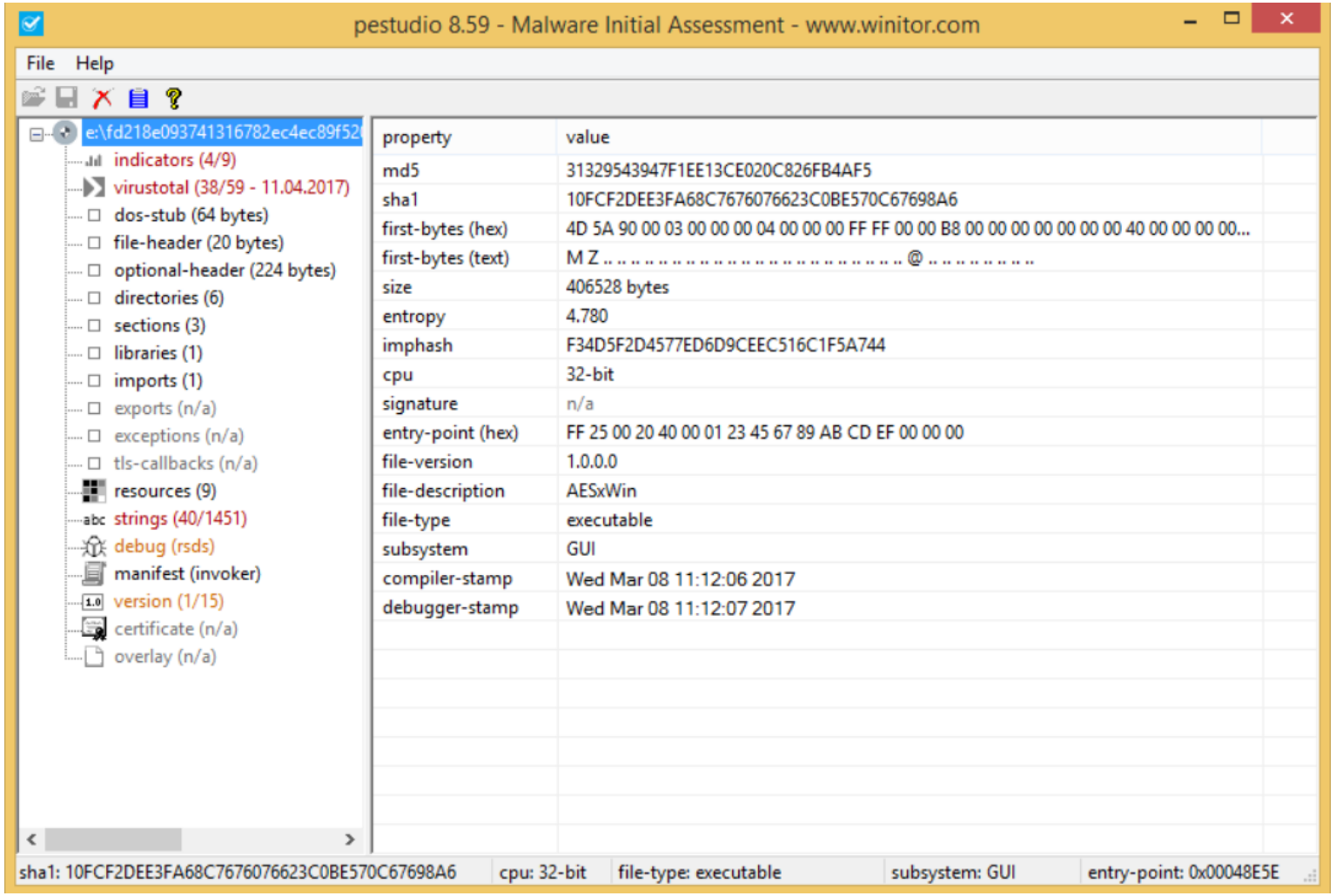
Autoruns


Autoruns - Sysinternals: www.sysinternals.com
File Entry Options Help
Filter:
Codecs
Boot Execute
Image Hijacks
AppInit
KnownDLLs
Winlogon
Winsock Providers
Print Monitors
LSA Providers
Network Providers
WMI
Sidebar Gadgets
Office
Everything
Logon
Explorer
Internet Explorer
Scheduled Tasks
Services
Drivers
Autorun Entry
Description
Publisher
Image Path
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
VBoxTray
VirtualBox Guest Additions ...
Oracle Corporation
c:\windows\system32\vboxtray.exe
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
CrashReportUpdater
c:\users\msm\desktop\kelihos.exe
HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components
Microsoft Windows
Windows Mail
Microsoft Corporation
c:\program files\windows mail\winmail.exe
HKLM\SOFTWARE\Wow6432Node\Microsoft\Active Setup\Installed Components
Microsoft Windows
Windows Mail
Microsoft Corporation
c:\program files (x86)\windows mail\winmail.exe
HKLM\Software\Classes\Directory\Background\ShellEx\ContextMenuHandlers
Gadgets
Sidebar droptarget
Microsoft Corporation
c:\program files\windows sidebar\sbdrop.dll
HKLM\Software\Wow6432Node\Classes\Directory\Background\ShellEx\ContextMenuHandlers
Gadgets
Sidebar droptarget
Microsoft Corporation
c:\program files (x86)\windows sidebar\sbdrop.dll
Task Scheduler
\Microsoft\Windows Defender\MP Sch...
Microsoft Malware Protectio...
Microsoft Corporation
c:\program files\windows defender\mpcmdrun.exe
\Microsoft\Windows\NetTrace\Gather...
c:\windows\system32\gathernetworkinfo.vbs
\Microsoft\Windows\Windows Media S...
Windows Media Player Net...
Microsoft Corporation
c:\program files\windows media player\wmpnscfg.exe
Ready.
Windows Entries Hidden.

Autoruns

- **Shows you what programs are configured to run during system bootup or login**
- Output:
  - Various Windows objects started at system boot:
    - Codecs
    - DLLs
    - Services
    - WMI objects
    - Scheduled tasks
    - Office sub-components

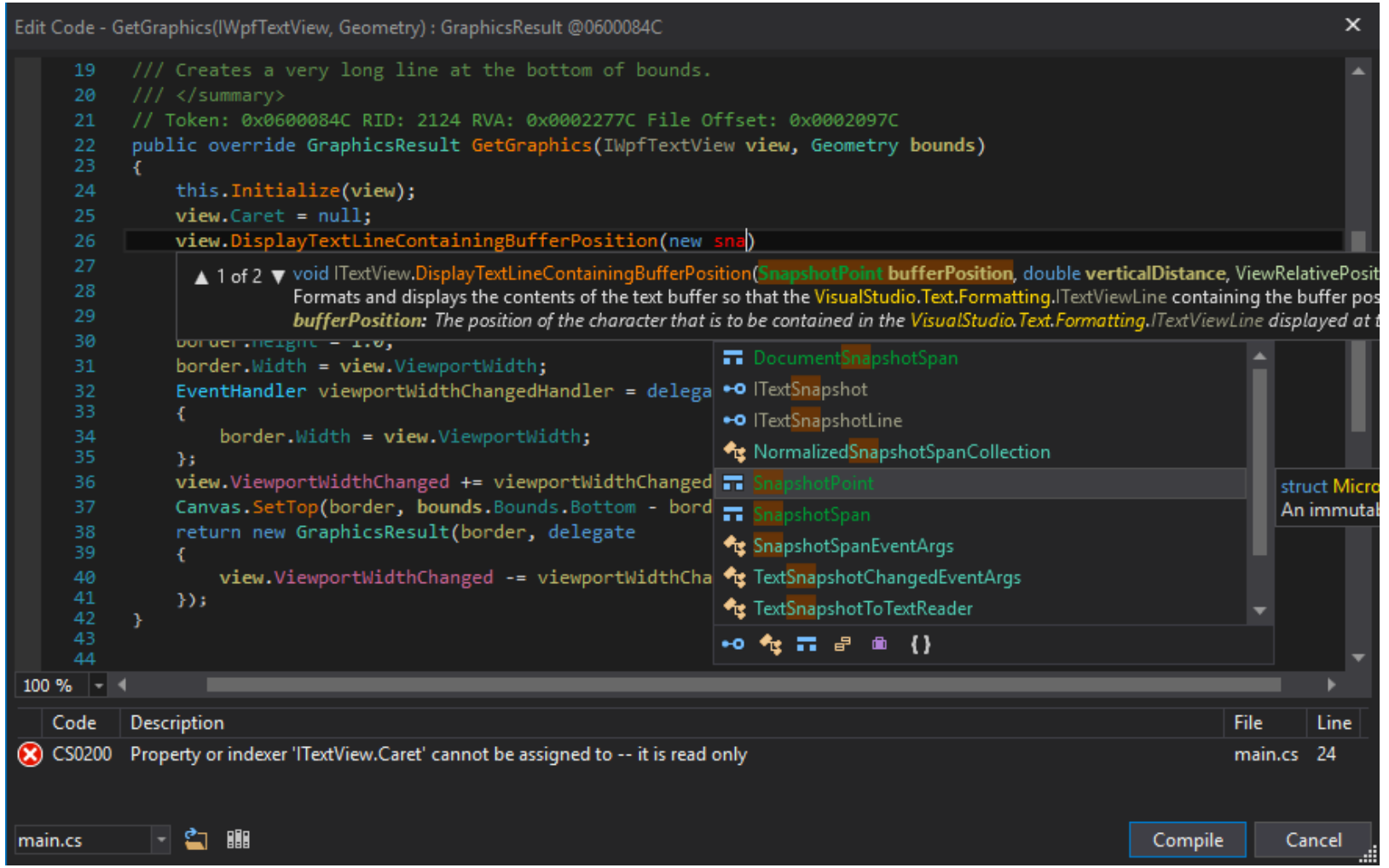
pestudio


pestudio 8.59 - Malware Initial Assessment - www.winitor.com
File Help
e:\fd218e093741316782ec4ec89f52
indicators (4/9)
virustotal (38/59 - 11.04.2017)
dos-stub (64 bytes)
file-header (20 bytes)
optional-header (224 bytes)
directories (6)
sections (3)
libraries (1)
imports (1)
exports (n/a)
exceptions (n/a)
tls-callbacks (n/a)
resources (9)
strings (40/1451)
debug (rsds)
manifest (invoker)
version (1/15)
certificate (n/a)
overlay (n/a)
property value
md5 31329543947F1EE13CE020C826FB4AF5
sha1 10FCF2DEE3FA68C7676076623C0BE570C67698A6
first-bytes (hex) 4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00 00...
first-bytes (text) M Z .. .. @ .. ..
size 406528 bytes
entropy 4.780
imphash F34D5F2D4577ED6D9CEEC516C1F5A744
cpu 32-bit
signature n/a
entry-point (hex) FF 25 00 20 40 00 01 23 45 67 89 AB CD EF 00 00 00
file-version 1.0.0.0
file-description AESxWin
file-type executable
subsystem GUI
compiler-stamp Wed Mar 08 11:12:06 2017
debugger-stamp Wed Mar 08 11:12:07 2017
sha1: 10FCF2DEE3FA68C7676076623C0BE570C67698A6
cpu: 32-bit
file-type: executable
subsystem: GUI
entry-point: 0x00048E5E

pestudio

- **Initial malware assessment**
- Input: Windows application binary file
- Output:
  - VirusTotal report
  - PE format informations:
    - Imports / Exports
    - Resources
    - Strings
    - Manifest
    - Digital certificates

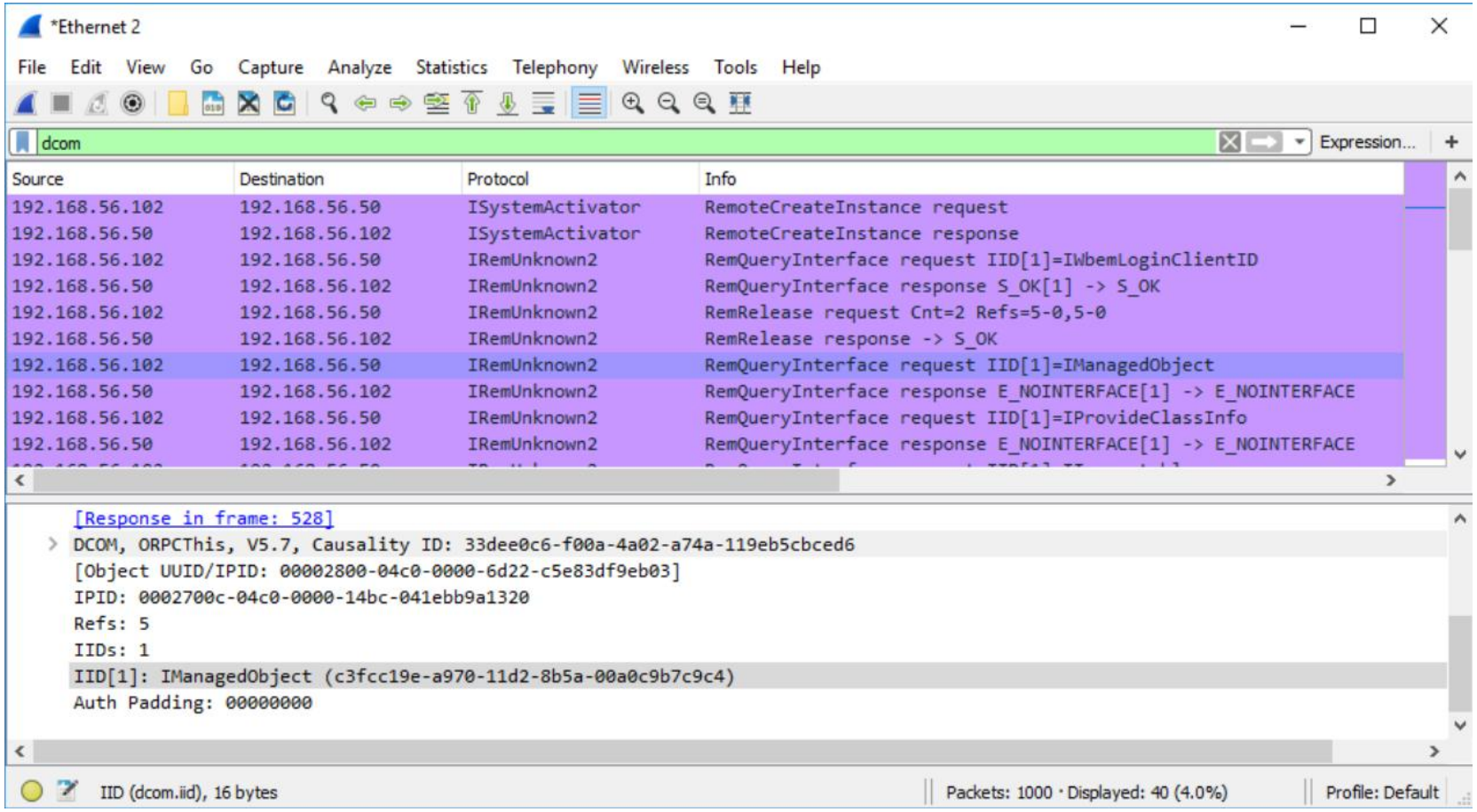
dnSpy & ILspy


Edit Code - GetGraphics(IWpfTextView, Geometry) : GraphicsResult @0600084C
/// Creates a very long line at the bottom of bounds.
/// </summary>
// Token: 0x0600084C RID: 2124 RVA: 0x0002277C File Offset: 0x0002097C
public override GraphicsResult GetGraphics(IWpfTextView view, Geometry bounds)
{
this.Initialize(view);
view.Caret = null;
view.DisplayTextLineContainingBufferPosition(new sna)
▲ 1 of 2 ▼ void ITextView.DisplayTextLineContainingBufferPosition(SnapshotPoint bufferPosition, double verticalDistance, ViewRelativePosit
Formats and displays the contents of the text buffer so that the VisualStudio.Text.Formatting.ITextViewLine containing the buffer pos
bufferPosition: The position of the character that is to be contained in the VisualStudio.Text.Formatting.ITextViewLine displayed at t
border.Width = view.ViewportWidth;
EventHandler viewportWidthChangedHandler = delega
{
border.Width = view.ViewportWidth;
};
view.ViewportWidthChanged += viewportWidthChanged
Canvas.SetTop(border, bounds.Bounds.Bottom - bord
return new GraphicsResult(border, delegate
{
view.ViewportWidthChanged -= viewportWidthCha
});
}
DocumentSnapshotSpan
ITextSnapshot
ITextSnapshotLine
NormalizedSnapshotSpanCollection
SnapshotPoint
SnapshotSpan
SnapshotSpanEventArgs
TextSnapshotChangedEventArgs
TextSnapshotToTextReader
struct Micro
An immutab
100 %
Code
Description
File
Line
CS0200
Property or indexer 'ITextView.Caret' cannot be assigned to -- it is read only
main.cs
24
main.cs
Compile
Cancel

dnSpy & ILspy

- **.NET assembly browsers and decompilers**
- Input: C# application binary file
- Possible output:
  - Editing all metadata of types (classes), methods, properties, events, fields
  - Adding, removing, renaming any type (class), method, property, event, field
  - Editing, adding, removing .NET resources and saving them to disk
  - Debugging any .NET assembly, no source code required

Wireshark


*Ethernet 2
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
dcom
Expression...
Source Destination Protocol Info
192.168.56.102 192.168.56.50 ISystemActivator RemoteCreateInstance request
192.168.56.50 192.168.56.102 ISystemActivator RemoteCreateInstance response
192.168.56.102 192.168.56.50 IRemUnknown2 RemQueryInterface request IID[1]=IWbemLoginClientID
192.168.56.50 192.168.56.102 IRemUnknown2 RemQueryInterface response S_OK[1] -> S_OK
192.168.56.102 192.168.56.50 IRemUnknown2 RemRelease request Cnt=2 Refs=5-0,5-0
192.168.56.50 192.168.56.102 IRemUnknown2 RemRelease response -> S_OK
192.168.56.102 192.168.56.50 IRemUnknown2 RemQueryInterface request IID[1]=IManagedObject
192.168.56.50 192.168.56.102 IRemUnknown2 RemQueryInterface response E_NOINTERFACE[1] -> E_NOINTERFACE
192.168.56.102 192.168.56.50 IRemUnknown2 RemQueryInterface request IID[1]=IProvideClassInfo
192.168.56.50 192.168.56.102 IRemUnknown2 RemQueryInterface response E_NOINTERFACE[1] -> E_NOINTERFACE
[Response in frame: 528]
DCOM, ORPCThis, V5.7, Causality ID: 33dee0c6-f00a-4a02-a74a-119eb5cbced6
[Object UUID/IPID: 00002800-04c0-0000-6d22-c5e83df9eb03]
IPID: 0002700c-04c0-0000-14bc-041ebb9a1320
Refs: 5
IIDs: 1
IID[1]: IManagedObject (c3fcc19e-a970-11d2-8b5a-00a0c9b7c9c4)
Auth Padding: 00000000
IID (dcom.iid), 16 bytes
Packets: 1000 · Displayed: 40 (4.0%)
Profile: Default

Wireshark

- **Wireshark is the world's foremost and widely-used network protocol analyzer**
- Output:
  - Dumping traffic from a live network connection or read from a file of already-captured packets
  - Data display can be refined using a display filter (powerful!).
  - Plug-ins can be created for dissecting new protocols.

Exercise #1
Check for Windows infection

Exercise #1.0 – Monitoring Windows behavior

Quick tips (once again):

- Check **MISP** for possible IoCs and threat actor information
- Open **HIVE** in new tab
- Take notes for them before start of your analysis ☺

Exercise #1.1 – Monitoring Windows behavior

- Based on **MISP** infromation about threat actors try to find malicious process on infected machine:
  - Run **ProcessExplorer**
    - Click on *View*, choose *Select Columns…*
      - Tick checkbox: *User Name*
    - Check all processes for *User Name*
      - Is there any Windows binaries running on logged user rights?
        - Hint: Column *Company Name*
    - Open one or two processes properties and check if there is additional tabs
      - *.NET Assemblies*
      - *.NET Performance*

Exercise #1.2 – Monitoring Windows behavior

- Run **ProcessMonitor**
  - Wait about 1-2 minutes
  - Stop event capturing with keys: CTRL+X
- Filter Events
  - Add filter for found process na
  - Registry activity
    - Add another filter condition: *Result* is *Success*
  - File activity
    - Add new filter: *Operation* is *WriteFile*
  - Network activity
    - Add new filter: *Operation* contains *TCP*

Exercise #1.3 – Monitoring Windows behavior

- Open **moloch.enisa.ex** in web browser
  - Log to Moloch with credentials: *admin/MOLOCH*
  - Navigate to *Sessions* tab
    - In Search textbox type:
      - **`dns.query.type == A`** (All DNS requests)
      - **`http.method == POST`** (All HTTP POST requests)
      - **http.bodymagic == application/json** (All HTTP requests with JSONs)

Exercise #2
Analyze malicious binary

Exercise #2.1 – Analyze malicious binary

- Run **pestudio** and open found binary
- Click on tabs:
  - Indicators
  - VirusTotal
  - Strings
  - Version
- Is there any information about programming language in binary?
  - C / C++
  - C# / Java
  - Python

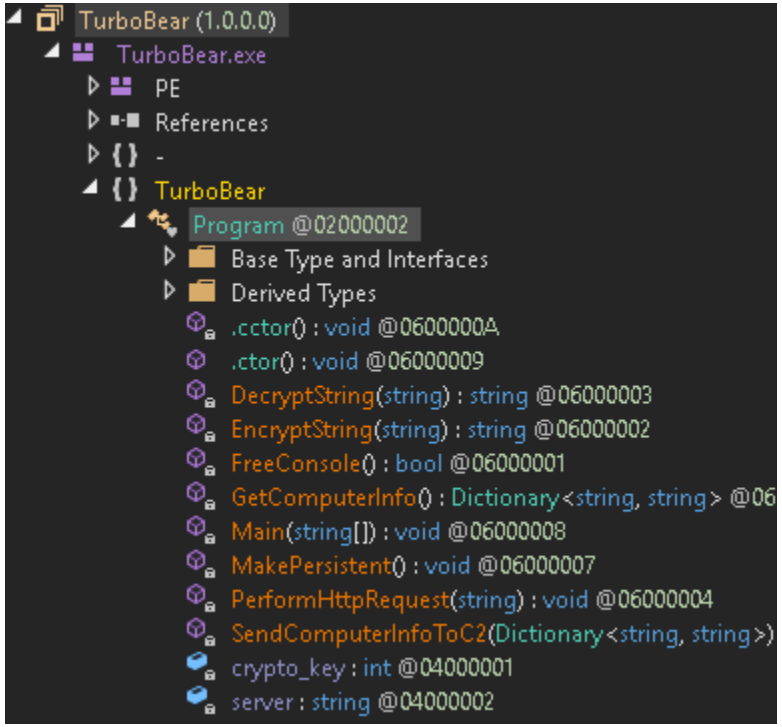
Exercise #2.2 – Analyze malicious binary

- What we know after quick look in previous tools?
  - Command-and-control servers?
  - Malware basic features?
  - How the network traffic is generated?
  - Persistence options?
  - Crypto keys & functions?
- Is this enough to create indicators of compromise?

(Optional) Exercise #3
Malicious binary decompilation

Exercise #3.1 – Malicious binary decompilation

- Run **dnSpy (C# decompiler)** and open found binary
  - Expand last position in menu (probably binary original name)



Exercise #3.2 – Malicious binary decompilation

- C# malware analysis tips & tricks
  - Windows autostart registry entry:
    HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
  - HTTP Requests
    - Request object creation:
      HttpWebRequest httpWebRequest = (HttpWebRequest)WebRequest.Create(*address*);
    - Selecting request method (GET / POST): **httpWebRequest.Method = "POST";**
    - Setting *Content type*: **httpWebRequest.ContentType = "application/json";**
    - Dictionary object serialization:
      string serialized_dict = new JavaScriptSerializer().Serialize(dict);
- Quick tip: Add this information to **HIVE** for further analysis

Exercise #3.3 – Malicious binary decompilation

- Click on *Search Assemblies* button (magnifier icon or CTRL+ Shift + K)
  - In *Options*, disable: *Search in GAC assemblies*
  - In *Search For:*, choose from list *Number / Strings* and *Selected Type*
    - Type in search bar:
      - Http
      - GET
      - POST
      - Windows
  - Change *Search For:* to *All of the Above*
    - Type in search bar:
      - crypto
      - server