MALWARE ANALYSIS

Basic Dynamic Analysis

Dr. Zeyad Safaa Younus Saffawi

Dynamic Analysis

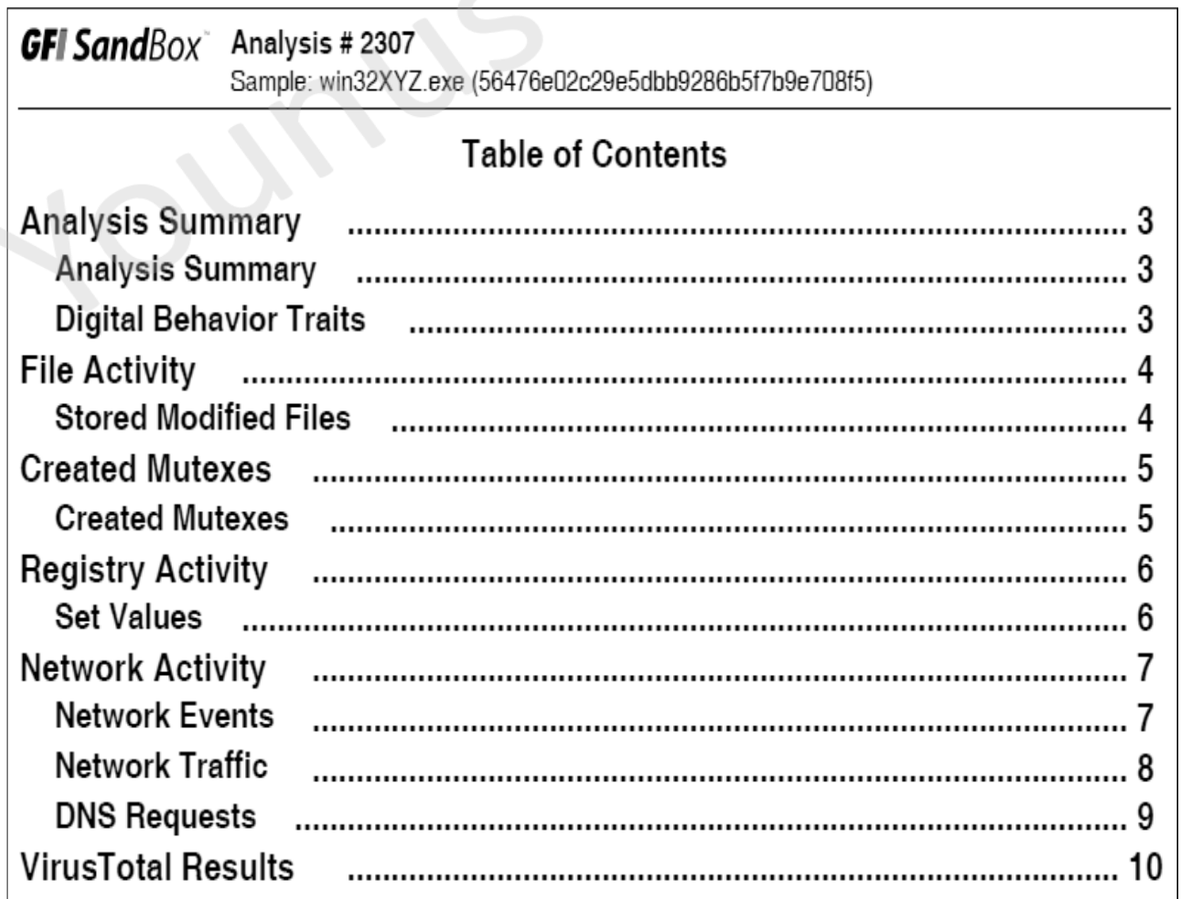
- **Dynamic analysis** involves examining malware during its execution to observe its actual behavior. This approach is highly effective because it reveals precisely how the malware interacts with the system, modifies files or the registry, communicates over the network, and performs other malicious actions.
- Dynamic analysis is generally conducted as the **second step in malware analysis**, following static analysis. It is particularly necessary when static analysis cannot provide sufficient information due to challenges such as:
  - **Obfuscation:** Techniques used by malware authors to hide program logic.
  - **Packing:** Compression or encryption of the executable code that prevents direct inspection.
  - **Limitations of static techniques:** Inability to observe runtime behavior, system interactions, or network communications.
- By executing malware in a controlled environment, analysts can overcome these obstacles and gain a comprehensive understanding of the malware's functionality.

Sandbox

- A sandbox is a safe, isolated environment used to run untrusted programs (like malware) for basic dynamic analysis without harming real systems.
- Sandboxes often simulate network services so malware can run normally.
- Examples: Norman Sandbox, GFI Sandbox, Anubis, Joe Sandbox, Joe Sandbox, ThreatExpert, BitBlaze, Comodo Instant Malware Analysis.
  - paid versions are in-house and expensive but easy to use.
  - They produce a nice PDF report of results

Using a Sandbox (Example: GFI Sandbox):

- Automates malware execution and generates reports.
- Key report sections:
  - **Analysis Summary:** static analysis information and a high-level overview of the dynamic analysis results
  - **File Activity:** Created, opened, or deleted files
  - **Created Mutexes:** Synchronization objects used by malware
  - **Registry Activity:** Changes to the Windows registry
  - **Network Activity:** Connections, setting up a listening port, or performing a DNS request
  - **VirusTotal Results:** Scan results for malware detection



GFI SandBox Analysis # 2307
Sample: win32XYZ.exe (56476e02c29e5dbb9286b5f7b9e708f5)

Table of Contents

Running Malware

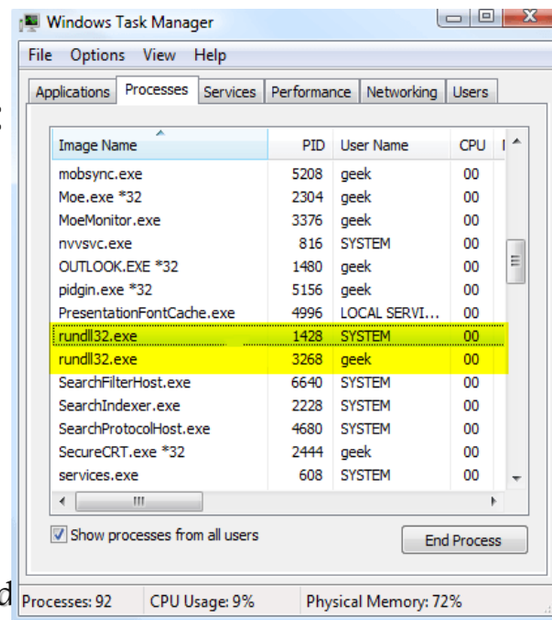
- Running malware is a critical step in **dynamic analysis**.
- Most **EXE** files can be run easily.
- **DLL** files require special methods because Windows does not execute them directly.
- Before running, it's important to **identify exported functions** (using tools like **PEview** or **PE Explorer,** Dependency Walker) to know how to launch the DLL correctly.
- **Running EXE Files**
  - Normally, you can double-click the file or use the command line:

    C:\> sample.exe
  - Before execution, always start your monitoring tools (ProcMon, Wireshark, Regshot…) and take a **snapshot** of the VM.
- **Running DLL Files using rundll32.exe**
  - rundll32.exe is built into Windows and allows executing an exported DLL function:

    C:\> rundll32.exe DLLname,ExportName arguments
  - If the function is exported by **ordinal number** instead of names, use #:

    C:\> rundll32.exe mydll.dll,#5
  - rip.dll has these export functions: **Install** and **Uninstall**

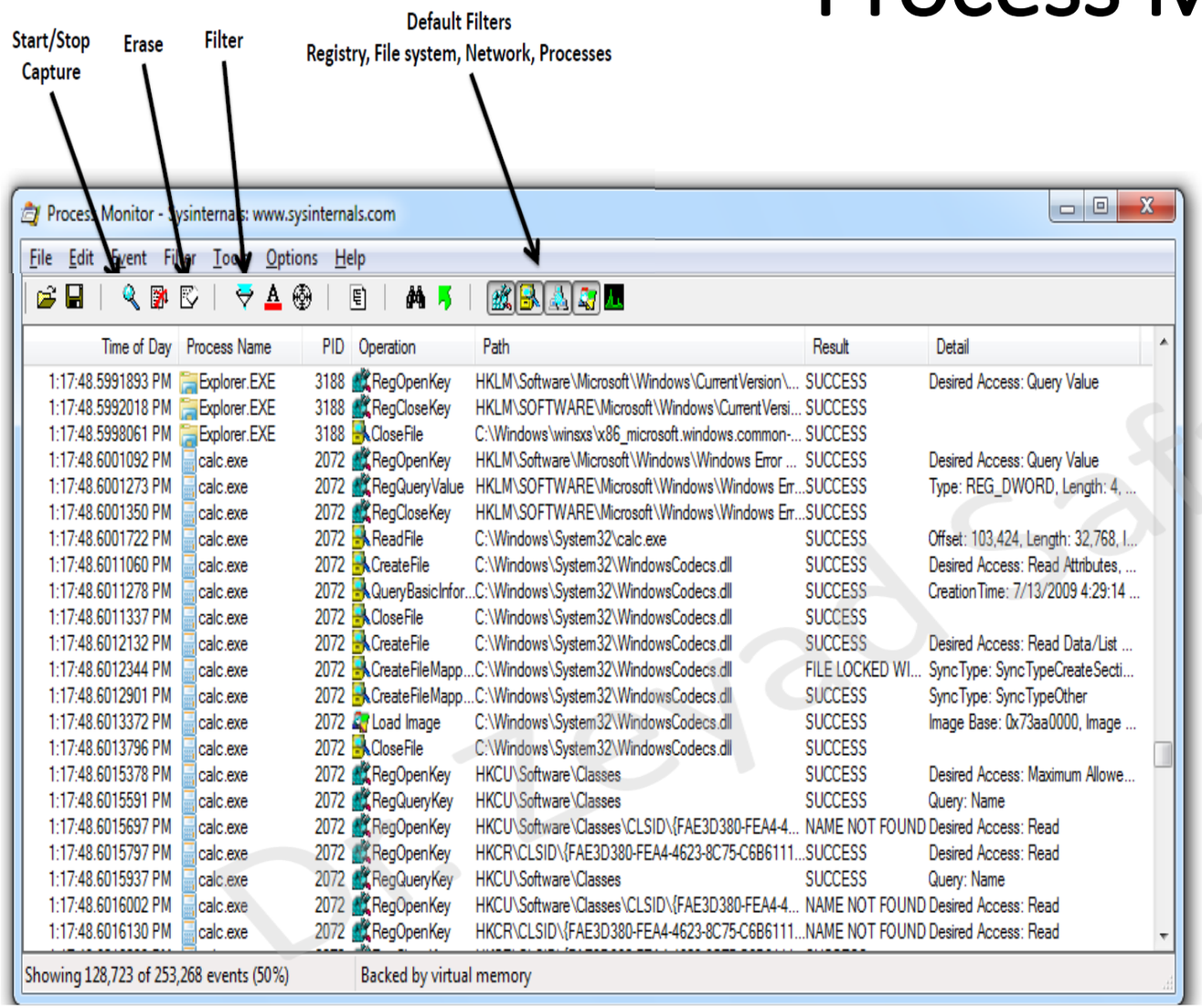
    C:\> rundll32.exe rip.dll, Install

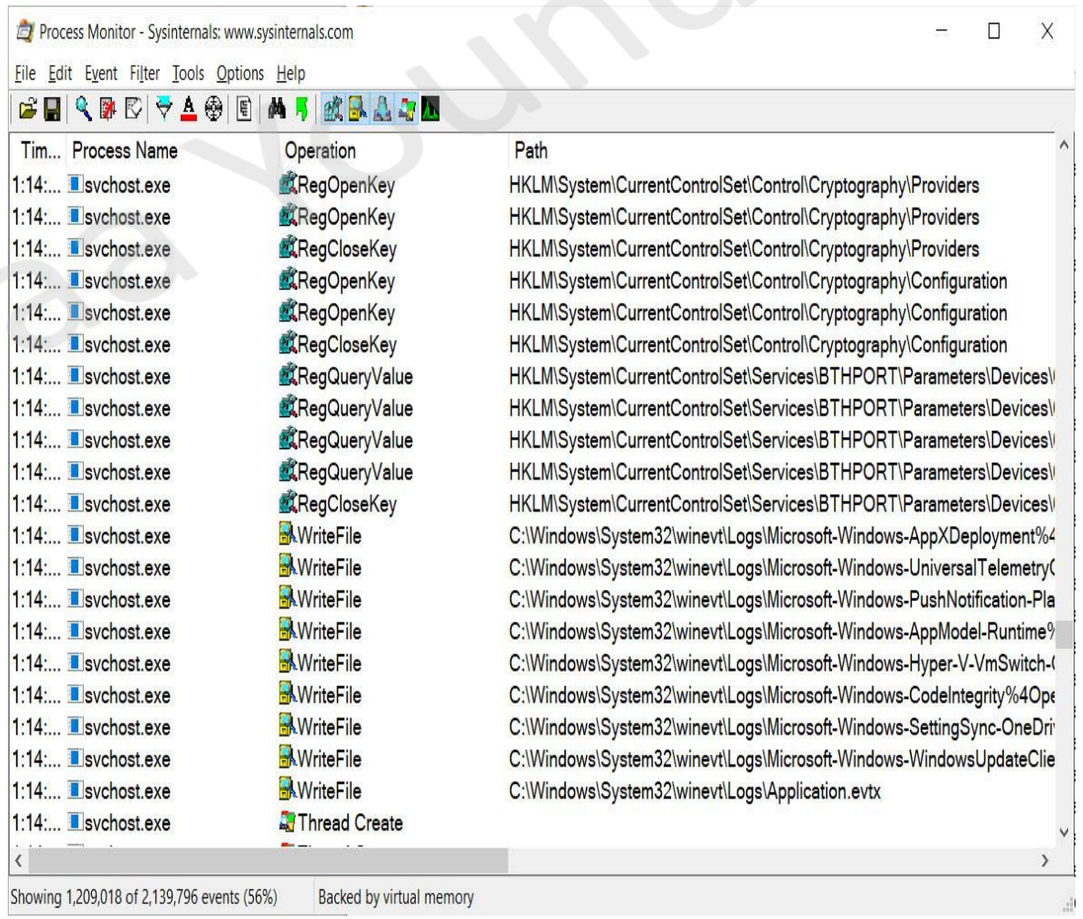


Process Monitor (Procmon)

- An advanced **Windows** monitoring tool that captures registry, file system, network, process/thread activity (combines FileMon + RegMon features)
- Starts capturing immediately when you enable capture; it buffers events in RAM until you stop.
- Because it can generate tens of thousands of events per minute, it can exhaust memory and crash a VM if left running too long.
- All recorded events are kept, but you can filter the display to make it easier to focus on relevant items

Process Monitor



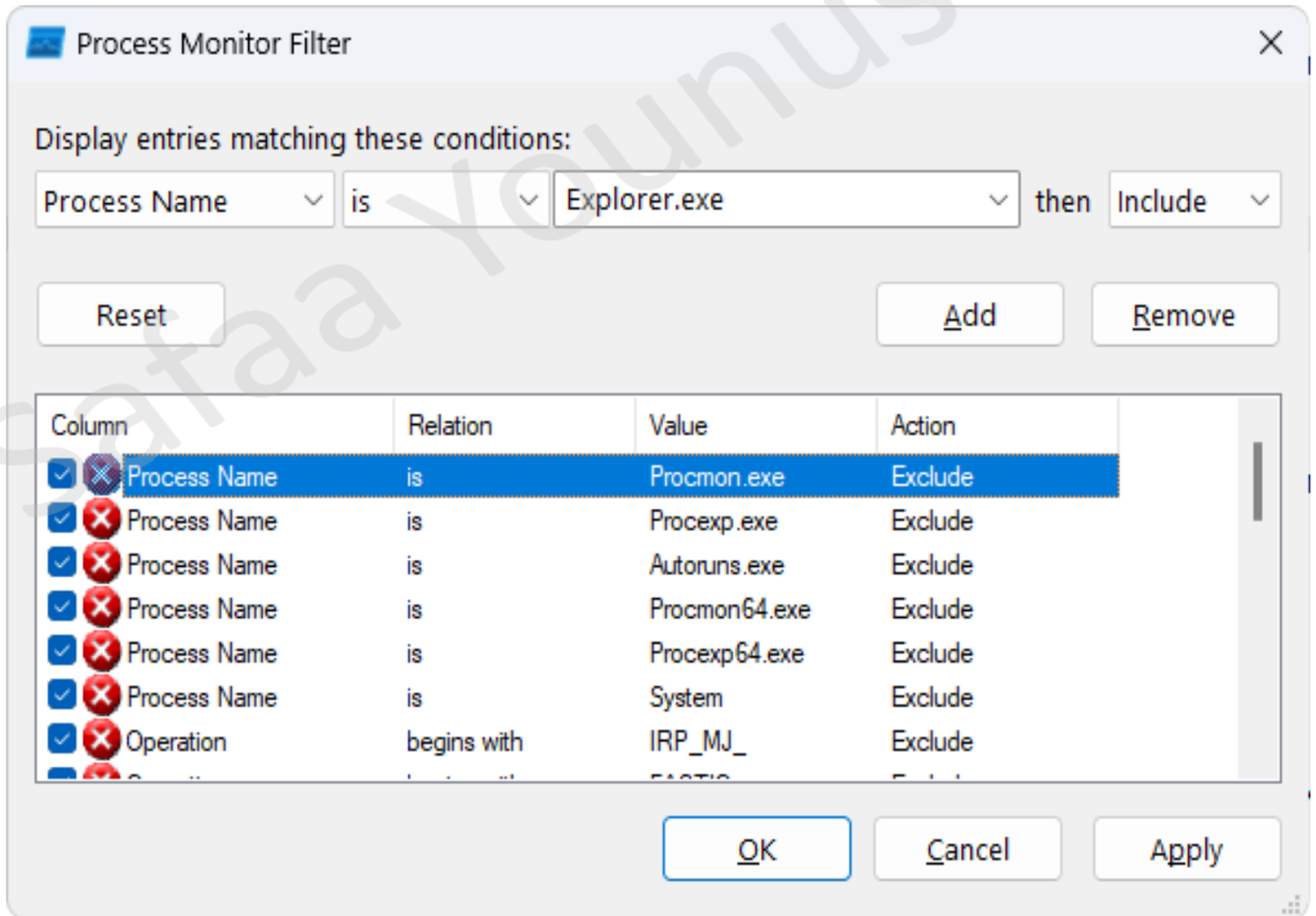


Filtering in Process Monitor tool

- When Procmon (Process Monitor) is running, it records **thousands of events**, making it hard to find relevant information.
- **Filtering** helps reduce noise and focus only on suspicious or important activities — especially during malware analysis.
  - Go to Filter → Filter.
  - Select a column (e.g., Process Name, Operation, or Detail).
  - Choose a condition (e.g., Is, Contains).
  - Decide to Include or Exclude the events.
  - Click Add, then Apply.

Filtering in Process Monitor tool

- Filtering with Exclude
  - hide normal activity before launching malware
  - Right-click each Process Name and click **Exclude**
- Filtering with Include
  - Most useful filters: Process Name, Operation, and Detail

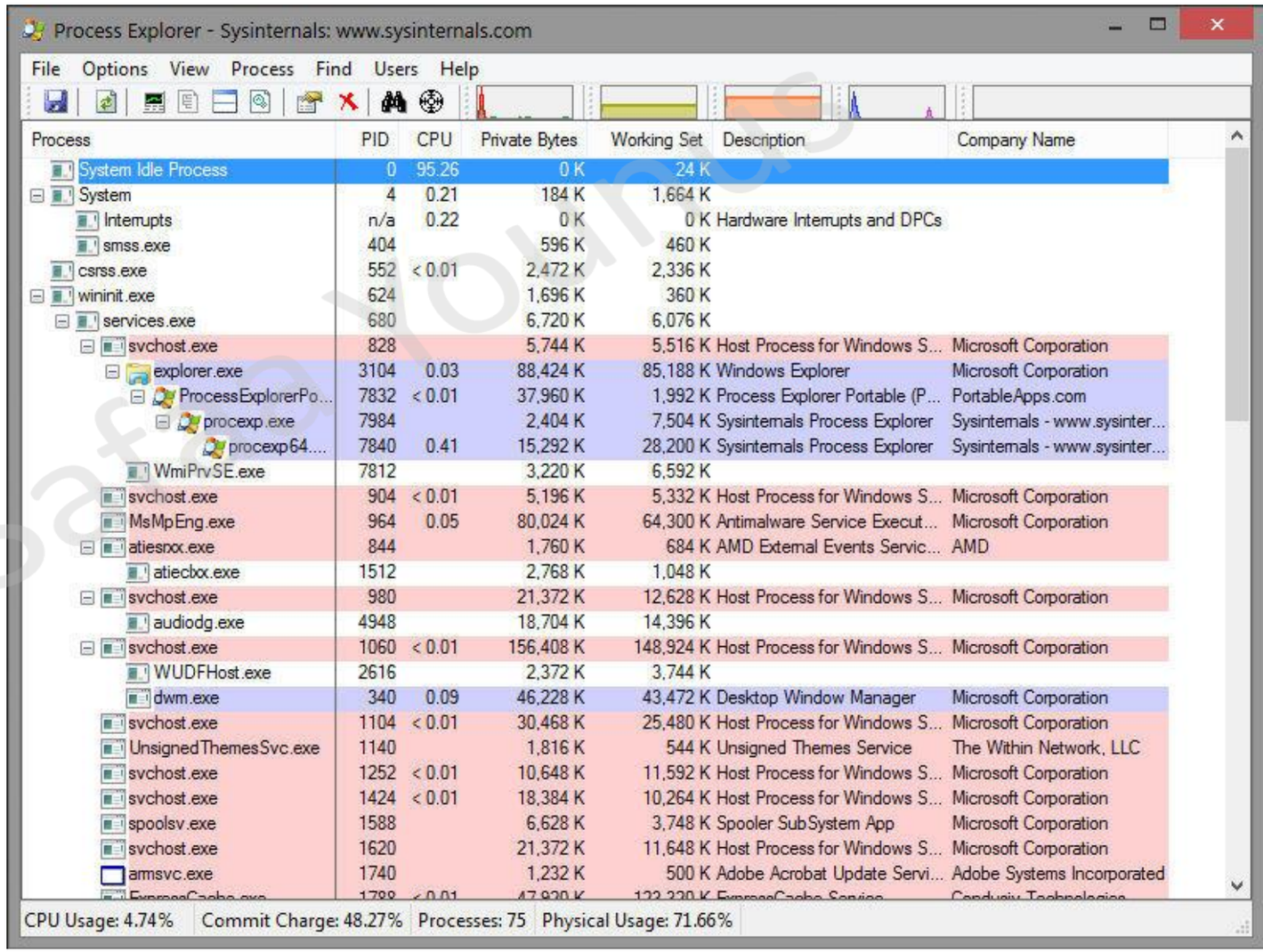


Viewing Processes with Process Explorer

- Process Explorer is a powerful task manager tool from Microsoft used in dynamic malware analysis. It shows processes in a tree structure
- It provides real-time visibility into:
  - Running processes
  - Loaded DLLs
  - System resource usage
  - Network connections
  - Process properties (threads, handles, etc.)
- Default columns:
  - Process
  - PID
  - CPU
  - Description
  - Company Name

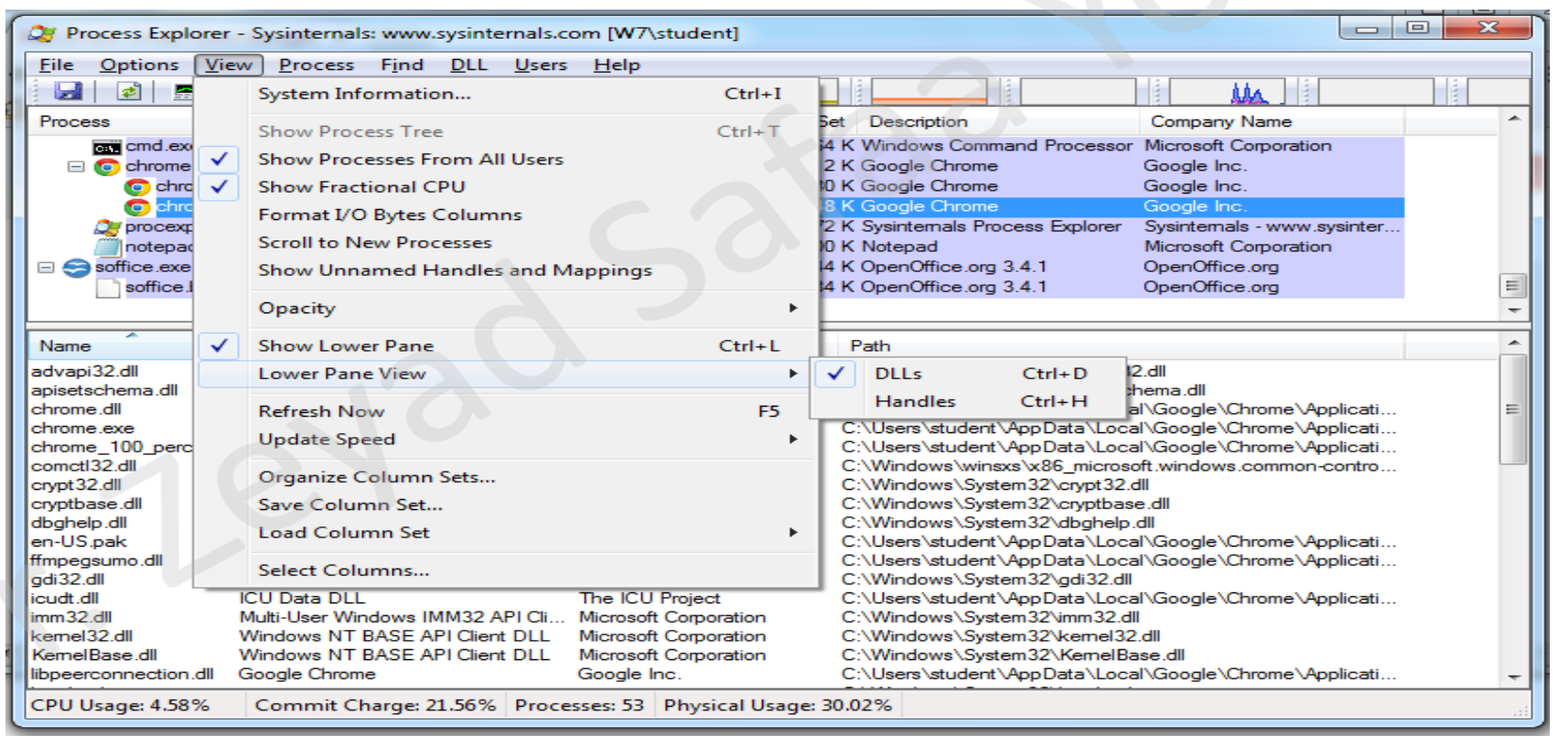
Color coding

- Pink = services
- Blue = processes
- Green = new processes
- Red = terminated processes.
- Helps track new or suspicious processes created y malware.



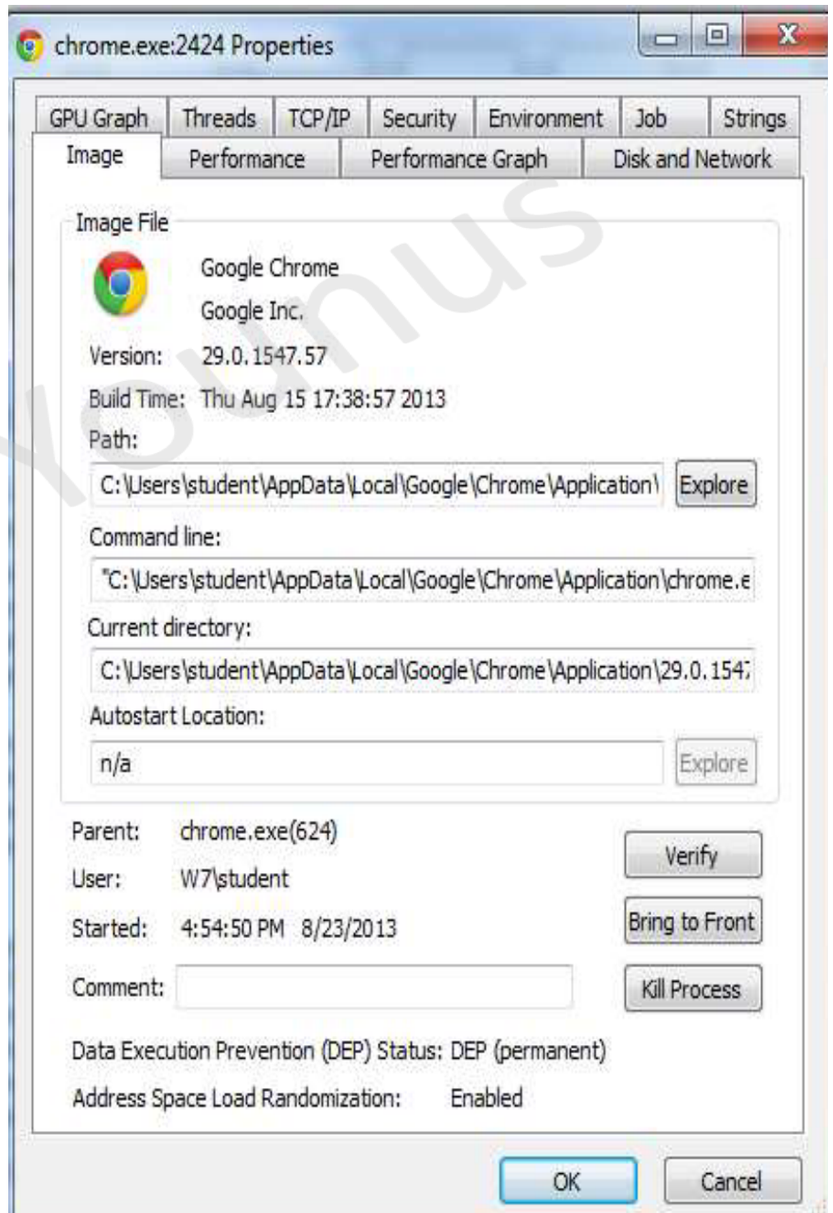
Inspecting Processes

- DLL View: Shows all DLLs loaded into a process.
- Handles View: Displays files, mutexes, events, and handles used.



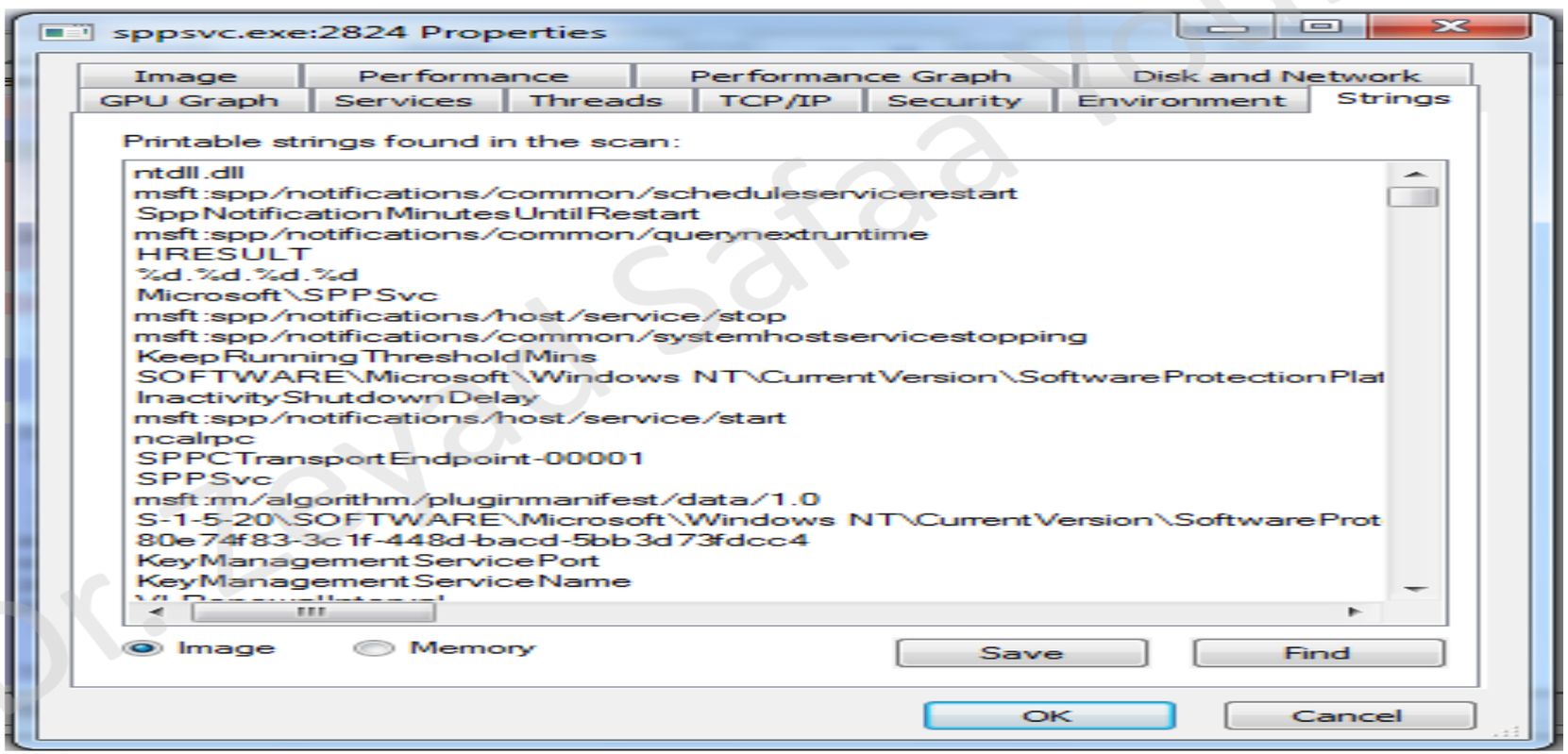
Properties

- Threads tab: Active threads
- TCP/IP tab: Active network connections or ports
- Image tab: File path on disk
  - Displays **DEP (Data Execution Prevention)** and **ASLR (Address Space Layout Randomization)** status.
- **Verify button** checks the **digital signature of the file on disk**.
  - Does **not verify the in-memory image**, so it **cannot detect process replacement**.



Properties

- Use the Strings tab to compare:
  - Disk image strings vs. memory strings.
  - A big difference may indicate process replacement (e.g., svchost.exe case).

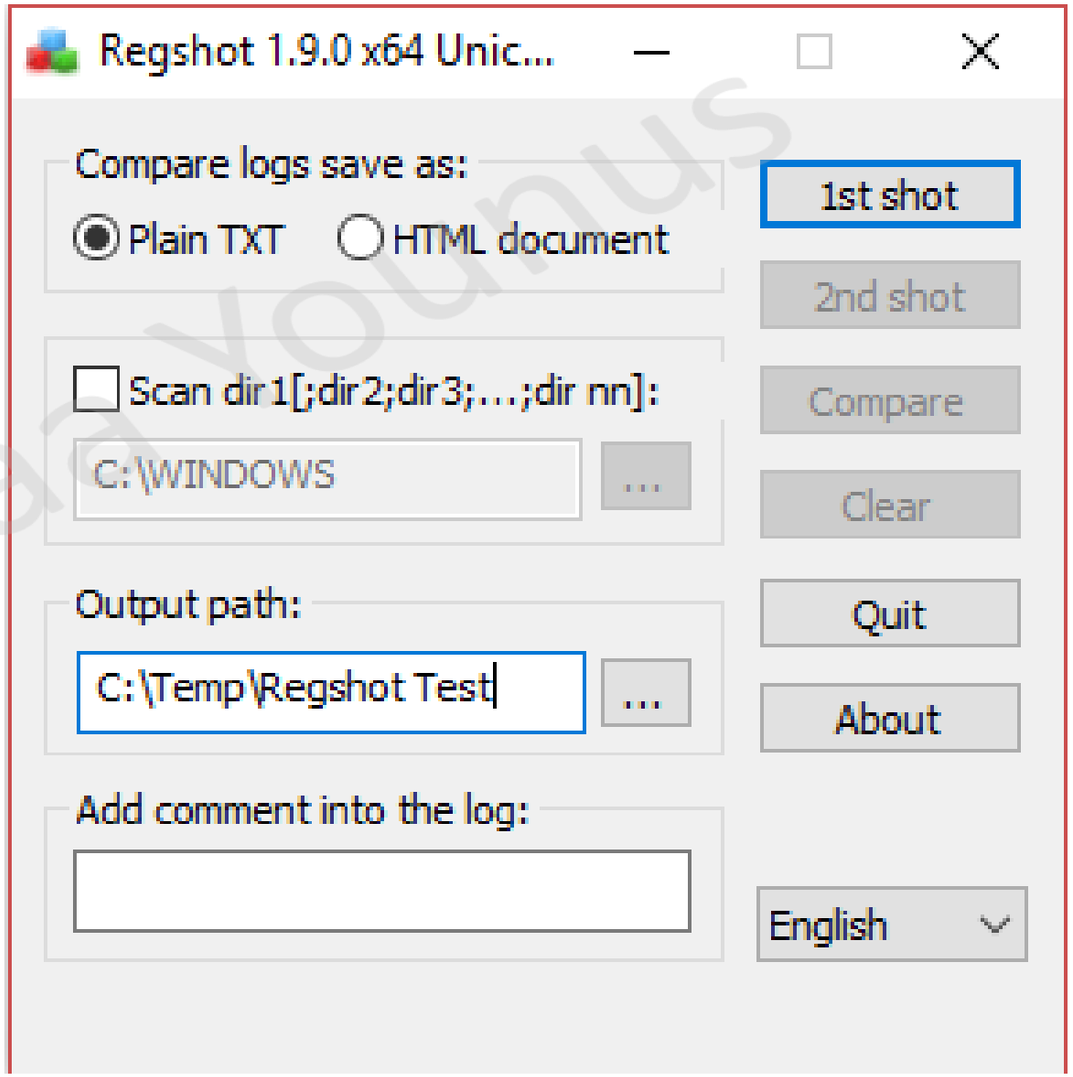


Detecting Malicious Documents

- Open suspicious **PDFs or Word** files while Process Explorer runs.
- Watch Process Explorer to see if it launches a process
- The Image tab of that process's Properties sheet will show where the malware is located

Regshot tool

- Regshot is an open-source tool for taking and comparing Windows registry snapshots.
- Identify changes made by malware to the registry, such as added or modified keys and values.
- **Regshot for malware analysis:**
- Take the **first snapshot** before running the malware by clicking **1st Shot**.
- Run the malware and let it complete its actions.
- Take the **second snapshot** by clicking **2nd Shot**.
- Click **Compare** to see all registry changes.

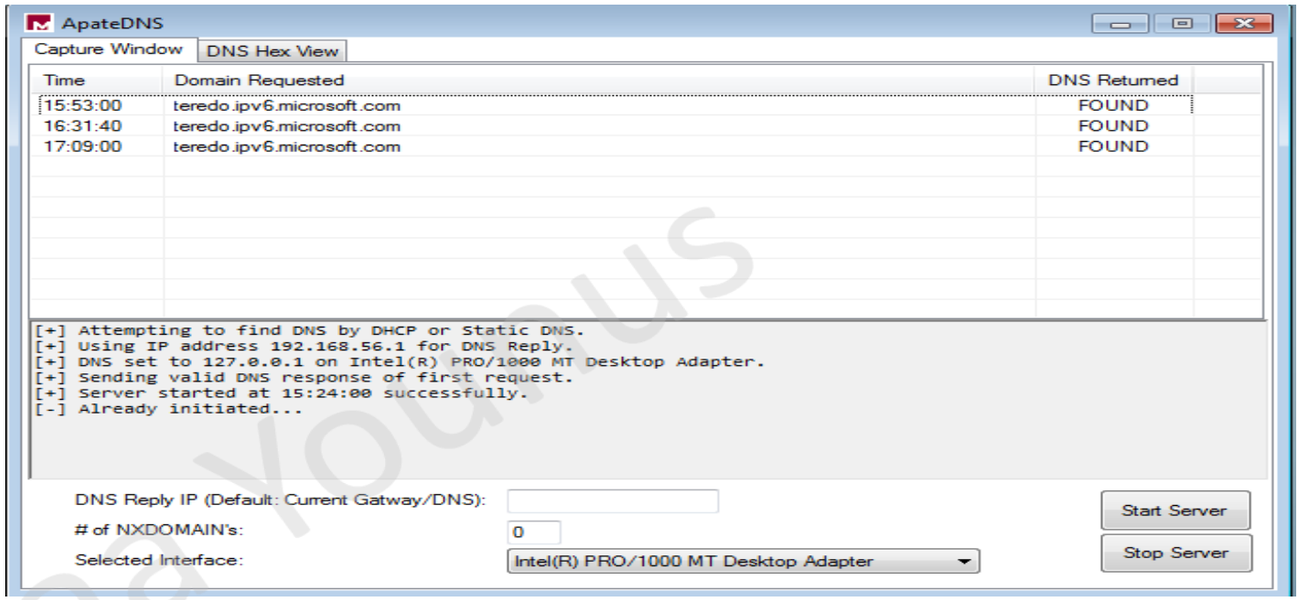


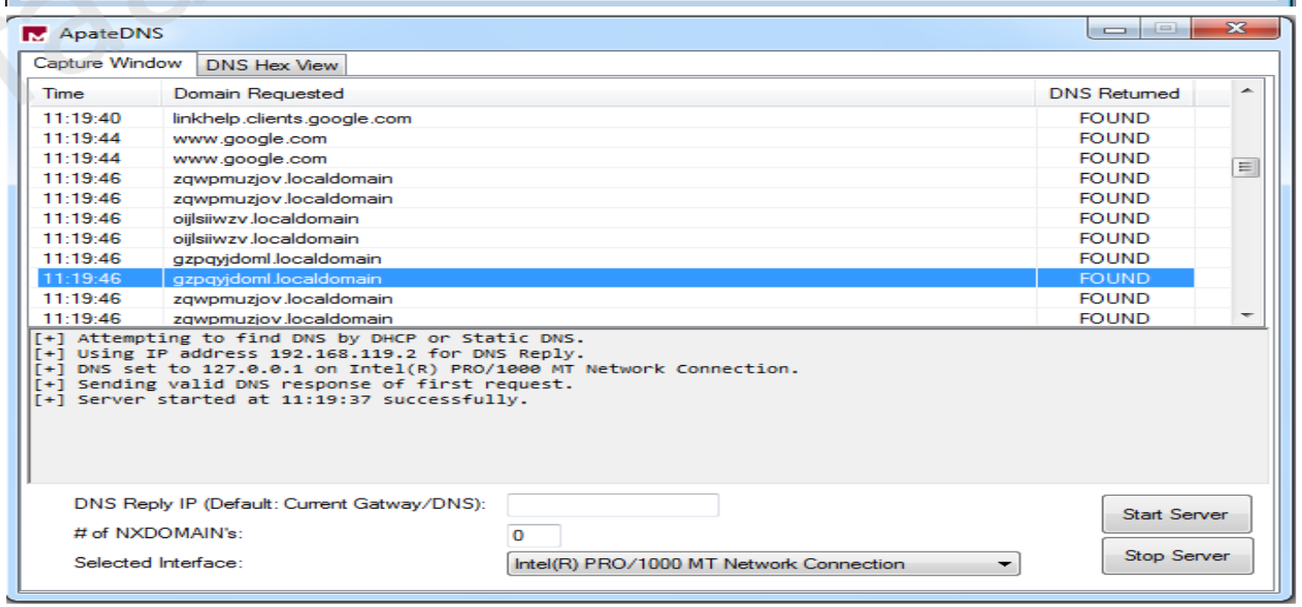
Faking a Network

- let malware believe it has normal Internet connectivity so it will perform network behavior (beacons, DNS lookups, C2 traffic) without actually touching the real Internet.
- Collect network indicators (domains, IPs, request patterns, packet signatures) safely for analysis and signature creation.
- tools
  - FakeNet-NG — lightweight network simulator that intercepts and responds to many protocols. Easy for quick sorting.
  - INetSim — more full-featured Internet services simulator (DNS, HTTP, SMTP, FTP, etc.).
  - **Network redirection/iptables and DNS spoofing** (for custom responses).
  - Proxy tools (mitmproxy) — for inspecting HTTP/HTTPS if you control certs.
  - Wireshark / tcpdump — capture traffic for offline analysis.

ApateDNS

- **ApateDNS** is a lightweight DNS responder/spoofer tool that listens on UDP/53 and replies to DNS queries with an IP address you choose.
- quickly see which domain names malware tries to resolve and direct those names to a local or fake server (so malware continues execution without touching the real Internet).





Monitoring with Netcat

- Netcat (nc) is a lightweight, flexible TCP/UDP tool — “the TCP/IP Swiss Army knife.”
- Two main modes:
  - Listen/server mode (-l) — waits for inbound connections.
  - Connect/client mode — initiates outbound connections to a target.
- Data read from stdin is sent over the network; data received is written to stdout.
- Run Netcat in listen mode on the expected port (e.g., port 80 for HTTP-based C2)
  - C2 (Command and Control): The malware uses the C2 channel to send commands (from the attacker) and receive results (from the victim).
- so you can capture whatever the malware sends:

nc -l -p 80

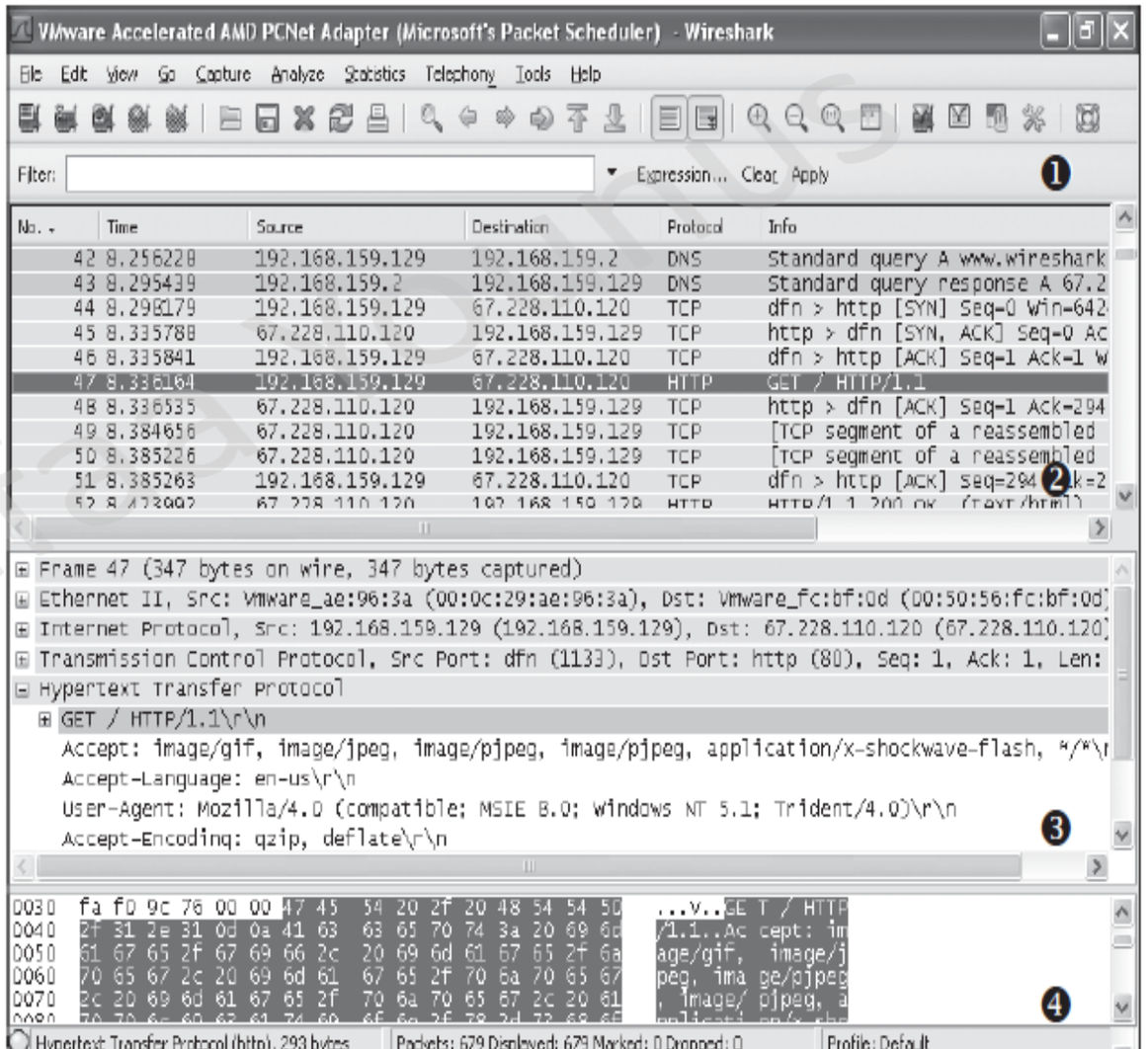
- Start the malware in the isolated guest VM. When the malware connects, Netcat will print the raw request (HTTP headers, POST body, or shell input) to your terminal.

```
POST /cq/frame.htm HTTP/1.1
Host: www.google.com ❷
User-Agent: Mozilla/5.0 (Windows; Windows NT 5.1; TWFsd2FyZUh1bnRlcg==;
rv:1.38)
Accept: text/html, application
Accept-Language: en-US, en;q=
Accept-Encoding: gzip, deflate
Keep-Alive: 300
Content-Type: application/x-form-urlencoded
Content-Length

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

Z:\Malware> ❸
```

Packet Sniffing with Wireshark

- **Wireshark is an** Open-source packet capture and analysis tool.
- Captures network traffic, lets you inspect packet details, follow sessions, and analyze protocols.

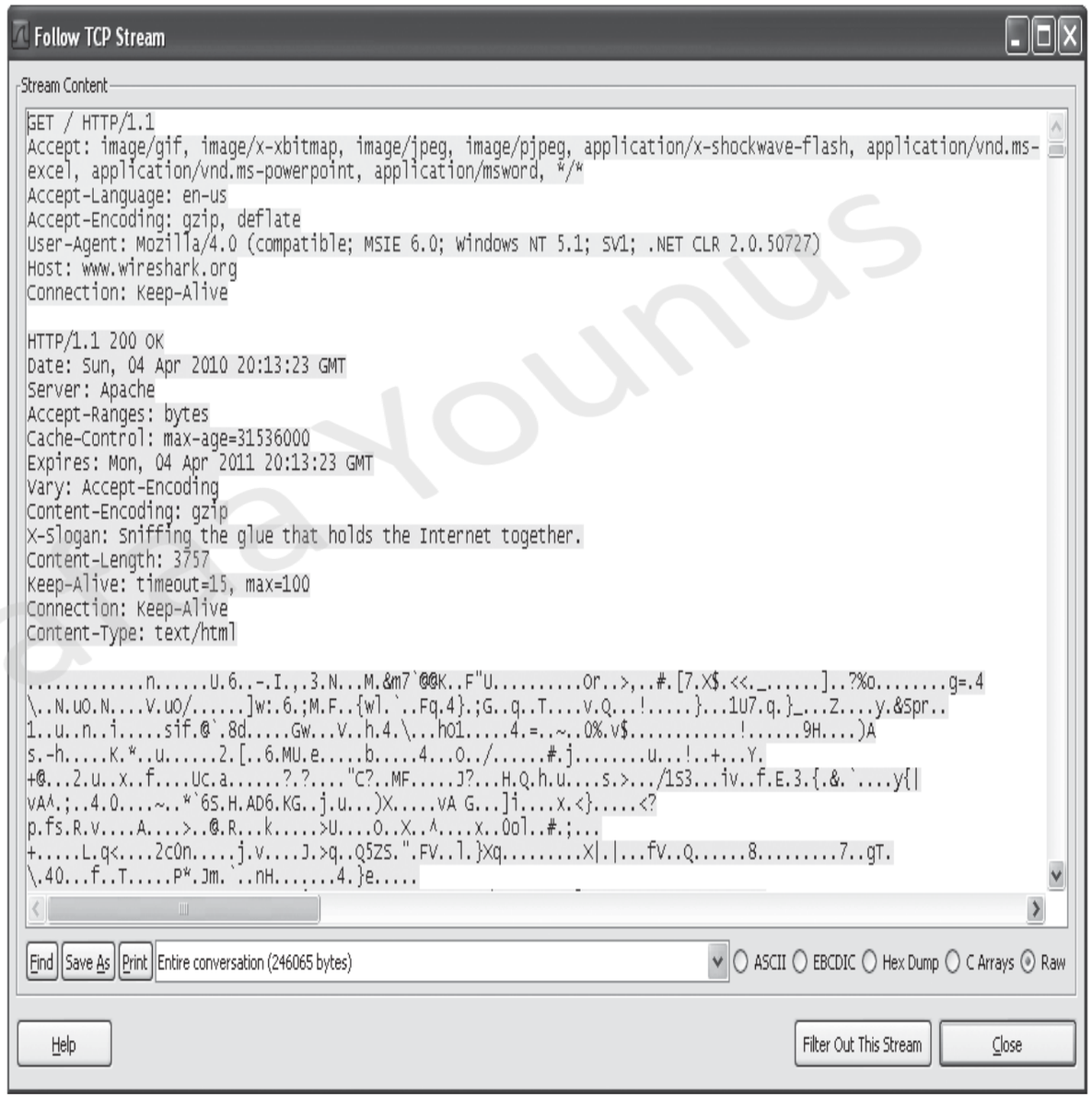
1. **Filter box** — enter display filters (e.g., http, ip.addr == 10.0.0.5, tcp.port == 80).
2. **Packet list** — all packets that match the display filter (one line per packet).
3. **Packet details** — decoded protocol tree for the selected packet.
4. **Hex / bytes pane** — raw packet bytes; linked to the decoded fields.



Follow a session: Right-click a TCP packet → **Follow** → **TCP Stream** to view the full conversation (colored per side). Great for reading HTTP POST bodies or command channels.

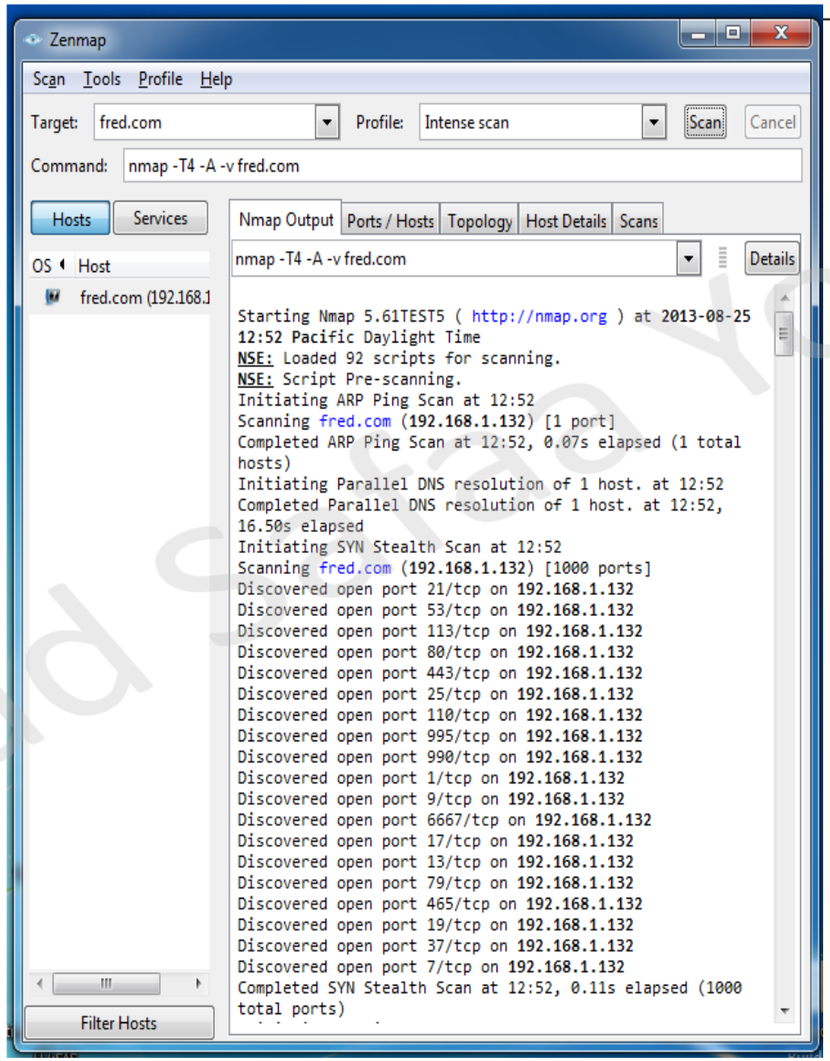
Save PCAP: File → Save/Export — keep PCAPs for later analysis or evidence.

Use with other tools: correlate PCAPs with Procmon, Netcat logs, and system artifacts.



INetSim

- INetSim is a free Linux-based suite that simulates common Internet services (HTTP, DNS, SMTP, FTP, IRC, etc.) for malware analysis.
- Run it on a Linux VM on the same isolated virtual network as your malware analysis VM.
- Let malware believe it has Internet access so it will perform network actions (beacons, C2, downloads) without touching the real Internet.
- Collects network indicators (domains, URIs, payloads) safely.
- Emulates many services by default (DNS 53, HTTP 80, HTTPS 443, SMTP 25, FTP 21, IRC 6667, TFTP 69, POP3 110, and many more).
- HTTP/HTTPS: serves almost any requested file (prevents 404s that would stop malware).




```
* dns 53/udp/tcp - started (PID 9992)
* http 80/tcp - started (PID 9993)
* https 443/tcp - started (PID 9994)
* smtp 25/tcp - started (PID 9995)
* irc 6667/tcp - started (PID 10002)
* smtps 465/tcp - started (PID 9996)
* ntp 123/udp - started (PID 10003)
* pop3 110/tcp - started (PID 9997)
* finger 79/tcp - started (PID 10004)
* syslog 514/udp - started (PID 10006)
* tftp 69/udp - started (PID 10001)
* pop3s 995/tcp - started (PID 9998)
* time 37/tcp - started (PID 10007)
* ftp 21/tcp - started (PID 9999)
* ident 113/tcp - started (PID 10005)
* time 37/udp - started (PID 10008)
* ftps 990/tcp - started (PID 10000)
* daytime 13/tcp - started (PID 10009)
* daytime 13/udp - started (PID 10010)
* echo 7/tcp - started (PID 10011)
* echo 7/udp - started (PID 10012)
* discard 9/udp - started (PID 10014)
* discard 9/tcp - started (PID 10013)
* quotd 17/tcp - started (PID 10015)
* quotd 17/udp - started (PID 10016)
* chargen 19/tcp - started (PID 10017)
* dummy 1/udp - started (PID 10020)
* chargen 19/udp - started (PID 10018)
* dummy 1/tcp - started (PID 10019)
```

Basic Dynamic Tools in Practice

- **Start Procmon** and **clear display**.
  - Add a filter for the malware process name (e.g., msts.exe) so you only see its events.
- **Open Process Explorer** to watch processes, DLLs, handles, network connections.
- **Take a registry snapshot** with Regshot (click **1st Shot**).
- **Prepare the virtual network**:
  - Run **INetSim** on a Linux analysis VM (HTTP/HTTPS/DNS/etc.).
  - Run **ApateDNS** on the Windows analysis host (redirect guest DNS to INetSim).
- **Start packet capture** with Wireshark on the internal/host-only network interface.

References

- Sikorski, M., & Honig, A. (2012). *Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software*. No Starch Press.
- Bowne, S. (n.d.). *CNIT 126 Ch 2: Malware Analysis in Virtual Machines & 3: Basic Dynamic Analysis* [PowerPoint slides]. SlideShare. https://www.slideshare.net/slideshow/cnit-126-2-malware-analysis-in-virtual-machines-3-basic-dynamic-analysis/71069826
- Bowne, S. (n.d.). *CNIT 126: Ch 2-3 Basic Dynamic Analysis* [Video]. YouTube. https://www.youtube.com/watch?v=Nsy8U8UAbHk