Malware Analysis

Basic Dynamic Analysis

Compiled by: Dharmesh D. Dave | Asst. Prof. | National Forensic Sciences University

Introduction

- Basic Dynamic Analysis is the next step after completion of basic static analysis, here the malware is executed in the controlled environment for monitoring it's behaviour, functionality and characteristics.
- Dynamic analysis lets you observe the malware's true functionality, because, for example, the existence of an action string in a binary does not mean the action will actually execute.
- Dynamic analysis may put your system and network at risk.
- Dynamic techniques do have their limitations, because not all code paths may execute when a piece of malware is run.

Limitation(s)

- In the case of command-line malware that requires arguments, each argument could execute different program functionality, and without knowing the options you wouldn’t be able to dynamically examine all of the program’s functionality.
- Cannot identify its programming logic of malicious functions and behavior.

Executing DLL

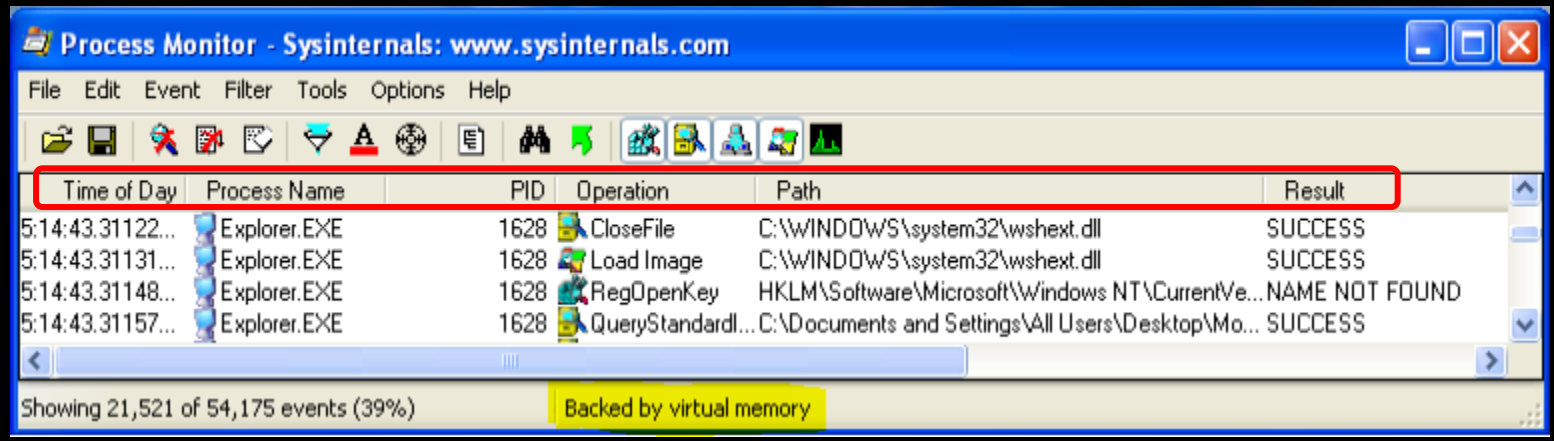
- Because malicious DLLs frequently run most of their code in DLLMain (called from the DLL entry point), and because DLLMain is executed whenever the DLL is loaded, you can often get information dynamically by forcing the DLL to load using rundll32.exe.
- It can be tricky to launch malicious DLLs because Windows doesn't know how to run them automatically.
- Rundll32.exe provides a container for running a DLL using this syntax in cmd:
- **>rundll32.exe *Dllname, Functionname(Ordinal) argument***
- Examples:
- **>rundll32.exe rip.dll, Install**
- >rundll32.exe xyzzy.dll, #5

Process Monitor

- Procmon is an advanced monitoring tool for Windows that provides a way to monitor certain registry, file system, network, process, and thread activity.
- It combines and enhances the functionality of two legacy tools: FileMon and RegMon.
- Although procmon captures a lot of data, it doesn't capture everything.
- For example, it *can miss the device driver activity of a user-mode component* talking to a rootkit via device I/O controls, as well as certain GUI calls, such as SetWindowsHookEx.

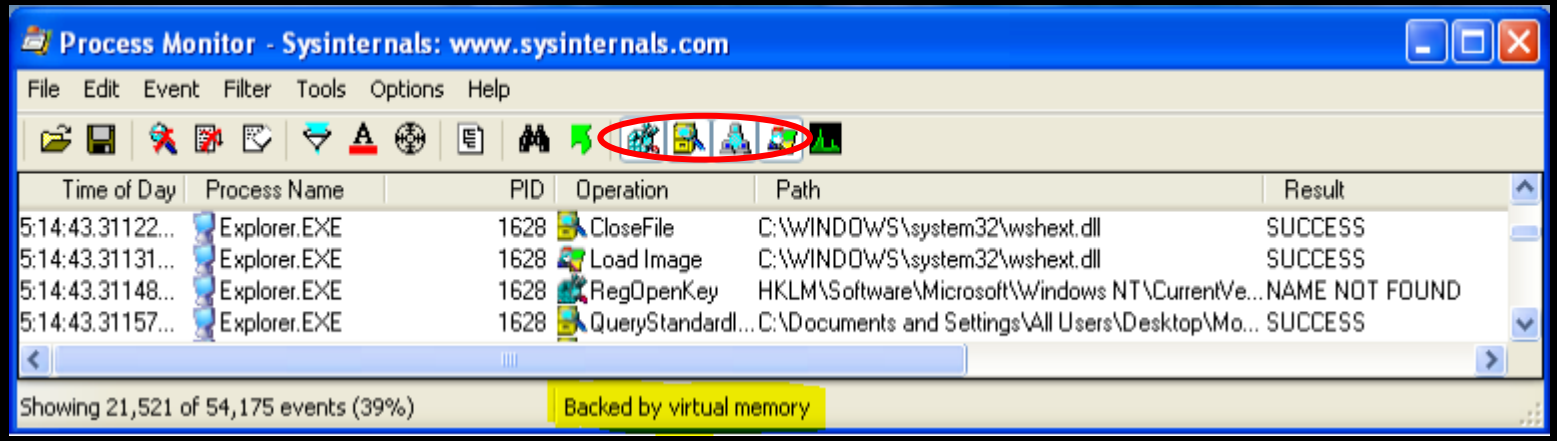
Process Monitor

- Procmon monitors all system calls it can gather as soon as it is run.
- Because many system calls exist on a Windows machine (sometimes more than 50,000 events a minute), it's usually impossible to look through them all.
- As a result, because procmon uses RAM to log events until it is told to stop capturing, it can crash a virtual machine using all available memory.



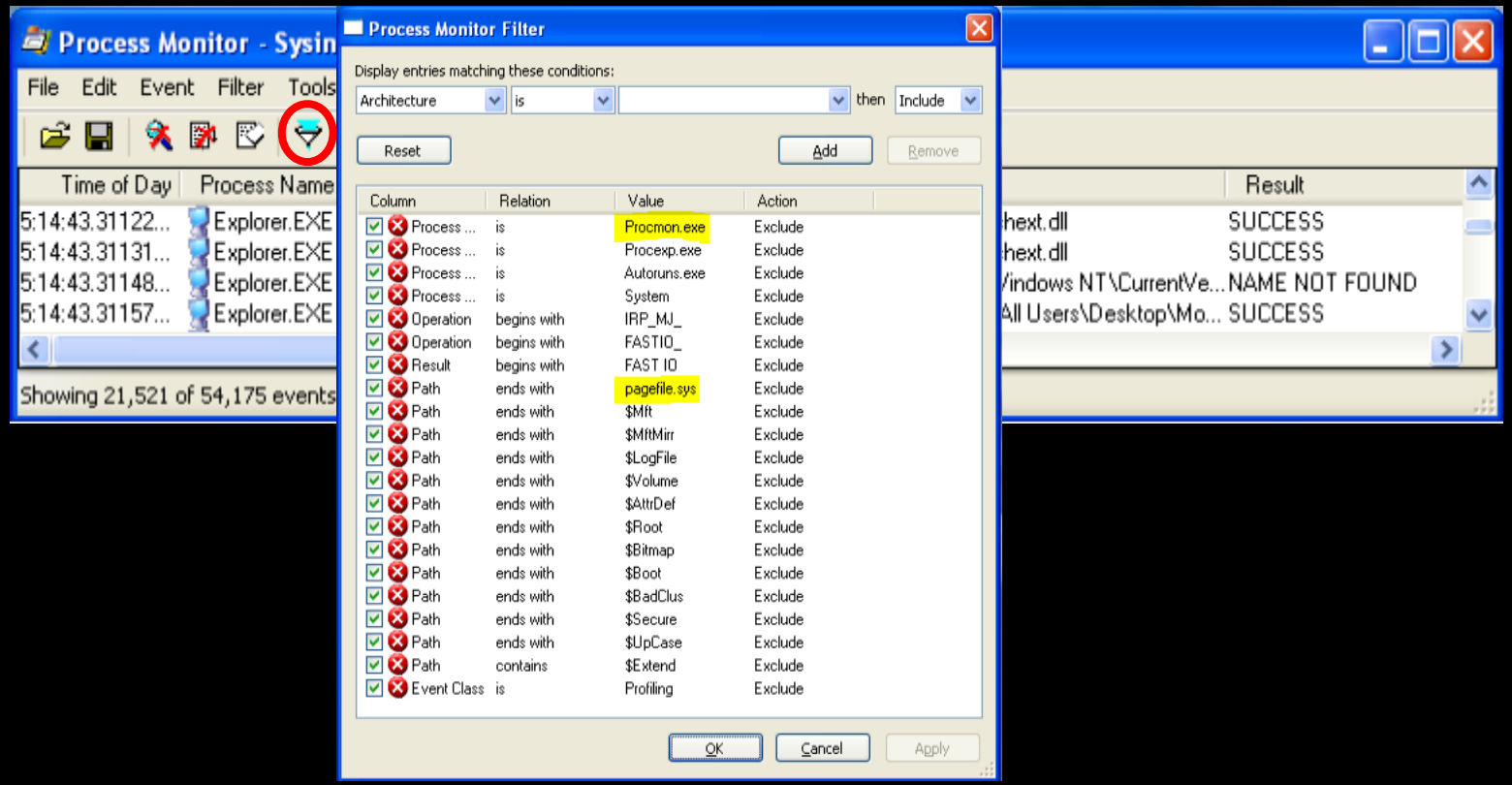
Process Monitor

- Automatic Filters of Procmon:
- **Registry** By examining registry operations, you can tell how a piece of malware installs itself in the registry.
- **File system** Exploring file system interaction can show all files that the malware creates or configuration files it uses.
- **Process activity** Investigating process activity can tell you whether the malware spawned additional processes.
- **Network** Identifying network connections can show you any ports on which the malware is listening.



Process Monitor

- Procmon uses some basic filters by default. For example, it contains a filter that excludes procmon.exe and one that excludes the pagefile from logging, because it is accessed often and provides no useful information.



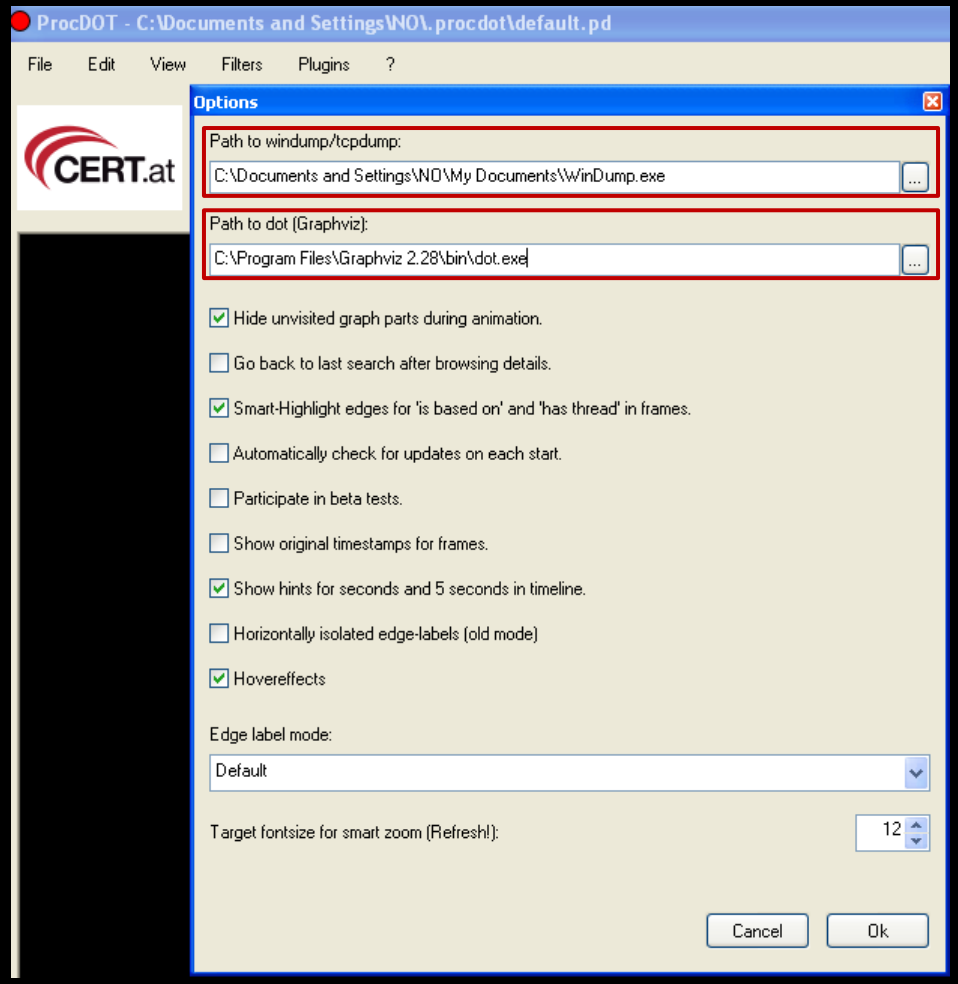
Process Monitor

- If your malware runs at boot time, use procmon's boot logging options to install procmon as a startup driver to capture startup events.

ProcDOT

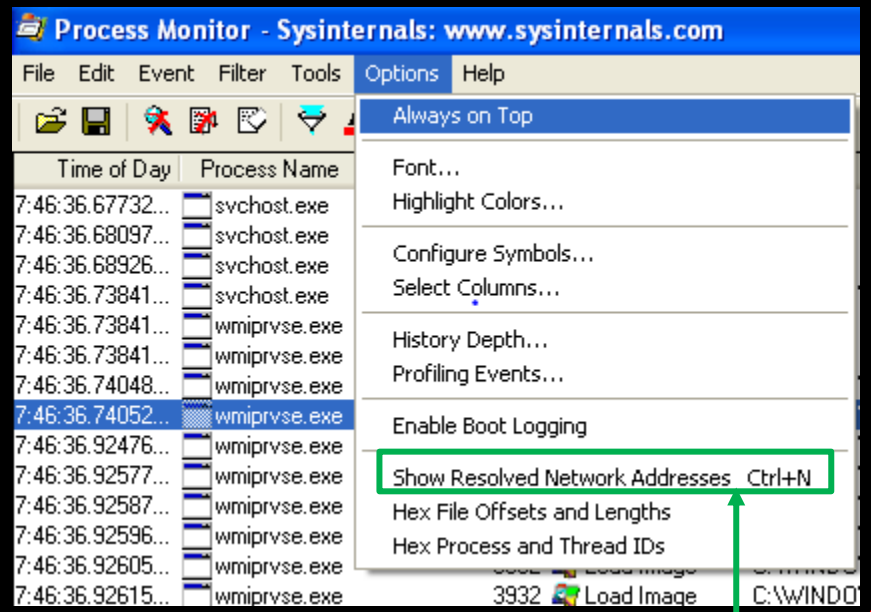
- ProcDOT is not necessarily a malware analysis tool. **ProcDOT is a tool that visualizes system activities in a very convenient way.**
- ProcDOT has some of the below features:
- something that lets us get an overall guts feeling for an entire situation within a glance,
- something that enables us to spot relevant parts and understand the correlation between them in minutes, and finally
- something that provides us a perfect starting point for deeper, detailed and even code level analysis if necessary.

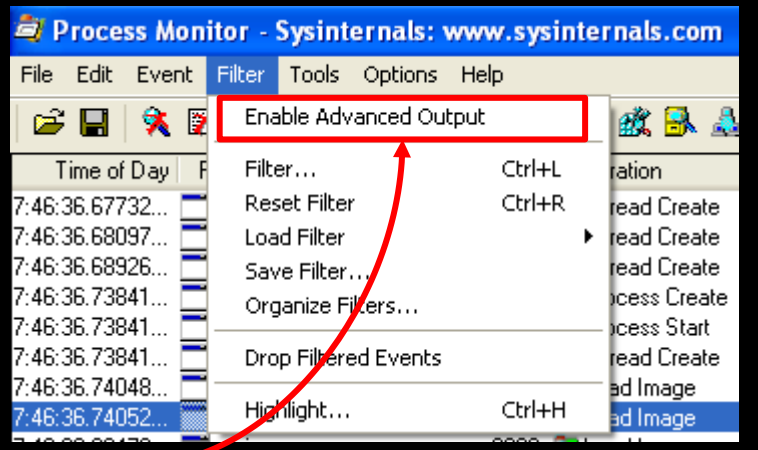
ProcDOT

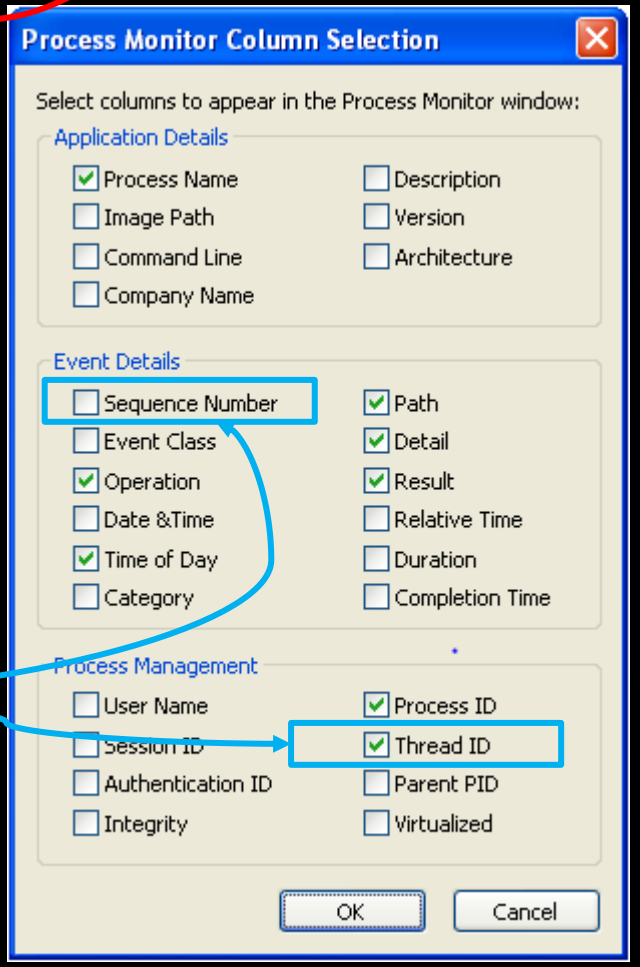


- **ProcDOT Prerequisites:**
- Read the readme file and follow it.
- Installing Windump *(Copy the downloaded executable (WinDump.exe) to a place of your taste (presumption: you need to provide this location in ProcDOT's options later)*
- Installing Graphviz

ProcDOT

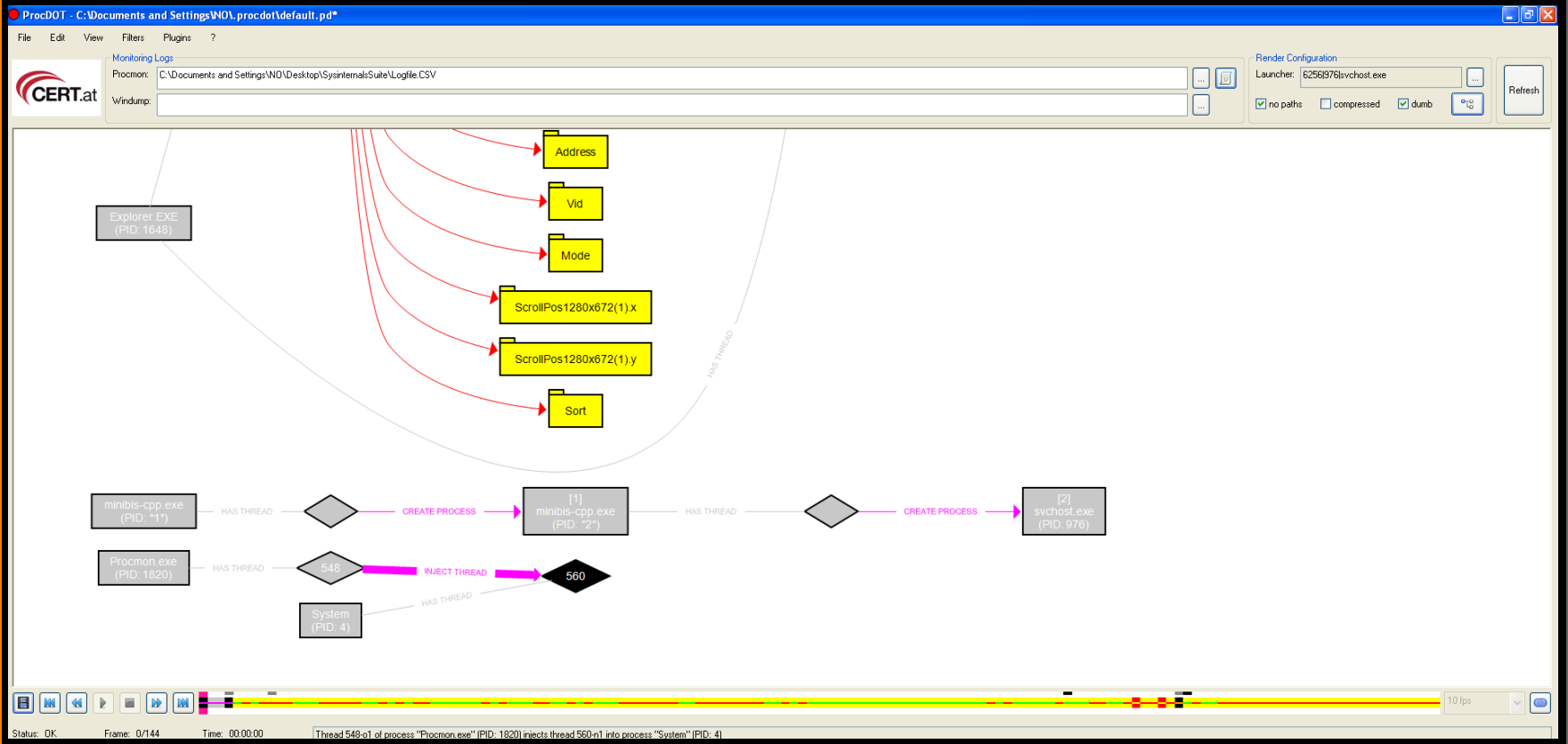






- **You need to adjust Procmon's configuration Therefore, in Procmon ...**
- **disable (uncheck) "Show Resolved Network Addresses" (Options)**
- **disable (uncheck) "Enable Advanced Output" (Filter)**
- adjust the displayed columns **(Options > Select Columns ...)**
  - **to not show the "Sequence" column**
  - **to show the "Thread ID" column**

ProcDOT

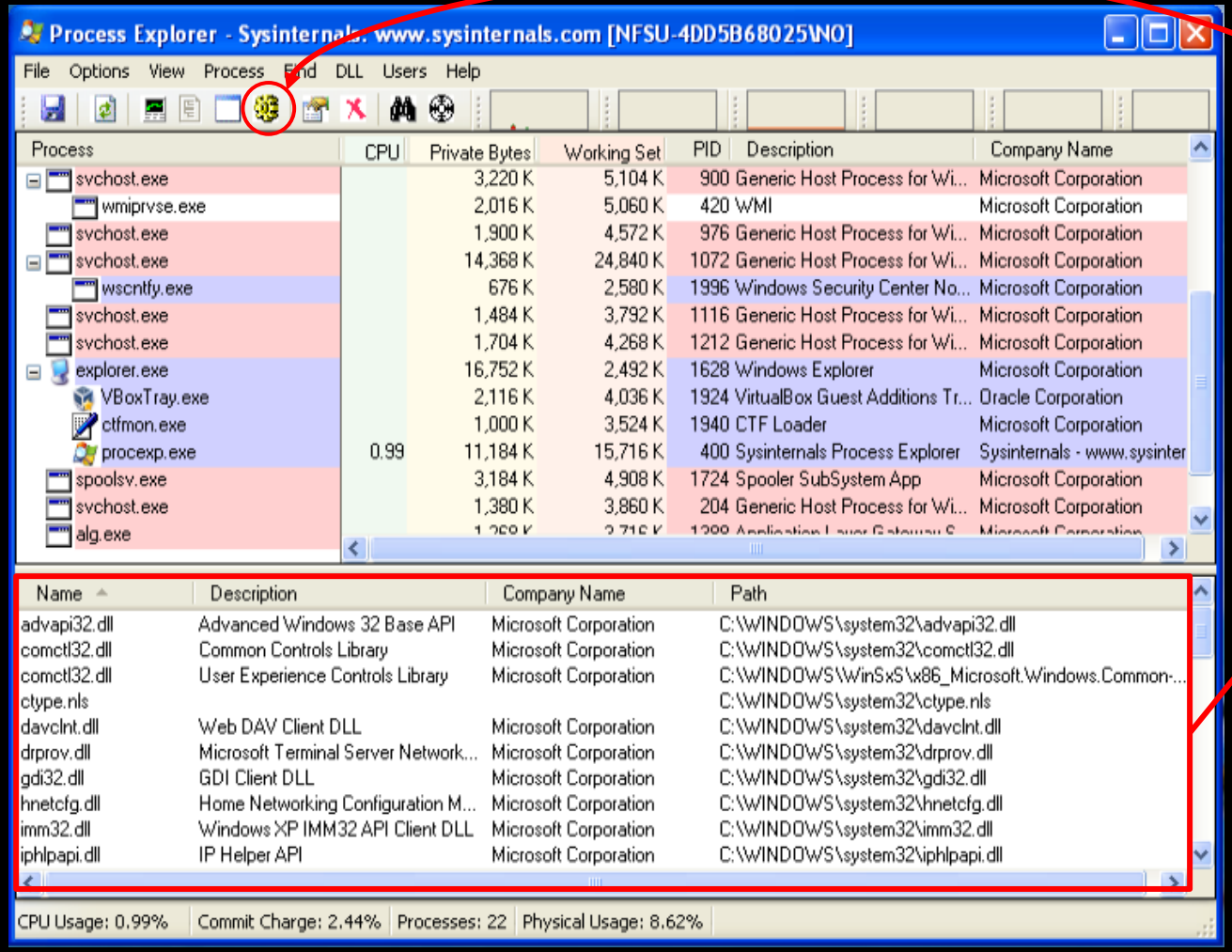


Save Procmon file and open in ProcDOT to visualize the system activities

Process Explorer

- Process explorer is an extremely powerful task manager. It can provide valuable insight into the processes currently running on a system.
- Process Explorer can be used to list active processes, DLLs loaded by a process, various process properties, and overall system information.
- It can also be used it to kill a process, log out users, and launch and validate processes.
- Process Explorer shows five columns: Process (the process name), PID (the process identifier), CPU (CPU usage), Description, and Company Name.

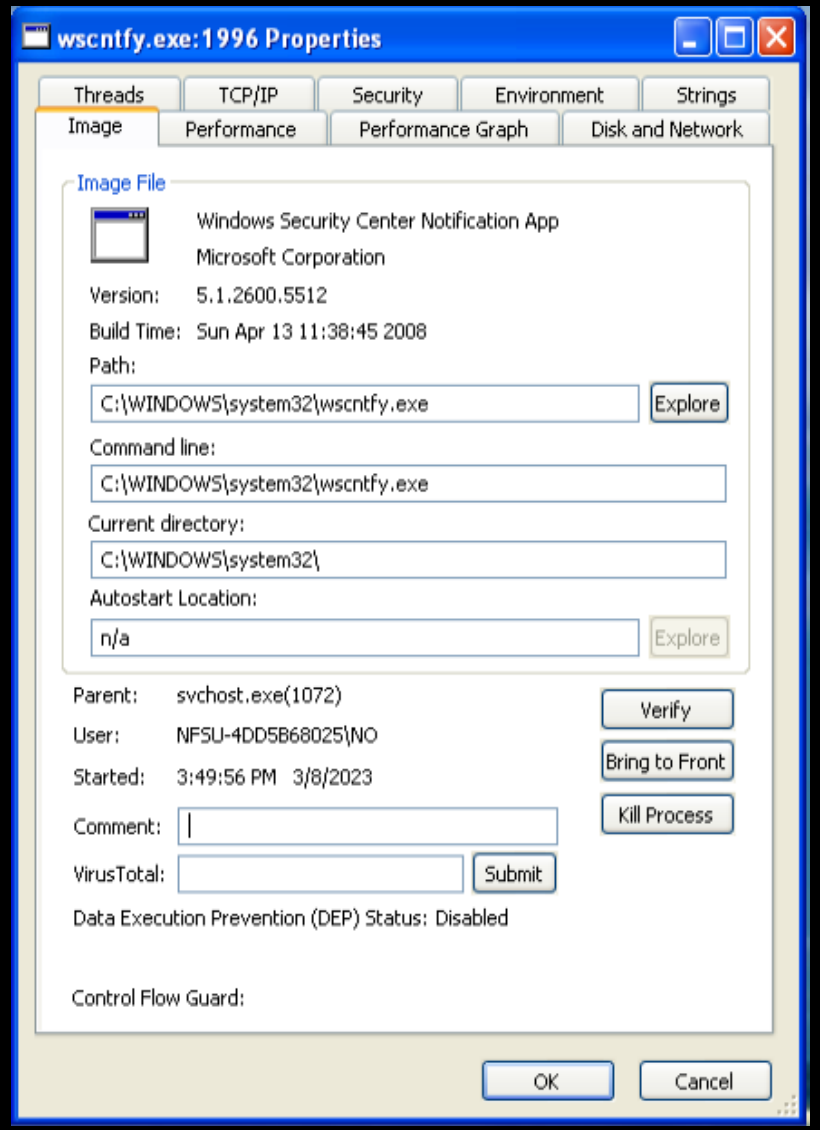
Process Explorer



Change the lower Panel details either **DLL** display OR **Handles** window.

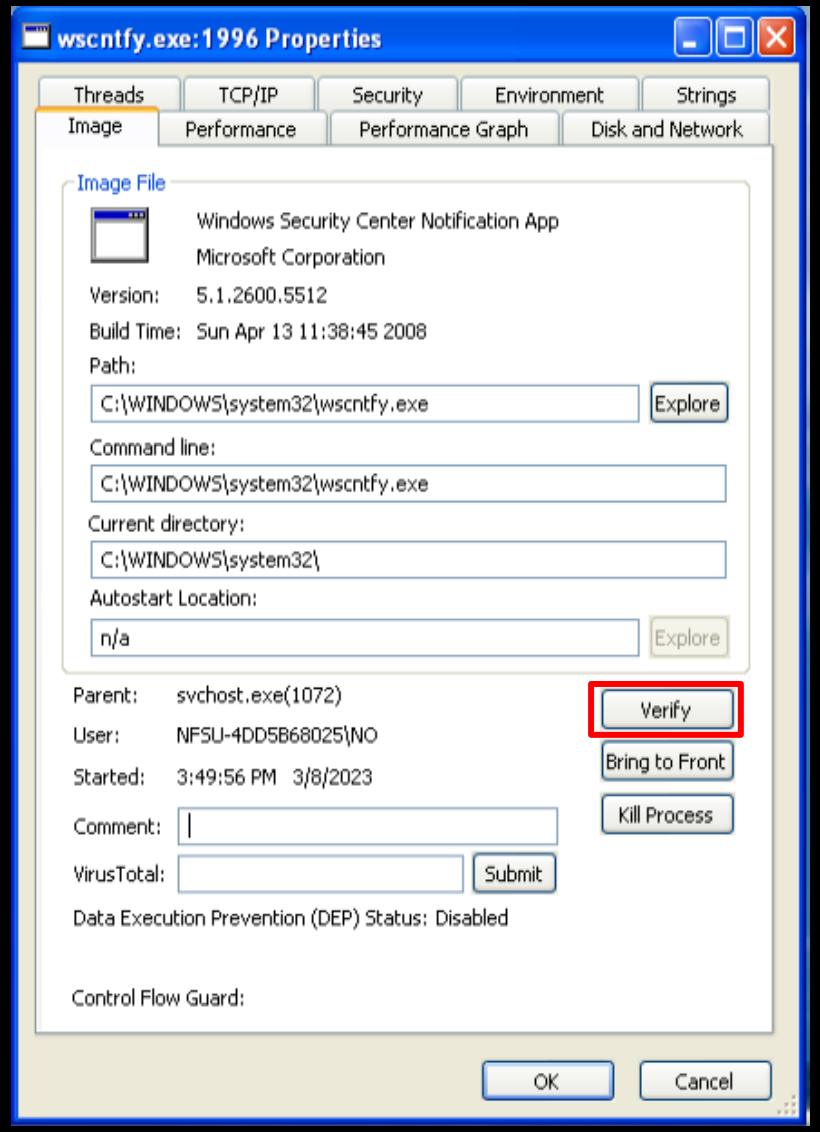
services are highlighted in pink, processes in blue, new processes in green, and terminated processes in red.

Process Explorer



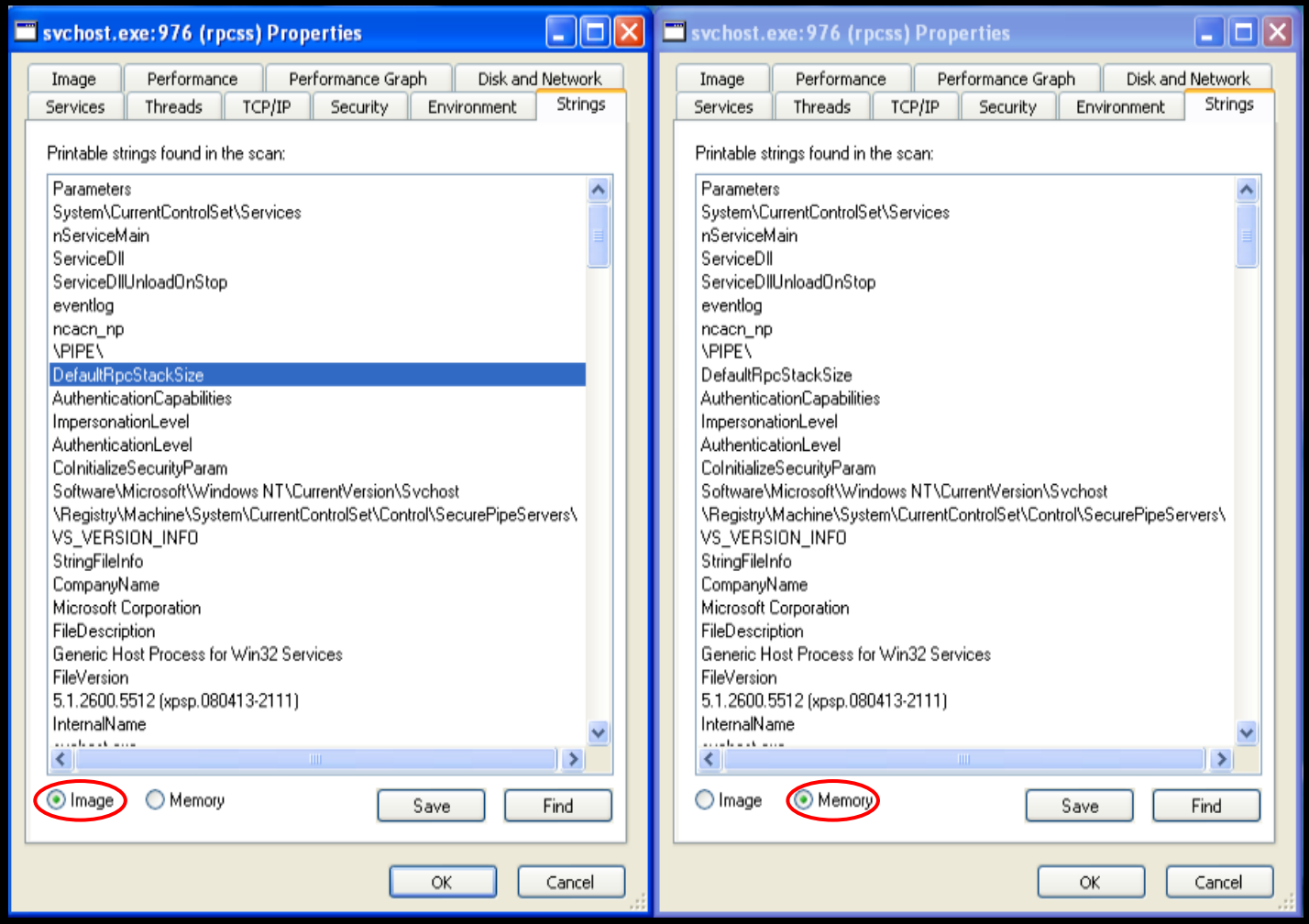
- This window (Property of Process) can provide some particularly useful information about the subject malware.
- The Threads tab shows all active threads
- The TCP/IP tab displays active connections or ports on which the process is listening, and
- the Image tab (opened in the figure) shows the path on disk to the executable.

Process Explorer



- Click this button to verify that the image on disk is, in fact, the **Microsoft signed binary**. Because Microsoft uses digital signatures for most of its core executables.
- The Verify button verifies the image on disk rather than in memory, and it is useless if an attacker uses process replacement, which involves running a process on the system and overwriting its memory space with a malicious executable.

Process Explorer



- One way to recognize process replacement is to use the Strings tab in the Process Properties window to **compare the strings** contained in the disk executable (image) against the strings in memory for that same executable running in memory. If observed drastically different then likely chance of process replacement.

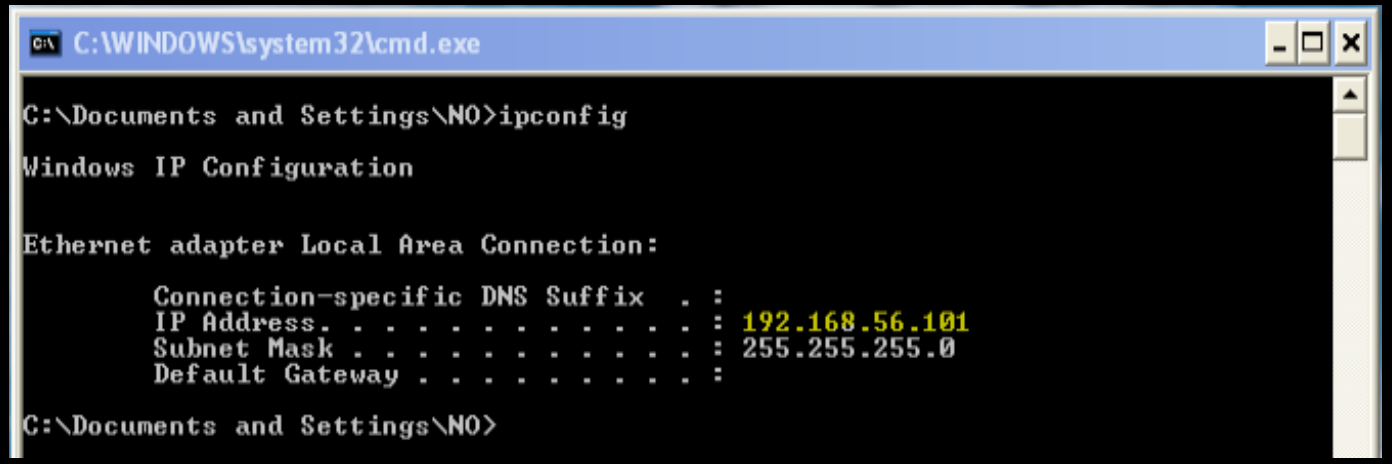
Analyse Malicious Document Process Explorer

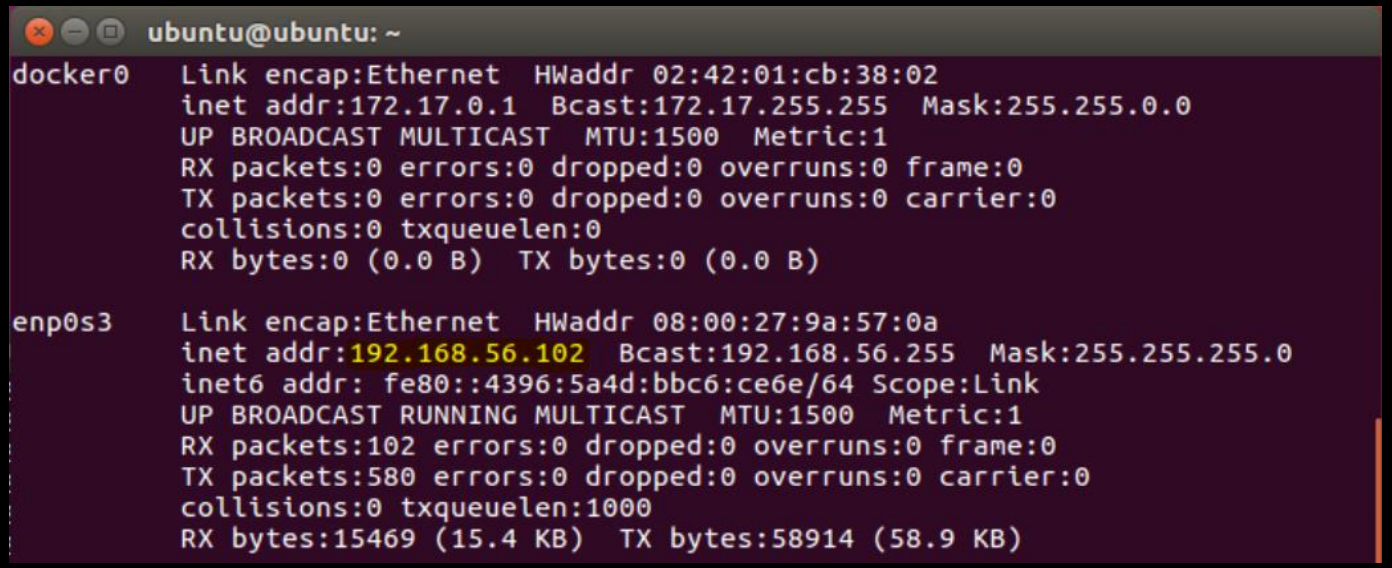
- A quick way to determine whether a document is malicious is to open Process Explorer and then open the suspected malicious document.
- If the document launches any processes, you should see them in Process Explorer, and be able to locate the malware on disk via the Image tab of the Properties window.

RegShot

- Regshot is an open source registry comparison tool that allows you to take and compare two registry snapshots.
- Simply take the first shot by clicking the 1st Shot button, and then run the malware and wait for it to finish making any system changes. Next, take the second shot by clicking the 2nd Shot button. Finally, click the Compare button to compare the two snapshots.
- You will find the difference between first shot and second shot for any modification, addition or deletion in the registry.

Faking Internet





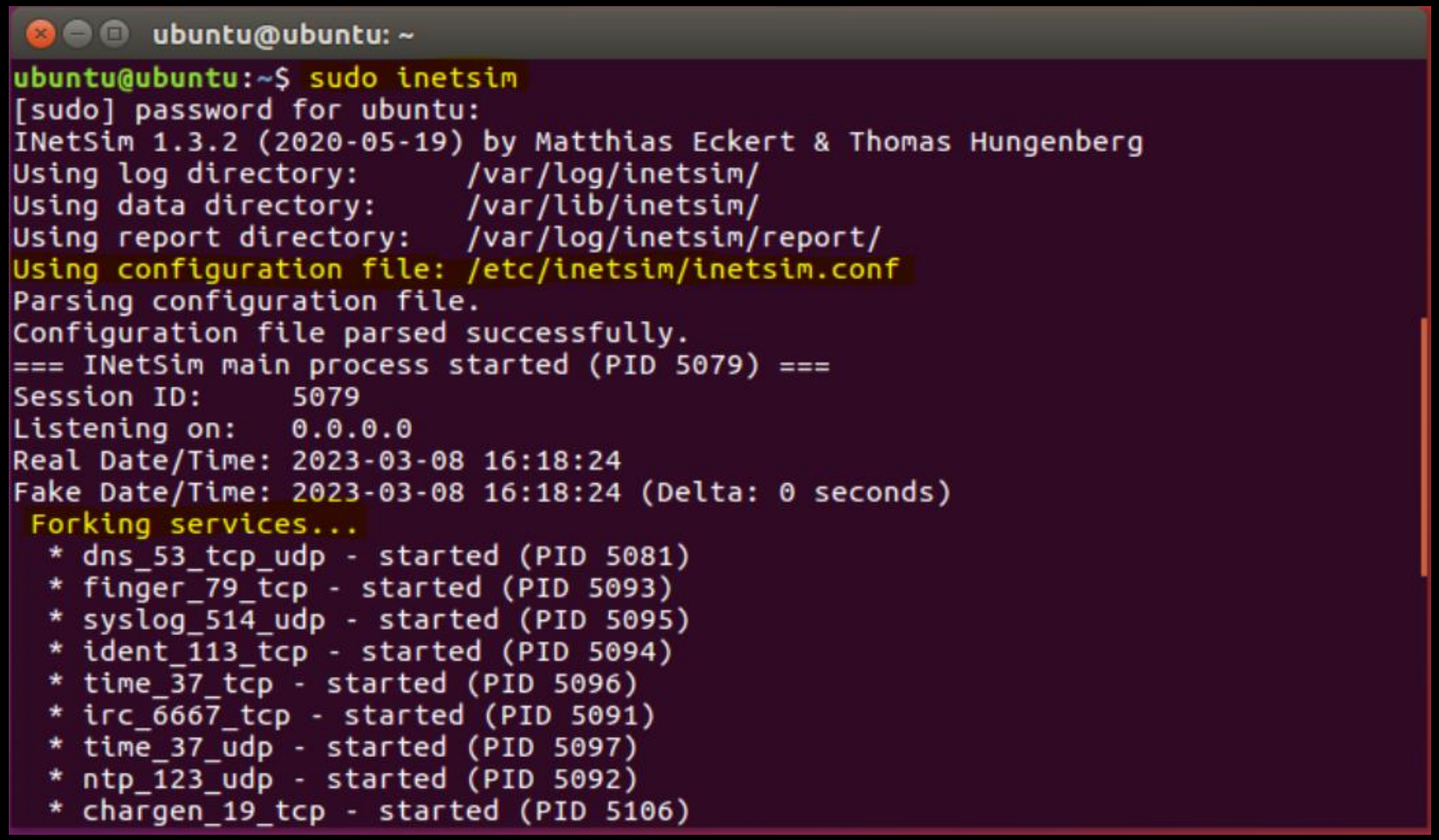
Faking Internet (inetsim)

- Install inetsim > apt-get install inetsim
- Make Changes in the inetsim.conf file:

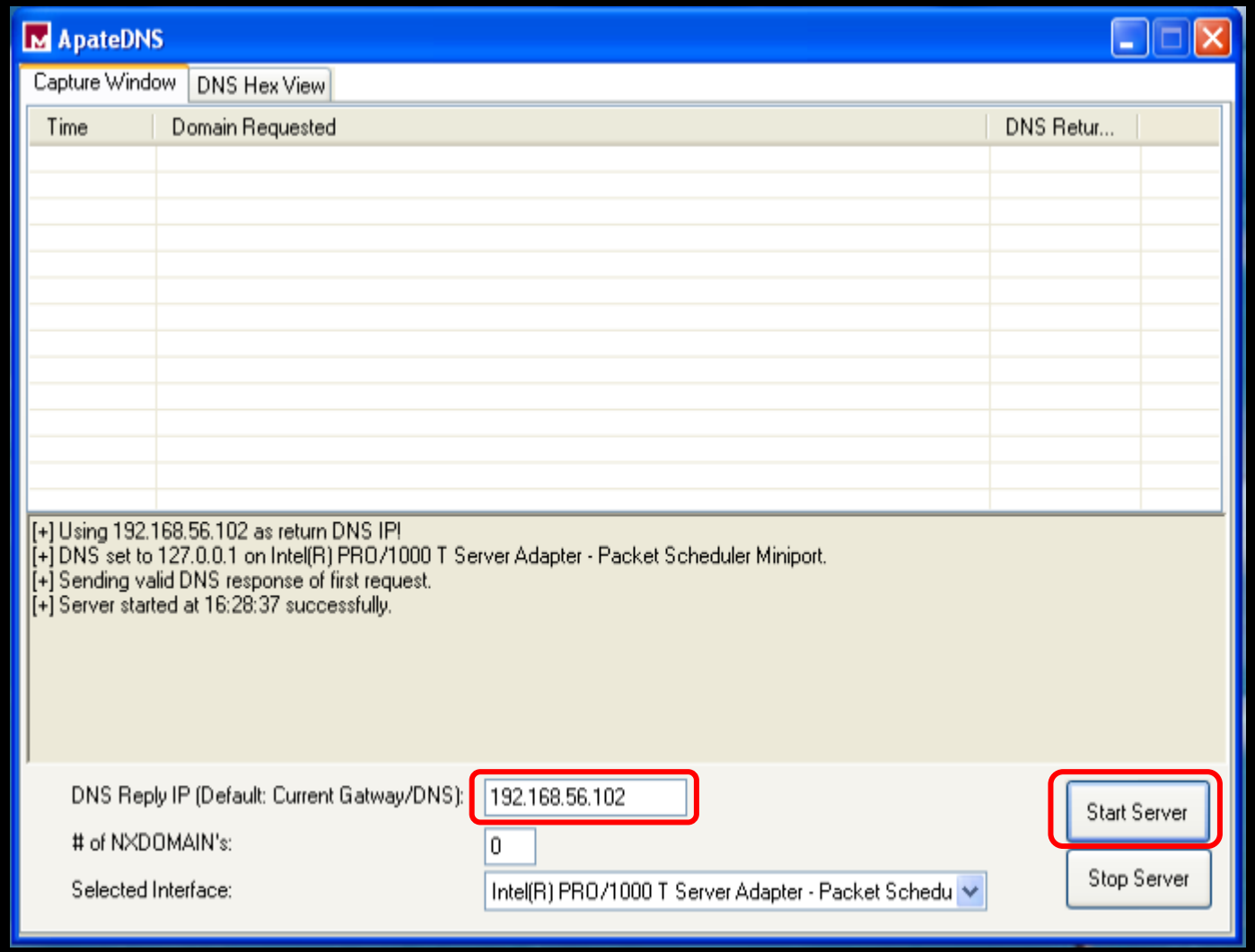
- #service_bind_address 10.10.10.1 Win Machine IP OR 0.0.0.0
- #dns_default_ip 10.10.10.1 Linux Machine IP
- #dns_default_domainname some.domain
- #dns_static www.foo.com 10.10.10.10

- Remove # while changing the inetsim.conf file.

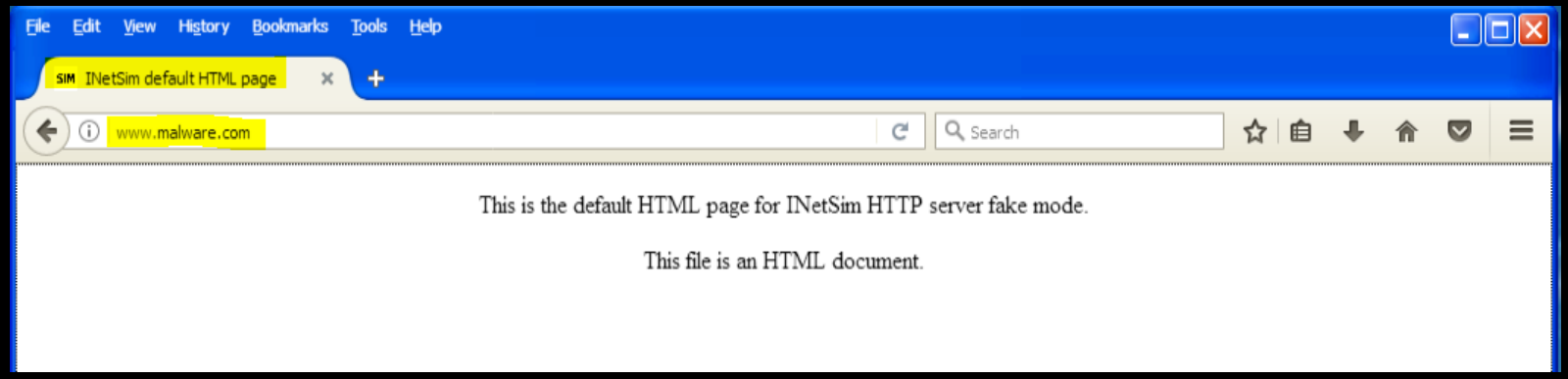
Faking Internet (inetsim)

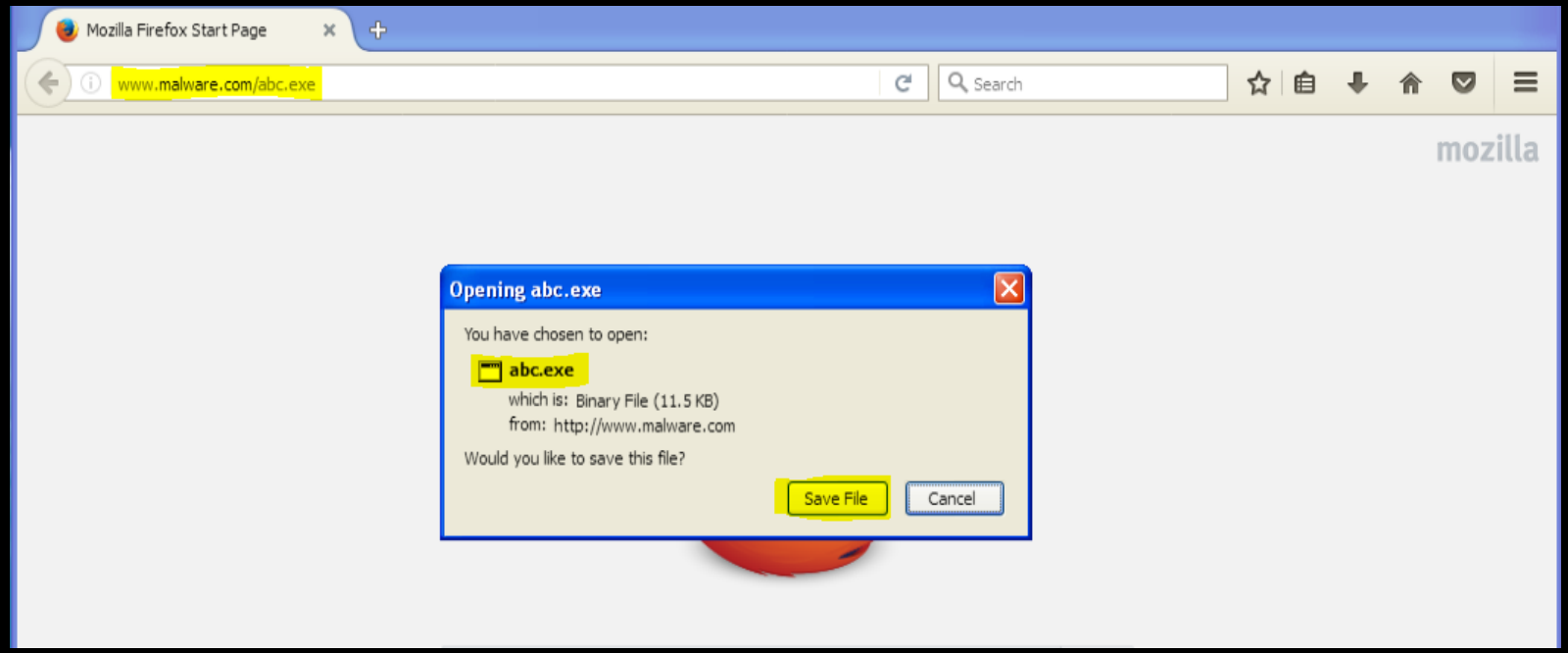


Faking Internet (ApateDNS)

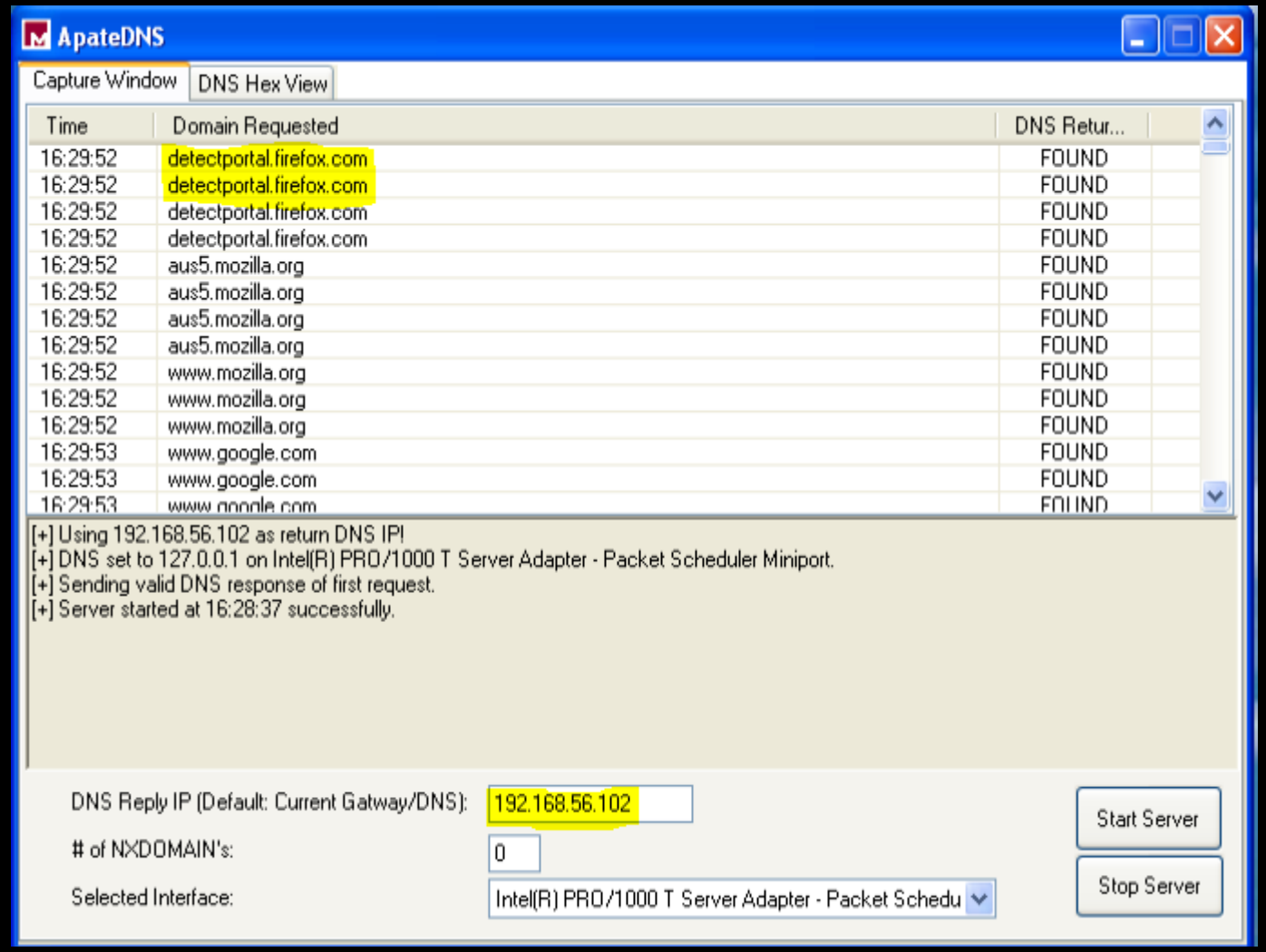


Faking Internet





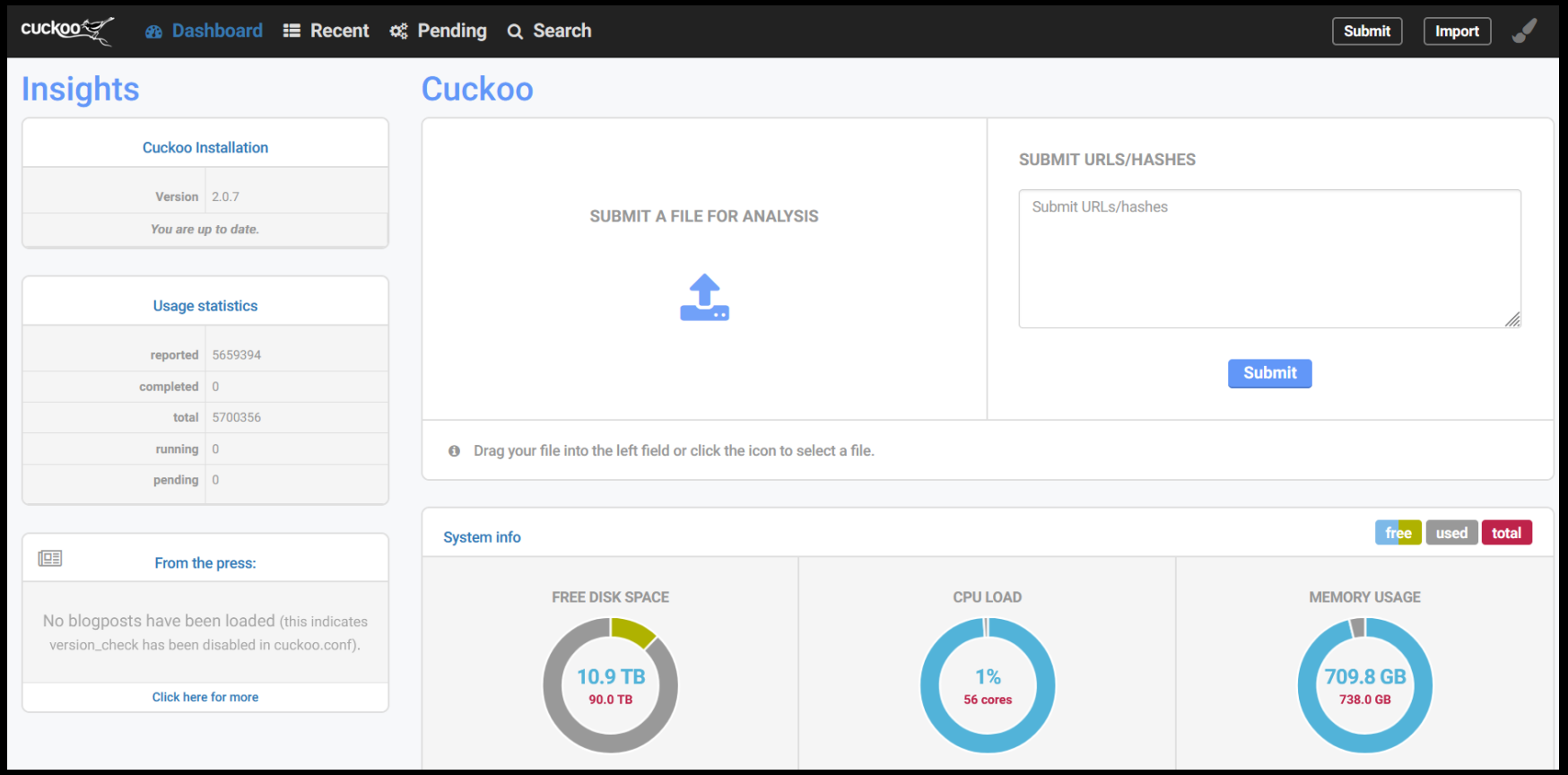
Faking Internet (ApateDNS)



Sandbox

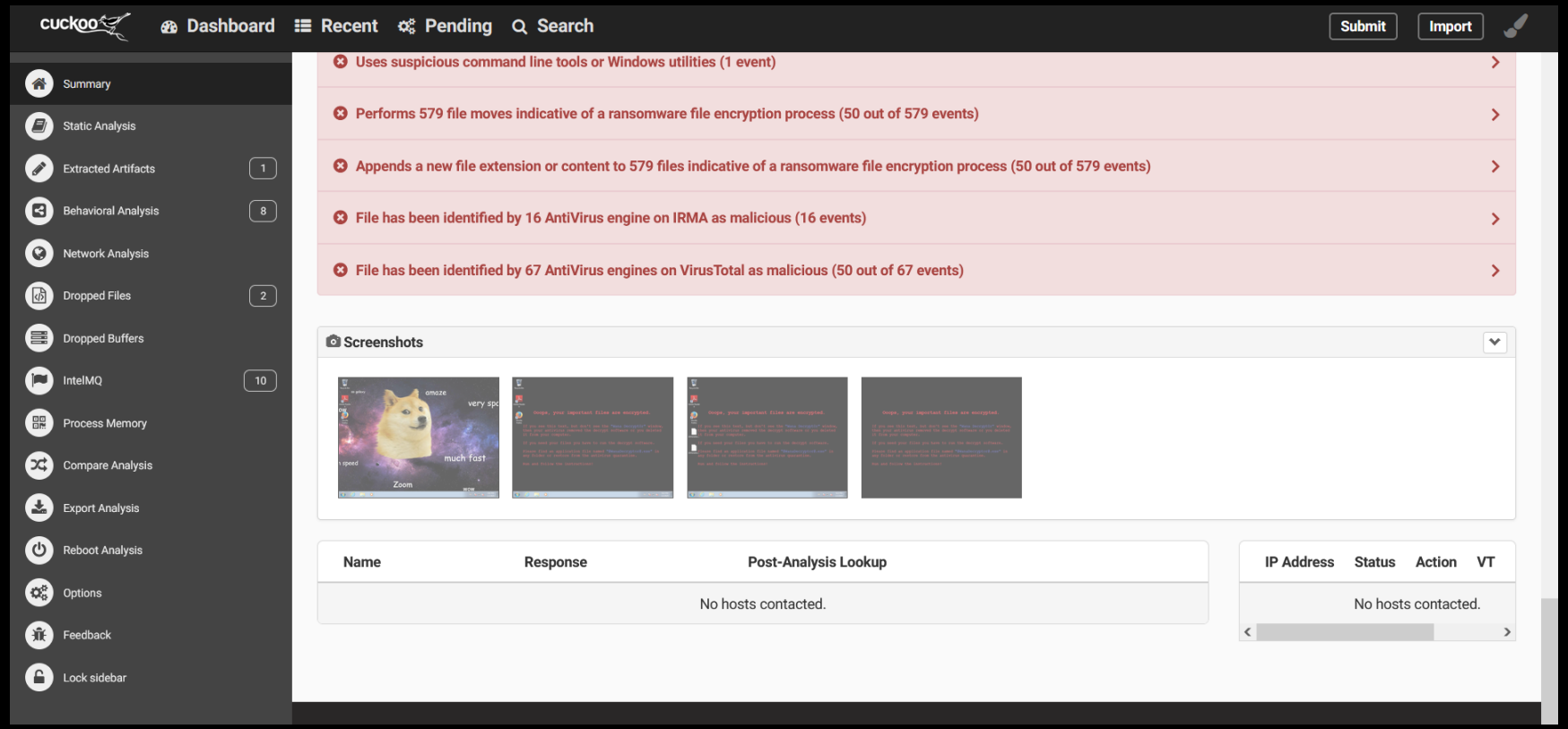
- "A system that allows an untrusted application to run in a highly controlled environment where the application's permissions are restricted to an essential set of computer permissions." – NIST
- "A sandbox is a security mechanism for running untrusted programs in a safe environment without fear of harming "real" systems."
- *The isolation metaphor is taken from the idea of children who do not play well together, so each is given their own sandbox to play in alone.*
- Ex. Norman SandBox, GFI Sandbox, Joe Sandbox, ThreatExpert, BitBlaze, Cuckoo and Comodo Instant Malware Analysis.

Sandbox (Cuckoo Sandbox)



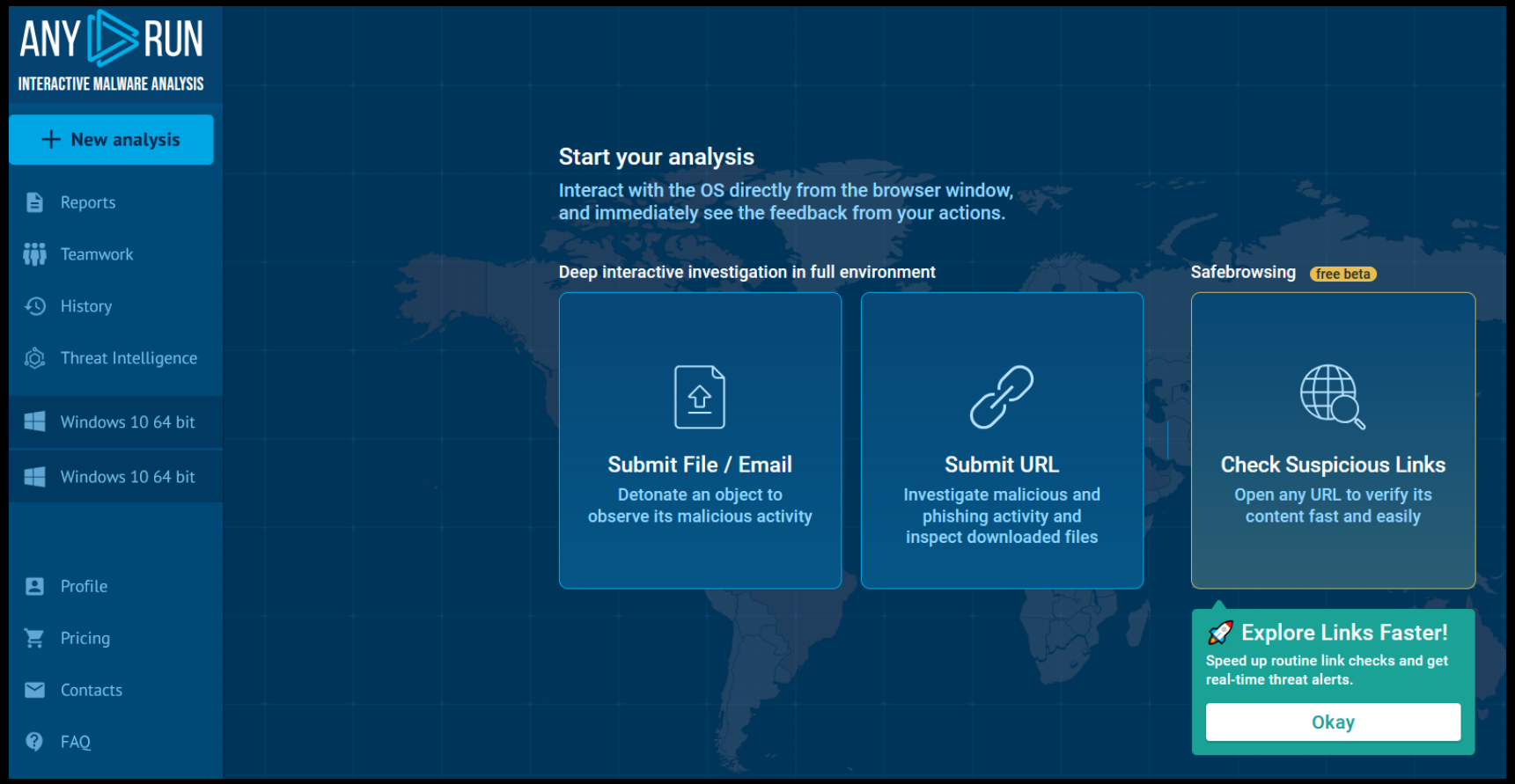
https://sandbox.pikker.ee/

Sandbox (Cuckoo Sandbox)



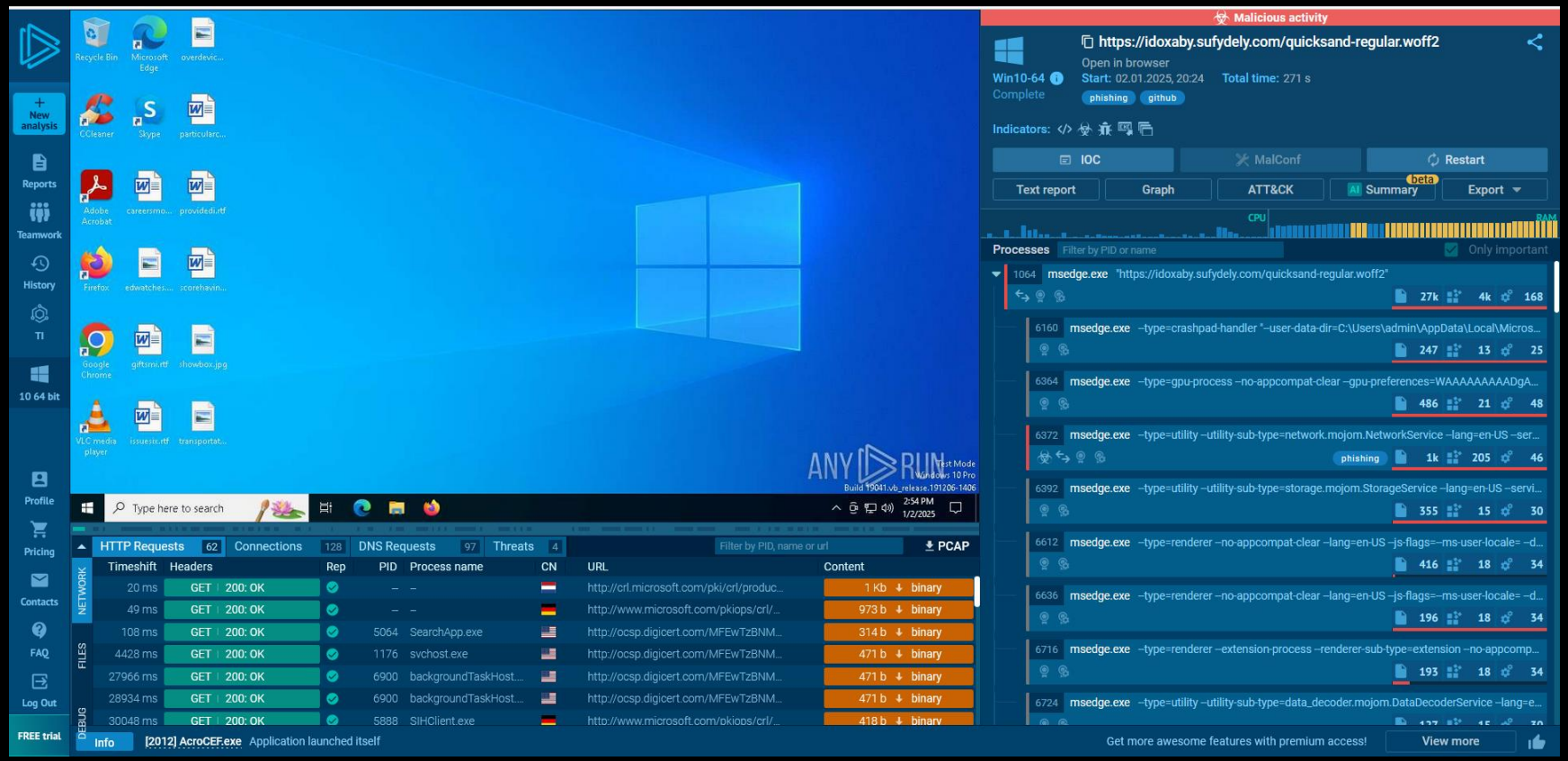
https://sandbox.pikker.ee/

Sandbox (Any Run)



https://any.run/

Sandbox (Any Run)



https://any.run/

Sandbox

Triage by Recorded Future
https://tria.ge/

Hybrid Analysis
https://www.hybrid-analysis.com/

Joe Sandbox
https://www.joesandbox.com/

Sandbox Drawbacks

- The sandbox simply runs the executable, without **command-line** options. If the malware executable requires command-line options, it will not execute any code that runs only when an option is provided.
- If malware is waiting for a **command-and-control** packet to be returned before launching a backdoor, the backdoor will not be launched in the sandbox.
- It may not record all events, because neither you nor the sandbox may wait long enough. Ex. if the malware is **set to sleep** for a day before it performs malicious activity, you may miss that event.

Sandbox Drawbacks

- Malware often detects when it is running in a virtual machine, and if a virtual machine is detected, the malware might stop running or behave differently.
- Some malware requires the presence of certain registry keys or files on the system that might not be found in the sandbox. These might be required to contain legitimate data, such as commands or encryption keys.
- If the malware is a DLL, certain exported functions will not be invoked properly, because a DLL will not run as easily as an executable.
- The sandbox environment OS may not be correct for the malware.
- A sandbox cannot tell you what the malware does.

References

- Practical Malware Analysis by Michael Sikorski