

# Mã độc

## Chương 2. Phân tích mã độc cơ bản

# Mục tiêu

- Nhắc lại một số kiến thức cơ bản cần thiết trong quá trình phân tích mã độc
- Giới thiệu các phương pháp phân tích mã độc
- Giới thiệu một số công cụ phân tích mã độc

# Tài liệu tham khảo

**[1] Michael Sikorski, Andrew Honig, 2012, Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software, No Starch Press, (ISBN: 978-1593272906).**

**[2] Sam Bowne, Slides for a college course at City College San Francisco,  
[https://samsclass.info/126/126\\_S17.shtml](https://samsclass.info/126/126_S17.shtml)**

# Nội dung

- 1. Các phương pháp phân tích mã độc**
- 2. Công cụ và kiến thức cơ sở**
- 3. Các nguyên tắc khuyến nghị trong phân tích mã độc**
- 4. Phân tích tĩnh cơ bản**
- 5. Phân tích động cơ bản**
- 6. Xây dựng môi trường phân tích mã độc**

# Nội dung

- 1. Các phương pháp phân tích mã độc**
- 2. Công cụ và kiến thức cơ sở**
- 3. Các nguyên tắc khuyến nghị trong phân tích mã độc**
- 4. Phân tích tĩnh cơ bản**
- 5. Phân tích động cơ bản**
- 6. Xây dựng môi trường phân tích mã độc**

# Mục đích của phân tích mã độc

- ☐ Liệt kê được tất cả các hành vi độc hại
- ☐ Phát hiện máy chủ điều khiển (nếu có)
- ☐ Gỡ bỏ phần mềm độc hại và khôi phục máy tính, tài liệu của người sử dụng về trạng thái như trước khi bị lây nhiễm.
- ☐ Đưa ra dấu hiệu để có thể rà soát, phát hiện phần mềm độc hại đó trên những máy tính khác.
- ☐ Đưa ra phương pháp để phòng trừ, ngăn chặn lây lan của phần mềm độc hại.

# Các phương pháp phân tích mã độc

- ❑ Phân tích tĩnh và phân tích động
- ❑ Phân tích cơ bản và phân tích nâng cao

# Các phương pháp phân tích mã độc

- ❑ Phân tích tĩnh và phân tích động
- ❑ Phân tích cơ bản và phân tích nâng cao



# Phân tích tĩnh

## Kỹ thuật phân tích tĩnh (Static Analysis)

- ❑ Tiến hành kiểm tra mã độc mà không cần phải thực thi mã độc
- ❑ Công cụ: VirusTotal, Strings, trình Disassembler như IDA Pro...



# Phân tích động

**Kỹ thuật phân tích động (Dynamic Analysis)**

**❑ Tiến hành kiểm tra mã độc bằng cách thực thi mã độc và theo dõi chúng**



# Phân tích động

**Kỹ thuật phân tích động (Dynamic Analysis)**

- ❑ Mã độc được phân tích trong môi trường máy ảo, cách ly với hệ thống thật và máy ảo được Snapshot lại các bản ghi.**
- ❑ Công cụ: RegShot, Process Monitor, Process Hacker, CaptureBAT...**

# Các phương pháp phân tích mã độc

- ❑ Phân tích tĩnh và phân tích động
- ❑ Phân tích cơ bản và phân tích nâng cao

# Phân tích tĩnh cơ bản

- ❑ Theo dõi mã độc mà không cần đi sâu vào việc phân tích mã Assembly
- ❑ Công cụ: VirusTotal, Strings...
- ❑ Với cách này thì chỉ phân tích được những mẫu mã độc đơn giản, với những mẫu phức tạp thì có thể sẽ không phát hiện được những hành vi của chúng.

# Phân tích động cơ bản

- ☐ Thực thi mã độc và theo dõi hoạt động
- ☐ Dễ dàng thực hiện nhưng đòi hỏi phải có môi trường phân tích an toàn
- ☐ Không cho kết quả tốt với tất cả các loại mã độc

# Phân tích nâng cao

**Phân tích tĩnh nâng cao:**

- ☐ Dịch ngược mã độc với các trình Disassembler
- ☐ Độ phức tạp cao, yêu cầu kiến thức về hợp ngữ - assembly

**Phân tích động nâng cao:**

- ☐ Chạy chương trình với các trình Debugger
- ☐ Kiểm tra trạng thái bên trong của một chương trình mã độc đang chạy.

# Quy trình phân tích một phần mềm mã độc

- ❑ Cài đặt và cấu hình môi trường an toàn cho việc phân tích mã độc,
- ❑ Phân tích tĩnh cơ bản,
- ❑ Phân tích động cơ bản,
- ❑ Phân tích chuyên sâu (phân tích tĩnh nâng cao và phân tích động nâng cao).



# Nội dung

1. Các phương pháp phân tích mã độc
- 2. Công cụ và kiến thức cơ sở**
3. Các nguyên tắc khuyến nghị trong phân tích mã độc
4. Phân tích tĩnh cơ bản
5. Phân tích động cơ bản
6. Xây dựng môi trường phân tích mã độc

# Công cụ và kiến thức cơ sở

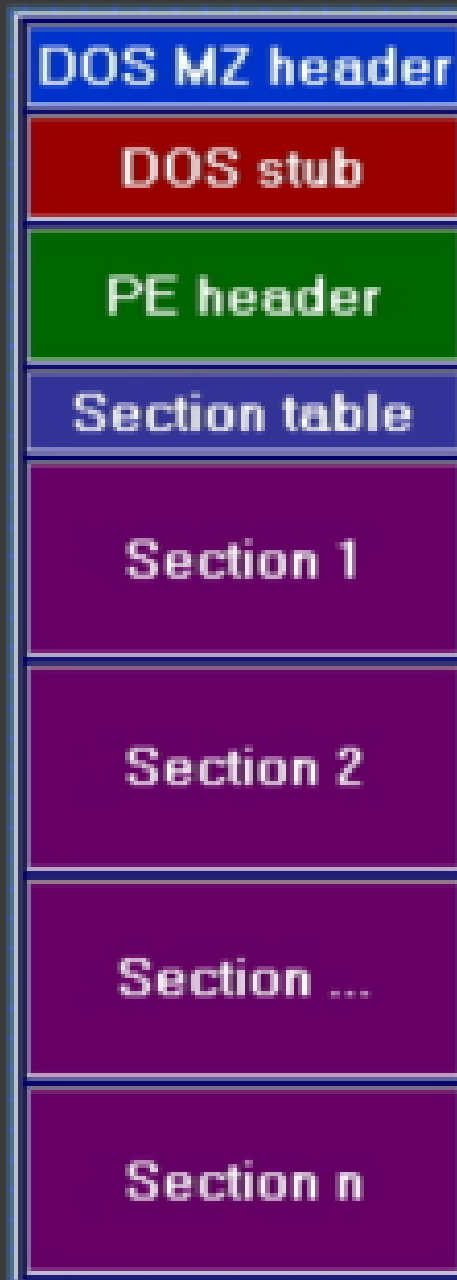
- ☐ Định dạng file thực thi trên Windows
- ☐ Liên kết các thư viện và hàm
- ☐ Môi trường phân tích mã độc

# Công cụ và kiến thức cơ sở

- ❑ Định dạng file thực thi trên Windows
- ❑ Liên kết các thư viện và hàm
- ❑ Môi trường phân tích mã độc

# PE File Format

- ❑ Là một định dạng file riêng của Win32. Hầu hết các file thực thi, Object file hay các file DLLs đều thuộc định dạng này.
- ❑ PE file là một cấu trúc dữ liệu mà Windows định nghĩa, chứa thông tin cần thiết cho windows nạp và thực thi tệp.
- ❑ Một PE file về cơ bản sẽ có các trường: DOS MZ Header, DOS Stub, PE Header, Section tables...



- Executable Code Section, named `.text` (Microsoft) or `CODE` (Borland)
- Data Sections, named `.data`, `.rdata`, or `.bss` (Microsoft) or `DATA` (Borland)
- Resources Section, named `.rsrc`
- Export Data Section, named `.edata`
- Import Data Section, named `.idata`
- Debug Information Section, named `.debug`

# PE Header

PE Header thực chất là một cấu trúc (Struct) **IMAGE\_NT\_HEADERS** bao gồm các thông tin cần cho quá trình nạp file lên bộ nhớ.

- ☐ Chứa thông tin về file
- ☐ Loại ứng dụng
- ☐ Các hàm và thư viện bắt buộc
- ☐ Không gian lưu trữ được yêu cầu.

# Important PE Sections

**Một số Section quan trọng cần chú ý:**

- ❑ .text – Vùng lưu giữ các tập chỉ thị của CPU (Mã nguồn đã được biên dịch)**
- ❑ .rdata – Các thư viện Import và Export ra các hàm xử lý**
- ❑ .data – Vùng chứa các biến toàn cục, static, hằng (Thực ra là chỉ lưu giữ địa chỉ của chúng còn giá trị thì lưu ở .rsrc)**
- ❑ .rsrc – Lưu trữ các chuỗi, icon, images, menus... trong chương trình.**

# PE View

PEview - C:\Windows\System32\notepad.exe

File View Go Help

notepad.exe

- IMAGE\_DOS\_HEADER
- MS-DOS Stub Program
- IMAGE\_NT\_HEADERS
  - Signature
  - IMAGE\_FILE\_HEADER
  - IMAGE\_OPTIONAL\_HEADER
- IMAGE\_SECTION\_HEADER .text
- IMAGE\_SECTION\_HEADER .data
- IMAGE\_SECTION\_HEADER .rsrc
- IMAGE\_SECTION\_HEADER .reloc
- BOUND\_IMPORT Directory Table
- BOUND\_IMPORT DLL Names
- SECTION .text
- SECTION .data
- SECTION .rsrc
- SECTION .reloc

Data	Description	Value
014C	Machine	IMAGE_FILE_MACHINE_I386
0004	Number of Sections	
4A5BC60F	Time Date Stamp	2009/07/13 Mon 23:41:03 UTC
00000000	Pointer to Symbol Table	
00000000	Number of Symbols	
00E0	Size of Optional Header	
0102	Characteristics	
		IMAGE_FILE_EXECUTABLE_IMAGE
		IMAGE_FILE_32BIT_MACHINE

Viewing IMAGE\_FILE\_HEADER



# IMAGE\_SECTION\_HEADER

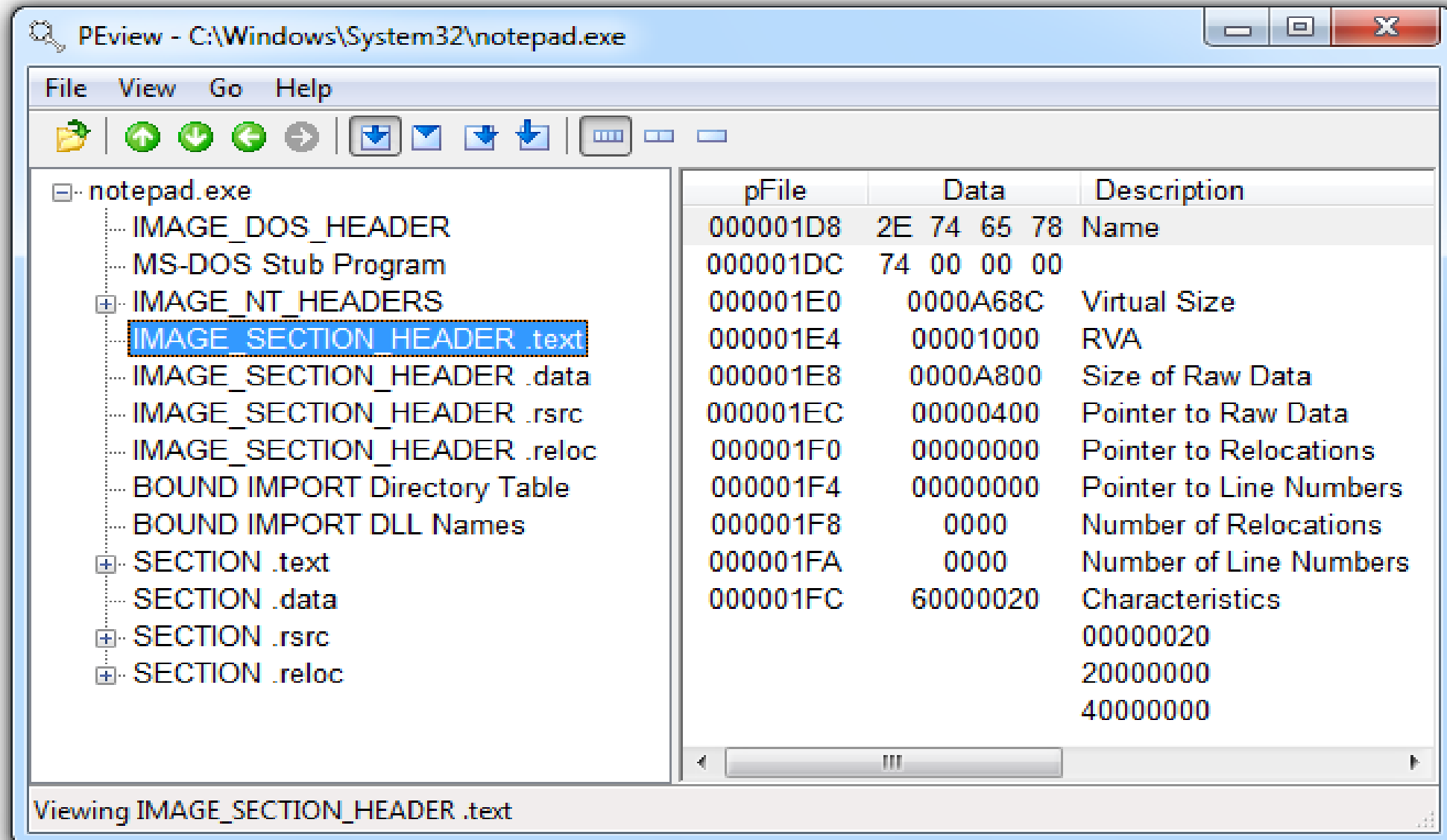
**Những thông tin cần chú ý của các Section như:**

- ❑ Virtual Size: Kích thước của file trên bộ nhớ RAM**
- ❑ Size of Raw Data: Kích thước của file trên ổ đĩa cứng**

# IMAGE\_SECTION\_HEADER

- ❑ Với .text Section thông thường kích thước không thay đổi nhiều
- ❑ Các tệp tin thực thi bị Packed sẽ cho *kích thước của Virtual Size lớn hơn nhiều lần so với kích thước của Size of Raw Data*. Đây cũng là dấu hiệu của mã độc.
- ❑ Các mã độc thường bị Packed lại để gây khó khăn cho người phân tích.

# Not Packed



# Packed

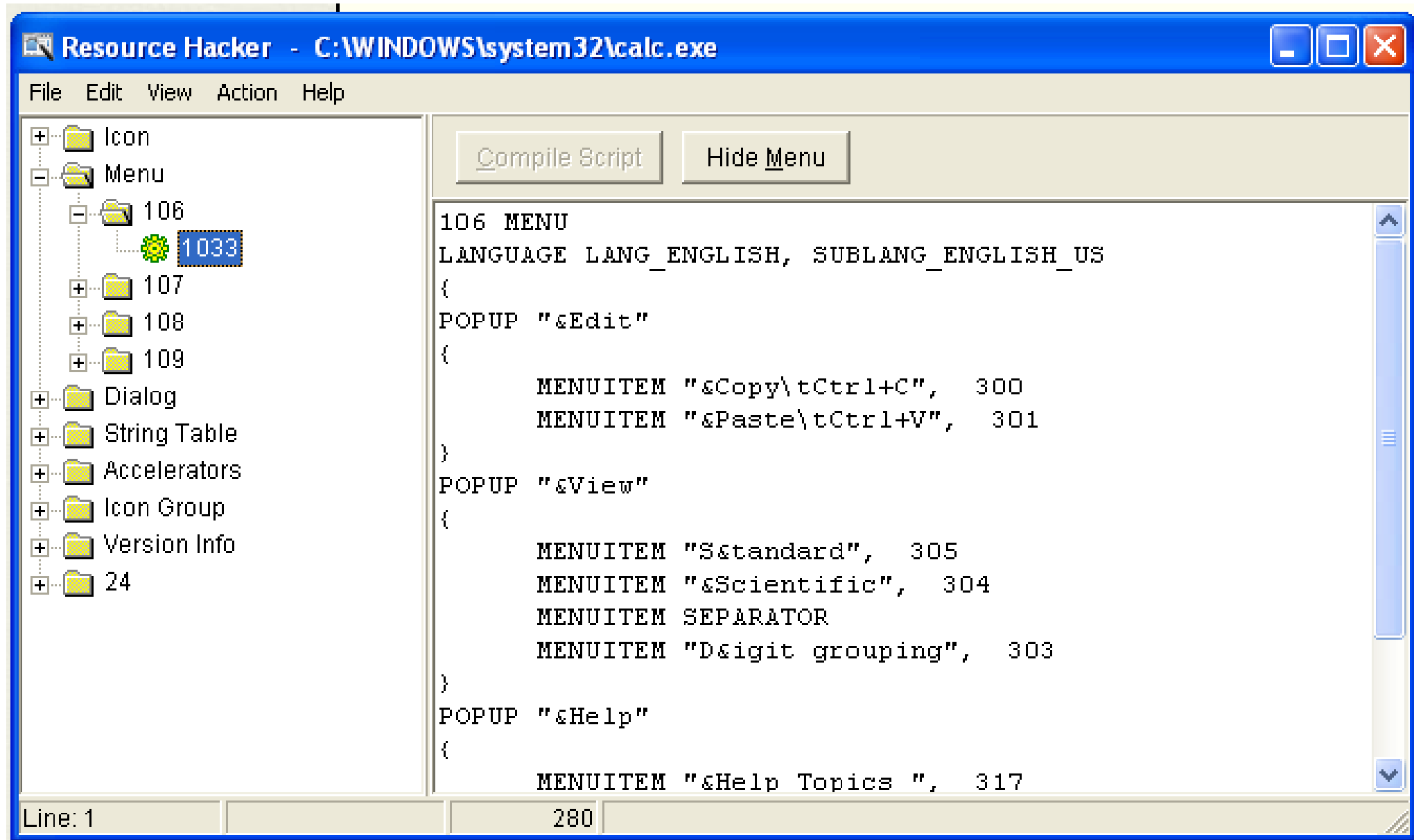
## *Section Information for PackedProgram.exe*

Name	Virtual size	Size of raw data
.text	A000	0000
.data	3000	0000
.rdata	4000	0000
.rsrc	19000	3400
Dijfpds	20000	0000
.sdfuok	34000	3313F
Kijijl	1000	0200

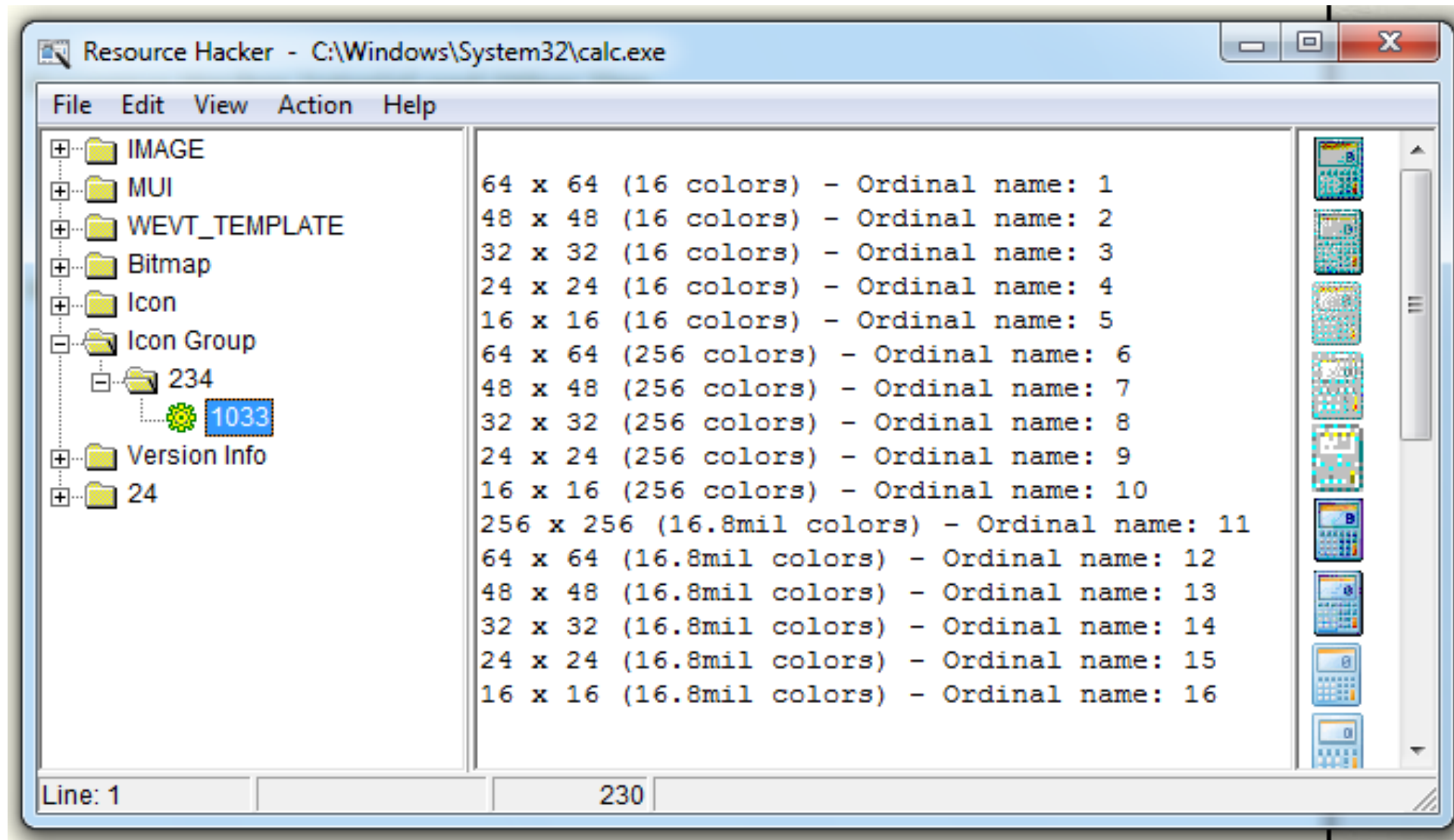
# Resource Hacker

- ❑ Là một chương trình cho phép duyệt qua phần .rsrc section
- ❑ Thông qua đó có thể chỉnh sửa, thêm, xóa các tài nguyên như: Strings, icons, menus... (VD: Mod Game)

# Resource Hacker in Windows XP



# Resource Hacker in Windows 7



# Công cụ và kiến thức cơ sở

- ❑ Định dạng file thực thi trên Windows
- ❑ Liên kết các thư viện và hàm
- ❑ Môi trường phân tích mã độc



# Liên kết các thư viện và hàm

- ❑ Các hàm được sử dụng trong chương trình được lưu trữ bởi một chương trình khác: chẳng hạn như các thư viện
- ❑ Windows có nhiều thư viện hỗ trợ lập trình viên. Các thư viện (DLL) chứa các hàm/API được viết sẵn.

# Liên kết các thư viện và hàm

❑ Khi biên dịch chương trình từ mã nguồn (C/C++) thu được các *Object file*. Cần phải có một *trình liên kết* (Linker) để liên kết các file Obj đó thành *file thực thi .exe*

❑ Các Linker có thể liên kết theo ba cách:

- **Statically**
- **At Runtime**
- **Dynamically**

# Static Linking

- ❑ Hiếm khi được sử dụng cho các file thực thi trên Windows, phổ biến trên nền tảng Unix/Linux.
- ❑ Tất cả các hàm trong thư viện đều được thêm vào file thực thi.
- ❑ Kích thước của file chạy sẽ lớn.

# Runtime Linking

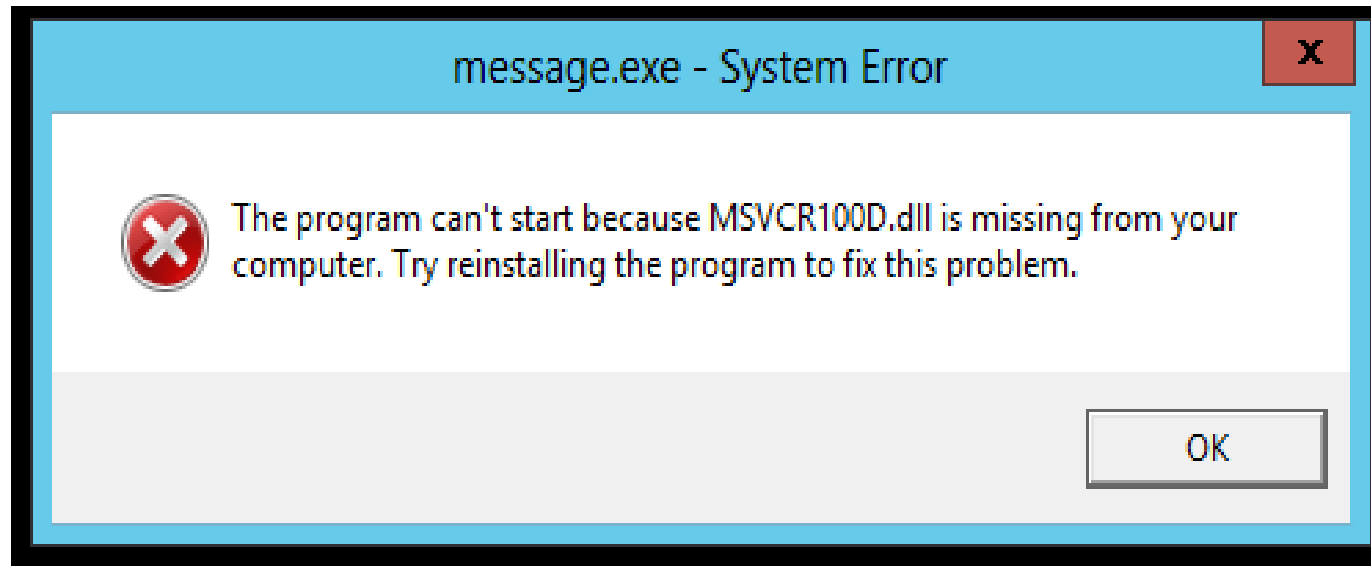
- ❑ Không phổ biến ở các chương trình/phần mềm thông thường.
- ❑ Thường gặp trong các phần mềm độc hại. Đặc biệt là với các chương trình độc hại bị packed (nén) và Obfuscate (làm rối).

# Runtime Linking

- ❑ Các chương trình sử dụng liên kết kiểu này thì khi phân tích sẽ không phát hiện được nhiều thư viện mà nó đã Import. Thay vào đó trong quá trình chạy thì nó mới import những thư viện cần thiết.
- ❑ Các hàm thường gặp: `LoadLibrary` và `GetProcAddress`.

# Dynamic Linking

- ❑ Là phương pháp phổ biến nhất.
- ❑ Các thư viện cần thiết sẽ được tìm kiếm khi chương trình được nạp.



# Công cụ và kiến thức cơ sở

- ❑ Định dạng file thực thi trên Windows
- ❑ Liên kết các thư viện và hàm
- ❑ Môi trường phân tích mã độc

# Môi trường phân tích mã độc

- ❑ Máy thật
- ❑ Máy ảo



# Máy thật

**Ưu điểm:**

☐ Một số mã độc phát hiện môi trường máy ảo và không chạy trên môi trường này, gây khó khăn trong phân tích.

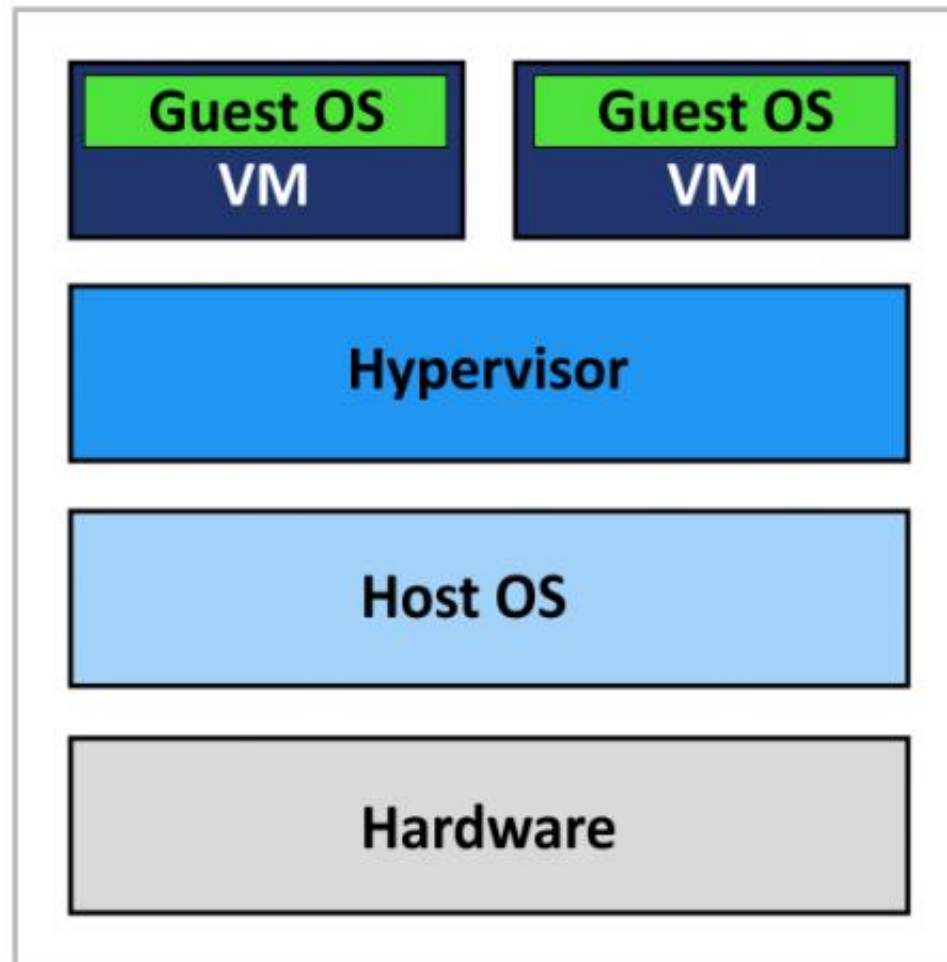
**Nhược điểm:**

☐ Một số loại mã độc có thể không hoạt động khi không có kết nối Internet

☐ Khó khăn trong việc loại bỏ hoàn toàn mã độc trên máy thật, do đó cần phải có những phương án dự phòng.

# Máy ảo

- ❑ Phương pháp phổ biến nhất
- ❑ Bảo vệ máy thật từ các phần mềm độc hại

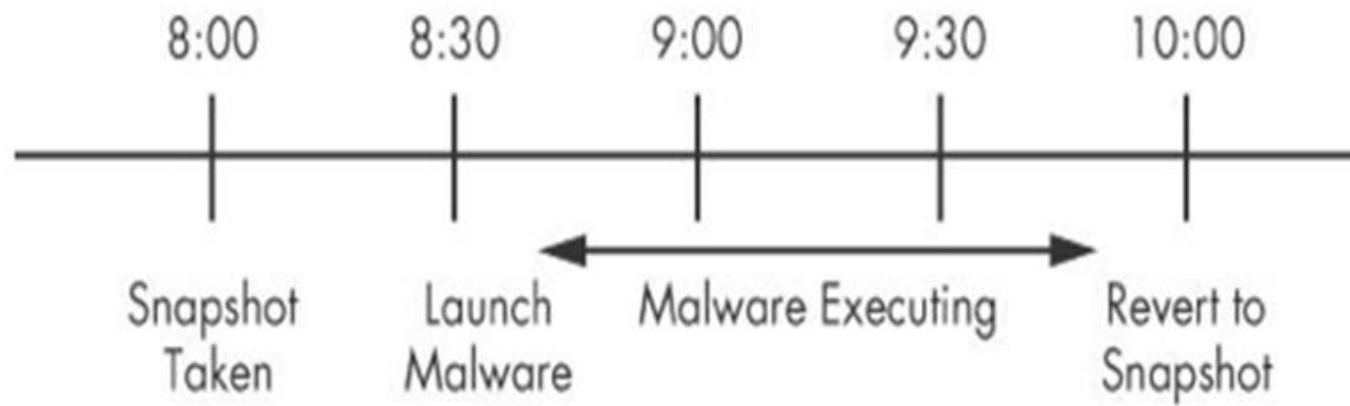


# Vmware Player

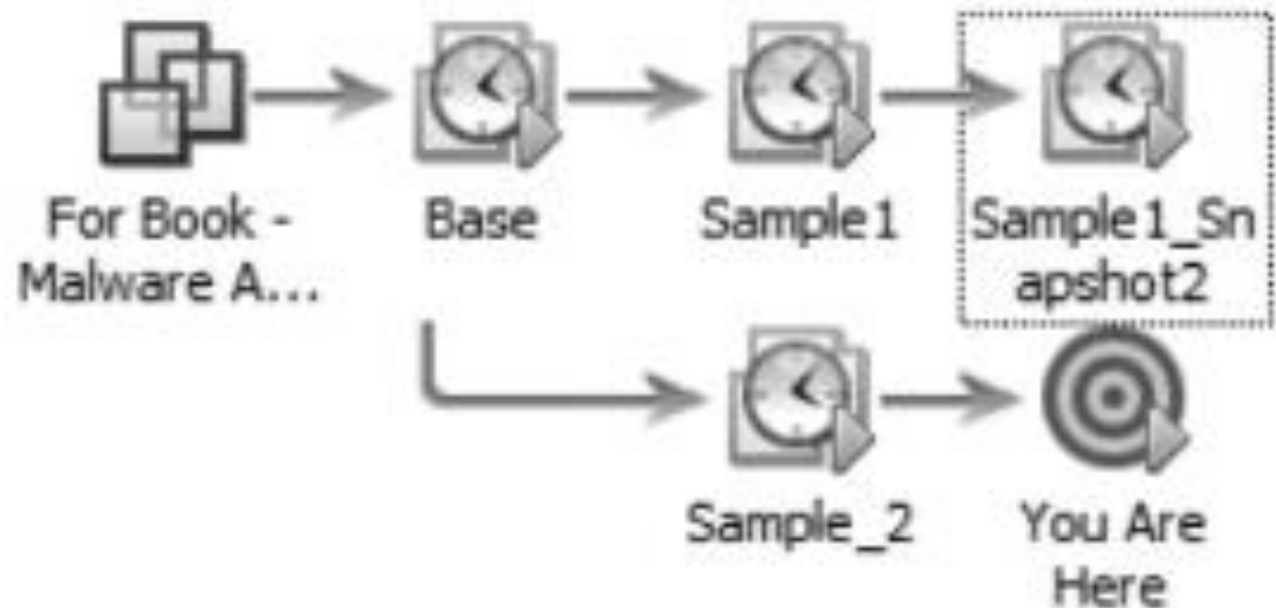
- ❑ Một phần mềm tạo máy ảo phổ biến
- ❑ Miễn phí nhưng giới hạn về chức năng so với bản thương mại.
- ❑ Một số phần mềm tạo máy ảo khác như: VirtualBox, Hyper-V, Xen, Parallels...



# Snapshots



*Snapshot timeline*



# Rủi ro của việc sử dụng VMware

- ❑ Mã độc có thể phát hiện được rằng nó đang nằm trong máy ảo và thay đổi hành động của nó.
- ❑ Phần mềm VMware có một số lỗ hổng bảo mật, mã độc có thể khai thác.
- ❑ Mã độc có thể lây nhiễm sang máy thật. Không nên lưu những dữ liệu nhạy cảm trên máy thật.

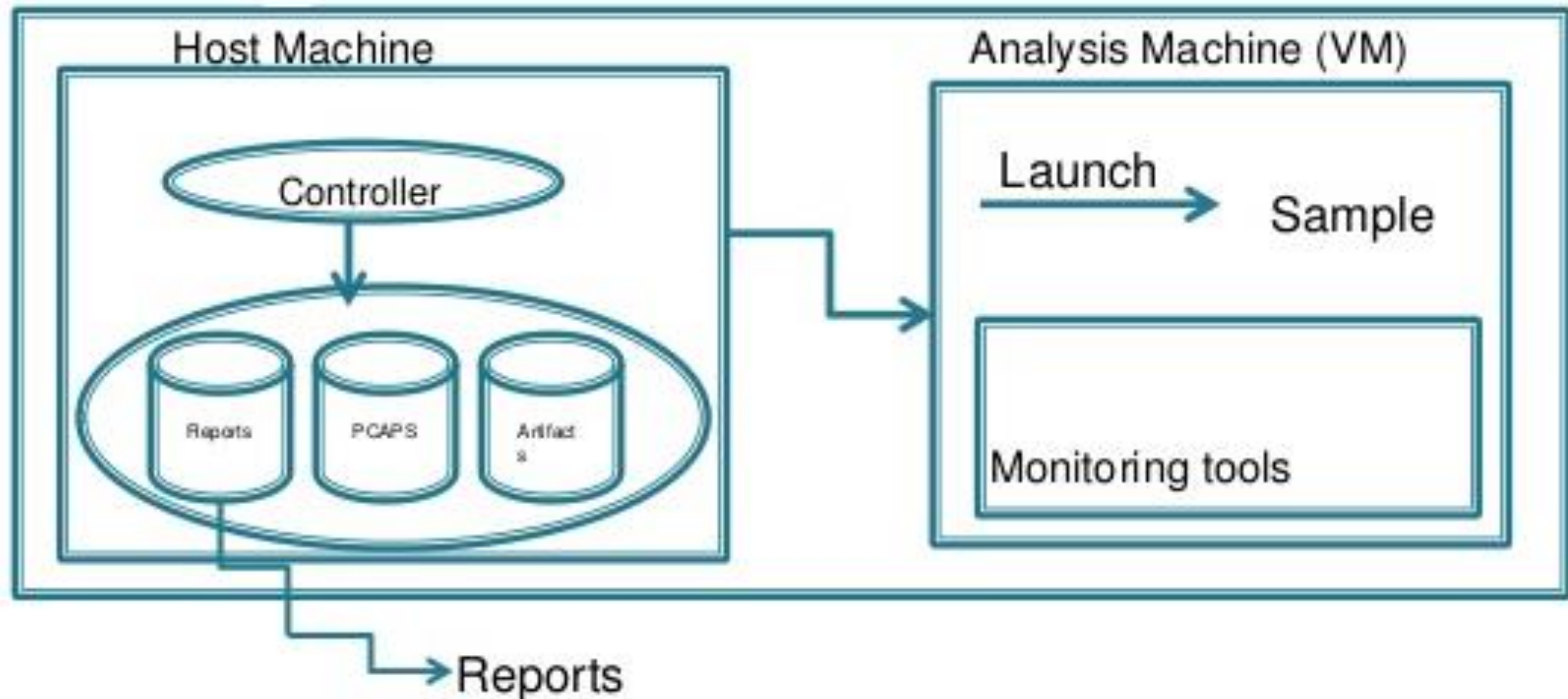
# Sandbox

- ❑ Môi trường phân tích mã độc, tích hợp nhiều công cụ phục vụ phân tích động cơ bản.
- ❑ Môi trường ảo hóa mô phỏng các dịch vụ mạng
- ❑ Hỗ trợ rất tốt trong việc lập báo cáo, thống kê...

**VD: CW Sandbox, GFI Sandbox, Anubis, Joe Sandbox, Comodo Instant Malware Analysis**

# Sandbox

## □ Sandbox architect



# Sandbox

## ❑ Comodo sandbox

### Verdict

#### Auto Analysis Verdict

Suspicious++

### Description

#### Suspicious Actions Detected

Copies self to other locations

Creates autorun records

Creates files in windows system directory

Deletes self

Injects code into other processes

### Mutexes Created or Opened

PId	Image Name	Address	Mutex Name
0x41c	C:\TEST\sample.exe	0x404b9b	L1T7X2S XK4V2RL
0x41c	C:\TEST\sample.exe	0x404b9b	L1T7X2S XK4V2RLUser15
0x41c	C:\TEST\sample.exe	0x404b9b	User5
0x7ec	C:\WINDOWS\system32\taskmgr.exe	0x404b9b	L1T7X2S XK4V2RL
0x7ec	C:\WINDOWS\system32\taskmgr.exe	0x404b9b	User5



# Sandbox

## ❑ CW sandbox

Scan Summary

File Changes

Registry Changes

▶ Submission Details

Date	25.02.2013 02:27:47
Sandbox Version	2.1.22
File Name	c:\servernoanti.exe
Submitting Email	
Comment	

▶ Summary Findings

Total Number of Processes	5
Termination Reason	NormalTermination
Start Time	00:00.110
Stop Time	00:16.860
Start Reason	AnalysisTarget

▶ Analysis HighLights

Spawned Processes	Found 4 Processes. (View Activity by Process)
Filesystem Changes	View File Changes
Registry Changes	View Registry Changes
Network Activity	View Network Activity

# Nội dung

1. Các phương pháp phân tích mã độc
2. Công cụ và kiến thức cơ sở
3. Các nguyên tắc khuyến nghị trong phân tích mã độc
4. Phân tích tĩnh cơ bản
5. Phân tích động cơ bản
6. Xây dựng môi trường phân tích mã độc

# Các nguyên tắc khuyến nghị

- ☐ Không tập trung vào chi tiết.
- ☐ Thử nhiều công cụ.
- ☐ Mã độc luôn được biến đổi, cải tiến.

# Nội dung

1. Các phương pháp phân tích mã độc
2. Công cụ và kiến thức cơ sở
3. Các nguyên tắc khuyến nghị trong phân tích mã độc
- 4. Phân tích tĩnh cơ bản**
5. Phân tích động cơ bản
6. Xây dựng môi trường phân tích mã độc

# Phân tích tĩnh cơ bản

- ☐ Antivirus scanning
- ☐ Tìm kiếm theo giá trị Hash
- ☐ Tìm kiếm theo chuỗi ký tự
- ☐ Xác định các thư viện và hàm được dùng

# Phân tích tĩnh cơ bản

## ☐ Antivirus scanning

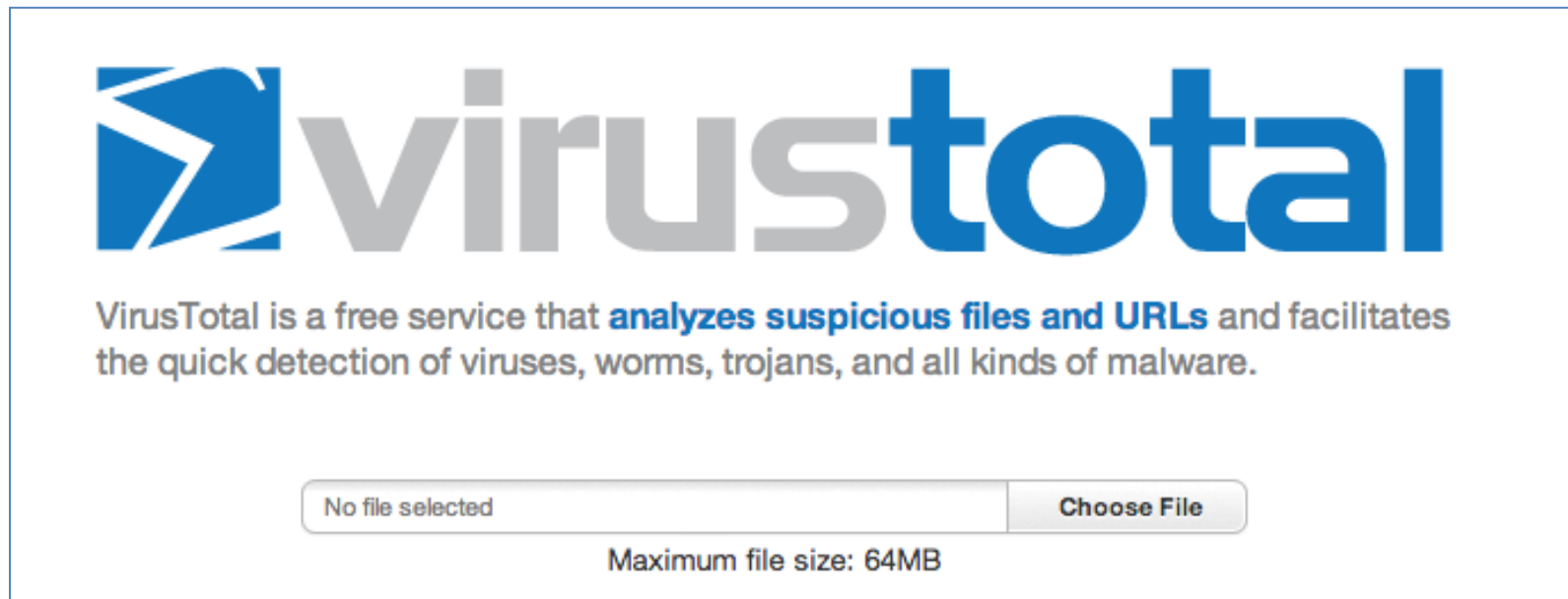
☐ Tìm kiếm theo giá trị Hash

☐ Tìm kiếm theo chuỗi ký tự

☐ Xác định các thư viện và hàm được dùng

# Antivirus scanning

- ❑ Là một trong những bước đầu của quá trình phân tích.
- ❑ Mã độc dễ dàng thay đổi chữ ký và có thể qua mặt được phần mềm chống virus.
- ❑ Một số công cụ Online: VirusTotal, Malwr...



# Phân tích tĩnh cơ bản

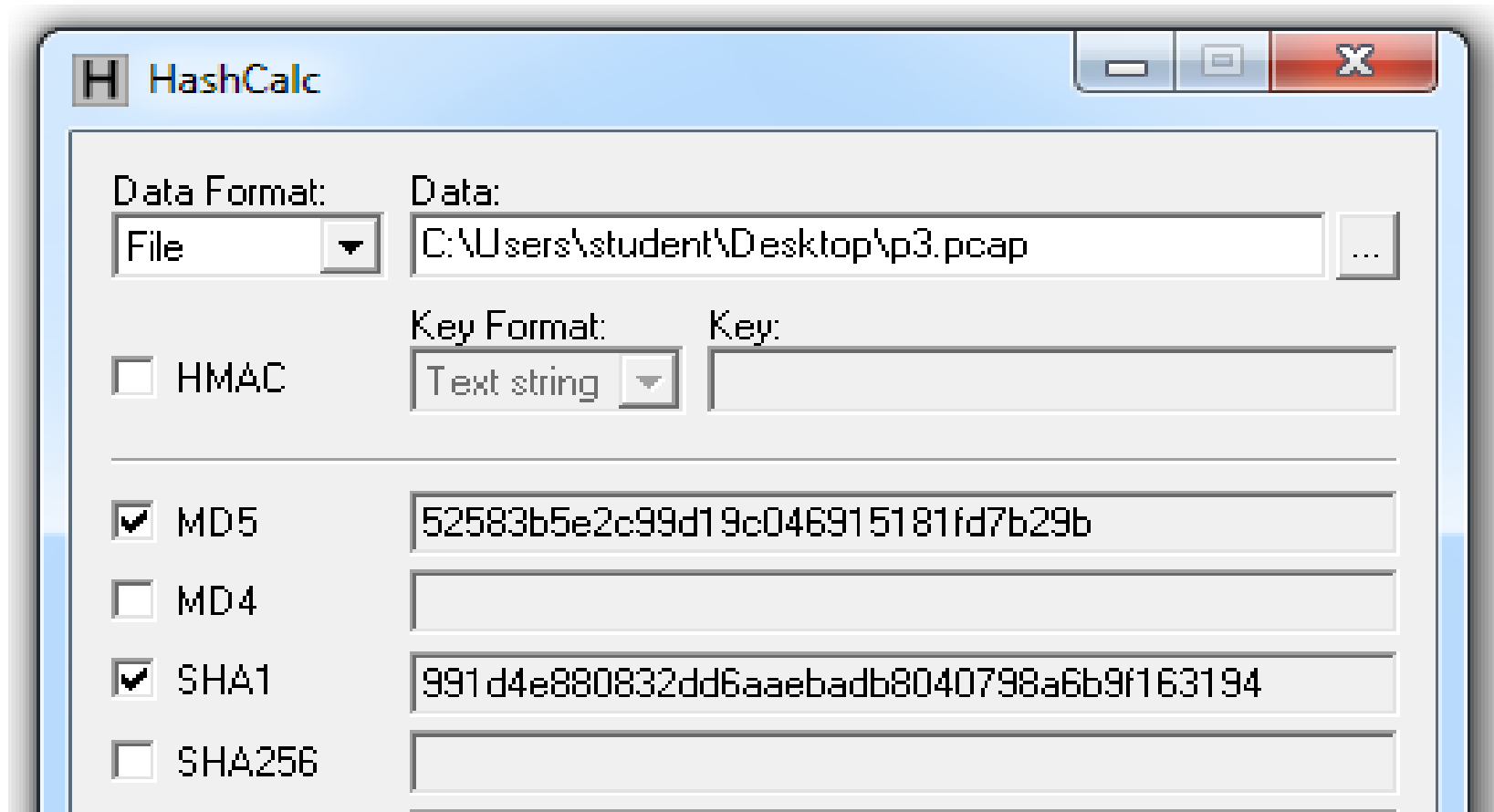
- ❑ Antivirus scanning
- ❑ Tìm kiếm theo giá trị Hash
- ❑ Tìm kiếm theo chuỗi ký tự
- ❑ Xác định các thư viện và hàm được dùng



# Tìm kiếm theo giá trị Hash

- ❑ Thuật toán băm thông dụng: MD5, SHA-1...
- ❑ Nếu tệp tin có giá trị hàm băm trùng với giá trị hàm băm của mẫu mã độc đã có thì có thể kết luận tệp tin là mã độc.

# HashCalc



# Phân tích tĩnh cơ bản

- ❑ Antivirus scanning
- ❑ Tìm kiếm theo giá trị Hash
- ❑ **Tìm kiếm theo chuỗi ký tự**
- ❑ Xác định các thư viện và hàm được dùng

# Tìm kiếm chuỗi ký tự

❑ Tìm kiếm chuỗi ký tự (String) trong file giúp người phân tích có thể biết được những thư viện, hàm, thông báo... có trong chương trình.

VD: Khi Finding String của một mẫu mã độc có thể xác định được địa chỉ IP của FTP Server mà mã độc kết nối tới, hoặc hành động ghi thêm một registry.

# The Strings command

❑ GetLayout và SetLayout

❑ GDI32.DLL

❑ 99.124.22.1

```
C:>strings bp6.ex_
```

```
VP3
```

```
VW3
```

```
t$@
```

```
D$4
```

```
99.124.22.1 4
```

```
e-@
```

```
GetLayout 1
```

```
GDI32.DLL 3
```

```
SetLayout 2
```

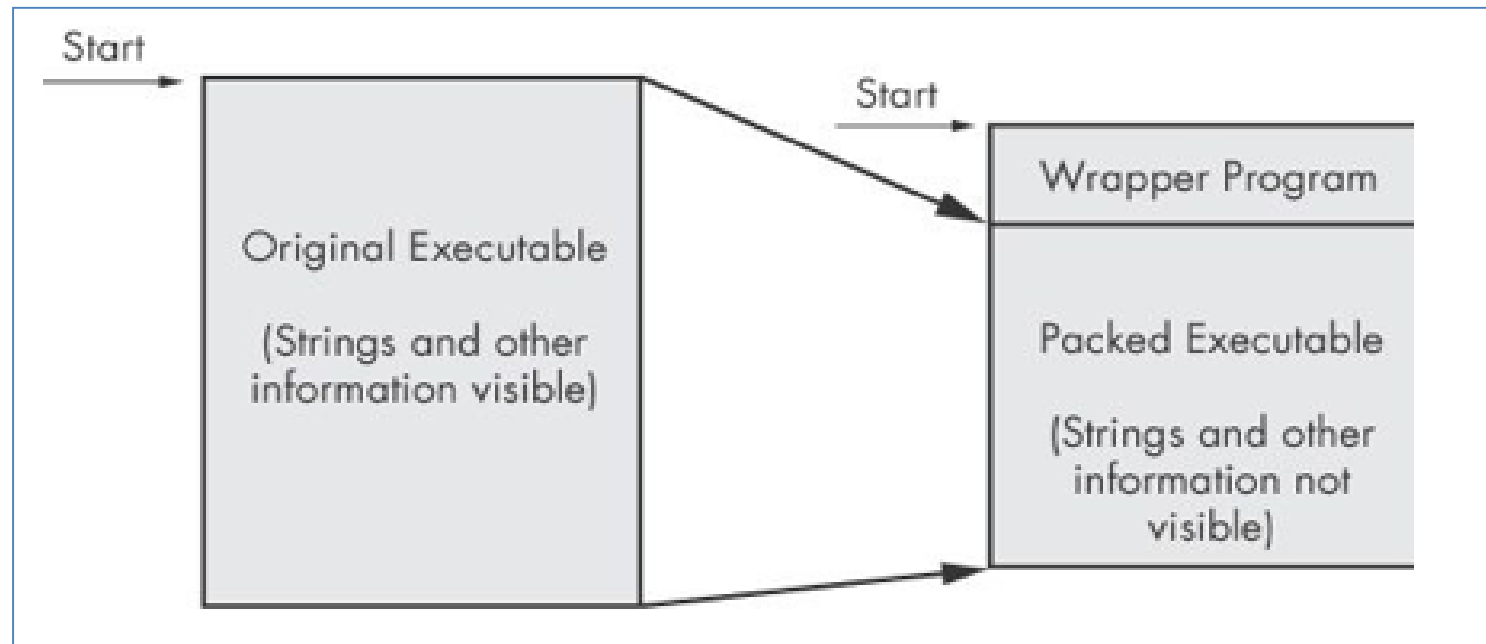
```
M}C
```

```
Mail system DLL is invalid.!Send Mail failed to  
send message. 5
```

**Khi mã độc bị đóng gói/ nén  
hoặc làm rồi?**

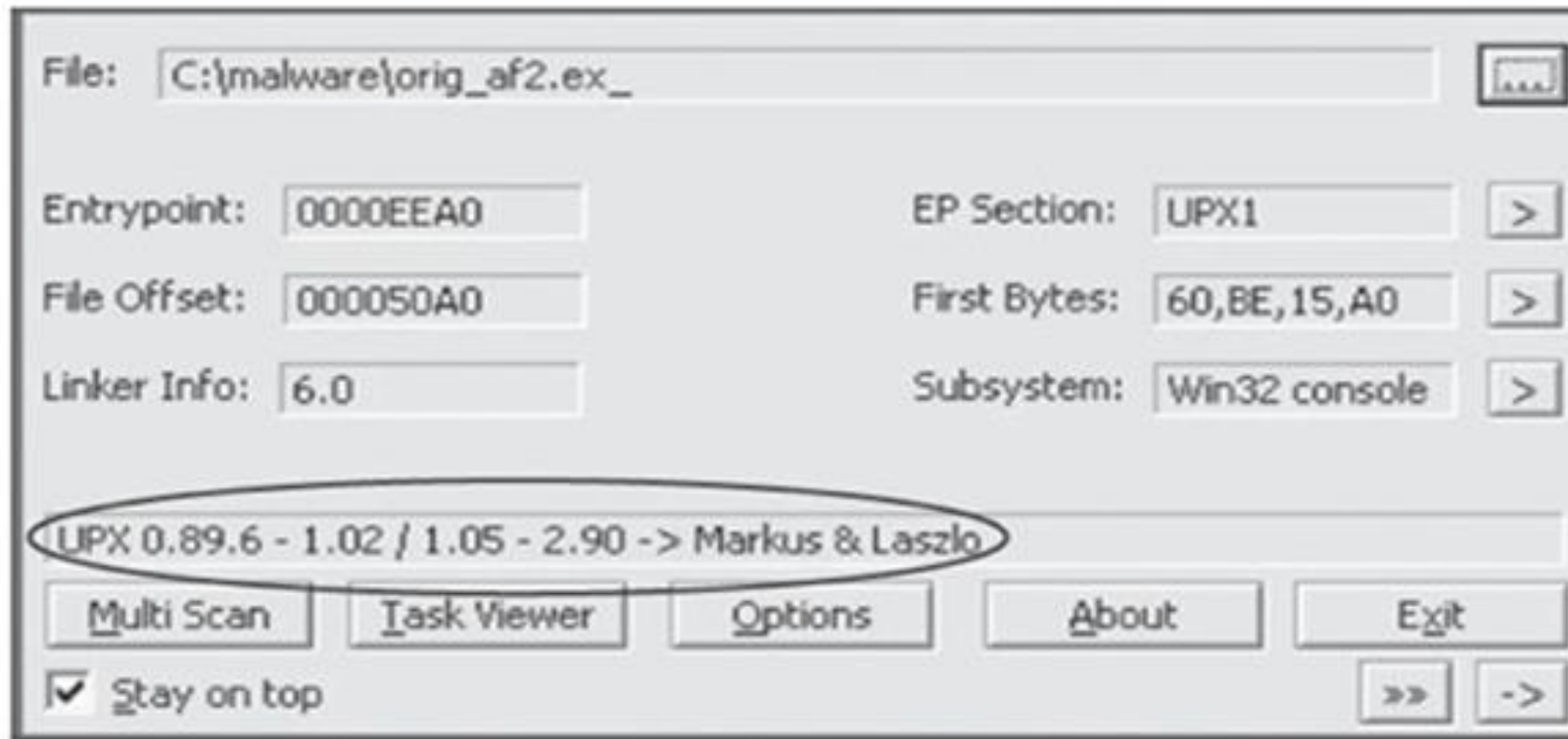
# Packing Files

- ❑ Làm khó quá trình Finding Strings và không thể đọc được mã dịch ngược của chương trình.
- ❑ Những gì nhìn thấy chỉ là một Wrapper – Một đoạn mã nhỏ dùng để giải nén khi tệp tin được chạy.



# Phát hiện Packers

- ❑ Một số công cụ giúp phát hiện chương trình bị Packed: PEiD, Exeinfo PE...



*The PEiD program*



# Phân tích tĩnh cơ bản

- ❑ Antivirus scanning
- ❑ Tìm kiếm theo giá trị Hash
- ❑ Tìm kiếm theo chuỗi ký tự
- ❑ Xác định các thư viện và hàm được dùng

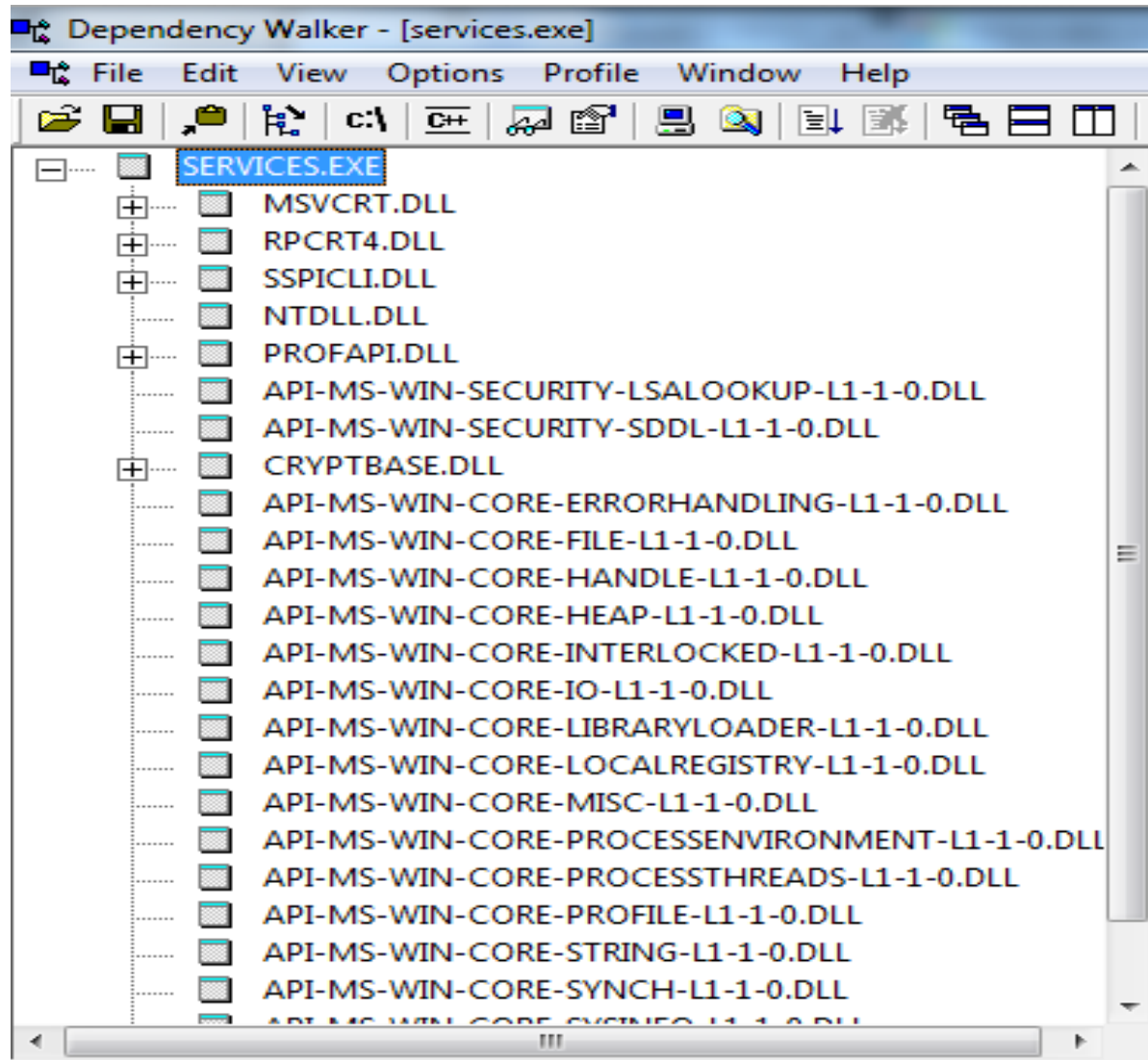
# Xác định các thư viện và hàm được dùng

- ❑ Trong PE Header liệt kê các thư viện và hàm mà chương trình sử dụng.
- ❑ Dựa vào tên và các hàm được gọi có thể phỏng đoán được chức năng của chương trình.
- ❑ VD: Hàm **URLDownloadToFile** chỉ ra rằng chương trình tải một cái gì đó.

# Dependency Walker

- ❑ Công cụ giúp hiển thị các hàm trong **Dynamic Linked**.
- ❑ Một chương trình bình thường sẽ có rất nhiều các DLL được nạp.
- ❑ Phần mềm độc hại/ mã độc thì thường có rất ít các DLL được nạp, nó sẽ nạp trong quá trình thực thi.

# Dependency Walker



**Services.exe (Normal)**



**Services.ex\_ (Malware) 68**

# Imports & Exports in Dependency Walker

**Imports (PI)**

**Exports (E)**

The screenshot shows the Dependency Walker window for LAB01-01.exe. The left pane shows the file structure with LAB01-01.EXE, KERNEL32.DLL, and MSVCRT.DLL. The right pane is split into two sections: Imports (PI) and Exports (E). The Imports section lists functions imported from MSVCRT.DLL, including malloc, exit, \_exit, \_XcptFilter, \_\_p\_\_initenv, \_\_getmainarg, \_\_initterm, and \_\_setusermath. The Exports section lists functions exported by MSVCRT.DLL, including \_XcptFilter, \_\_getmainarg, \_\_p\_\_initenv, \_\_p\_\_commoc, \_\_p\_\_fmode, \_\_set\_app\_typ, and \_\_setusermath.

PI^	Ordinal	Hint	Function
	N/A	657 (0x0291)	malloc
	N/A	585 (0x0249)	exit
	N/A	211 (0x00D3)	_exit
	N/A	72 (0x0048)	_XcptFilter
	N/A	100 (0x0064)	__p__initenv
	N/A	88 (0x0058)	__getmainarg
	N/A	271 (0x010F)	__initterm
	N/A	121 (0x0082)	__setusermath

E^	Ordinal	Hint	Function
	108 (0x006C)	106 (0x006A)	_XcptFilter
	147 (0x0093)	145 (0x0091)	__getmainarg
	181 (0x00B5)	179 (0x00B3)	__p__initenv
	187 (0x00BB)	185 (0x00B9)	__p__commoc
	192 (0x00C0)	190 (0x00BE)	__p__fmode
	212 (0x00D4)	210 (0x00D2)	__set_app_typ
	214 (0x00D6)	212 (0x00D4)	__setusermath

# Các DLL phổ biến

DLL	Description
<i>Kernel32.dll</i>	This is a very common DLL that contains core functionality, such as access and manipulation of memory, files, and hardware.
<i>Advapi32.dll</i>	This DLL provides access to advanced core Windows components such as the Service Manager and Registry.
<i>User32.dll</i>	This DLL contains all the user-interface components, such as buttons, scroll bars, and components for controlling and responding to user actions.
<i>Gdi32.dll</i>	This DLL contains functions for displaying and manipulating graphics.

# Các DLL phổ biến

<i>Ntdll.dll</i>	This DLL is the interface to the Windows kernel. Executables generally do not import this file directly, although it is always imported indirectly by <i>Kernel32.dll</i> . If an executable imports this file, it means that the author intended to use functionality not normally available to Windows programs. Some tasks, such as hiding functionality or manipulating processes, will use this interface.
<i>WSock32.dll</i> and <i>Ws2_32.dll</i>	These are networking DLLs. A program that accesses either of these most likely connects to a network or performs network-related tasks.
<i>Wininet.dll</i>	This DLL contains higher-level networking functions that implement protocols such as FTP, HTTP, and NTP.



# Imports & Exports

- ❑ Các file thư viện DLL sẽ Export ra các hàm.
- ❑ Các file thực thi \*.EXE sẽ Import các hàm vào từ thư viện.
- ❑ Các hàm, thư viện được Import hay Export đều được liệt kê trong PE Header của file.



# Ví dụ: Keylogger

❑ Imports **User32.dll** và sử dụng hàm **SetWindowsHookEx**. Nó là một hàm rất phổ biến trong các keylogger để nhận đầu vào từ bàn phím.

❑ Nó Exports **LowLevelKeyboardProc** và **LowLevelMouseProc** để gửi dữ liệu đi nơi khác.

❑ Nó sử dụng **RegisterHotKey** để xác định một hành động gõ phím đặc biệt như: Ctrl + Shift + P để thu thập dữ liệu.

# Ví dụ: Chương trình bị Packed

- ❑ Rất ít các hàm, thư viện được gọi
- ❑ Tất cả những gì thấy được chỉ là những hàm dùng cho việc giải nén/unpacker

*DLLs and Functions Imported from  
PackedProgram.exe*

Kernel32.dll	User32.dll
GetModuleHandleA	MessageBoxA
LoadLibraryA	
GetProcAddress	
ExitProcess	
VirtualAlloc	
VirtualFree	

# Nội dung

1. Các phương pháp phân tích mã độc
2. Công cụ và kiến thức cơ sở
3. Các nguyên tắc khuyến nghị trong phân tích mã độc
4. Phân tích tĩnh cơ bản
5. Phân tích động cơ bản
6. Xây dựng môi trường phân tích mã độc

# Phân tích động cơ bản

**Một số lưu ý khi tiến hành phân tích động:**

- ☐ **Thực thi/ chạy phần mềm độc hại đồng thời theo dõi kết quả,**
- ☐ **Yêu cầu cần có một môi trường an toàn cho việc phân tích,**
- ☐ **Cách li với môi trường máy thật, tránh lây nhiễm sang các máy khác,**
- ☐ **Máy thật có thể ngắt kết internet hoặc kết nối với các máy khác.**

# Phân tích động

- ❑ **Giám sát hệ thống và theo dõi các tiến trình độc hại đồng thời theo dõi kết quả**
- ❑ **So sánh Registry**

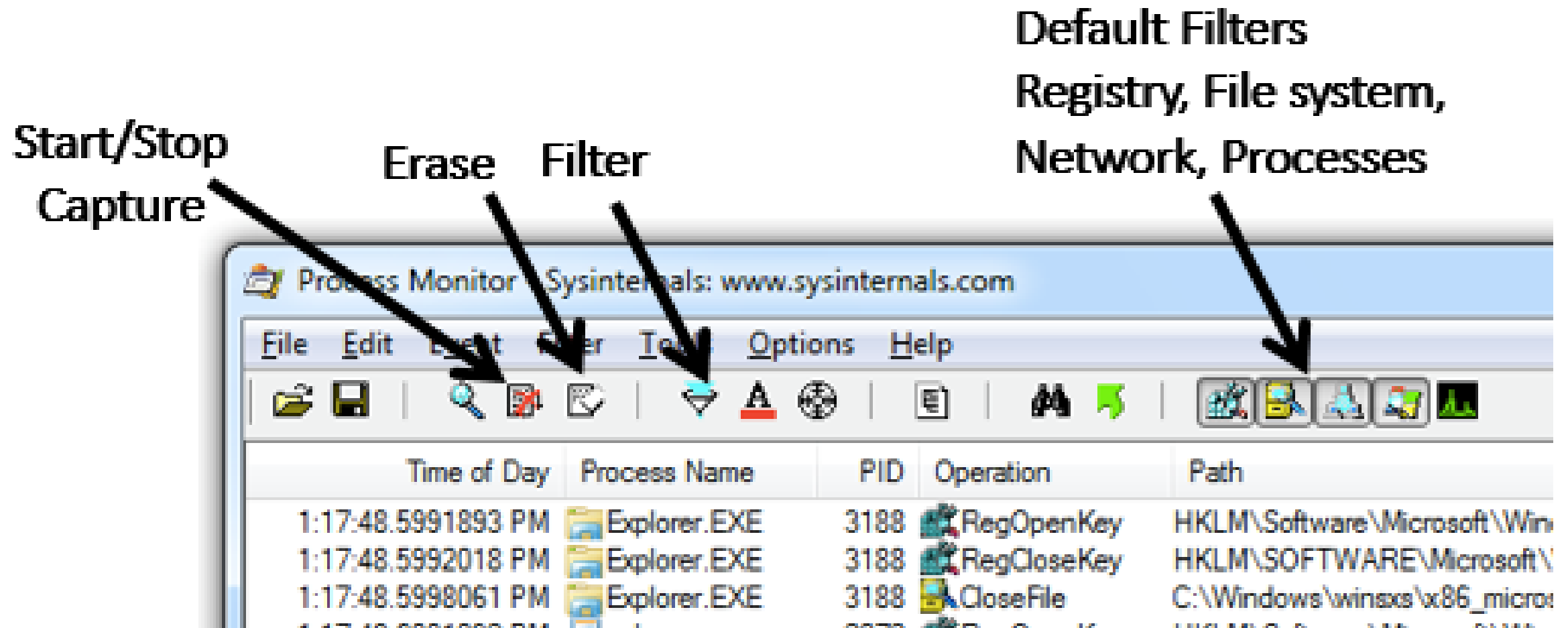
# Phân tích động

- ❑ **Giám sát hệ thống và theo dõi các tiến trình độc hại đồng thời theo dõi kết quả**
- ❑ **So sánh Registry**

# Process Monitor

- ☐ Theo dõi Registry, tệp tin hệ thống, mạng, tiến trình và các hoạt động của luồng...
- ☐ Ghi lại tất cả các sự kiện, dễ dàng tìm kiếm và lọc kết quả.
- ☐ Khi chạy lâu sẽ chiếm dụng RAM càng nhiều, dẫn đến treo hoặc tắt máy.

# Process Monitor Toolbar



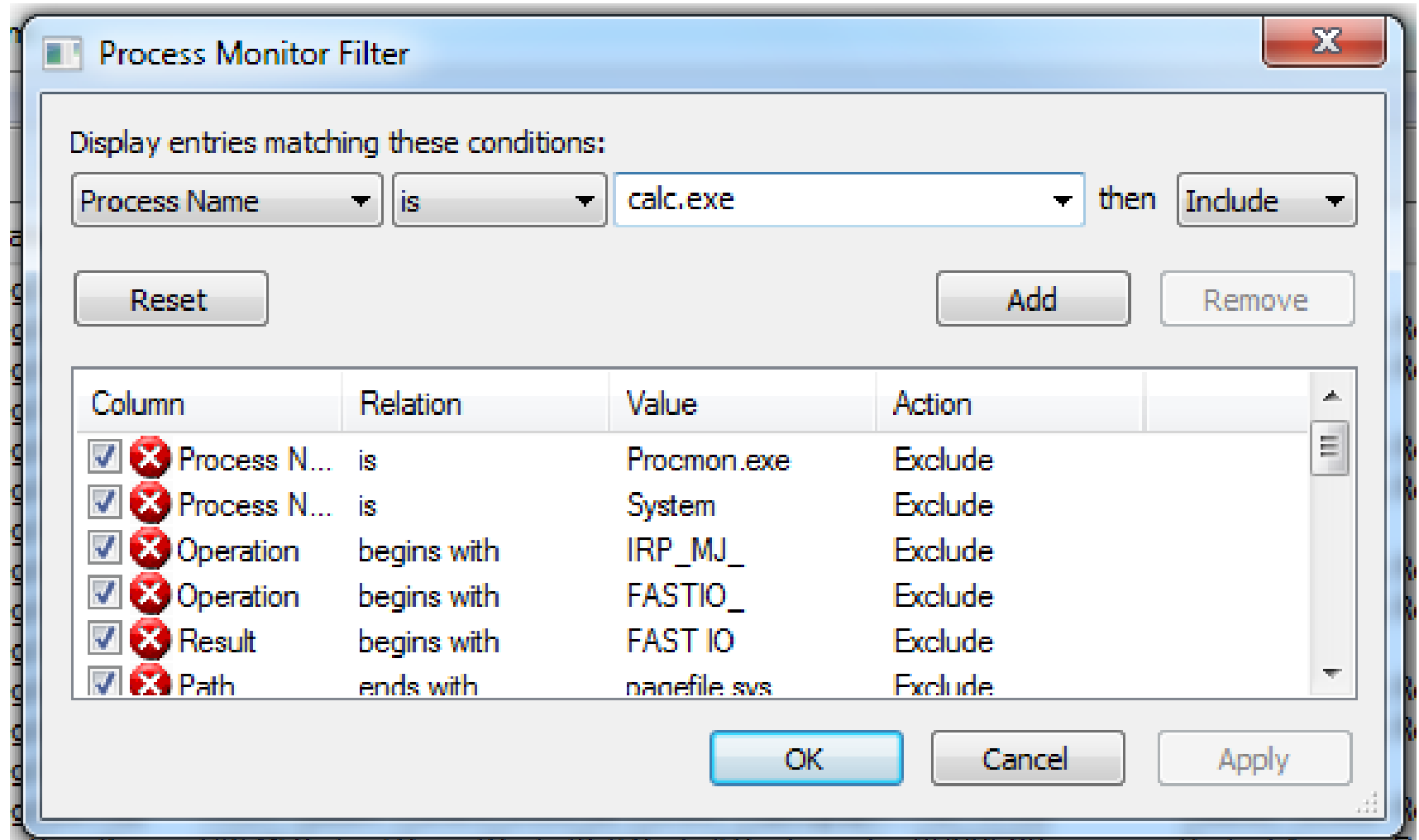


# Lọc và loại trừ kết quả

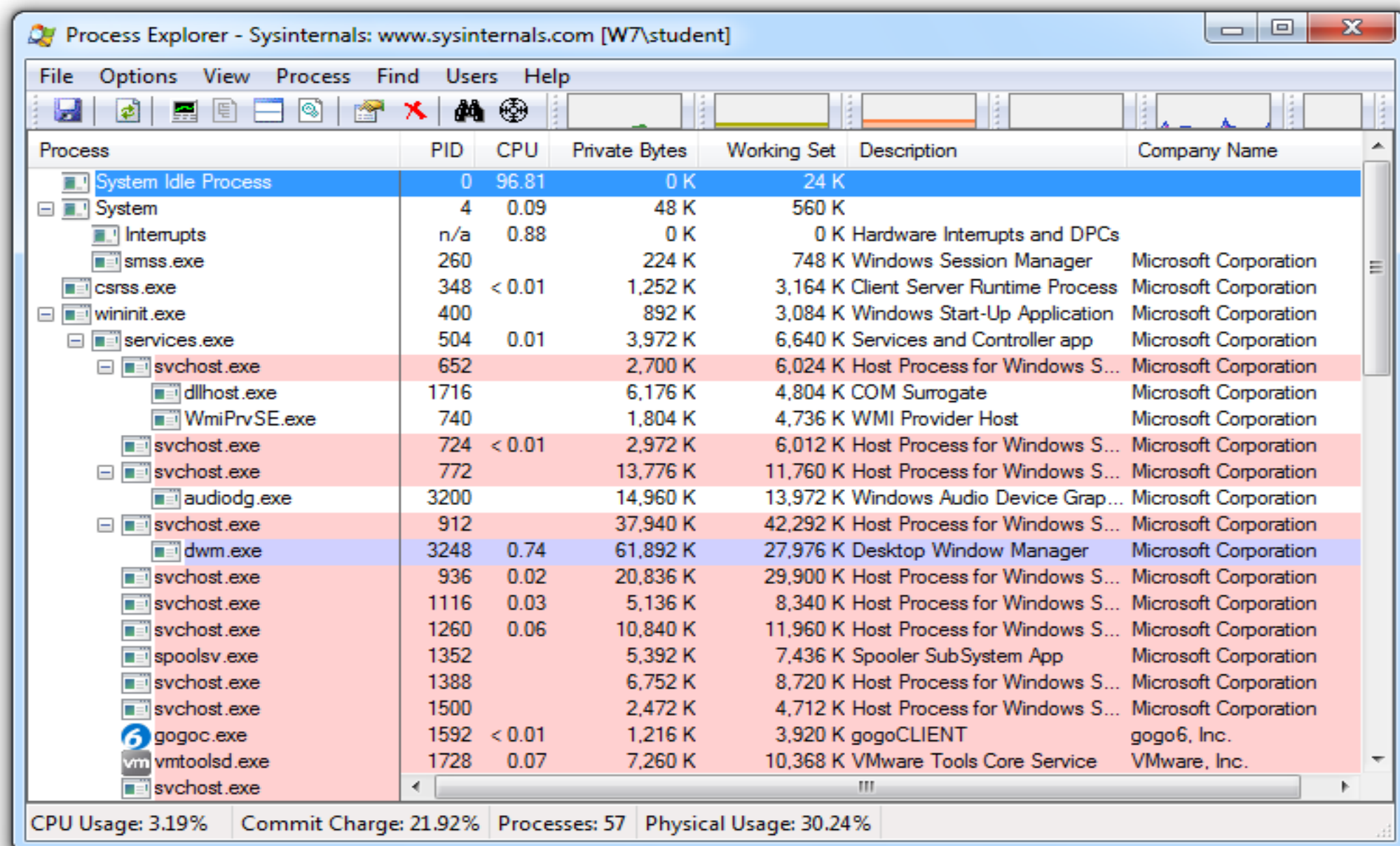
- ☐ Ấn các hoạt động thông thường/tin cậy trước khi thực thi mã độc
- ☐ Click chuột phải chọn tiến trình và chọn Exclude
- ☐ Không phải lúc nào cũng hoạt động ổn với tất cả các mẫu mã độc

# Filtering with Include

❑ Những bộ lọc hữu ích: Process Name, Operation and Detail



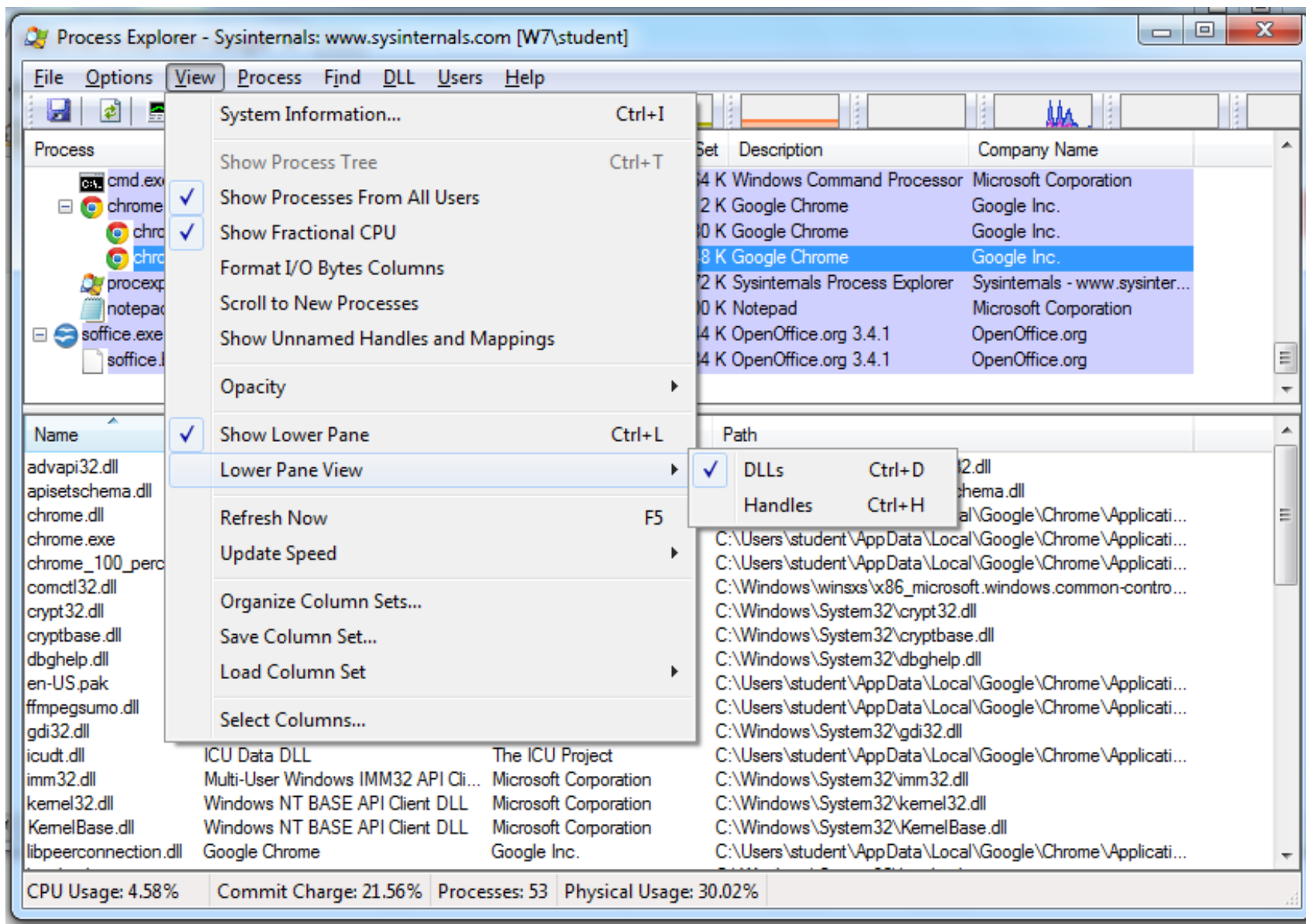
# Process Explorer



Process	PID	CPU	Private Bytes	Working Set	Description	Company Name
System Idle Process	0	96.81	0 K	24 K		
System	4	0.09	48 K	560 K		
Interrupts	n/a	0.88	0 K	0 K	Hardware Interrupts and DPCs	
smss.exe	260		224 K	748 K	Windows Session Manager	Microsoft Corporation
csrss.exe	348	< 0.01	1,252 K	3,164 K	Client Server Runtime Process	Microsoft Corporation
wininit.exe	400		892 K	3,084 K	Windows Start-Up Application	Microsoft Corporation
services.exe	504	0.01	3,972 K	6,640 K	Services and Controller app	Microsoft Corporation
svchost.exe	652		2,700 K	6,024 K	Host Process for Windows S...	Microsoft Corporation
dllhost.exe	1716		6,176 K	4,804 K	COM Surrogate	Microsoft Corporation
WmiPrvSE.exe	740		1,804 K	4,736 K	WMI Provider Host	Microsoft Corporation
svchost.exe	724	< 0.01	2,972 K	6,012 K	Host Process for Windows S...	Microsoft Corporation
svchost.exe	772		13,776 K	11,760 K	Host Process for Windows S...	Microsoft Corporation
audiodg.exe	3200		14,960 K	13,972 K	Windows Audio Device Grap...	Microsoft Corporation
svchost.exe	912		37,940 K	42,292 K	Host Process for Windows S...	Microsoft Corporation
dwm.exe	3248	0.74	61,892 K	27,976 K	Desktop Window Manager	Microsoft Corporation
svchost.exe	936	0.02	20,836 K	29,900 K	Host Process for Windows S...	Microsoft Corporation
svchost.exe	1116	0.03	5,136 K	8,340 K	Host Process for Windows S...	Microsoft Corporation
svchost.exe	1260	0.06	10,840 K	11,960 K	Host Process for Windows S...	Microsoft Corporation
spoolsv.exe	1352		5,392 K	7,436 K	Spooler SubSystem App	Microsoft Corporation
svchost.exe	1388		6,752 K	8,720 K	Host Process for Windows S...	Microsoft Corporation
svchost.exe	1500		2,472 K	4,712 K	Host Process for Windows S...	Microsoft Corporation
gogoc.exe	1592	< 0.01	1,216 K	3,920 K	gogoCLIENT	gogo6, Inc.
vmtoolsd.exe	1728	0.07	7,260 K	10,368 K	VMware Tools Core Service	VMware, Inc.
svchost.exe						

CPU Usage: 3.19%    Commit Charge: 21.92%    Processes: 57    Physical Usage: 30.24%

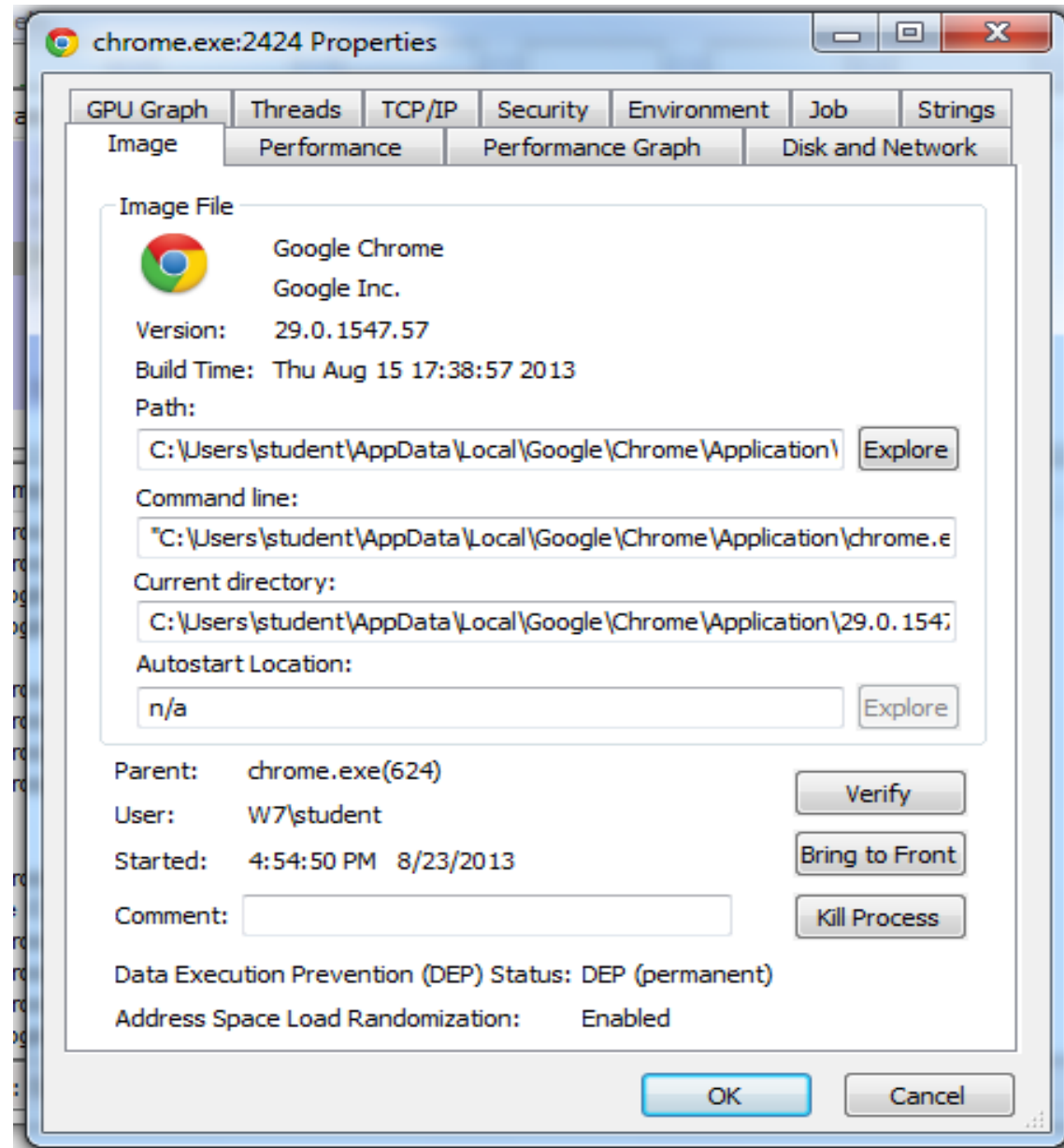
# DLL Mode - Process Explorer



# Properties - Process Explorer

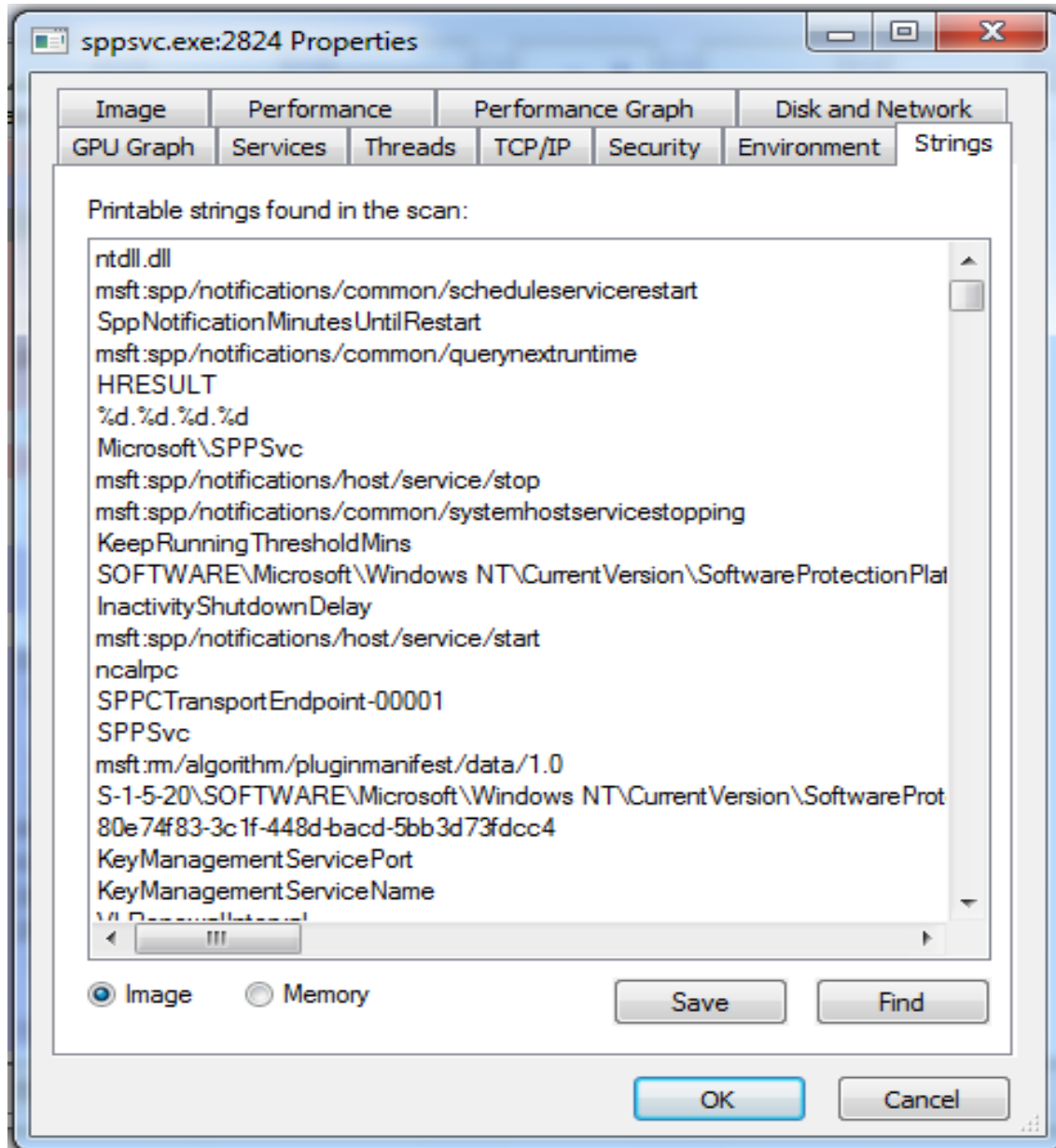
❑ Hiện thị trạng thái DEP và ASLR có được bật hay không.

❑ Kiểm tra chữ ký của Windows trên tệp tin.



# Strings - Process Explorer

❑ So sánh ảnh với chuỗi được nạp trên bộ nhớ, nếu chúng khác xa nhau, nó có thể chỉ ra quá trình thay thế.



# Phát hiện mã độc dạng tài liệu

- ❑ Các tệp tài liệu có thể chứa mã độc hay đoạn mã khai thác chính những chương trình đọc tài liệu (nếu có lỗ hổng bảo mật).
- ❑ Có thể thông qua Process Explorer để xem nó có khởi chạy một tiến trình nào đó hay không.
- ❑ Tab Image của Process Explorer Properties sẽ hiển thị vị trí của phần mềm độc hại.

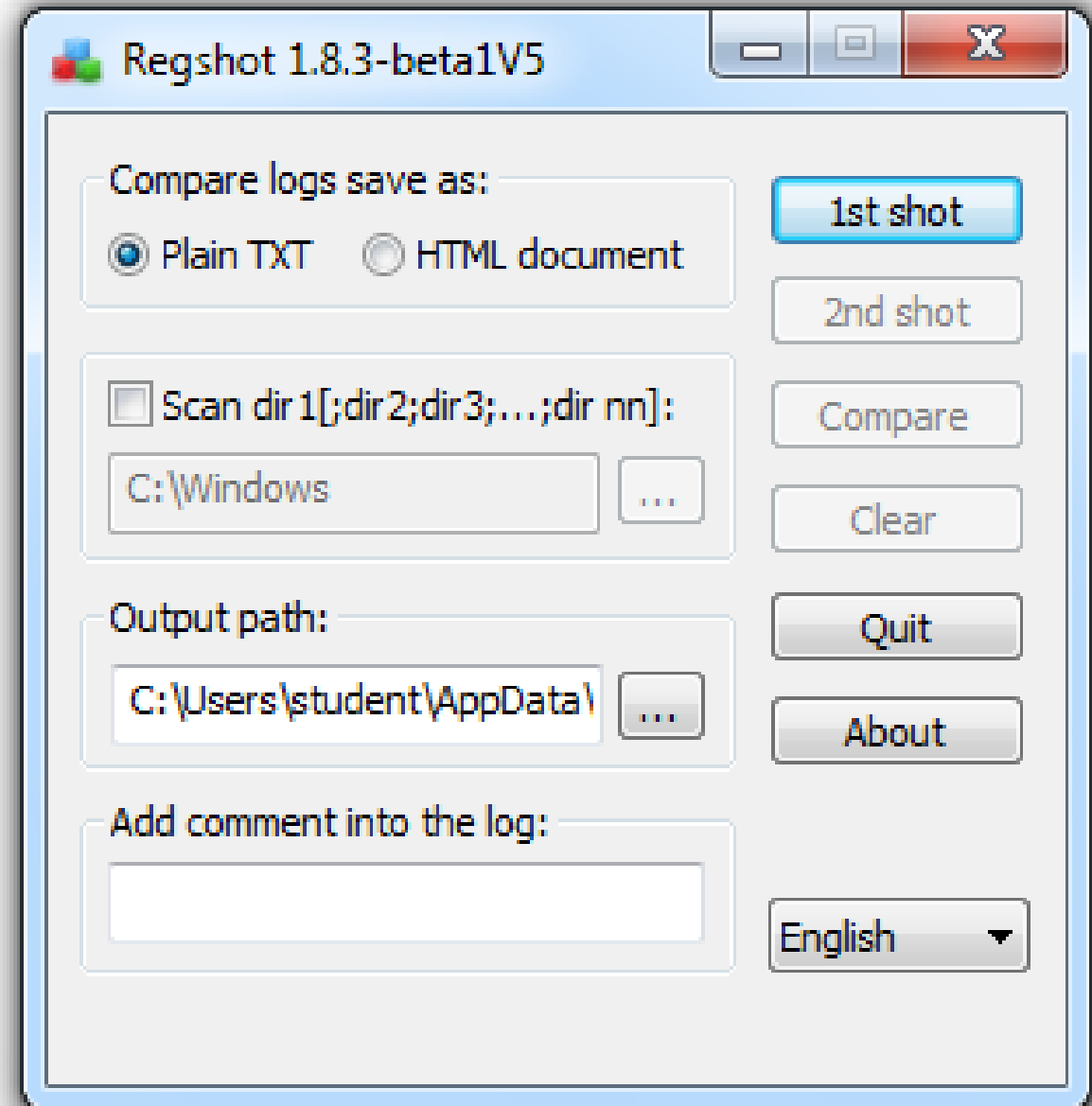
# Phân tích động

- ❑ **Giám sát hệ thống và theo dõi các tiến trình độc hại đồng thời theo dõi kết quả**
- ❑ **So sánh Registry**



# Regshot

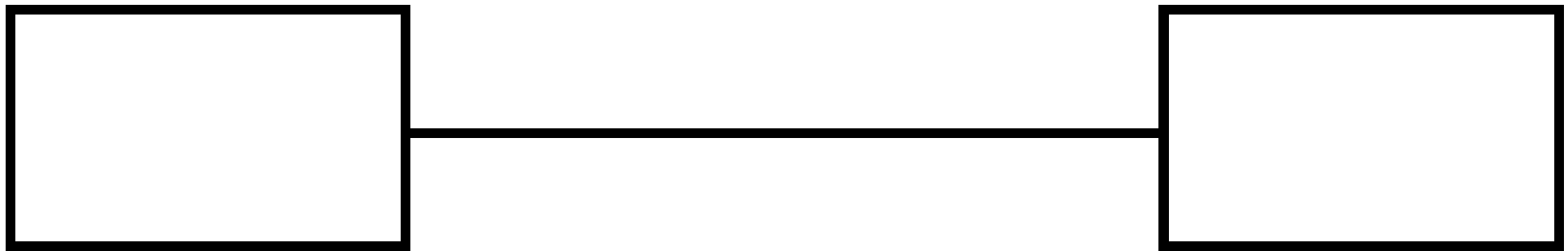
- ❑ Regshot hỗ trợ chụp lại Registry trước và sau khởi chạy mã độc
- ❑ Quan sát và so sánh xem các khóa Registry đã bị thay đổi



# Nội dung

1. Các phương pháp phân tích mã độc
2. Công cụ và kiến thức cơ sở
3. Các nguyên tắc khuyến nghị trong phân tích mã độc
4. Phân tích tĩnh cơ bản
5. Phân tích động cơ bản
6. Xây dựng môi trường phân tích mã độc

# Xây dựng môi trường phân tích mã độc



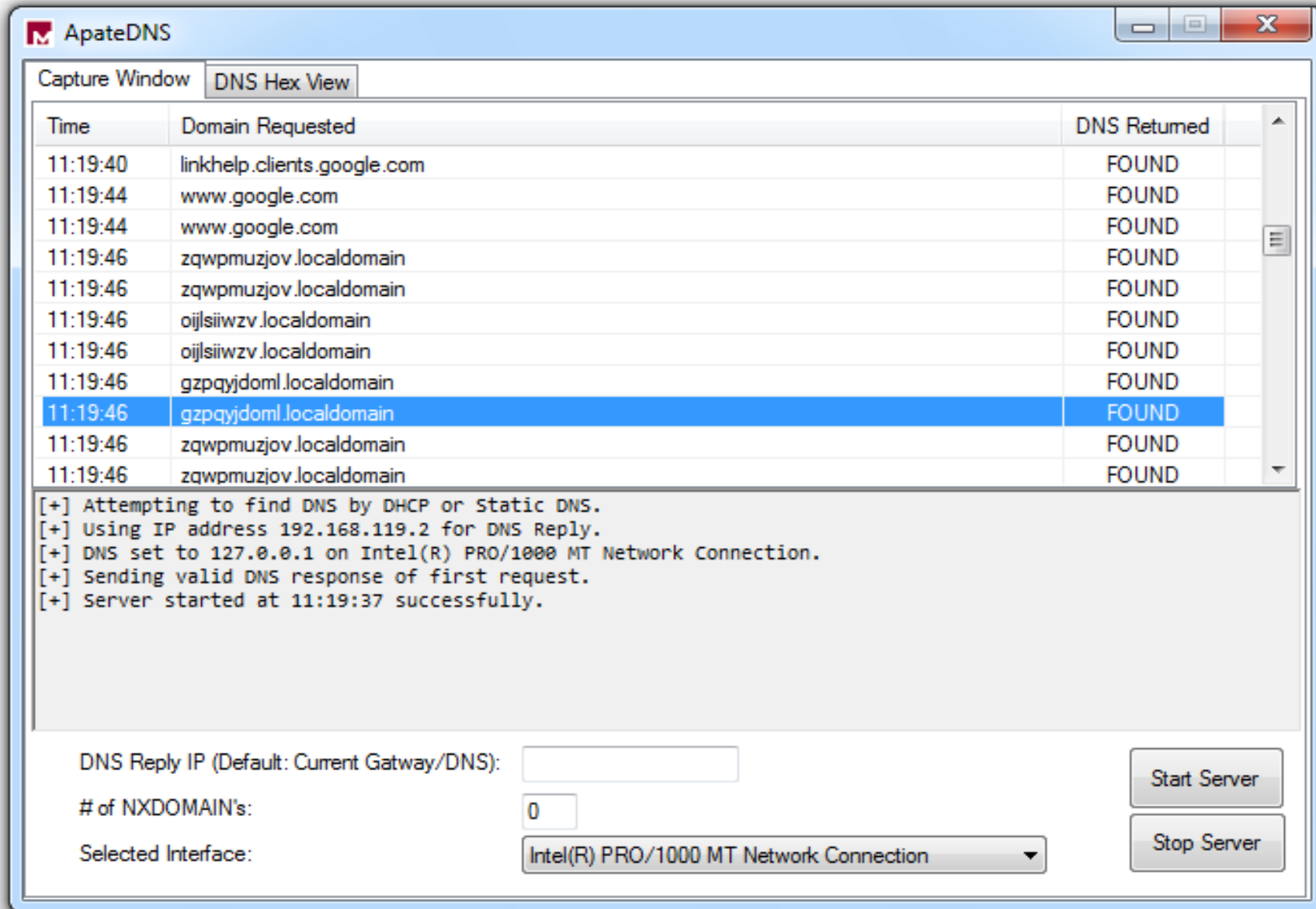
WinXP / Win 2008 server

Tools

Kali Linux

DNS

# Sử dụng ApateDNS để chuyển hướng luồng DNS



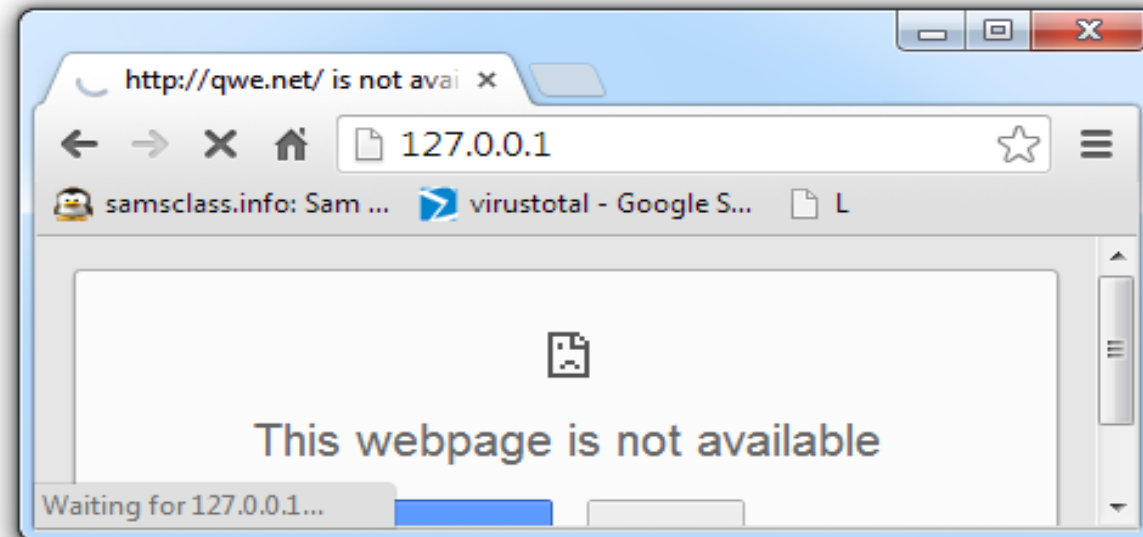
# ApateDNS

- ❑ Không hoạt động trên Windows XP và Windows 7
- ❑ Nslookup có thể hoạt động nhưng không thể nhìn thấy bất cứ điều gì trong trình duyệt hoặc với công cụ ping
- ❑ Công cụ thay thế: INetSim

# Monitoring with Ncat

```
Administrator: cmd - Shortcut (2) - ncat -l 80

C:\Windows\System32>ncat -l 80
GET / HTTP/1.1
Host: 127.0.0.1
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/29.0.1547.57 Safari/537.36
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-US,en;q=0.8
```



# Packet Sniffing with Wireshark

Capturing from Intel(R) PRO/1000 MT Network Connection

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: http Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
1101	7.515707	192.168.119.154	23.65.1.224	HTTP	GET /f.gif?_u=137745237561
1106	7.537336	18.181.0.31	192.168.119.154	HTTP	HTTP/1.1 200 OK (PNG)
1108	7.557449	93.184.216.139	192.168.119.154	HTTP	[TCP Retransmission] Cont
1110	7.590291	23.65.1.224	192.168.119.154	HTTP	HTTP/1.1 200 OK (GIF89a)
1111	7.691258	23.65.1.224	192.168.119.154	HTTP	[TCP Retransmission] HTTP/
1189	36.858744	192.168.119.154	199.16.156.21	HTTP	GET /widgets/timelines/pag
1193	36.881799	192.168.119.154	199.16.156.21	HTTP	GET /widgets/timelines/pag
1196	36.954204	199.16.156.21	192.168.119.154	HTTP	HTTP/1.1 200 OK (applicat
1199	37.045979	199.16.156.21	192.168.119.154	HTTP	HTTP/1.1 200 OK (applicat
1369	96.750725	192.168.119.154	199.16.156.21	HTTP	GET /widgets/timelines/pag
1373	96.772892	192.168.119.154	199.16.156.21	HTTP	GET /widgets/timelines/pag
1376	96.846439	199.16.156.21	192.168.119.154	HTTP	HTTP/1.1 200 OK (applicat
1381	96.944497	199.16.156.21	192.168.119.154	HTTP	HTTP/1.1 200 OK (applicat

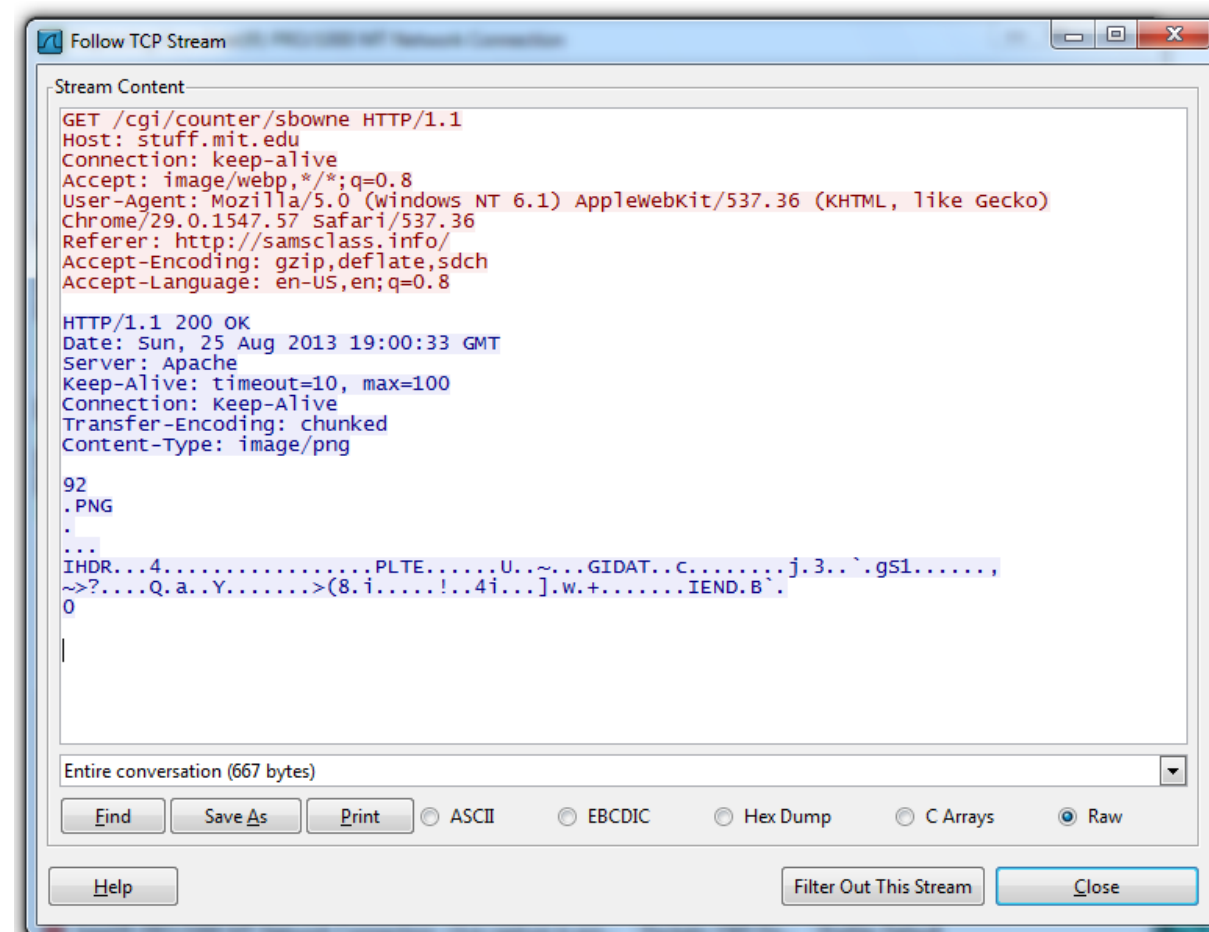
Frame 48: 437 bytes on wire (3496 bits), 437 bytes captured (3496 bits)

Ethernet II, Src: Vmware\_52:34:92 (00:0c:29:52:34:92), Dst: Vmware\_e3:22:f1 (00:50:56:00:00:00)

Internet Protocol Version 4, Src: 192.168.119.154 (192.168.119.154), Dst: 141.101.11

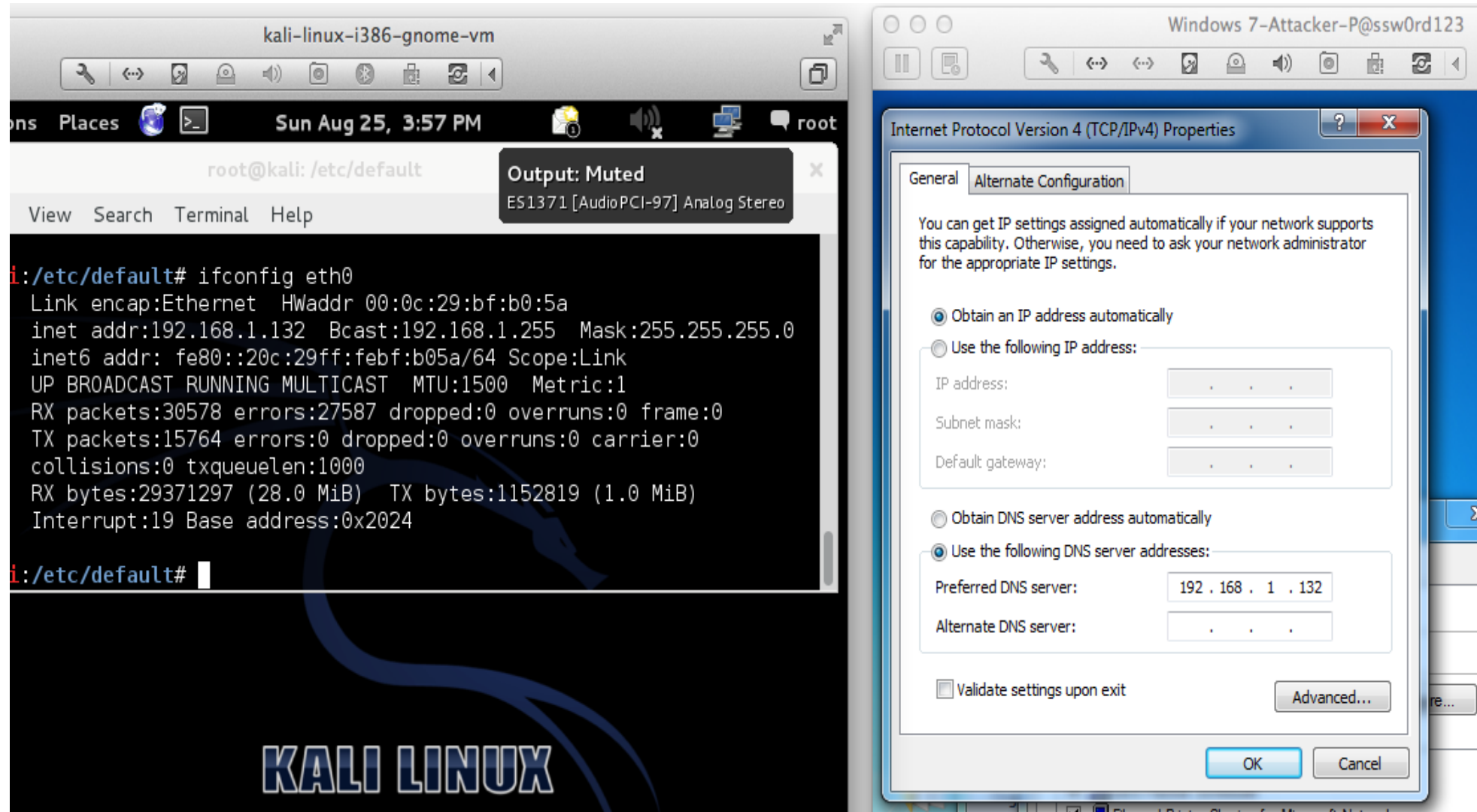
# Follow TCP Stream

Có thể lưu lại file từ stream

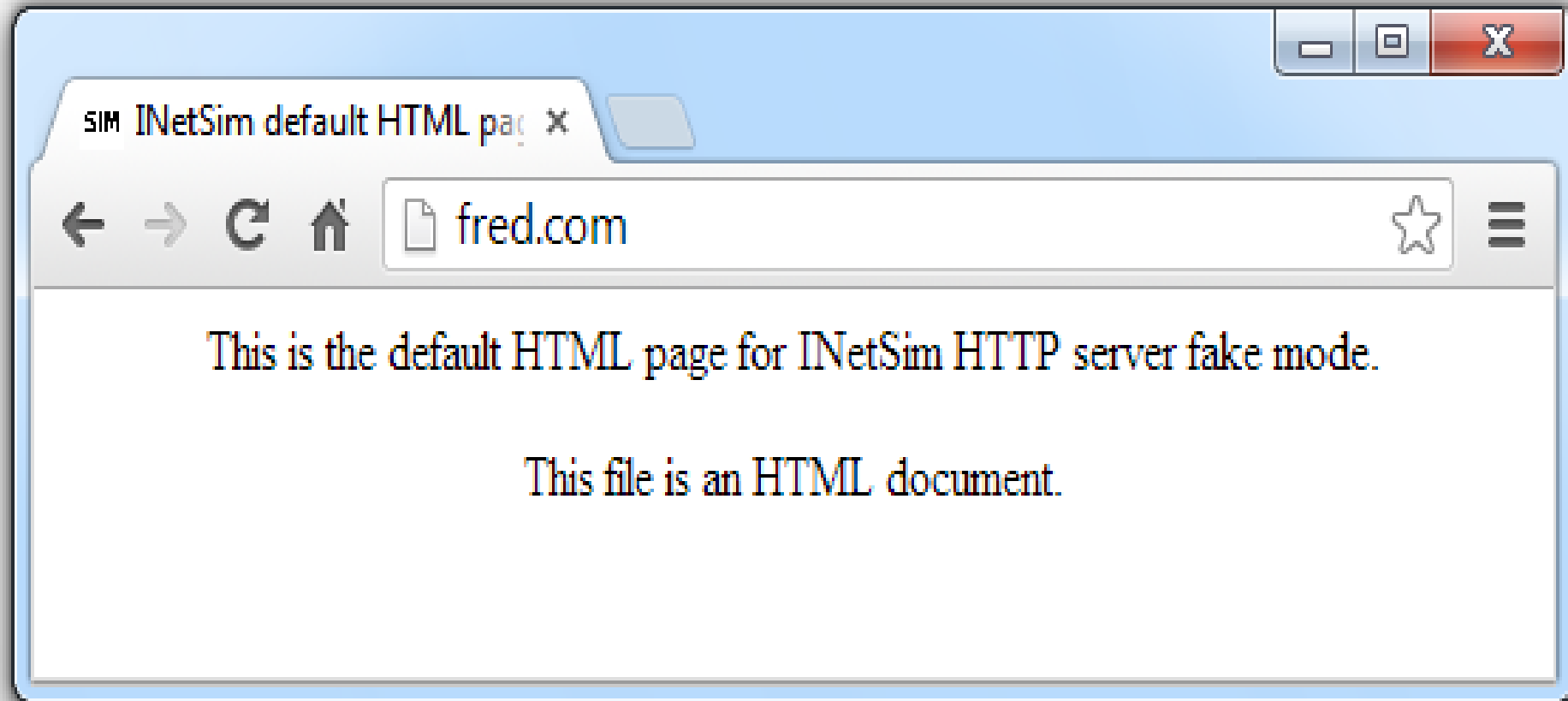




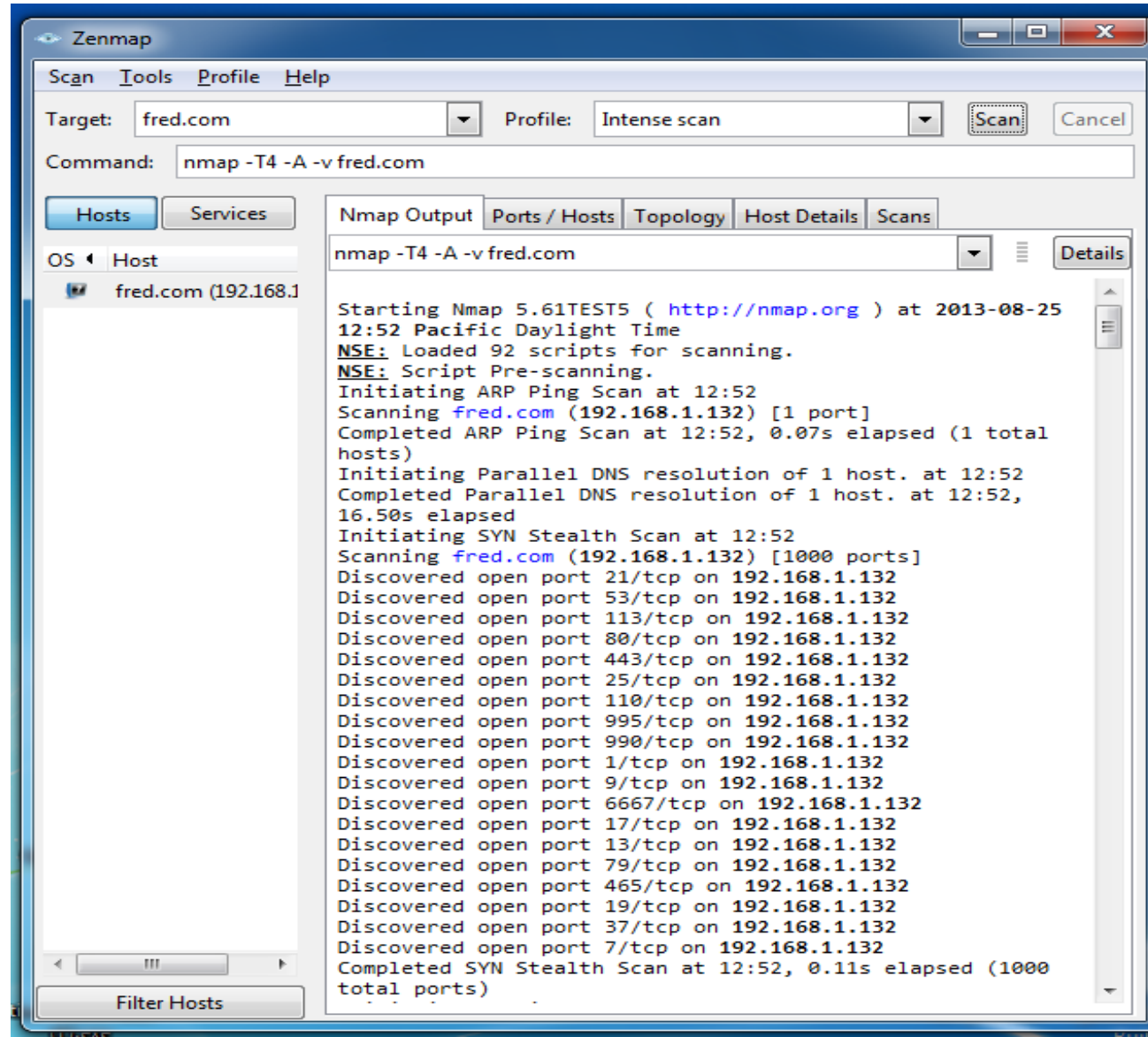
# INetSim



# INetSim Fools a Browser



# INetSim Fools Nmap



# Nội dung

- 1. Các phương pháp phân tích mã độc**
- 2. Công cụ và kiến thức cơ sở**
- 3. Các nguyên tắc khuyến nghị trong phân tích mã độc**
- 4. Phân tích tĩnh cơ bản**
- 5. Phân tích động cơ bản**
- 6. Xây dựng môi trường phân tích mã độc**