



**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**  
Posts and Telecommunications Institute of Technology

# PHÂN TÍCH MÃ ĐỘC

**KHOA AN TOÀN THÔNG TIN**  
**TS. ĐÌNH TRƯỜNG DUY**



**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**  
Posts and Telecommunications Institute of Technology

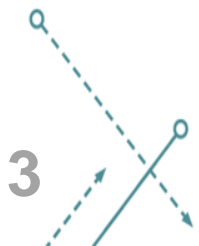
# PHÂN TÍCH MÃ ĐỘC

## Windows internal

**KHOA AN TOÀN THÔNG TIN**  
**TS. ĐINH TRƯỜNG DUY**

# Giới thiệu

1. Windows Registry
2. Important Directories
3. Windows Processes
4. Windows Services
5. Syscal



# Windows Registry

- Không chỉ hệ điều hành, mà hầu hết phần mềm cũng sử dụng registry để lưu trữ thông tin cấu hình và cài đặt liên quan đến phần mềm của họ. Các mục nhập trong registry được sắp xếp theo cấu trúc cây với các gốc cấp cao được gọi là **hives**.
- Hives là các gốc cấp cao trong cấu trúc cây của registry và chứa các khối dữ liệu liên quan. Windows sử dụng năm hives chính sau:
  1. HKEY\_CLASSES\_ROOT (HKCR)
  2. HKEY\_LOCAL\_MACHINE (HKLM)
  3. HKEY\_USERS (HKU)
  4. HKEY\_CURRENT\_CONFIG (HKCC)
  5. HKEY\_CURRENT\_USER (HKCU)

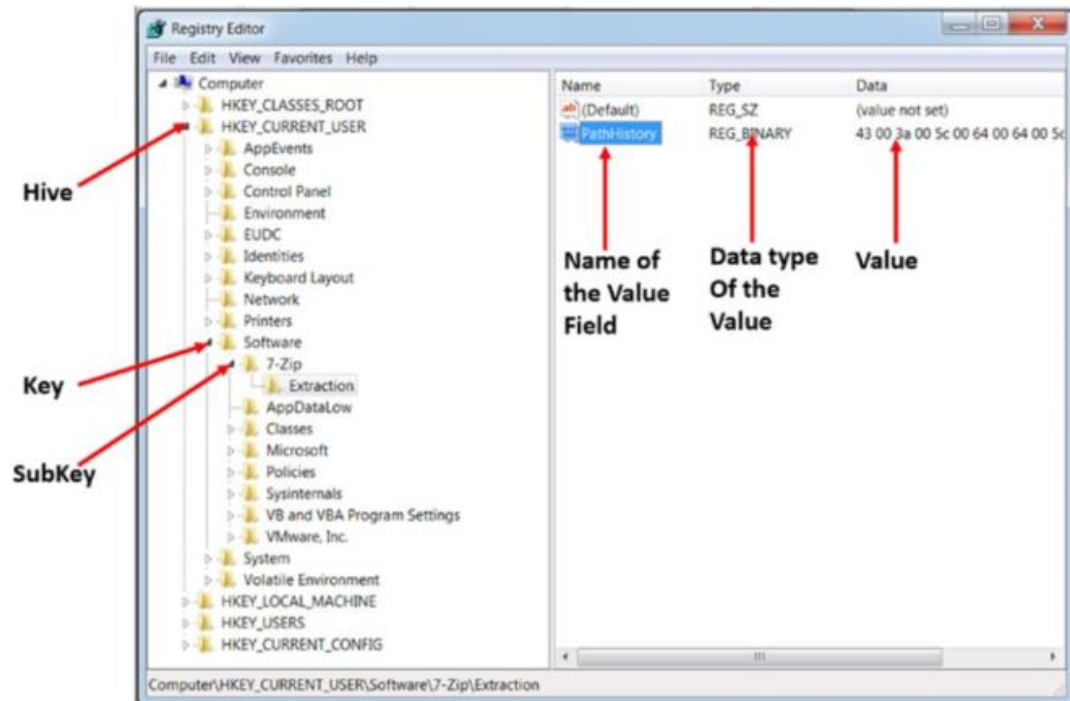


# Windows Registry

- HKEY\_CLASSES\_ROOT (HKCR): Chứa thông tin về các loại tệp và các phần mở rộng tệp, cũng như các thông tin về các đối tượng COM (Component Object Model) và các khóa đăng ký liên quan đến phần mở rộng shell.
- HKEY\_CURRENT\_USER (HKCU): Chứa các cấu hình đăng nhập của người dùng hiện tại, bao gồm thông tin về giao diện người dùng, cài đặt ứng dụng và các tùy chọn cá nhân.
- HKEY\_LOCAL\_MACHINE (HKLM): Chứa thông tin về cấu hình hệ thống và phần mềm, bao gồm cài đặt hệ thống, thông tin phần cứng, cấu hình ứng dụng và các khóa đăng ký liên quan đến hệ thống.
- HKEY\_USERS (HKU): Chứa thông tin về các tài khoản người dùng được tạo trên máy tính, bao gồm các cấu hình người dùng và các tùy chọn cá nhân.
- HKEY\_CURRENT\_CONFIG (HKCC): Chứa thông tin về cấu hình hiện tại của phần cứng và hệ thống, bao gồm các cấu hình liên quan đến hiển thị, âm thanh, mạng và các thiết bị khác.

# Windows Registry

- Mỗi hive chứa các khóa (keys) và các giá trị (values) liên quan đến cấu hình và cài đặt của phần mềm.
- Các khóa và giá trị trong registry được sử dụng để lưu trữ thông tin như đường dẫn tới tệp, các cài đặt ứng dụng, cấu hình mạng và rất nhiều thông tin khác.



# Windows Registry

- Mã độc có thể sửa đổi thông tin registry để thay đổi hành vi hệ thống theo ý muốn của nó, và nó thực hiện điều này bằng cách sử dụng các API Win32.
- Những API thông thường nhất mà mã độc sử dụng là sửa đổi các giá trị registry được dùng để thực thi phần mềm trong quá trình khởi động hệ thống hoặc đăng nhập người dùng.
- Mã độc sửa đổi các giá trị này để hệ thống tự động khởi động mã độc trong quá trình khởi động hệ thống.
- Các kỹ thuật này được gọi là cơ chế duy trì tính bền vững (persistence mechanisms) trong Windows.
- Nếu hệ điều hành được cài đặt trên một máy ảo để phân tích mã độc, các dấu vết của máy ảo sẽ có trong registry.
- Mã độc có thể truy vấn các khóa registry này và tìm hiểu xem hệ điều hành mục tiêu của nạn nhân có được cài đặt trên máy ảo hay không.
- Trong trường hợp này, mã độc có thể không thể hiện hành vi thực sự của nó và có thể đánh lừa người phân tích.

"HKEY\_LOCAL\_MACHINE\HARDWARE\DESCRIPTION\System\BIOS\SystemProductName" hoặc  
"HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Virtual Machine\Guest\Parameters\PhysicalHostName"

# Important Directories

- Thư mục **system32** hoặc đường dẫn **C:\Windows\system32** chứa hầu hết các chương trình và công cụ hệ thống
- Các chương trình khác: *smss.exe*, *svchost.exe*, *services.exe*, *explorer.exe*, *winlogon.exe*, *calc.exe*...
- **Program files** or path **C:\Program Files** or **C:\Program Files (x86)** lưu các chương trình người dùng cài đặt
- Các thư mục như: AppData và Roaming thường được sử dụng bởi mã độc để sao chép chính nó vào các thư mục này và thực thi chúng
  - My Documents C:\Users\<user>\Documents
  - Desktop C:\Users\<user>\Desktop
  - AppData C:\Users\<user>\AppData
  - Roaming C:\Users\<user>\AppData\Roaming





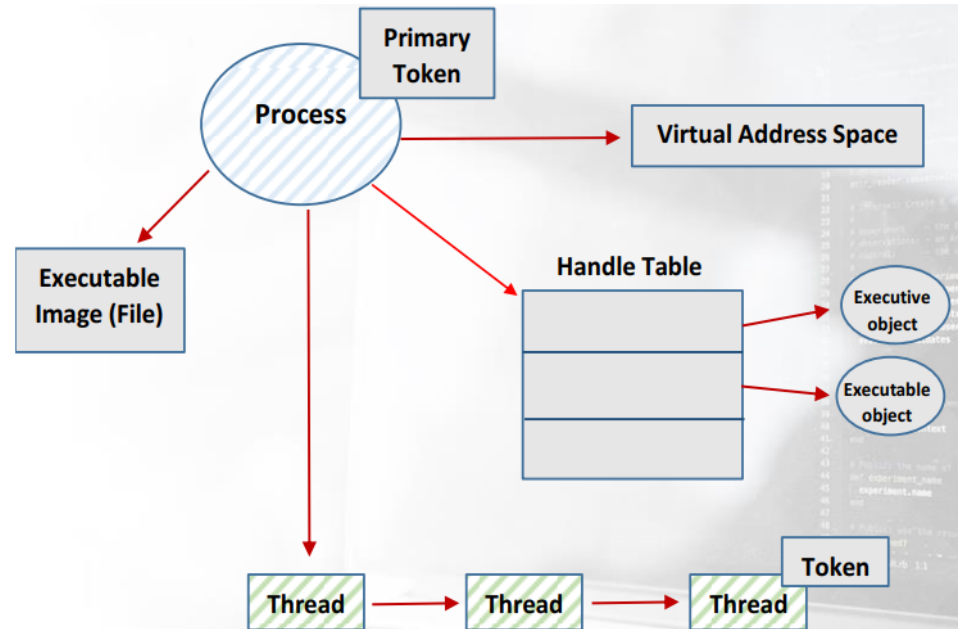
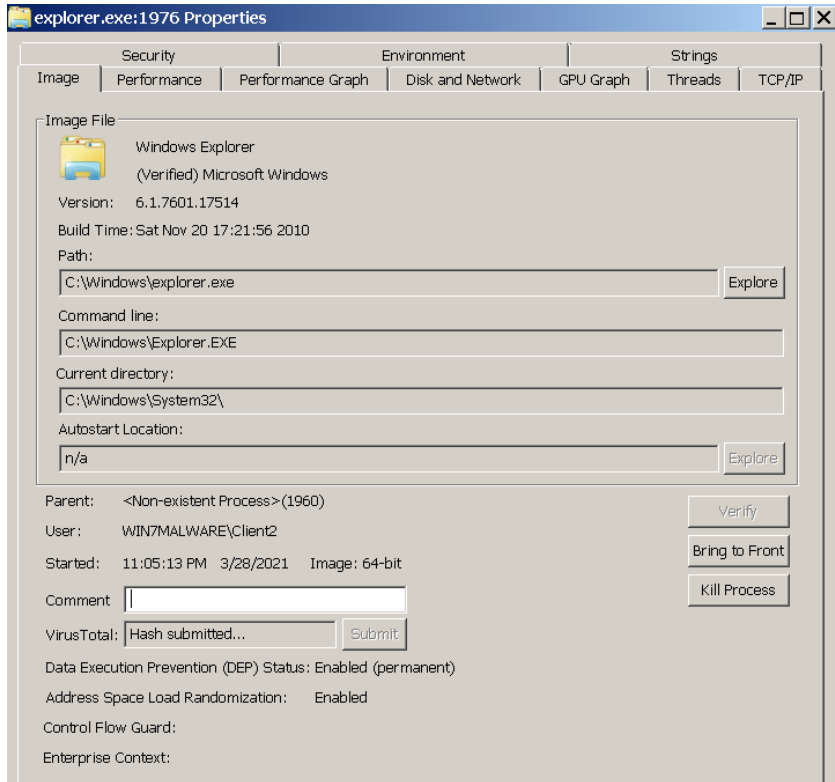
# Windows processes

- Theo mặc định, hđh Windows chạy nhiều tiến trình hệ thống.
- Hầu hết các tiến trình này được tạo ra từ các chương trình nằm trong thư mục **system32**.
- Phần mềm độc hại chạy trên hệ thống bằng cách:
  - Giả mạo thành một tiến trình hệ thống
  - Sửa đổi các tiến trình hệ thống đang chạy để thực hiện mục đích độc hại. Các kỹ thuật như tiêm mã (code injection) và process hollowing được sử dụng để thực hiện việc này.

Một vài tiến trình hệ thống quan trọng:

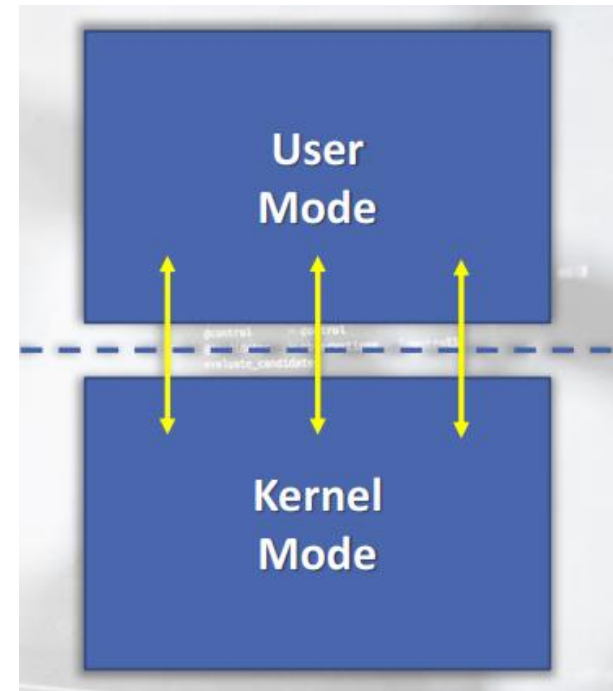
- **smss.exe**
- **wininit.exe**
- **winlogon.exe**
- **explorer.exe**
- **csrss.exe**
- **userinit.exe**
- **services.exe**
- **lsmd.exe**
- **lsass.exe**
- **svchost.exe**

# Windows processes



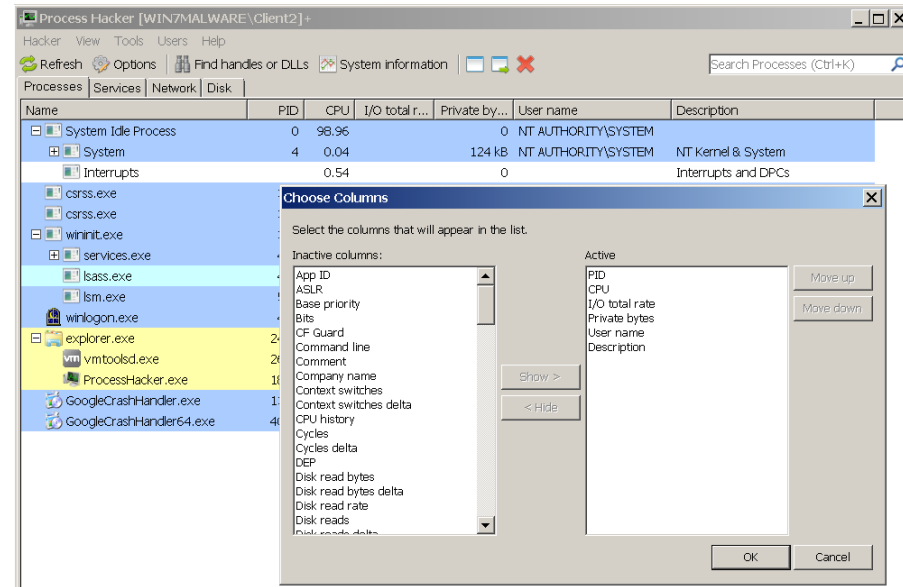
# Windows processes

- Các tiến trình có thể chạy bằng các mode sau:
  - User-mode
  - Kernel-mode
- CPU sẽ chuyển giữa 2 mode này phụ thuộc vào mã đang thực thi
- Khi chương trình chạy nó sẽ được triển khai trên user-mode
- CPU sẽ chuyển sang kernel-mode → Khi cần thực hiện một hoạt động có đặc quyền để làm việc với tài nguyên hệ thống thông qua kernel (ví dụ: mở một tệp trên đĩa)



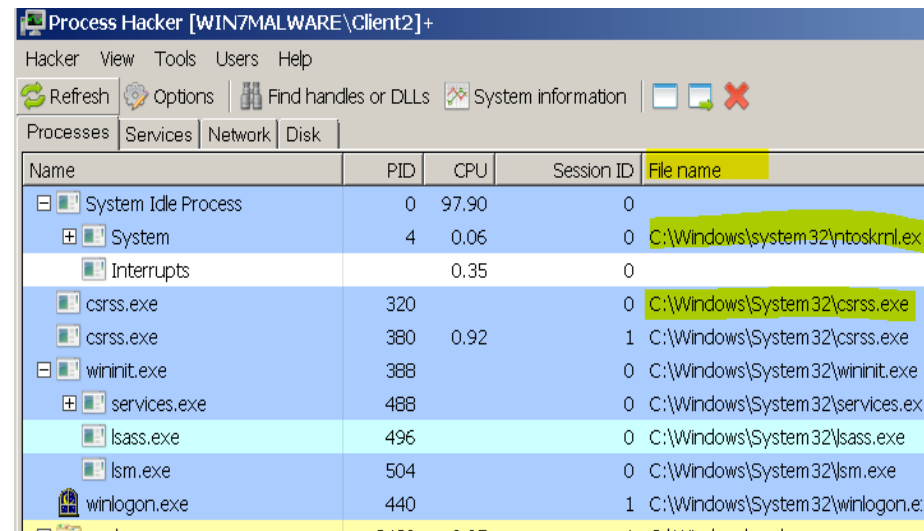
# Windows processes

- Một tiến trình có thể có nhiều thuộc tính: PID, tiến trình cha, đường dẫn thực thi, bộ nhớ ảo...
- Công cụ: Task Manager, **Process Explorer**, **Process Hacker**, CurrProcess...
- Cần đảm bảo rằng các cột sau cần phải được active: *PID, CPU, Session ID, File name, User name, Private bytes, Description, and I/O total rate*



# Windows processes

- Đây là đường dẫn của chương trình mà tiến trình tạo ra
- Theo mặc định thì tệp nhị phân của các tiến trình của hệ thống được lưu trong:  
**C:\Windows\system32.**

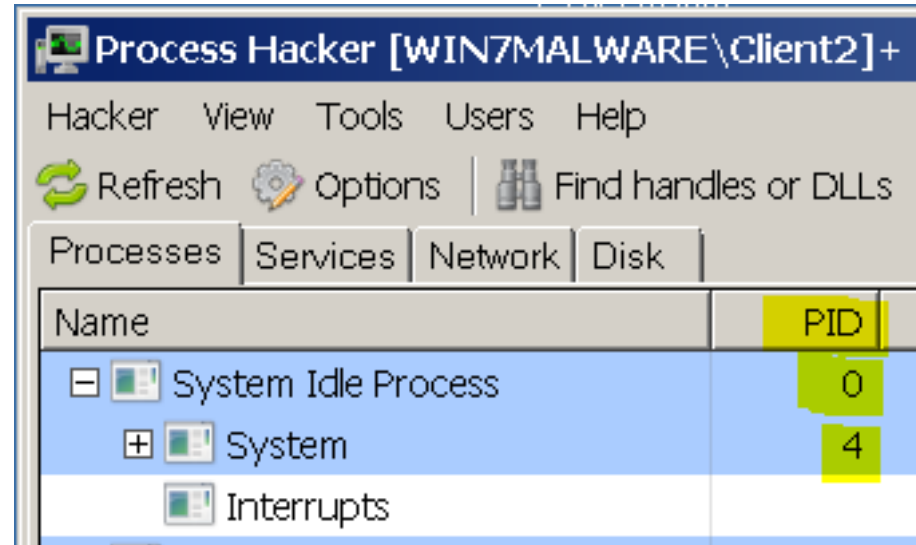


Name	PID	CPU	Session ID	File name
System Idle Process	0	97.90	0	
System	4	0.06	0	C:\Windows\system32\ntoskrnl.ex
Interrupts		0.35	0	
csrss.exe	320		0	C:\Windows\System32\csrss.exe
csrss.exe	380	0.92	1	C:\Windows\System32\csrss.exe
wininit.exe	388		0	C:\Windows\System32\wininit.exe
services.exe	488		0	C:\Windows\System32\services.ex
lsass.exe	496		0	C:\Windows\System32\lsass.exe
lsmd.exe	504		0	C:\Windows\System32\lsmd.exe
winlogon.exe	440		1	C:\Windows\System32\winlogon.e

→ Nếu tiến trình tồn tại trong hệ thống nhưng đường dẫn lại không có trong C:\Windows\system32 → **ngghi ngờ**

# Windows processes

- PID là ID độc nhất của một tiến trình
  - Hai tiến trình hệ thống đã cố định PID là **SYSTEM IDLE PROCESS** có giá trị 0 và **SYSTEM** có giá trị 4.
- Hệ thống nên chỉ chạy một phiên bản của các tiến trình này trên hệ thống.



The screenshot shows the Process Hacker application window. The title bar reads "Process Hacker [WIN7MALWARE\Client2] +". The menu bar includes "Hacker", "View", "Tools", "Users", and "Help". Below the menu bar are buttons for "Refresh", "Options", and "Find handles or DLLs". The "Processes" tab is selected, showing a list of processes. The list has two columns: "Name" and "PID". The "PID" column is highlighted in yellow. The processes listed are "System Idle Process" (PID 0), "System" (PID 4), and "Interrupts".

Name	PID
System Idle Process	0
System	4
Interrupts	

# Windows processes

- Session được tạo ra cho mỗi người dùng đăng nhập vào hệ thống và xác định bởi ID Session.
  - Khi Windows khởi động, nó tạo ra một session 0 mặc định (phiên phi tương tác).
  - Session 1 và sau đó sẽ được tạo do người dùng đầu tiên đăng nhập.
- Không người dùng nào có thể đăng nhập với session 0
- Tất cả các dịch vụ khởi động và chương trình hệ thống của Windows sẽ được khởi động bằng session 0

Process Hacker [WIN7MALWARE\Client2]+

Hacker View Tools Users Help

Refresh Options Find handles or DLLs System information

Processes Services Network Disk

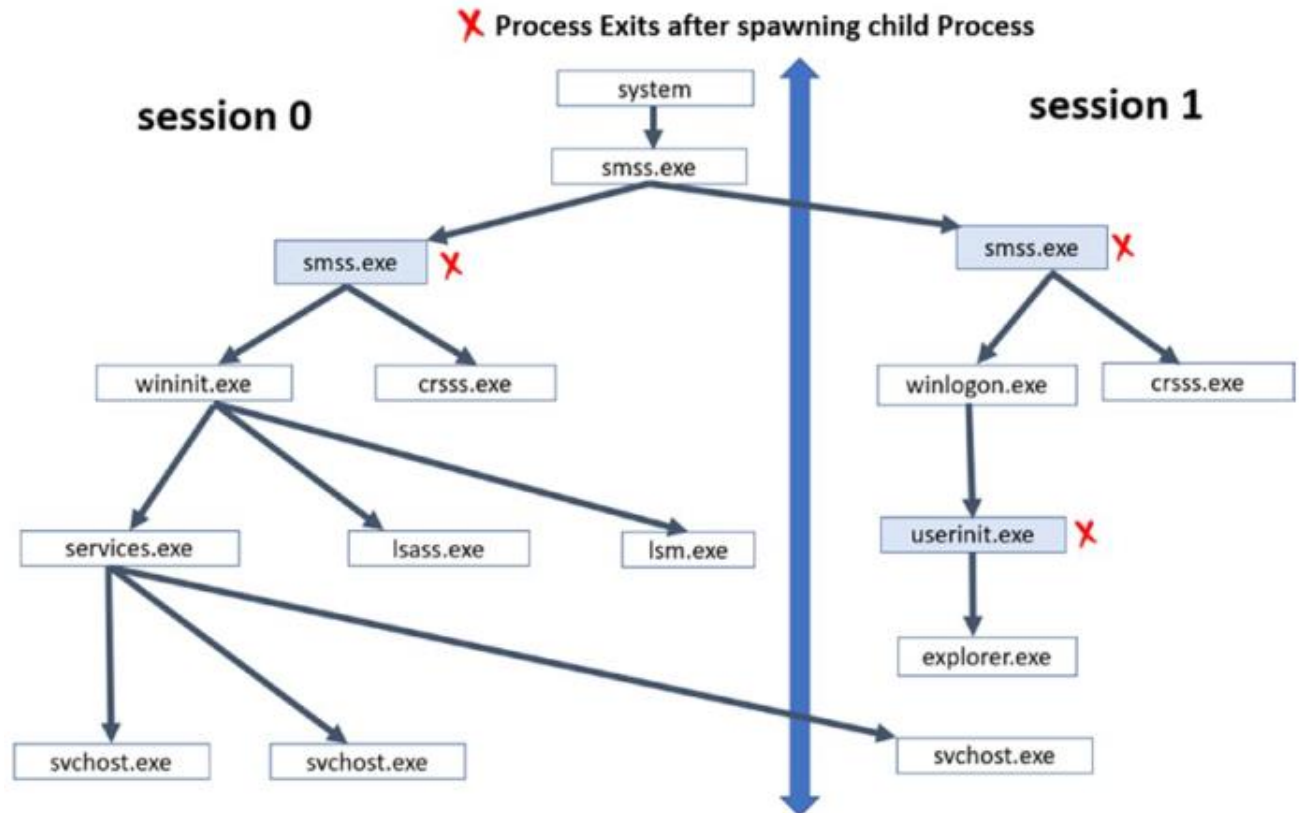
Name	PID	CPU	Session ID	File
System Idle Process	0	97.05	0	
System	4	0.03	0	C:\
Interrupts		1.35	0	
csrss.exe	320		0	C:\
csrss.exe	380	0.13	1	C:\
wininit.exe	388		0	C:\
services.exe	488		0	C:\
lsass.exe	496		0	C:\
lsim.exe	504		0	C:\
winlogon.exe	440		1	C:\
explorer.exe	2420	0.29	1	C:\
GoogleCrashHandler.exe	1368		0	C:\
GoogleCrashHandler64.exe	4024		0	C:\

**svchost.exe & other system processes → session 0**

**Winlogon.exe, csrss.exe, explorer.exe, taskhost.exe → user session**

# Windows processes

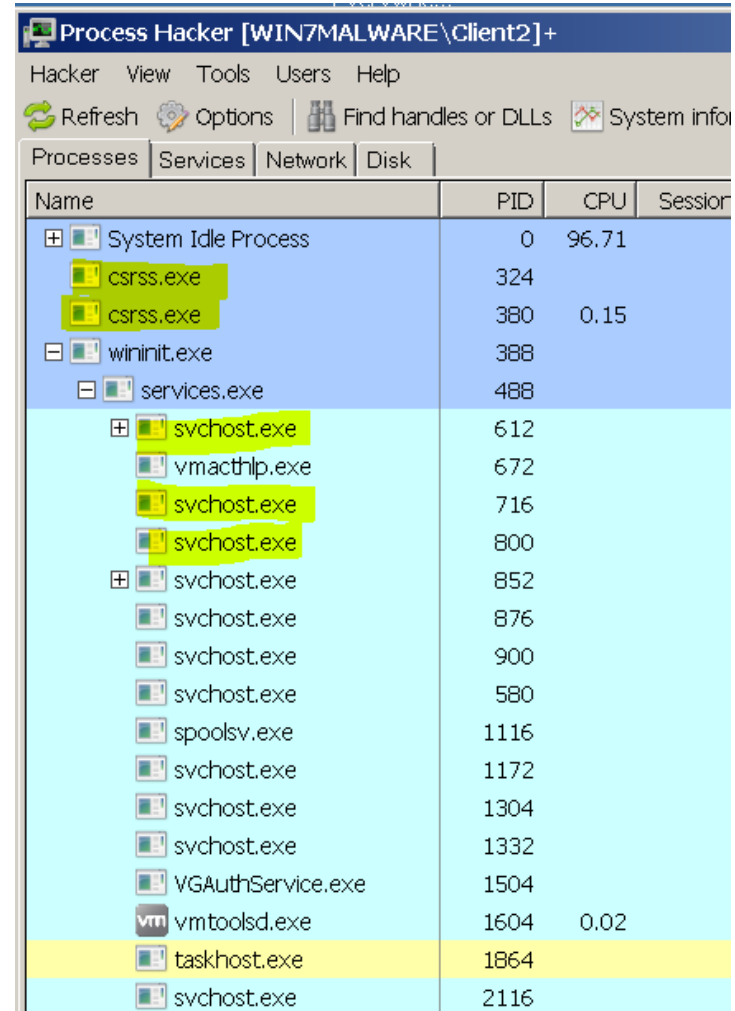
- Một vài tiến trình hệ thống có các tiến trình cha đặc biệt
- Nếu một tiến trình với cùng tên với tiến trình hệ thống nhưng tiến trình cha không khớp với cây tiến trình → **nghi ngờ mã độc**





# Windows processes

- Hầu hết các tiến trình hệ thống chỉ có một phiên bản (instance) đang chạy vào một thời điểm.
  - Ngoại lệ duy nhất là **csrss.exe**, có hai instance đang chạy.
  - Một ngoại lệ khác là **svchost.exe**, có thể có nhiều instance đang chạy.
- Nếu phát hiện nhiều hơn 2 instance của **csrss.exe** hoặc nhiều hơn một instance của bất kỳ tiến trình hệ thống nào khác (trừ svchost.exe), thì có khả năng cao các phiên bản tiến trình thừa này là các phiên bản của phần mềm độc hại.



Process Hacker [WIN7MALWARE\Client2] +

Hacker View Tools Users Help

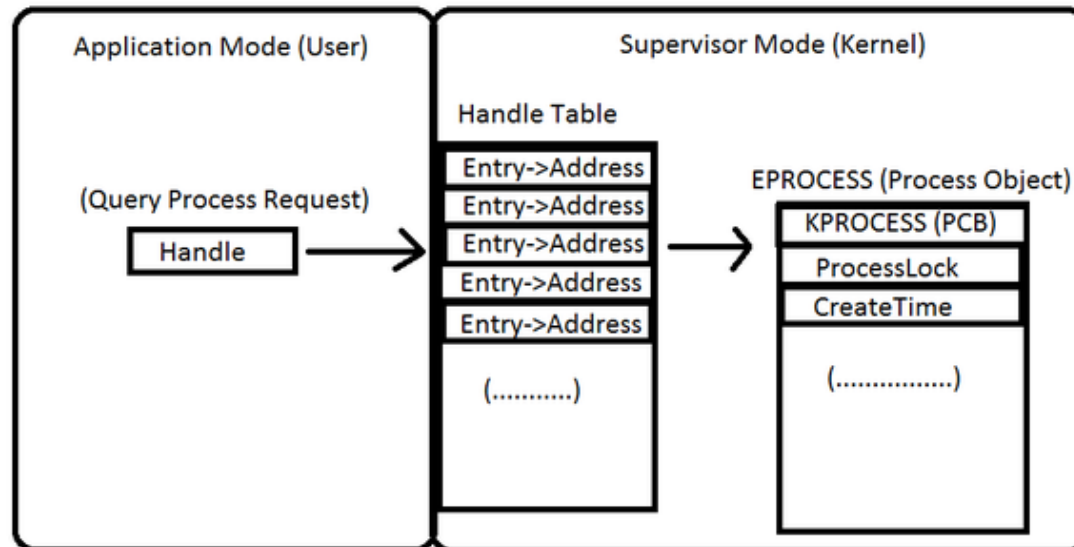
Refresh Options Find handles or DLLs System info

Processes Services Network Disk

Name	PID	CPU	Session
System Idle Process	0	96.71	
csrss.exe	324		
csrss.exe	380	0.15	
wininit.exe	388		
services.exe	488		
svchost.exe	612		
vmacthlp.exe	672		
svchost.exe	716		
svchost.exe	800		
svchost.exe	852		
svchost.exe	876		
svchost.exe	900		
svchost.exe	580		
spoolsv.exe	1116		
svchost.exe	1172		
svchost.exe	1304		
svchost.exe	1332		
VGAuthService.exe	1504		
vmtoolsd.exe	1604	0.02	
taskhost.exe	1864		
svchost.exe	2116		

# Windows processes

- Handles (bộ xử lý) thường là các tham chiếu đến các đối tượng hệ thống, nhưng các đối tượng này không thể được truy cập trực tiếp từ các yêu cầu chế độ người dùng. Thay vào đó, các handle đóng vai trò như các con trỏ gián tiếp đến các đối tượng trong không gian kernel.
- Mỗi tiến trình cũng có một bảng, bảng này là của riêng từng tiến trình và trỏ đến các đối tượng kernel.

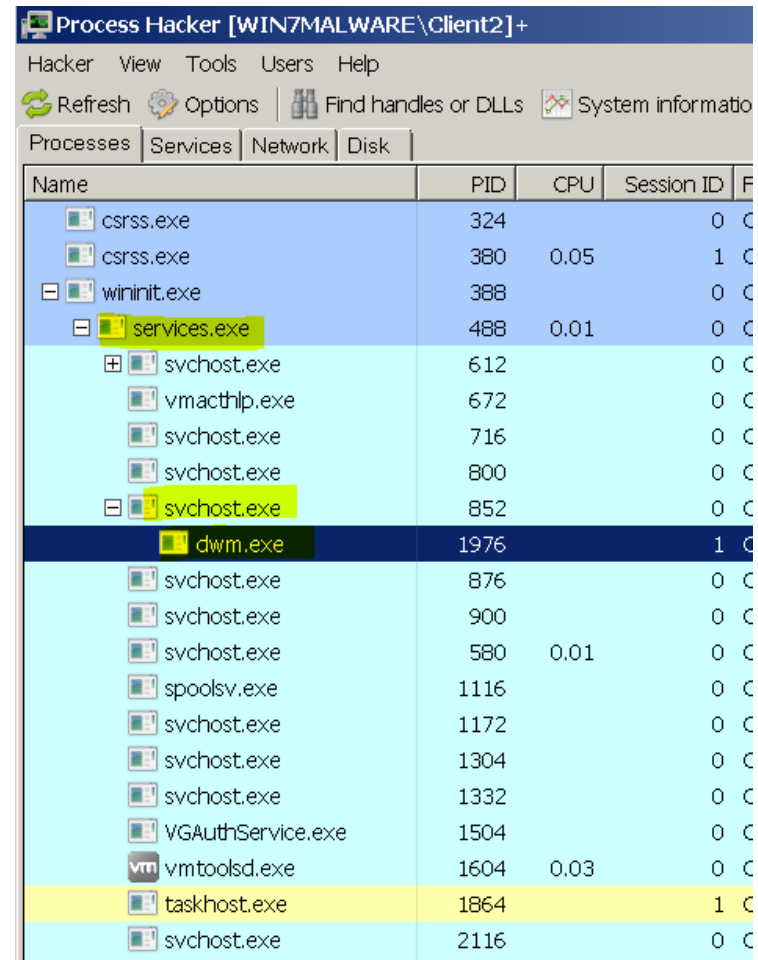


# Windows processes

- Ví dụ, giả sử chúng ta có một tiến trình muốn ghi vào một tệp. Khi đó, tiến trình sẽ yêu cầu truy cập vào tệp thông qua kernel. Kernel sẽ chịu trách nhiệm mở tệp và tạo một handle, sau đó thêm handle đó vào bảng handle để tiến trình có thể sử dụng nó để truy cập vào tệp.
- Nhờ có handle này, tiến trình có thể sử dụng các hàm hệ thống được cung cấp bởi kernel để thực hiện các hoạt động trên tệp, chẳng hạn như đọc, ghi hoặc đóng tệp. Handle đóng vai trò như một tham chiếu giữa tiến trình và đối tượng kernel, cho phép tiến trình tương tác với đối tượng đó thông qua các yêu cầu gửi tới kernel.

# Windows SERVICES

- Services là những tiến trình đặc biệt, chúng có thể chạy nền và được quản lý bởi HĐH.
- Mỗi dịch vụ đã được đăng ký có thể là một tệp thực thi (executable file) hoặc một tệp DLL (DLL file).
  - Tất cả các dịch vụ đã được đăng ký đều được chạy bởi tiến trình `services.exe` trong trường hợp của một tệp thực thi.
  - Hoặc bằng cách sử dụng tiến trình `svchost.exe`.

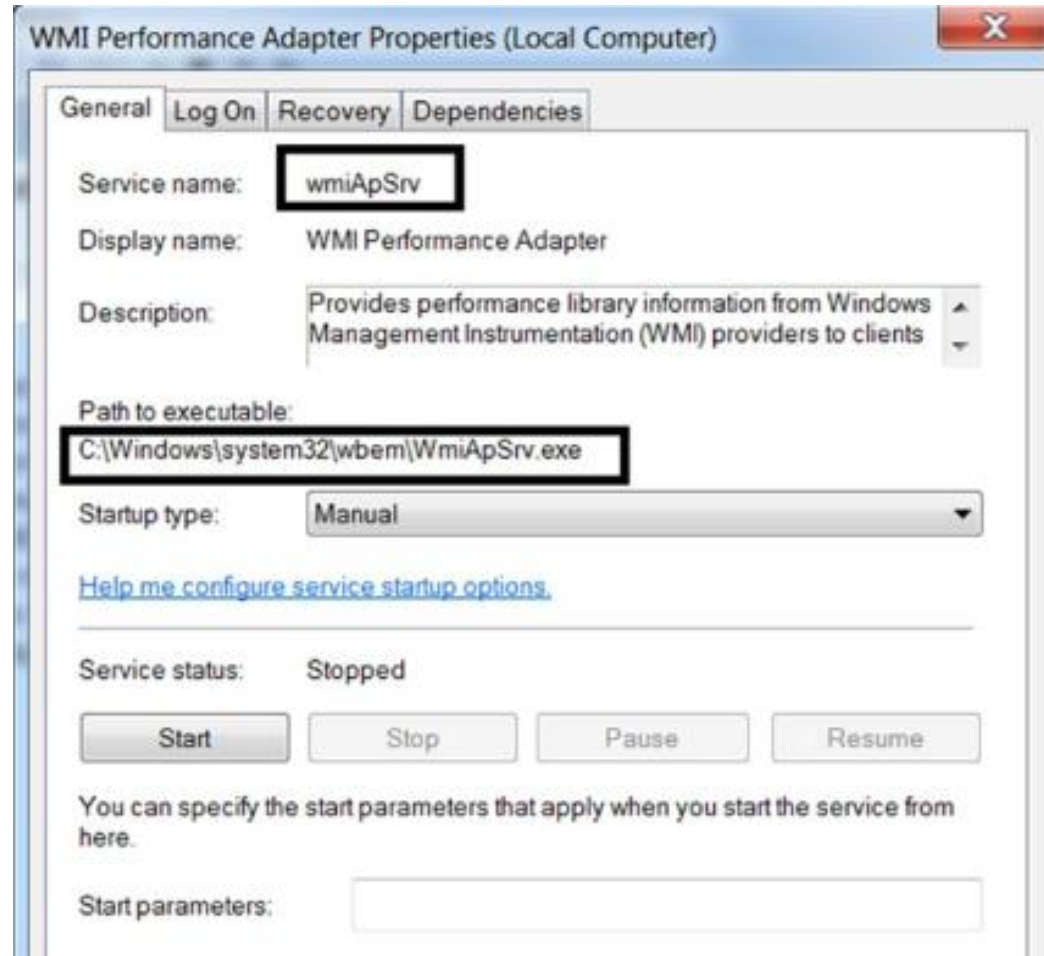


Name	PID	CPU	Session ID	Parent
csrss.exe	324		0	System
csrss.exe	380	0.05	1	System
wininit.exe	388		0	System
services.exe	488	0.01	0	System
svchost.exe	612		0	services.exe
vmacthlp.exe	672		0	System
svchost.exe	716		0	services.exe
svchost.exe	800		0	services.exe
svchost.exe	852		0	services.exe
dwm.exe	1976		1	System
svchost.exe	876		0	services.exe
svchost.exe	900		0	services.exe
svchost.exe	580	0.01	0	services.exe
spoolsv.exe	1116		0	services.exe
svchost.exe	1172		0	services.exe
svchost.exe	1304		0	services.exe
svchost.exe	1332		0	services.exe
VGAAuthService.exe	1504		0	services.exe
vmtoolsd.exe	1604	0.03	0	System
taskhost.exe	1864		1	System
svchost.exe	2116		0	services.exe

Tất cả các dịch vụ đang chạy sử dụng **svchost.exe** với tiến trình cha là **services.exe**

# Windows SERVICES

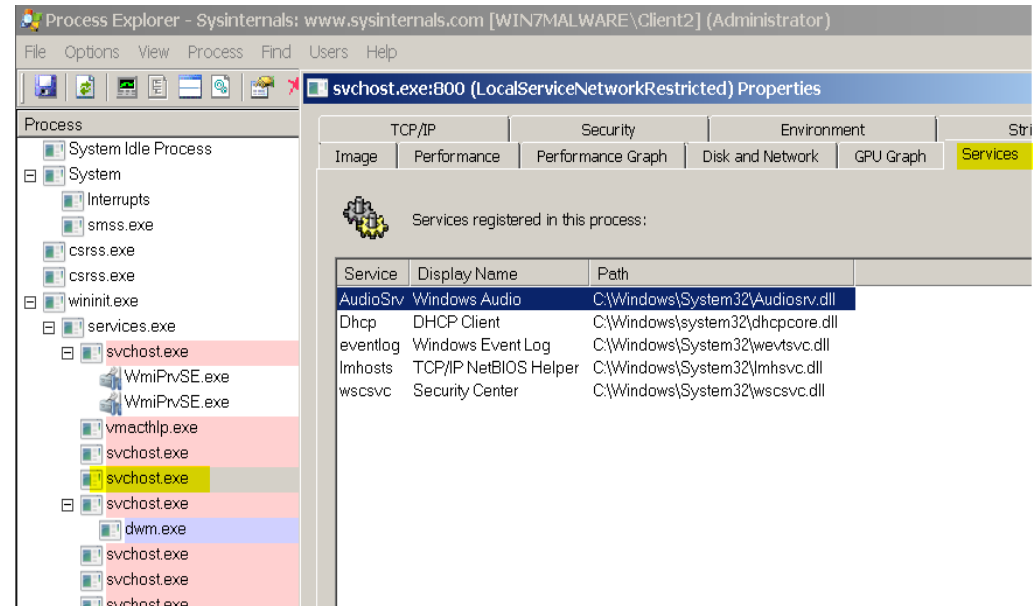
- Đối với **executable file** được đăng ký như là một dịch vụ ta sẽ thấy nó được chạy như là tiến trình con riêng của **svchost.exe**



# Windows SERVICES

- Các dịch vụ cũng có thể được chứa dưới dạng các tệp DLL trong svchost.exe.
- Nếu dịch vụ đã được đăng ký là một tệp DLL → không có quy trình con riêng được tạo dưới svchost.exe.

→ DLL dịch vụ được chạy như một phần của một phiên bản svchost.exe mới hoặc một trong các phiên bản svchost.exe hiện có, nơi nó được tải vào bộ nhớ và sử dụng một luồng để thực thi.



Danh sách các dịch vụ DLL đang được thực thi bởi tiến trình svchost.exe

# Windows SERVICES

- Mã độc thường được thấy như một dịch vụ dưới dạng executable service hoặc DLL service
- 3 loại điển hình nhất là:
  - **regsvr32.exe** command
  - **sc.exe** command
  - using Win32 APIs
- **regsvr32.exe** command và **sc.exe** command cần phải đăng ký là một dịch vụ

## Command-Line Tools to Register a Service

`sc.exe create SNAME start= auto binpath= <path_to_service_exe>`  
where, SNAME is the name of the new service

`regsvr32.exe <path_to_service_dll>`

Một số khóa registry mà hệ thống sử dụng để lưu trữ thông tin về các mục dịch vụ bao gồm:

- **HKLM\SYSTEM\CurrentControlSet\services**
- **HKLM\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce**
- **HKLM\Software\Microsoft\Windows\CurrentVersion\RunServices**

# Windows SERVICES

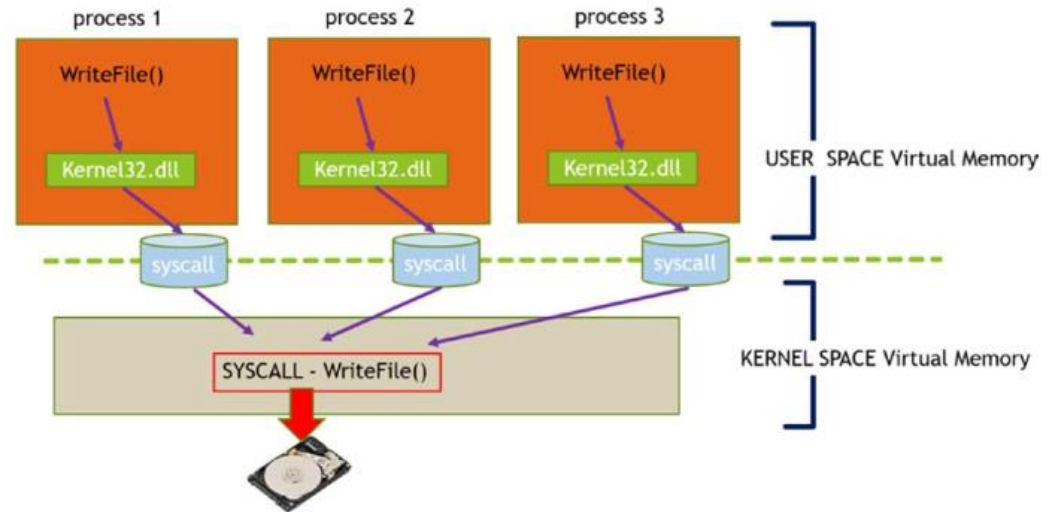
→ Khi phân tích một mẫu phần mềm độc hại, hãy xem xét xem mẫu đó có đăng ký chính nó như một dịch vụ sử dụng các lệnh như `regsvr32.exe` và `sc.exe`. Sau đó, hãy theo dõi đường dẫn của tệp thực thi hoặc tệp DLL đã được đăng ký như một dịch vụ. **Thông thường, phần mềm độc hại sẽ đăng ký các tải/tệp thực thi bổ sung dưới dạng dịch vụ.**





# SYSCALL

- Ở vùng user, các chương trình không được phép tương tác với phần cứng trực tiếp
- Để vùng user được giao tiếp với phần cứng thì kernel phải tạo ra các **syscall**.
- Các syscalls sẽ giao tiếp với các tài nguyên phần cứng thực tế thông qua các drivers một cách có kiểm soát.



Chuyển đổi từ Chế độ User sang Chế độ Kernel bằng cách sử dụng syscalls.

# SYSCALL

- Mọi thứ trong Windows đều là một đối tượng (object). Một trong số các đối tượng quan trọng đó là **Mutex** (Mutual Exclusion).
- Mutex là một đối tượng đồng bộ hóa, trong đó hai hoặc nhiều tiến trình hoặc luồng có thể đồng bộ hóa các hoạt động và thực thi của chúng.
- Để đảm bảo rằng chỉ có một instance duy nhất của tiến trình chương trình đang chạy tại một thời điểm.
- Rất nhiều phần mềm độc hại không muốn có nhiều phiên bản của chính nó được chạy, có lẽ là vì nó không muốn tái nhiễm máy tính cùng một lần nữa.
- Hãy chú ý đến mutex được tạo bằng cách xem tab **Handles** trong **Process Hacker**, ở đó nếu có mutex, nó sẽ được liệt kê như một handle.

