

Chasing Ghosts? Return of the Stealth Malware

F-SECURE[®]



Gergely Erdélyi

Gergely.Erdelyi@F-Secure.com

The logo for F-Secure, featuring the text "F-SECURE" in a bold, black, sans-serif font. Below the text is a stylized, purple and black shield-like symbol with a white 'F' shape inside. The logo is set against a circular background that appears to be a globe with a grid of latitude and longitude lines.

F-SECURE

Introduction

- What is stealth malware?
- What is the purpose of stealth malware?
- How do stealth features work?

The logo for F-Secure, featuring the text "F-SECURE" in a bold, black, sans-serif font. Below the text is a stylized, purple and black shield-like symbol with a white 'F' shape inside. The logo is set against a circular background that appears to be a globe or a grid pattern.

F-SECURE

Brain

- Brain was the first PC virus
- It was discovered in January, 1986
- Brain is a stealth boot virus
 - It hooks Int 13h
 - Hides the infected boot sector

Hiding behind complexity

- Windows is increasingly complex
- Size of Windows XP:
 - 40 million lines of code
 - around 10.000 files
 - approximately 1 gigabyte (including data)
- It contains a number of files with unclear purpose

The logo for F-Secure, featuring the text "F-SECURE" in a bold, black, sans-serif font. Below the text is a stylized, dark blue and black shield-like symbol with a white 'F' shape inside. The logo is set against a circular background that appears to be a globe or a grid pattern.

Know the components (?)

Filename: *CMS32.DLL*

File description: *'Console Messaging Subsystem Library'*

Filename: *WOW32.DLL*

File description: *'32-bit WOW Subsystem Library'*

Which one does not belong to Windows?

The logo for F-Secure, featuring the text "F-SECURE" in a bold, black, sans-serif font above a stylized, purple and black shield-like symbol. The shield has a white 'F' shape inside it.

F-SECURE

Optical tricks

Can you see the difference?

kernel32.dll

kerne132.dll



Hiding behind the file manager

- Windows Explorer does not show extensions by default
- Files with HIDDEN and/or SYSTEM attributes are not shown
- The behaviour of Explorer is controlled through registry:

```
[HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced]
```

```
Hidden, SuperHidden, HideFileExt
```

The logo for F-SECURE, featuring the text "F-SECURE" in a bold, sans-serif font above a stylized, inverted triangle graphic with a smaller triangle inside it.

Hiding processes as services (Win9x)

Win32 API call:

```
DWORD RegisterServiceProcess(  
    DWORD dwProcessId,  
    DWORD dwType);
```

dwProcessId

- ID of the process to register

dwType

- 0/1 register/deregister the process

The logo for F-SECURE, featuring the text "F-SECURE" in a bold, black, sans-serif font above a stylized, dark blue and black shield-like symbol. The shield has a white 'F' shape inside it. The entire logo is set against a circular background with a grid pattern, resembling a globe or a technical diagram.

Hiding processes as services (WinNT)

- Services are registered to Service Control Manager
- Using SCM is more complex:

```
OpenSCManager ( ) ;
```

```
CreateService ( ) ;
```

```
StartService ( ) ;
```

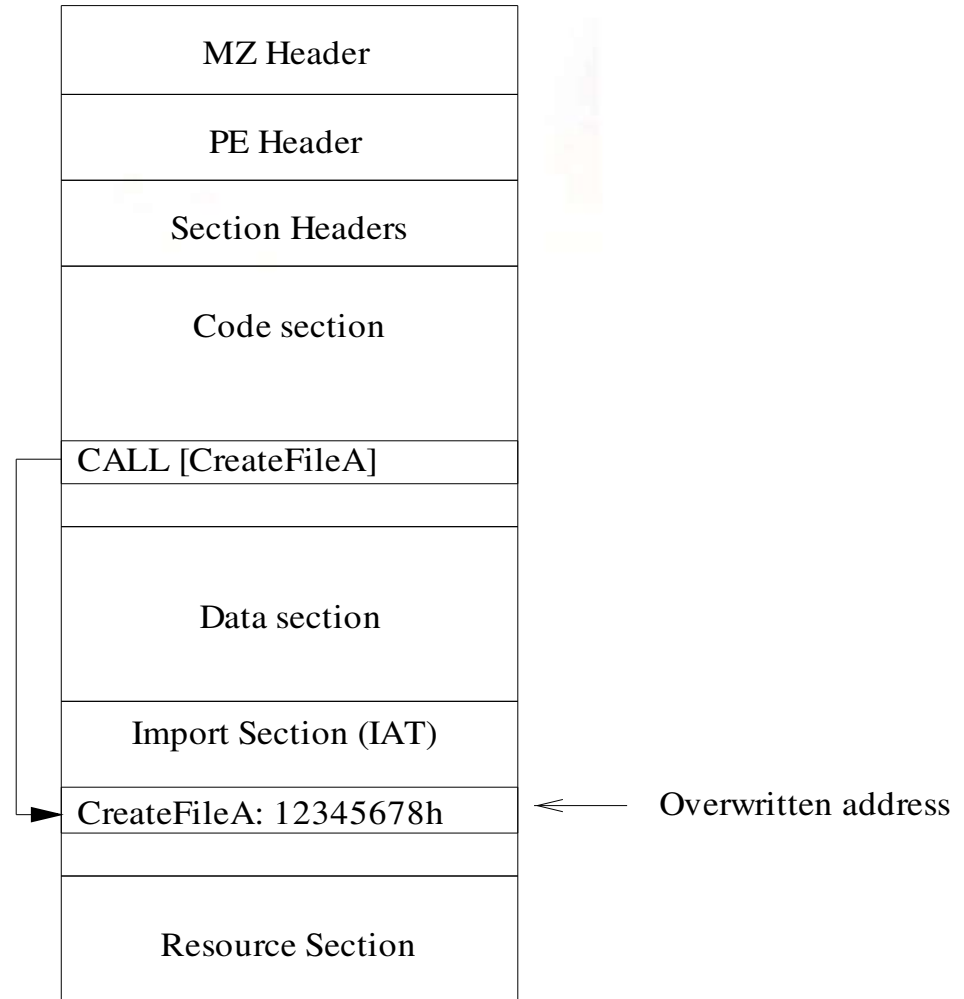
```
CloseServiceHandle ( ) ;
```

Hiding processes as services (WinNT)

- Services must register a handler to SCM:

```
DWORD WINAPI HandlerEx(  
    DWORD dwControl,  
    DWORD dwEventType,  
    LPVOID lpEventData,  
    LPVOID lpContext);
```

Import Address Table modification



Dynamic code patching (before patching)

FindNextFileA:

```
195D6: 55          PUSH  EPB
195D7: 8B EC      MOV   EBP, ESP
195D9: 81 EC 60 02 00 00  SUB   ESP, 260

195DF: 53          PUSH  EBX
195E0: 8D 85 A0 FD FF FF  LEA  EAX, [EBP-260]
```

Dynamic code patching (after patching)

FindNextFileA:

195D6: E9 78 56 34 12 JMP 12345678h

195DB: 60 02 00 00 XXX

195DF: 53 PUSH EBX

195E0: 8D 85 A0 FD FF FF LEA EAX, [EBP-260]

The logo for F-Secure, featuring a stylized 'F' inside a shield-like shape, with the text 'F-SECURE' above it.

F-SECURE

Dynamic code patching (Hook code)

```
NEW_FindNextFileA(arguments)
{
    Process_Arguments();
    Restore_First_Bytes(Hooked_Function);
    FindNextFileA();
    Alter_Data();
    Patch_First_Bytes(Hooked_Function);
}
```

Dynamic code patching with instruction analysis

FindNextFileA:

```
195D6: 55          PUSH    EPB
195D7: 8B EC      MOV     EBP, ESP
195D9: 81 EC 60 02 00 00  SUB    ESP, 260

195DF: 53          PUSH    EBX
195E0: 8D 85 A0 FD FF FF  LEA    EAX, [EBP-260]
```

Dynamic code patching with instruction analysis

FindNextFileA:

```
195D6: E9 78 56 34 12      JMP      12345678h
195DB: 90                      NOP
195DC: 90                      NOP
195DD: 90                      NOP
195DE: 90                      NOP
```

FindNextFileA_Cont:

```
195DF: 53                      PUSH    EBX
195E0: 8D 85 A0 FD FF FF      LEA    EAX, [EBP-260]
```

Dynamic code patching with instruction analysis

Original_FindNextFileA:

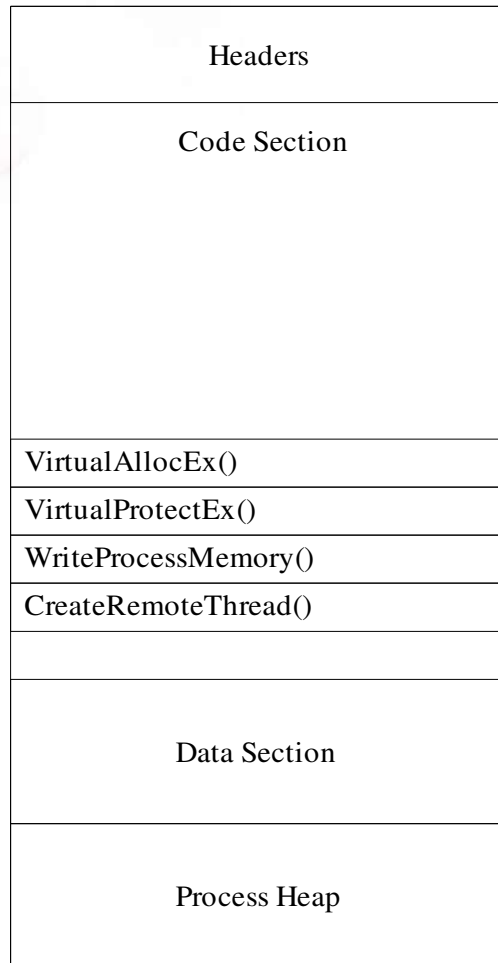
```
20000: 55                PUSH EBP
20001: 8BEC             MOV  EBP, ESP
20003: 81EC60020000    SUB  ESP, 260
20009: E9XXXXXXXX      JMP  FindNextFileA_Cont
```

HOOKED_FindNextFileA(arguments)

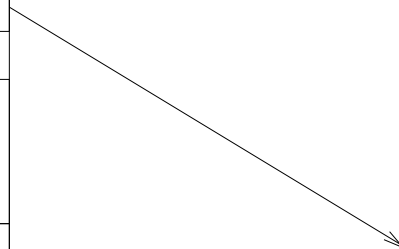
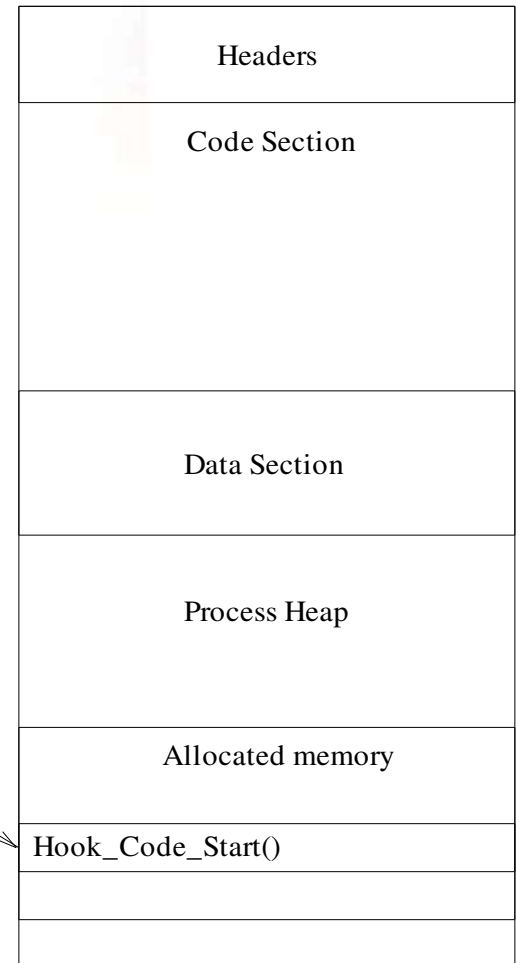
```
{
    Process_Arguments();
    Original_FindNextFileA(
    Alter_Data();
}
```

Installing the hooks (WinNT)

Attacking process



Victim



The logo for F-SECURE, featuring the text "F-SECURE" in a bold, black, sans-serif font above a stylized, black-outlined letter "F" that is filled with a purple-to-blue gradient. The logo is set against a circular background with a grid pattern, resembling a globe.

F-SECURE

Installing the hooks (WinNT)

```
LPVOID VirtualAllocEx(  
    HANDLE hProcess,  
    LPVOID lpAddress,  
    SIZE_T dwSize,  
    DWORD flAllocationType,  
    DWORD flProtect);
```

```
VirtualProtectEx();  
WriteProcessMemory();  
CreateRemoteThread();
```



DLL injection

- The hooks and install routine is placed in a DLL
- The attacker injects a `LoadLibrary("Nasty.dll")` call
- Using `CreateRemoteThread()` the code is executed
- The system loads the DLL and calls `DIIMain()` in it
- `DIIMain()` installs the hooks
- `Nasty.dll` is active in the remote process and monitors it

Direct memory writing

- The attacker uses `VirtualAllocEx()` to allocate memory
- The hooks and installer is copied using `WriteProcessMemory()`
- The installer is started with `CreateRemoteThread()`
- The injected code must be position independent

The logo for F-Secure, featuring the text "F-SECURE" in a bold, black, sans-serif font. Below the text is a stylized, purple and black shield-like symbol with a white 'F' shape inside. The logo is set against a circular background that looks like a globe with latitude and longitude lines.

F-SECURE

Kernel space hooks

- Code running in kernel space has more control
- Kernel space hooks are more difficult to detect
- Writing kernel code is harder
- Any mistake can cause total system failure
- Kernel code is highly OS version dependent

File system hooks (Windows 9x/ME)

```
IFSMgr_InstallFileSystemApiHook(  
    pIFSFileHookFunc HookFunc  
);
```

```
FileSystemApiHookFunction(  
    pIFSFunc FSDFnAddr,  
    int FunctionNum, →  
    int Drive,  
    int ResourceFlags,  
    int CodePage,  
    pioreq pir  
);
```

IFSFN_OPEN
IFSFN_CLOSE
IFSFN_READ
IFSFN_WRITE
IFSFN_SEEK
IFSFN_SEARCH
IFSFN_ENUMHANDLE
IFSFN_FINDOPEN
IFSFN_FINDNEXT

The logo for F-Secure, featuring the text "F-SECURE" in a bold, black, sans-serif font. Below the text is a stylized, dark blue and black shield-like symbol with a white 'F' shape inside. The logo is set against a circular background with a grid pattern, resembling a globe or a technical diagram.

Registry and other API hooks (Windows 9x/ME)

- Using Virtual Machine Manager (VMM)
- `Hook_Device_Service()` installs hooks
- Device Driver Kit (DDK) has the headers for most common services (e.g. registry)

Installing kernel space hooks (Windows 9x/ME)

- Loading a device driver (VxD)
 - Hooks are placed to a VxD file
 - VxD can be loaded with `CreateFile("\\\\.\\hook.vxd")`
 - VxD stays loaded until next restart
- Ring3 to Ring0 jump
 - Works by modifying the Interrupt Descriptor Table
 - Executes code in kernel space
 - Used by the Zerg and Sma viruses for example

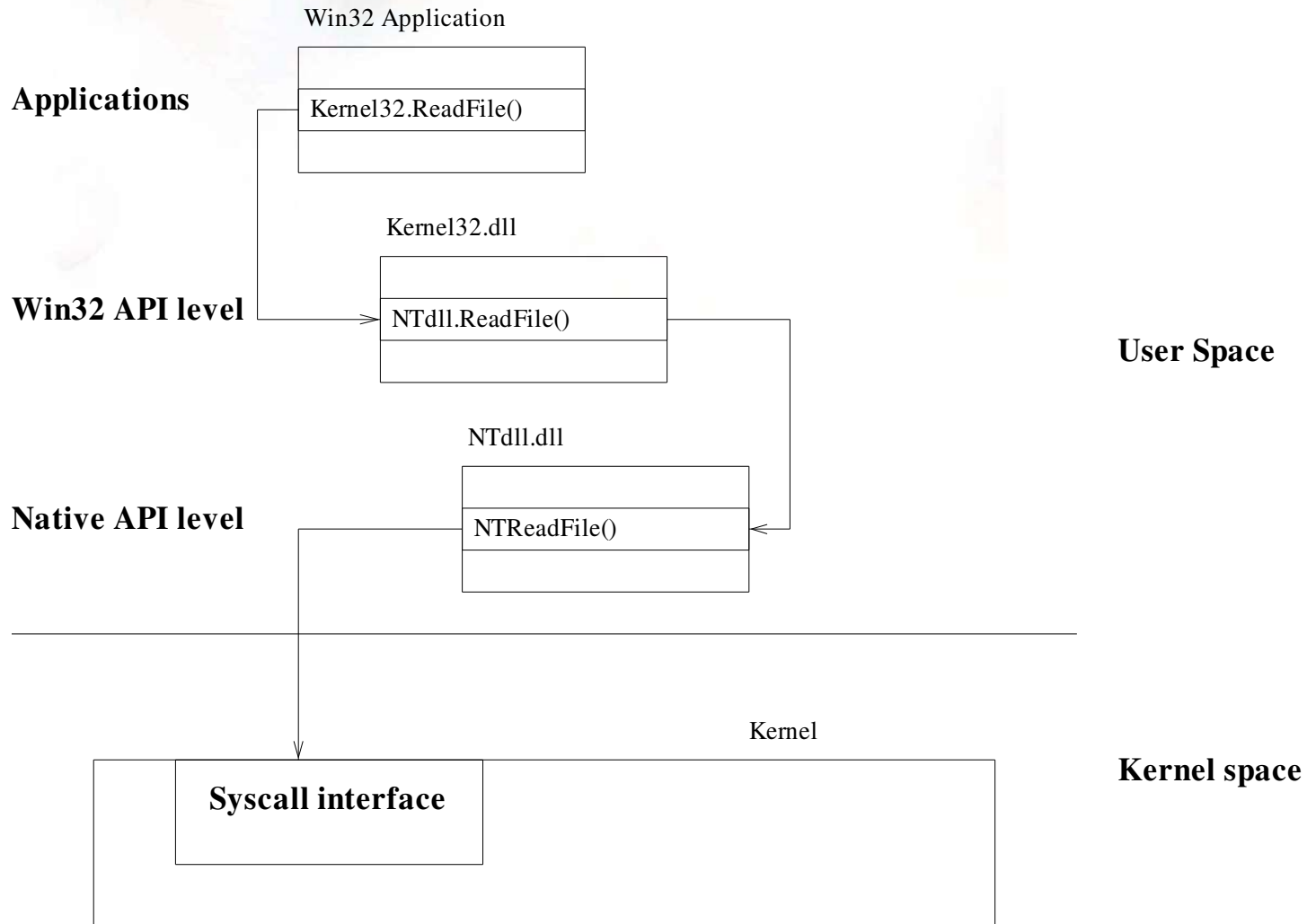
The logo for F-Secure, featuring the text "F-SECURE" in a bold, black, sans-serif font. Below the text is a stylized, three-dimensional shield or triangle shape composed of overlapping blue and black geometric forms. The entire logo is set against a circular background with a grid pattern, resembling a globe or a technical diagram.

Kernel space hooks in Windows NT/2k/XP

- NT is based on a microkernel architecture
- On the top of the kernel several user subsystems are running (Win32, OS/2, Posix)
- The subsystems use the Native NT API to communicate with the kernel
- Hooks in the Native NT API provide global control



A Win32 API call

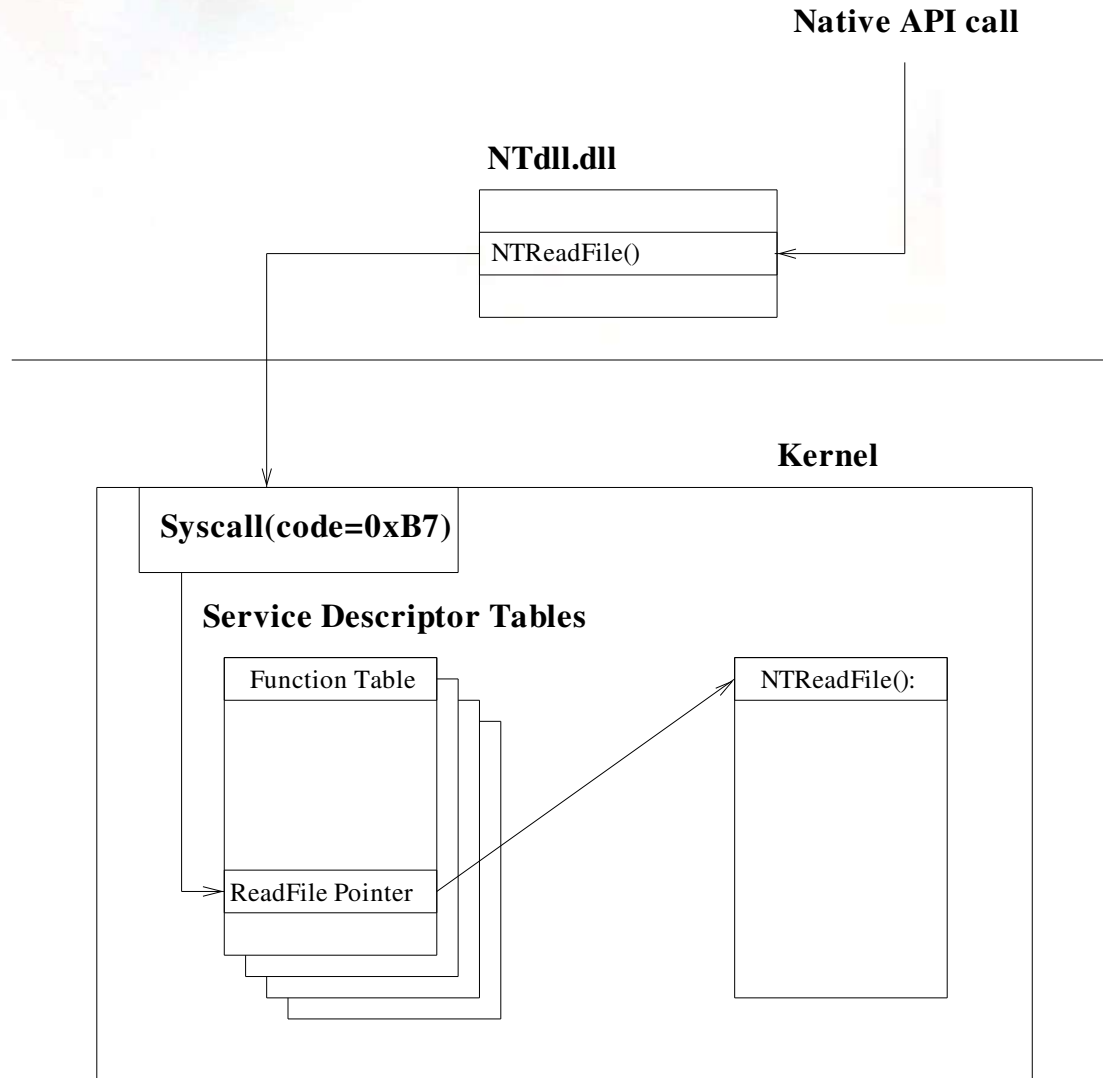


Altering the Kernel Service Table

- Entry points to system calls are stored in the System Service Descriptor Table (SSDT)
- SSDT stores services in four groups:
 - 0 - Core services (exported from NTDLL.DLL)
 - 1 - GUI services
 - 2 - Reserved
 - 3 - Reserved
- SSDT is write protected in Windows XP but that can be circumvented by disabling the processor's WP bit



NT System Call



The logo for F-Secure, featuring the text "F-SECURE" in a bold, black, sans-serif font. Below the text is a stylized shield or triangle shape composed of several overlapping, nested shapes in shades of purple and black. The logo is set against a circular background that resembles a globe with latitude and longitude lines.

Installable File System

- NT file system drivers can be created using the Installable File System API (IFS)
- IFS kit is available from Microsoft
- Through IFS It is possible to supervise all file operations
- IFS filters can be layered on top of each other
- Antivirus applications use IFS too

The logo for F-SECURE, featuring the text "F-SECURE" in a bold, black, sans-serif font above a stylized, dark blue and black shield-like symbol. The shield has a white outline and a dark blue interior with a white 'F' shape. The logo is set against a circular background with a grid pattern.

F-SECURE®

Direct kernel object modification

- Requires knowledge on undocumented kernel internals
- Fu backdoor uses this method for different purposes:
 - Hiding processes
 - Adjust privileges
- This approach is highly OS version dependent and error prone

Installing the hooks

- Hooks can be added as standard device drivers
 - With `CreateService()` using `SERVICE_KERNEL_DRIVER` flag
 - Drivers are loaded/unloaded automatically by the system
- Using `NtSetSystemInformation()`
 - The function `SystemLoadAndCallImage` is undocumented
 - It does not need registering to Service Control Manager
 - Loads and starts a driver in a running system

The logo for F-Secure, featuring the text "F-SECURE" in a bold, black, sans-serif font. Below the text is a stylized, purple and black shield-like symbol with a white 'F' shape inside. The logo is set against a circular background that looks like a globe with latitude and longitude lines.

F-SECURE®

Real World Hooks

- Not all programs using these techniques are malicious
- Malware (viruses, backdoors, etc) do misuse them
- Things they try to hide:
 - Files, directories
 - Processes
 - Registry keys, values
 - Services
 - Open network ports

Examples of stealth malware

- HxDef (Hacker Defender) which hides:
 - Processes
 - Services
 - Registry keys, values
 - Files, directories
 - Network traffic
- HE4Hook package (available in source code) hides:
 - Processes
 - Files, directories

The logo for F-Secure, featuring the text "F-SECURE" in a bold, black, sans-serif font. Below the text is a stylized, dark blue and black shield-like symbol with a white 'F' shape inside. The logo is set against a circular background with a grid pattern, resembling a globe.

Detection of stealth malware

- User space stealth can be detected with kernel based scanner code
- Clean booting
- Detection of malware's communication channel
 - Mail slot
 - Socket
 - Other IPC mechanism
- Detection of symptoms not hidden by the malware

The logo for F-Secure, featuring the text "F-SECURE" in a bold, black, sans-serif font. Below the text is a stylized, purple and black shield-like symbol with a white outline, set against a circular background that resembles a globe with latitude and longitude lines.

Conclusion

- Stealth code for Windows is reality
- It is becoming more and more common
- Most often it's used in backdoors/rootkits
- Traditional hacking/cracking activities drive the development of stealth techniques
- We can expect more tricky solutions day by day

F-SECURE®



The Hide'n'Seek has begun...



The logo for F-Secure, featuring the text "F-SECURE" in a bold, black, sans-serif font. To the right of the text is a small registered trademark symbol (®). Below the text is a stylized, geometric logo consisting of a large, dark blue triangle pointing downwards, with a smaller, lighter blue triangle nested inside it, creating a sense of depth and movement.

F-SECURE®

Questions?