The Evolution of Windows Spyware Techniques

Birdman
birdman@chroot.org
birdman@xsolve.com
birdman@gmail.com
HIT2005

Welcome!

- Hello everyone, This is Birdman.

- WARNING - Contents of this presentation are for "Educational Purposes ONLY". It is strongly suggested that you do not use this knowledge for illegal purposes!.......plz 😊
1. What is Spyware?

- Too Many Fake Jargons !?
  - In many newspapers, magazines or reports, you must have heard about the following terms:
    - Joke, Logic Bomb, Trojan, Backdoor, Worm, Dropper, Germ, Intended, Malware, Riskware, Spyware, Adware, Ghostware, Keylogger, Rookit, Harmful Program.
  - But What’s It !?
    - Don’t care about those dazed words !! Because it is very difficult to make proper definitions, they are just advertisement words.
Evil Level of Malware

Virtuous

Normal Programs

Virus

Trojan

Backdoor

Worm

Rootkit

Evil

My Taxonomy of Malware

- Klaus Brunnstein
  - He writes about the difficulties of defining Malware. He regards the traditional definitions as self-contradicting and not exhaustive. Therefore he proposes a new way of defining the term, which he calls intentionally dysfunctional software. His definition is meant to distinguish normal dysfunctionalities from intentionally malevolent ones

- Spyware are not products, It are just functions!
  - Rootkit, Backdoor, Adware, Keylogger and Password Dummper ... all of them are features of Malware
2. The Techniques In The Past

- Famous Malware
3. The Spyware of Nowadays

- Connect-back Backdoor
- Portless Spyware
- DLL-Based Spyware
- Spyware + Rootkit

Portless Spyware - RawSocket

- Raw-Socket Backdoor
  - A raw socket is one that allows access to the underlying transport protocol.
  - Raw socket use “Device\RawIp” and normal socket use “Device\Tcp” or “Device\Udp.”
    Therefore, they have no any ports!
  - Local Sniffer: Use WSAIoctl to set SIO_RCVALL
  - Famous Backdoor
    - Ackcmd
      - HTTP TCP(Ack) tunneling
    - hKdoor
Portless Spyware - LSP

- LSP Backdoor (SPI Backdoor)
  - LSP = Layered Service Provider
  - Registry
    - `System\CurrentControlSet\Services\WinSock2\Parameters\Protocol_Catalog9\Catalog_Entries`

DLL-Based Spyware

- As our observation, DLL-based Spyware are popular among the Spyware Coder.
  1. It resides in processes, thus it can bypass many scanning (including the personal firewall).
  2. Everyone watch the Process and EXE-file, but no one care about DLLs.
  3. Up now on, there are no effective Anti-Virus or Anti-Hacking tools to against them !!!

- Install Component
  - ActiveX, LSP …
- DLL Injection
- Replacement System DLL (Proxy DLL)
Rootkit

- RootKits are hacker tools that modify existing operating system software so that an attacker can gain access to and hide on a machine.
- This rootkit patches Windows API to hide certain objects from being listed.
  1. Processes
  2. Handles
  3. Modules
  4. Files & Folders
  5. Registry Values
  6. Services
  7. TCP/UDP Sockets

4 Stealth Tricks

- Hooking
- Non-Hooking
4-1 Stealth With Hooking

- The Hooking Techniques are the most important stealth tricks, these tricks are also the popular among the Hackers.

- What is Hooking?
  - Hooking = Execution Path Change

- Types of Hooking
  - Function Pointer Change
  - Raw-Code Change

The API Calling Path

- Int 2E/SYSENTER
- Ntoskrnl.exe
- KiSystemServiceTable
- Ntoskrnl.exe
- NtCreateFile

CreateFile(...)
Call [00456189]

Address | Data
--|---
00456189 | BFFF9321
00456193 | .......

The API Entry

The IAT

Call API

KiSystemService

Ntoskrnl.exe

NtCreateFile

Ntdll.dll

KSystemService

Reference

Int 2E/SYSENTER

Call API

Application

IAT

CreateFile(...)
Call [00456189]
**Hooking Type**

- **Function Pointer Change**
  - IAT Modification
  - EAT Modification
  - SDT Modification

- **Raw-Code Change**
  - Calls to the target function are replaced with calls to the malicious code by modifying application binaries.
  - Insert JMP
  - Insert CALL

- **Breakpoint Trapping**
  - Insert INT3

---

**Performance Issue**

<table>
<thead>
<tr>
<th>Interception Technique</th>
<th>Intercepted Function</th>
<th>Time (microseconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Empty Function</td>
<td></td>
</tr>
<tr>
<td>Direct</td>
<td></td>
<td>0.113 µs</td>
</tr>
<tr>
<td>Call Replacement</td>
<td></td>
<td>0.143 µs</td>
</tr>
<tr>
<td>DLL Redirection</td>
<td></td>
<td>0.143 µs</td>
</tr>
<tr>
<td>Detours Library</td>
<td></td>
<td>0.145 µs</td>
</tr>
<tr>
<td>Breakpoint Trap</td>
<td></td>
<td>229.564 µs</td>
</tr>
<tr>
<td></td>
<td>CoCreate-Instance</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>14.836 µs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15.193 µs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15.194 µs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>265.851 µs</td>
</tr>
</tbody>
</table>

- **Detours: Binary Interception of Win32 Functions**
The Well-Known Ways for Hooking API

1. Replacing Files (DLLs)  
2. Hooking IAT  
3. Patching API Entry  
4. Hook Export Directory  
5. Hooking IDT 2Eh Entry  
6. Hooking KiSystemService  
7. Hooking SDT  
8. Hooking SST (KiServiceTable)  
9. Hook NativeAPI Export_Directory  
10. Patching NativeAPI Entry

User Mode

Kernel Mode

The Flow Path After “Patching API”

Application

 Kernel32.DLL

API Entry

BFFF9321 : API Entry

Jmp my_stub

Call API

My stub

Call [00456189]
4-1-1 Process Hiding

- Intruders are interested in staying invisible, they always use such functionality to cover their other spyware. Therefore, almost every rootkit provides such stealth trick.

- API-Hooking
  - ToolHelp API
  - PS API
  - Performance API
  - WMI API
  - Native API
    - ZwQuerySystemInformation

- DKOM
  - DKOM:EPROCESS:ActiveProcessLinks

4-1-2 TCP/UDP Port Hiding

- For hiding the port, we have many methods to do that:
  1. By SNMP Functions (such as netstat)
  2. By Query TCP Handles (such as FPort, Arbiter)
- There is an example, which will hide the certain “Port” by hooking SDT. It control a Native API, ZwDeviceIOPControlFile.
Hook It~

- Therefore, we could break in them!
  - Hook IpHelper APIs
    - GetTcpTable
    - AllocateAndGetTcpTableFromStack
    - AllocateAndGetUdpTableFromStack
    - AllocateAndGetTcpExTableFromStack (New for WinXP)
    - AllocateAndGetUdpExTableFromStack (New for WinXP)
  
  ➢ Hook DeviceIOControl API
    - IOCTL_TCP_QUERY_INFORMATION
    - IOCTL_TCP_QUERY_INFORMATION_EX (New for WinXP)

4-1-3e Registry Hiding

- Win32 API
  - RegEnumKeyA/W
  - RegEnumKeyExA/W
  - RegEnumValueA/W
  - RegQueryMultipleValuesA/W

- Native API
  - ZwEnumerateKey
  - ZwEnumerateValueKey
4-1-4 File/Directory Hiding

- **Win32 API**
  - FindFirstFileA/W, FindNextFileA/W
- **Native API**
  - ZwQueryDirectoryFile

4-1-5 Service Hiding

- **Advapi32.dll**
  - EnumServicesStatusA
4-2 Stealth With No-Hooking

- Recently, No-Hooking tricks are more and more popular, because there are many mature ways to detect Hooking.
- The ultimate stealth is nothing to hide!
- DKOM
  - Direct Kernel Object Manipulation
  - Jamie Butler, http://www.hbgary.com

Fu rootkit

All active processes in the system are kept on the single list. This list is implemented by pair of pointers in each EPROCESS block:
- Win2K:EPROCESS.ActiveProcessLinks (offset +0xa0)
- WinXP:EPROCESS.ActiveProcessLinks (offset +0x88)
Interesting Stealth Techniques

- Zero Registry Spyware
- Stealth Module Trick
- Code Injection
  - Shellcode Injection
  - DLL Injection
- Playing PE Loader

Zero-Registry Spyware

- There is a new popular trick to make Spyware become more stealth. Some DLL-based Spyware replace system service DLL, therefore they don’t modify any registry. It is difficult to discover them!
  - Packetdoor
    - Stop Auto-update service
    - Replace wuauserv.dll with packetdoor’s dll
    - Start Auto-update service
  - BDR.UC.Backdoor
Stealth Module Trick

- As soon as it is loaded into a process, the Rootkit hides its DLL. Rootkit modify the PEB_LDR_DATA (PEB=FS:0x30) to unlink
  - InLoadOrderModuleList, InMemoryOrderModuleList, InInitializationOrderModuleList
  - The technique used below is very efficient against all programs that rely on the windows API for enumerating modules. Due to the fact that EnumProcessModules/Module32First/Module32Next/... depend on NtQuerySystemInformation
  - Rootkit : vanquish-0.2.0

```
FS:30
PEB
Ntdll.dll
kernel32
Spyware
```

Code Injection

- DLL Injection (Win2K/XP)
  1. Open the target process.
  2. Prepare the "Inject-code" and "Inject-data" in our local process.
  3. Allocate memory in the remote process address space.
  4. Change the page permission of the allocated memory.
  5. Write a copy of our inject-code and a inject-data to the remote process.
  6. Create a thread in the remote process to invoke our inject-code.
**Playing PE Loader**

- There is a variation of DLL-Injection. It could make the DLL become invisible. I show you:

```
Spyware Loader

OpenProcess() -> Target Process
VirtualAllocEx() -> Trojan.dll
CreateRemoteThread() -> C:\Trojan.dll

Copy Image
LoadLibrary() -> Trojan.dll
FreeLibrary
Restore Image
```

```
Spyware Loader

OpenProcess() -> Target Process
VirtualAllocEx() -> Trojan.dll
CreateRemoteThread() -> C:\Trojan.dll

Copy Image
LoadLibrary() -> Trojan.dll
FreeLibrary
Restore Image
```
5. Spyware Analysis and Detection Techniques

- Detect Hidden Processes
- Detect Hidden Files
- Detect Hidden Registry

Tools: Procexp
Tools: Rootkit Revealer


Tools: Blacklight

- F-Secure
Tools: IceSword

Tools: Archon
**Anti-Rootkit Tools**

- KAV
- Rootkit-Revealer
- Blacklight
- IceSword
- Archon
- VICE

- **How about …**
  - Pc-cillin, Norton, CA, Spy Sweeper …

**Demonstration**

1. **Backdoor**
   1. Keylogger
   2. SPI Backdoor

2. **Rootkit + Backdoor**
   1. Hxdef 1.0 + BirdSPY4
   2. Pro-Agent
   3. Hidden Process (FU like)
   4. AFX2005

3. **Stealth Module Backdoor**
   1. BdrUCB
   2. Keylogger + Vanquish
   3. ByShell

4. **Is Adware just a Adware !?**
   1. 中國人的好幫手？ – 3721
   2. Adware-Example2
6. Conclusion

- Trend of Spyware
  - Spyware is rootkitlized !!
  - DLL-based Spyware is difficult to detect.
  - No effective Anti-Spyware tools could fright rootkit.
  - DKOM and Physical Memory Usage techniques are more popular among Rootkit.
  - EXE In-Process-Execution

User Mode Rootkit become more popular.
Kernel Mode Rootkit become more powerful.
Last Words

- I'd like to emphasize that I am not responsible for anyone using that sample code with his/her homemade Trojan to leech porn from his friend's PC. Seriously, this is just a sample for educational purposes, it should not be used for any kind of illegal purpose.

7. Reference

- Thx Rootkit Guru :D
  - Greg Hoglund
  - Jamie Butler
  - fuzen_op
  - Joanna Rutkowska
  - Chew Keong
- Books
  - Windows95 System Programming Secrets, Matt Pietrek
  - Systems Programming for Windows95, Walter Oney
  - Programming Applications - Fourth Edition, Jeffrey Richter
  - Windows Internals 4th, David A. Solomon & Mark E. Russionovich
  - Undocumented NT, Prasad Dabak, Milind Borate & Sandeep Phadke
  - Undocumented Windows 2000 Secrets, Sven B. Schreiber
  - Windows NT/2000 Native API Reference, Gary Nebbett
- Articles & Codes
  - Win32API Spying Techniques, Yariv Kaplan
  - Tracing NT Kernel-Mode Calls, Dmitri Leman
  - Detours SDK package, Microsoft Research
  - ForceLibrary 1.4, yoda
  - APIHooks 5.5, EiICZ