Mã độc

Chương 6. Phân tích một số cơ chế và hành vi thông thường của mã độc

Mục tiêu

- Giới thiệu một số cơ chế hoạt động thường gặp của mã độc
- Phân tích một số cơ chế hoạt động thường gặp của 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

Nội dung

1. Downloader
2. Backdoor
3. Công cụ đánh cắp thông tin
4. Các cơ chế duy trì hiện diện
5. Leo thang đặc quyền
6. Các kỹ thuật trong Rootkit

Nội dung

7. Launcher
8. Tiêm vào tiến trình
9. Thay thế tiến trình
10. Tiêm vào hook
11. Detour
12. Tiêm vào APC

Nội dung

1. Downloader
2. Backdoor
3. Công cụ đánh cắp thông tin
4. Các cơ chế duy trì hiện diện
5. Leo thang đặc quyền
6. Các kỹ thuật trong Rootkit

Downloader

- Tải một chương trình độc hại khác về.
- Che giấu nó trước các anti virus.
- Thường sử dụng windows API *URLDownloadtoFileA* theo sau đó là một lời gọi đến *WinExec*

7

Loaders

Chuẩn bị một phần mềm độc hại khác và thực thi nó một cách bí mật

- Có thể thực thi ngay lập tức hoặc lúc nào đó
- Lưu trữ mã độc ở những nơi không mong muốn, chẳng hạn như .rsrc section của PE file

8

Nội dung

1. Downloader
2. Backdoor
3. Công cụ đánh cắp thông tin
4. Các cơ chế duy trì hiện diện
5. Leo thang đặc quyền
6. Các kỹ thuật trong Rootkit

9

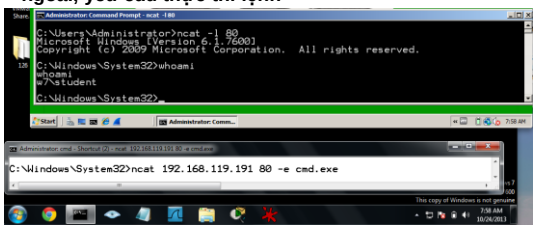
Backdoor

- Tạo một truy cập từ xa đến máy nạn nhân
- Đây là loại mã độc phổ biến nhất
- Thông thường mã độc này nhắm đến cổng 80
- Khả năng chung: Thao tác trên Registry, liệt kê các cửa sổ hiển thị, tạo các thư mục, tìm kiếm file,...

10

Reverse Shell

- Máy tính bị lây nhiễm sẽ gọi kẻ tấn công từ bên ngoài, yêu cầu thực thi lệnh



11

Windows Reverse Shells

Hành động cơ bản

- Gọi CreateProcess và thao tác với cấu trúc STARTUPINFO
- Tạo socket cho điều khiển máy từ xa
- Sau đó gắn socket với standard input, output, và error cho cmd.exe
- CreateProcess chạy cmd.exe với cửa sổ bị chặn lại để ẩn nó

12

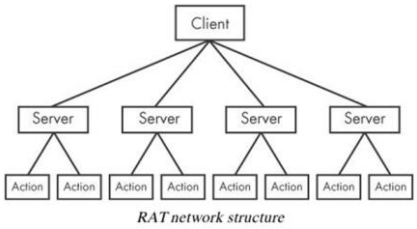
Windows Reverse Shells

- ❑ Đa luồng – Multithreaded
- ❑ Tạo một socket, hai pipe và hai thread
- ❑ Tìm các lời gọi API đến CreateThread và CreatePipe
- ❑ Một thread cho stdin, một cho stdout

13

RATs (Remote Administration Tools)

- ❑ Ví dụ Poison Ivy

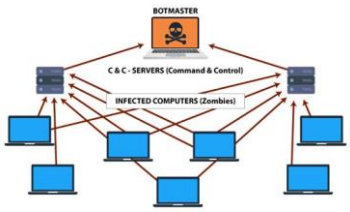


RAT network structure

14

Botnets

- ❑ Một tập hợp các máy bị nhiễm mã độc
  - ▪ Được gọi là bots hoặc zombies



15

Botnets v. RATs

- ❑ Botnet gồm nhiều máy, RATs kiểm soát ít máy hơn
- ❑ Tất cả bots/zombies đều được kiểm soát cùng một lúc; RATs điều khiển từng nạn nhân một
- ❑ RATs dành cho các cuộc tấn công có mục tiêu rõ ràng; Botnets được dùng cho các cuộc tấn công nhắm đến nhiều đối tượng chung: DDOS, Spam,...

16

Nội dung

1. Downloader
2. Backdoor
3. Công cụ đánh cắp thông tin
4. Các cơ chế duy trì hiện diện
5. Leo thang đặc quyền
6. Các kỹ thuật trong Rootkit

17

Đánh cắp thông tin

Ba cách:

- ❑ Tấn công vào cơ chế đăng nhập và lấy cắp thông tin
- ❑ Dump dữ liệu lưu trữ, chẳng hạn như Passwprd hashes
- ❑ Lưu lại các thao tác gõ phím của victim (Keylloger)

18

GINA Interception

- Nhận dạng và xác thực của bên thứ ba (GINA)
  - Cho phép bên thứ ba tùy chỉnh tiến trình đăng nhập cho RFID hoặc thẻ thông minh
  - Mã độc có thể chặn bắt thông tin gửi đến tiến trình xác thực để đánh cắp thông tin
- GINA có trong msgina.dll
  - Được load bởi Winlogon thực thi trong quá trình đăng nhập
- WinLogon cũng load những tùy chỉnh của bên thứ ba trong DLLs load giữa WinLogon và GINA.

19

GINA Registry Key

- HKLM\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\Winlogon\GinaDLL
- Chứa các DLL bên thứ ba được WinLogon nạp

Malicious fsgina.dll sits in between the Windows system files to capture data.

20

MITM Attack

Mã độc phải export tất cả các hàm trong msgina.dll, hoạt động như một MITM

- Có hơn 15 hàm, hầu hết đều bắt đầu với wlx
- Mã độc export rất nhiều hàm wlx, có thể để chặn bắt
- Là dấu hiệu nhận biết

21

WlxLoggedOutSAS

- Hầu hết các export chỉ đơn giản là gọi qua các hàm trong msgina.dll
- Tại (2) mã độc logs lại các thông tin vào:%SystemRoot%\system32\drivers\tcpudp.sys

GINA DLL WlxLoggedOutSAS export function for logging stolen credentials

```
100014A0 WlxLoggedOutSAS
100014A0         push    esi
100014A1         push    edi
100014A2         push    offset aWlxloggedout_0 ; "WlxLoggedOutSAS"
100014A7         call    Call_msgina_dll_function 1
...
100014FB         push    eax ; Args
100014FC         push    offset aUSDSPSOpS ;"U: %s D: %s P: %s OP: %s"
10001501         push    offset aDRIVERS ; "drivers\tcpudp.sys"
10001503         call    Log_To_File 2
```

22

Hash Dumping

- Mật khẩu đăng nhập được lưu trữ dưới dạng LM hoặc NTLM hashes
  - Hashes có thể được sử dụng trực tiếp để xác thực (pass-the-hash-attack)
  - Hoặc cracked offline để tìm password
- Pwdump and Pass-the-Hash Toolkit
  - Công cụ hacking miễn phí cung cấp việc hash dumping
  - Mã nguồn mở
  - Được sử dụng lại trong nhiều mã độc, sửa đổi để bypass qua anti-virus

23

Pwdump

- Injects một DLL vào LSASS (Local Security Authority Subsystem Service)
- Get hashes từ SAM (Security Account Manager)
- Inject DLL chạy bên trong những tiến trình khác
- Lấy tất cả các quyền của tiến trình
- LSASS là mục tiêu phổ biến
  - Đặc quyền cao
  - Truy cập vào nhiều API hữu ích

24

Pwdump

- Injects lsaext.dll vào lsass.exe
  - Gọi hàm GetHash, export của lsaext.dll
  - Hash extraction, Sử dụng những hàm không cung cấp tài liệu của windows
- Kẻ tấn công có thể thay đổi tên của hàm GetHash

25

Pwdump

- Sử dụng các thư viện:
  - *samsrv.dll* để truy cập vào SAM
  - advapi32.dll để truy cập vào các hàm chưa được imported vào lsass.exe
  - Hashes trích xuất bởi SamIGetPrivateData
  - Giải mã với SystemFunction025 và SystemFunction027
- Tất cả các hàm đều không có tài liệu nào nói về nó

26

Pwdump

Unique API calls used by a pwdump variant's export function GrabHash

```
1000123F        push    offset LibFileName      ; "samsrv.dll" 1
10001244        call    esi ; LoadLibraryA
10001248        push    offset aAdvapi32_dll_0  ; "advapi32.dll" 2
...
10001251        call    esi ; LoadLibraryA
...
1000125B        push    offset ProcName         ; "SamIConnect"
10001260        push    ebx                     ; hModule
10001265        call    esi ; GetProcAddress
...
10001281        push    offset aSamrqu ; "SamrQueryInformationUser"
10001286        push    ebx                     ; hModule
1000128C        call    esi ; GetProcAddress
...
100012C2        push    offset aSamigetpriv ; "SamIGetPrivateData"
100012C7        push    ebx                     ; hModule
100012CD        call    esi ; GetProcAddress
...
100012CF        push    offset aSystemfunctl  ; "SystemFunction025" 3
100012D4        push    edi                     ; hModule
100012DA        call    esi ; GetProcAddress
100012DC        push    offset aSystemfunl_0  ; "SystemFunction027" 4
100012E1        push    edi                     ; hModule
100012E7        call    esi ; GetProcAddress
```

27

Pwdump

Unique API calls used by a whosthere-alt variant's export function TestDump

```
10001119        push    offset LibFileName ; "secur32.dll"
1000111E        call    ds:LoadLibraryA
10001130        push    offset ProcName ; "LsaEnumerateLogonSessions"
10001135        push    esi             ; hModule
10001136        call    ds:GetProcAddress 1
...
10001670        call    ds:GetSystemDirectoryA
10001676        mov     edi, offset aMsv1_0_dll ; \\msv1_0.dll
...
100016A6        push    eax             ; path to msv1_0.dll
100016A9        call    ds:GetModuleHandleA 2
```

28

Keystroke Logging

Kernel-Based Keyloggers

- Khó phát hiện với những ứng dụng ở user-mode
- Thường là một phần của Rootkits
- Hoạt động như drivers của bàn phím
- Vượt qua các chương trình bảo vệ người dùng ở user-space

29

Keystroke Logging

- User-Space Keyloggers
  - Sử dụng Windows API
  - Thực hiện với các hooking hoặc polling
- Hooking
  - Sử dụng hàm SetWindowsHookEx để thông báo cho mã độc mỗi lần nhấn phím
- Polling
  - Dùng hàm GetAsyncKeyState và hàm GetForegroundWindow để liên tục thăm dò trạng thái của các phím

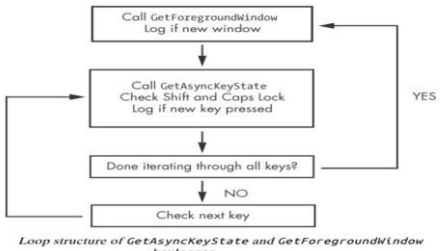
30

Polling Keyloggers

- GetAsyncKeyState
  - Xác định xem một phím đã được nhấn hay không
- GetForegroundWindow
  - Xác định cửa sổ foreground

31

Polling Keyloggers



Loop structure of GetAsyncKeyState and GetForegroundWindow keylogger

32

Polling Keyloggers

```
[Up]
[Num Lock]
[Down]
[Right]
[UP]
[Left]
[PageDown]
```

33

Nội dung

1. Downloader
2. Backdoor
3. Công cụ đánh cắp thông tin
4. Các cơ chế duy trì hiện diện
5. Leo thang đặc quyền
6. Các kỹ thuật trong Rootkit

34

Các cơ chế duy trì hiện diện

- Sửa đổi Registry: Run key
- Các Registry entries quan trọng
  - AppInit_DLLs
  - Winlogon Notify
  - SvcHost DLLs

35

Sửa đổi Registry

- Run key
  - HKEY_LOCAL_MACHINE\ SOFTWARE\ Microsoft\ Windows\ CurrentVersion\ Run
  - Nhiều thứ khác, với Autoruns
- Công cụ *Process Monitor* hiển thị các sửa đổi Registry (Sửa những giá trị trong registry, thêm key...)

36

APPINIT DLLS

AppInit_DLLs được nạp vào mọi tiến trình có sử dụng User32.dll

- HKEY_LOCAL_MACHINE\ SOFTWARE\ Microsoft\ Windows NT\ CurrentVersion\ Windows
- Registry này chứa một danh sách các DLL
- Có nhiều tiến trình load chúng
- Mã độc sẽ gọi DLLMain để kiểm tra tiến trình trước khi nó khởi chạy payload

37

Winlogon Notify

Giá trị của Notify có trong

- HKEY_LOCAL_MACHINE\ SOFTWARE\ Microsoft\ Windows
- Những DLLs handles sẽ xử lý các sự kiên của winlogon.exe
- Mã độc thường gắn với các sự kiện như đăng nhập, khởi động cùng hệ thống, khóa màn hình,...
- Nó thậm chí có thể khởi chạy ở cả trong chế độ Safe mode của windows.

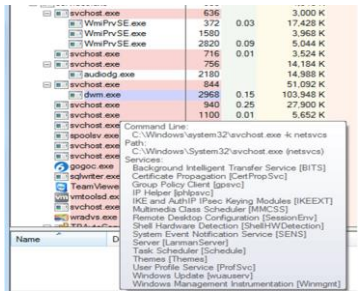
38

SvcHost DLLs

- Svchost là một tiến trình chung cho các service khác nhau
- Nhiều tiến trình Svchost có thể chạy cùng một lúc
- Các Group được xác định tại
  - HKEY_LOCAL_MACHINE\ SOFTWARE\ Microsoft\ Windows NT\ CurrentVersion\ Svchost
- Các Services được xác định tại
  - HKEY_LOCAL_MACHINE\ System\ CurrentControlSet\ Services\ ServiceName

39

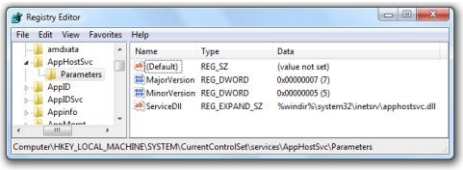
Process Explorer



40

ServiceDLL

- Tất cả các svchost.exe DLL chứa một tham số key với một giá trị ServiceDLL
- Mã độc sẽ set ServiceDLL đến vị trí của các DLL độc hại



41

Groups

- Mã độc thường tự add nó vào một Group đang tồn tại
  - Hoặc ghi đè lên một nonvital service
  - Thường thì một service rất hiếm khi được sử dụng bởi nhóm netsvcs
- Phát hiện điều này với phân tích động và theo dõi Registry
- Hoặc tìm các hàm của service giống như CreateServiceA trong lúc Disassembly

42

Trojanized System Binaries

- ❑ Mã độc tiến hành Patch các byte của một binary trên hệ thống để buộc hệ thống thực thi mã độc
- ❑ Một khi bị nhiễm thì những lần sau hệ thống đều load mã độc
- ❑ Các DLLs thường là mục tiêu phổ biến
- ❑ Thông thường thì các entry function được sửa đổi
- ❑ Nhảy tới đoạn mã được chèn vào một phần trống của binary
- ❑ Sau đó thực thi DLL như bình thường

Trojanized System Binaries

rtutils.dll's DLL Entry Point Before and After Trojanization

Original code	Trojanized code
`DllEntryPoint(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpReserved)`	`DllEntryPoint(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpReserved)`

`mov edi, edi`
`push ebp`
`mov ebp, esp`
`push ebx`
`mov ebx, [ebp+8]`
`push esi`
`mov esi, [ebp+0Ch]`
	`jmp DllEntryPoint_0`

DLL Load-Order Hijacking

The default search order for loading DLLs on Windows XP is as follows:

1. The directory from which the application loaded
2. The current directory
3. The system directory (the `GetSystemDirectory` function is used to get the path, such as *.../Windows/System32/*)
4. The 16-bit system directory (such as *.../Windows/System/*)
5. The Windows directory (the `GetWindowsDirectory` function is used to get the path, such as *.../Windows/*)
6. The directories listed in the `PATH` environment variable

KnownDLLs Registry Key

- ❑ Chứa danh sách các vị trí DLL cụ thể
- ❑ Ghi đè trình tự tìm kiếm các DLL đã được liệt kê
- ❑ Thay đổi thứ tự nạp các DLL,
  - ▪ Chiếm quyền ưu tiên nạp các DLL của thư mục System32
  - ▪ Không được bảo vệ bởi KnownDLLs

explorer.exe

- ❑ Nằm tại /Windows
- ❑ Load ntshrui.dll từ System32
- ❑ Không biết ntshrui.dll nằm ở đâu -> thực hiện việc tìm kiếm
- ❑ Một ntshrui.dll độc hại trong /Windows sẽ được nạp thay thế

Vulnerable DLLs

- ❑ Bất kỳ những binary không được tìm thấy trong /System32 thì đều có thể gây ra sự nguy hiểm
- ❑ explorer.exe có khoảng 50 DLL dễ gây nguy hiểm
- ❑ Các DLL biết đến thì hầu hết không được bảo vệ đầy đủ, vì:
- ❑ Nhiều DLL lại load các DLL khác
- ❑ Import đệ quy tuân theo trình tự tìm kiếm mặc định

Nội dung

1. Downloader
2. Backdoor
3. Công cụ đánh cắp thông tin
4. Các cơ chế duy trì hiện diện
5. Leo thang đặc quyền
6. Các kỹ thuật trong Rootkit

49

Leo thang đặc quyền

- ❑ Các tiến trình chạy bởi người dùng không thể làm được mọi thứ
- ❑ Những hàm giống như TerminateProcess hoặc CreateRemoteThread yêu cầu quyền của hệ thống (trên quản trị viên)
- ❑ SeDebugPrivilege là quyền dùng để debug
- ❑ Cho phép các tài khoản quản trị cục bộ có thể leo thang lên quyền của hệ thống (System privileges)

50

Leo thang đặc quyền

Setting the access token to SeDebugPrivilege

```
00401003  lea     eax, [esp+1Ch+TokenHandle]
00401006  push    eax                     ; TokenHandle
00401007  push    (TOKEN_ADJUST_PRIVILEGES | TOKEN_QUERY)
; DesiredAccess
00401009  call    ds:GetCurrentProcess
0040100F  push    eax                     ; ProcessHandle
00401010  call    ds:OpenProcessToken 1
00401016  test    eax, eax
00401018  jz      short loc_401080
0040101A  lea     ecx, [esp+1Ch+Luid]
0040101E  push    ecx                     ; lpLuid
0040101F  push    offset Name             ; "SeDebugPrivilege"
00401024  push    0                       ; lpSystemName
00401026  call    ds:LookupPrivilegeValueA
0040102C  test    eax, eax
0040102E  jnz     short loc_40103E
```

51

Leo thang đặc quyền

```
...
0040103E  mov     eax, [esp+1Ch+Luid.LowPart]
00401042  mov     ecx, [esp+1Ch+Luid.HighPart]
00401046  push    0                       ; ReturnLength
00401048  push    0                       ; PreviousState
0040104A  push    10h                     ; BufferLength
0040104C  lea     edx, [esp+28h+NewState]
00401050  push    edx                     ; NewState
00401051  mov     [esp+2Ch+NewState.Privileges.Luid.LowPt], eax 3
00401055  mov     eax, [esp+2Ch+TokenHandle]
00401059  push    0                   ; DisableAllPrivileges
0040105B  push    eax                 ; TokenHandle
0040105C  mov     [esp+34h+NewState.PrivilegeCount], 1
00401064  mov     [esp+34h+NewState.Privileges.Luid.HighPt], ecx 4
00401068  mov     [esp+34h+NewState.Privileges.Attributes],
SE_PRIVILEGE_ENABLED 5
00401070  call    ds:AdjustTokenPrivileges 2
```

52

Nội dung

1. Downloader
2. Backdoor
3. Công cụ đánh cắp thông tin
4. Các cơ chế duy trì hiện diện
5. Leo thang đặc quyền
6. Các kỹ thuật trong Rootkit

53

User-Mode Rootkits

- ❑ Sửa đổi các hàm cục bộ của hệ điều hành
- ❑ Ẩn tập tin, kết nối mạng, tiến trình,...

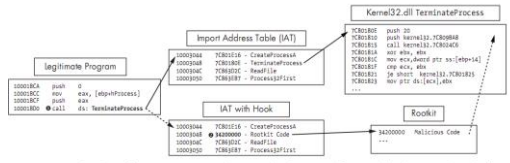
54

IAT (Import Address Table) Hooking

- Có thể sửa đổi
  - IAT (Import Address Table)
  - EAT (Export Address Tables)
- Các thành phần của PE File
- Làm đầy trong bộ Loader

55

IAT Hooking



IAT hooking of TerminateProcess. The top path is the normal flow, and the bottom path is the flow with a rootkit.

56

Inline Hooking

- Ghi đè các API
- Chứa trong các Import DLLs
- Thay đổi những mã trong actual function, không phải con trỏ

57

Các kỹ thuật khởi chạy

58

Nội dung

7. Launcher
8. Tiêm vào tiến trình
9. Thay thế tiến trình
10. Tiêm vào hook
11. Detour
12. Tiêm vào APC

59

Launcher

- Sets chính bản thân nó là một phần của mã độc
- Có thể thực thi ngay tức thì hoặc một thời điểm nào đó
- Nó che giấu những hành vi của mã độc trước người dùng
- Thường chứa mã độc mà họ đang load lên
  - Một file thực thi hoặc DLL trong phần rsrc của nó
- Các items thông thường có bên trong . rsrc
  - Icon, images, menus, trings,...

60

Mã hóa hoặc giải mã

- Phần .rsrc có thể được mã hóa hoặc nén
- Extraction phần rsrc sẽ sử dụng các API như:
  - FindResource
  - LoadResource
  - SizeofResource
- Thường sẽ có những đoạn code để leo thang đặc quyền

61

Nội dung

7. Launcher
8. Tiêm vào tiến trình
9. Thay thế tiến trình
10. Tiêm vào hook
11. Detour
12. Tiêm vào APC

62

Tiêm vào tiến trình

- Là một trong những kỹ thuật khởi chạy của mã độc phổ biến nhất
- Injects code vào một chương trình đang chạy
- Che giấu những hành vi nguy hiểm
- Có thể vượt qua tường lửa và các cơ chế bảo vệ
- Các API thường được gọi:
  - VirtualAllocEx để cấp phát vùng nhớ
  - WriteProcessMemory để ghi lên nó

63

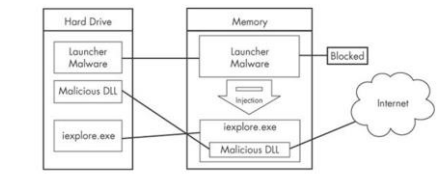
Tiêm vào tiến trình

- Tiêm code vào một tiến trình từ xa với việc gọi LoadLibrary
- Khi load, hệ điều hành tự động gọi DLLMain chứa những đoạn mã độc hại

64

Tiêm vào tiến trình

- Code của mã độc có được đặc quyền giống như mã được tiêm vào



DLL injection—the launcher malware cannot access the Internet until it injects into iexplore.exe.

65

Tiêm vào tiến trình

CreateRemoteThread sử dụng ba tham số

- Tiến trình xử lý hProcess
- Điểm bắt đầu lpStartAddress (LoadLibrary)
- Tham số lpParameter tên của DLL độc hại

C Pseudocode for DLL injection

```
hVictimProcess = OpenProcess(PROCESS_ALL_ACCESS, 0, victimProcessID );

pNameInVictimProcess = VirtualAllocEx(hVictimProcess,...,sizeof(maliciousLibraryName),...,...);
WriteProcessMemory(hVictimProcess,...,maliciousLibraryName, sizeof(maliciousLibraryName),...);
GetModuleHandle("Kernel32.dll");
GetProcAddress(...,"LoadLibraryA");
CreateRemoteThread(hVictimProcess,...,...,LoadLibraryAddress,pNameInVictimProcess,...,...);
```

66

Direct Injection

- ❑ Tiêm mã trực tiếp vào tiến trình từ xa
- ❑ Không sử dụng DLL
- ❑ Linh hoạt hơn so với tiêm vào DLL
- ❑ Yêu cầu nhiều về tùy chỉnh mã
- ❑ Đảm bảo chạy được mà không ảnh hưởng đến tiến trình
- ❑ Khó thực hiện

67

Nội dung

7. Launcher
8. Tiêm vào tiến trình
9. Thay thế tiến trình
10. Tiêm vào hook
11. Detour
12. Tiêm vào APC

68

Thay thế tiến trình

- ❑ Mã độc ghi đè lên vùng nhớ của một đối tượng đang chạy
- ❑ Che giấu mã độc và làm nó giống như một tiến trình hợp lệ
- ❑ Tránh nguy cơ bị crash giữa một tiến trình với tiến trình tiêm
- ❑ Mã độc nhận được các quyền của tiến trình thay thế
- ❑ *svchost.exe* thường là mục tiêu.

69

Suspended State

- ❑ Trong trạng thái suspended, tiến trình vẫn được nạp vào bộ nhớ nhưng primary Thread của nó thì cũng bị suspended theo.
- ❑ Mã độc có thể ghi đè mã của nó lên đó trước khi tiến trình thoát khỏi trạng thái suspended và chạy trở lại.
- ❑ Sử dụng giá trị CREATE_SUSPENDED trong tham số dwCreationFlags trong khi gọi đến các hàm CreateProcess .

70

Thay thế tiến trình

Assembly code showing process replacement

```
00401535    push    edi             ; lpProcessInformation
00401536    push    ecx             ; lpStartupInfo
00401537    push    ebx             ; lpCurrentDirectory
00401538    push    ebx             ; lpEnvironment
00401539    push    CREATE_SUSPENDED ; dwCreationFlags
0040153B    push    ebx             ; bInheritHandles
0040153C    push    ebx             ; lpThreadAttributes
0040153D    lea     edx, [esp+94h+CommandLine]
00401541    push    ebx             ; lpProcessAttributes
00401542    push    edx             ; lpCommandLine
00401543    push    ebx             ; lpApplicationName
00401544    mov     [esp+0A0h+StartupInfo.dwFlags], 101h
0040154F    mov     [esp+0A0h+StartupInfo.wShowWindow], bx
00401557    call    ds:CreateProcessA
```

71

Thay thế tiến trình

C pseudocode for process replacement

```
CreateProcess(...,"svchost.exe",...,CREATE_SUSPEND,...);
ZwUnmapViewOfSection(...);
VirtualAllocEx(...,ImageBase,SizeOfImage,...);
WriteProcessMemory(...,headers,...);
for (i=0; i < NumberOfSections; i++) {
  WriteProcessMemory(...,section,...);
}
SetThreadContext();
...
ResumeThread();
```

- ❑ ZwUnmapViewOfSection giải phóng tất cả bộ nhớ được trỏ bởi một section
- ❑ VirtualAllocEx – Cấp phát lại bộ nhớ
- ❑ WriteProcessMemory – Đẩy/Ghi mã độc vào nó

72

Thay thế tiến trình

C pseudocode for process replacement

```
CreateProcess(...,"svchost.exe",...,CREATE_SUSPEND,...);
ZwUnmapViewOfSection(...);
VirtualAllocEx(...,ImageBase,SizeOfImage,...);
WriteProcessMemory(...,headers,...);
for (i=0; i < NumberOfSections; i++) {
  WriteProcessMemory(...,section,...);
}
SetThreadContext();
...
ResumeThread();
```

- SetThreadContext – Khôi phục lại môi trường của tiến trình nạn nhân
- ResumeThread – Thực thi mã độc hại

73

Nội dung

7. Launcher
8. Tiêm vào tiến trình
9. Thay thế tiến trình
10. Tiêm vào hook
11. Detour
12. Tiêm vào APC

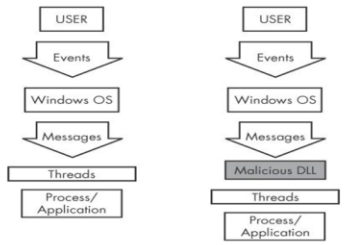
74

Hooks

- Windows Hook sẽ chặn bắt các thông điệp trao đổi giữa các ứng dụng
- Malicious hooks
  - Đảm bảo rằng mã độc sẽ chạy bất cứ khi nào mà và một thông điệp trao đổi có thể bị chặn
  - Đảm bảo rằng một DLL sẽ được nào vào vùng nhớ của tiến trình phía nạn nhân

75

Tiêm vào hook



Event and message flow in Windows with and without hook injection

76

Local and Remote Hooks

- *Local Hook*: Sẽ chặn bắt và sửa đổi các thông điệp bên trong một tiến trình cục bộ
- *Remote Hook*: Sẽ chặn bắt và sửa đổi các thông điệp được trao đổi giữa các tiến trình ở xa với nhau

77

High-Level and Low-Level Remote Hooks

- High-level remote hooks
  - Yêu cầu các thủ tục hook exported các hàm có trong DLL
  - Mapped bởi hệ điều hành vào không gian tiến trình của một hook thread hoặc tất cả các thread
- Low-level remote hooks
  - Yêu cầu các thủ tục hook được chứa trong quá trình cài đặt các hook

78

Keyloggers dử dụng Hooks

- Các thao tác trên bàn phím sẽ được chụp lại bởi các hook mức cao hoặc mức thấp, sử dụng các thủ tục WH_KEYBOARD hoặc WH_KEYBOARD_LL

79

Keyloggers dử dụng *SetWindowsHookEx*

- Các tham số
  - idHook – Loại hook cài đặt
  - Lpfn - points to hook procedure
  - hMod – Xử lý các DLL hoặc các module cục bộ, thủ tục lpfn được xác định
  - dwThreadId – Thread liên kết các hook với nhau. Zero = tất cả các thread
- Thủ tục hook phải gọi CallNextHookEx để truyền đến các hook tiếp theo

80

Thread Targeting

- Load tất cả các thread có thể làm giảm hiệu suất của hệ thống
- Cũng có thể kích hoạt một IPS
- Keyloggers load tất cả các thread để lấy được tất cả các hành động từ bàn phím
- Có những mã độc nhằm vào thread đơn
- Thông thường một mục tiêu Message của windows hiếm khi được sử dụng, chẳng hạn như WH_CBT (a computer-based training message)

81

Keylogger

Hook injection, assembly code

```
00401100        push    esi
00401101        push    edi
00401102        push    offset LibFileName ; "hook.dll"
00401107        call    LoadLibraryA
0040110D        mov     esi, eax
0040110F        push    offset ProcName ; "MalwareProc"
00401114        push    esi             ; hModule
00401115        call    GetProcAddress
0040111B        mov     edi, eax
0040111D        call    GetNotepadThreadId
00401122        push    eax             ; dwThreadId
00401123        push    esi             ; hmod
00401124        push    edi             ; lpfn
00401125        push    WH_CBT   ; idHook
00401127        call    SetWindowsHookExA
```

82

Keylogger

- Malicious DLL sẽ nạp hook.dll
- Thu được địa chỉ của các Malicious hook
- Thủ tục hook chỉ gọi CallNextHookEx
- Một thông điệp WH_CBT được gửi đến Notepad thread
- Buộc hook.dll được load bởi Notepad
- Nó sẽ bắt đầu hook các thông điệp gõ phím từ chương trình Notepad khi mà Notepad bắt đầu chạy

83

Nội dung

7. Launcher
8. Tiêm vào tiến trình
9. Thay thế tiến trình
10. Tiêm vào hook
11. Detour
12. Tiêm vào APC

84

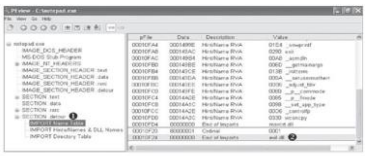
Detour

- Detours giúp các nhà phát triển ứng dụng dễ dàng sửa đổi ứng dụng trên hệ điều hành
- Các mã độc sử dụng nó để thêm các DLL mới vào các binary có trên đĩa
- Sửa đổi cấu trúc PE để tạo một .detour section
- Chứa PE header gốc với một bảng địa chỉ được Import

85

Detour

- setdll là một công cụ của Microsoft được sử dụng để trỏ PE vào bảng địa chỉ import mới
- Có nhiều cách khác nhau để thêm một .detour section



A PEview of Detours and the evil.dll

86

Nội dung

7. Launcher
8. Tiêm vào tiến trình
9. Thay thế tiến trình
10. Tiêm vào hook
11. Detour
12. Tiêm vào APC

87

Asynchronous Procedure Call (APC)

- Trực tiếp một thread được thực thi code khác trước khi thực thi theo đường dẫn thông thường của nó
- Mỗi thread đều có hàng đợi APC đi kèm với nó
- Chúng được xử lý khi thread trong trạng thái có thể thay đổi, chẳng hạn như các hàm này được gọi:
  - WaitForSingleObjectEx
  - WaitForMultipleObjectsEx
  - Sleep

88

Các loại APC

- Kernel-Mode APC
  - Tạo cho hệ thống hoặc driver
- User-Mode APC
  - Tạo cho một ứng dụng
- APC Injection được sử dụng trong cả hai trường hợp

89

Chèn APC từ chế độ người dùng

- Sử dụng API QueueUserAPC
- Thread phải ở trạng thái có thể thay đổi
- WaitForSingleObjectEx là một trong những hàm được gọi phổ biến trong Windows API
- Nhiều Thread ở trạng thái có thể thay đổi

90

Chèn APC từ chế độ người dùng

- Mở một trình xử lý Thread
- QueueUserAPC được gọi với pfnAPC từ LoadLibraryA (Load một DLL)
- dwData chứa tên của DLL (*dbnet.dll*)
- Svchost.exe thường là mục tiêu của APC injection

APC injection from a user-mode application

```
00401DA9        push    [esp+4+dwThreadId]      ; dwThreadId
00401DAD        push    0                       ; bInheritHandle
00401DAF        push    10h                     ; dwDesiredAccess
00401DB1        call    ds:OpenThread 1
00401DB7        mov     esi, eax
00401DB9        test    esi, esi
00401DBB        jz      short loc_401DCE
00401DBD        push    [esp+4+dwData]          ; dwData = dbnet.dll
00401DC1        push    esi                     ; hThread
00401DC2        push    ds:LoadLibraryA 2       ; pfnAPC
00401DC8        call    ds:QueueUserAPC
```


Chèn APC từ nhân

- Các driver độc hại và Rootkits thường muốn thực thi code ở không gian người dùng
- Điều này là khó thực hiện
- Một phương pháp là APC injection để có được không gian phía người dùng
- Mục tiêu hay được sử dụng nhất là *svchost.exe*
- Các hàm sử dụng
  - KeInitializeApc
  - KeInsertQueueApc

Chèn APC từ nhân

User-mode APC injection from kernel space

```
000119BD        push    ebx
000119BE        push    1 1
000119C0        push    [ebp+arg_4] 2
000119C3        push    ebx
000119C4        push    offset sub_11964
000119C9        push    2
000119CB        push    [ebp+arg_0] 3
000119CE        push    esi
000119CF        call    ds:KeInitializeApc
000119D5        cmp     edi, ebx
000119D7        jz      short loc_119EA
000119D9        push    ebx
000119DA        push    [ebp+arg_C]
000119DD        push    [ebp+arg_8]
000119E0        push    esi
000119E1        call    edi         ;KeInsertQueueApc
```


Nội dung

1. Downloader
2. Backdoor
3. Công cụ đánh cắp thông tin
4. Các cơ chế duy trì hiện diện
5. Leo thang đặc quyền
6. Các kỹ thuật trong Rootkit

Nội dung

7. Launcher
8. Tiêm vào tiến trình
9. Thay thế tiến trình
10. Tiêm vào hook
11. Detour
12. Tiêm vào APC