جامعة البعث
كلية الهندسة المعلوماتية
قسم هندسة البرمجيات

Protecting Windows Systems from Rootkits

تقديم :
هادي السالم – يوسف إبراهيم – محمد نور شحيمة

إشراف :
د. محسن حسين

2017 -2018

Project Goals

The Principle Gools(Build a Rootkit) :

1. Hide Processes
2. Hide Files.
3. Hide Registry Entry
4. Build Key logger
5. Build Remote control unite
6. Stay hidden in the system

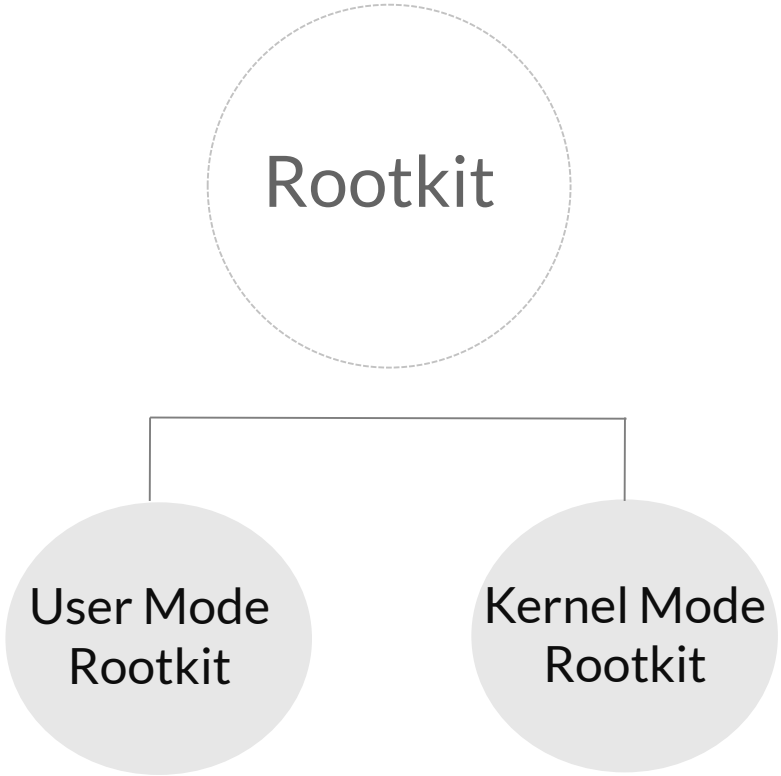
The Final Gool:

1. Build Rootkits Scanner

Rootkit
Defenation

- الروتكيت مجموعة من البرمجيات التي تؤمن التواجد الخفي داخل النظام .
- الروتكيت ظهرت لانظمة Unix وتعني Root + Kit .
- ظهر مفهوم الروتكيت منذ حوالي 20 سنة
- الروتكيت تعني التخفي الغير قابل للكشف.
- الروتكت هي سلاح ذو حدين يمكن استخدامه بشكل قانوني.
- الروتكيت ليس فايروس.
- الروتكيت لا تتطلب استغلال ثغرات ضمن النظام الهدف.
- الروتكيت تستخدم تقنية التعديل Modification .

Rootkit
Types



Rootkit
Types

- Kernel Mode : هي الروتكيت التي تندمج مع نواة نظام التشغيل ويصبح من المستحيل كشفها, بالنسبة لانظمة Windows اصبح من الصعب تطبيق هذا النوع بعد صدور حماية الPatch Guard .

- User Mode : تستخدم API نظام التشغيل في عملها وتستخدم تقنيات الDll Injection وال Code Injection .

Rootkit

Example (Stuxnet)

Rootkit
Examples

Vanquish
User Mode

LTD-4
Kernel Mode

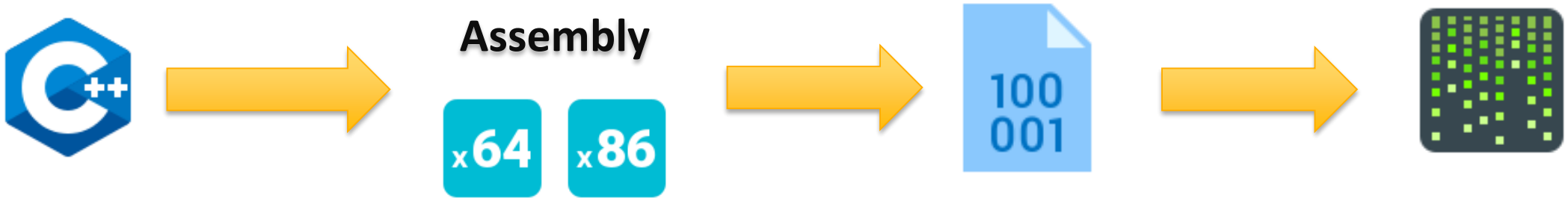
Blue Pill
Kernel Mode

FU
Kernel Mode

Shellcode
Defenation

- الShellcode هو كود مكتوب بلغة الاسيمبلي وبصيغة Binary .
- الShellcode يتم حقنه ضمن Process تعمل ضمن النظام بحيث يتم تنفيذ الShellcode .
- عند كتابة الShellcode يجب الانتباه الى حماية ASLR() والى العناوين الذاكرية بشكل عام.
- يجب ان يكون الShellcode صغير قدر المستطاع .
- الShellcode يختلف من نظام لاخر ومن اصدار لاخر .
- يمكن تنفيذ الShellcode على الجهاز الهدف اما من خلال :
  1. استغلال ثغرة ضمن النظام او ضمن البرامج التي تعمل ضمن النظام .
  2. من خلال القيام بحقن Process عاملة ضمن النظام .
- الShellcode يجب ان يكون خالي من Bad Characters. (\0 , \f , \r , \n)

Shellcode
Defenation



Shellcode
Linux VS Windows

- يوجد System Call محددة وثابتة يمكن استدعاءها بسهولة.
- النظام مفتوح المصدر والكود المصدري بمتناول الجميع.
- إمكانية التعديل والاضافة على المشاريع والبرامج البرمجية بسهولة من قبل الجميع.

- لا يوجد System Call محددة يمكن استدعاءها .
- النظام مغلق واحتكاري (*Un Documented API*).
- يوجد قيود صارمة على البرامج التي يتم اضافتها على ال *Microsoft Store*

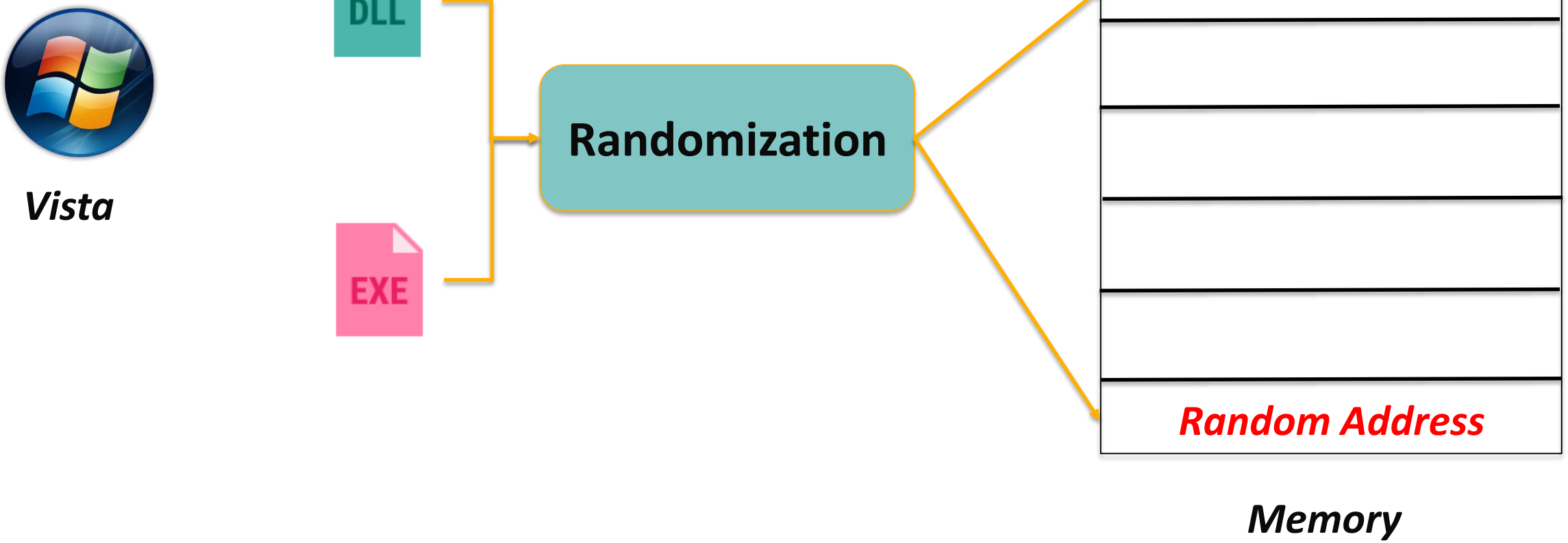
Shellcode

ASLR(Address Space Layout Randomization)

- في النسخ القديمة من Windows كان الMemory Manager يقوم بتحميل الBinaries في نفس الفضاء الذاكري .
- حيث يدعم ال Linker ميزة تمكن المبرمجين من وضع Base Address المفضل للملف (مثلا DLL) ضمن Header الملف نفسه .
- في حال لم يتم وضع العنوان المفضل سيتم تحميل ملفات (*.exe) الى العنوان الافتراضي (0x40000000) .
- ومن أجل الملفات (*.DLL) سيتم التحميل الى العنوان الافتراضي (0x10000000).
- في حال كانت الذاكرة غير متاحة لتطابق العناوين السابقة سيتم التحميل الى مناطق أخرى من الذاكرة.

Shellcode

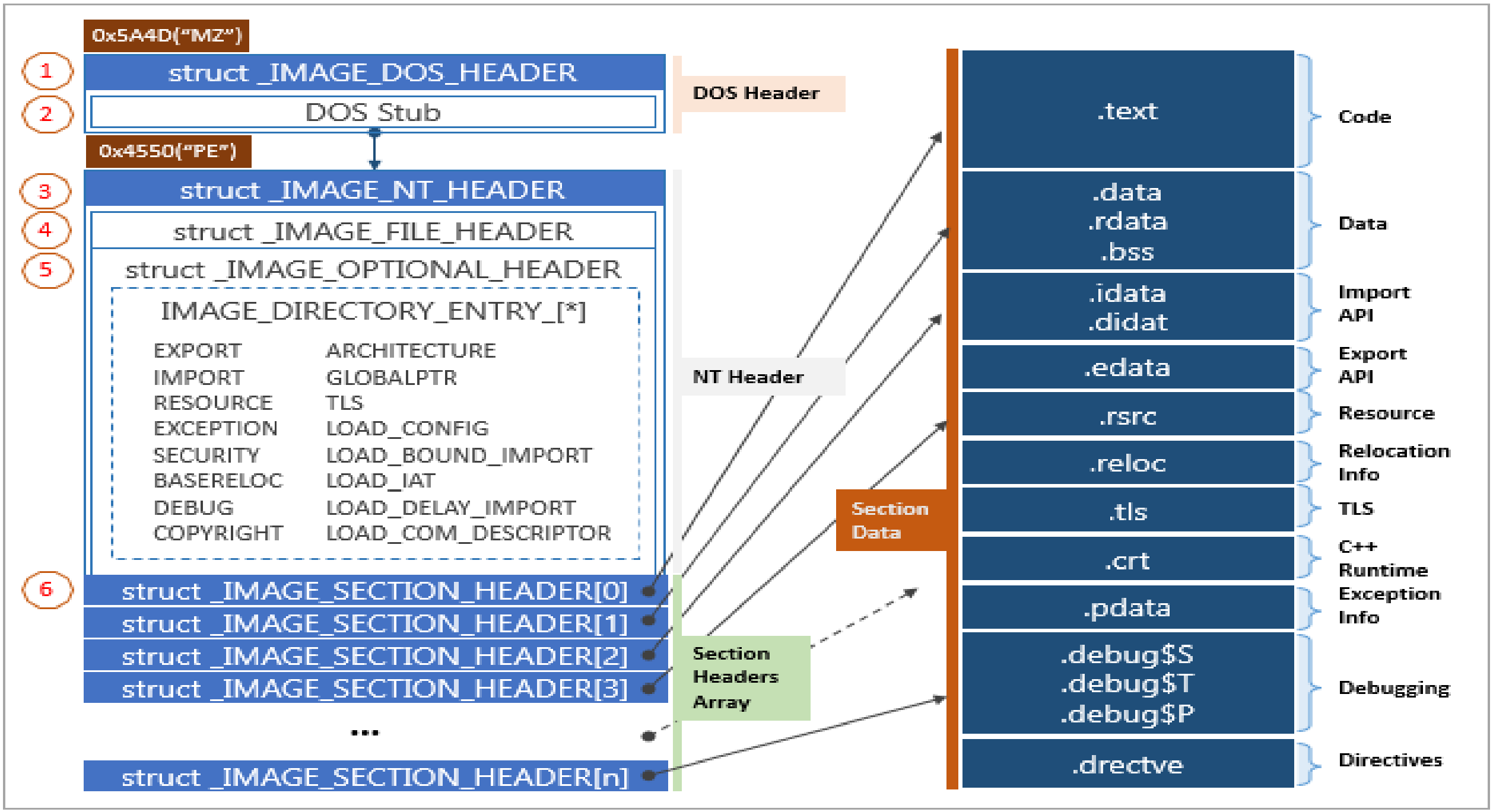
ASLR(Address Space Layout Randomization)



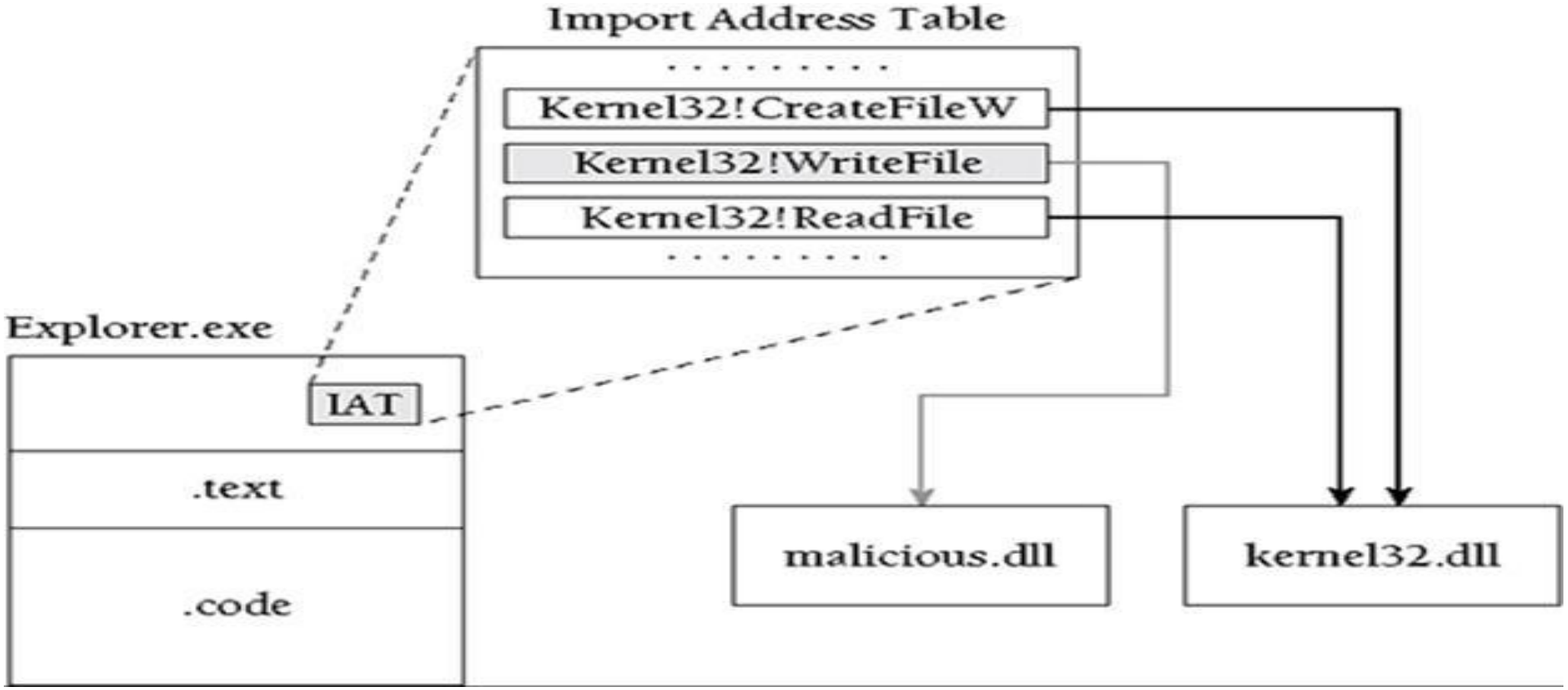
PE Files
Anatomy

- أي ملف تنفيذي يحوي على Header يحوي على جدول IAT (Import Address Table)
- جدول الIAT يحوي على عناوين الدوال التي سيستخدمها البرنامج .
- بتطبيق عملية ال DLL Injection سيتم تعديل عناوين الدوال التي يستخدمها البرنامج .

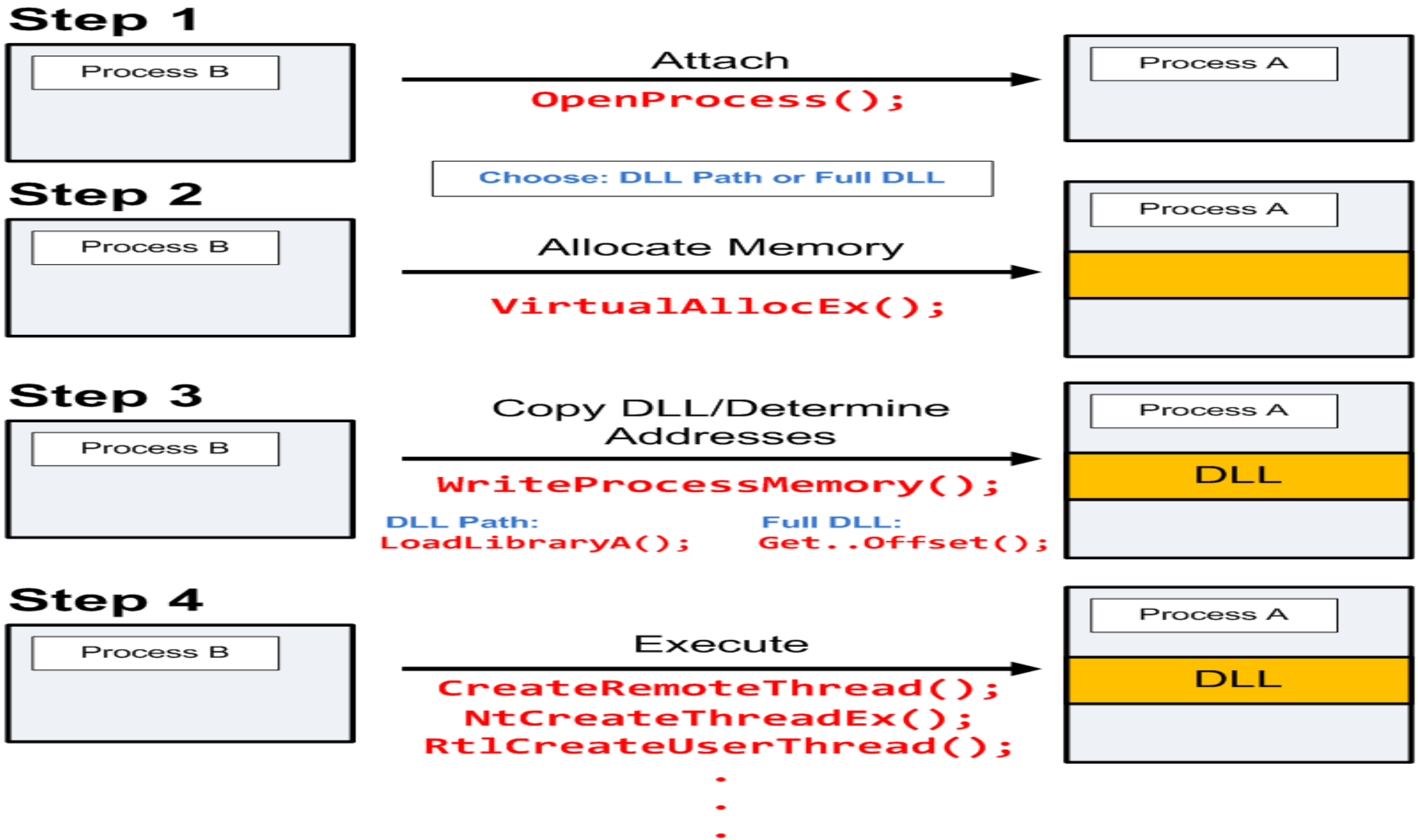
PE Format


0x5A4D("MZ")
1
struct _IMAGE_DOS_HEADER
2
DOS Stub
0x4550("PE")
3
struct _IMAGE_NT_HEADER
4
struct _IMAGE_FILE_HEADER
5
struct _IMAGE_OPTIONAL_HEADER
IMAGE_DIRECTORY_ENTRY_[*]
EXPORT
IMPORT
RESOURCE
EXCEPTION
SECURITY
BASERELOC
DEBUG
COPYRIGHT
ARCHITECTURE
GLOBALPTR
TLS
LOAD_CONFIG
LOAD_BOUND_IMPORT
LOAD_IAT
LOAD_DELAY_IMPORT
LOAD_COM_DESCRIPTOR
6
struct _IMAGE_SECTION_HEADER[0]
struct _IMAGE_SECTION_HEADER[1]
struct _IMAGE_SECTION_HEADER[2]
struct _IMAGE_SECTION_HEADER[3]
...
struct _IMAGE_SECTION_HEADER[n]
DOS Header
NT Header
Section Headers Array
Section Data
.text
.data
.rdata
.bss
.idata
.didat
.edata
.rsrc
.reloc
.tls
.crt
.pdata
.debug$S
.debug$T
.debug$P
.drectve
Code
Data
Import API
Export API
Resource
Relocation Info
TLS
C++ Runtime Exception Info
Debugging
Directives

PE Files
Anatomy



Overview

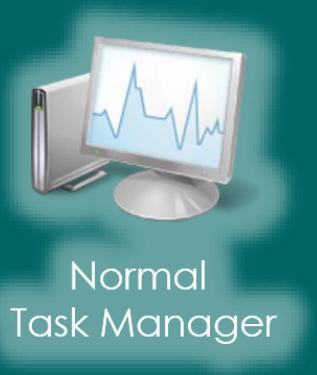

Step 1
Process B
Attach
OpenProcess();
Process A
Choose: DLL Path or Full DLL
Step 2
Process B
Allocate Memory
VirtualAllocEx();
Process A
Step 3
Process B
Copy DLL/Determine Addresses
WriteProcessMemory();
DLL Path:
LoadLibraryA();
Full DLL:
Get..Offset();
Process A
DLL
Step 4
Process B
Execute
CreateRemoteThread();
NtCreateThreadEx();
RtlCreateUserThread();
.
.
.
Process A
DLL

Hiding
Processes(DLL Injection)

- سيتم كتابة مكتبة DLL بحيث سيتم حقنها ببرنامج الTask Manager
- وظيفة ملف DLL هو التعديل على الدالة :

ZwQuerySystemInformation()

- بحيث يكون لدينا مؤشران :
  1. مؤشر يؤشر على العنصر الحالي
  2. مؤشر يؤشر على العنصر السابق



Normal
Task Manager

Infected
Task Manager

Process A

Process B

Process C

Process A

Process B

Process C

ZWQuerySystemInformation()

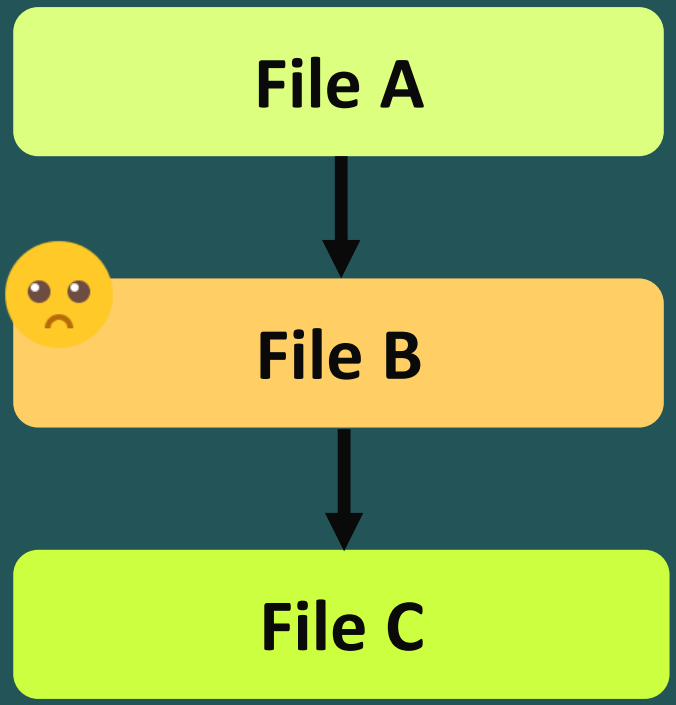
Hiding
Files (DLL Injection)

- سيتم تنفيذ عملية Hooking لدالتين :
  1. الدالة FindFirstFileA() : تعيد معلومات عن اول ملف ضمن المجلد
  2. الدالة FindNextFile() : دالة بوليانية (Enumeration).

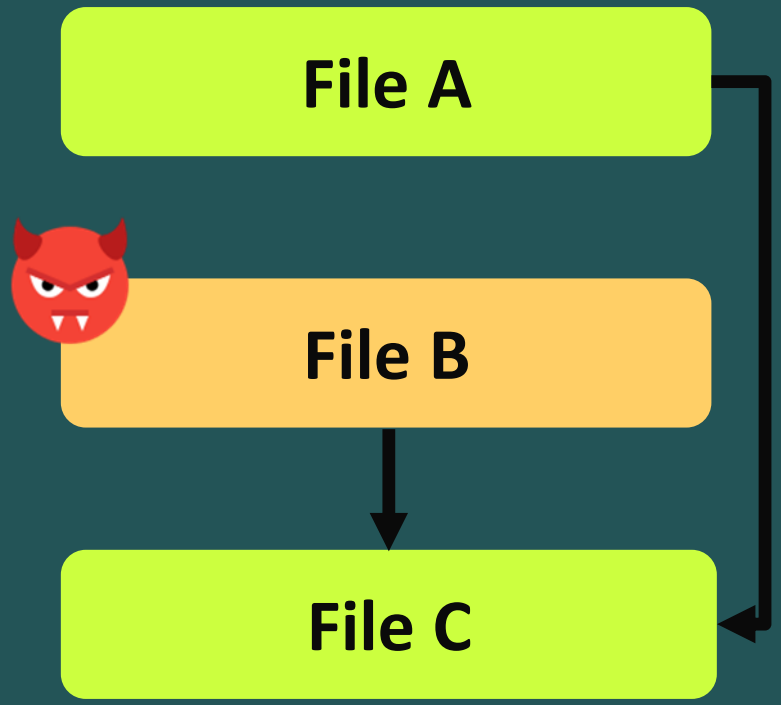
- اذا اعادت الدالة FindNextFile() معلومات عن الملف المراد اخفاءه سنستدعي الدالة مرة أخرى.

- سيتم تخطي الملف المراد اخفاءه (لن يتم حذفه) .


Normal
Files Manager


File A
File B
File C


Infected
Files Manager


File A
File B
File C

Hiding
Registry Entries

- سيتم عمل Hooking :
  1. للدالة RegEnumValueA() : تعطي مجموعة المدخلات (Entries) لمفتاح (Key) معين .
- نقوم باستدعاء الدالة RegEnumValueA() حتى نحصل على الخطأ ERROR_NO_MORE_ITEMS .
- والذي يدل على عدم وجود أي Entry جديد يتبع للمفتاح المعطى .
- نقوم بعملية مقارنة مع اسم الEntry المراد اخفاءه .
- في حال المطابقة سيتم استدعاء الدالة السابقة ولكن مع زيادة الIndex الخاص بالEntry بمقدار 1 .

Normal Registries Viewer

Entry A

Entry B

Entry C

Infected Registries Viewer

Entry A

Entry B

Entry C

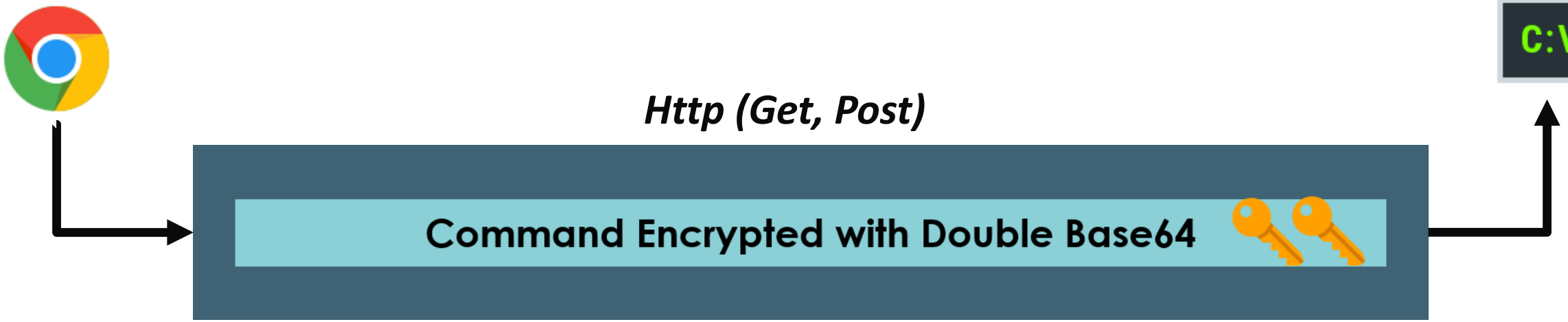
Backdooring

The System

- سيتم بناء برنامج قادر على توفير الوصول البعيد الى برنامج CMD.exe موجود على جهاز الضحية (Victim)
- سيتم ارسال الأوامر الى سيرفر HTTP تابع لجهاز الHacker .
- يقوم الHacker بالتحكم بجهاز الضحية عن طريق المتصفح .
- سيتم تشفير الRequest/Reply بترميز Base64 مزدوج .
- كل 60 ثانية يقوم جهاز الضحية بفحص هل يوجد امر جديد يريد الHacker تنفيذه .

Backdooring

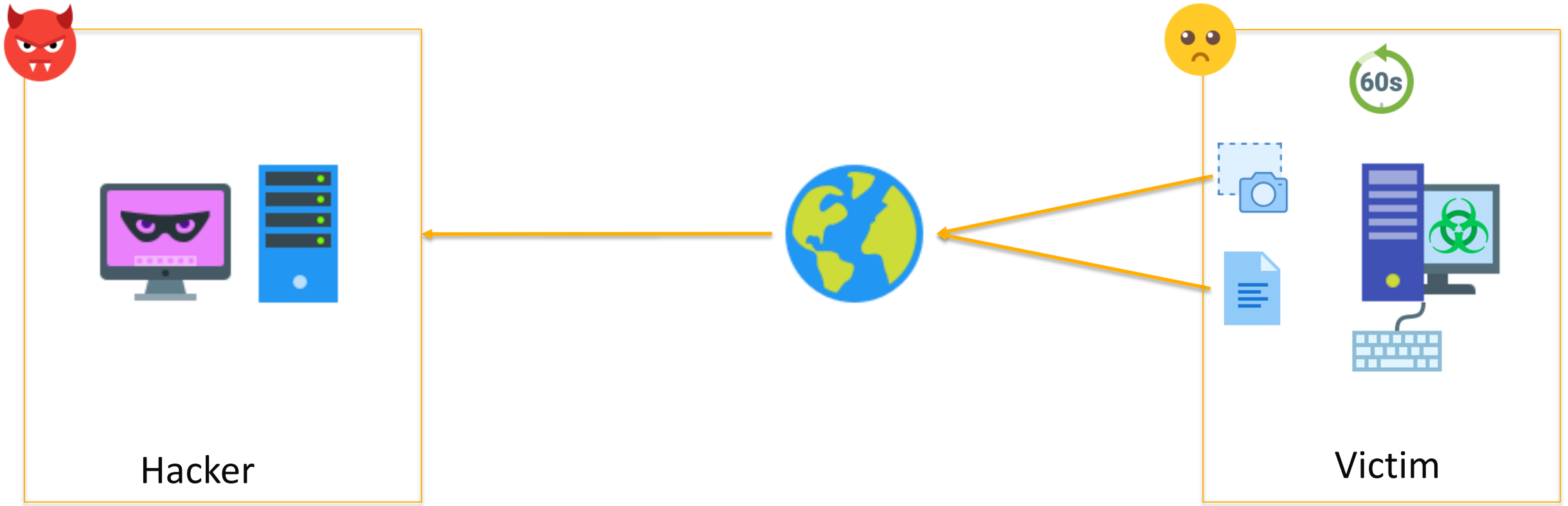
The System



KeyLogger

- سيتم انشاء Key Logger يقوم بتسجيل ضربات لوحة المفاتيح ثم يقوم بتخزينها في ملف نصي ليتم لاحقا ارسالها الى Server الHacker .
- أيضا سيتم اخذ Screen Shot لسطح مكتب الضحية كل 60 ثانية .
- يتم ارسال ملف الصورة والملف النصي الى سيرفر FTP خاص بالHacker .
- يتم تزويد الKeylogger باسم مستخدم وكلمة مرور لحساب الFTP بشكل مضمن داخل الكود.
- نستخدم الدالة : SetWindowsHookEx () ومنها نستخدم الدالة KeyboardProc ()

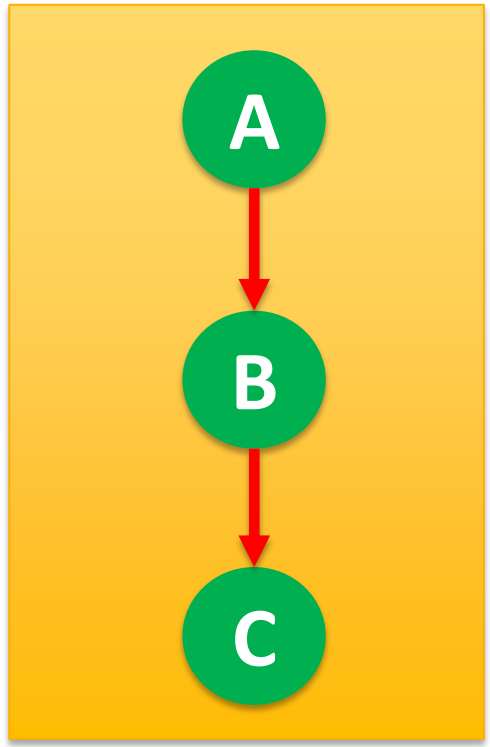
KeyLogger



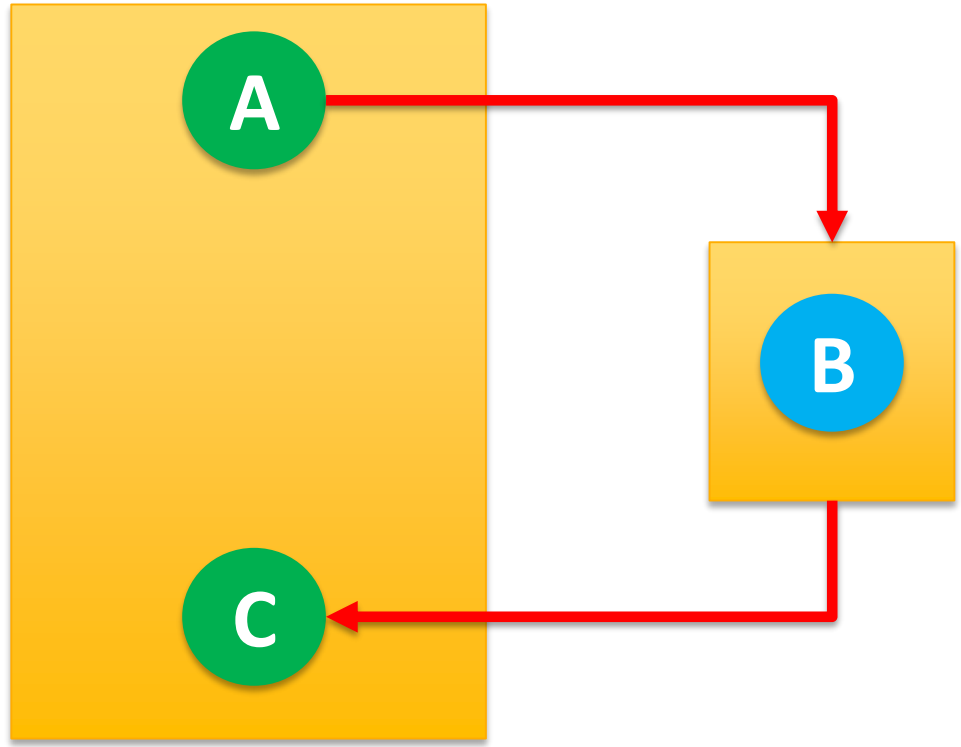
Hiding
inside the System

- سيتم استخدام تقنية Code cave في التخفي حيث سيتم اختيار برنامج يقلع مع اقلاع النظام ليتم حقنه بملف DLL .
- ملف الDLL يقوم بأنشاء قسم جديد ضمن الHeader البرنامج المصاب.
- يتم نسخ الShellcode للقسم الجديد .
- الshellcode مسؤول عن تشغيل الRootkit.
- عندما يقلع النظام في المرة القادة ستقلع الRootkit مع النظام.

CodeCave

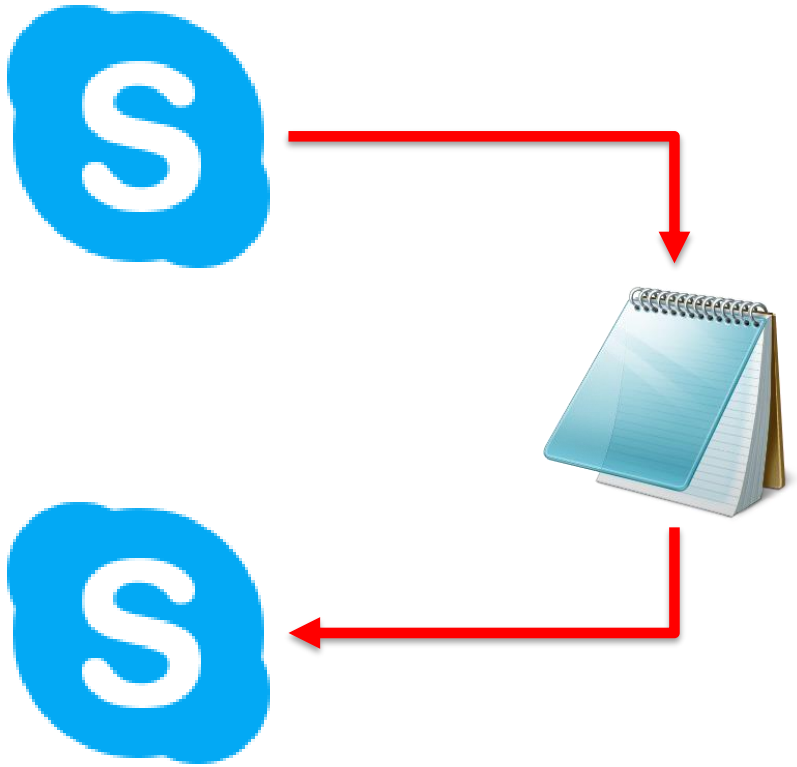


Normal Program



CodeCave

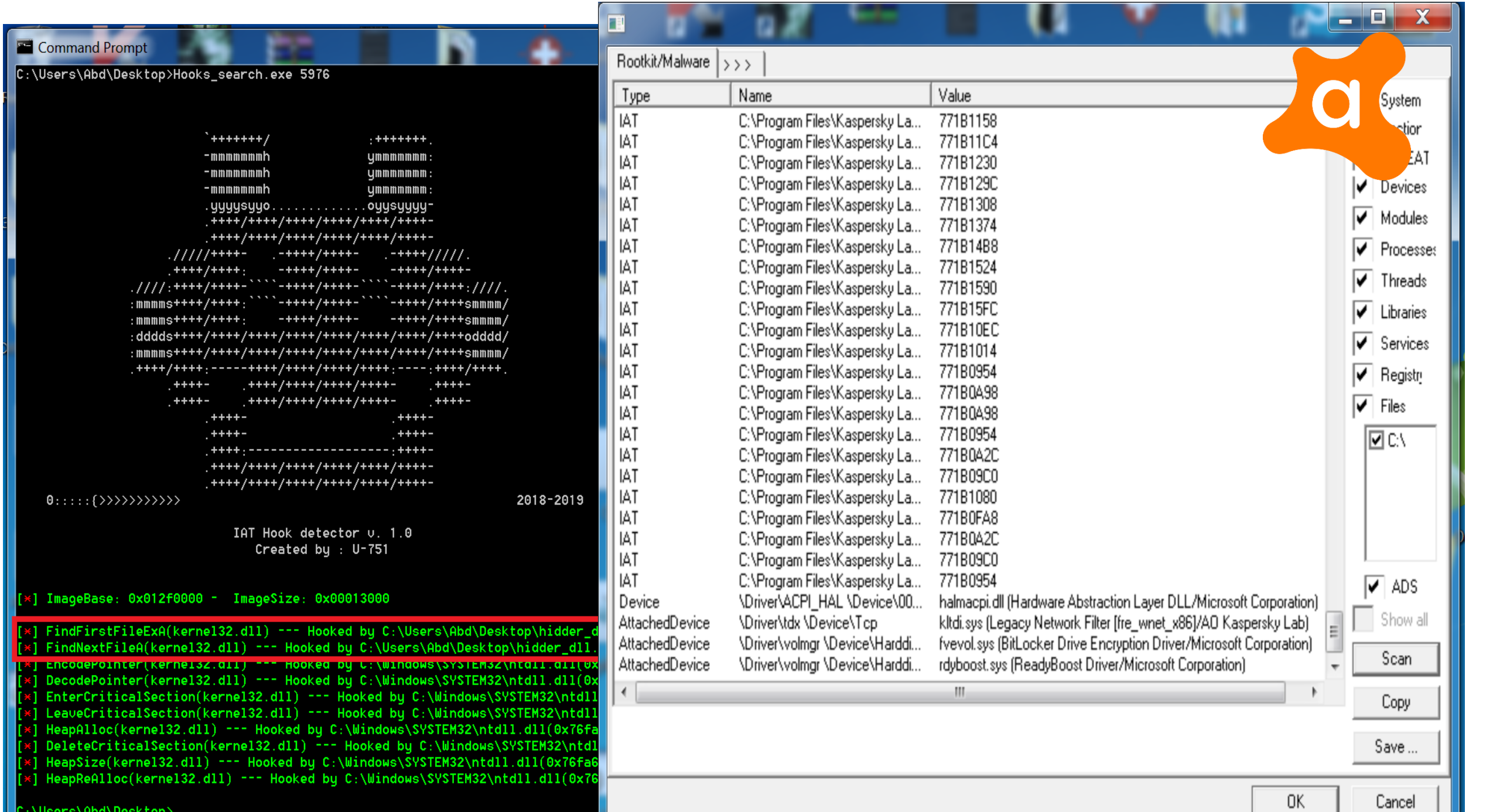
CodeCave



- سيتم انشاء قسم جديد ضمن الHeader لبرنامج Skype .
- ويتم حقن Shellcode بداخل القسم الجديد.
- انشاء نسخة جديدة من برنامج Skype .
- عندما يتم تشغيل البرنامج الجديد سيتم تنفيذ الShellcode .
- وينفذ برنامج Notepad + Skype.

Rootkits
Scanner

- عندما يتم تطبيق تقنية ال IAT Hooking من قبل الRootkit عندها ستؤشر الى عناوين مكاتب DLL مختلفة عن مكاتب النظام الاصلية .
- يجب ان يقع العنوان المؤشر عليه من قبل الدالة ضمن : [Image Base , Image Base + Image Size]
- في حال وقع العنوان خارج المجال السابق سيتم اعتبار ان الدالة قد تعرضت لعملية Hooking .
- من خلال هذه الطريقة سنحصل على الكثير من الإنذارات الكاذبة (False Alarm) .
- يمكن التخلص من 90 بالمئة من الإنذارات الكاذبة من خلال اعتبار ان برامج النظام ومكتباته التي تقوم بعملية Hook لدالة ما انها آمنة .
- أيضا يمكن اعتبار عمليات الHook التي تقوم بها الAnti-virus أمنة.


Command Prompt
C:\Users\Abd\Desktop>Hooks_search.exe 5976
2018-2019
IAT Hook detector v. 1.0
Created by : U-751
[*] ImageBase: 0x012f0000 - ImageSize: 0x00013000
[*] FindFirstFileExA(kernel32.dll) --- Hooked by C:\Users\Abd\Desktop\hidder_d
[*] FindNextFileA(kernel32.dll) --- Hooked by C:\Users\Abd\Desktop\hidder_dll.
[*] DecodePointer(kernel32.dll) --- Hooked by C:\Windows\SYSTEM32\ntdll.dll(0x
[*] EnterCriticalSection(kernel32.dll) --- Hooked by C:\Windows\SYSTEM32\ntdll
[*] LeaveCriticalSection(kernel32.dll) --- Hooked by C:\Windows\SYSTEM32\ntdll
[*] HeapAlloc(kernel32.dll) --- Hooked by C:\Windows\SYSTEM32\ntdll.dll(0x76fa
[*] DeleteCriticalSection(kernel32.dll) --- Hooked by C:\Windows\SYSTEM32\ntdl
[*] HeapSize(kernel32.dll) --- Hooked by C:\Windows\SYSTEM32\ntdll.dll(0x76fa6
[*] HeapReAlloc(kernel32.dll) --- Hooked by C:\Windows\SYSTEM32\ntdll.dll(0x76
Rootkit/Malware
Type
Name
Value
IAT C:\Program Files\Kaspersky La... 771B1158
IAT C:\Program Files\Kaspersky La... 771B11C4
IAT C:\Program Files\Kaspersky La... 771B1230
IAT C:\Program Files\Kaspersky La... 771B129C
IAT C:\Program Files\Kaspersky La... 771B1308
IAT C:\Program Files\Kaspersky La... 771B1374
IAT C:\Program Files\Kaspersky La... 771B14B8
IAT C:\Program Files\Kaspersky La... 771B1524
IAT C:\Program Files\Kaspersky La... 771B1590
IAT C:\Program Files\Kaspersky La... 771B15FC
IAT C:\Program Files\Kaspersky La... 771B10EC
IAT C:\Program Files\Kaspersky La... 771B1014
IAT C:\Program Files\Kaspersky La... 771B0954
IAT C:\Program Files\Kaspersky La... 771B0A98
IAT C:\Program Files\Kaspersky La... 771B0A98
IAT C:\Program Files\Kaspersky La... 771B0954
IAT C:\Program Files\Kaspersky La... 771B0A2C
IAT C:\Program Files\Kaspersky La... 771B09C0
IAT C:\Program Files\Kaspersky La... 771B1080
IAT C:\Program Files\Kaspersky La... 771B0FA8
IAT C:\Program Files\Kaspersky La... 771B0A2C
IAT C:\Program Files\Kaspersky La... 771B09C0
IAT C:\Program Files\Kaspersky La... 771B0954
Device \Driver\ACPI_HAL \Device\00... halmacpi.dll (Hardware Abstraction Layer DLL/Microsoft Corporation)
AttachedDevice \Driver\tdx \Device\Tcp kltdi.sys (Legacy Network Filter [fre_wnet_x86]/AO Kaspersky Lab)
AttachedDevice \Driver\volmgr \Device\Harddi... fvevol.sys (BitLocker Drive Encryption Driver/Microsoft Corporation)
AttachedDevice \Driver\volmgr \Device\Harddi... rdyboost.sys (ReadyBoost Driver/Microsoft Corporation)
System
Devices
Modules
Processes
Threads
Libraries
Services
Registry
Files
C:\
ADS
Show all
Scan
Copy
Save ...
OK
Cancel

Our Tests:

Future Views

- سيتم بناء واجهة رسومية لRootkit Scanner لتسهل تعامل المستخدم مع البرنامج.
- سيتم تطوير البرنامج بحيث يغطي أنظمة Linux و Mac OS .
- سيتم تزويد البرنامج بقاعدة ضخمة من تواقيع لبرمجيات خبيثة .
- سيتم دمج البرنامج بنظام كشف تطفل (NIDS) لمواجهة التهديدات القادمة من الشبكة .

References

- *Rootkits: Subverting the Windows Kernel : July 22, 2005*
  Greg Hoglund

- *Hacking Exposed: Malware and Rootkits: 2009*
  Aaron LeMasters, Michael A. Davis, Sean Bodmer

- *The Rootkit Arsenal : May 4, 2009*
  Bill Blunden

- *Malware Analyst's Cookbook and DVD: Tools and Techniques: 2010*
  Steven Adair, Michael Hale Ligh, Blake Hartstein, Matthew Richard