

# CTF presentation

Jason Mcneice

# Code functionality

Criteria	In my Code I Implemented
Persistence	Adds values to below registry's. <pre>"HKCU\\Software\\Microsoft\\Windows\\CurrentVersion\\Run\\</pre> <pre>\\HKLM\\Software\\Microsoft\\Windows\\CurrentVersion\\Run\\</pre>

-Runs CMD commands to add values to the registry. This will cause the program to run on start up

# Code functionality

Criteria	In my Code I Implemented
Network Connectivity	Sends Post request to dummy server, c2.JasonMcneice.com
Encoding	XOR
Flag 2 – has your name & ends with _4484 and is not visible as a string	JasonMcneice_Flag2_4484

- Source code only contains encrypted message
- Before sending the message to the server the program will decrypt the message
- The second flag is contained within the message

# Code functionality

Criteria	In my Code I Implemented
Code Obfuscation	Anti-reverse engineering technique used 1. Jump Instruction with a Constant Condition 2. Random junk code
Anti-Debugging	Uses C function “IsDebuggerPresent()” if true the program will display a message
Anti-Virtual Machine & Sandboxing	Uses the cpuid instruction to detect if the program is in a VM. Will display a message if so.
Packing	Uses UPX
Flag 1 – has your name and ends with _4484	JasonMcneice_Flag1_4484

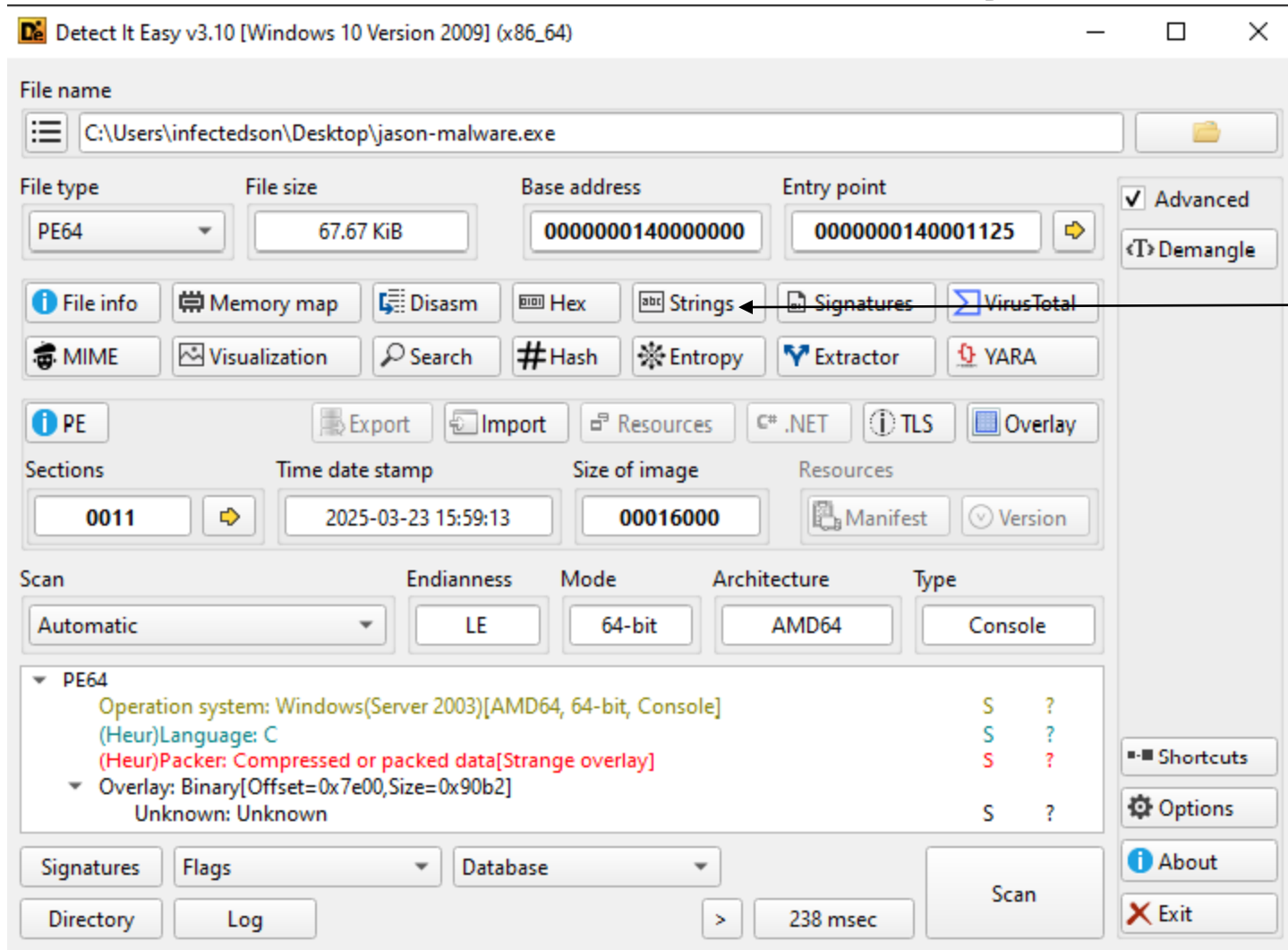
# Part 1 – Basic static analysis

```
FLARE-VI Sun 05/25/2025 18:30:37.58
C:\Users\infectedson\Desktop>upx -d jason-malware.exe
                Ultimate Packer for eXecutables
                Copyright (C) 1996 - 2024
UPX 4.2.4      Markus Oberhumer, Laszlo Molnar & John Reiser    May 9th 2024

      File size      Ratio      Format      Name
      -----
      69298 <-    49330    71.19%    win64/pe    jason-malware.exe

Unpacked 1 file.
```

# Part 1 – Basic static analysis



# Part 1 – Basic static analysis

Number ▾	Offset	Address		Size	Typ	String
Filter	Filter	Filter	Filter	Filter	F...	4484
48	00003488	0000000140005...	Section(2)...	17	U	JasonMcneice_Flag1_4484

Number ▾	Offset	Address		Size	Typ	String
Filter	Filter	Filter	Filter	Filter	F...	c2
54	0000358c	0000000140005...	Section(2)...	13	A	c2.JasonMcneice.com

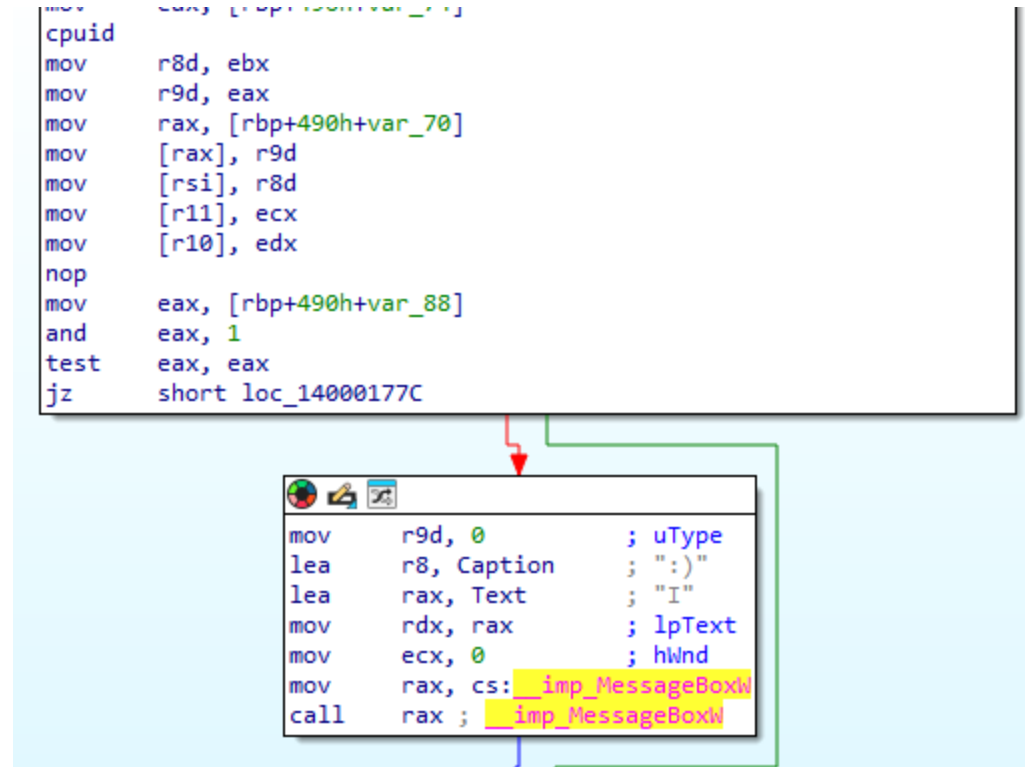
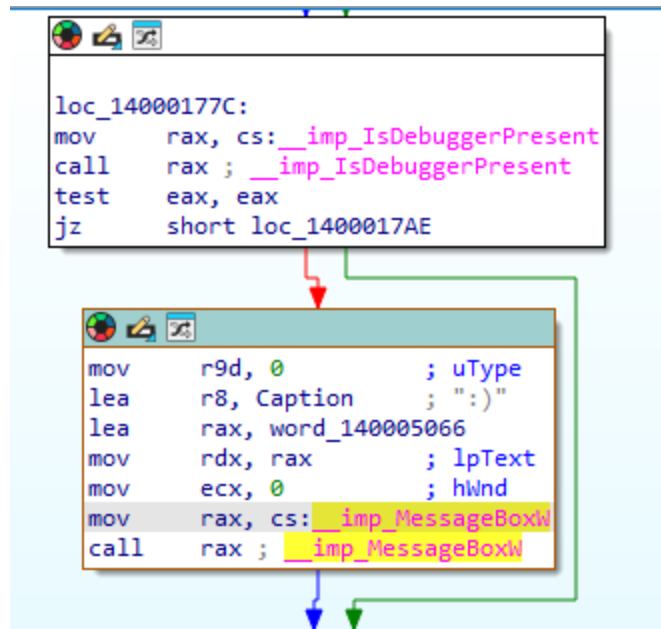
# Part 2 – Bypass anti analysis with IDA pro

- Goal
  - set the instruction pointer to a location past all the anti-analysis techniques.

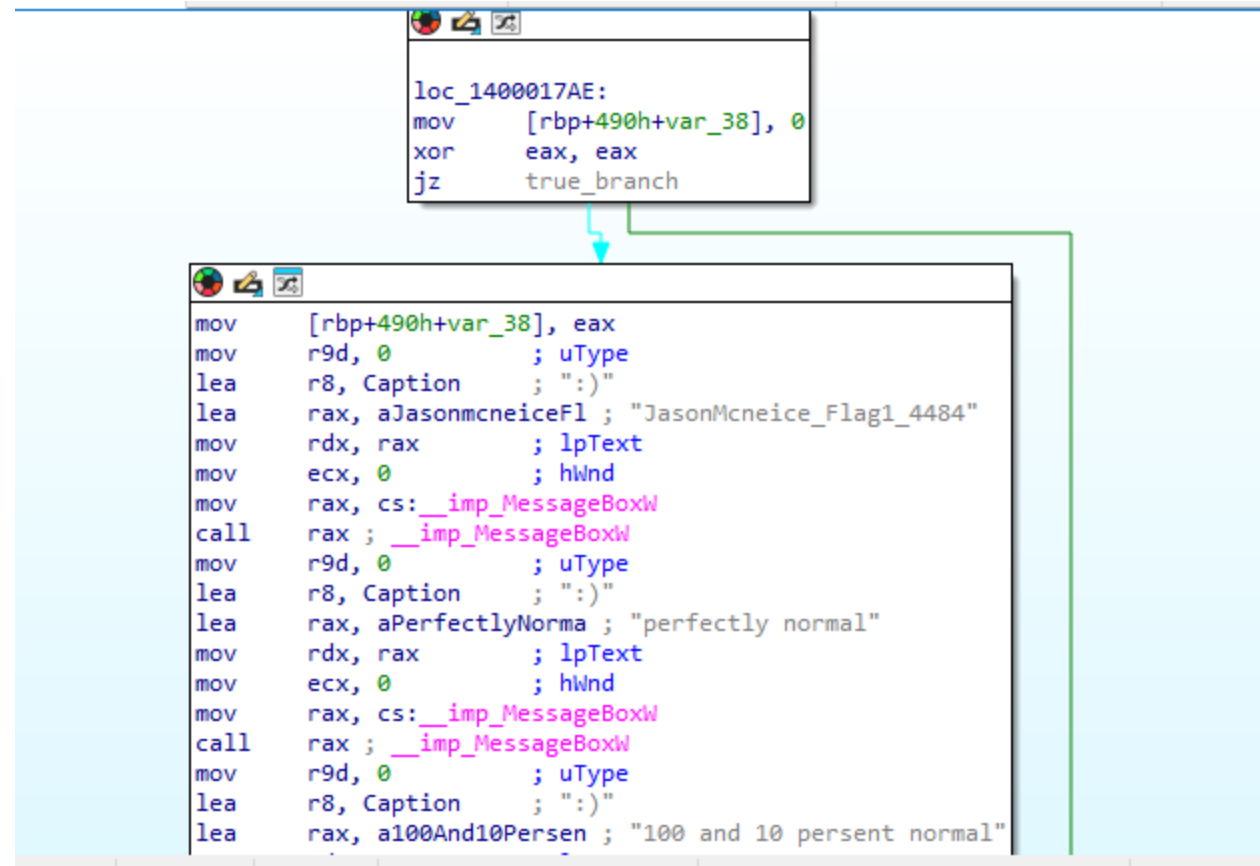




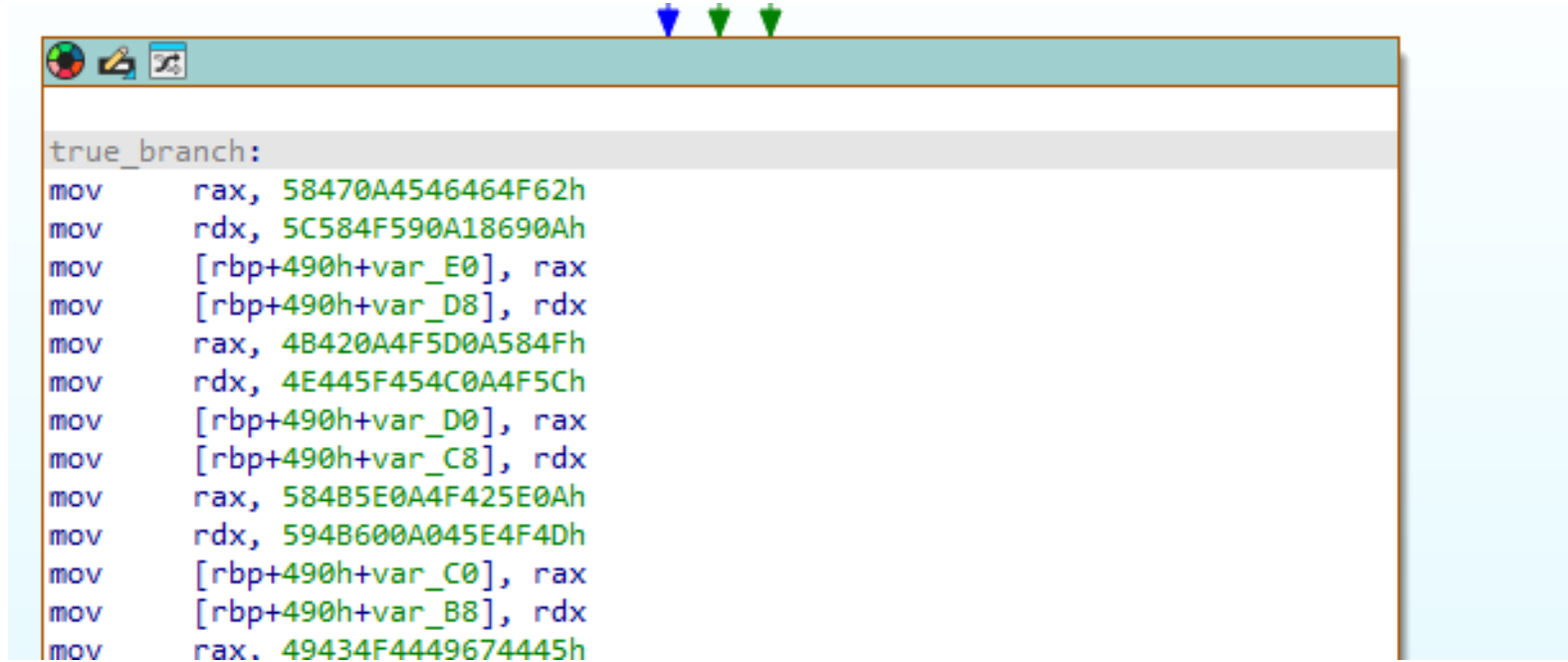
# Part 2 – Bypass anti analysis with IDA pro



# Part 2 – Bypass anti analysis with IDA pro



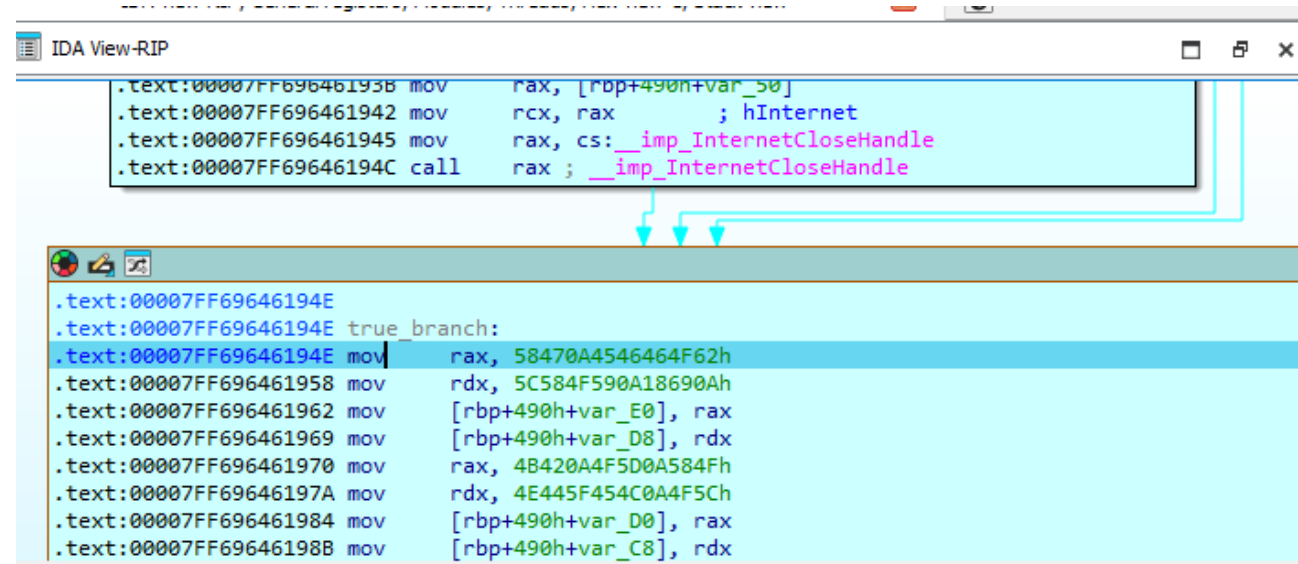
## Part 2 – Bypass anti analysis with IDA pro



```
true_branch:  
mov     rax, 58470A4546464F62h  
mov     rdx, 5C584F590A18690Ah  
mov     [rbp+490h+var_E0], rax  
mov     [rbp+490h+var_D8], rdx  
mov     rax, 4B420A4F5D0A584Fh  
mov     rdx, 4E445F454C0A4F5Ch  
mov     [rbp+490h+var_D0], rax  
mov     [rbp+490h+var_C8], rdx  
mov     rax, 584B5E0A4F425E0Ah  
mov     rdx, 594B600A045E4F4Dh  
mov     [rbp+490h+var_C0], rax  
mov     [rbp+490h+var_B8], rdx  
mov     rax, 49434F4449674445h
```

# Part 2 – Bypass anti analysis with IDA pro

```
push    r12
push    rsi
push    rbx
sub     rsp, 4E0h
lea     rbp, [rsp+80h]
call    __main
lea     rax, [rbp+490h+var_90]
mov     [rbp+490h+var_70], rax
mov     [rbp+490h+var_74], 1
mov     rax, [rbp+490h+var_70]
lea     rsi, [rax+4]
mov     rax, [rbp+490h+var_70]
lea     r11, [rax+8]
mov     rax, [rbp+490h+var_70]
lea     r10, [rax+0Ch]
mov     eax, [rbp+490h+var_74]
cpuid
mov     r8d, ebx
mov     r9d, eax
mov     rax, [rbp+490h+var_70]
mov     [rax], r9d
mov     [rsi], r8d
mov     [r11], ecx
mov     [r10], edx
nop
mov     eax, [rbp+490h+var_88]
and     eax, 1
test    eax, eax
```

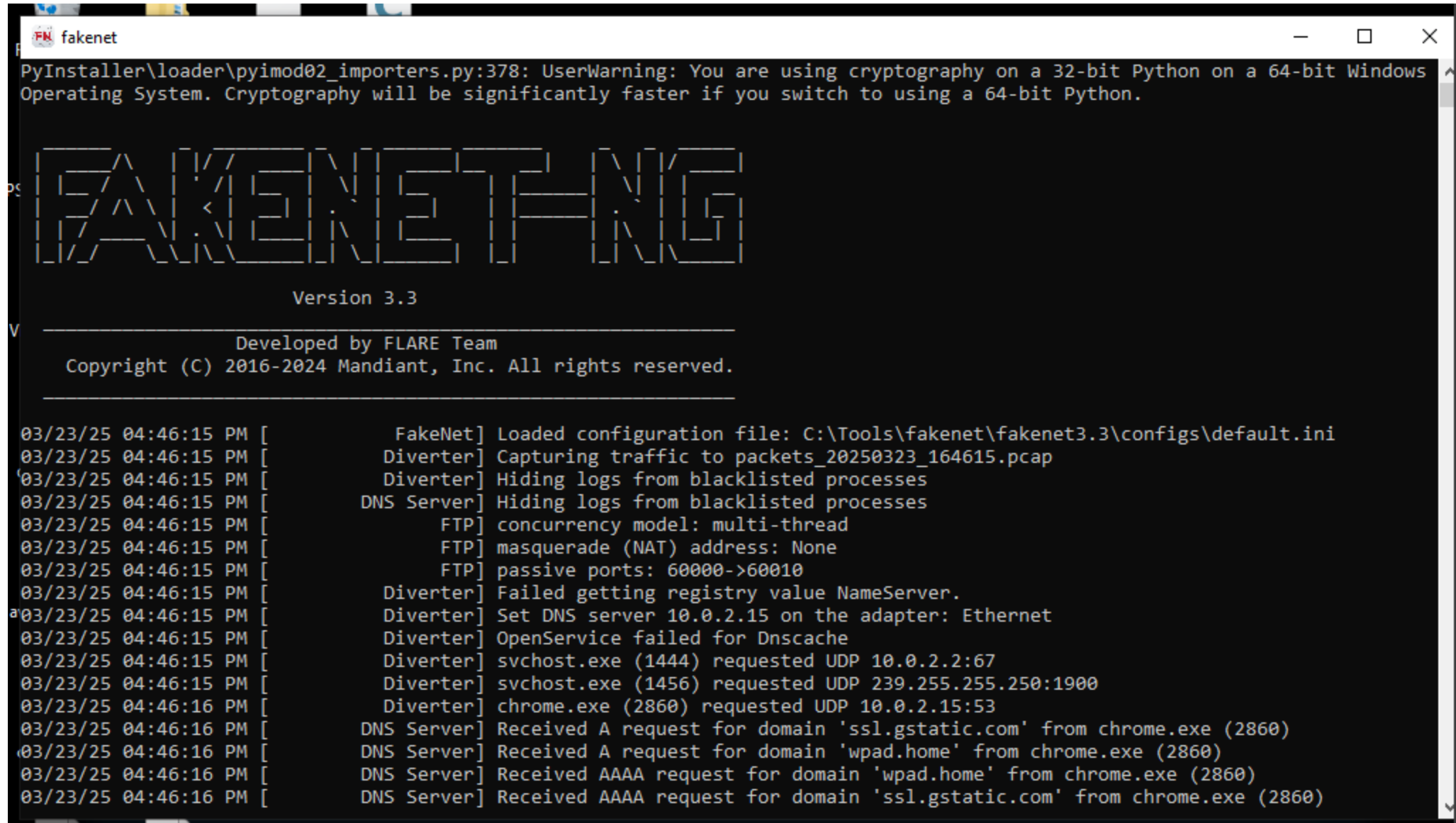


The screenshot shows the IDA Pro interface. The top pane displays assembly code with a call instruction highlighted. The bottom pane shows the disassembly of the call instruction, including a true\_branch label and several mov instructions.

```
IDA View-RIP
.text:00007FF69646193B mov     rax, [rbp+490h+var_50]
.text:00007FF696461942 mov     rcx, rax ; hInternet
.text:00007FF696461945 mov     rax, cs:__imp_InternetCloseHandle
.text:00007FF69646194C call    rax ; __imp_InternetCloseHandle

.true_branch:
.text:00007FF69646194E mov     rax, 58470A4546464F62h
.text:00007FF696461958 mov     rdx, 5C584F590A18690Ah
.text:00007FF696461962 mov     [rbp+490h+var_E0], rax
.text:00007FF696461969 mov     [rbp+490h+var_D8], rdx
.text:00007FF696461970 mov     rax, 4B420A4F5D0A584Fh
.text:00007FF69646197A mov     rdx, 4E445F454C0A4F5Ch
.text:00007FF696461984 mov     [rbp+490h+var_D0], rax
.text:00007FF69646198B mov     [rbp+490h+var_C8], rdx
```

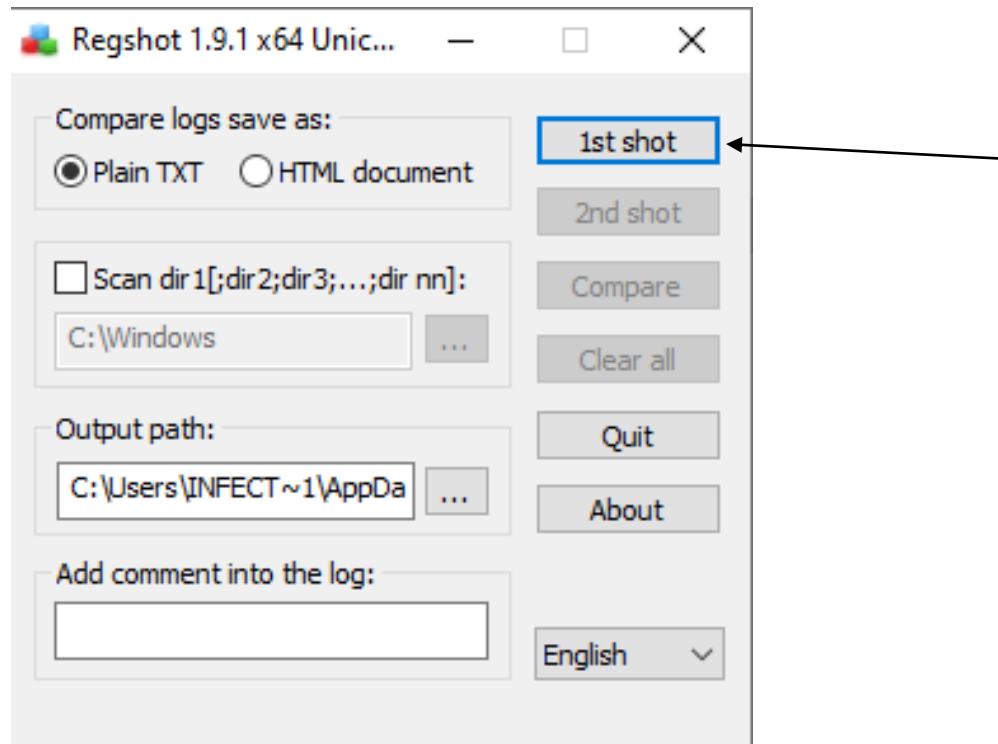
# Part 3 - Dynamic analysis



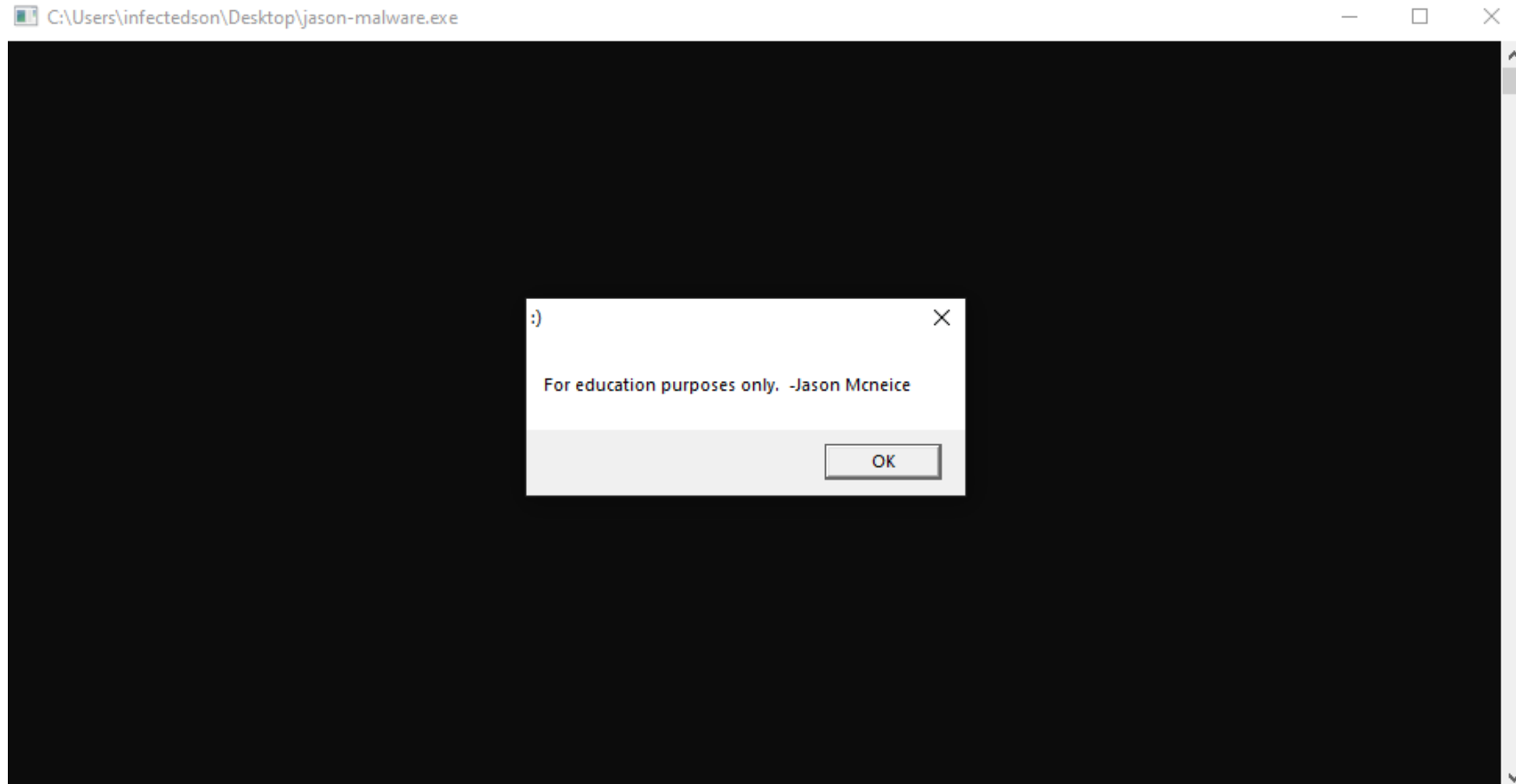
The screenshot shows the FakeNet application window. At the top, a warning message states: "PyInstaller\loader\pyimod02\_importers.py:378: UserWarning: You are using cryptography on a 32-bit Python on a 64-bit Windows Operating System. Cryptography will be significantly faster if you switch to using a 64-bit Python." Below this is the "FAKENET" logo in a stylized, blocky font, followed by "Version 3.3". A horizontal line separates the header from the footer, which reads: "Developed by FLARE Team" and "Copyright (C) 2016-2024 Mandiant, Inc. All rights reserved." The main area of the window displays a log of network traffic. The log entries are as follows:

```
03/23/25 04:46:15 PM [ FakeNet] Loaded configuration file: C:\Tools\fakenet\fakenet3.3\configs\default.ini
03/23/25 04:46:15 PM [ Divertor] Capturing traffic to packets_20250323_164615.pcap
03/23/25 04:46:15 PM [ Divertor] Hiding logs from blacklisted processes
03/23/25 04:46:15 PM [ DNS Server] Hiding logs from blacklisted processes
03/23/25 04:46:15 PM [ FTP] concurrency model: multi-thread
03/23/25 04:46:15 PM [ FTP] masquerade (NAT) address: None
03/23/25 04:46:15 PM [ FTP] passive ports: 60000->60010
03/23/25 04:46:15 PM [ Divertor] Failed getting registry value NameServer.
03/23/25 04:46:15 PM [ Divertor] Set DNS server 10.0.2.15 on the adapter: Ethernet
03/23/25 04:46:15 PM [ Divertor] OpenService failed for Dnscache
03/23/25 04:46:15 PM [ Divertor] svchost.exe (1444) requested UDP 10.0.2.2:67
03/23/25 04:46:15 PM [ Divertor] svchost.exe (1456) requested UDP 239.255.255.250:1900
03/23/25 04:46:16 PM [ Divertor] chrome.exe (2860) requested UDP 10.0.2.15:53
03/23/25 04:46:16 PM [ DNS Server] Received A request for domain 'ssl.gstatic.com' from chrome.exe (2860)
03/23/25 04:46:16 PM [ DNS Server] Received A request for domain 'wpad.home' from chrome.exe (2860)
03/23/25 04:46:16 PM [ DNS Server] Received AAAA request for domain 'wpad.home' from chrome.exe (2860)
03/23/25 04:46:16 PM [ DNS Server] Received AAAA request for domain 'ssl.gstatic.com' from chrome.exe (2860)
```

# Part 3 - Dynamic analysis



# Part 3 - Dynamic analysis

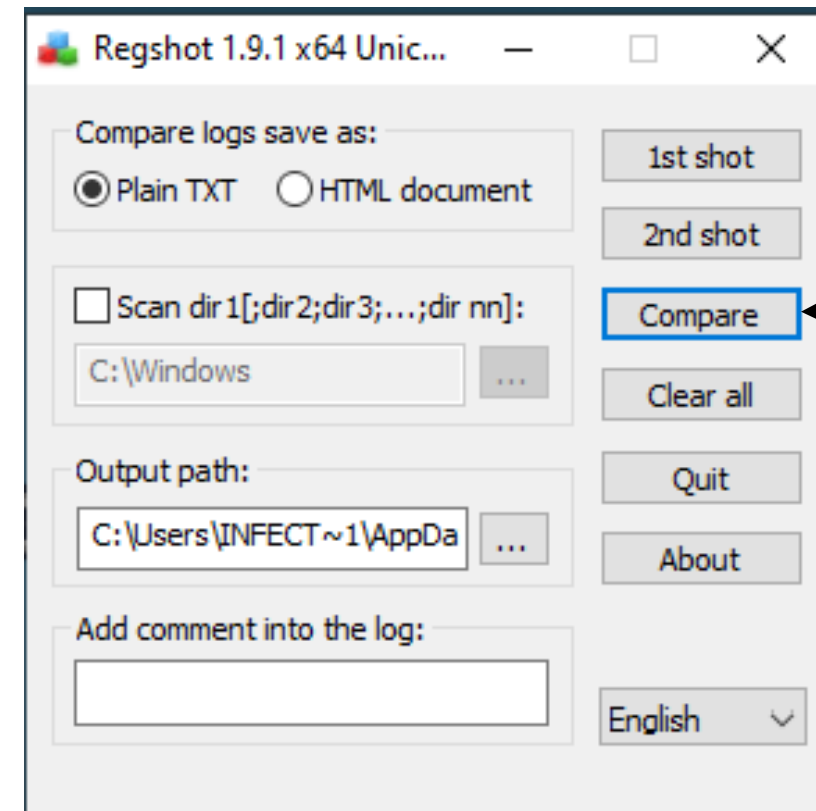
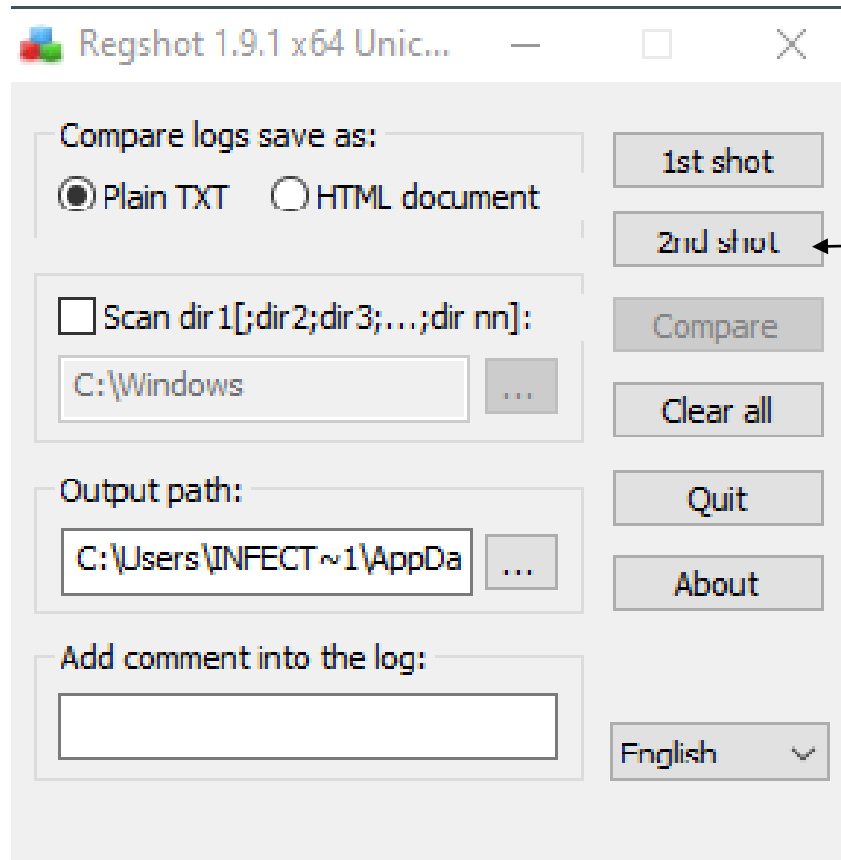


# Part 3 - Dynamic analysis

```
05:10:33 PM [      DNS Server] Received AAAA request for domain c2.jasonmcneice.com from jason-malware.exe
05:10:33 PM [      Diverter] jason-malware.exe (5096) requested TCP 192.0.2.123:80
05:10:33 PM [ HTTPListener80] POST / HTTP/1.1
05:10:33 PM [ HTTPListener80] User-Agent: thing
05:10:33 PM [ HTTPListener80] Host: c2.JasonMcneice.com
05:10:33 PM [ HTTPListener80] Content-Length: 69
05:10:33 PM [ HTTPListener80] Cache-Control: no-cache
05:10:33 PM [ HTTPListener80]
05:10:33 PM [ HTTPListener80]
05:10:33 PM [ HTTPListener80] b' Hello mr C2 server we have found the target. JasonMcneice_Flag2_4484\x00'
05:10:33 PM [ HTTPListener80] Storing HTTP POST headers and data to http-20250322-171033.txt
```



# Part 3 - Dynamic analysis



# Part 3 - Dynamic analysis

```
HKU\S-1-5-21-3269865406-3658123218-2710846053-1001\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\notSus:  
"C:\Users\infectedson\Desktop\jason-malware.exe"
```

# Summary

- Flags: JasonMcneice\_Flag1\_4484 & JasonMcneice\_Flag2\_4484
- We also have confirmed that this malware changes the registry, and sends a post request to a c2 server