Windows malware reversing

Alexandre-Xavier Labonté-Lamoureux & Thomas Dupuy

Why?

- Protection of customers
- Deep analysis about malware families
- Follow evolution of these families
- Collaboration with law enforcement

How?

- Static
- Dynamic

Tools

- IDA Pro
- JEB
- Ghidra
- Radare/Cutter
- Hopper
- Binary Ninja
- x64dbg
- GDB

Type of malware

- financial (ransomware, banker...)
- espionnage (keylogger, spyware...)
- disruption (DDos, wiper...)

Vectors of compromise

- email
- website
- document
- vulnerability
- ...

Identifying potential malicious functionalities

- persistence
- suspicious connections
- obfuscation (packer, control flow flattening…)

x86 assembly

Instruction	Meaning	Instruction	Meaning
mov dst, src	dst = src *(copy)*	imul dst, src	dst *= src
push src	push src to stack	idiv dst, src	dst /= src
pop dst	pop from stack to dst	and dst, src	dst &= src
lea dst, src	load src mem addr to dst	or dst, src	dst |= src
add dst, src	dst += src	xor dst, src	dst ^= src
sub dst, src	dst -= src	shl dst, src	dst <<= src
inc dst	dst++	shr dst, src	dst >>=src
dec dst	dst--	not dst	dst = -dst

Reference: https://www.cs.virginia.edu/~evans/cs216/guides/x86.html

x86 assembly code flow

Instruction	Meaning	Instruction	Meaning
jmp addr	jump to addr	cmp a, b	comparison
je/jz addr	jump if ==	call func	function call
jne/jnz addr	jump if !=	nop	does nothing
ja/jnbe/jg/jnle addr	jump if >		
jae/jnb/jge/jnl addr	jump if >=		
jb/jnae/jl/jnge addr	jump if <		
jbe/jna/jle/jng addr	jump if <=		

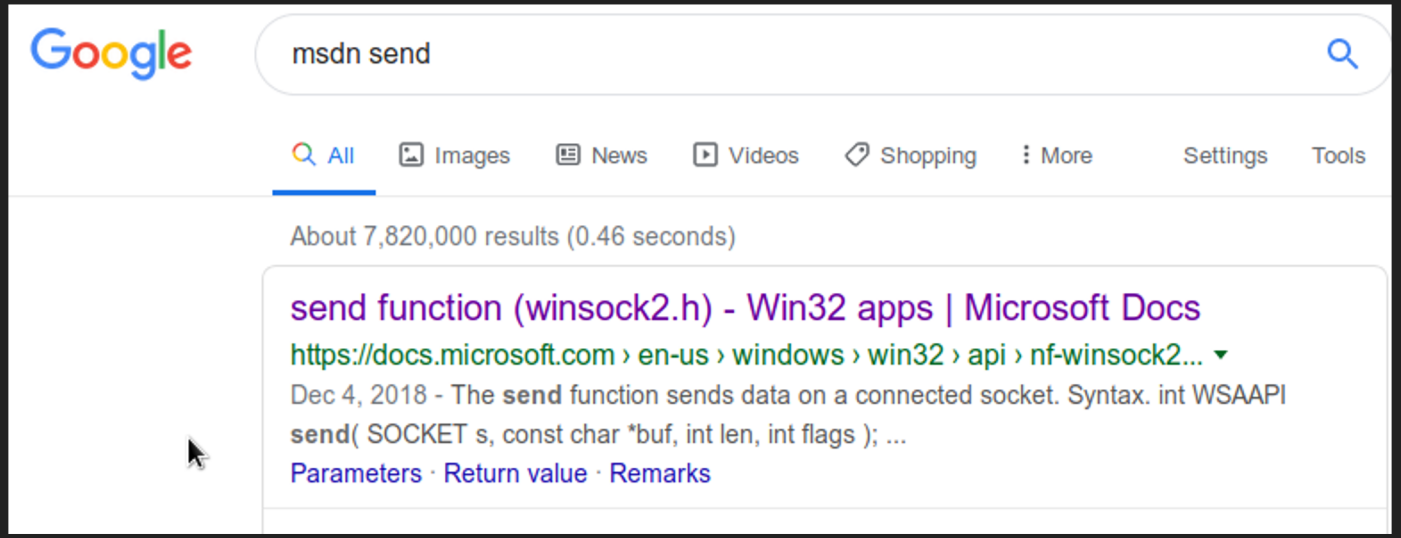
Reference: https://www.cs.virginia.edu/~evans/cs216/guides/x86.html

Windows API functions

- Filesystem
  - CreateFile
  - ReadFile, WriteFile
  - CreateFileMapping, MapViewOfFile
- Windows Registry
  - RegOpenKey
  - RegSetValue
  - RegGetValue
- Networking
  - WSAStartup
  - bind, listen
  - accept, connect
  - recv, send

Windows API functions

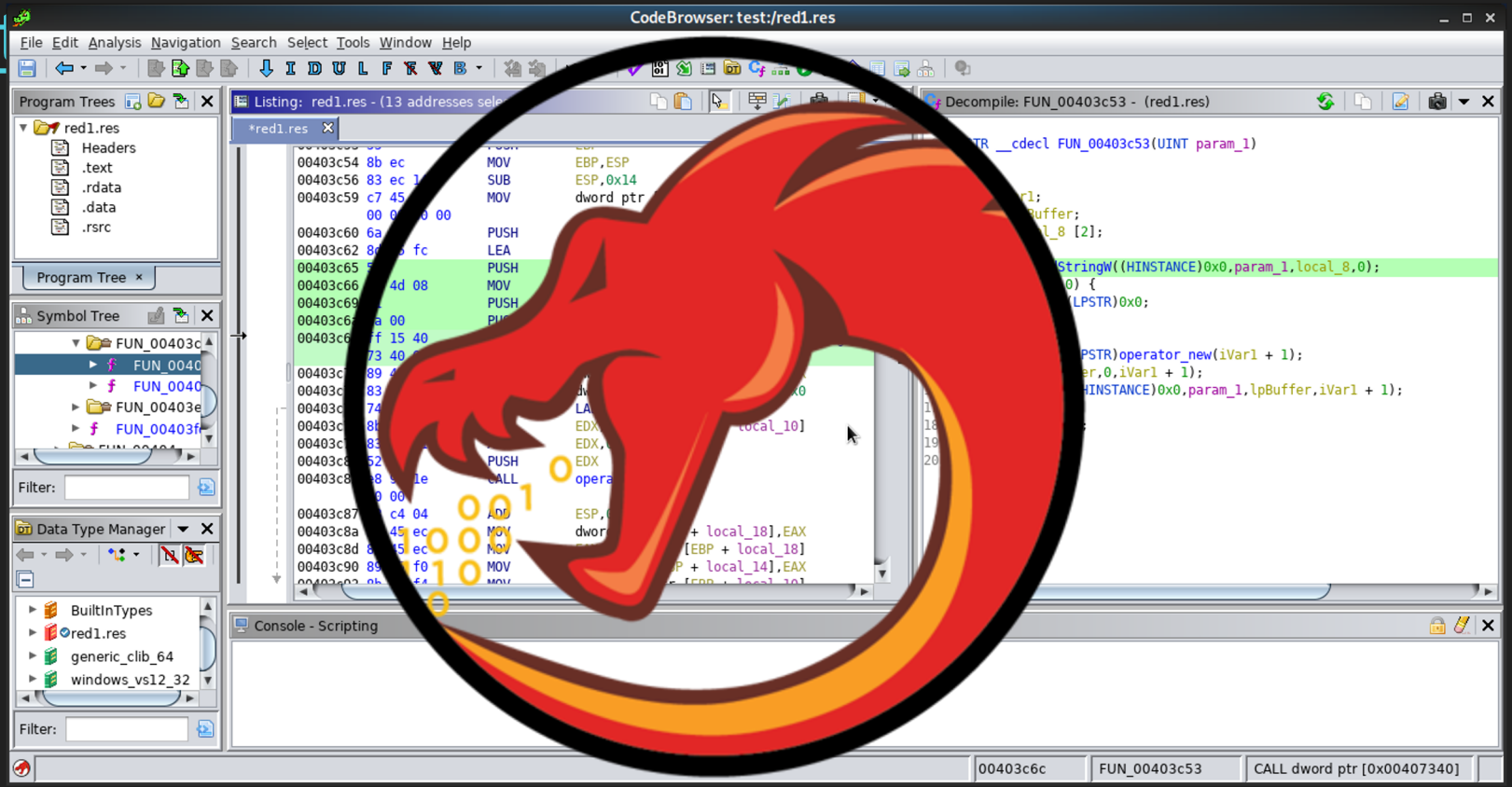
You can easily find what a Windows API function does by searching for it on MSDN



Ghidra

Download here:

htt



Working with Ghidra

How to start reverse engineering with Ghidra:

1. “File” menu → “New Project”
2. Select “Non-shared project”
3. Select a directory and project name
4. Your project was created. Now open the “File” menu and click “ImportFile”
5. Ghidra shows you info about the executable. Click “Ok” to continue.
6. Click on the executable’s name in the project folder. There will be a *cool* dragon animation.
7. Ghidra will ask if you want to analyze the file. Click “Yes”, then “Analyze”.

Working with Ghidra

Shortcuts

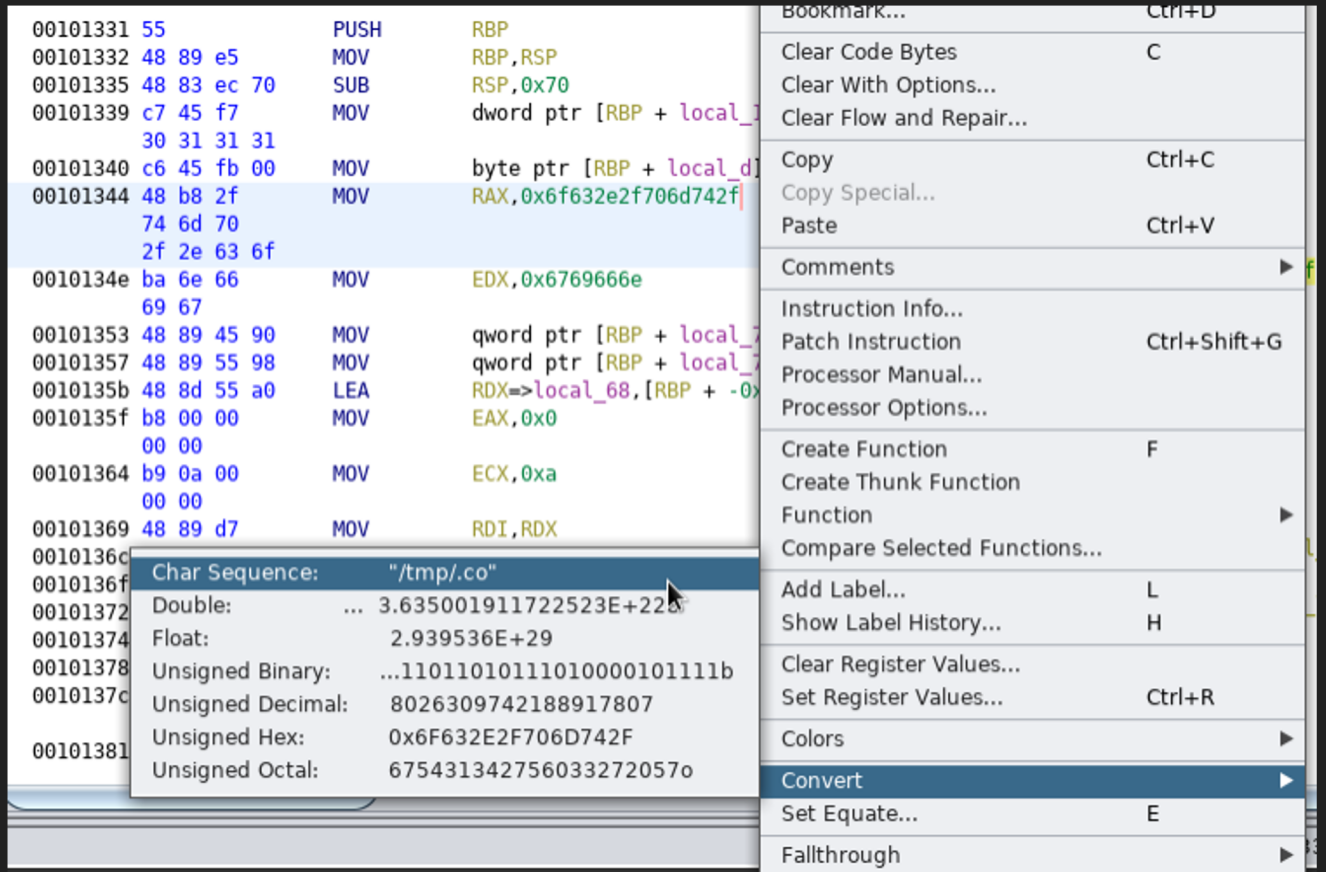
- “L”: Edit label, rename function or variable
- “;”: Add a comment
- “g”: Goto address
- “Ctrl+F”: Find
- “Ctrl+Z”: Undo
- “Ctrl+Shift+Z”: Redo
- “Alt+Left”: Go to previous location
- “Alt+Right”: Go to next location

Cheat sheet: https://ghidra-sre.org/CheatSheet.html

Working with Ghidra

Converting data

Note: You can only convert data via the disassembly listing



Working with Ghidra

Highlighting similar symbols

Click the middle mouse button

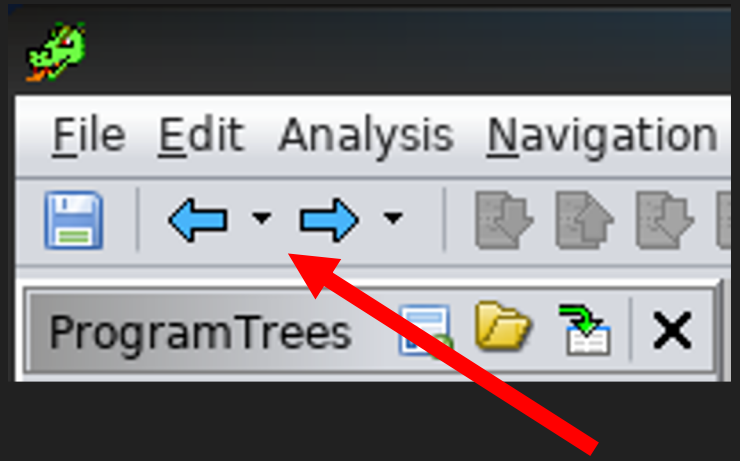
Note: You can change the button in: Edit -> ToolOptions -> ListingFields -> CursorTextHighlight

Working with Ghidra

Navigation

Go to your previous location or your next location.

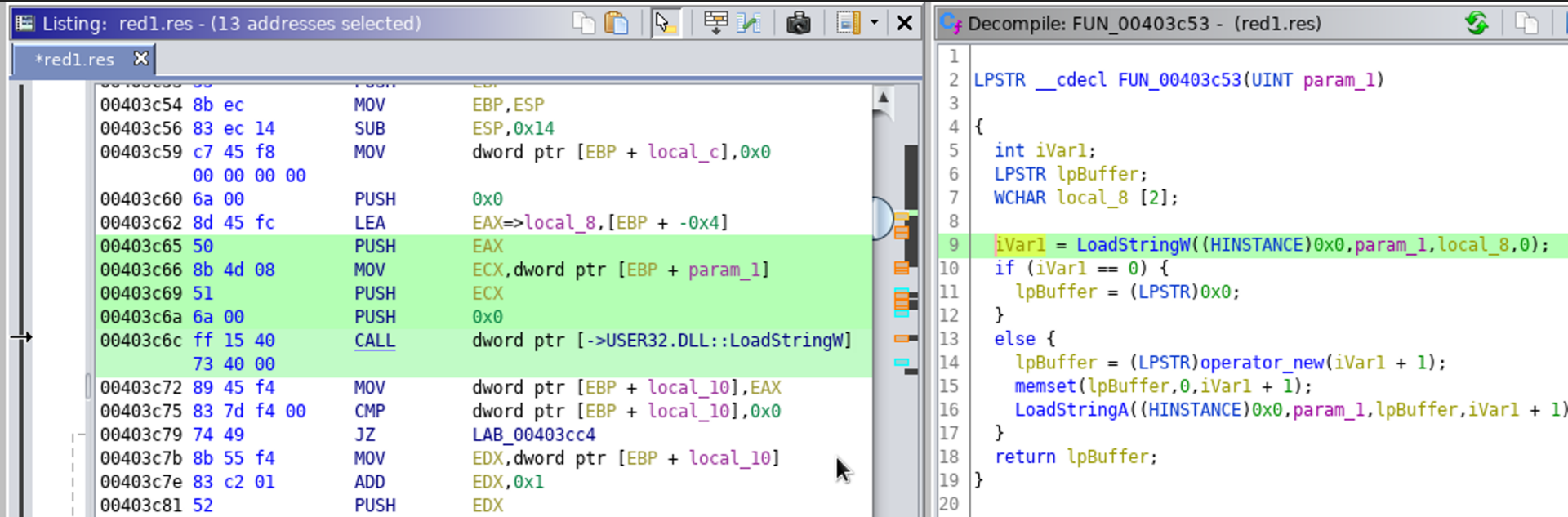
It’s like undo and redo, but it’s for the function that you were visiting.



Working with Ghidra

Matching assembly code with decompiled code

Highlight the code in one panel so the matching code highlights in the other panel.



Scenario

A customer sends you an email saying that he received a weird email with an attachment. He already opened it, but nothing happened. He is wondering if it's malicious or not.

You can find the document compressed (password: montrehack) here:

Link: http://fuckfuckgo.xyz/document.zip

```
85b556bb944bdd0ae8de86f099ba71231cf734fb  document.zip
a57e52449d041121f502220cb0d3e403a9f7909f  facture.doc
```

Questions

- What are the conditions for the malware to be executed?
- Can you see any mistakes from the operator?

Download the next stage… (please don't upload it on sharing platforms like VirusTotal, VirusShare, VirusBay...)

- What is the condition for the malware to be executed?
- What is the purpose of the function located at 0x402230?
- Starting at 0x401297, what does the malware do?
- How does the malware persists on the system?
- Explain what the malware does at 0x4015EB.

Document

link: http://fuckfuckgo.xyz/document.zip
85b556bb944bdd0ae8de86f099ba71231cf734fb document.zip
a57e52449d041121f502220cb0d3e403a9f7909f facture.doc

What are the conditions for the malware to be executed?
Can you see any mistakes from the operator?

Download the next stage… (please don't upload it on sharing platforms like VirusTotal, VirusShare, VirusBay...)

v1
link: TODO
a3116dac8de52f7e614a6327402c3e88af0d09ba ph.zip
ff9ba2fa2da1db170edad5223e98f21eecff9e07 v1

What is the condition for the malware to be executed?
What is the purpose of the function located at 0x402230?
Starting at 0x401297 what does the malware do?
How does the malware persists on the system?
Explain what the malware does at 0x4015EB.

Bonus (v2)
link: TODO

Unpack the binary.
Write a script to automate the task.

Bonus

TODO <link>

- Unpack the binary.
- Write a script to automatize the task.

Links

- https://www.welivesecurity.com/2019/08/08/varenyky-spambot-campaigns-france
- http://www.leparisien.fr/faits-divers/escroquerie-au-crypto-porno-un-pirate-doue-mais-instable-psychologiquement-14-12-2019-8217504.php
- https://krebsonsecurity.com/2019/12/nuclear-bot-author-arrested-in-sextortion-case/