Incident Response Threat Analysis

Prepared for

E Corp

SUNSHUTTLE – Go Malware

[28-Feb-2023]

SUMMARY

I made this report to present a Go malware analysis (and not in response to alerts from a security software/ use case asked by a customer). This report includes findings and recommended actions (Details about the analysis given in the annex).

FINDINGS

Attack Vector

For the samples analyzed, the infection vector is not known.

The alert originated from the following device:

1. Computer Name: {**Enter Device Name**}
2. IP Address: {Enter IP Address}
3. Assigned User: {Enter User's First name & Last name}
4. Date & Time of Event: {Enter date/time event occurred}
5. Last Seen Date/Time Stamp: {Enter the Last Logon Date/Time stamp}

This is what happened. The following action(s) caused the device to become compromised:

Action / Infection Vector	True? Yes	Comments
Browsing the Web		
Malicious link		
Browser Exploit		
File Download		
Clicking Malicious Link(s)		
Link in e-mail	N/A	
Link in file attachment		
Link in chat application		
Downloading Malware		
From chat application		
From e-mail attachment		
From removable media or USB disk		
From website		
Opening Malicious Attachment(s)/File(s)		
From e-mail		
From removable media or USB disk		

This table shows what we know.

Indicator	Present?	Notes
File Names	sunshuttle	NA
File Paths	N/A	
MD5 Hash	5DB340A70CB5D90601516DB89E629E43	
SHA 1 Hash	576BE6824FEE2F41767A039514EDB66C9002EB5	
SHA 256 Hash	478B04C20BBF6717D10EE978B99339B7C4664FEBC8BCFDAF86C3F0FBFC83A5C5	
Infection Vector	N/A	
Packer	UPX	
Language	Go	
Malware/Family	SUNSHUTTLE	
Setup of the program	1. Net-interfaces 2. main_define_internal_settings 3. main_GetMD5hash 4. base64_sdt_encoding 5. URL_base64_encoding	Anti-VM technique
File/DLL File	Config.dat.tmp	
URL	cdn.mxpnl.com code.jquery.com facebook.com nikeoutletinc.org twitter.com bing.com nikeoutletinc.org	decoy request decoy request decoy request decoy request decoy request decoy request C2 [the requests are passed in the cookies]
Scheduled Task	Backdoor [orders/request]	Detailed in "orders and requests"
cryptography	AES-256 in CFB mode encryption then two rounds of base64 encoding 1st round use the standard algorithm and the second one uses the custom alphabet "URL_SAFE". 2 GET requests are necessary to obtain a session key from the C2. The session key is encoded in transit: It's generated by the C2, and then provided to the backdoor to secure all future communications between the backdoor and the C2. This session key is encrypted with asymmetric cryptography AES_OAEP	AES-256 in CFB mode encryption + 2 rounds of base64 (std+custom) AES_OAEP
cookies		Hard-coded value:110 Request type Hard-coded value gWJpnph... Backdoor identifier

Orders from the c2	No operation: This cookie is missing, so the C2 try to send a new order to the backdoor until this cookie will be present Set a new min delay configuration value Enable/disable the decoy traffic Update the user-agent used in requests Send the victim machine local time Update the expiration timestamp download of the file from the c2 upload of the file from the c2 execute a Go command execute a Go command and send the output	MHcoJGpHMiluTS IJVG2jnYvnE jktstagnsC5pCmfuTM3 fbbCInzwmzFQZmDIgc AT18tInio8H7mM513 7c2poatKYsADKC7xxjBMb0m2 [timestamp] String|ID main_wget_file BAFsekJMz / main_send_file_part pCznRnh06xI5phVR Os_exec_command 9L2BCqKRM1z3Nf0Lt
Requests to the c2	Missing cookie – obtain an order to execute Creation of the session key Retrieve the session key Download a file from the C2 Upload a file to the C2 Send the command result to the C2	GC6BzibvrlkYoFr uWrGEdYLq 3qnehpTGrqr9x j9PQsNev gtpUjmrowl S4KUbT6H8NY

Behavior summary

Anti_VM [Net_interfaces]→ Create config file [Main_internal]→ call to time_Now→date conversion→launch legitimate traffic →→connection with the c2 [negociation of the session_key, derived to the date of the first launch/ will be used to encrypt the future communications]→main_beconing [main loop of the program] where the malware asks the orders to the C2, and execute them.

RECOMMENDED ACTIONS

First, make sure all your computers are running updated security solution.

Vulnerability	Comments
Malicious file(s)	**Possible Incident Response Steps:**
	1. Check the Findings section for a list of infected files. 2. Check the web proxy for similar files or hashes 3. Check AV solution for hashes 4. Run additional AV scans on machines involved in isolated event 5. Speak with user to see if there is a reoccurring theme, such as a repeated site visited. 6. Determine if that computer has had any other triggered events from other security products in the last 48 hours leading up to isolated event. 7. Remote administration services use strongly encrypted protocols and only accept connections from authorized users or locations. 8. Check an unusual increase of traffic send to legitimate sites defined by the request decoys

REFERENCES

This report may contain information that is available on the Internet. For more information, please refer to the following websites:

Title	Author	URLs	Date
Sunburst backdoor –code overlaps with Kazuar	KASPERSKY GReAT : Georgy Kucherin, Igor Kuznetsov, Costin RAIU	https://securelist.com/sunburst-backdoor-kazuar/99981/	11 Jan 2021
New SUNSHUTTLE Second-Stage Backdoor	Mandiant: Lindsay Smith, Jonathan Leathery, Ben Read	https://www.mandiant.com/resources/sunshuttle-second-stage-backdoor-targeting-us-based-entity	
MAR-10327841-1.v1 – SUNSHUTTLE	The Department of Homeland Security (DHS)	https://www.cisa.gov/uscert/ncas/analysis-reports/ar21-105a	15 apr 2021
Virustotal		https://www.virustotal.com/gui/file/478b04c20bbf6717d10ee978b99339b7c4664febc8bcfdaf86c3f0fbfc83a5c5/details [the hash submitted, not the file]	25 Oct 2021
		Other useful references – Team Sharing Information Purpose	**TYPE**
PE102	Corkami	https://raw.githubusercontent.com/corkami/pics/master/binary/pe102/pe102.svg	
Go Cryptography	Anish Nat	https://leanpub.com/cryptog https://play.google.com/store/books/details/Anish_Nath_Go_Lang_Cryptography?id=TxJ9DwAAQBAJ	book
GoReSym	Mandiant	https://github.com/mandiant/GoReSym (Not used but could be useful)	Go symbol parser

CONTACT US

For additional assistance, please contact Natacha BAKIR.

- **Phone number** – on demand
- **Email address** – on demand
- **GitHub**: Alphabot42

Annexes

Analyzing a Go Malware

Why Go?

More and more security tools are written in this language [Such as red teaming]
Threat actor also chose this language to develop their own tools.
There's been a 2,000% increase of new malware written in Go over the past few years [source: Intezer]
"Go supports an easy process for cross-platform compilation. This allows malware developers to write code once and compile binaries from the same codebase for multiple platforms [targeting Windows, Mac, and Linux from the same codebase]."

What analyzing Go malware is different?

There is no exception mechanism in Go, but functions can have multiple return values, which is important, because pure static analysis is inefficient. The Good news is that the standard library provided by Google is extensive and very well documented.

The approach

Try to rewrite the original scripts by tracking the library calls and their corresponding arguments.
When we encounter a function that we don't know, we will be able to look it up in the official documentation. Another point of interest is that strings and constants using the program get mashed together and put together in the binary.

Notes:

No prologue/No epilogue

The usual EB push ebp move ebp ESP is nowhere to be seen. Instead, you have this move RCX Gs 28. And then a comparison here to make sure that the stack is big enough. And if it's not, and you have this call to runtime.

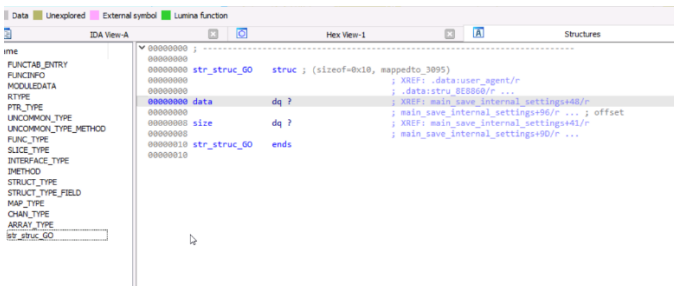
How the arguments are passed?

When the argument is passed, it is moved directly on the top of the stack. And then you have other arguments, that comes immediately after. So, with Go, arguments are not pushed or passed through registers, they are directly copied onto the stack at the correct position. there is no prologue and epilogue, the compiler knows exactly where on the stack those arguments are going to be.
So, a value , ESP+8, for instance is not equal to another esp+8, it depends on the value stored in ESP.

Methodology / Tips

That's the reason why, renaming variables really won't help you, because those variables are going to be used for very different things throughout the lifetime of the function. So instead, you can press Q on those to just have references to direct offsets on the stack, try to locate all the relevant API calls, put a breakpoint there, and check the returned value in the debugger, print ln, and we will just look at the arguments and look at the return values.

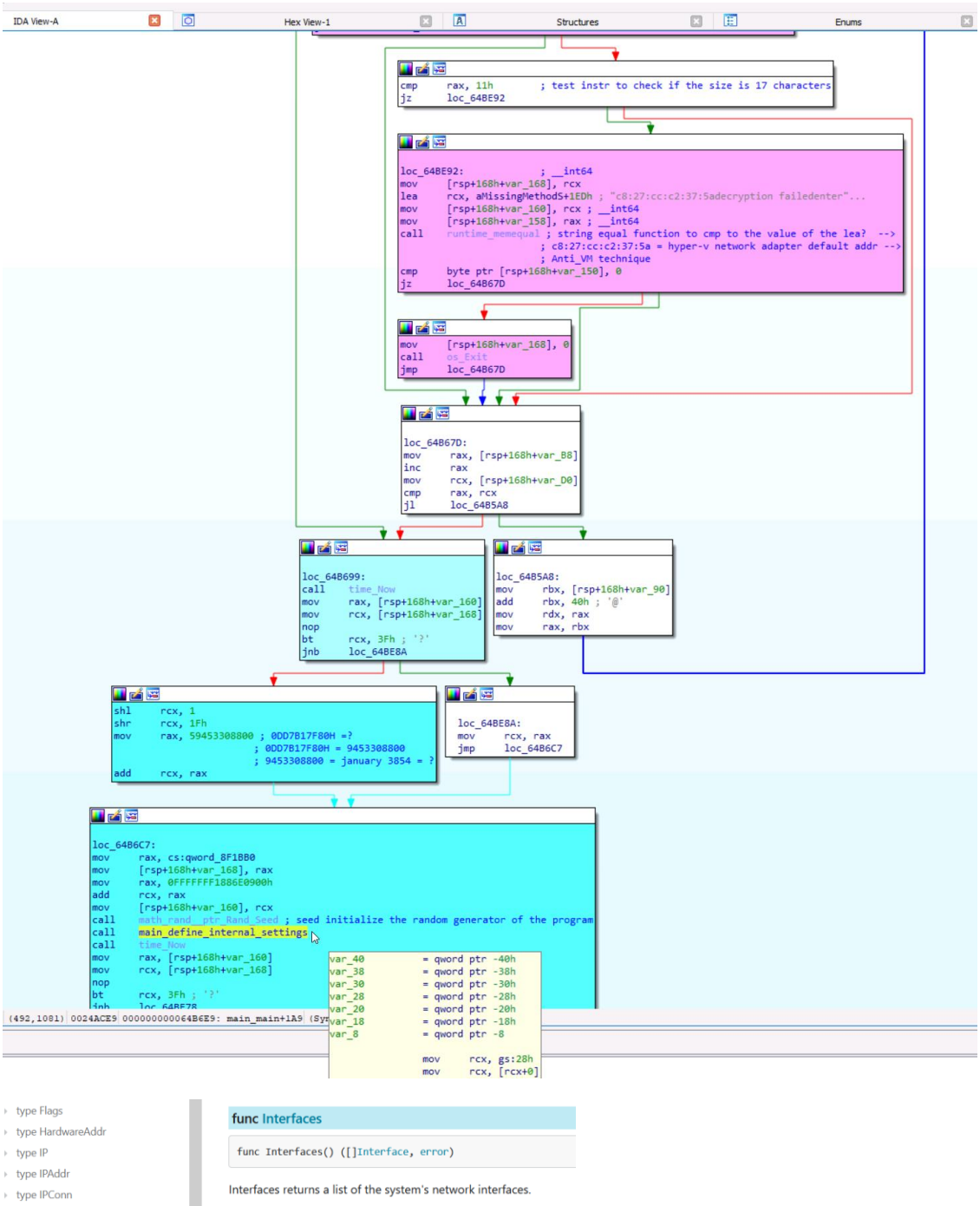
Note: Create the Go string structure to apply (data+size)



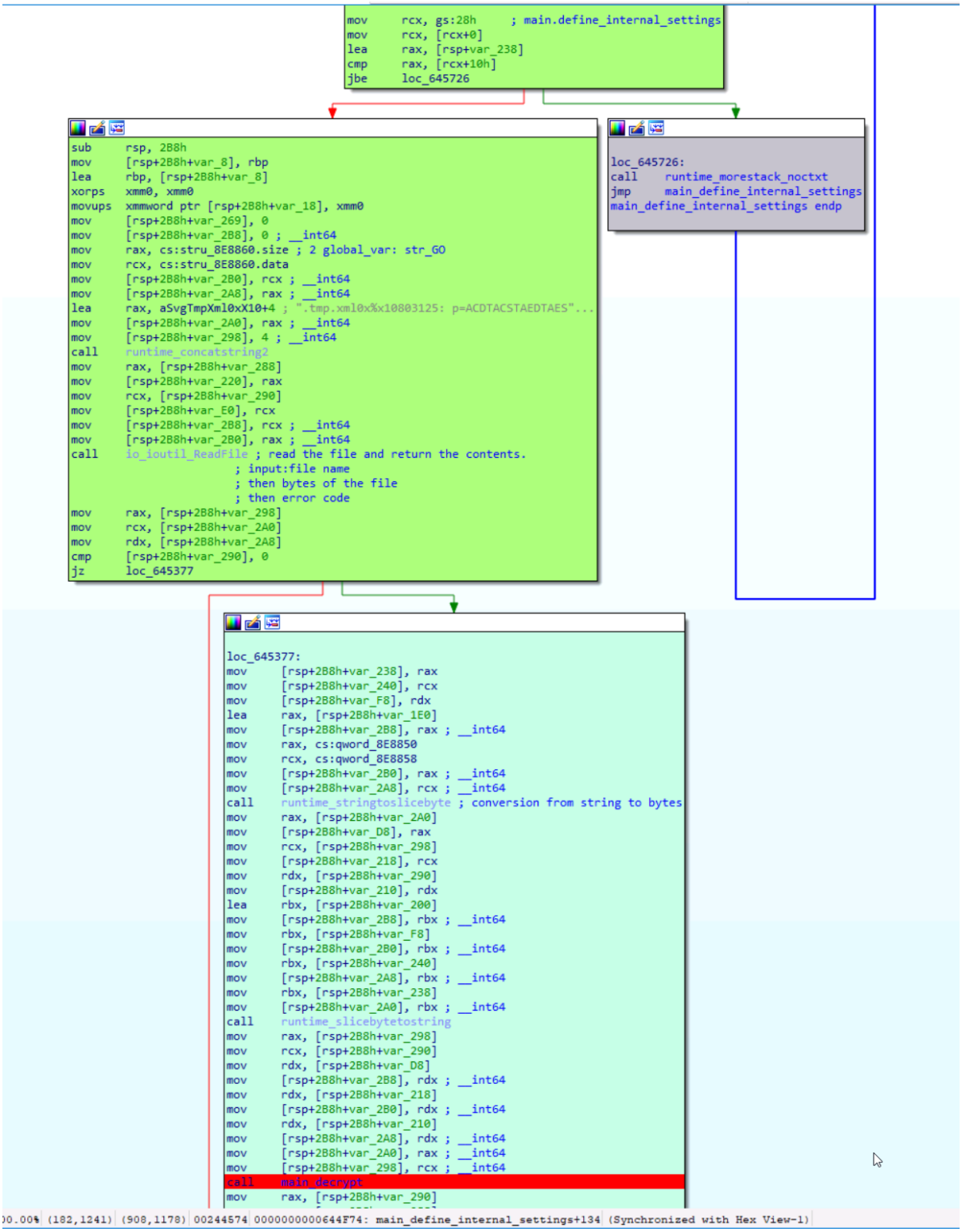
Screenshot of the analysis

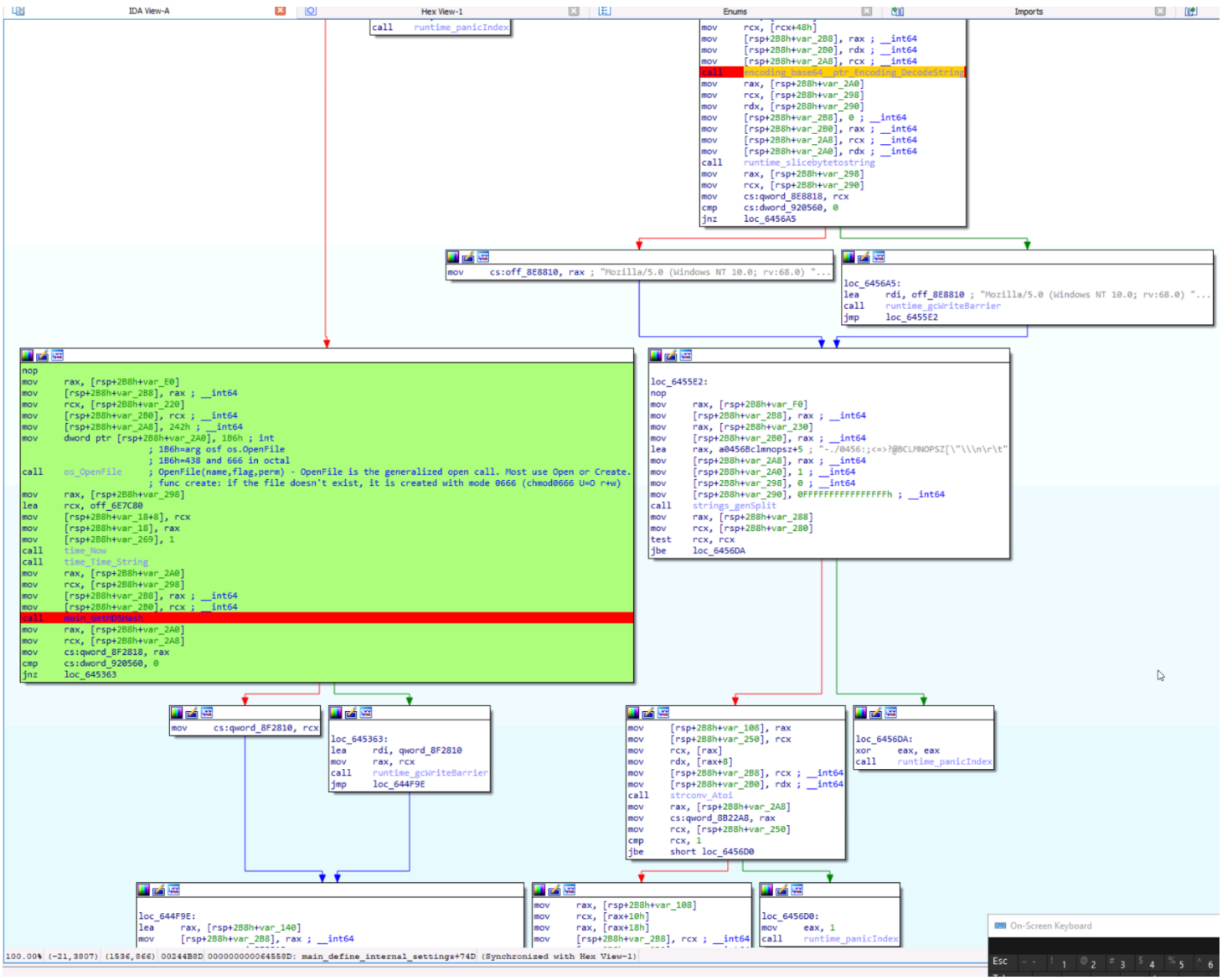
The functions analyzed are net_interfaces, main_define_internat_settings, main_GetMD5hash, main_encrypt, main_false_requesting, main_request_session_key, main_retrieve_session_key, URL_base64_encoding, Main_beconing, main_resolve_command, Main_decrypt, main_send_command, main_send_file

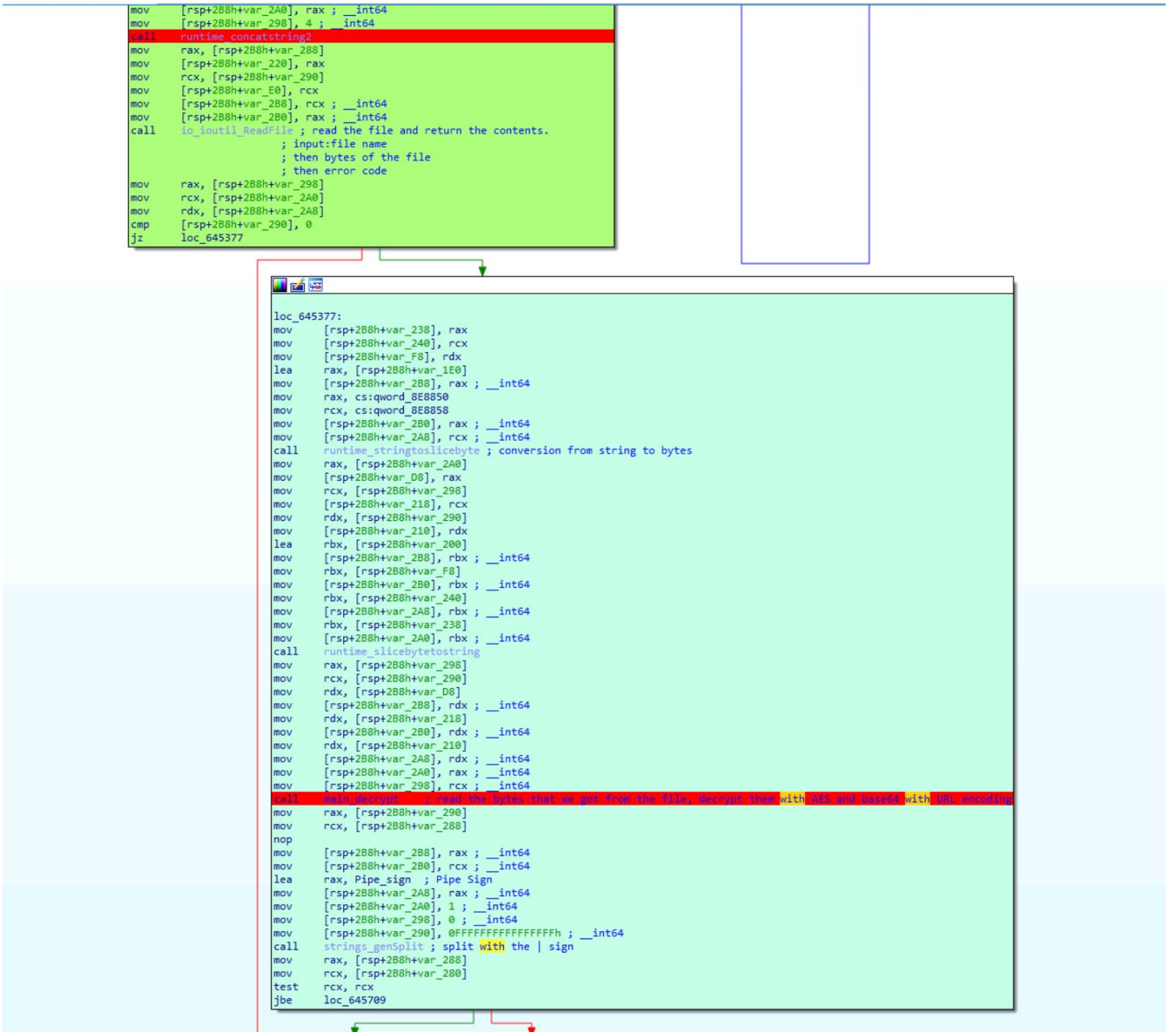
Net-interfaces



main_define_internal_settings



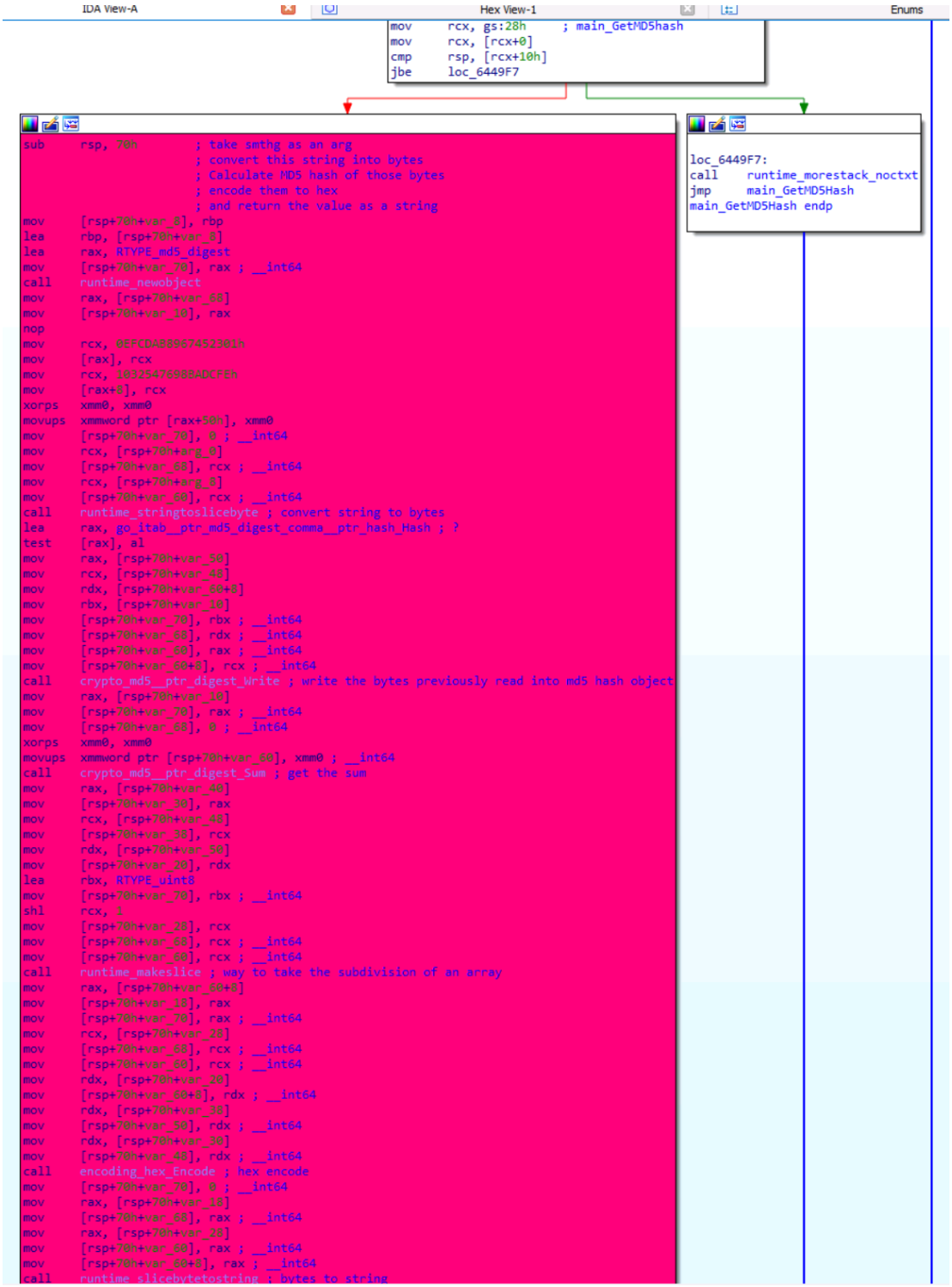

IDA View-A
Hex View-1
Enums
Imports
call runtime_panicIndex
mov rcx, [rcx+48h]
mov [rsp+2B8h+var_2B8], rax ; __int64
mov [rsp+2B8h+var_2B0], rdx ; __int64
mov [rsp+2B8h+var_2A8], rcx ; __int64
call encoding_base64__ptr_Encoding_DecodeString
mov rax, [rsp+2B8h+var_2A0]
mov rcx, [rsp+2B8h+var_298]
mov rdx, [rsp+2B8h+var_290]
mov [rsp+2B8h+var_2B8], 0 ; __int64
mov [rsp+2B8h+var_2B0], rax ; __int64
mov [rsp+2B8h+var_2A8], rcx ; __int64
mov [rsp+2B8h+var_2A0], rdx ; __int64
call runtime_slicebytetostring
mov rax, [rsp+2B8h+var_298]
mov rcx, [rsp+2B8h+var_290]
mov cs:qword_8E8818, rcx
cmp cs:dword_920560, 0
jnz loc_6456A5
mov cs:off_8E8810, rax ; "Mozilla/5.0 (Windows NT 10.0; rv:68.0) "...
loc_6456A5:
lea rdi, off_8E8810 ; "Mozilla/5.0 (Windows NT 10.0; rv:68.0) "...
call runtime_gcWriteBarrier
jmp loc_6455E2
nop
mov rax, [rsp+2B8h+var_E0]
mov [rsp+2B8h+var_2B8], rax ; __int64
mov rcx, [rsp+2B8h+var_220]
mov [rsp+2B8h+var_2B0], rcx ; __int64
mov [rsp+2B8h+var_2A8], 242h ; __int64
mov dword ptr [rsp+2B8h+var_2A0], 1B6h ; int
; 1B6h=arg osf os.OpenFile
; 1B6h=438 and 666 in octal
call os_OpenFile ; OpenFile(name,flag,perm) - OpenFile is the generalized open call. Most use Open or Create.
; func create: if the file doesn't exist, it is created with mode 0666 (chmod0666 U=O r+w)
mov rax, [rsp+2B8h+var_298]
lea rcx, off_6E7C80
mov [rsp+2B8h+var_18+8], rcx
mov [rsp+2B8h+var_18], rax
mov [rsp+2B8h+var_269], 1
call time_Now
call time_Time_String
mov rax, [rsp+2B8h+var_2A0]
mov rcx, [rsp+2B8h+var_298]
mov [rsp+2B8h+var_2B8], rax ; __int64
mov [rsp+2B8h+var_2B0], rcx ; __int64
call main_GetMD5Hash
mov rax, [rsp+2B8h+var_2A0]
mov rcx, [rsp+2B8h+var_2A8]
mov cs:qword_8F2818, rax
cmp cs:dword_920560, 0
jnz loc_645363
loc_6455E2:
nop
mov rax, [rsp+2B8h+var_F0]
mov [rsp+2B8h+var_2B8], rax ; __int64
mov rax, [rsp+2B8h+var_230]
mov [rsp+2B8h+var_2B0], rax ; __int64
lea rax, a0456bclmnopsz+5 ; "-./0456:;<=>?@BCLMNOPSZ[\"\\\n\r\t"
mov [rsp+2B8h+var_2A8], rax ; __int64
mov [rsp+2B8h+var_2A0], 1 ; __int64
mov [rsp+2B8h+var_298], 0 ; __int64
mov [rsp+2B8h+var_290], 0FFFFFFFFFFFFFFFFh ; __int64
call strings_genSplit
mov rax, [rsp+2B8h+var_2B8]
mov rcx, [rsp+2B8h+var_2B0]
test rcx, rcx
jbe loc_6456DA
mov cs:qword_8F2810, rcx
loc_645363:
lea rdi, qword_8F2810
mov rax, rcx
call runtime_gcWriteBarrier
jmp loc_644F9E
mov [rsp+2B8h+var_108], rax
mov [rsp+2B8h+var_250], rcx
mov rcx, [rax]
mov rdx, [rax+8]
mov [rsp+2B8h+var_2B8], rcx ; __int64
mov [rsp+2B8h+var_2B0], rdx ; __int64
call strconv_Atoi
mov rax, [rsp+2B8h+var_2A8]
mov cs:qword_8B22A8, rax
mov rcx, [rsp+2B8h+var_250]
cmp rcx, 1
jbe short loc_6456D0
loc_6456DA:
xor eax, eax
call runtime_panicIndex
loc_644F9E:
lea rax, [rsp+2B8h+var_140]
mov [rsp+2B8h+var_2B8], rax ; __int64
mov rax, [rsp+2B8h+var_108]
mov rcx, [rax+10h]
mov rax, [rax+18h]
mov [rsp+2B8h+var_2B8], rcx ; __int64
loc_6456D0:
mov eax, 1
call runtime_panicIndex
On-Screen Keyboard
100.00% (-21,3807) (1536,866) 00244B8D 000000000064558D: main_define_internal_settings+74D (Synchronized with Hex View-1)

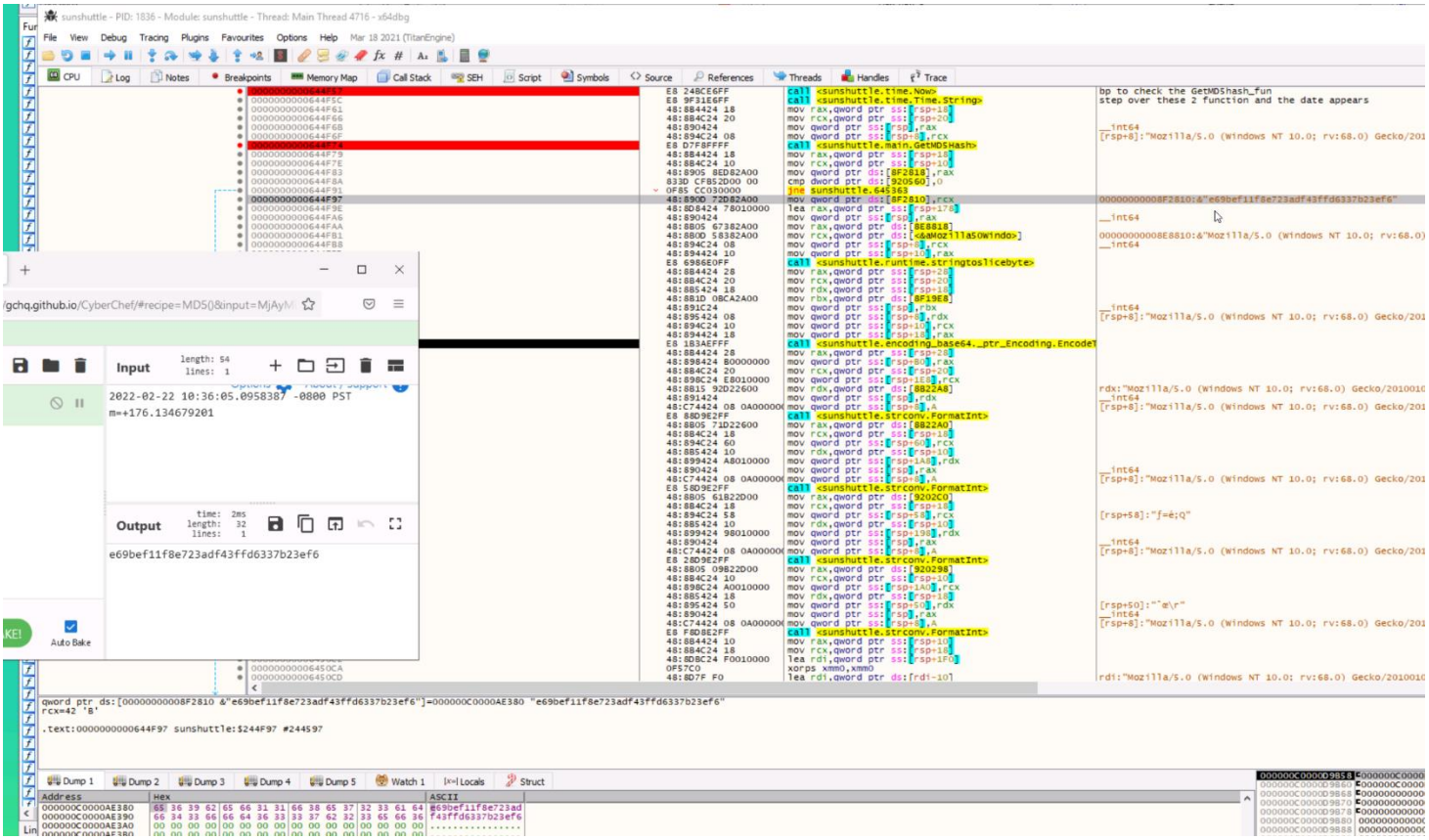


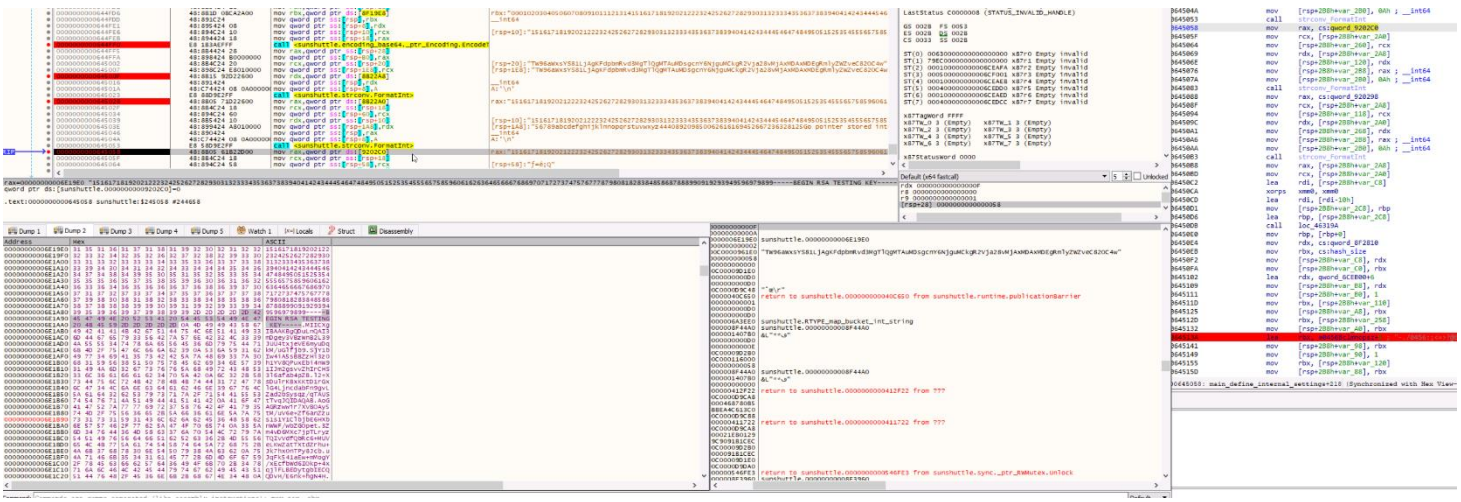
*The pipe sign is used to format the Sunshuttle's configuration

A|B|C|D|E
ID| delay_min-delay_max|main_false_requesting_enabled|expiration_timestamp|user_agent

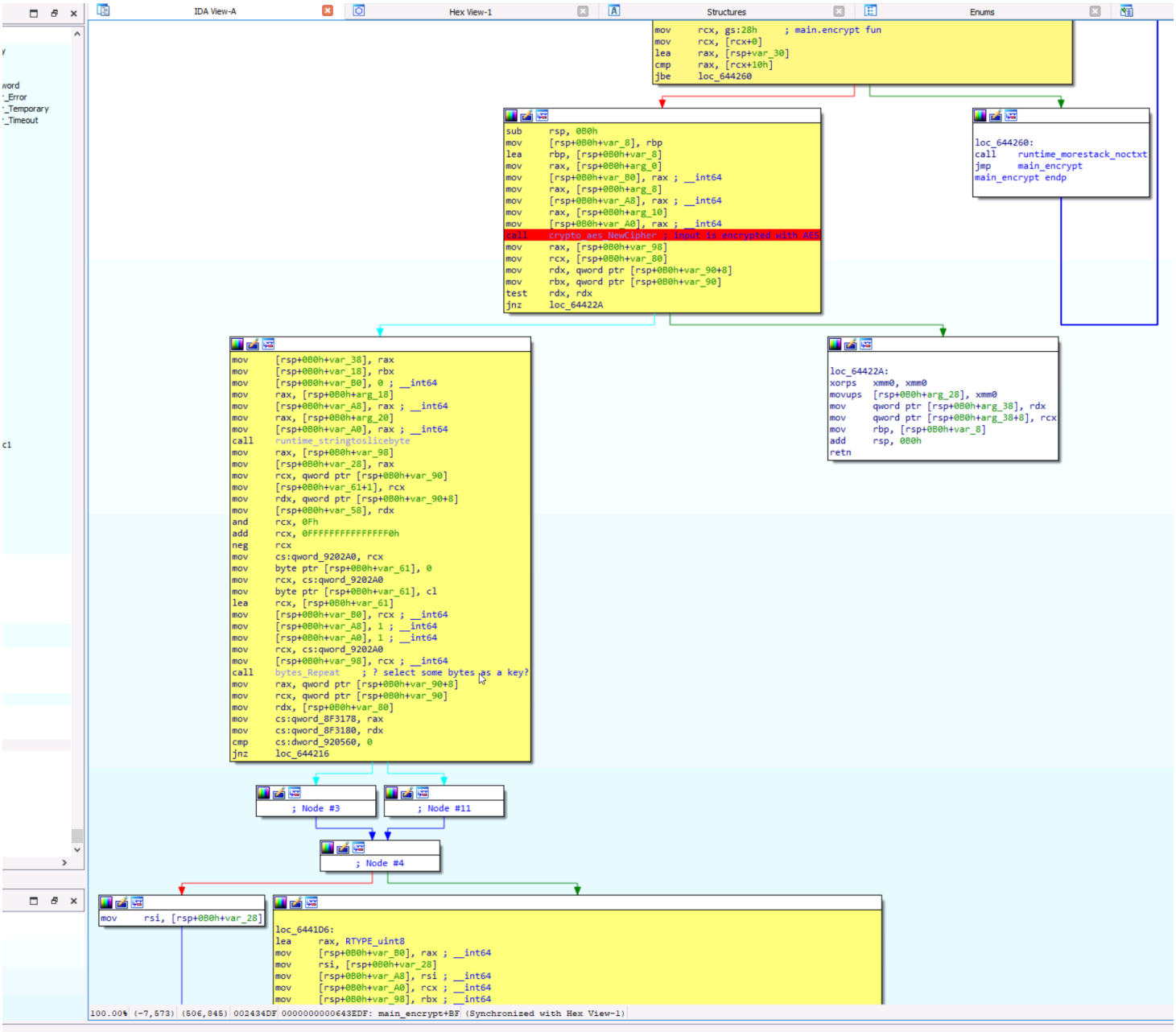
main_GetMD5hash

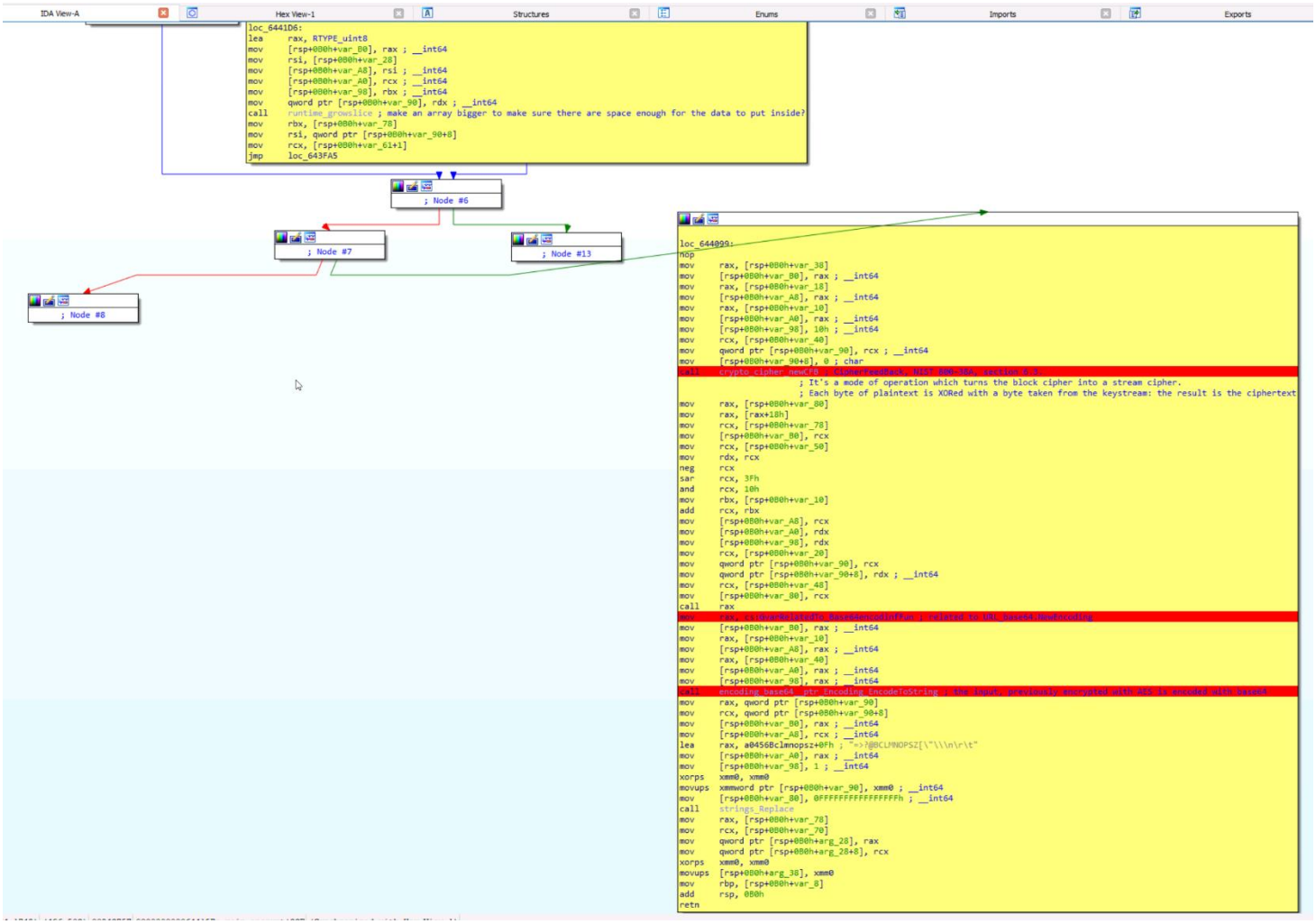


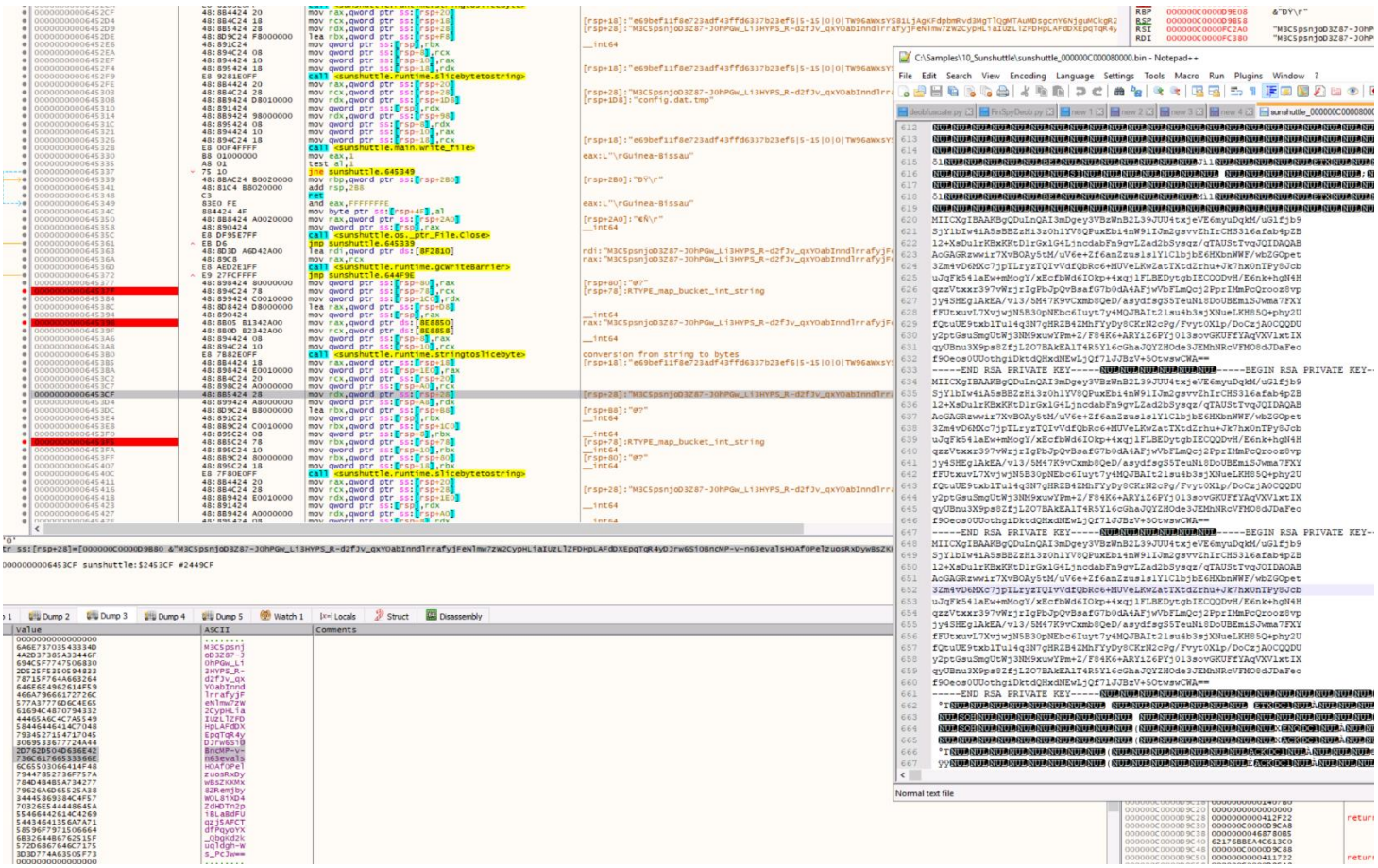




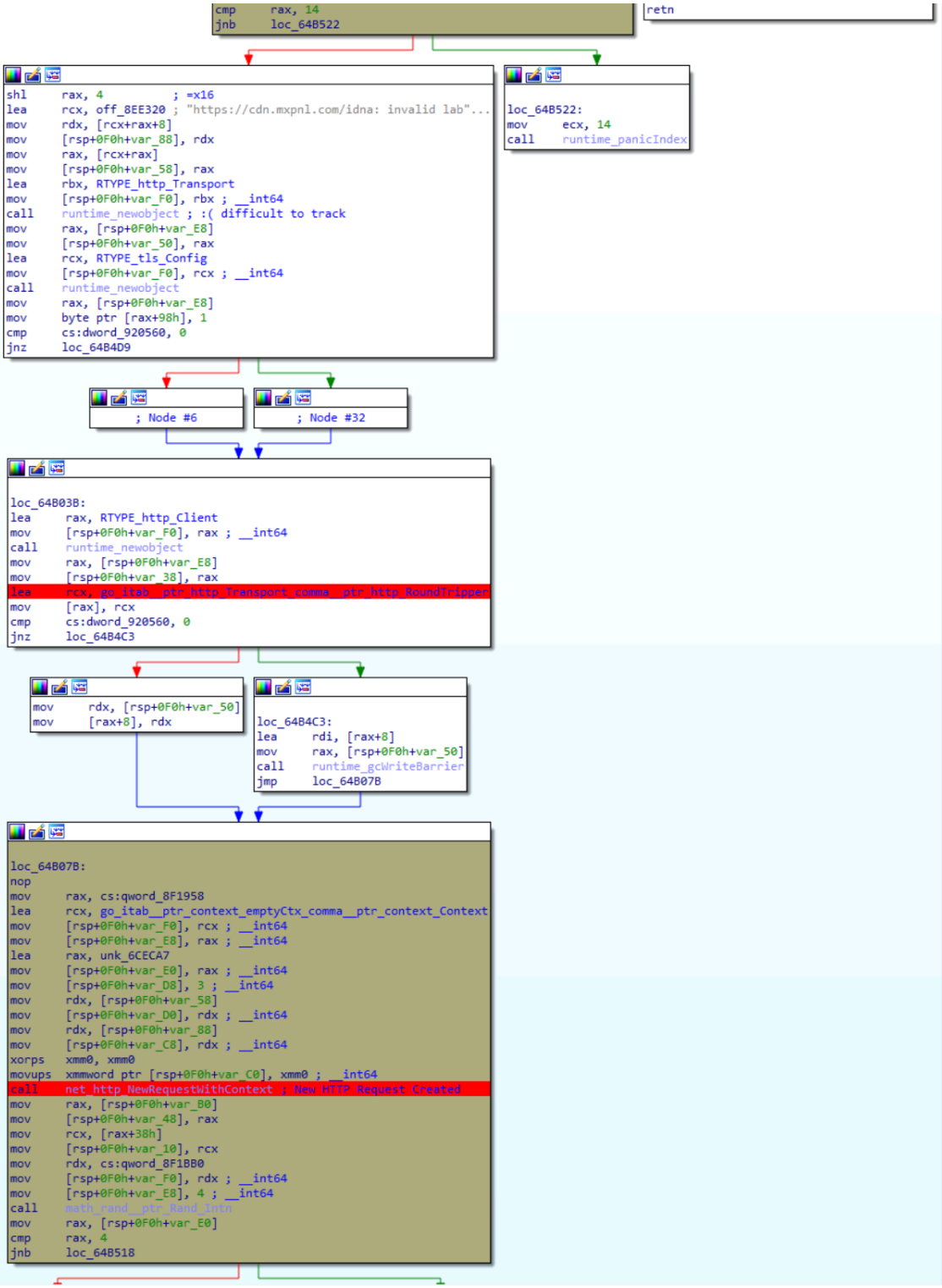
Main_encrypt

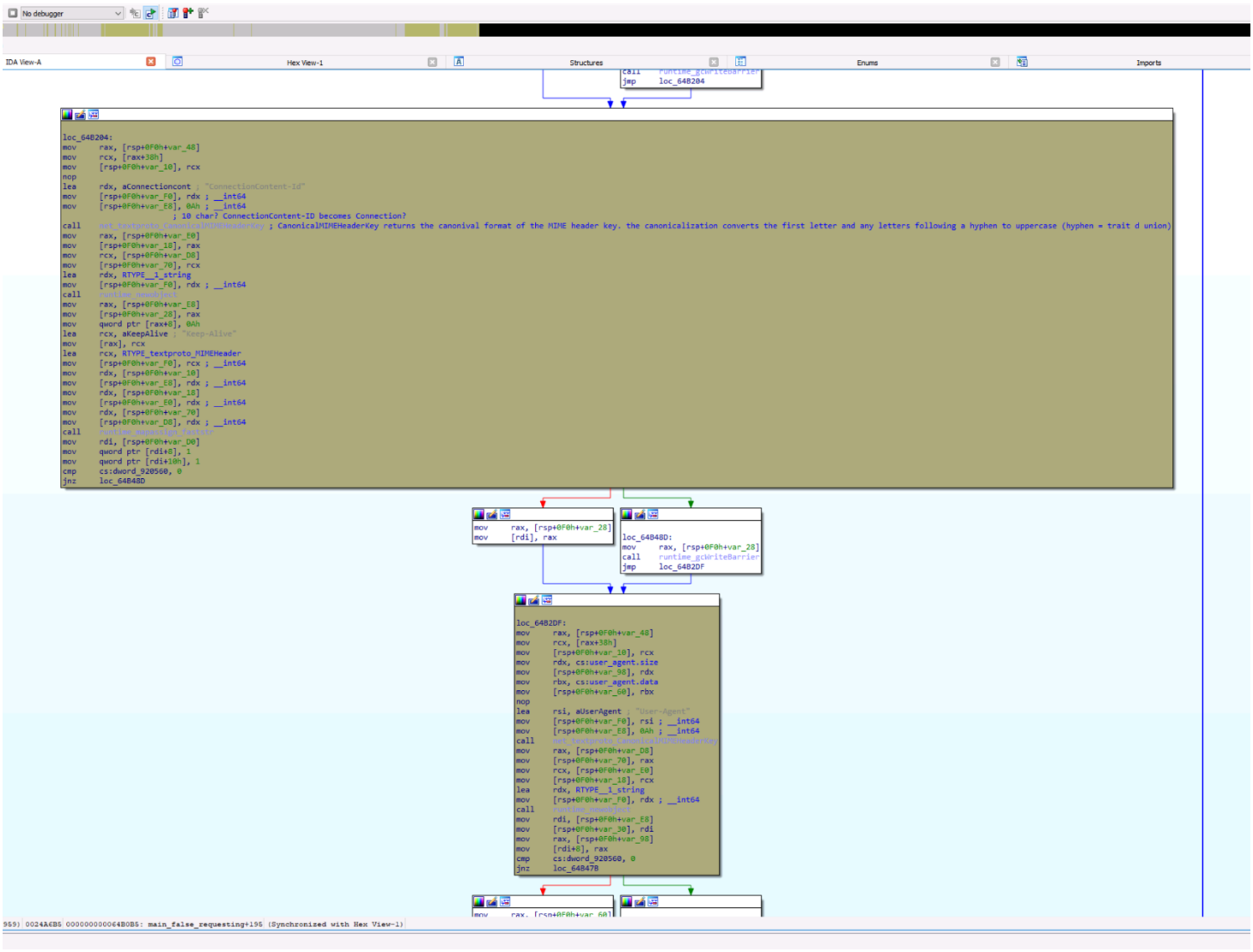


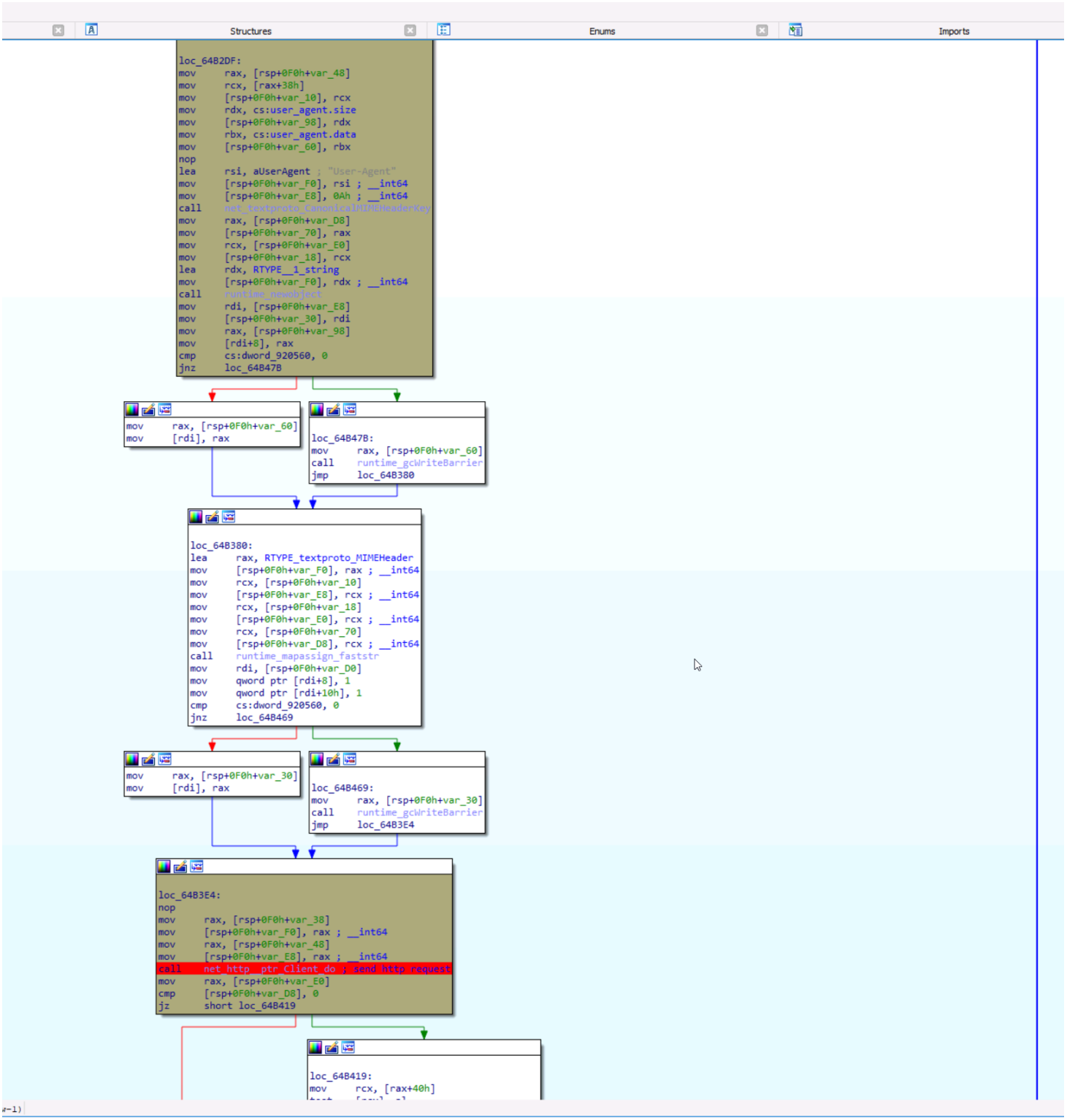

IDA View-A
Hex View-1
Structures
Enums
Imports
Exports
loc_6441D6:
lea rax, RTYPE_uint8
mov [rsp+080h+var_B0], rax ; __int64
mov rsi, [rsp+080h+var_28]
mov [rsp+080h+var_A8], rsi ; __int64
mov [rsp+080h+var_A0], rcx ; __int64
mov [rsp+080h+var_98], rbx ; __int64
mov qword ptr [rsp+080h+var_90], rdx ; __int64
call runtime_growslice ; make an array bigger to make sure there are space enough for the data to put inside?
mov rbx, [rsp+080h+var_78]
mov rsi, qword ptr [rsp+080h+var_90+8]
mov rcx, [rsp+080h+var_61+1]
jmp loc_643FA5
; Node #6
; Node #7
; Node #13
; Node #8
loc_644099:
nop
mov rax, [rsp+080h+var_38]
mov [rsp+080h+var_B0], rax ; __int64
mov rax, [rsp+080h+var_18]
mov [rsp+080h+var_A8], rax ; __int64
mov rax, [rsp+080h+var_10]
mov [rsp+080h+var_A0], rax ; __int64
mov [rsp+080h+var_98], 10h ; __int64
mov rcx, [rsp+080h+var_40]
mov qword ptr [rsp+080h+var_90], rcx ; __int64
mov [rsp+080h+var_90+8], 0 ; char
call crypto_cipher_newCFB ; CipherFeedBack, NIST 800-38a, section 6.3.
; It's a mode of operation which turns the block cipher into a stream cipher.
; Each byte of plaintext is XORed with a byte taken from the keystream: the result is the ciphertext
mov rax, [rsp+080h+var_80]
mov rax, [rax+18h]
mov rcx, [rsp+080h+var_78]
mov [rsp+080h+var_80], rcx
mov rcx, [rsp+080h+var_50]
mov rdx, rcx
neg rcx
sar rcx, 3Fh
and rcx, 10h
mov rbx, [rsp+080h+var_10]
add rcx, rbx
mov [rsp+080h+var_A8], rcx
mov [rsp+080h+var_A0], rdx
mov [rsp+080h+var_98], rdx
mov rcx, [rsp+080h+var_20]
mov qword ptr [rsp+080h+var_90], rcx
mov qword ptr [rsp+080h+var_90+8], rdx ; __int64
mov rcx, [rsp+080h+var_48]
mov [rsp+080h+var_80], rcx
call rax
mov rax, xsidvackmlstedlo_Base64EncodingFun ; related to URL_base64.NewEncoding
mov [rsp+080h+var_B0], rax ; __int64
mov rax, [rsp+080h+var_10]
mov [rsp+080h+var_A8], rax ; __int64
mov rax, [rsp+080h+var_40]
mov [rsp+080h+var_A0], rax ; __int64
mov [rsp+080h+var_98], rax ; __int64
call encoding_base64__ptr_Encoding_EncodeToString ; the input, previously encrypted with AES is encoded with base64
mov rax, qword ptr [rsp+080h+var_90]
mov rcx, qword ptr [rsp+080h+var_90+8]
mov [rsp+080h+var_B0], rax ; __int64
mov [rsp+080h+var_A8], rcx ; __int64
lea rax, a0456bclmnopsz+0Fh ; "~>?@CLMNOPSZ[\"\\\n\r\t"
mov [rsp+080h+var_A0], rax ; __int64
mov [rsp+080h+var_98], 1 ; __int64
xorps xmm0, xmm0
movups xmmword ptr [rsp+080h+var_90], xmm0 ; __int64
mov [rsp+080h+var_80], 0FFFFFFFFFFFFFFFFh ; __int64
call strings_Replace
mov rax, [rsp+080h+var_78]
mov rcx, [rsp+080h+var_70]
mov qword ptr [rsp+080h+arg_28], rax
mov qword ptr [rsp+080h+arg_28+8], rcx
xorps xmm0, xmm0
movups [rsp+080h+arg_30], xmm0
mov rbp, [rsp+080h+var_8]
add rsp, 080h
retn

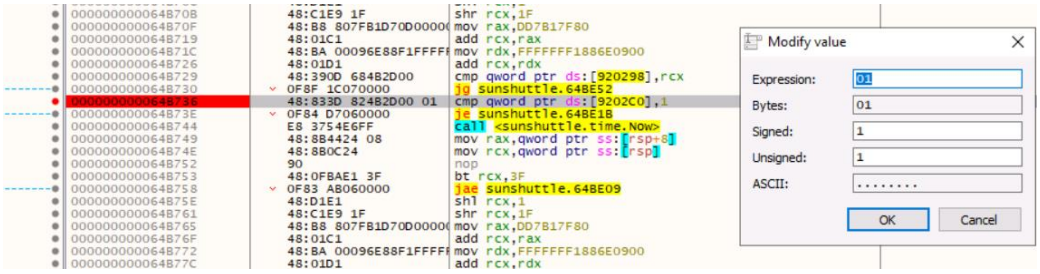


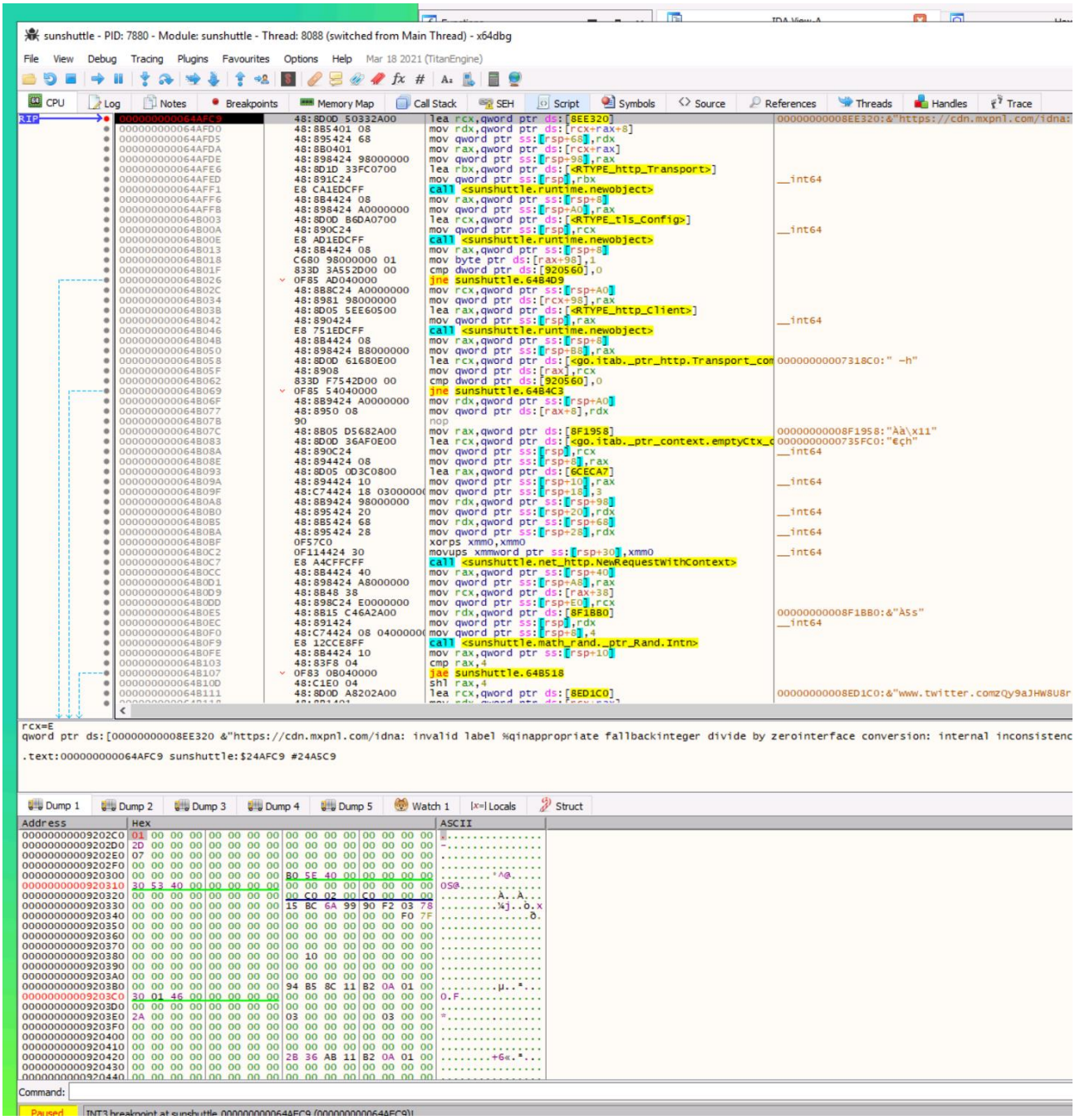
Main_false_requesting and C2 server




No debugger
IDA View-A
Hex View-1
Structures
Enums
Imports
loc_64B204:
mov rax, [rsp+0F0h+var_48]
mov rcx, [rax+38h]
mov [rsp+0F0h+var_10], rcx
nop
lea rdx, aConnectioncont ; "ConnectionContent-Id"
mov [rsp+0F0h+var_F0], rdx ; __int64
mov [rsp+0F0h+var_E8], 0Ah ; __int64
; 10 char? ConnectionContent-ID becomes Connection?
call net_textproto_CanonicalMIMEHeaderKey ; CanonicalMIMEHeaderKey returns the canonival format of the MINE header key. the canonicalization converts the first letter and any letters following a hyphen to uppercase (hyphen = trait d union)
mov rax, [rsp+0F0h+var_E0]
mov [rsp+0F0h+var_18], rax
mov rcx, [rsp+0F0h+var_D8]
mov [rsp+0F0h+var_70], rcx
lea rdx, RTYPE__1_string
mov [rsp+0F0h+var_F0], rdx ; __int64
call runtime_newobject
mov rax, [rsp+0F0h+var_E8]
mov [rsp+0F0h+var_28], rax
mov qword ptr [rax+8], 0Ah
lea rcx, aKeepAlive ; "Keep-Alive"
mov [rax], rcx
lea rcx, RTYPE_textproto_MIMEHeader
mov [rsp+0F0h+var_F0], rcx ; __int64
mov rdx, [rsp+0F0h+var_10]
mov [rsp+0F0h+var_E8], rdx ; __int64
mov rdx, [rsp+0F0h+var_18]
mov [rsp+0F0h+var_E0], rdx ; __int64
mov rdx, [rsp+0F0h+var_70]
mov [rsp+0F0h+var_D8], rdx ; __int64
call runtime_mapassign_fast64
mov rdi, [rsp+0F0h+var_D0]
mov qword ptr [rdi+8], 1
mov qword ptr [rdi+10h], 1
cmp cs:dword_920560, 0
jnz loc_64B48D
mov rax, [rsp+0F0h+var_28]
mov [rdi], rax
loc_64B48D:
mov rax, [rsp+0F0h+var_28]
call runtime_gcWriteBarrier
jmp loc_64B2DF
loc_64B2DF:
mov rax, [rsp+0F0h+var_48]
mov rcx, [rax+38h]
mov [rsp+0F0h+var_10], rcx
mov rdx, cs:user_agent.size
mov [rsp+0F0h+var_90], rdx
mov rbx, cs:user_agent.data
mov [rsp+0F0h+var_60], rbx
nop
lea rsi, aUserAgent ; "User-Agent"
mov [rsp+0F0h+var_F0], rsi ; __int64
mov [rsp+0F0h+var_E8], 0Ah ; __int64
call net_textproto_CanonicalMIMEHeaderKey
mov rax, [rsp+0F0h+var_D8]
mov [rsp+0F0h+var_70], rax
mov rcx, [rsp+0F0h+var_E0]
mov [rsp+0F0h+var_18], rcx
lea rdx, RTYPE__1_string
mov [rsp+0F0h+var_F0], rdx ; __int64
call runtime_newobject
mov rdi, [rsp+0F0h+var_E8]
mov [rsp+0F0h+var_30], rdi
mov rax, [rsp+0F0h+var_90]
mov [rdi+8], rax
cmp cs:dword_920560, 0
jnz loc_64B47B
00024A6B5 0000000064B0B5: main_false_requesting+195 (Synchronized with Hex View-1)


Structures
Enums
Imports
loc_64B2DF:
mov rax, [rsp+0F0h+var_48]
mov rcx, [rax+38h]
mov [rsp+0F0h+var_10], rcx
mov rdx, cs:user_agent.size
mov [rsp+0F0h+var_98], rdx
mov rbx, cs:user_agent.data
mov [rsp+0F0h+var_60], rbx
nop
lea rsi, aUserAgent ; "User-Agent"
mov [rsp+0F0h+var_F0], rsi ; __int64
mov [rsp+0F0h+var_E8], 0Ah ; __int64
call net_textproto_CanonicalMIMEHeaderKey
mov rax, [rsp+0F0h+var_D8]
mov [rsp+0F0h+var_70], rax
mov rcx, [rsp+0F0h+var_E0]
mov [rsp+0F0h+var_18], rcx
lea rdx, RTYPE__1_string
mov [rsp+0F0h+var_F0], rdx ; __int64
call runtime_newobject
mov rdi, [rsp+0F0h+var_E8]
mov [rsp+0F0h+var_30], rdi
mov rax, [rsp+0F0h+var_98]
mov [rdi+8], rax
cmp cs:dword_920560, 0
jnz loc_64B47B
mov rax, [rsp+0F0h+var_60]
mov [rdi], rax
loc_64B47B:
mov rax, [rsp+0F0h+var_60]
call runtime_gcWriteBarrier
jmp loc_64B380
loc_64B380:
lea rax, RTYPE_textproto_MIMEHeader
mov [rsp+0F0h+var_F0], rax ; __int64
mov rcx, [rsp+0F0h+var_10]
mov [rsp+0F0h+var_E8], rcx ; __int64
mov rcx, [rsp+0F0h+var_18]
mov [rsp+0F0h+var_E0], rcx ; __int64
mov rcx, [rsp+0F0h+var_70]
mov [rsp+0F0h+var_D8], rcx ; __int64
call runtime_mapassign_faststr
mov rdi, [rsp+0F0h+var_D0]
mov qword ptr [rdi+8], 1
mov qword ptr [rdi+10h], 1
cmp cs:dword_920560, 0
jnz loc_64B469
mov rax, [rsp+0F0h+var_30]
mov [rdi], rax
loc_64B469:
mov rax, [rsp+0F0h+var_30]
call runtime_gcWriteBarrier
jmp loc_64B3E4
loc_64B3E4:
nop
mov rax, [rsp+0F0h+var_38]
mov [rsp+0F0h+var_F0], rax ; __int64
mov rax, [rsp+0F0h+var_48]
mov [rsp+0F0h+var_E8], rax ; __int64
call net_http_ptr_Client_do ; send http request
mov rax, [rsp+0F0h+var_E0]
cmp [rsp+0F0h+var_D8], 0
jz short loc_64B419
loc_64B419:
mov rcx, [rax+40h]

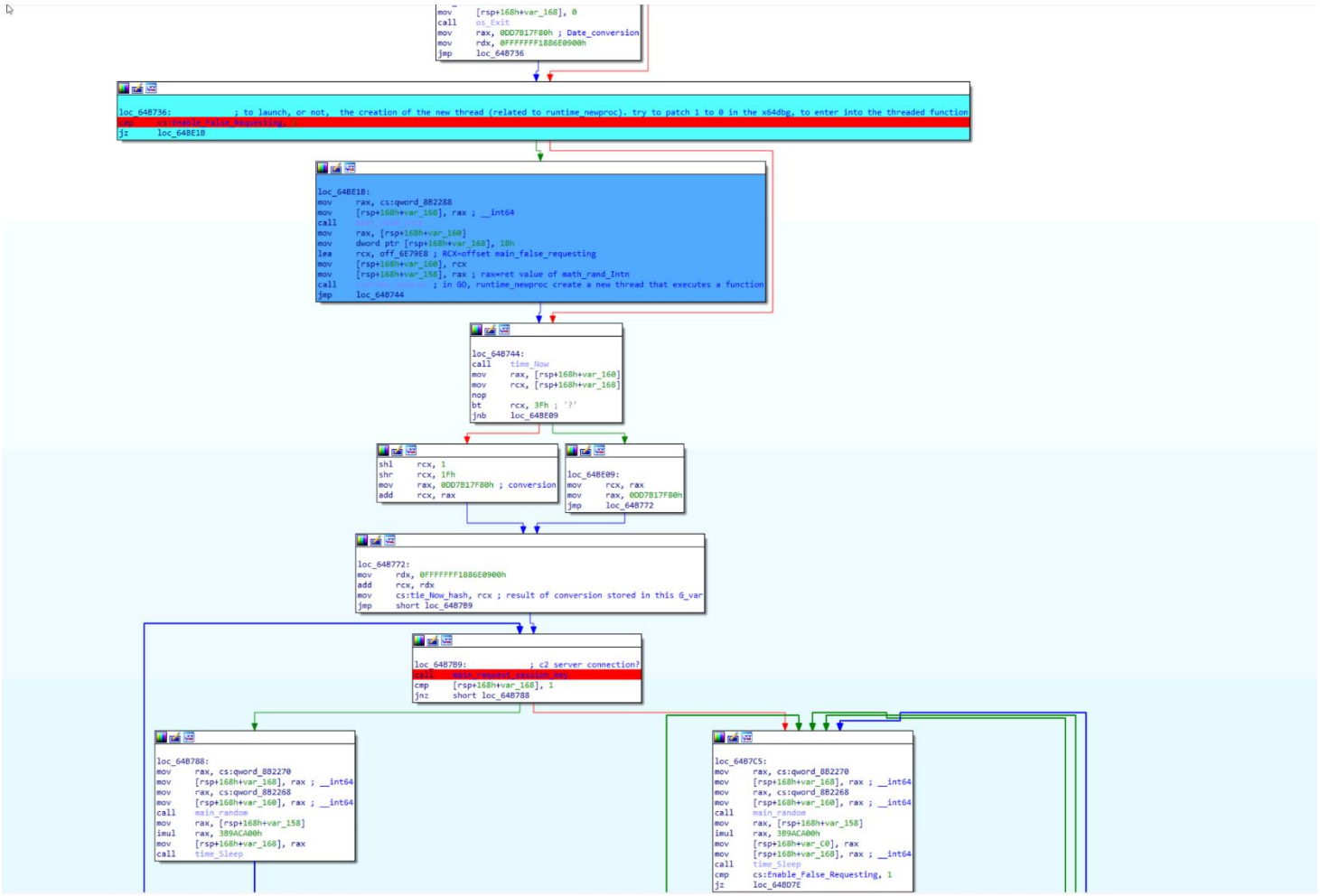

Modify value
Expression:
01
Bytes:
01
Signed:
1
Unsigned:
1
ASCII:
.....
OK
Cancel


sunshuttle - PID: 7880 - Module: sunshuttle - Thread: 8088 (switched from Main Thread) - x64dbg
File View Debug Tracing Plugins Favourites Options Help Mar 18 2021 (TitanEngine)
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Handles Trace
Dump 1 Dump 2 Dump 3 Dump 4 Dump 5 Watch 1 Locals Struct
Command:

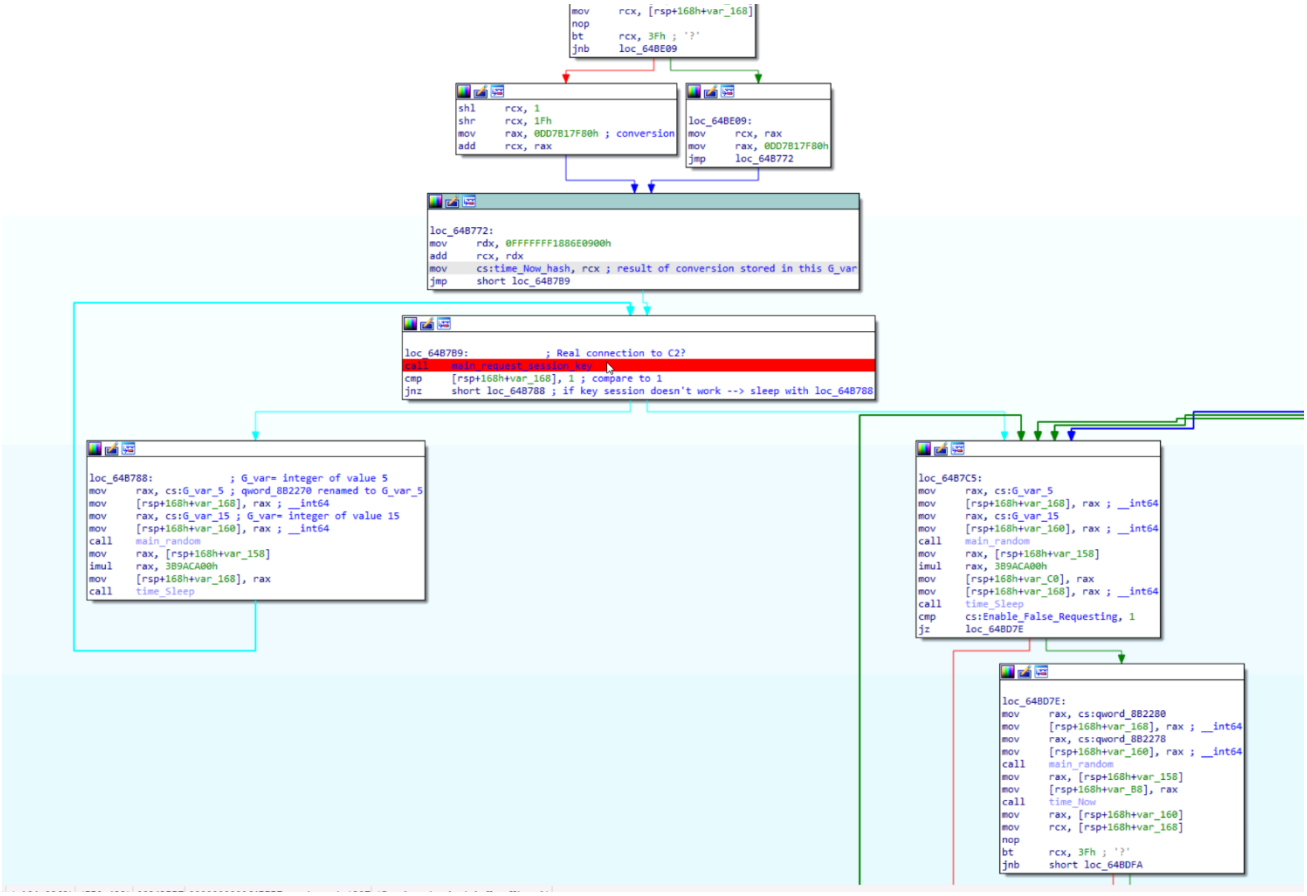
C2:

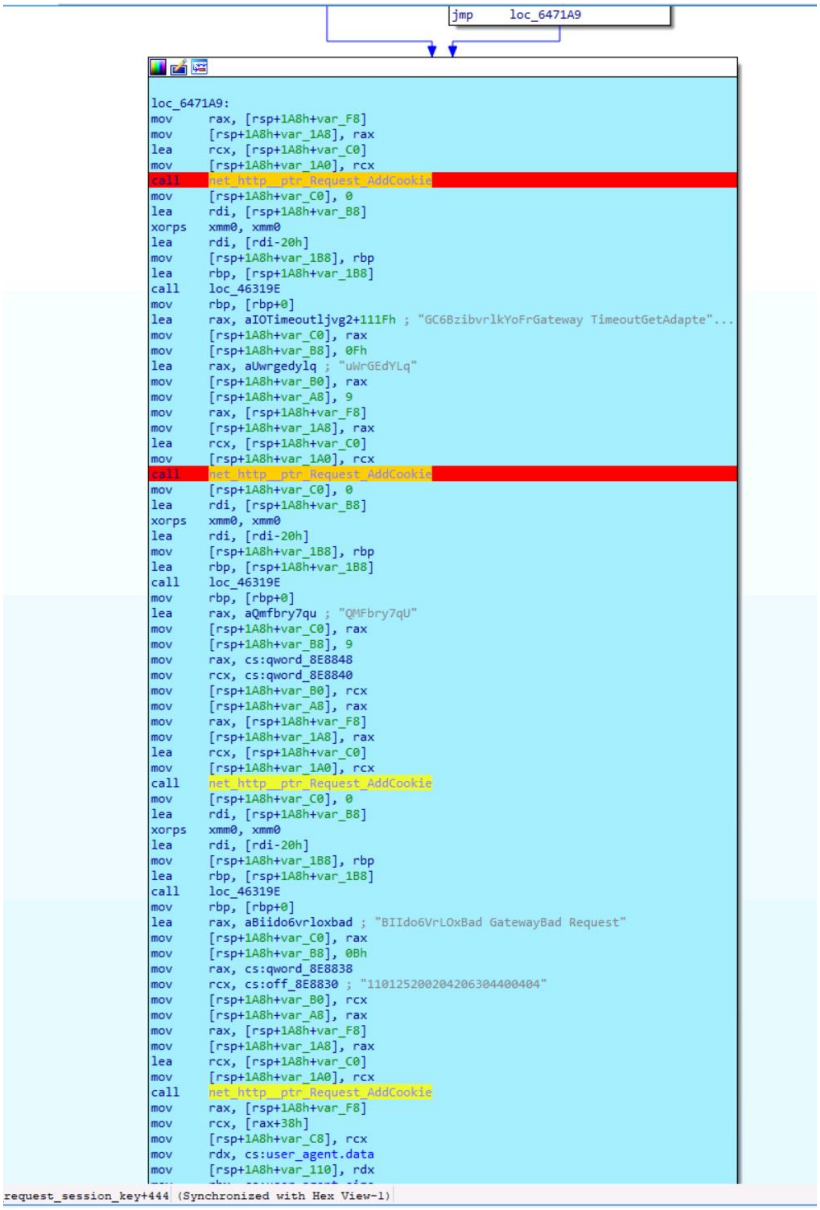
```
0000000000000005
00000C00011C5A0
000000000000015
```

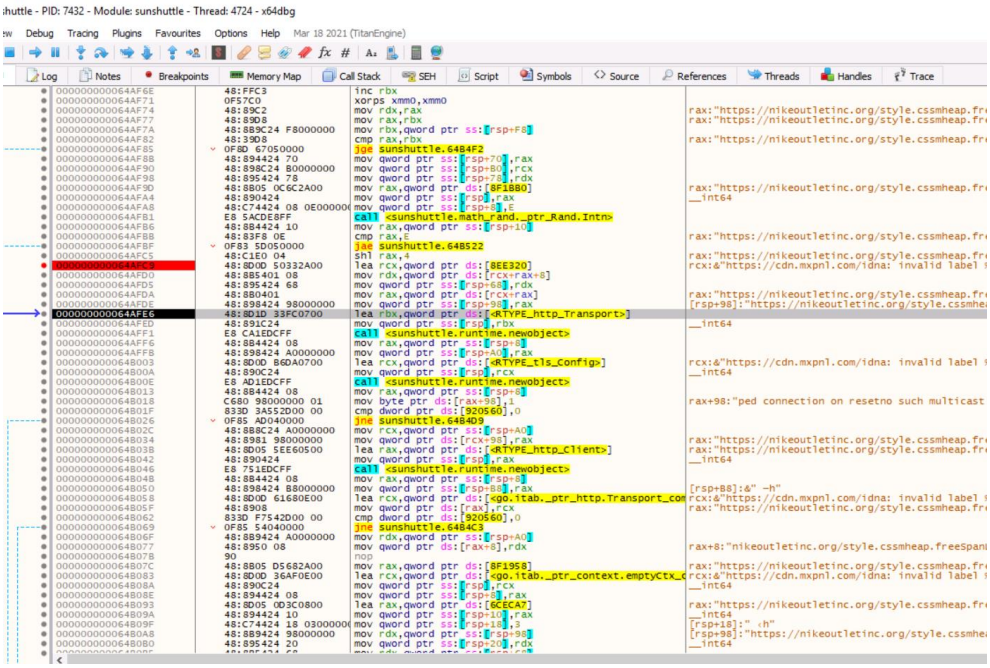
"nikeoutletinc.org:443"

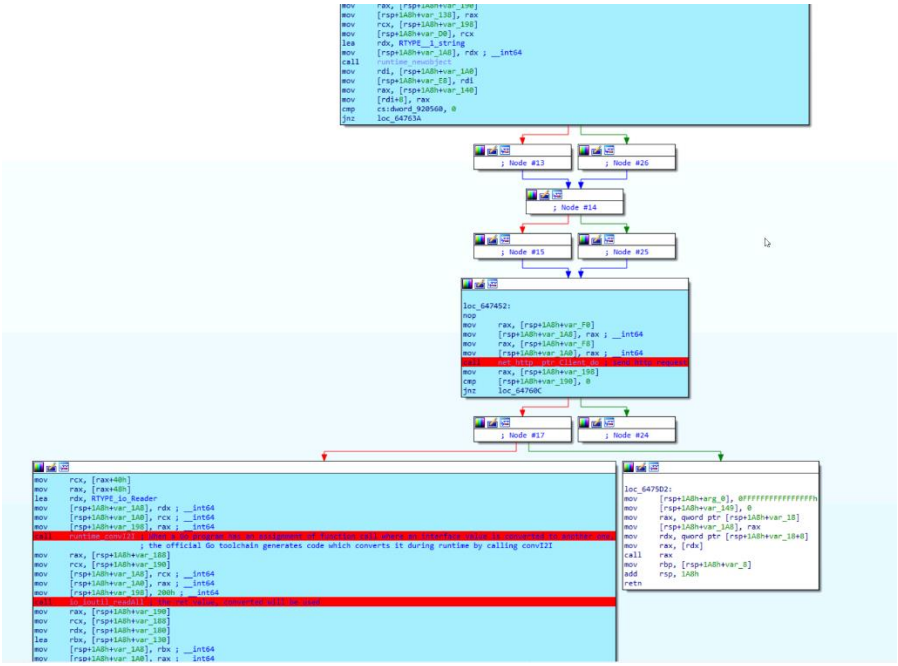


Main_request_session_key [creation of a C2 python script]




jmp loc_6471A9
loc_6471A9:
mov rax, [rsp+1A8h+var_F8]
mov [rsp+1A8h+var_1A8], rax
lea rcx, [rsp+1A8h+var_C0]
mov [rsp+1A8h+var_1A0], rcx
call net_http__ptr_Request_AddCookie
mov [rsp+1A8h+var_C0], 0
lea rdi, [rsp+1A8h+var_B8]
xorps xmm0, xmm0
lea rdi, [rdi-20h]
mov [rsp+1A8h+var_1B8], rbp
lea rbp, [rsp+1A8h+var_1B8]
call loc_46319E
mov rbp, [rbp+0]
lea rax, aIOTimeoutljvg2+111Fh ; "GC6BzibvrlkYoFrGateway TimeoutGetAdapte"...
mov [rsp+1A8h+var_C0], rax
mov [rsp+1A8h+var_B8], 0Fh
lea rax, aUwrgedylq ; "uWrGEdYLq"
mov [rsp+1A8h+var_B0], rax
mov [rsp+1A8h+var_A8], 9
mov rax, [rsp+1A8h+var_F8]
mov [rsp+1A8h+var_1A8], rax
lea rcx, [rsp+1A8h+var_C0]
mov [rsp+1A8h+var_1A0], rcx
call net_http__ptr_Request_AddCookie
mov [rsp+1A8h+var_C0], 0
lea rdi, [rsp+1A8h+var_B8]
xorps xmm0, xmm0
lea rdi, [rdi-20h]
mov [rsp+1A8h+var_1B8], rbp
lea rbp, [rsp+1A8h+var_1B8]
call loc_46319E
mov rbp, [rbp+0]
lea rax, aQmfbry7qu ; "QMFbry7qU"
mov [rsp+1A8h+var_C0], rax
mov [rsp+1A8h+var_B8], 9
mov rax, cs:qword_8E8848
mov rcx, cs:qword_8E8840
mov [rsp+1A8h+var_B0], rcx
mov [rsp+1A8h+var_A8], rax
mov rax, [rsp+1A8h+var_F8]
mov [rsp+1A8h+var_1A8], rax
lea rcx, [rsp+1A8h+var_C0]
mov [rsp+1A8h+var_1A0], rcx
call net_http__ptr_Request_AddCookie
mov [rsp+1A8h+var_C0], 0
lea rdi, [rsp+1A8h+var_B8]
xorps xmm0, xmm0
lea rdi, [rdi-20h]
mov [rsp+1A8h+var_1B8], rbp
lea rbp, [rsp+1A8h+var_1B8]
call loc_46319E
mov rbp, [rbp+0]
lea rax, aBiido6vrloxbad ; "BIIdo6VrLOxBad GatewayBad Request"
mov [rsp+1A8h+var_C0], rax
mov [rsp+1A8h+var_B8], 0Bh
mov rax, cs:qword_8E8838
mov rcx, cs:off_8E8830 ; "11012520020420630440040 4"
mov [rsp+1A8h+var_B0], rcx
mov [rsp+1A8h+var_A8], rax
mov rax, [rsp+1A8h+var_F8]
mov [rsp+1A8h+var_1A8], rax
lea rcx, [rsp+1A8h+var_C0]
mov [rsp+1A8h+var_1A0], rcx
call net_http__ptr_Request_AddCookie
mov rax, [rsp+1A8h+var_F8]
mov rcx, [rax+38h]
mov [rsp+1A8h+var_C8], rcx
mov rdx, cs:user_agent.data
mov [rsp+1A8h+var_110], rdx
request_session_key+444 (Synchronized with Hex View-1)

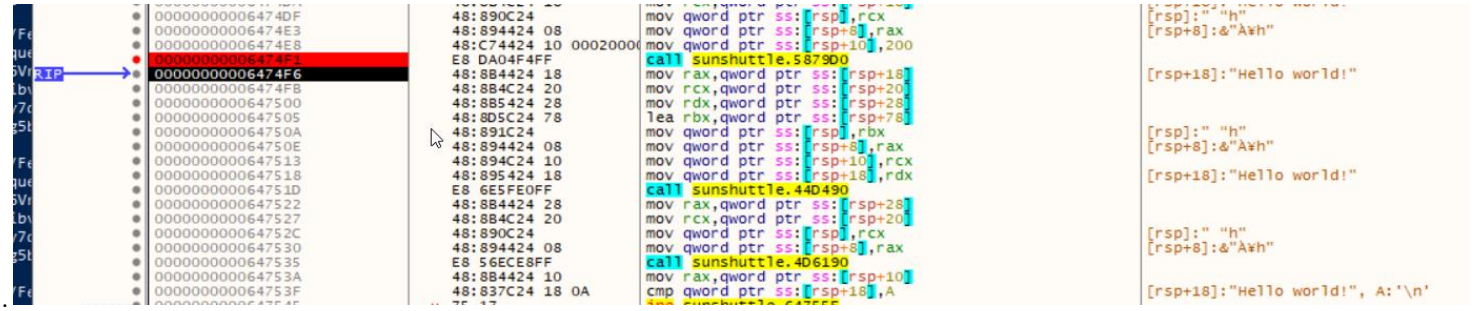

ihuttle - PID: 7432 - Module: sunshuttle - Thread: 4724 - x64dbg
Debug Tracing Plugins Favourites Options Help Mar 18 2021 (TitanEngine)
Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Handles Trace



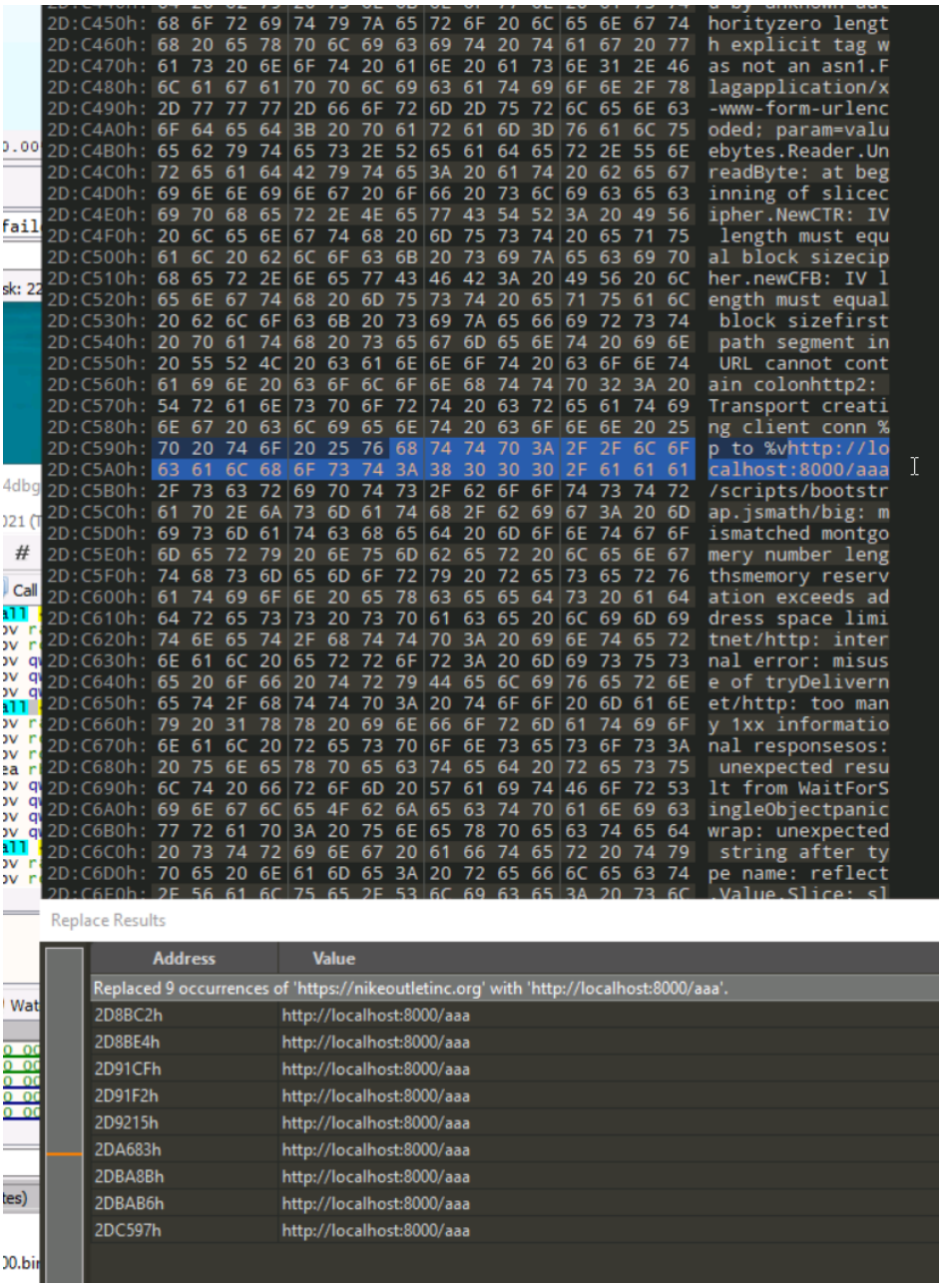
As the server have been seized, create a c2 server with a python script

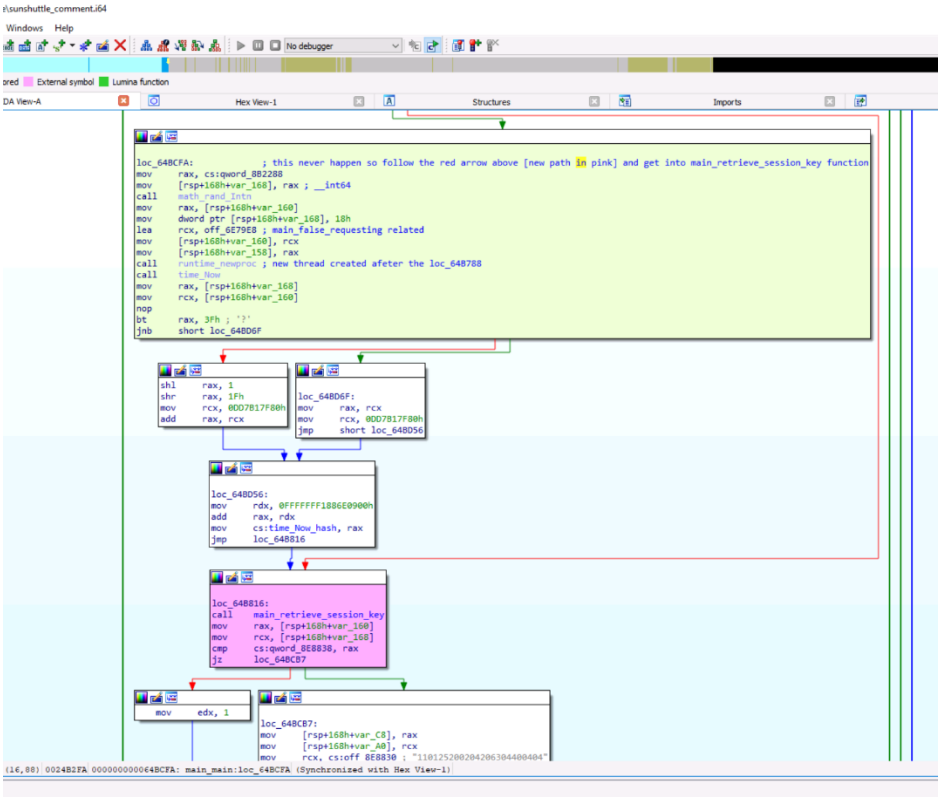
```python
import http.server
import sys
import http.cookies

HOSTNAME = "localhost"
PORT = 8000

class SunshuttleHandler(http.server.BaseHTTPRequestHandler):
    def do_GET(self):
        #Add your code here
        cookies = http.cookies.SimpleCookie(self.headers.get('Cookie'))
        print(cookies, "\r\n")
        sys.stdout.flush()
        """Respond to a GET request."""
        self.send_response(200)
        self.send_header("Content-type", "text/html")
        self.end_headers()

        self.wfile.write(b"Hello world!")

        print(f"Received a GET request on {self.path}!")
        sys.stdout.flush()


def run(server_class=http.server.HTTPServer, handler_class=SunshuttleHandler):
    server_address = (HOSTNAME, PORT)
    httpd = server_class(server_address, handler_class)
    try:
        httpd.serve_forever()
    except KeyboardInterrupt:
        print("Bye!")


if __name__ == '__main__':
    run()
```

```
127.0.0.1 - - [26/Feb/2022 08:55:50] "GET /aaa/polls/db/ HTTP/1.1" 200 -
Received a GET request on /aaa/polls/db/!
Set-Cookie: BIIdo6VrLOx=110
Set-Cookie: GC6BzibvrlkYoFr=uWrGEdYLq
Set-Cookie: QMFbry7qU=gWJpnphrWoSL4ciMCxB0e5wxXpyznr5k
Set-Cookie: vw3LTg5bR3fb=19e2378af4de73053a4cd40721dbf427

127.0.0.1 - - [26/Feb/2022 08:55:56] "GET /aaa/polls/db/ HTTP/1.1" 200 -
Received a GET request on /aaa/polls/db/!
```

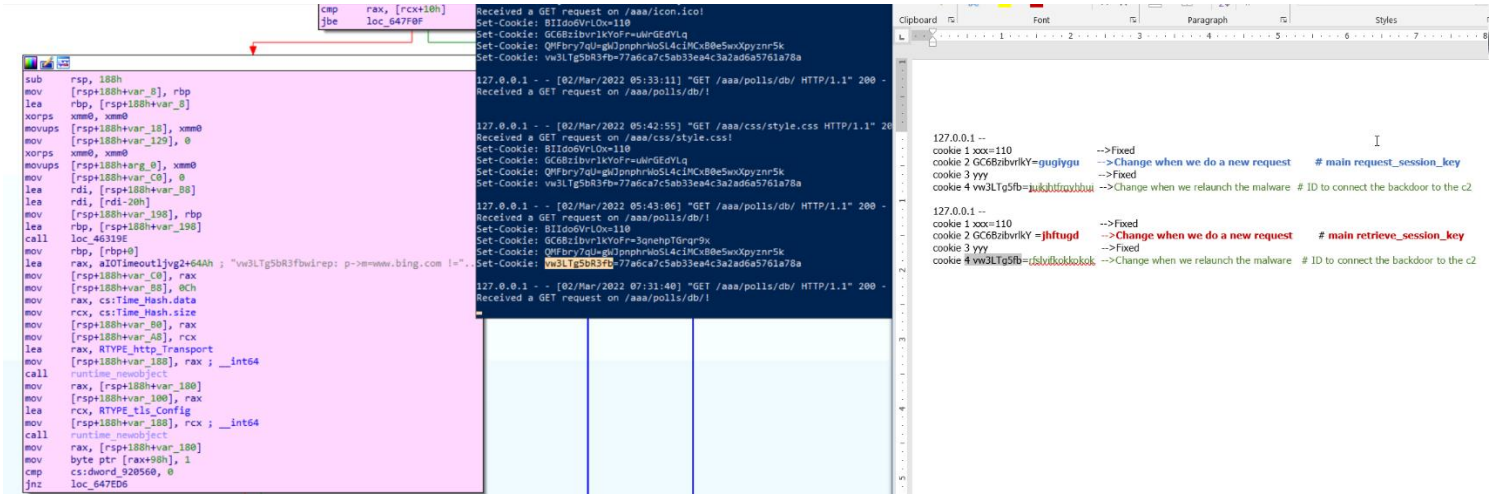


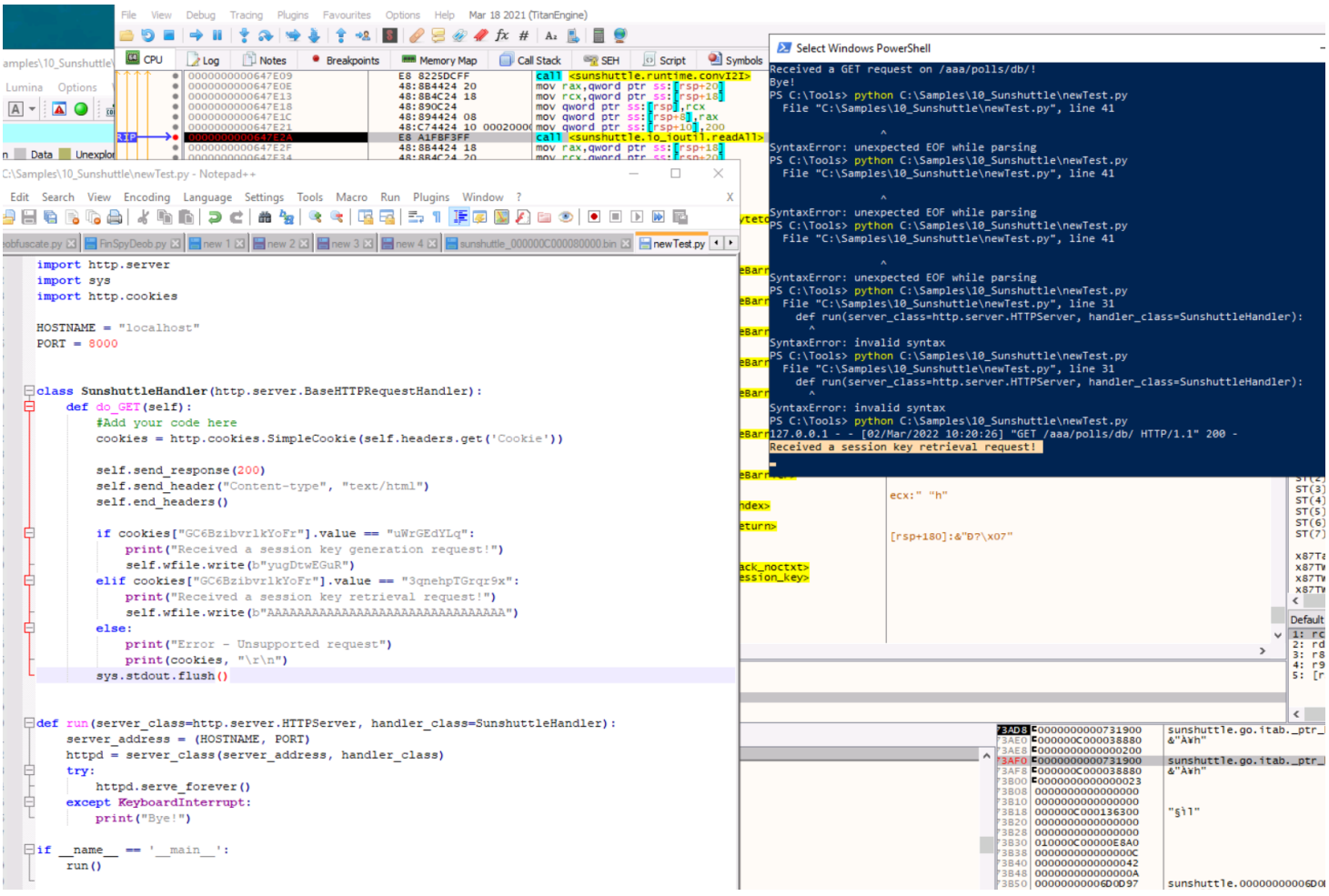
Change the c2 server in sunshuttle: open sunshuttle sample with 010 editor, look for the c2 server as a string, and replace the malicious domain name by another one that we control (downgrading from https to http in order to avoid SSL certificate settings, and respecting the size of the original string. Cf. structure of string in go), for instance replace https://nikeoutletinc.org by http://localhost:8888/aaa



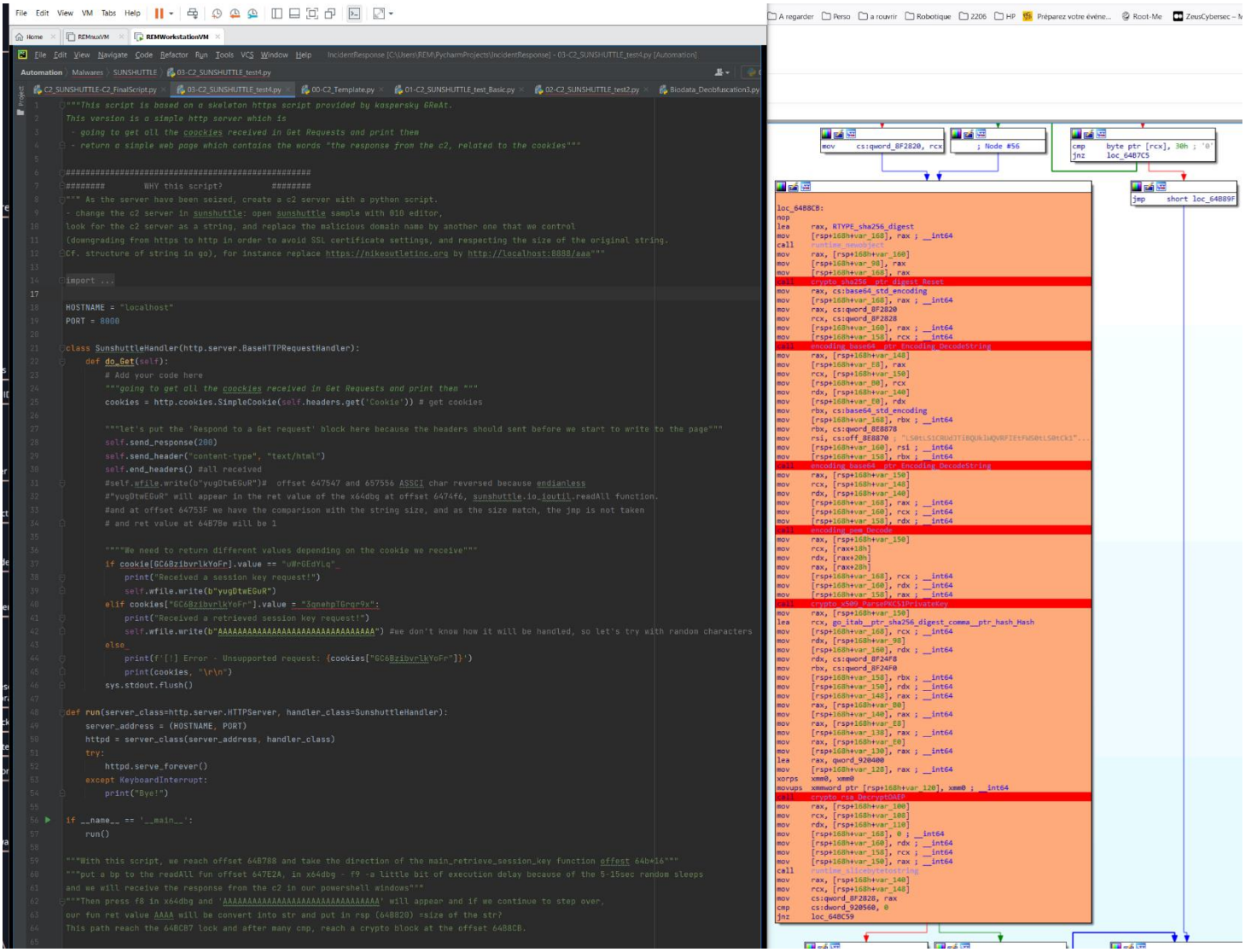

loc_64BCFA: ; this never happen so follow the red arrow above [new path in pink] and get into main_retrieve_session_key function
loc_64BD6F:
loc_64BD56:
loc_64B816:
loc_64BCB7:

Main_retrieve_session_key

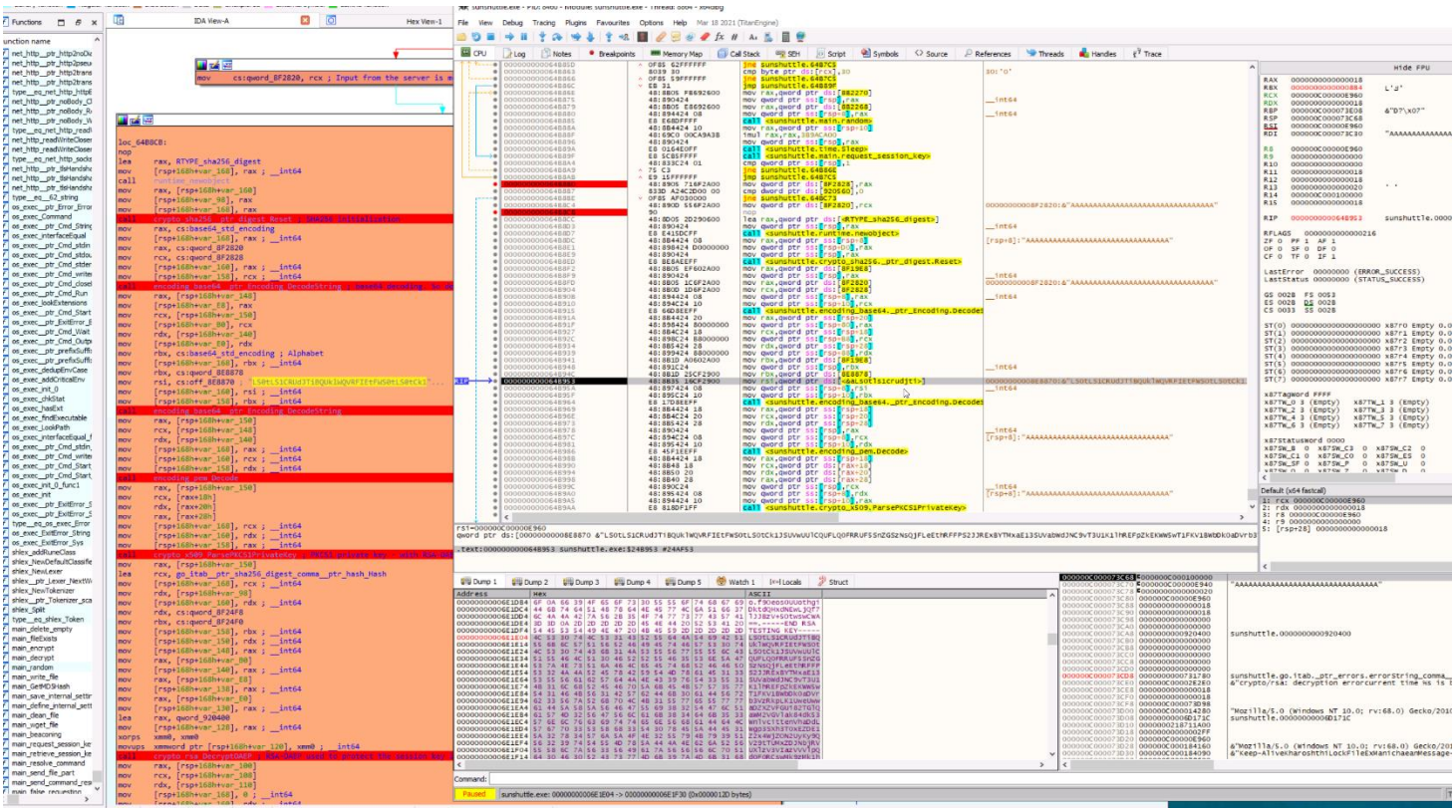


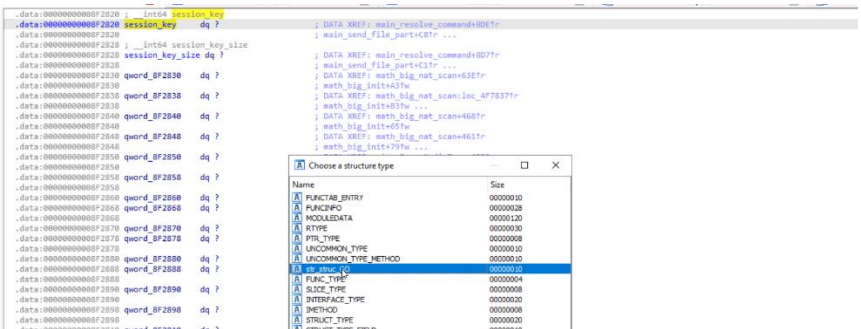


Crypto block



```
.text:000000000064A2A9 mov     rdx, [rsp+3C0h+var_398]
.text:000000000064A2AE lea     rbx, [rsp+3C0h+var_2F8]
.text:000000000064A2B6 mov     [rsp+3C0h+var_3C0], rbx ; __int64
.text:000000000064A2BA mov     [rsp+3C0h+var_3B8], rcx ; __int64
.text:000000000064A2BF mov     [rsp+3C0h+var_3B0], rax ; __int64
.text:000000000064A2C4 mov     [rsp+3C0h+var_3A8], rdx ; __int64
.text:000000000064A2C9 call    runtime_slicebytetostring
.text:000000000064A2CE mov     rax, [rsp+3C0h+var_398]
.text:000000000064A2D3 mov     [rsp+3C0h+var_350], rax
.text:000000000064A2D8 mov     rcx, [rsp+3C0h+var_3A0]
.text:000000000064A2DD mov     [rsp+3C0h+var_288], rcx
.text:000000000064A2E5 lea     rdx, [rsp+3C0h+var_318]
.text:000000000064A2ED mov     [rsp+3C0h+var_3C0], rdx ; __int64
.text:000000000064A2F1 mov     rdx, cs:session_key.size ;                    session key being used
.text:000000000064A2F8 mov     rbx, cs:session_key.data
.text:000000000064A2FF mov     [rsp+3C0h+var_3B8], rbx ; __int64
.text:000000000064A304 mov     [rsp+3C0h+var_3B0], rdx ; __int64
.text:000000000064A309 call    runtime_stringtoslicebyte
.text:000000000064A30E mov     rax, [rsp+3C0h+var_398]
.text:000000000064A313 mov     rcx, [rsp+3C0h+var_3A0]
.text:000000000064A318 mov     rdx, [rsp+3C0h+var_3A8]
.text:000000000064A31D mov     [rsp+3C0h+var_3C0], rdx ; __int64
.text:000000000064A321 mov     [rsp+3C0h+var_3B8], rcx ; __int64
.text:000000000064A326 mov     [rsp+3C0h+var_3B0], rax ; __int64
.text:000000000064A32B mov     rax, [rsp+3C0h+var_288]
.text:000000000064A333 mov     [rsp+3C0h+var_3A8], rax ; __int64
.text:000000000064A338 mov     rax, [rsp+3C0h+var_350]
.text:000000000064A33D mov     [rsp+3C0h+var_3A0], rax ; __int64
.text:000000000064A342 call    main_encrypt    ;                          data encryption
.text:000000000064A342                         ; fun saw before -->NewCipher.
.text:000000000064A342                         ; probably with the session key that was negociated
.text:000000000064A342                         ; And then CFB mode created,
.text:000000000064A342                         ; and then some base64_url_encoding, and then
.text:000000000064A342                         ; whatever whis has been encoded, is ret
.text:000000000064A347 mov     rax, [rsp+3C0h+var_390] ; the data in rax is doing stuffs,
.text:000000000064A34C mov     rcx, [rsp+3C0h+var_398]
.text:000000000064A351 lea     rdx, [rsp+3C0h+var_338]
.text:000000000064A359 mov     [rsp+3C0h+var_3C0], rdx ; __int64
.text:000000000064A35D mov     [rsp+3C0h+var_3B8], rcx ; __int64
.text:000000000064A362 mov     [rsp+3C0h+var_3B0], rax ; __int64
.text:000000000064A367 call    runtime_stringtoslicebyte
.text:000000000064A36C mov     rax, [rsp+3C0h+var_398]
.text:000000000064A371 mov     rcx, [rsp+3C0h+var_3A0]
.text:000000000064A376 mov     rdx, [rsp+3C0h+var_3A8]
.text:000000000064A37B mov     rbx, cs:base64_std_encoding
.text:000000000064A382 mov     [rsp+3C0h+var_3C0], rbx ; __int64
.text:000000000064A386 mov     [rsp+3C0h+var_3B8], rdx ; __int64
.text:000000000064A38B mov     [rsp+3C0h+var_3B0], rcx ; __int64
.text:000000000064A390 mov     [rsp+3C0h+var_3A8], rax ; __int64
.text:000000000064A395 call    encoding_base64__ptr_Encoding_EncodeToString ;   base64_std_Encoding
.text:000000000064A39A mov     rax, [rsp+3C0h+var_398]
.text:000000000064A39F mov     [rsp+3C0h+var_360], rax
.text:000000000064A3A4 mov     rcx, [rsp+3C0h+var_3A0]
.text:000000000064A3A9 mov     [rsp+3C0h+var_2B8], rcx
.text:000000000064A3B1 mov     [rsp+3C0h+var_200], 0
.text:000000000064A3BD lea     rdi, [rsp+3C0h+var_1F8]
.text:000000000064A3C5 xorps   xmm0, xmm0
.text:000000000064A3C8 lea     rdi, [rdi-20h]
.text:000000000064A3CC mov     [rsp+3C0h+var_3D0], rbp
.text:000000000064A3D1 lea     rbp, [rsp+3C0h+var_3D0]
.text:000000000064A3D6 call    loc_46319E
.text:000000000064A3DB mov     rbp, [rbp+0]
.text:000000000064A3DF lea     rdx, aIOTimeoutljvg2+6Ah ; "vw3LIg5bR3fbwirep: p->m=www.bing.com !="...
.text:000000000064A3E6 mov     [rsp+3C0h+var_200], rdx ; we see the usual cookies we have seen before
.text:000000000064A3EE mov     [rsp+3C0h+var_1F8], 0Ch
.text:000000000064A3FA mov     rdx, cs:Time_Hash.data ; the cookie with the hash of the current time...
.text:000000000064A401 mov     rbx, cs:Time_Hash.size
.text:000000000064A408 mov     [rsp+3C0h+var_1F0], rdx
.text:000000000064A410 mov     [rsp+3C0h+var_1E8], rbx
.text:000000000064A418 lea     rdx, RTYPE_http_Transport
.text:000000000064A41F mov     [rsp+3C0h+var_3C0], rdx ; __int64
.text:000000000064A423 call    runtime_newobject
.text:000000000064A428 mov     rax, [rsp+3C0h+var_3B8]
.text:000000000064A42D mov     [rsp+3C0h+var_2A0], rax
.text:000000000064A435 lea     rcx, RTYPE_tls_Config
.text:000000000064A43C mov     [rsp+3C0h+var_3C0], rcx ; __int64
.text:000000000064A440 call    runtime_newobject
.text:000000000064A445 mov     rax, [rsp+3C0h+var_3B8]
.text:000000000064A44A mov     byte ptr [rax+98h], 1
```

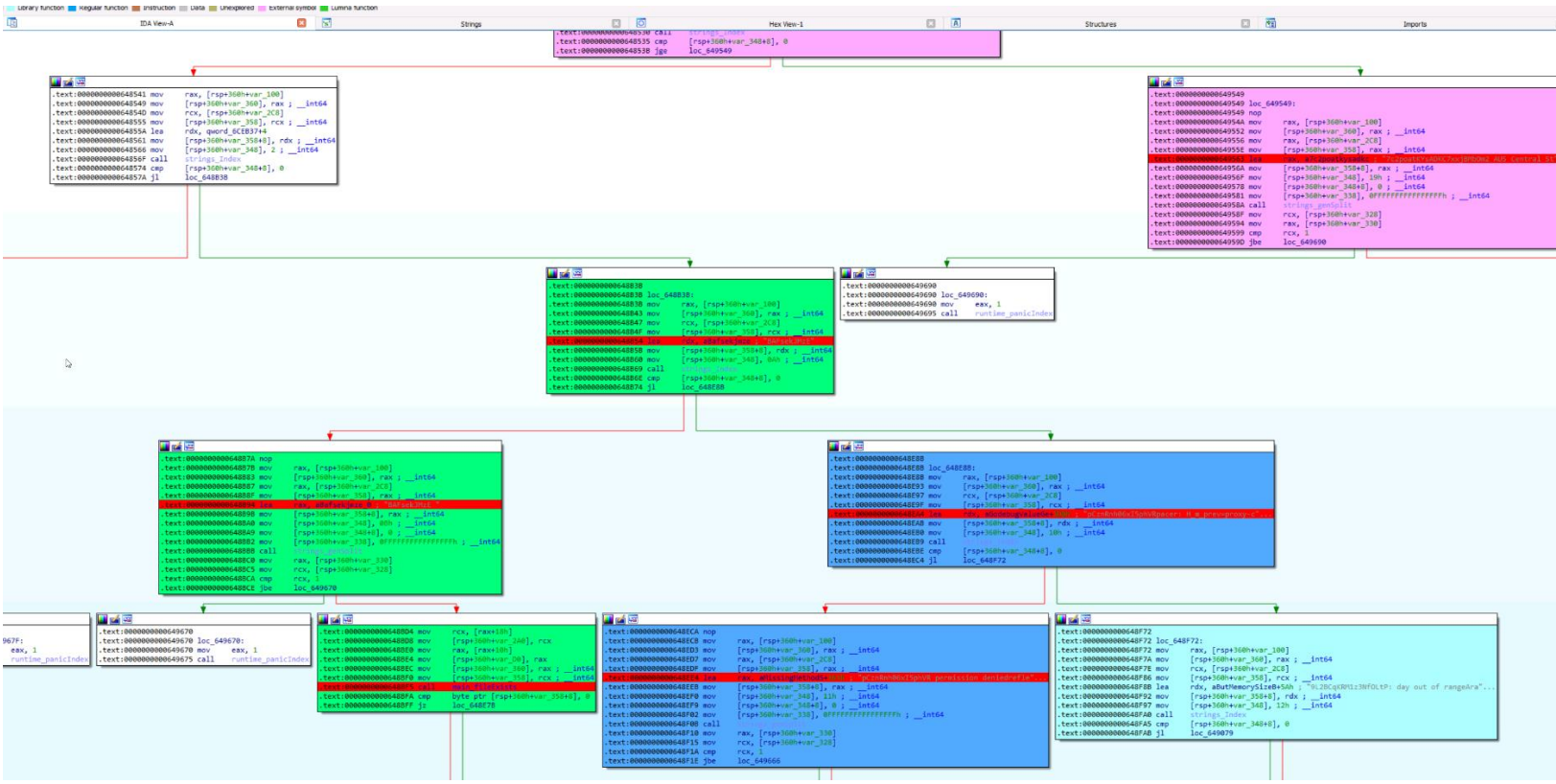


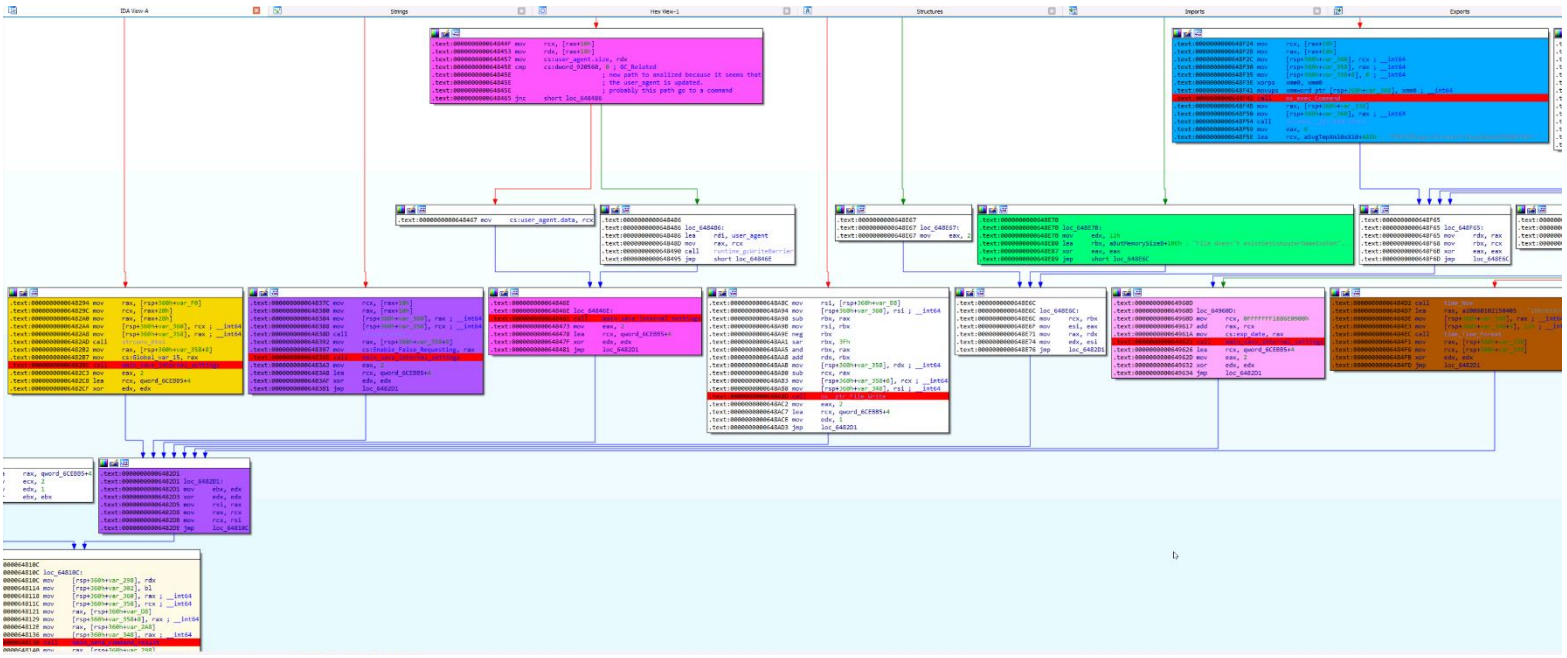


```
>>> s = "WGlwdHFST2dPSUFUR3Z2dTdzQkJhNlBqbVJkYmJRei1xZC1TWl9KLXJNcw=="
>>> import base64
>>> from Crypto.cipher import PKCS1_OAEP, AES
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'Crypto.cipher'
>>> from Crypto.Cipher import PKCS1_OAEP, AES
>>> base64.b64decode(s)
b'XiptqROgOIATGvvu7sBBa6PjmRdbbQz-qd-SZ_J-rMs'
>>> s = base64.b64decode(s)
>>> base64.urlsafe_b64decode(s)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "C:\Python37\lib\base64.py", line 133, in urlsafe_b64decode
    return b64decode(s)
  File "C:\Python37\lib\base64.py", line 87, in b64decode
    return binascii.a2b_base64(s)
binascii.Error: Incorrect padding
>>> base64.urlsafe_b64decode(s + b"===")
b'^*m\xa9\x13\xa08\x80\x13\x1a\xfb\xee\xee\xc0Ak\xa3\xe3\x99\x17[m\x0c\xfe\xa9\xdf\x92g\xf2~\xac\xcb'
>>> s = base64.urlsafe_b64decode(s + b"===")
>>> SESSION_KEY = b"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
>>> cipher = AES.new(SESSION_KEY, AES.MODE_CFB, segment_size=128, iv=s[0:16])
>>> cipher.decrypt(s[16:])
b'\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10'
>>>
```

```
Received POST data from the client
Set-Cookie: GC6BzibvrlkYoFr=S4KUbT6H8NY
Set-Cookie: QMFbry7qU=gWJpnphrWoSL4ciMCxB0e5wxXpyznr5k
Set-Cookie: vw3LTg5bR3fb=77a6ca7c5ab33ea4c3a2ad6a5761a78a
{b'QnwddoTIoKDj2u': [b'WTF0R0syMm1vS1JMNmI3MXQ4ZWxOanAzcnA5aF9Qd0w2V3o3aGtsaWhnRQ=='], b'zQy9aJHW8U8rpQN': [b'WHATEVER']
}
127.0.0.1 - - [07/Mar/2022 13:59:31] "GET /aaa/polls/db/ HTTP/1.1" 200 -
REQUEST without the expected cookie!
Set-Cookie: QMFbry7qU=gWJpnphrWoSL4ciMCxB0e5wxXpyznr5k
Set-Cookie: vw3LTg5bR3fb=77a6ca7c5ab33ea4c3a2ad6a5761a78a
```

Commands





Python Scripts

The main scripts are available in the Sunshuttle_scripts.zip attached