

Incident Response Threat Analysis

Prepared for

E Corp



SUNSHUTTLE – Go Malware

[28-Feb-2023]

SUMMARY	3
FINDINGS	3
<i>Attack Vector</i>	3
This table shows what we know.....	4
<i>Behavior summary</i>	5
RECOMMENDED ACTIONS	5
REFERENCES.....	6
CONTACT US	6
Annexes	7
Analyzing a Go Malware	7
Why Go?	7
What analyzing Go malware is different?	7
The approach	7
Methodology / Tips	7
Screenshot of the analysis.....	8
Net-interfaces.....	8
main_define_internal_settings	9
main_GetMD5hash	12
Main_encrypt.....	14
Main_false_requesting and C2 server	17
Main_request_session_key [creation of a C2 python script]	22
Main_retrieve_session_key	28
Crypto block	29
Commands	33
Python Scripts	34

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		
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	478B04C20BBF6717D10EE978B99339B7C4664FEB8BCFDAF86C3F0FBFC83A5C5	
Infection Vector	N/A	
Packer	UPX	
Language	Go	
Malware/Family	SUNSHUTTLE	
Setup of the program	<ul style="list-style-type: none"> 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 1 st 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	<pre>PS C:\Tools> python C:\Samples\10_Sunshuttle\c2_Test.py 127.0.0.1 - - [02/Mar/2022 05:33:11] "GET /aaa/icon.ico HTTP/1.1" 200 - Received a GET request on /aaa/icon.ico! Set-Cookie: BIIdo6VrLOx=110 Set-Cookie: GC6Bzibvr1kYoFr=uWrGEEdYLq Set-Cookie: QMFbry7qu=gWJpnphrwSL4ciMCxB0e5wxXpyznr5k Set-Cookie: vw3LTg5bR3fb=77a6ca7c5ab33ea4c3a2ad6a5761a78a</pre>	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 UVG2jnYvnE jktstagnsC5pCmfuTM3 fbbClnzwrmzFQZmDlgc AT18tInio8H7mM513 7c2poatKYsADKC7xxjBMb0m 2 [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/analvsis-reports/ar21-105a	15 apr 2021
Virustotal		https://www.virustotal.com/gui/file/478b04c20bbf6717d10ee978b99339b7c4664feb8bcfdaf86c3f0fbfc83a5c5/details [the hash submitted, not the file]	25 Oct 2021
PE102	Corkami	Other useful references – Team Sharing Information Purpose	TYPE
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)

```

Data Unexplored External symbol Lumina function
IDA View-A Hex View-1 Structures
00000000 00000000 str_struct_00 struct ; (sizeof=0x10, mappedto_30995)
00000000 00000000 ; XREF: .data$user_agent/r
00000000 00000000 ; .data$stru_BE000000 ...
00000000 00000000 ; main$main_save_Internal_settings+48/r
00000000 00000000 ; main$main_save_Internal_settings+96/r ...
00000000 00000000 ; main$main_save_Internal_settings+90/r ...
00000000 00000000 ends
00000010
00000010
00000000 00000000 str_struct_00

```

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

```

loc_64BE92:    ; _int64
    cmp    rax, 11h      ; test instr to check if the size is 17 characters
    jz     loc_64B692

loc_64B692:    ; _int64
    mov    [rsp+168h+var_168], rcx
    lea    rcx, aMissingMethods+1EDh ; "c8:27:cc:c2:37:5adecryption failedenter"...
    mov    [rsp+168h+var_168], rcx ; _int64
    mov    [rsp+168h+var_158], rax ; _int64
    call   runtime_memequal ; string equal function to cmp to the value of the lea? -->
    ; c8:27:cc:c2:37:5a = hyper-v network adapter default addr -->
    ; Anti_VM technique
    cmp    byte ptr [rsp+168h+var_150], 0
    jz     loc_64B67D

loc_64B67D:
    mov    [rsp+168h+var_168], 0
    call   os.Exit
    jmp     loc_64B67D

loc_64B699:
    call   time.Now
    mov    rax, [rsp+168h+var_160]
    mov    rcx, [rsp+168h+var_168]
    nop
    bt     rcx, 3Fh ; '?'
    jnb    loc_64BE8A

loc_64B65A8:
    mov    rbx, [rsp+168h+var_90]
    add    rbx, 40h ; '@'
    mov    rdx, rax
    mov    rax, rbx

loc_64BE8A:
    mov    rcx, rax
    jmp     loc_64B6C7

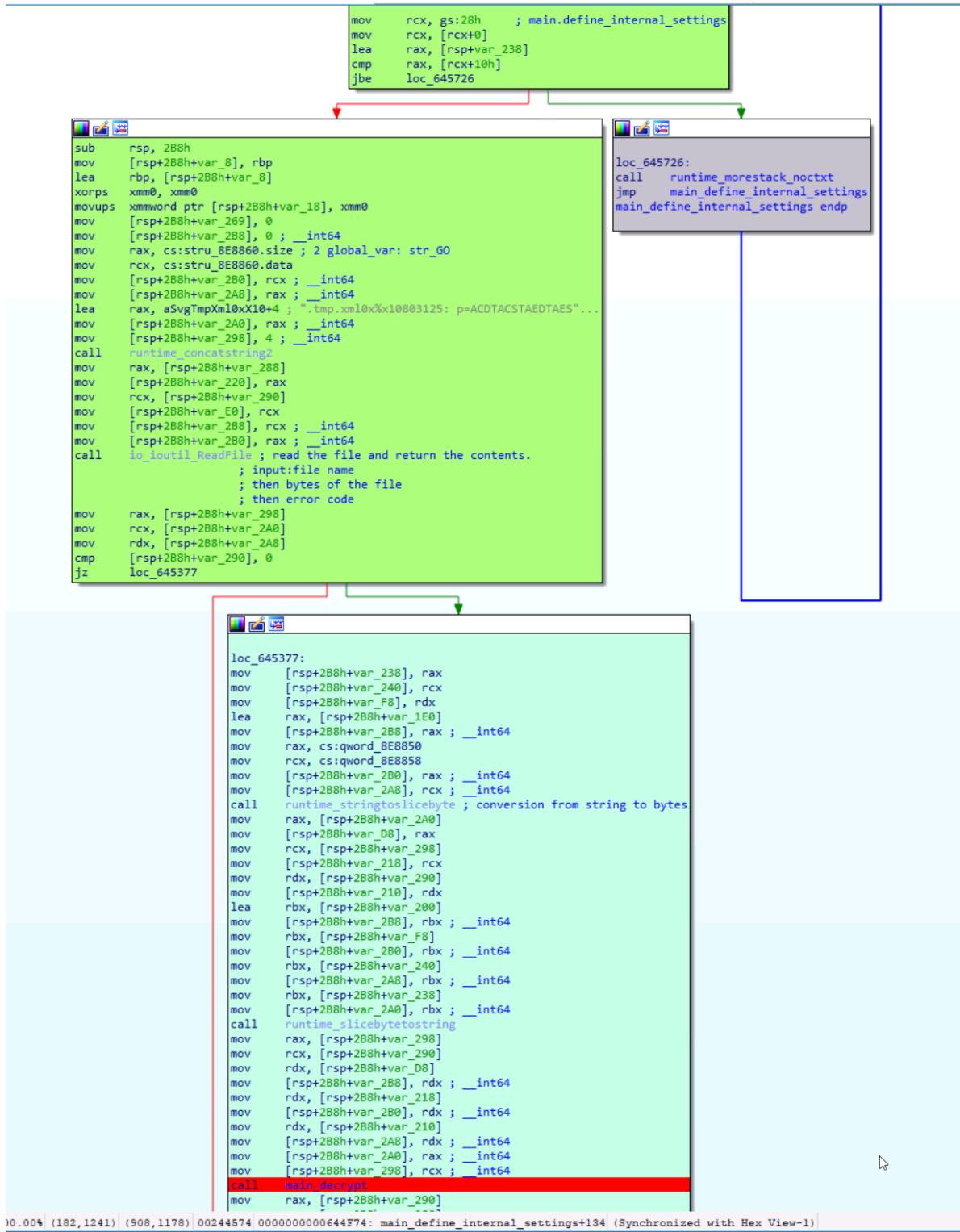
loc_64B6C7:
    shr    rcx, 1Fh
    mov    rax, 59453308800 ; 0DD7B17F80H = ?
    add    rcx, rax
    mov    rax, 9453308800 ; 9453308800 = january 3854 = ?
    add    rcx, rax

loc_64B6C8:
    mov    rax, cs:qword_8F1B80
    mov    [rsp+168h+var_168], rax
    mov    rax, 0FFFFFFF1886E0900h
    add    rcx, rax
    mov    [rsp+168h+var_168], rcx
    call   math.rand_ptr Random.Seed ; seed initialize the random generator of the program
    call   main_define_internal_settings
    call   time.Now
    mov    rax, [rsp+168h+var_160]
    mov    rcx, [rsp+168h+var_168]
    nop
    bt     rcx, 3Fh ; '?'
    jnh    loc_64B678
    var_40      = qword ptr -40h
    var_38      = qword ptr -38h
    var_30      = qword ptr -30h
    var_28      = qword ptr -28h
    var_20      = qword ptr -20h
    var_18      = qword ptr -18h
    var_8       = qword ptr -8h
    mov    rcx, gs:28h
    mov    rcx, [rcx+0]

func Interfaces()
    ([]Interface, error)

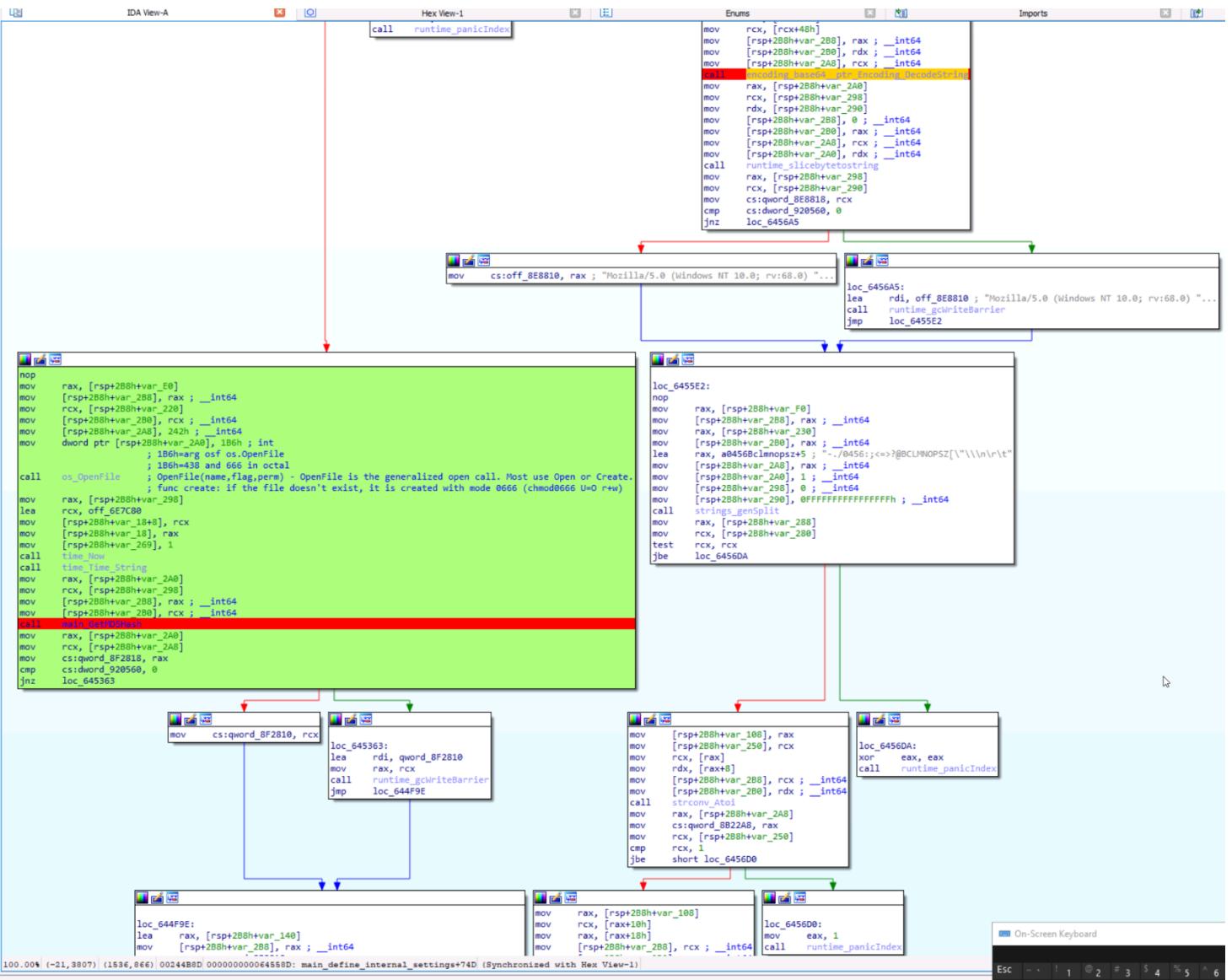
```

Interfaces returns a list of the system's network interfaces.

main_define_internal_settings

Incident Response Threat Analysis for E Corp

E CORP



The screenshot shows a debugger interface with assembly code. The top section is highlighted in green and contains the following assembly:

```

mov    [rsp+288h+var_2A0], rax ; _int64
mov    [rsp+288h+var_298], 4 ; _int64
call   runtime_concatstring2
mov    rax, [rsp+288h+var_288]
mov    [rsp+288h+var_220], rax
mov    rcx, [rsp+288h+var_290]
mov    [rsp+288h+var_E0], rcx
mov    [rsp+288h+var_2B8], rcx ; _int64
mov    [rsp+288h+var_2B0], rax ; _int64
call   io_ioutil_Readfile ; read the file and return the contents.
; input: file name
; then bytes of the file
; then error code
mov    rax, [rsp+288h+var_298]
mov    rcx, [rsp+288h+var_2A0]
mov    rdx, [rsp+288h+var_2A8]
cmp    [rsp+288h+var_290], 0
jz    loc_645377

```

The bottom section is highlighted in light blue and contains the following assembly:

```

loc_645377:
mov    [rsp+288h+var_238], rax
mov    [rsp+288h+var_240], rcx
mov    [rsp+288h+var_F8], rdx
lea    rax, [rsp+288h+var_1E0]
mov    [rsp+288h+var_2B8], rax ; _int64
mov    rax, cs:qword_8E8850
mov    rcx, cs:qword_8E8858
mov    [rsp+288h+var_2B0], rax ; _int64
mov    [rsp+288h+var_2A8], rcx ; _int64
call   runtime_stringtoslicebyte ; conversion from string to bytes
mov    rax, [rsp+288h+var_2A0]
mov    [rsp+288h+var_D8], rax
mov    rcx, [rsp+288h+var_298]
mov    [rsp+288h+var_218], rcx
mov    rdx, [rsp+288h+var_290]
mov    [rsp+288h+var_210], rdx
lea    rbx, [rsp+288h+var_200]
mov    [rsp+288h+var_2B8], rbx ; _int64
mov    rbx, [rsp+288h+var_F8]
mov    [rsp+288h+var_2B0], rbx ; _int64
mov    rbx, [rsp+288h+var_240]
mov    [rsp+288h+var_2A8], rbx ; _int64
mov    rbx, [rsp+288h+var_238]
mov    [rsp+288h+var_2A0], rbx ; _int64
call   runtime_slicebytetostring
mov    rax, [rsp+288h+var_298]
mov    rcx, [rsp+288h+var_290]
mov    rdx, [rsp+288h+var_D8]
mov    [rsp+288h+var_2B8], rdx ; _int64
mov    rdx, [rsp+288h+var_218]
mov    [rsp+288h+var_2B0], rdx ; _int64
mov    rdx, [rsp+288h+var_210]
mov    [rsp+288h+var_2A8], rdx ; _int64
mov    [rsp+288h+var_2A0], rax ; _int64
mov    [rsp+288h+var_298], rcx ; _int64
call   main_decrypt ; read the bytes that we got from the file, decrypt them with AES and base64 with URL encoding
mov    rax, [rsp+288h+var_290]
mov    rcx, [rsp+288h+var_288]
nop
mov    [rsp+288h+var_2B8], rax ; _int64
mov    [rsp+288h+var_2B0], rcx ; _int64
lea    rax, Pipe_sign ; Pipe Sign
mov    [rsp+288h+var_2A8], rax ; _int64
mov    [rsp+288h+var_2A0], 1 ; _int64
mov    [rsp+288h+var_298], 0 ; _int64
mov    [rsp+288h+var_290], 0xFFFFFFFFFFFFFFFh ; _int64
call   strings_genSplit ; split with the | sign
mov    rax, [rsp+288h+var_288]
mov    rcx, [rsp+288h+var_2B0]
test   rcx, rcx
jbe    loc_645709

```

*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

IDA View-A

Hex View-1

```

mov    rcx, gs:28h ; main_GetMD5hash
mov    rcx, [rcx+0]
cmp    rsp, [rcx+10h]
jbe    loc_6449F7

```

Enums

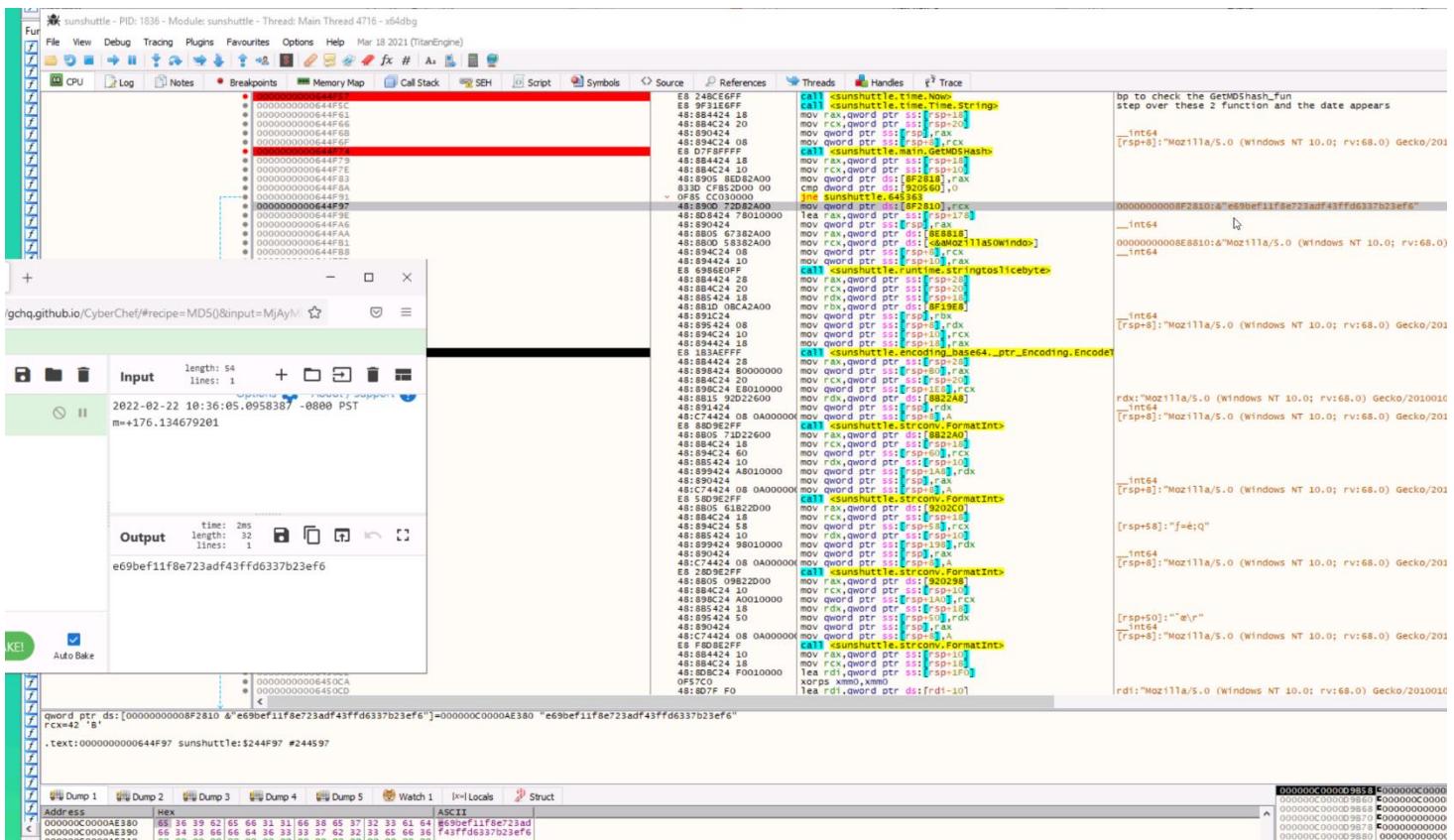
```

sub    rsp, 70h      ; take smthg as an arg
       ; convert this string into bytes
       ; Calculate MD5 hash of those bytes
       ; encode them to hex
       ; and return the value as a string
mov    [rsp+70h+var_8], rbp
lea    rbp, [rsp+70h+var_8]
lea    rax, RTYPE_md5_digest
mov    [rsp+70h+var_70], rax ; __int64
call   runtime_newobject
mov    rax, [rsp+70h+var_68]
mov    [rsp+70h+var_10], rax
nop
mov    rcx, 0EPCDA88967452301h
mov    [rax], rcx
mov    rcx, 10325476988ADCFEh
mov    [rax+8], rcx
xorps xmm0, xmm0
movups xmmword ptr [rax+50h], xmm0
mov    [rsp+70h+var_70], 0 ; __int64
mov    rcx, [rsp+70h+arg_0]
mov    [rsp+70h+var_68], rcx ; __int64
mov    rcx, [rsp+70h+arg_8]
mov    [rsp+70h+var_60], rcx ; __int64
call   runtime_stringtoslicebyte ; convert string to bytes
lea    rax, go_itab_ptr_md5_digest_comma_ptr_hash_Hash ; ?
test  al
mov    rax, [rsp+70h+var_50]
mov    rcx, [rsp+70h+var_48]
mov    rdx, [rsp+70h+var_60+8]
mov    rbx, [rsp+70h+var_10]
mov    [rsp+70h+var_70], rbx ; __int64
mov    [rsp+70h+var_68], rdx ; __int64
mov    [rsp+70h+var_60], rax ; __int64
mov    [rsp+70h+var_60+8], rcx ; __int64
call   crypto_md5_ptr_digest_Write ; write the bytes previously read into md5 hash object
mov    rax, [rsp+70h+var_10]
mov    [rsp+70h+var_70], rax ; __int64
mov    [rsp+70h+var_68], 0 ; __int64
xorps xmm0, xmm0
movups xmmword ptr [rsp+70h+var_60], xmm0 ; __int64
call   crypto_md5_ptr_digest_Sum ; get the sum
mov    rax, [rsp+70h+var_40]
mov    [rsp+70h+var_30], rax
mov    rax, [rsp+70h+var_40]
mov    [rsp+70h+var_30], rax
mov    rdx, [rsp+70h+var_50]
mov    [rsp+70h+var_20], rdx
lea    rbx, RTYPE_uint8
mov    [rsp+70h+var_70], rbx ; __int64
shl    rcx, 1
mov    [rsp+70h+var_20], rcx
mov    [rsp+70h+var_60], rcx ; __int64
mov    [rsp+70h+var_60], rcx ; __int64
call   runtime_makeslice ; way to take the subdivision of an array
mov    rax, [rsp+70h+var_60+8]
mov    [rsp+70h+var_10], rax
mov    [rsp+70h+var_70], rax ; __int64
mov    rcx, [rsp+70h+var_20]
mov    [rsp+70h+var_60], rax ; __int64
mov    rdx, [rsp+70h+var_60], rdx ; __int64
mov    [rsp+70h+var_60], rax ; __int64
mov    rdx, [rsp+70h+var_20]
mov    [rsp+70h+var_60+8], rdx ; __int64
mov    rdx, [rsp+70h+var_30]
mov    [rsp+70h+var_50], rdx ; __int64
mov    rdx, [rsp+70h+var_30]
mov    [rsp+70h+var_40], rdx ; __int64
call   encoding_hex_Encode ; hex encode
mov    [rsp+70h+var_70], 0 ; __int64
mov    rax, [rsp+70h+var_10]
mov    [rsp+70h+var_60], rax ; __int64
mov    rax, [rsp+70h+var_20]
mov    [rsp+70h+var_60], rax ; __int64
mov    [rsp+70h+var_60+8], rax ; __int64
call   runtime_slicebytetostring ; bytes to string

```

Incident Response Threat Analysis for E Corp

E CORP



Main_encrypt

```

mov    rcx, gs:28h ; main.encrypt fun
mov    rcx, [rcx+0]
lea    rax, [rsp+var_30]
cmp    rax, [rcx+10h]
jbe    loc_644260

sub    rsp, 080h
mov    [rsp+080h+var_8], rbp
lea    rbp, [rsp+080h+var_8]
mov    rax, [rsp+080h+arg_0]
mov    [rsp+080h+var_B0], rax ; _int64
mov    rax, [rsp+080h+arg_8]
mov    [rsp+080h+var_A8], rax ; _int64
mov    rax, [rsp+080h+arg_10]
mov    [rsp+080h+var_A0], rax ; _int64
test   rdx, rdx
jnz    loc_64422A

call   crypto aes NewCipher ; input is encrypted with AES
mov    rax, [rsp+080h+var_98]
mov    rcx, [rsp+080h+var_80]
mov    rdx, qword ptr [rsp+080h+var_90+8]
mov    rbx, qword ptr [rsp+080h+var_90]
test   rdx, rdx
jnz    loc_64422A

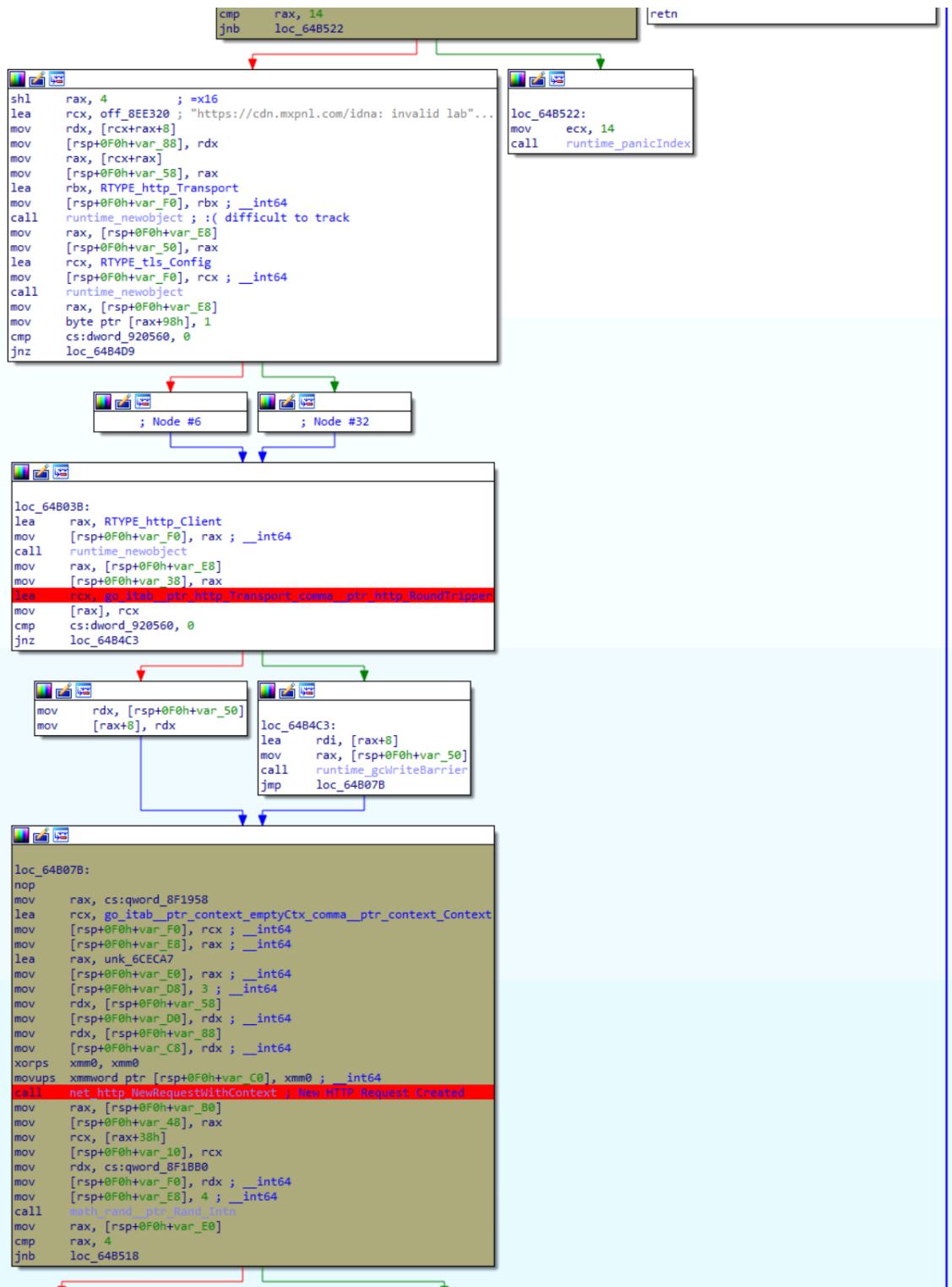
loc_64422A:
xorps  xmm0, xmm0
movups [rsp+080h+arg_28], xmm0
mov    qword ptr [rsp+080h+arg_38], rdx
mov    qword ptr [rsp+080h+arg_38+8], rcx
mov    rbp, [rsp+080h+var_8]
add    rbp, 080h
retn

; Node #3
; Node #11
; Node #4

loc_6441D6:
rax, RTYPE_uint8
mov    rax, [rsp+080h+var_B0], rax ; _int64
mov    rsi, [rsp+080h+var_28]
mov    [rsp+080h+var_A8], rsi ; _int64
mov    [rsp+080h+var_A0], rax ; _int64
mov    [rsp+080h+var_98], rbx ; _int64

```


1 Page 2 Page 3 Page 4 Page 5

Main_false_requesting_and_C2_server

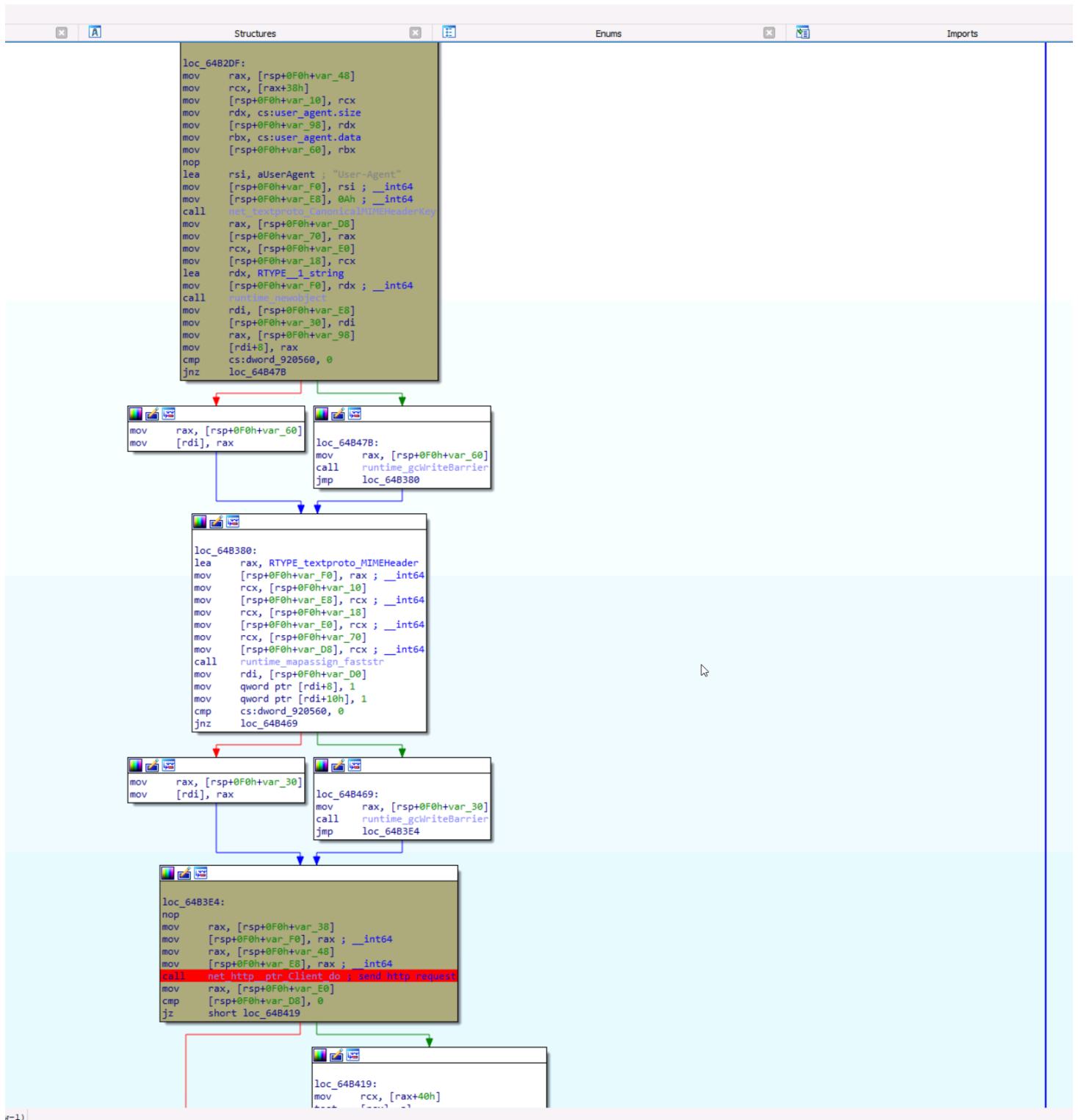
IDA View-A Hex View-1 A Structures Enums Imports

```

loc_64B204:
    mov    rax, [rsp+0h+var_48]
    mov    rcx, [rax+30h]
    mov    [rsp+0fh+var_10], rcx
    nop
    lea    rdx, aConnectionContent ; "ConnectionContent-Id"
    mov    [rsp+0fh+var_F0], rdx ; __int64
    mov    [rsp+0fh+var_E0], 1
    mov    [rsp+0fh+var_D0], rdx
    lea    rdx, RTYPE__1_string
    mov    [rsp+0fh+var_F0], rdx ; __int64
    call   runtime._emitObject
    mov    rax, [rsp+0fh+var_E0]
    mov    [rsp+0fh+var_D0], rax
    mov    quord ptr [rax+8], 0Ah
    lea    rcx, aKeepAlive ; "keep-Alive"
    mov    [rax], rcx
    lea    rax, rcpType_textrproto_NDHEHeader
    mov    [rsp+0fh+var_70], rax ; __int64
    mov    rdx, [rsp+0fh+var_10]
    mov    [rsp+0fh+var_E0], rdx ; __int64
    mov    rdx, [rsp+0fh+var_10]
    mov    [rsp+0fh+var_E0], rdx ; __int64
    mov    rdx, [rsp+0fh+var_E0]
    mov    [rsp+0fh+var_E0], rdx ; __int64
    mov    rdx, [rsp+0fh+var_D0]
    mov    [rsp+0fh+var_D0], rdx ; __int64
    call   runtime._emitObject
    mov    rdi, [rsp+0fh+var_00]
    mov    quord ptr [rdi+8], 1
    mov    quord ptr [rdi+10], 1
    cap   csidword g20560, 0
    jnz   loc_64B480
    loc_64B480:
    mov    rax, [rsp+0fh+var_28]
    mov    [rdi], rax
    call   runtime._emitObject
    jmp   loc_64B2DF
    loc_64B2DF:
    mov    rax, [rsp+0fh+var_48]
    mov    rcx, [rax+30h]
    mov    rdx, csUserAgent_size
    mov    [rsp+0fh+var_98], rdx
    mov    rbx, csUserAgent_data
    mov    [rsp+0fh+var_60], rbx
    nop
    lea    rsi, UserAgent ; "User-Agent"
    mov    [rsp+0fh+var_F0], rsi ; __int64
    mov    [rsp+0fh+var_E0], 0Ah ; __int64
    call   runtime._emitObject
    mov    rax, [rsp+0fh+var_D0]
    mov    [rsp+0fh+var_70], rax
    mov    rcx, [rsp+0fh+var_E0]
    mov    [rsp+0fh+var_10], rcx
    lea    rdx, RTYPE__1_string
    mov    [rsp+0fh+var_F0], rdx ; __int64
    call   runtime._emitObject
    mov    rdi, [rsp+0fh+var_E0]
    mov    [rsp+0fh+var_30], rdi
    mov    rax, [rsp+0fh+var_98]
    [rdi+8], rax
    cmp   rax, csidword g20560, 0
    inc   rax
    loc_64B470:
    mov    rax, [rsp+0fh+var_A0]

```

955) 0024A4B5 000000000064B0B5: main_false_requesting+195 (Synchronized with Hex View-1)



Incident Response Threat Analysis for E Corp

E CORP

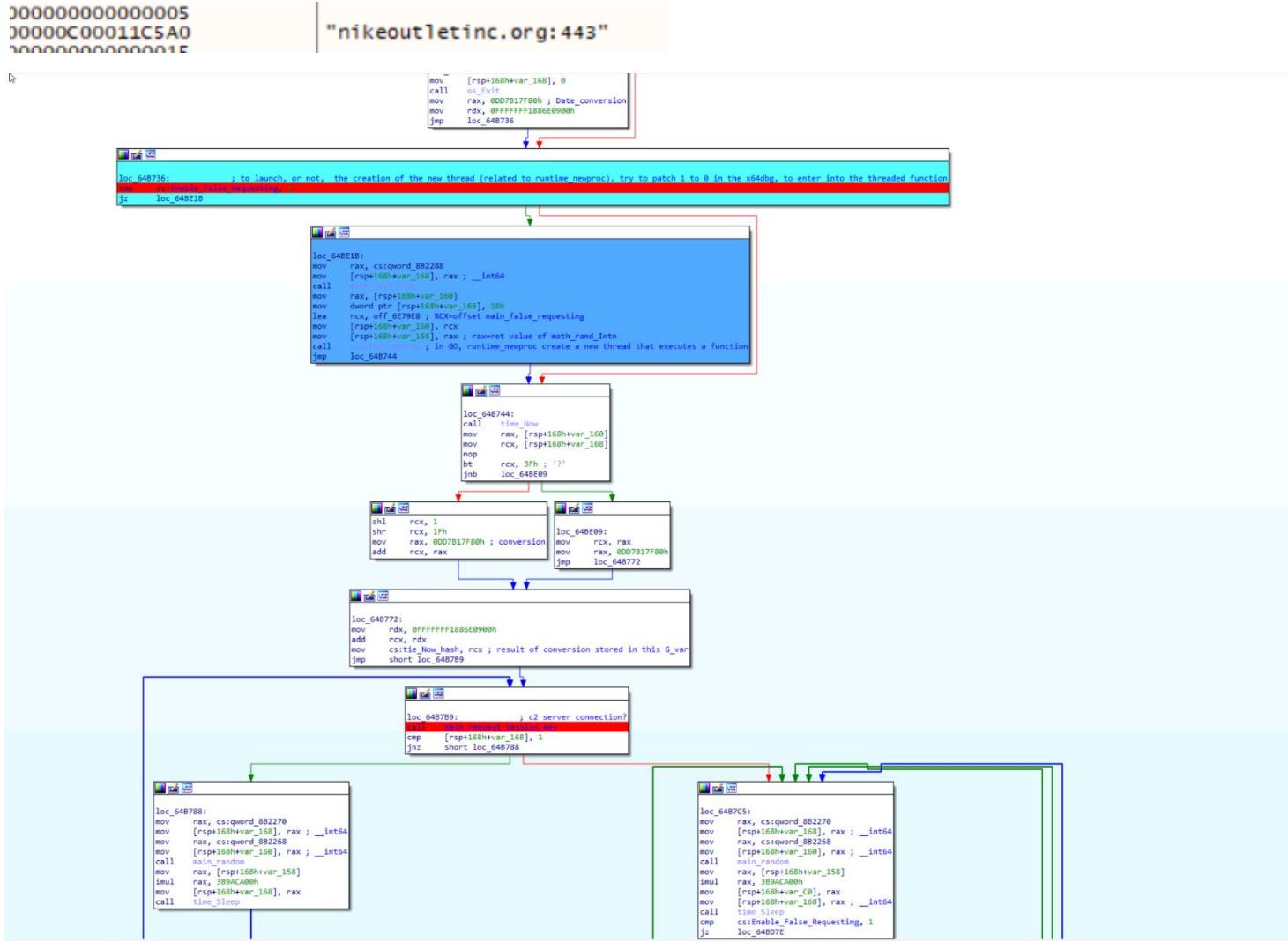
Assembly pane:

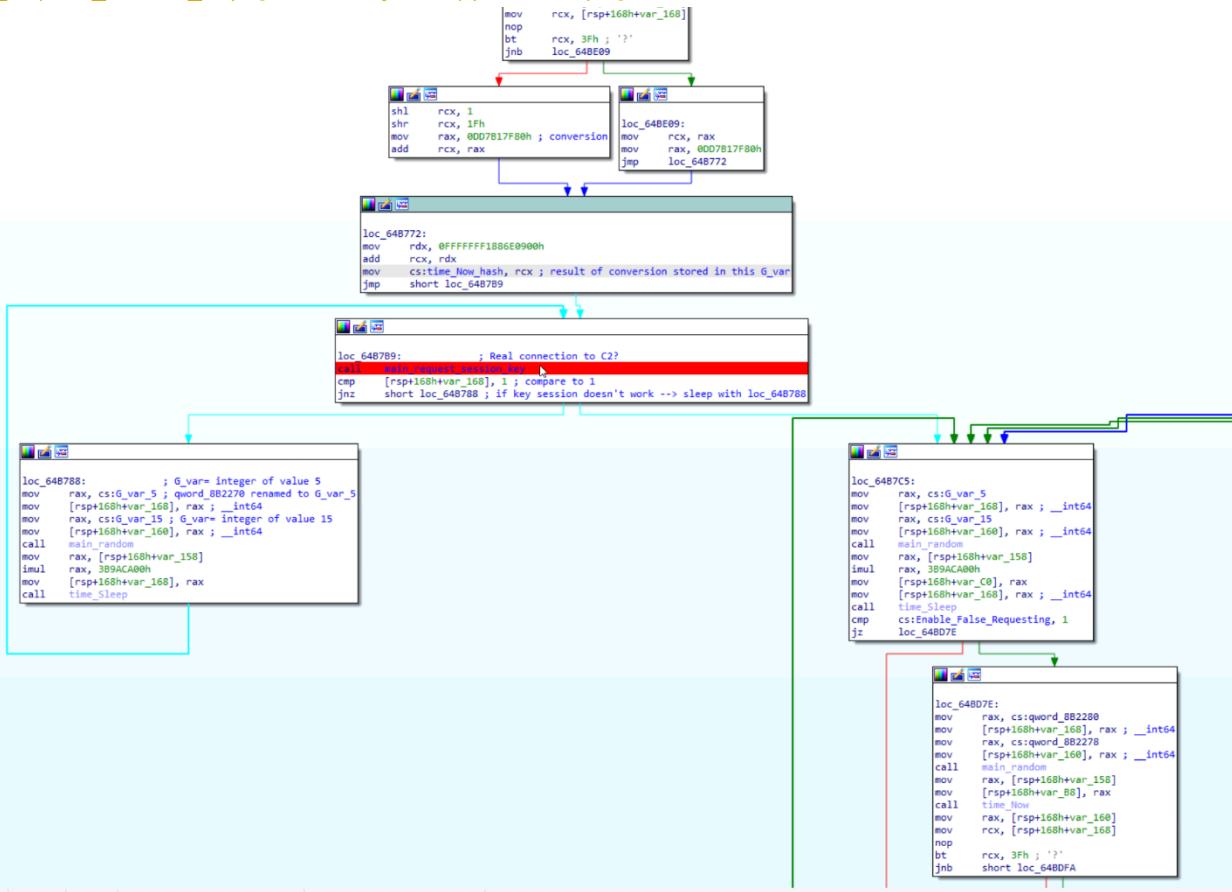
```
48:C1E9 1F shr    rcx,1F  
48:B8 807FB1D70D0000 mov    rax,DD7B17F80  
48:D1E1 add    rcx,rcx  
48:A0 00906E88F1FFFFF mov    rdx,FFFFFFF1886E0900  
48:01D1 add    rcx,rdx  
48:390D 684B2D00 cmp    qword ptr ds:[820298],rcx  
48:0F8F 1C070000 jae    sunshuttle.64BE52  
48:833D 824B2D00 01 cmp    qword ptr ds:[9202C0].1  
48:0F84 D7060000 jae    sunshuttle.64BE1B  
48:E8 3754E6F mov    rax,qword ptr ss:[ESP+8]  
48:88A4 2408 call   <sunshuttle.time.now>  
48:880C 24 mov    rcx,qword ptr ss:[ESP+8]  
48:90          bt    rcx,3F  
48:98 0048EAE1 3F jae    sunshuttle.64BE09  
48:F0 83 AB060000 shl    rcx,1  
48:D1E1 add    rcx,rcx  
48:C1E9 1F shr    rcx,1F  
48:B8 807FB1D70D0000 mov    rax,DD7B17F80  
48:01C1 add    rcx,rcx  
48:A0 00906E88F1FFFFF mov    rdx,FFFFFFF1886E0900  
48:01D1 add    rcx,rdx
```

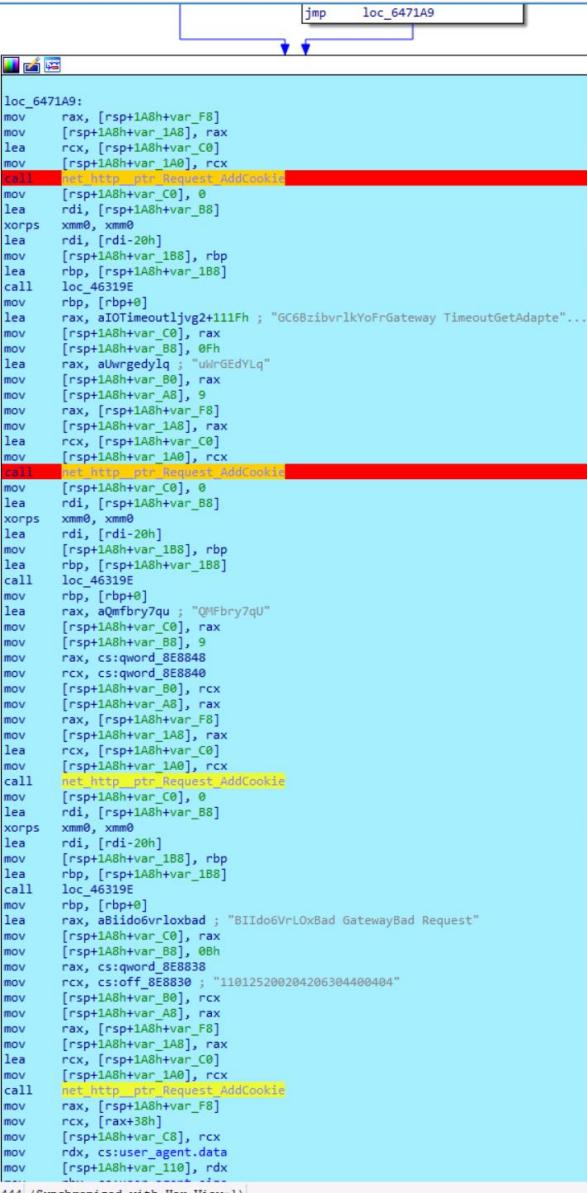
The screenshot shows the x64dbg debugger interface with the assembly window active. The assembly pane displays assembly code for the sunshuttle module. A specific instruction at address 48:8D0D 5032A00 is highlighted in yellow, labeled as the current target. The assembly code includes various CPU instructions such as lea, mov, and cmp, along with memory operations involving registers like rax, rcx, and rdx, and memory addresses like [rsp+8]. The stack pane shows the current state of the stack, and the memory dump pane shows the contents of memory at specific addresses. The interface includes standard debugger navigation and search tools.

2022-nbr
SUNSHUTTLE Malware Analysis Report

20

C2:

Main_request_session_key [creation of a C2 python script]



```

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, aUkrgedyLq ; "UlkGEdYLq"
mov    [rsp+1A8h+var_B8], 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_B8], 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, aBIldo6VrLoxbad ; "BIldo6VrLoxbad GatewayBad Request"
mov    [rsp+1A8h+var_C0], rax
mov    [rsp+1A8h+var_B8], 0Bh
mov    rax, cs:qword_8E8838
mov    rcx, cs:off_8E8830 ; "110125200204206304400404"
mov    [rsp+1A8h+var_B8], 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)

```

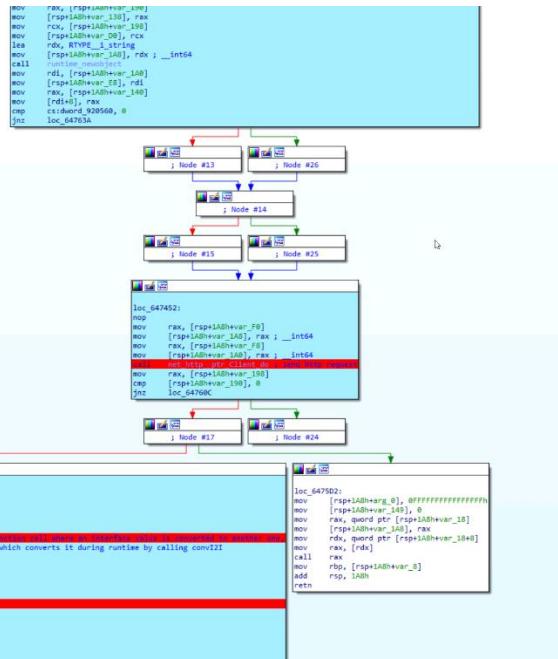
Incident Response Threat Analysis for E Corp

E CORP

:shuttle - PID: 7432 - Module: sunshuttle - Thread: 4724 - x64dbg

Debug Tracing Plugins Favourites Options Help Mar 18 2021 (TitanEngine)

The screenshot displays the Immunity Debugger interface with the assembly view active. The assembly pane shows assembly code for a function starting at address 48:0F5C3. The code includes instructions like inc rbx, xorps xmm0, xmm0, mov rax, mov rax, mov rax, qword ptr ss:[rsp+8], and a call instruction to `sunshuttle.lib484F2`. Several memory locations are highlighted in blue or red, such as 48:0F5C0, 48:0F5D0, and 48:0F5E0. A red arrow points to the instruction at 48:0F5E0, which is `lea rbx, qword ptr [rax + 40h]`. The registers pane shows CPU, MMU, and Registers sections. The stack pane shows the current stack state. The memory map pane shows the program's memory layout. The breakpoints pane shows the current breakpoints. The log pane shows the command history. The notes pane shows any user notes. The handles pane shows open handles. The trace pane shows the current trace information.



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

```

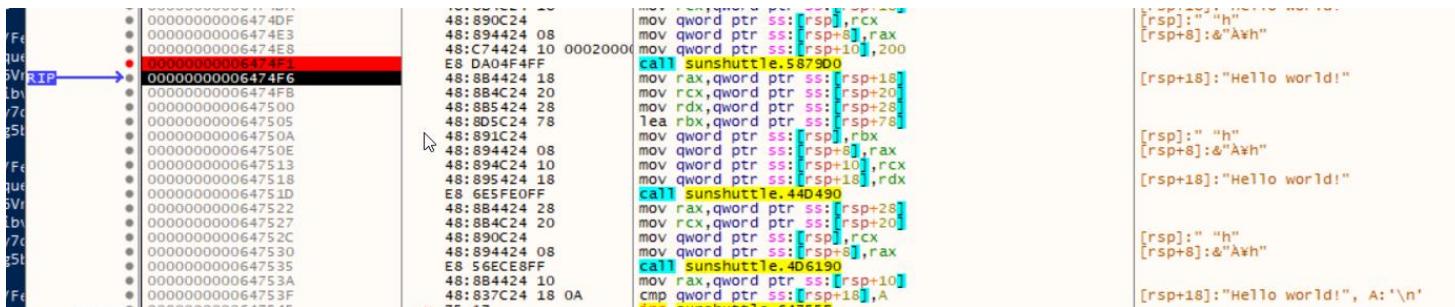
1 import http.server
2 import sys
3 import http.cookies
4
5 HOSTNAME = "localhost"
6 PORT = 8000
7
8
9 class SunshuttleHandler(http.server.BaseHTTPRequestHandler):
10     def do_GET(self):
11         #Add your code here
12         cookies = http.cookies.SimpleCookie(self.headers.get('Cookie'))
13         print(cookies, "\r\n")
14         sys.stdout.flush()
15         """Respond to a GET request."""
16         self.send_response(200)
17         self.send_header("Content-type", "text/html")
18         self.end_headers()
19
20         self.wfile.write(b"Hello world!")
21
22         print(f"Received a GET request on {self.path}!")
23         sys.stdout.flush()
24
25
26     def run(server_class=http.server.HTTPServer, handler_class=SunshuttleHandler):
27         server_address = (HOSTNAME, PORT)
28         httpd = server_class(server_address, handler_class)
29         try:
30             httpd.serve_forever()
31         except KeyboardInterrupt:
32             print("Bye!")
33
34
35 if __name__ == '__main__':
36     run()
37
38

```

```

Set-Cookie: VNSB7gJ8R5t0-19c1978048c7305564646721807427
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: B1D06VrL0x=110
Set-Cookie: GC6BzibvrlkYOf=rUwGEyDlq
Set-Cookie: QMFdry7Qu=gWJpnphrWoSL4ciMcxB0e5wxXpyznr5k
Set-Cookie: vw3LTgBr3fb=19e2378af4de73053a4cd0721dbf427
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>

```

0.00 2D:C450h: 68 6F 72 69 74 79 7A 65 72 6F 20 6C 65 6E 67 74 horityzero lenght
2D:C460h: 68 20 65 78 70 6C 69 63 69 74 20 74 61 67 20 77 h explicit tag w
2D:C470h: 61 73 20 6E 6F 74 20 61 6E 20 61 73 6E 31 2E 46 as not an asn1.F
2D:C480h: 6C 61 67 61 70 70 6C 69 63 61 74 69 6F 6E 2F 78 lagapplication/x
2D:C490h: 2D 77 77 77 2D 66 6F 72 6D 2D 75 72 6C 65 6E 63 -www-form-urlencoded; param=valu
2D:C4A0h: 6F 64 65 64 3B 20 70 61 72 61 6D 3D 76 61 6C 75 oded; param=valu
2D:C4B0h: 65 62 79 74 65 73 2E 52 65 61 64 65 72 2E 55 6E ebytes.Reader.Un
2D:C4C0h: 72 65 61 64 42 79 74 65 3A 20 61 74 20 62 65 67 readByte: at beg
2D:C4D0h: 69 6E 6E 69 6E 67 20 6F 66 20 73 6C 69 63 65 63 inning of slicec
2D:C4E0h: 69 70 68 65 72 2E 4E 65 77 43 54 52 3A 20 49 56 ipher.NewCTR: IV
2D:C4F0h: 20 6C 65 6E 67 74 68 20 6D 75 73 74 20 65 71 75 length must equ
2D:C500h: 61 6C 20 62 6C 6F 63 6B 20 73 69 7A 65 63 69 70 al block sizecip
2D:C510h: 68 65 72 2E 6E 65 77 43 46 42 3A 20 49 56 20 6C her.newCFB: IV l
2D:C520h: 65 6E 67 74 68 20 6D 75 73 74 20 65 71 75 61 6C enghth must equal
2D:C530h: 20 62 6C 6F 63 6B 20 73 69 7A 65 66 69 72 73 74 block sizefirst
2D:C540h: 20 70 61 74 68 20 73 65 67 6D 65 6E 74 20 69 6E path segment in
2D:C550h: 20 55 52 4C 20 63 61 6E 6E 6F 74 20 63 6F 6E 74 URL cannot cont
2D:C560h: 61 69 6E 20 63 6F 6C 6F 6E 68 74 74 70 32 3A 20 ain colonhttp2:
2D:C570h: 54 72 61 6E 73 70 6F 72 74 20 63 72 65 61 74 69 Transport creati
2D:C580h: 6E 67 20 63 6C 69 65 6E 74 20 63 6F 6E 6E 20 25 ng client conn %
2D:C590h: 70 20 74 6F 20 25 76 68 74 74 70 3A 2F 2F 6C 6F p to %vhttp://lo
2D:C5A0h: 63 61 6C 68 6F 73 74 3A 38 30 30 30 2F 61 61 61 calhost:8000/aaa /scripts/bootstr
2D:C5B0h: 2F 73 63 72 69 70 74 73 2F 62 6F 6F 74 73 74 72 /scripts/bootstr
2D:C5C0h: 61 70 2E 6A 73 6D 61 74 68 2F 62 69 67 3A 20 6D ap.jsmath/big: m
2D:C5D0h: 69 73 6D 61 74 63 68 65 64 20 6D 6F 6E 74 67 6F ismatched montgo
# 2D:C5E0h: 6D 65 72 79 20 6E 75 6D 62 65 72 20 6C 65 6E 67 mery number leng
2D:C5F0h: 74 68 73 6D 65 6D 6F 72 79 20 72 65 73 65 72 76 thsmemory reserv
2D:C600h: 61 74 69 6F 6E 20 65 78 63 65 65 64 73 20 61 64 ation exceeds ad
2D:C610h: 64 72 65 73 73 20 73 70 61 63 65 20 6C 69 6D 69 dress space limi
2D:C620h: 74 6E 65 74 2F 68 74 74 70 3A 20 69 6E 74 65 72 tnet/http: inter
2D:C630h: 6E 61 6C 20 65 72 72 6F 72 3A 20 6D 69 73 75 73 nal error: misus
2D:C640h: 65 20 6F 66 20 74 72 79 44 65 6C 69 76 65 72 6E e of tryDelivern
2D:C650h: 65 74 2F 68 74 74 70 3A 20 74 6F 6F 20 6D 61 6E et/http: too man
2D:C660h: 79 20 31 78 78 20 69 6E 66 6F 72 6D 61 74 69 6F y 1xx informatio
2D:C670h: 6E 61 6C 20 72 65 73 70 6F 6E 73 65 73 6F 73 3A nal responsesos:
2D:C680h: 20 75 6E 65 78 70 65 63 74 65 64 20 72 65 73 75 unexpected resu
2D:C690h: 6C 74 20 66 72 6F 6D 20 57 61 69 74 46 6F 72 53 lt from WaitForS
2D:C6A0h: 69 6E 67 6C 65 4F 62 6A 65 63 74 70 61 6E 69 63 ingleObjectpanic
2D:C6B0h: 77 72 61 70 3A 20 75 6E 65 78 70 65 63 74 65 64 wrap: unexpected
2D:C6C0h: 20 73 74 72 69 6E 67 20 61 66 74 65 72 20 74 79 string after ty
2D:C6D0h: 70 65 20 6E 61 6D 65 3A 20 72 65 66 6C 65 63 74 pe name: reflect
2D:C6E0h: 2F 56 61 6C 75 65 2F 53 6C 69 63 65 3A 20 73 6C .Value.Slice: sl

```

Replace Results

Address	Value
Replaced 9 occurrences of 'https://nikeoutletinc.org' with 'http://localhost:8000/aaa'.	
2D8BC2h	http://localhost:8000/aaa
2D8BE4h	http://localhost:8000/aaa
2D91CFh	http://localhost:8000/aaa
2D91F2h	http://localhost:8000/aaa
2D9215h	http://localhost:8000/aaa
2DA683h	http://localhost:8000/aaa
2DBA8Bh	http://localhost:8000/aaa
2DBAB6h	http://localhost:8000/aaa
2DC597h	http://localhost:8000/aaa

\sunshuttle_comment.i64

Windows Help

pred External symbol Lumina function

DA View-A Hex View-1 Structures Imports

```

loc_64BCFA: ; this never happen so follow the red arrow above [new path in pink] and get into main_retrieve_session_key function
    mov    rax, cs:qword_8822B8
    mov    [rsp+188h+var_188], rax ; _int64
    call   _main
    mov    rax, [rsp+168h+var_160]
    mov    dword ptr [rsp+168h+var_160], 18h
    lea    rcx, off_6E79E8 ; main_false_requesting related
    mov    [rsp+188h+var_158], rcx
    call   runtime_request; new thread created after the loc_648788
    call   time_Now
    mov    rax, [rsp+168h+var_168]
    mov    rcx, [rsp+168h+var_160]
    nop
    bt    rax, 3Fh ; '?'
    jnb   short loc_64BD0F

    ; Red arrow points here
    shl   rax, 1
    shr   rax, 1Fh
    mov    rcx, 80D7B17F80h
    add    rax, rcx
    mov    rax, loc_64BD0F

    ; Pink box highlights this section
    loc_64BD0F:
    mov    rax, rcx
    mov    rcx, 80D7B17F80h
    jmp   short loc_64BD056

    ; Pink box highlights this section
    loc_64BD056:
    mov    rdx, 0FFFFFFF1886E0900h
    add    rax, rdx
    mov    cstime_Now_hash, rax
    jmp   loc_648816

    ; Pink box highlights this section
    loc_648816:
    call   main_retrieve_session_key
    mov    rax, [rsp+188h+var_160]
    mov    rcx, [rsp+188h+var_160]
    cmp   cstime_Now_hash, rax
    je    loc_648C87

    ; Pink box highlights this section
    loc_648C87:
    mov    edx, 1
    loc_648CB7:
    mov    rax, [rsp+188h+var_C8], rax
    mov    [rsp+188h+var_A0], rcx
    mov    rcx, cs:off 8E8834 ; "110125200204206304400404"

```

(16,88) 0024B1FA 000000000064BCFA: main_main:loc_64BCFA [Synchronized with Hex View-1]

Main retrieve session key

Received a GET request on /aaa/icon.ico
Set-Cookie: 11D0d6VtL0x110
Set-Cookie: GC6Bz1bvlrvkYfOr=uvn6EdYlQ
Set-Cookie: QMfbny7qUn4gJpmphW0SL4cIMcxBoE5wxXpyznr5k
Set-Cookie: vu3LTg5R3fb+77a6ca7c5ab3e4c3a2ad6a5761a78a

127.0.0.1 - [02/Mar/2022 05:33:11] "GET /aaa/polls/db/ HTTP/1.1" 200 - Received a GET request on /aaa/polls/db/!

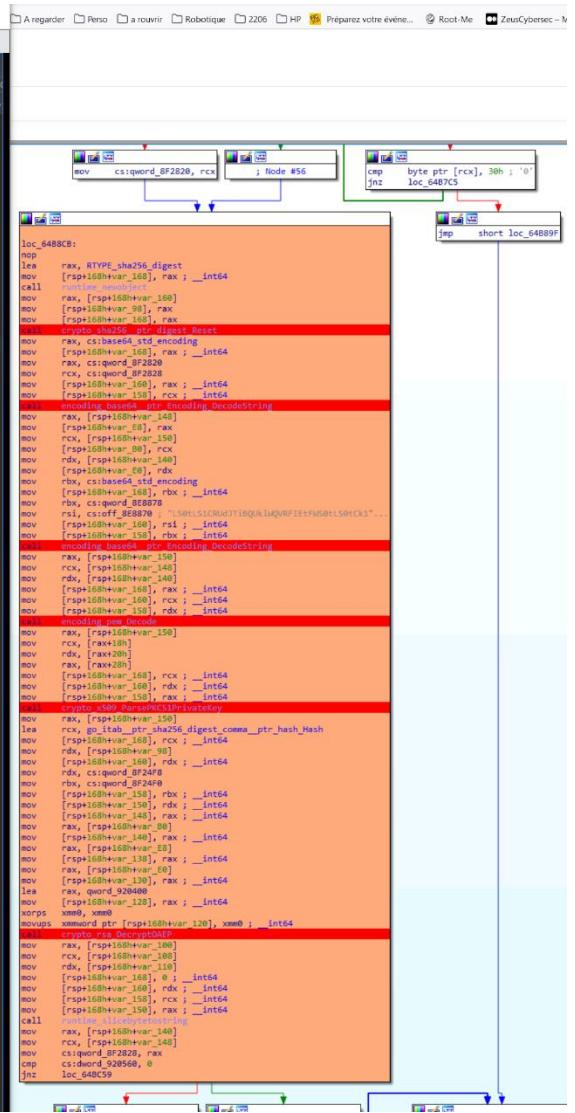
Received a GET request on /aaa/css/style.css!
Set-Cookie: B11D0d6VtL0x110
Set-Cookie: GC6Bz1bvlrvkYfOr=uvn6EdYlQ
Set-Cookie: QMfbny7qUn4gJpmphW0SL4cIMcxBoE5wxXpyznr5k
Set-Cookie: vu3LTg5R3fb+77a6ca7c5ab3e4c3a2ad6a5761a78a

127.0.0.1 - [02/Mar/2022 05:42:55] "GET /aaa/css/style.css HTTP/1.1" 200 - Received a GET request on /aaa/css/style.css!
Set-Cookie: B11D0d6VtL0x110
Set-Cookie: GC6Bz1bvlrvkYfOr=uvn6EdYlQ
Set-Cookie: QMfbny7qUn4gJpmphW0SL4cIMcxBoE5wxXpyznr5k
Set-Cookie: vu3LTg5R3fb+77a6ca7c5ab3e4c3a2ad6a5761a78a

127.0.0.1 - [02/Mar/2022 05:43:06] "GET /aaa/polls/db/ HTTP/1.1" 200 - Received a GET request on /aaa/polls/db/!
Set-Cookie: B11D0d6VtL0x110
Set-Cookie: GC6Bz1bvlrvkYfOr=3qnehpTqr9x
Set-Cookie: QMfbny7qUn4gJpmphW0SL4cIMcxBoE5wxXpyznr5k
Set-Cookie: vu3LTg5R3fb+77a6ca7c5ab3e4c3a2ad6a5761a78a

127.0.0.1 - [02/Mar/2022 07:31:40] "GET /aaa/polls/db/ HTTP/1.1" 200 - Received a GET request on /aaa/polls/db/!

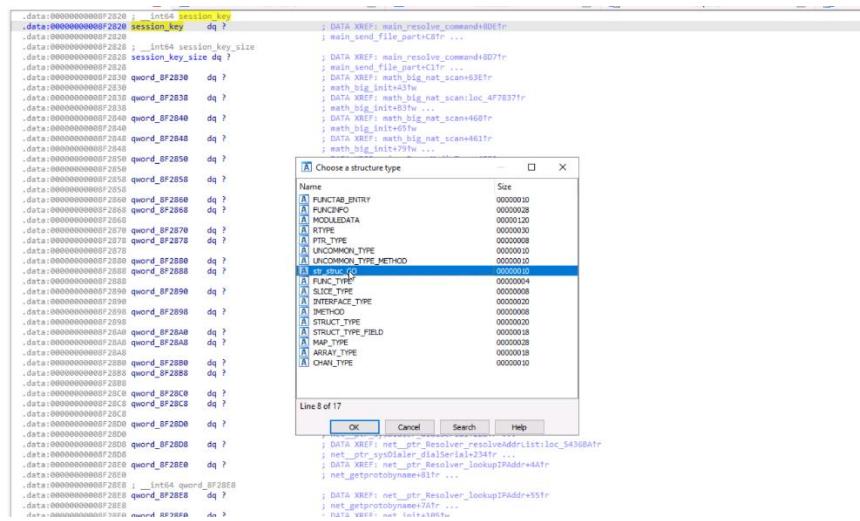
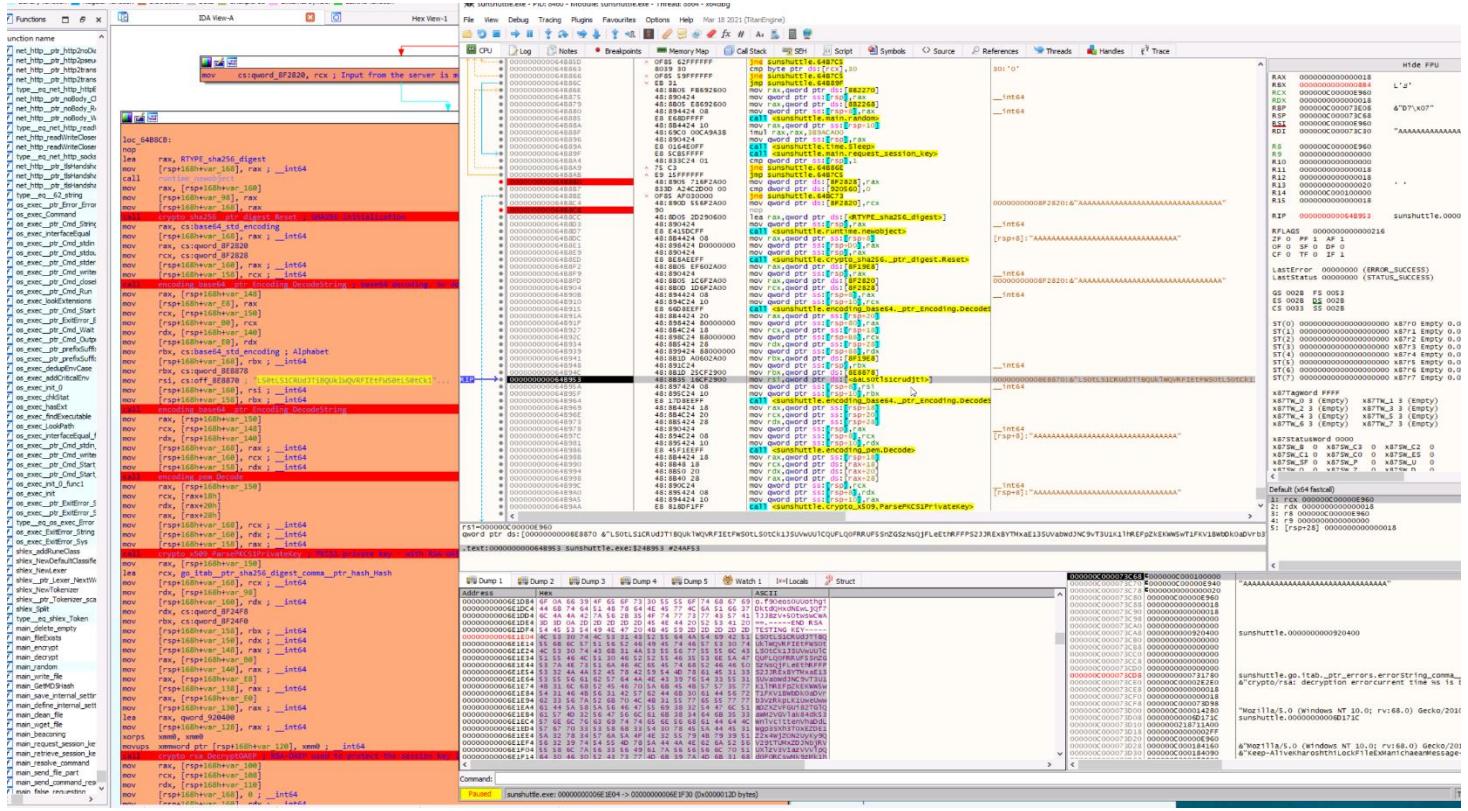
Crypto block



Address	OpCode	Registers	Comments
.text:000000000064A2A9	mov	rdx, [rsp+3C0h+var_398]	
.text:000000000064A2AE	lea	rbx, [rsp+3C0h+var_2F8]	
.text:000000000064A2B6	mov	[rsp+3C0h+var_3C0], rcx ; __int64	
.text:000000000064A2BA	mov	[rsp+3C0h+var_3B8], rax ; __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:000000000064A378	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_288], 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, nT0TimeOut)jvg2+s6h ; "vw3!Jg5hR3fhwirep: 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	

Incident Response Threat Analysis for E Corp

E CORP

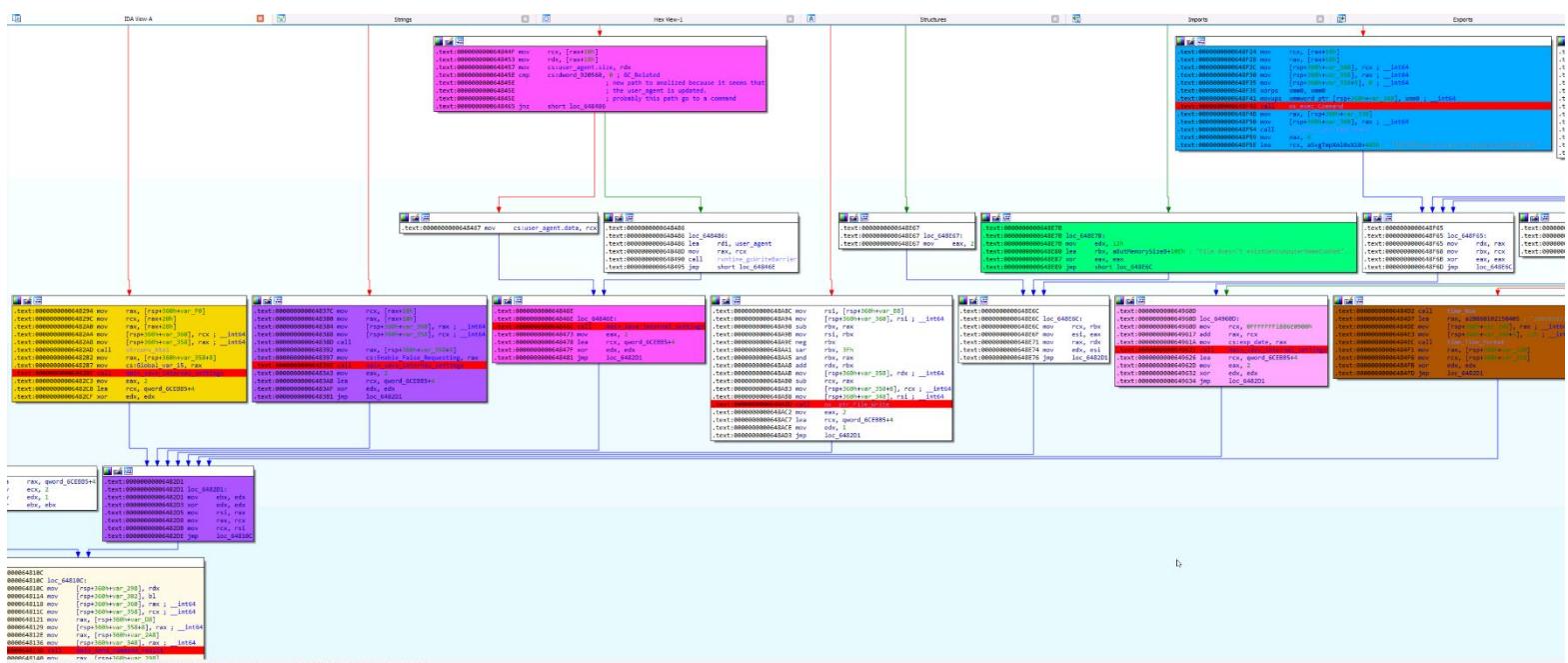


```
Received POST data from the client
Set-Cookie: GC6Bzibvr1kYoFr=S4KUbT6H8NY
Set-Cookie: QMFbry7qU=gWJpnphrWoSL4ciMCxB0e5wxXpyznr5k
Set-Cookie: vw3LTg5bR3fb=77a6ca7c5ab33ea4c3a2ad6a5761a78a
{b'QnwddoTiOkDj2u': [b'WTF0R0syMm1vS1JMNnI3MXQ4ZlxOanAzcnA5aF9Qd0w2V3o3aGtsaWhnRQ=='], b'zQy9aJHW8U8rpQN': [b'WHATEVER']}
}
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

The screenshot displays the IDA Pro interface with several windows open:

- IDA View-A**: Shows assembly code for a function. The assembly code includes instructions like mov, cmp, and jne, with comments such as "runtime_panicIndex".
- Strings**: A window showing various strings from the memory dump.
- Hex View-1**: A window showing the hex dump of memory starting at address 0x649540.
- Structures**: A window showing the memory dump.
- Imports**: A window showing imported functions.
- Memory dump windows (multiple windows):** These windows show memory dump sections for different ranges: 0x648830-0x64883F, 0x649090-0x64909F, 0x649695-0x64969F, 0x649E20-0x649E2F, 0x649F72-0x649F7F, and 0x649F80-0x649F8F. Each window contains assembly code, memory dump, and registers.



Python Scripts

The main scripts are available in the Sunshuttle_scripts.zip attached