# Secrets behind the Lazarus's VHD ransomware

seguranca-informatica.pt/secrets-behind-the-lazaruss-vhd-ransomware/

August 17, 2021



## Introduction

<u>Data encryption malware</u> is one of the most popular malware families in recent years and targets mass volumes of users and companies around the world. In this article, we will take a deep dive into a new VHD ransomware distributed in the wild by the Lazarus group — the criminals behind the <u>WannaCry</u> incident in 2017.

The VHD ransomware is one of the most dangerous threats today. This is because it tries to change system settings, disable some programs and functions of the target operating systems, and demands payment in cryptocurrency via a ransom note dropped during the ransomware execution. The Lazarus group have been using this ransomware, which uses methods typical of APT attacks but specialized in financial cybercrime.

The activity of the Lazarus Group surged in 2014 and 2015, where custom-tailored malware was used in its cyberattacks. The group has been linked to several major cyberattacks, including the 2014 Sony Pictures hack, several SWIFT banking attacks since 2016, and the WannaCry ransomware infection in May 2017.

# Lazarus's VHD ransomware

Researchers from Kaspersky <u>detailed</u> a new VHD ransomware used by the group between March and May 2020. The analyzed samples have been deployed over the network of the target enterprises, brute-forcing the SMB service on every discovered machine and using the

MATA malware framework documented by Kaspersky <u>here</u> and also used by the Lazarus group.

Initially, criminals gain access to the target infrastructures via widespread botnet infections such as Emotet and Trickbot families or by exploiting vulnerable VPN gateways. When criminals have a good understanding of the target's finances and IT processes and network, they escalated their privileges on the compromised systems and installed a backdoor that is part of the MATA malware framework. The criminals leverage the backdoor to control the Active Directory server and use it to deliver VHD ransomware to all systems in the target network within 10 hours using the help of a Python-based loader.

After finishing the infection chain with the ransomware deployment, criminals enter a new phase: negotiation.

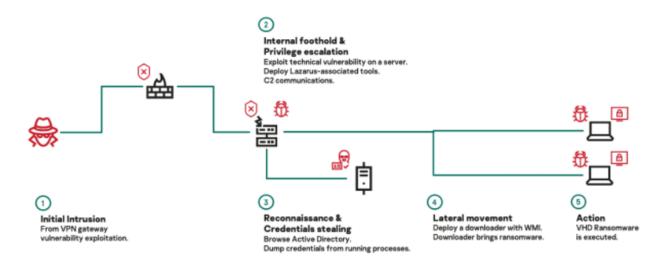


Figure 1: High-level diagram of the Hakuna MATA and VHD ransomware.

VHD is a standard ransomware tool that spreads through the drives connected to the target device, encrypts all the files and deletes all System Volume Information folders that prevent the impacted system could be restored. The ransomware binary is protected with VMProtect packer in order to prevent its analysis. The binary sections vmp0, vmp1 are the first signal of this packer.

.text	00013505	00001000	00013600
.rdata	000053C0	00015000	00005400
.data	000035F0	0001B000	00001400
.vmp0	003306CF	0001F000	00330800
.vmp1	0026F350	00350000	0026F400
.reloc	000030EC	005C0000	00003200
.rsrc	000001B2	005C4000	00000200

As observed in the PorEx diagram below, some parts of the binary were obfuscated, such as the malware config, ransom note details and so on.

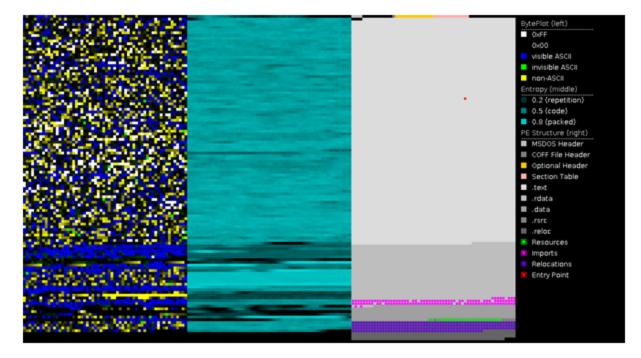


Figure 3: PortEx diagram — VHD ransomware.

One interesting detail this threat is related to some services that are suspended to preserve important files from modification such as Microsoft Exchange and SQL server. Next, the services suspended during the ransomware operation are presented.

.rdata:00418B0E	align 10h
.rdata:00418B10 aScStopMicrosof	db 'sc stop "Microsoft Exchange Active Directory Toplogy"',0
.rdata:00418B10	; DATA XREF: .text:00405131To
.rdata:00418846	align 4
.rdata:00418B48 aScStopMicros 0	db 'sc stop "Hicrosoft Exchange Anti-span Update"',0
.rdata:00418848	; DATA XREF: .text:00405139 <sup>†</sup> 0
.rdata:00418876	align 4
.rdata:00418B78 aScStopMicros 1	db 'sc stop "Microsoft Exchange Compliance Audit"',0
.rdata:00418B78	; DATA XREF: .text:004051411o
.rdata:00418BA6	align 4
.rdata:00418BA8 aScStopMicros_2	db 'sc stop "Hicrosoft Exchange Compliance Service"',0
.rdata:00418BA8	; DATA XREF: .text:0040514910
.rdata:00418BD8 aScStopMicros_3	db 'sc stop "Microsoft Exchange DAG Management"',0
.rdata:00418BD8	; DATA XREF: .text:004051511o
.rdata:00418C04 aScStopMicros_4	db 'sc stop "Hicrosoft Exchange Diagnostics"',0
.rdata:00418C04	; DATA XREF: .text:00405159To
.rdata:00418C2D	align 10h
.rdata:00418C30 aScStopMicros_5	db 'sc stop "Microsoft Exchange EdgeSync"',0

Figure 4: Services stoped during the ransomware execution.

Here's a complete list of stoped services during VHD ransomware execution:

```
sc stop "Microsoft Exchange Anti-spam Update"
sc stop "Microsoft Exchange Active Directory Toplogy"
sc stop "Microsoft Exchange Compliance Audit"
sc stop "Microsoft Exchange Compliance Service"
sc stop "Microsoft Exchange DAG Management"
sc stop "Microsoft Exchange Diagnostics"
sc stop "Microsoft Exchange EdgeSync"
sc stop "Microsoft Exchange Frontend Transport"
sc stop "Microsoft Exchange Health Manager"
sc stop "Microsoft Exchange IMAP4"
sc stop "Microsoft Exchange Health Manager Recovery"
sc stop "Microsoft Exchange Information Store"
sc stop "Microsoft Exchange IMAP4 Backend"
sc stop "Microsoft Exchange Mailbox Assistants"
sc stop "Microsoft Exchange Mailbox Transport Delivery"
sc stop "Microsoft Exchange POP3"
sc stop "Microsoft Exchange POP3 Backend"
sc stop "SQL Server Agent (TESTINSTANCE)"
sc stop "SQL Server (TESTINSTANCE)"
```

Some important Windows calls are also executed during the malware operation. Some of them are described in a brief way below to a better understanding of how the malware works.

#### Calls loaded from kernel32.dll

CreateFileW()	Creates or opens a file or I/O device during malware execution.
FindFirstFileW()	Searches a directory for a file or subdirectory with a name that matches a specific name (or partial name if wildcards are used).
FindNextFileW()	Continues a file search from a previous call to the FindFirstFile, FindFirstFileEx, or FindFirstFileTransacted functions.
GetLogicalDrives()	Retrieves a bitmask representing the currently available disk drives.
IsDebuggerPresent()	Determines whether the calling process is being debugged by a user-mode debugger (used to prevent malware running in virtual machines).
IsProcessorFeaturePresent()	Determines whether the specified processor feature is supported by the current computer (used in anti-debug techniques).
CreateMutexW()	Creates or opens a named or unnamed mutex object (used to prevent further ransomware infections).

**SHEmptyRecycleBinW()** Empties the Recycle Bin on the specified drive.

Call loaded from user32.dll

**ShowWindow()** Sets the specified window's show state. Used to hide the ransomware window during its execution.

Every time a file is encrypted, it is renamed. The extension "**.vhd**" is appended to the end of the file name.

tout 0000007 loo b0007.		
.text:00404BF7 loc_404BF7: .text:00404BF7		; CODE XREF: .text:00404A73†j
.text:00404BF7	5011	; .text:00404A87Ťj
.text:00404BFD	nov	edx, [ebp-4B84h]
	nov	eax, [ebp-4B80h]
.text:00404C03	push	edx
.text:00404C04	push	eax
.text:00404C05	lea	edi, [ebp-120h]
.text:00404C0B	mov	esi, ebx
.text:00404C0D	call	sub_405AD0
.text:00404C12	push	104h
	lea	
.text:00404C22	push	104h
.text:00404C27	push	ecx
.text:00404C28	call	sub_4076F3
.text:00404C2D	push	104h
.text:00404C32	push	offset a_vhd ; ".vhd"
.text:00404C37	lea	edx, [ebp-328h]
.text:00404C3D	push	104h
.text:00404C42	push	edx
.text:00404C43	call	sub_407B1F
.text:00404C48	add	
.text:00404C4B	push	ebx
.text:00404C4C	call	sub_734F29
.text:00404C17 .text:00404C1C .text:00404C22 .text:00404C27 .text:00404C28 .text:00404C20 .text:00404C32 .text:00404C37 .text:00404C30 .text:00404C42 .text:00404C48 .text:00404C48 .text:00404C48	push push call push lea push call add push	offset unk_41CF20 ecx, [ebp-328h] 104h ecx sub_4076F3 104h offset a_vhd ; ".vhd" edx, [ebp-328h] 104h edx   sub_407B1F esp, 28h ebx

*Figure 5:* Ransomware extension (.vhd) is appended to the file name whenever a file is encrypted.

The ransom note is also dropped into the desktop folder "*HowToDecrypt.txt*" with the instructions for paying the ransom and recovering the damaged files.

Becycle Bin ADRLBerg DIWM_XZ., PKB944-y0., wBP971M wARD ots	
Czuwyłs Bw-IBRADe., cytyc.0.gr OdyżółaC., Wilitizując	
Acrobat Reader DC Reader DC	davidsummary.jpg, handsalmost.png.v purchaseboth.jpg.v runeven.jpg.vhd vhd hd hd
Maatila Frieda Frieda	HowToDecrypttxt-Notepad File Edit Format View Help All data on your pc were encrypted with strongest encryption method.
Geogle COLIKEAN LUNXEggi TVuqOTLAL SLOVING Choine	The only way to get your data back is to purchase unique key for you. * You can get cheaper price if you contact us as soon as possible. *
LBYERDING, DapHerrHo., LBAX12W., VB-siglade zTu30-Hk.,	After three days from now, it will be difficult to recover your data. Good Luck.
Sliclafou,-vWRQM. nj0Y.jog vMXVrG2bn. splasov32	contact address: miclejaps@msgden.net stevenjoker@msgden.net
SQIPISHMA _E MIZHAT PATRIES,N., WARDONG., HowToDec	

Figure 6: Damage files and ransom note dropped into the desktop folder.

The ransomware is written in C++ and the files are encrypted with a combination of AES-256 in ECB mode and RSA-2048. The ransomware uses Mersenne Twister as a source of randomness, but unfortunately for the victims, the RNG is reseeded every time new data is consumed. The combination of ECB and AES is not semantically secure, which means the patterns of the original clear data are preserved upon encryption.

VHD implements a mechanism to resume operations if the encryption process is interrupted. For instance, for files larger than 16MB, the ransomware stores the current cryptographic materials on the hard drive, in cleartext. This information is not deleted securely afterwards, which implies there may be a chance to recover some of the damaged files. Chrysanthemum.jpg.vhd

000D6A70       5D 39 4D 3B 33 03 31 18 2C 6A C5 D1 15 17 E8 86       ]9M;3.1.,jÅÑè^         000D6A80       FA 1D 5A 87 0A 2C 34 FF 30 94 73 04 23 85 40 14       ú.2‡.,4ÿ0"s.‡@.         000D6A90       B0 BD F7 B4 89 D1 E6 02 2B A4 5D FA 28 0E 5A 33       °4±÷ 'tiŇe.+*]ú(.23         000D6A80       7C AE 8B EE 97 87 06 02 DC DF 11 FA E4 8E 5B C6       [@xi=tÜB.úáž[E         000D6A80       79 33 8A 38 6E 43 A9 D8 8B 1E B5 7F 08 21 8F 64       y3Š8nC@Ø<.p!.d         000D6A80       FF F6 06 5E 8C A3 48 C1 93 6C 90 67 10 AC D9 0C       yö.^G£HÁf1.gŋù.         000D6A20       FF F6 06 5E 9C A3 48 C1 93 6C 90 67 10 AC D9 0C       yö.^G£HÁf1.gŋù.         000D6A20       67 39 36 5B 90 5E 23 EA D6 C2 A4 2A D2 AD C; 096[.^#ėĉÅxė.       Geof[.^#ėĉÂx*č.         000D6A20       67 61 22 33 12 29 2E 6C 02 07 90 01 4A 3A 37 DD       ia"3.).1J:7Ý         000D6AF0       9B BE 3F 07 6C 48 EC 7B 0F 2B E4 C5 85 06 A5 D8       >%?.1Hì(.+äÅ¥Ø	7 08 09 0A 0B 0C 0D 0E 0F Decoded text	
000D6A80       FA 1D 5A 87 0A 2C 34 FF 30 94 73 04 23 85 40 14          ú.2\$±, 4ÿ0"s.\$±@.          000D6A90       B0 BD F7 B4 89 D1 E6 02 2B A4 5D FA 28 0E 5A 33 <sup>0</sup> 4±; 'tN&±.+#]ú(.Z3          000D6A00       FC AE 8B EE 97 87 06 02 DC DF 11 FA E4 8E 5B C6           [\$\$\scircles\$tilde{1}\$=\$tilde{1}\$\$\$U\$.\$\$\scircles\$tilde{2}\$ti	8 2C 6A C5 D1 15 17 E8 88 19M:3.1 jÅNe*	
000D6A90       B0 BD F7 B4 89 D1 E6 02 2B A4 5D FA 28 0E 5A 33       **±÷ 'thme.+=1)ú(.23         000D6AA0       7C AE 6B EE 97 87 06 02 DC DF 11 FA E4 6E 5B C6        @xi=+.UB.úaž[E         000D6AB0       79 33 8A 38 6E 43 A9 D8 8B 1E B5 7F 08 21 8F 64       y358nC©Ø<.µ!.d		
000D6AA0       7C AE 8B EE 97 87 06 02 DC DF 11 FA E4 8E 5B C6       [@k1-#ÜB.úäŽ[Æ         000D6AB0       79 33 8A 38 6E 43 A9 D8 8B 1E B5 7F 08 21 8F 64       y3Š8nC©Ø<.µ!.d		
000D6AB0 79 33 8A 38 6E 43 A9 D8 8B 1E B5 7F 08 21 8F 64 y3Š8nCCØ<.u!.d 000D6AC0 FF F6 06 5E 8C A3 48 C1 83 6C 90 67 10 AC D9 0C y0.^C£HÁf1.gÙ. 000D6AD0 C7 3A 6F 39 36 5B 90 5E 23 EA D6 C2 A4 2A D2 AD Ç:096[.^#êô¤*Ò. 000D6AE0 69 61 22 33 12 29 2E 6C 02 07 90 01 4A 3A 37 DD ia"3.).1J:7Ý 000D6AF0 9B BE 3F 07 6C 48 EC 7B 0F 2B E4 C5 85 06 A5 D8 >%?.1Hì{.+āÂ¥Ø		
000D6AC0         FF F6         06         5E         8C         A3         48         C1         83         6C         90         67         10         AC         D9         OC         90%.^G£HÁf1.gÚ.         AES256(FileKey, FileContents)           000D6AD0         C7         3A         6F         39         36         5B         90         5E         23         EA         D6         C2         A4         2A         D2         AD         C:096[.^##êÔŤ*Ò.         AES256(FileKey, FileContents)	R RR IF R5 75 08 21 RF 54 V3580C004 H L d	
000D6AD0 C7 3A 6F 39 36 5B 90 5E 23 EA D6 C2 A4 2A D2 AD Ç:096[.^#êĈä*Ò. 000D6AE0 69 61 22 33 12 29 2E 6C 02 07 90 01 4A 3A 37 DD ia"3.).1J:7Ý 000D6AF0 9B BE 3F 07 6C 48 EC 7B 0F 2B E4 C5 85 06 A5 D8 ≫4?.1Hì{.+āÅ¥Ø	ALS256(EleKey EleContents)	
000D6AE0 69 61 22 33 12 29 2E 6C 02 07 90 01 4A 3A 37 DD ia"3.).1J:7Ý 000D6AF0 9B BE 3F 07 6C 48 EC 7B 0F 2B E4 C5 85 06 A5 D8 >4?.1Hì{.+aÅ¥Ø		_
000D6AF0 9B BE 3F 07 6C 48 EC 7B 0F 2B E4 C5 85 06 A5 D8 >4?.1H1(.+äÅ¥Ø		
	B OF 2B E4 C5 85 06 A5 D8 >4?.1Hi(.+äÅ¥Ø	
000D6B00 A0 7D EA B5 22 05 06 EE 00 A3 B0 9E 87 6E DF 1D }êu"î.£°ž‡nß.		
000D6B10 E3 FC CD F2 6F 70 E0 90 E1 64 D2 E6 D1 9D 1D 3A āufóopà.ádðæÑ:	O E1 64 D2 E6 D1 9D 1D 3A ăüÍòopà.ádÒæÑ:	
000D6B20 A0 51 ED 28 6D 0E 2F 1C 90 44 E6 88 B7 6A 55 19 Qi(m./Dæ^ jU.	C 90 44 E6 88 B7 6A 55 19 Qi(m./Dæ^-jU.	
000D6B30 22 6B 0D 00 00 00 00 00 08 8F 8B 59 29 FD 45 A3 8k	0 08 8F 8B 59 29 FD 45 A3 Sk	
000D6B40 08 D4 BF 4C E9 C8 4A 39 7B A0 AF D0 2F D9 CF 00 .02LéÈJ9{ D/ÙI. Original Filesize		
000D6B50 AC 40 D1 CD E6 A2 5A B9 36 17 E3 67 55 93 27 45 ⊣@ÑÍæcZ³6.āgU™'E	9 36 17 E3 67 55 93 27 45 ¬@Ňĺæ¢Z³6.ãgU™'E	
000D6B60 8C 19 EC 76 83 4C 3D E9 FB FD 00 80 B4 3B 0A 40 €.ivfL=éûý.€';.0	9 FB FD 00 80 B4 3B 0A 40 €.ìvfL=éûý.€´;.@	
000D6B70 BB D4 4B 47 80 AB B2 E1 84 3D 27 C7 52 86 9A B5 »ÔKG€«*á,,="ÇR†šµ RSA1024(PublicKey, FileKey)	1 84 3D 27 C7 52 86 9A B5 »ÔKG€«°á"='ÇR†ăµ RSA1024(PublicKey, FileKey)	
000D6B80 98 9F 82 5E ED 2C 02 F4 46 8B EB BD C2 02 EB 03 *Y,^i,.oF(=44.e.	4 46 8B EB BD C2 02 EB 03 "Ŷ,^i,.ôF<ë\Â.ë.	_
000D6B90 5C F7 FC 86 F8 41 92 30 F6 A8 B4 51 E3 7E 5C 8D \+üt@A'08""Qā~\.	0 F6 A8 B4 51 E3 7E 5C 8D \÷ütøÅ'0ö"'Qã~\.	
000D6BA0 30 25 B3 68 C8 EC 52 5E D2 5C FF AE 77 8F B0 B8 0%⇒hÈìR^Ò\ŷ@w.°,	E D2 5C FF AE 77 8F B0 B8 0%°hÈìR^Ò\ý®w.°,	
000D6BB0 58 37 A8 2D C7 10 0A DF BB D0 61 7D F8 D0 F8 CA X7"-Ç8»Đa}@D@Ê	F BB D0 61 7D F8 D0 F8 CA X7"-Çß»Đa}øĐøÊ	
000D6BC0 <mark>E9 4D 38 F5 63 6B DC 10 4E 00 19 94 9B FA FE F8</mark> éM8ðckÜ.N"×úþø	0 4E 00 19 94 9B FA FE F8 éM8őckÜ.N">úpø	
000D6BD0 57 C9 2E 04 F3 12 BD 3A 35 0E CE 82 BE E3 10 F6 WÉó.%:5.Î,%&.ö		
000D6BE0 3F CA C4 69 EC 5F C7 D5 46 3C 69 2C 49 AE CB 74 ?ÊĂil_ÇÕF <i,i⊗ët< td=""><td>5 46 3C 69 2C 49 AE CB 74 ?ÊĂiì_ÇÕF<i,i®ët< td=""><td></td></i,i®ët<></td></i,i⊗ët<>	5 46 3C 69 2C 49 AE CB 74 ?ÊĂiì_ÇÕF <i,i®ët< td=""><td></td></i,i®ët<>	
000D6BF0 8D F8 60 E4 C3 E9 4F D9 E1 9C 76 0D CA F9 62 1D .ø`äÅéOÙáœv.Êùb.	9 E1 9C 76 0D CA F9 62 1D .ø`äÅéOÙáœv.Êùb.	
000D6C00 08 42 DB 88 28 7A 35 EB 31 89 A8 52 86 A9 2F 49 .BÛ<(z5ël%"R+©/I		
000D6C10 16 3B 7D F5 79 9D A0 4F 0E 08 69 3F 17 A1 54 17 .;}õy. 0i?.;T.		
000D6C20 85 7E BD CF 4F 11 65 6D E4 D4 70 E5 FE FC 48 67		
000D6C30 CE 90 35 66 F0 CE 27 53 1.5181'S	1.5fðî'S	
about and the sector 11 means to take (interim the sector)		
<pre>1unsigned intstdcall mersenne_twister(int min, int max)</pre>	stdcall mersenne_twister(int min, int max)	

```
2 {
     int seed; // eax
  3
  4 unsigned int seed_; // ecx
  5 int i; // eax
  6 int v6; // [esp+8h] [ebp-1390h] BYREF
     int v7[1248]; // [esp+Ch] [ebp-138Ch]
  7
     int v8; // [esp+138Ch] [ebp-Ch]
  8
  9
      seed = std::tr1::_Random_device();
10
     v7[0] = seed;
11
12
     seed = seed;
13
     v8 = -1;
14
     for (i = 1; i < 624; ++i)
 15
      {
16
       seed = i + 0x6C078965 * ((seed >> 30) ^ seed_);
17
      v7[i] = seed ;
 18
     }
19
     v6 = 624;
      return mersenne_twister_get_next_(min, max, (int)&v6);
20
21 }
```

*Figure 7:* Details about cryptographic algorithms and Mersenne Twister block with the RNG seed.

A strategy often used by malware developers is the usage of mutex entries to prevent further infections when a target device has already been infected. VHD ransomware creates during its execution the mutex *"AEEAEE SET"*, as observed below.

"\Sessions\1\BaseNamedObjects\AEEAEE SET" "AEEAEE SET"

The malware's plan is not very complex, as presented by the MITRE ATT&CK matrix below. Ony some functions to hide its presence and calls to scan the target device and drives and also to access the system files are used by the malicious binary.

### Preventing against ransomware incidents

There is not a perfect formula to fight ransomware incidents, but some measures can be enumerated to prevent some situations:

- Provide social engineering training and explain to employees how following simple rules helps to avoid ransomware incidents.
- Ensure that software, applications and systems are up to date.
- Use endpoint protection solutions to prevent malicious infections.
- Use vulnerability management and monitoring systems to identify potential unpatched flaws and to detect incidents in real time.
- Use canary files to detect ransomware early.
- Perform cybersecurity audits and mitigate any weaknesses discovered in order to prevent attacks in the wild, both from the external and internal perspective.

The article was initially published by Pedro Tavares on <u>resources.infosecinstitute.com</u>. All rights reserved ® infosecinstitute.com



Pedro Tavares

**Pedro Tavares** is a professional in the field of information security working as an Ethical Hacker/Pentester, Malware Researcher and also a Security Evangelist. He is also a founding member at CSIRT.UBI and Editor-in-Chief of the security computer blog <u>seguranca-informatica.pt</u>.

In recent years he has invested in the field of information security, exploring and analyzing a wide range of topics, such as pentesting (Kali Linux), malware, exploitation, hacking, IoT and security in Active Directory networks. He is also Freelance Writer (Infosec. Resources Institute and Cyber Defense Magazine) and developer of the <u>0xSI\_f33d</u> – a feed that compiles phishing and malware campaigns targeting Portuguese citizens.

Read more here.