Quick Heal

Security Simplified

Deconstructing RansomExx

aka Defray777 Ransomware – Memory run fileless malware

Author Aravind Raj

Quick Heal
Security Simplified

Table of Content

Introduction

RansomExx operation, also known as Defray777 (variants of the RansomEXX ransomware family.), is operated by the Gold Dupont threat group that has been active since 2018. The usual targets of these multi-staged human-operated attacks are Government and Educational Institutions, Healthcare, Manufacturing, Construction, and Engineering sectors. The high-profile attacks in 2020 against the Texas Department of Transportation (TxDOT), Brazilian government court, Konica Minolta, and others witnessed these destructive ransomware attacks.

These type of operations uses multi-vector extortion techniques to increase the chances of payments. The hackers then use the encrypted data to blackmail the target organization after exfiltrating sensitive information. In 2021, their targets, CNT Ecuador and the Italian Luxury fashion house Ermenegildo Zegna, apparently didn't pay the ransom, which resulted in their sensitive data being leaked on the dark web.

Interestingly, this attack also has a Linux variant, a first for ransomware. It also allows them to target a more comprehensive set of organizations. And this additional support for Linux systems makes this attack significantly more lethal.

Infection Chain

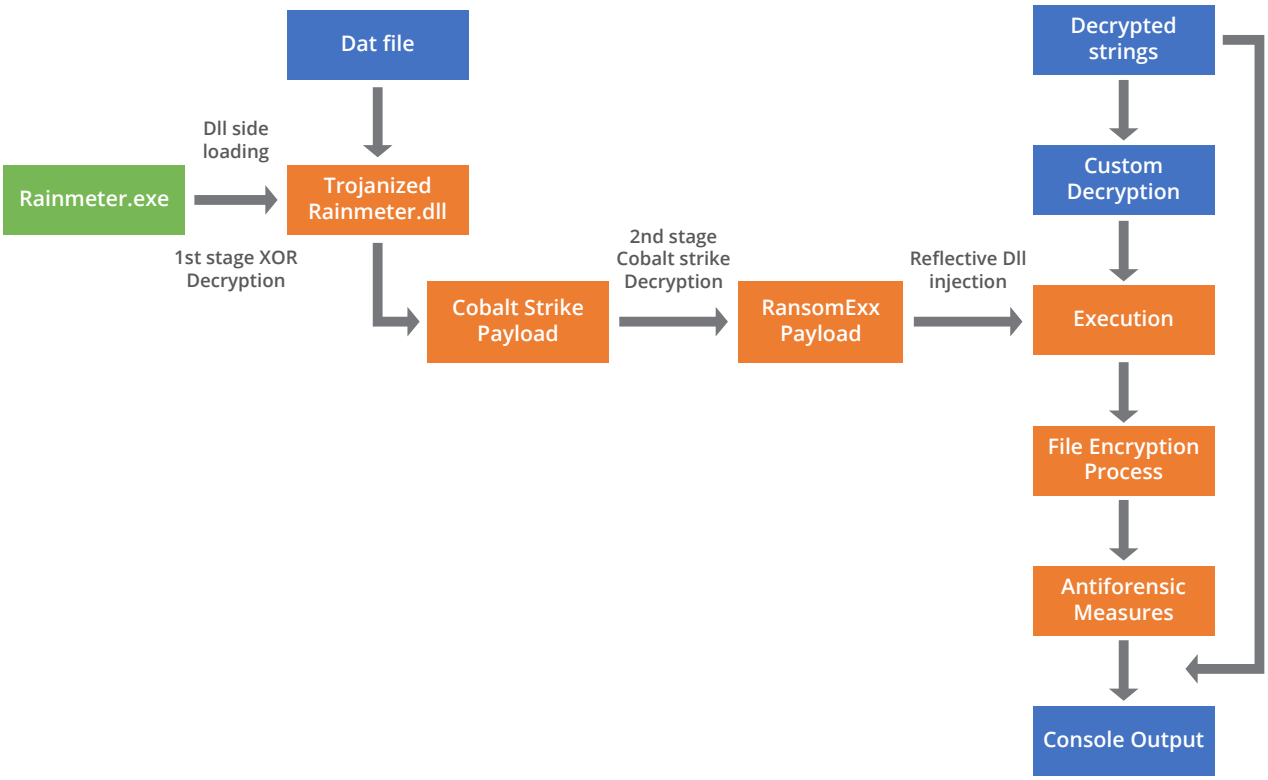


Figure 1 Infection Chain of the attack

The initial intrusion starts with the delivery of Vatet, a custom loader for the Cobalt Strike framework. The loader is usually spread via

- Water Hole attack,
- Exploit CVE-2019-19781 (Citrix Application Delivery Controller (ADC)),
- Brute force RDP endpoints

And drops the trojanized version of Notepad++, Rainmeter.exe, Upgrade.exe, etc.

Detailed Technical Analysis

1. Vatet Loader:

Let's take the example of a Vatet Loader sample of trojanized Notepad++ executable. The source code of the open-source Notepad++ is modified just with a few lines of code to look out for a DAT file (payload file) in a specific location. If it is present, it is decrypted, and control is transferred to the decrypted payload.

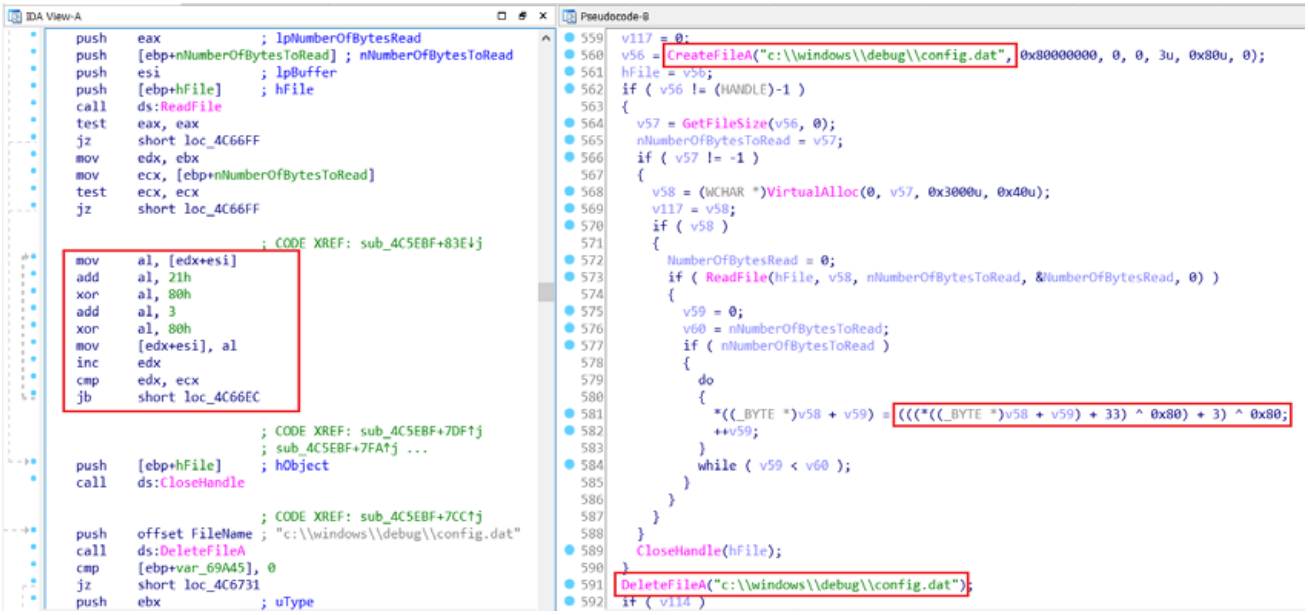


Figure 2 Vatet Loader, Loading Payload and decrypting

From "Figure 2," we see that the code is looking out for a payload file from its hard-coded location "**c:\\windows\\debug\\config.dat**". The result from API - "**CreateFileA**" is compared with "-1"; if the file is not found, the code is switched to regular notepad++ operation. Else the contents are copied to memory. The decryption process happens, an XOR operation with key "**0x80**," which varies depending upon the loader to decrypt, and the original config.dat file is deleted.

2. Rainmeter.dll

In another scenario of the Vatet loader, we have seen the use of **the DLL side loading technique** to execute the ransomware code. Attackers can use this technique to execute malicious DLLs that mimic legitimate ones. This technique has been used in many APTs to avoid detection here. The DLL is trojanized, containing the malicious code to load the payload file from the hardcoded location and decrypt it.

```
87 LABEL_15:
88   if ( FindResourceW((HMODULE)0x400000, (LPCWSTR)1, (LPCWSTR)3) )
89   {
90     v15 = LoadLibraryW(L"Rainmeter.dll");
91     if ( v15 && (v16 = GetProcAddress(v15, (LPCSTR)1)) != 0 )
92     {
93       result = ((int (__cdecl *)(int))v16)(v4);
94     }
95     else
96     {
```

Figure 3 Dll loading in Rainmeter.exe

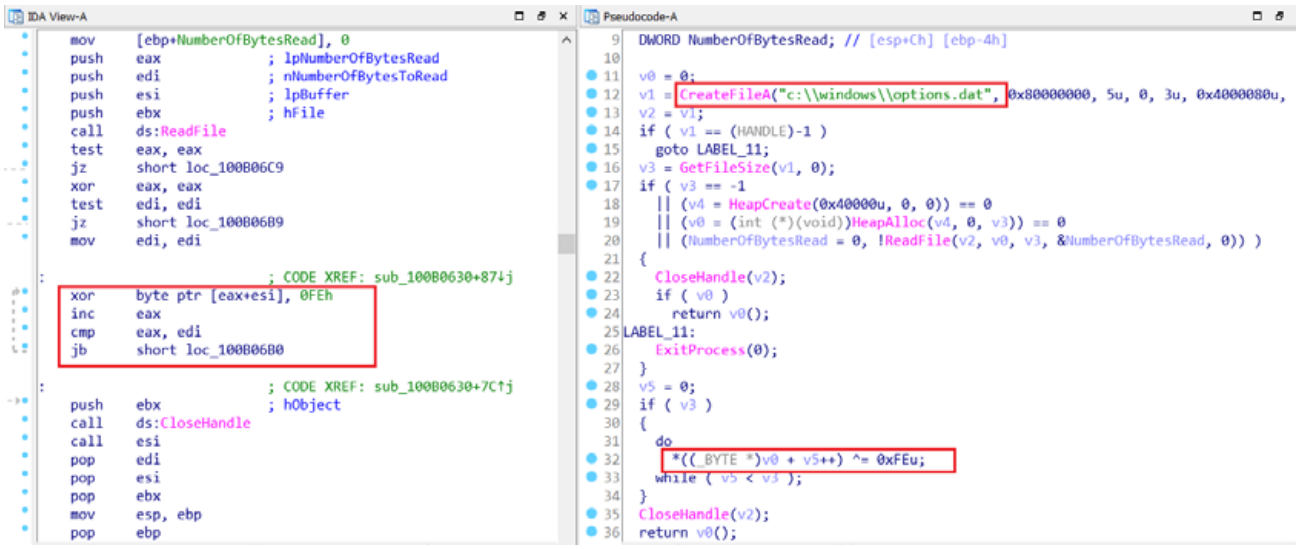


Figure 4 RainMeter.dll Decryption Loop

In this Variant the DLL payload is present in the hard-coded location "**c:\\windows\\options.dat**" and also the XOR key is "0xFE".

3. 2nd stage Decryption

After the XOR operation, injects the config.dat decrypted code into its memory and then executes the payload.

Again, there is a decryption loop in the payload that resembles the similar pattern of Cobalt Strike Beacon.

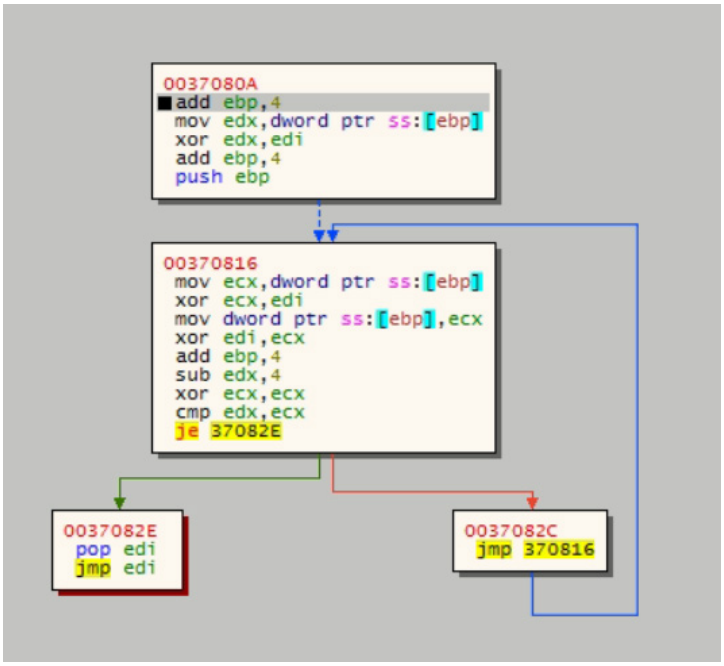


Figure 5 Cobalt strike Decryption loop

The decryption for payload can be done for these specific files using the python code given below

```
import struct

def xor(a, b):
        return bytearray([a[0]^b[0], a[1]^b[1], a[2]^b[2], a[3]^b[3]])

with open("Payload", "rb") as file:
        data = file.read()

Adrs = 0x4c
key = data[Adrs:Adrs+4]
size = struct.unpack("I", xor(key, data[Adrs+4:Adrs+8]))[0]

out = bytearray()
i = Adrs+8
while i < (len(data) - Adrs - 8):
        d = data[i:i+4]
        out += xor(d, key)
        key = d
        i += 4

with open("Decrypted_Payload.out", "wb+") as file:
        file.write(out)
```

For decoding config data for other variance, it can be done using the web page's code [1]. The decrypted payload and decryption process may differ depending upon the payload dropped by the Malware authors.

Detailed Analysis of RansomExx

1. Memory Execution

This malware is executed in memory by Cobalt strike and delivered by Vatet loader. After the second decryption stage, we get the payload file, the ransomware file.
This malware is reflectively loaded using DLL "?ReflectiveLoader@@YGKPAX@Z". In this technique, the malware is executed directly from memory rather than from disk, making it a file-less malware. This also means that there is no file dropped in the system. Thus, it evades static generic detections done by AV products making it only targetable by behavior and memory-based detections.

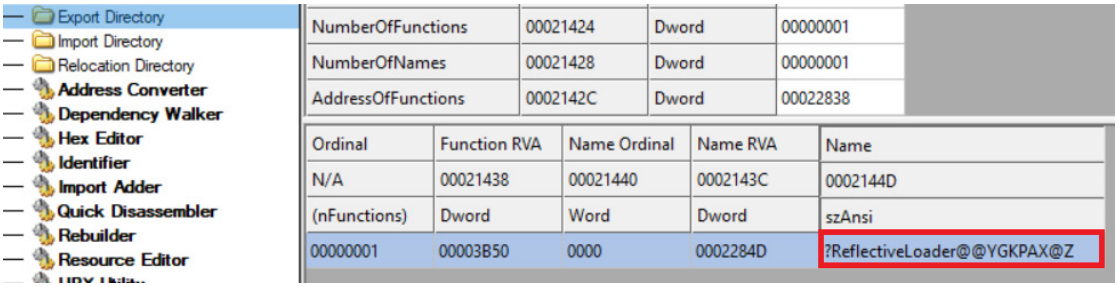


Figure 6 Exported Reflective DLL

2. Ransom Execution

Next, let's look at the step-by-step process of ransomware activities done by Ransomware.

1. The execution starts by getting the name of the computer. If it fails, then it assigns it to "DEFAULT COMPNAME "and continues to get the number of processors available in the system.
2. Then Compute the MD5 Hash of the computer name.
3. Convert the result to a string with API - "StringFromGUID2".
4. Create mutex using the string with a call to API - "CreateMutexW."

```
15  v0 = 0;
16  if ( sub_F43B0(*(const BYTE **)dword_1183BC, strlen(*(const char **)dword_1183BC), (BYTE *)&rguid) )
17  {
18    LOBYTE(rguid.Data1) = 120;
19    if ( StringFromGUID2(&rguid, &sz, 260) )
20    {
21      CreateMutexW(0, 0, &sz);
22      if ( GetLastError() == 183 )
23      {
```

Figure 7 Mutex Creation

1. Gets a list of all logical drives on the system using API - "GetLogicalDriveStringsW."
2. Since RansomEXX is a custom-packed malware, it decrypts some strings necessary for its operation (displays in console output).

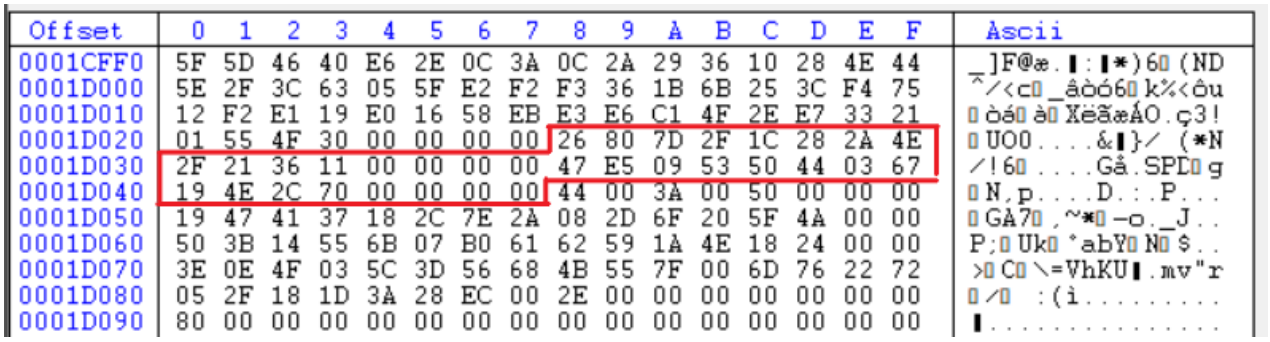
3. Custom String Decryption

The encrypted strings are present in the file as ciphers, which are decrypted by using the custom decryption method of using Bitwise "**AND**" operation of "**7F**" with index (to not exceed ASCII letters). After that, you can add them with Cypher1, and the resultant value is computed **XOR** with Cypher2, which is present in the file. It gives us decrypted strings.

```
do
{
  v5 = v19;
  *((_BYTE *)&v12 + v4) = byte_11DE438[v4] ^ (byte_11DE428[v4] + (v4 & 0x7F));
  byte_11DE42A[v5 + v4] = byte_11DE439[v4] ^ (byte_11DE429[v4] + ((v4 + 1) & 0x7F));
  byte_11DE42B[v18 + v4] = byte_11DE43A[v4] ^ (byte_11DE42A[v4] + ((v4 + 2) & 0x7F));
  byte_11DE42C[v17 + v4] = byte_11DE43B[v4] ^ (byte_11DE42B[v4] + ((v4 + 3) & 0x7F));
  byte_11DE42D[v16 + v4] = byte_11DE43C[v4] ^ (byte_11DE42C[v4] + ((v4 + 4) & 0x7F));
  v6 = byte_11DE42D[v4] + ((v4 + 5) & 0x7F);
  v4 += 6;
  *((_BYTE *)&v12 + v4 - 1) = byte_11DE437[v4] ^ v6;
}
while ( v4 < 12 );
```

Figure 8 Custom Decryption Loop

The Encrypted Cypher are present inside the file hardcoded –



Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Ascii
0001CFF0	5F	5D	46	40	E6	2E	0C	3A	0C	2A	29	36	10	28	4E	44	_]F@æ.▌:▌*)6▯(ND
0001D000	5E	2F	3C	63	05	5F	E2	F2	F3	36	1B	6B	25	3C	F4	75	^/<c▯_âòó6▯k%<ôu
0001D010	12	F2	E1	19	E0	16	58	EB	E3	E6	C1	4F	2E	E7	33	21	▯òá▯à▯XëãæÁO.ç3!
0001D020	01	55	4F	30	00	00	00	00	26	80	7D	2F	1C	28	2A	4E	▯UO0....&▌}/ (*N
0001D030	2F	21	36	11	00	00	00	00	47	E5	09	53	50	44	03	67	/!6▯....Gå.SPD▯g
0001D040	19	4E	2C	70	00	00	00	00	44	00	3A	00	50	00	00	00	▯N,p....D.:.P...
0001D050	19	47	41	37	18	2C	7E	2A	08	2D	6F	20	5F	4A	00	00	▯GA7▯,~*▯-o._J..
0001D060	50	3B	14	55	6B	07	B0	61	62	59	1A	4E	18	24	00	00	P;▯Uk▯°abY▯N▯$..
0001D070	3E	0E	4F	03	5C	3D	56	68	4B	55	7F	00	6D	76	22	72	>▯O▯\=VhKU▌.mv"r
0001D080	05	2F	18	1D	3A	28	EC	00	2E	00	00	00	00	00	00	00	▯/▯ :(i........
0001D090	80	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	▌...............

Figure 9 Hardcoded Crypted Strings

Sample of decryption operation -

code	Key	Index	Decryption operation	Decrypted Ascii
0x26	0x47	0	((0 & 7F) + 0x26) ^ 0x47	a
0x80	0xE5	1	((1 & 7F) + 0x80) ^ 0xE5	d
0x7D	0x9	2	((2 & 7F) + 0x7D) ^ 0x9	v
0x2F	0x53	3	((3 & 7F) + 0x2F) ^ 0x53	a
0x1C	0x50	4	((4 & 7F) + 0x1C) ^ 0x50	p
0x28	0x44	5	((5 & 7F) + 0x28) ^ 0x44	i
0x2A	0x3	6	((6 & 7F) + 0x2A) ^ 0x3	3
0x4E	0x67	7	((7 & 7F) + 0x4E) ^ 0x67	2
0x2F	0x19	8	((8 & 7F) + 0x2F) ^ 0x19	.
0x21	0x4E	9	((9 & 7F) + 0x21) ^ 0x4E	d
0x36	0x2C	A	((A & 7F) + 0x36) ^ 0x2C	l
0x11	0x70	B	((B & 7F) + 0x11) ^ 0x70	l

```
0031485A   8B5D F8          mov ebx,dword ptr ss:[ebp-8]
0031485D   8AD0             mov dl,al
0031485F   80E2 7F          and dl,7F
00314862   0290 28E43200    add dl,byte ptr ds:[eax+32E428]
00314868   3290 38E43200    xor dl,byte ptr ds:[eax+32E438]
0031486E   889406 29E43200  mov byte ptr ds:[esi+eax+32E429],dl
00314875   8D50 01          lea edx,dword ptr ds:[eax+1]
00314878   80E2 7F          and dl,7F
0031487B   0290 29E43200    add dl,byte ptr ds:[eax+32E429]
00314881   3290 39E43200    xor dl,byte ptr ds:[eax+32E439]
00314887   889403 2AE43200  mov byte ptr ds:[ebx+eax+32E42A],dl
0031488E   8B5D F4          mov ebx,dword ptr ss:[ebp-C]
00314891   8D50 02          lea edx,dword ptr ds:[eax+2]

Hide FPU
EAX   00000006
EBX   FFE3142B
ECX   0015F854   "advapi"
EDX   00000006
EBP   0015F8BC
ESP   0015F854   "advapi"
ESI   FFE3142B
EDI   0015F864   "ConvertStrir

EIP   0031485F   b2bc74d95c8bc

EFLAGS   00000297
```

Figure 10 Decryption Loop

Extracted strings

Already active [%s]
+%u (%u) files done [%s] [%u KB/s]
Started (PID: %u; Workers: %u; AES-%s) [%s]
Complete (+%u (%u) files done) [%s]
Work time: %d:%02d:%02d
Unable to get computer name
CryptoGuard
kernel32.dll
ConvertStringSecurityDescriptorToSecurityDescriptorW
advapi32.dll
IsWow64Process
SystemDrive
KiUserExceptionDispatcher

3. It then gets a list of processes running using API - “CreateToolhelp32Snapshot”.
4. It then terminates the processes and services that may conflict with its execution. It excludes files and folder paths relevant to its execution or contains system drivers by comparing them with the hard-coded list of process names.

Processes that are terminated

javaw	infopath	MSSQL$PRACTICEMGT	SQLBrowser
java	exchange	MSSQL$PRACTTICEBGC	SQLSERVERAGENT
sage	excel	MSSQL$PROD	SQLSafeOLRService
ks_action	encsvc	MSSQL$PROFXENGAGEMENT	SQLTELEMETRY
ks_email	duplicati	MSSQL$SBSMONITORING	SQLTELEMETRY$ECWDB2
ks_copy	devenv	MSSQL$SHAREPOINT	SQLWriter
ks_sched	dbsnmp	MSSQL$SOPHOS	SQLsafe Backup Service
ks_serv	dbeng50	MSSQL$SQLEXPRESS	SQLsafe Filter Service
ks_web	database	MSSQL$SQL_2008	SamSs
ks_report	backup	MSSQL$SYSTEM_BGC	SepMasterService
ks_im	atom	MSSQL$TPS	ShMonitor
ks_db	arw	MSSQL$TPSAMA	SmcService
pvxiosvr	agntsvcencsvc	MSSQL$VEEAMSQL2008R2	Smcinst
pvxwin32	agntsvcagntsvc	MSSQL$VEEAMSQL2012	SntpService
xfssvccon	agntsvc	MSSQLFDLauncher	Sophos Agent
wordpad	ARSM	MSSQLFDLauncher$ PROFXENGAGEMENT	Sophos AutoUpdate Service
wlmail	AcrSch2Svc	MSSQLFDLauncher$ SBSMONITORING	Sophos Clean Service
winword	Acronis VSS Provider	MSSQLFDLauncher$ SHAREPOINT	Sophos Device Control Service
vmwp	AcronisAgent	MSSQLFDLauncher$ SQL_2008	Sophos File Scanner Service
vmware-vmx	AcronixAgent	MSSQLFDLauncher$ SYSTEM_BGC	Sophos Health Service
vmms	Antivirus	MSSQLFDLauncher$ TPS	Sophos MCS Agent
vmconnect	BackupExecAgent Accelerator	MSSQLFDLauncher$ TPSAMA	Sophos MCS Client
vmcompute	BackupExecAgent Browser	MSSQLSERVER	Sophos Message Router
visio	BackupExecDevice MediaService	MSSQLServerADHelper	Sophos Safestore Service
veeam	BackupExecJob Engine	MSSQLServerADHelper100	Sophos System Protection Service
tv_x64	BackupExec ManagementService	MSSQLServerOLAPService	Sophos Web Control Service
tv_w32	BackupExecRPC Service	McAfeeEngineService	SstpSvc
tomcat	BackupExecVSS Provider	McAfeeFramework	Symantec System Recovery
thunderbird	DCAgent	McAfeeFrameworkMcAfee Framework	TmCCSF

thebat64	DbxSvc	McShield	TrueKey
thebat	EPSecurityService	McTaskManager	TrueKeyScheduler
teamviewer	EPUpdateService	MongoDB	TrueKeyServiceHelper
tbirdconfig	ESHASRV	MsDtsServer	UI0Detect
tasklist	EhttpSrv	MsDtsServer100	Veeam Backup Catalog Data Service
taskmgr	Enterprise Client Service	MsDtsServer110	VeeamBackupSvc
synctime	EraserSvc11710	MySQL57	VeeamBrokerSvc
sublime_text	EsgShKernel	MySQL80	VeeamCatalogSvc
stream	FA_Scheduler	NetMsmqActivator	VeeamCloudSvc
steam	IISAdmin	OracleClientCache80	VeeamDeploySvc
sqbcoreservice	IMAP4Svc	OracleServiceXE	VeeamDeploymentService
screenconnect	KAVFS	OracleXETNSListener	VeeamEnterpriseManagerSvc
ruby	KAVFSGT	PDVFSService	VeeamHvIntegrationSvc
qbw32	MBAMService	POP3Svc	VeeamMountSvc
pythonw	MBEndpointAgent	RESvc	VeeamNFSSvc
python	MSExchangeAB	ReportServer	VeeamRESTSvc
processhacker	MSExchangeAD Topology	ReportServer$SQL_2008	VeeamTransportSvc
powerpnt	MSExchangeAntispam Update	ReportServer$SYSTEM_BGC	W3Svc
postgres	MSExchangeES	ReportServer$TPS	WRSVC
php	MSExchangeEdgeSync	ReportServer$TPSAMA	Zoolz 2 Service
outlook	MSExchangeFBA	SAVAdminService	bedbg
oracle	MSExchangeFDS	SAVService	ekrn
onenote	MSExchangeIS	SDRSVC	kavfsslp
om8start	MSExchangeMGMT	SMTPSvc	klnagent
om8	MSExchangeMTA	SNAC	macmnsvc
ocssd	MSExchangeMail Submission	SQL Backups	masvc
ocomm	MSExchangeMailbox Assistants	SQLAgent$BKUPEXEC	mfefire
ocautoupds	MSExchangeMailbox Replication	SQLAgent$CITRIX_METAFRAME	mfemms
notepad	MSExchangeProtected ServiceHost	SQLAgent$CXDB	mfevtp
notepad++	MSExchangeRPC	SQLAgent$ECWDB2	mozyprobackup
node	MSExchangeRepl	SQLAgent$PRACTTICEBGC	msftesql$PROD
nginx	MSExchangeSA	SQLAgent$PRACTTICEMGT	ntrtscan
ncsvc	MSExchangeSRS	SQLAgent$PROD	sacsvr
ncs	MSExchangeSearch	SQLAgent$PROFX ENGAGEMENT	sophossps

mydesktop service	MSExchangeService Host	SQLAgent$SBSMONITORING	svcGenericHost
mydesktopqos	MSExchangeThrottling	SQLAgent$SHAREPOINT	swi_filter
mspub	MSExchangeTransport	SQLAgent$SOPHOS	swi_service
msacces	MSExchangeTransport LogSearch	SQLAgent$SQLEXPRESS	swi_update
mongod	MSOLAP$SQL_2008	SQLAgent$SQL_2008	swi_update_64
metiix	MSOLAP$SYSTEM_BGC	SQLAgent$SYSTEM_BGC	tmlisten
mdccom	MSOLAP$TPS	SQLAgent$TPS	wbengine
mbarw	MSOLAP$TPSAMA	SQLAgent$TPSAMA	
mail	MSSQL$BKUPEXEC	SQLAgent$VEEAMSQL2008R2	
i_view32	MSSQL$ECWDB2	SQLAgent$VEEAMSQL2012	

Processes Excluded from termination

vmnat.exe	explorer.exe	wefault.exe	rundll32.exe	powershell.exe

Folders Excluded from encryption

Excludes encryption if found this string in file path

\windows\system32\	\appdata\local\	:\perflogs\	:\$recycle.bin\
\windows\syswow64\	\appdata\locallow\	:\programdata\	crypt_detect
\windows\system\	\all users\microsoft\	:\drivers\	cryptolocker
\windows\winsxs\	\inetpub\logs\	:\wsus\	ransomware
\appdata\roaming\	:\boot\	:\efstmpwp\	

```
26  int v24; // [esp+60h] [ebp-4h]
27
28  result = L"\\windows\\system32\\";
29  v3 = L"\\windows\\syswow64\\";
30  v4 = L"\\windows\\system\\";
31  v5 = L"\\windows\\winsxs\\";
32  v6 = L"\\appdata\\roaming\\";
33  v7 = L"\\appdata\\local\\";
34  v8 = L"\\appdata\\locallow\\";
35  v9 = L"\\all users\\microsoft\\";
36  v10 = L"\\inetpub\\logs\\";
37  v11 = L":\\boot\\";
38  v12 = L":\\perflogs\\";
39  v13 = L":\\programdata\\";
40  v14 = L":\\drivers\\";
41  v15 = L":\\wsus\\";
42  v16 = L":\\efstmpwp\\";
43  v17 = L":\\$recycle.bin\\";
44  v18 = L"crypt_detect";
45  v19 = L"cryptolocker";
46  v20 = L"ransomware";
47  v21 = &unk_2E83C0;
48  v22 = Buffer;
49  v23 = word_2E87D0;
50  v24 = 0;
51  v2 = 0;
52  while ( !*result || !StrStrIW(a1, result) )
53  {
```

Figure 11

Files Excluded from encryption

Some files are excluded from encryption

iconcache.db	thumbs.db	ransomware	ransom
debug.txt	boot.ini	desktop.ini	autorun.inf
ntuser.dat	ntldr	ntdetect.com	bootfont.bin
bootsect.bak	!TXDOT_READ_ME!.txt		

```
24  result = 0;
25  v20 = 0;
26  v6 = L"iconcache.db";
27  v7 = L"thumbs.db";
28  v8 = L" ransomware ";
29  v9 = L" ransom ";
30  v10 = L"debug.txt";
31  v11 = L"boot.ini";
32  v12 = L"desktop.ini";
33  v13 = L"autorun.inf";
34  v14 = L"ntuser.dat";
35  v15 = L"ntldr";
36  v16 = L"ntdetect.com";
37  v17 = L"bootfont.bin";
38  v18 = L"!TXDOT_READ_ME!.txt";
39  v19 = 0;
40  if ( a1 )
```

Figure 12

Excluded Extensions

Some necessary system files are excluded from encryption by comparing extensions.

.ani	.cab	.cpl	.cur	.diagcab	.diagpkg	.dll	.drv	.hlp	.icl	.icns	.ico	.iso	.ics	.lnk
.idx	.mod	.mpa	.msc	.nomedia	.msstyles	.msu	.msp	.ocx	.prf	.rtp	.scr	.shs	.spl	.sys
.exe	.bat	.cmd	.url	.theme	.themepack	.mui								

```
51  v4 = PathFindExtensionW(a1);
52  if ( !v4
53    || !*v4
54    || (v5 = StrStrIW(
55              L".ani|.cab|.cpl|.cur|.diagcab|.diagpkg|.dll|.drv|.hlp|.icl|.icns|.ico|.iso|.ics|.lnk|.idx|.mod|.mpa|.msc"
56               "|.msp|.msstyles|.msu|.nomedia|.ocx|.prf|.rtp|.scr|.shs|.spl|.sys|.theme|.themepack|.exe|.bat|.cmd|.url|.mui",
57              v4) == 0,
58        result = 1,
59        v5) )
```

Figure 13

Encryption methodology

5. It creates a separate **thread background** to do the encryption process.
6. Files are encrypted using BlockChain **AES** mode with a key of **256 bits** generated uniquely for each Victim.
7. The AES key is encrypted with Public **RSA-4096**, a 0x200 byte ciphertext hardcoded in the file.

```
    v5 = sub_F6F60(&v23, (int)&v38);
    if ( !v5 )
      ++v25;
    break;
  }
}
sub_103EB0((__m128i *)&v38, 0, 0x30u);
sub_103EB0((__m128i *)&v21, 0, 0x10u);
if ( !v5
  && !sub_F79B0(
        &v34,
        "BB45A5C97A75102E2C0030CB9A4851D89026721CA327A27A7E0645FD427586ACF46B43DCB7719F3CD6071D559FF3C1CF6C2CAADE33"
        "B930DD60DBFDBB9CE01C3F9C6DE8303F2A780029FD71E2DC792D6A791503A9545D719F408896DEFDB05E451476D5F0E890E14D56A1"
        "C2361AD44F8E8996EA7FF09501A56FF7F8D3F5742395FAC3C77680B957C5D2006986BF6BB83F1F5519C0A9C153F7A7BC61B2749121"
        "4E33B897A626F2FD3F792088D4C34722033B95ED10487EB54CDC87F14C357F04CAADBBE48C46082AA2FEB58DB4FF58728470FD4254"
        "A7F0F163961E22D20983501C7A316710129AB4567581931C43CC7016F7FD944557D10D05E767F6DD08A66088D0BDE7C6442222F026"
        "645201791DBFD462ABDDA8B2CB650E2D487386D544FBE4805664E5A3ECC2EC4A7FD9161566D9B83185AF27F4C935A1B17F0A607E6E"
        "33BC187DA9FD968E4DF5B1D80E644E2AFA607F03115A1A42D89A4A87CC1611A6526BC2BFBDEA8F9D7F97726DBD0858D0CC9735F3F3"
        "9A50DFD80BCDD3CA5CACA0828BF4374F5DA939B130D8042BF901A10BDB51E535164A601164D440A9F8E24F4929691CCC9992A3158F"
        "D96DCE3BD9F97EF7C6EE18D98BFAC3D5C7A5DC4A4CD10698DC0A2F58A4AFD779C014669A223CFC3AC0694EBD0C4C1EB2BE7589584C"
        "2B6497B1F0DF2BEBF286AE9462BDFB0FB9321C0C0B09694A1C1AEFE6DE61D849A7E80F")
  && !sub_F79B0(&v35, "010001") )
{
  v33 = (unsigned int)(sub_F7920() + 7) >> 3;
  if ( !v36 )
  {
    v7 = sub_F59A0(&v23, (int)&v32, (int)&v22, v43);
```

Figure 14 Hardcoded Public Key RSA- 4096

8. The AES key which is encrypted with RSA-4096 is added to the end of every encrypted file.

```
29  StrCpyW((LPWSTR)&v11, psz2);
30  StrCatW((LPWSTR)&v11, L".txd0t");
31  if ( PathFileExistsW((LPCWSTR)&v11) )
32    return v3;
33  SetFileAttributesW(psz2, 0x80u);
34  v6 = CreateFileW(psz2, 0xC0000000, 0, 0, 3u, 0x8000080u, 0);
35  if ( v6 != (HANDLE)-1 )
36  {
37    v15 = 0i64;
38    GetFileSizeEx(v6, &v15);
39    if ( v15 && v15 >= 16 )
40    {
41      sub_413EB0(v13, 0, 0x118u);
42      EnterCriticalSection(&stru_425F70);
43      qmemcpy(&Buffer, &unk_425F88, 0x200u);
44      sub_4060E0(v13, (int)&unk_425F50);
45      LeaveCriticalSection(&stru_425F70);
46      if ( sub_404E40(2, (int)v6, 0, 0) )
47      {
48        NumberOfBytesWritten = 0;
49        if ( WriteFile(v6, &Buffer, 0x200u, &NumberOfBytesWritten, 0) )
50        {
51          if ( NumberOfBytesWritten == 512 )
52          {
```

Figure 15 Command to write to end of Encrypted every file

9. Each victim is targeted separately and unique RSA public key is used.

Post Encryption

10. Uses API - "WNetOpenEnumW" and "WNetEnumResourceW" to search for file shares that may contain files that could be encrypted.
11. Ransomware adds customized extensions to the end of the file.
12. Ransom notes are created in every folder – "**!TXDOT_READ_ME!.txt.**"

```
`
  FlushFileBuffers(v9);
  NumberOfBytesWritten = 0;
  if ( WriteFile(
         v9,
         L"Greetings, Texas Department of Transportation!\r\n"
          "\r\n"
          "Read this message CAREFULLY and contact someone from IT department.\r\n"
          "Your files are securely ENCRYPTED.\r\n"
          "No third party decryption software EXISTS.\r\n"
          "MODIFICATION or RENAMING encrypted files may cause decryption failure.\r\n"
          "\r\n"
          "You can send us an encrypted file (not greater than 400KB) and we will decrypt it FOR FREE,\r\n"
          "so you have no doubts in possibility to restore all files from all affected systems ANY TIME.\r"
          "\n"
          "Encrypted file SHOULD NOT contain sensitive information (technical, backups, databases, large "
          "documents).\r\n"
          "The rest of data will be available after the PAYMENT.\r\n"
          "Infrastructure rebuild will cost you MUCH more.\r\n"
          "\r\n"
          "Contact us ONLY if you officially represent the whole affected network.\r\n"
          "The ONLY attachments we accept are non archived encrypted files for test decryption.\r\n"
          "Speak ENGLISH when contacting us.\r\n"
          "\r\n"
          "Mail us: txdot911@protonmail.com\r\n"
          "We kindly ask you not to use GMAIL, YAHOO or LIVE to contact us.\r\n"
          "The PRICE depends on how quickly you do it.\r\n",
         0x7F0u,
         &NumberOfBytesWritten,
         0) )
  {
    if ( NumberOfBytesWritten == 2032 )
      FlushFileBuffers(v9);
  }
```

Figure 16

13. Executes a series of Anti-forensic measures that remove backup and restore options.
14. It displays the time taken to encrypt and the files encrypted. It closes the console.

Console Output

The console output gives the number of Workers, which represents the number of processors present and the time taken for the encryption process to complete after the entire process is done.

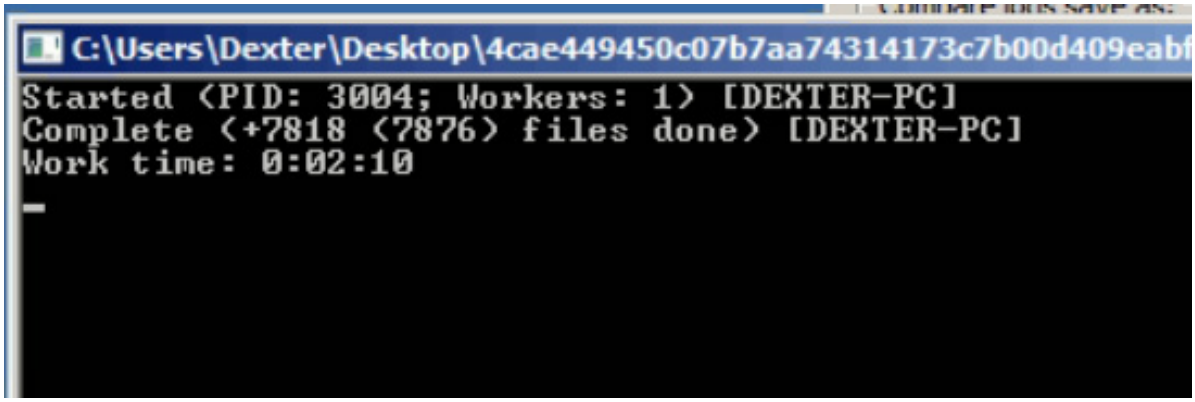


Figure 17 Console Output

Special Features

1. **Multi-Threading** - Uses Multi-threading to improve performance
2. **Process Priority Boost** - Uses API - "SetProcessPriorityBoost" to boost up the processing speed

```
  }
  while ( v1 < 36 );
}
v19 = *(_DWORD *)dword_4283BC;
v18 = v23;
v4 = GetCurrentProcessId();
sub_401A20((LPCSTR)&v20, v4, v18, v19);
v5 = GetCurrentProcess();
SetProcessPriorityBoost(v5, 0);
ThreadId = 0;
for ( i = CreateThread(0, 0, (LPTHREAD_START_ROUTINE)StartAddress, 0, 0, &ThreadId);
      !i;
      i = CreateThread(0, 0, (LPTHREAD_START_ROUTINE)StartAddress, 0, 0, &ThreadId) )
{
  Sleep(0x3E8u);
}
CloseHandle(i);
v8 = alloca(64);
if ( &v17 )
```

Figure 18 ProcessPriorityBoost

3. **Thread Priority Boost** - Uses API – "ThreadAffinityMask" and "SetThreadPriorityBoost" for dynamic priority boostSetThreadAffinityMask is set to 0
   SetThreadPriorityBoost is assigned to 0
   Threads work by Round Robin fashion without any priority
   The system does not boost the priority of threads with a base priority level between 16 and 31. Only threads with a base priority between 0 and 15 receive dynamic priority boosts.

```
  v11 = CreateEventA(0, 0, 1, 0);
  lpHandles[v0] = v11;
  if ( !v11 )
    break;
  v12 = CreateEventA(0, 0, 0, 0);
  *(_DWORD *)(dword_425F4C + 4 * v0) = v12;
  if ( !v12 )
    break;
  v13 = GetProcessHeap();
  v14 = v3(v13, 8u, 0x7D80u);
  *(_DWORD *)(dword_425F30 + 4 * v0) = v14;
  if ( !v14 )
    break;
  v15 = sub_402230() + 128;
  v16 = GetProcessHeap();
  v17 = v3(v16, 8u, v15);
  *(_DWORD *)(dword_425F28 + 4 * v0) = v17;
  if ( !v17 )
    break;
  v19 = (void *)sub_404C10(v18, 0);
  if ( v19 )
  {
    v20 = GetCurrentThread();
    SetThreadAffinityMask(v20, v0)
    SetThreadPriorityBoost(v19, 0)
    CloseHandle(v19);
  }
  if ( (signed int)++v0 >= (signed int)nCount )
    return 1;
  v3 = HeapAlloc;
}
return 0;
```

Figure 19 Allocation of Dynamic Priority

4. Additional Encryption Methods – Checks for the availability of SSE2 instructions

```
012656FC
012656FC
012656FC
012656FC sub_12656FC proc near
012656FC push    0Ah             ; ProcessorFeature
012656FE call    ds:IsProcessorFeaturePresent
01265704 mov     dword_1279DF4, eax
01265709 xor     eax, eax
0126570B retn
0126570B sub_12656FC endp
0126570B
```

Figure 20

PF_XMMI_INSTRUCTIONS_AVAILABLE 6	The SSE instruction set is available.
PF_XMMI64_INSTRUCTIONS_AVAILABLE 10	The SSE2 instruction set is available. **Windows 2000:** This feature is not supported.
PF_XSAVE_ENABLED 17	The processor implements the XSAVE and XRSTOR instructions. **Windows Server 2008, Windows Vista, Windows Server 2003 and Windows XP/2000:** This feature is not supported until Windows 7 and Windows Server 2008 R2.
PF_ARM_V8_INSTRUCTIONS_AVAILABLE 29	This ARM processor implements the the ARM v8 instructions set.

Figure 21

SSE2 (Streaming SIMD Extensions 2) is one of the Intel SIMD (Single Instruction, Multiple Data) processor supplementary instruction sets
SIMD instructions can greatly increase performance

Anti-forensic measures

The Authors have implemented a few Anti-Forensic measures to ensure not retrieving the data from machines post encryption, including deleting Event logs to overwrite space on the C drive.

```
dwTyp = 0;
if ( sub 4049:() )
  Wow64DisableWow64FsRedirec(&dwTyp);
v14 = L"bcdedit.ex;
v16 = L"bcdedit.ex;
v2 = L"wbadmin.ex;
v12 = L"wbadmin.ex;
v13 = L"delete catalog -qui;
v15 = L"/set {default} recoveryenable;
v17 = L"/set {default} bootstatuspolicy ignoreallfa;
v18 = L"schtasks.ex;
v19 = L"/Change /TN \"\\Microsoft\\Windows\\SystemRestore\\SR\" ;
v20 = L"wevtutil.ex;
v21 = L"cl Applicatic;
v22 = L"wevtutil.ex;
v23 = L"cl Syster;
v24 = L"wevtutil.ex;
v25 = L"cl Setup;
v26 = L"wevtutil.ex;
v27 = L"cl Securit;
v28 = L"wevtutil.ex;
v29 = L"sl Security /e:fal;
v30 = L"fsutil.ex;
v31 = L"usn deletejournal /D ;
v32 = 0;
v33 = 0;
v3 = 0;
do
```

Figure 22.1

```
while ( v2 );
v5 = "SOFTWARE\\Policies\\Microsoft\\Windows NT\\System;
ExecInf.fMas = (ULONG)"DisableConfi;
ExecInf.lpParamete = (LPCWST)"DisableConfi;
ExecInf.lpFil = (LPCWST)"SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Syste;
ExecInf.lpDirecto = (LPCWST)"SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Syste;
v6 = 0;
ExecInf.cbSiz = (DWORD)"SOFTWARE\\Policies\\Microsoft\\Windows NT\\System;
ExecInf.hwn = (HWND)"SOFTWARE\\Policies\\Microsoft\\Windows NT\\System;
ExecInf.lpVer = (LPCWST)"DisableSf;
ExecInf.nSho = (int)"DisableSf;
ExecInf.hInstAp = 0;
ExecInf.lpIDLis = 0;
v7 = 0;
do
```

Figure 22.2

The malware uses Living off the land binaries which is present inside "C:\Windows\System32\" folder to do the operations

Commands	Actions
"wbadmin.exe" delete catalog -quiet	Backup Catalogue are deleted
"bcdedit.exe" /set {default} bootstatuspolicy ignoreallfailures "bcdedit.exe" /set {default} recoveryenabled no	Disable Recovery Mode
"schtasks.exe" /Change /TN "\Microsoft\Windows\SystemRestore\SR" /disable	disable system restore using scheduled task
"wevtutil.exe" cl Setup "wevtutil.exe" cl System "wevtutil.exe" cl Application "wevtutil.exe" cl Security "wevtutil.exe" sl Security /e:false	Clear Event logs and disable Security Event Logs
"fsutil.exe" "usn delete journal /D C: "	delete the Update Sequence Number journal

Also creates a thread to overwrite deleted files and unused space on C Drive

"C:\Windows\System32\cipher.exe" /w %s

Registry entry

The registry entries are made to disable System restore

Key	Name	Value
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\NT\CurrentVersion\SystemRestore	DisableConfig	1
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\NT\CurrentVersion\SystemRestore	DisableSR	1
HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows NT\SystemRestore	DisableConfig	1
HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows NT\SystemRestore	DisableSR	1

Ransom Note

Finally, a Ransom note is displayed to the user

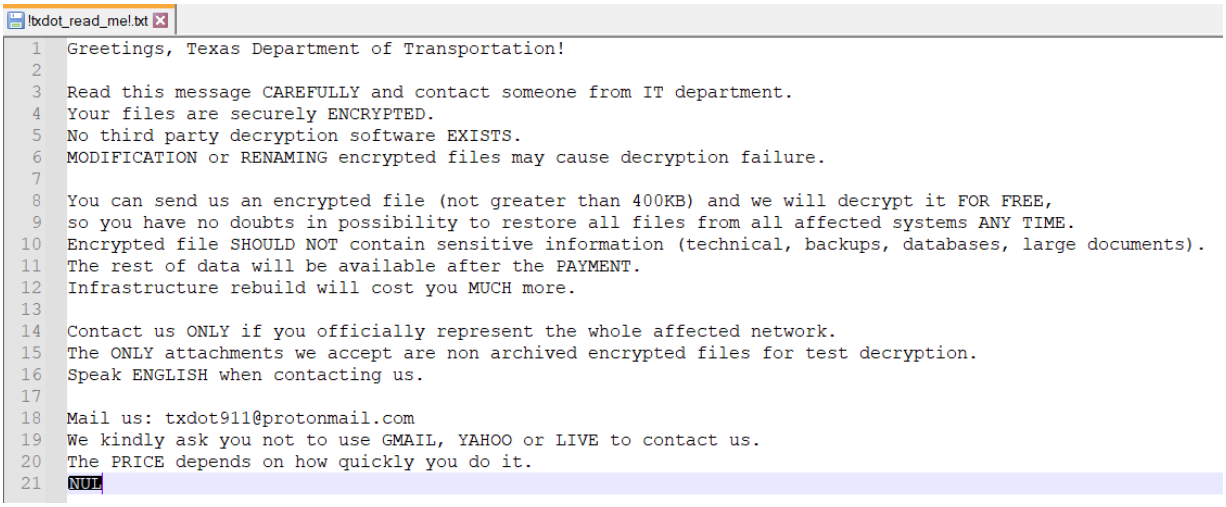


Figure 23

The Encryption note for CNT of Ecuador happened in July 2021 is given below

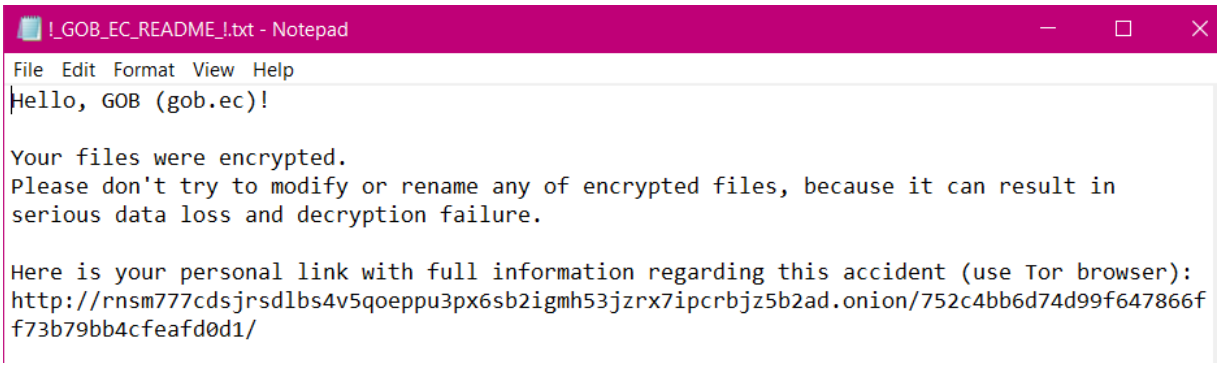


Figure 24

Older Variant – Defray777

In the earlier versions of Defray777 in 2017, they started by leveraging software vulnerabilities of Microsoft documents delivered via Phishing emails to targeted people. These drop backdoor malware Vatet loaders are leading to the execution of Defray777 by using cobalt strike. Via these attacks, they targeted only the UK and US.
It also adds specific 32-byte strings to the end of each file as a marker to indicate the file is encrypted and connects to C&C to report infection information. It also uses VSS admin to delete volume shadow copy which is not used in newer versions

```
Don't panic, read this and contact someone from IT department.
Your computer has been infected with a virus known as ransomware.
All files including your personal or business documents, backups and projects are encrypted.
Encryption is very sophisticated and without paying a ransom you won't get your files back.
You could be advised not to pay, but you should anyway get in touch with us.
Ransom value for your files is 5000$ to be paid in digital currency called Bitcoin.
If you have questions, write us.
If you have doubts, write us.
If you want to negotiate, write us.
If you want to make sure we can get your files back, write us.

glushkov@protonmail.ch
glushkov@tutanota.de
igor.glushkov.83@mail.ru

In case we don't respond to an email within one day, download application called BitMessage and reach to us for the fastest response.
BitMessage BM-2cVPKqFb5ZRaMUYdryqxsMNxFMud1bvnY6

################################################################################

 To someone from IT department

 This is custom developed ransomware, decrypter won't be made by an antivirus
 company. This one doesn't even have a name. It uses AES-256 for encrypting
 files, RSA-2048 for storing encrypted AES-256 password and SHA-2
 for keeping the encrypted file integrity. It's written in C++ and have passed
 many quality assurance tests. To prevent this next time use offline backups.

################################################################################
```

Figure 25 Defray777 Ransom Note

Linux Variant

The Linux variant has existed since July 2020 and uses the same logic for encryption as Windows, encrypts file extension that is only explicitly mentioned.

The code for the Linux version of NASDAQ stock exchange targeted malware shows that the author did not even try to hide the function names for the Malware.

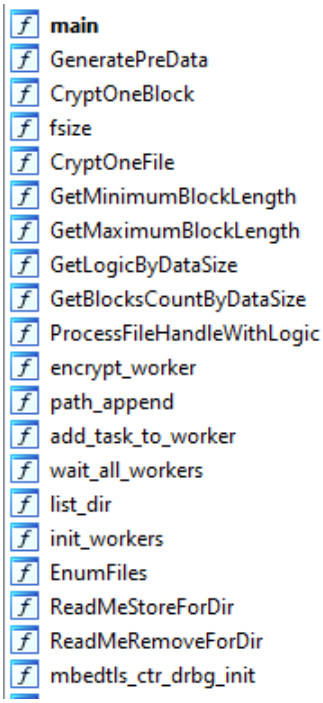


Figure 26

Uses ELF executable port of the 777 ransomware to encrypt Linux and other Unix-like systems

```
strcat(dest, "!R1_RCM_README!.txt");
if ( (unsigned int)stat64(dest, &v2) == -1 )
{
  stream = fopen64(dest, "w");
  if ( stream )
  {
    fwrite(
      "Hello R1 RCM (NASDAQ: RCM)!!!\r\n"
      "\r\n"
      "Inspect this message CLOSELY and contact someone from technical division.\r\n"
      "Your data is securely ENCRYPTED.\r\n"
      "CORRECTION names or content of encrypted items (*.r1rcm911) can make recovering problems.\r\n"
      "\r\n"
      "Mail us any encrypted document (smaller than 800KB) and we would restore it.\r\n"
      "Affected file SHOULD NOT have sensitive intelligence.\r\n"
      "The rest of data will be available behind PAYING.\r\n"
      "\r\n"
      "We ask you not to contact cops as they will BLOCK your bank accounts to inhibit payment.\r\n"
      "Reach us BUT if you responsible for all business.\r\n"
      "\r\n"
      "r1rcm911@protonmail.com",
      1uLL,
      0x24BuLL,
      stream);
```

Figure 27

Conclusion

Cybercriminals improve their approaches constantly to exfiltrate data while moving laterally in the infrastructure. RansomExx is an apt example of how a ransomware family can evolve in finding new ways of infecting machines.

The new variant compared to Defray777 shows that the threat actors are improvising with sophisticated modern methods to evade detection by running file-less malware and using advanced intrusion techniques combined with anti-forensic measures.

Using the double extortion method makes the threat group even more dangerous. The expansion into other additional platforms must be seen as a greater danger to companies. When domain control is compromised, it can be used to deploy Windows and Linux builds to all devices connected to the network, leading to a devastating disaster.

As users, we can protect ourselves from these threats by enabling the latest security features with regular offline and remote backups and keeping the OS updated with the latest patch.

Quick Heal Detection details

Quick Heal and Seqrite protects these kinds of ransomware in multiple stages, including URL filtering, web protection, Anti Malware protection, Behaviour, Cloud, and Anti Ransomware protection.

Ransom.RansomExx.S24986107
Ransom.RansomExx.S24986107
Trojan.MauvaiseRI.S5250997
Trojan.MauvaiseRI.S5250997
Trojan.Vatet
Trojan.VatetIH.S24673002
Trojan.VatetIH.S24673003
Trojan.VatetRI.S24672989
Trojan.VatetRI.S24672990
Trojan.VatetRI.S24672991
Trojan.VatetRI.S24672992
Trojan.VatetRI.S24672993
Trojan.VatetRI.S24672994

Trojan.VatetRI.S24672997
Trojan.VatetRI.S24672998
Trojan.VatetRI.S24673000
ELF.Trojan.45068.GC
Elf.Trojan.A1190002
ELF.Trojan.39879.GC
Trojan.Ghanarava.162944294493b342
Trojan.Ghanarava.16259071563d35a7
Trojan.Ghanarava.16259014162b3696
Trojan.Ghanarava.1629442944e4e97d
Trojan.Ghanarava.162582110332ee9b
Trojan.Ghanarava.16294429441aace4
Trojan.Ghanarava.16294429445f202d

Quick Heal also has additional security to block these kinds of Ransomware attacks in Behavioural and Anti-Ransomware Protection.

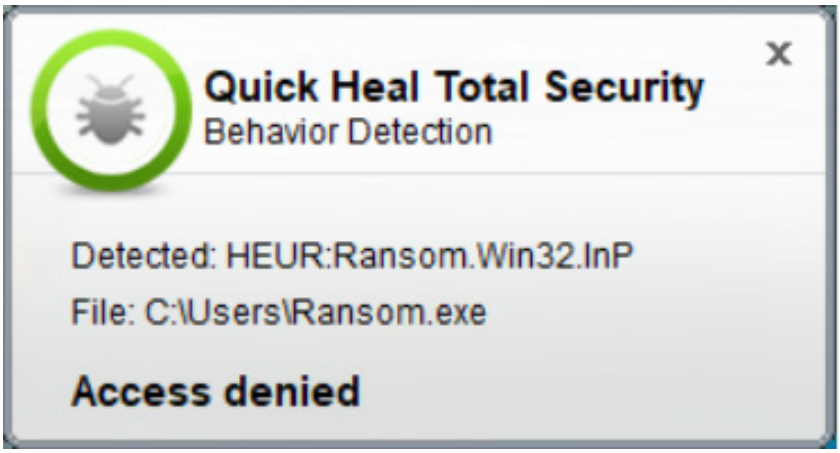


Figure 28

Mitre Attack Framework

T1189	Drive-by Compromise
T1620	Reflective Code Loading
T1057	Process Discovery
T1055	Process Injection
T1574	Hijack Execution Flow: DLL Side-Loading
T1083	File and Directory Discovery
T1140	Deobfuscate/Decode Files or Information
T1485	Data Destruction
T1486	Data Encrypted for Impact
T1490	Inhibit System Recovery
T1120	Peripheral Device Discovery
T1135	Network Share Discovery
T1489	Service Stop
T1059	Command and Scripting Interpreter: Windows Command Shell
T1047	Windows Management Instrumentation

Indicator of Compromise

E1E8725F45C5E42C7899B825739234CD
FCD21C6FCA3B9378961AA1865BEE7ECB
D13F890034A68CCB4AF4E0BF51E2B5EC
9D0BE1D1B94D984EEAA4433FA13B2C7C
4BB2F87100FCA40BFBB102E48EF43E65
F7C4CB42780B03303CA4B8535BB27207
AA1DDF0C8312349BE614FF43E80A262F
210F47C8F47DED8525DA927710ABC6AD

Vatet Loader

001DB136683CE2ACF62CE8F3D6D5B4C8
039E75CDD8787394789D11CA6D2C7711
088D29B4A238A650E12F5CE97EC58289
164B162F8CD59ACF9D3DA0BEC7EA1C52
23594AD0BA8EC37AD5EAEC84AEE9CECD
23DAE47577CDA08DFC82E65E1217CBEE
31DC5267D3DAF057BAAA37F8D5D59229
3EECB3D41523E5C29E8ACE24DB7931C6
497AD5FACD7764702CEC9A221D299572
6363CBA1430BF8A617D789B49E275975
643FBCDA0041C2B57A2740BB02E16DB0
6932DFCD3789F88E828D939174183446
808C956808D1A47B50F51DF08D45F391
94B27B9DE692308CDB07AA6CC31391F1
A40F5C5438F7DA071B0DF586B7329438
B90FBB7AE572ECA2F64D14C0E0DC4A21
D81995FDC06F5FC5268F78E1F1A7EBBB
DE2F1524EAD077C3BB0AA592BEF99E0C
FE180737BFB5436A592581DE52ED9368
3AC2C1BA866675A1E8D111F8435CB4B2
3ECB971E7FA1EE6357DDA4CE8384ACBC
64627BA9B3737C530C4571EA819D9E1B
84A336D3629FB9BB1DCEB41523057778

9F9A4BE7B29DA64E215321BC1DC626BF
A8944382DA44F326B14C6E3304DFF7B0
B5A22F43252B89DB6EAB109C6CCF9962
E4A15BF88200EEBD417912F9DCFB9A16
2DDB52A96CE4D6121A53B6F7FFB6EA3C
369B086DF0B72B46070D010C360F86EE
3ECB971E7FA1EE6357DDA4CE8384ACBC
64627BA9B3737C530C4571EA819D9E1B
84A336D3629FB9BB1DCEB41523057778
9F9A4BE7B29DA64E215321BC1DC626BF
A8944382DA44F326B14C6E3304DFF7B0
B5A22F43252B89DB6EAB109C6CCF9962
E4A15BF88200EEBD417912F9DCFB9A16
1D191D54CDD3ADB4621B5C3A13D1EA91
26E4A7443332461D330E6DC4E9A22F5B
2F6340654F5D07C7A5D19B9D228DABB1
4B3064C24CB16361027233138FD539DC
4EF817562DC042E616AE26A2C8773F23
81BA4107943BB4AD2EC351BA2417F987
DDF9E95123D9B585FA9E164236BFD338
FC2FEFB951BFBFDB1E337C9019968C8D
9F95BAD2FFDF61D0587CBA710BA0F2F4
9F95BAD2FFDF61D0587CBA710BA0F2F4

38F9CB4BC4F1F92186BB63A7E995C648
988B54D62C2163CDB5398FF6571E3C80
9935435529057201DAC86957275A43E9
13CC74A4168AAB6C63B5E44358F47604
AA0BF0045C4FAA988815117CEBCACDEB
B5D6214C223B3F6BC4A77C47E0E2A864
E5B622B9864D3A2E31A4EDAC46C1CB0C
4D1B52E30629477A12DCF2BBBC196E88
3EFD8ADA55DF9D41D439E10E3F02CFEA
6F6A04E60AF90862B2CED5864B6B23F9
C7E84D5C86F51A349445AD126C42FD89
68CB520D2084020638790187E34638EA
05D24DD80B9A39E2148E94C742F8F16B
B18EE982DE606ADC6715E7A52648B63C
E0D2C9AAC9A8489A2154AFF6E0ABCB6E
0EA9B7A283E7D4601FB7DBD63493B342
1F937CBAE354345087860C7D33E0E61D
2133B1C7BB6145CDD121EB8C423D35A7

225747A368357A5EAFAAC5337EE56C9A
25E8D46D27E0A1034804ABA00BA75D38
41EFF4CD049A8B5DEBF437B229E7C044
4BEE85530D15BE0A9E6C8672E355DDC6
4F2C11EE45CE87EEEE7789B43CC91AC3
615292E183CF11759B672148998BFA18
7031A1138E1892FB09BFBDF518DBA07B
77E9031A6BA4AFEECDA915E914A352DF
8041965231306E1C2DFF3695D6327524
976D4CB0F4DE0B02AD46DE2C862B3696
9D4C4AF4B600BB90E92A5C0B86551507
AE07F0B180BC52B39000F50353E4E97D
CA4682A32CDAAF2C0357A2A79E32EE9B
DBA03B64B963B77FE966238C261AACE4
DCBA8D6CF6B336AC96DB500AD99B0013
E2B15234DEE641B74EE7959DF2AE2E43
E843170E564321228FC88B9291A4265C
EB885E485049EE4516BBDF6D9C5F202D

Vatet Payload

D0D18D3C0F6286FACC308378A25832AD
049DDD7117E62F321359679E0675FB70
1D184002E50BF5A5BF1D0F6F745ED549
00EBDE844D52ED6477340FF46C7C9AA7
05667FF47E0396428BA7D7B9D0F012DE
14A9EF86992FE16560FE7D96204513DB
12B57C866C75FC5235E841682075D880
BFA080919A4868D5CE8398E171E2DDD7
234F4C627BA32E9B983385FB6524F3D0
BAF05F022919871E150411F1B5CCD7A4
1B80363C4DE80D4AACFF86AE67A867BC
32FA9177C2DCA009F31D9968BE74D21B

Appendix

[1]
https://raw.githubusercontent.com/Doneone/happy_cs/0788279a219756f10bbb03e9e7e960d3ce5bbaeb/lib.py

Quick Heal

Security Simplified

Quick Heal Technologies Ltd.
Marvel Edge, Office No. 7010 C & D, 7th Floor, Viman Nagar, Pune,
Maharashtra, India - 411014.
Phone: 1800 212 7377 | info@quickheal.co.in | www.quickheal.com