Investigating BlackSuit Ransomware’s Similarities to Royal

trendmicro.com/en_us/research/23/e/investigating-blacksuit-ransomwares-similarities-to-royal.html

May 31, 2023

Ransomware

In this blog entry, we analyze BlackSuit ransomware and how it compares to Royal Ransomware.

By: Katherine Casona, Ivan Nicole Chavez, Ieriz Nicolle Gonzalez, Jeffrey Francis Bonaobra May 31, 2023 Read time: (words)

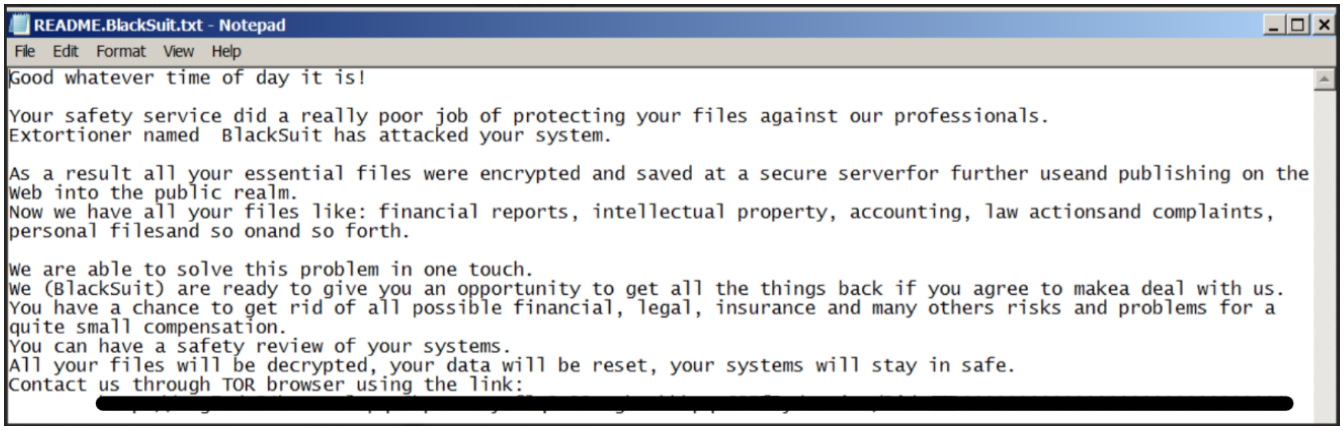
Royal ransomware, which is already one of the most notable ransomware families of 2022, has gained additional notoriety in early May 2023 after it was used to attack IT systems in Dallas, Texas. Around the same period, several researchers on Twitter came across a new ransomware family called BlackSuit that targeted both Windows and Linux users. Additional Twitter posts mentioned connections between BlackSuit and Royal, which piqued our interest. We managed to retrieve and analyze a Windows 32-bit sample of the ransomware from Twitter.

In this blog entry, we analyze BlackSuit ransomware and how it compares to Royal Ransomware.

Encryption and leak site details

Before delving into the main comparison between the two ransomware families, let’s first examine the encryption and leak site details of BlackSuit.

BlackSuit appends the *blacksuit* file extension to the files it encrypts, drops its ransom note into the directory, and lists its TOR chat site in the ransom note along with a unique ID for each of its victims.



```
Good whatever time of day it is!

Your safety service did a really poor job of protecting your files against our professionals.
Extortioner named  BlackSuit has attacked your system.

As a result all your essential files were encrypted and saved at a secure serverfor further useand publishing on the
Web into the public realm.
Now we have all your files like: financial reports, intellectual property, accounting, law actionsand complaints,
personal filesand so onand so forth.

We are able to solve this problem in one touch.
We (BlackSuit) are ready to give you an opportunity to get all the things back if you agree to makea deal with us.
You have a chance to get rid of all possible financial, legal, insurance and many others risks and problems for a
quite small compensation.
You can have a safety review of your systems.
All your files will be decrypted, your data will be reset, your systems will stay in safe.
Contact us through TOR browser using the link:
[redacted]
```

Figure 1. The BlackSuit ransom note

Its operators also set up a data leak site as part of their two-pronged extortion strategy to coerce victims into paying the ransom demand. Note that there is just a single victim currently listed on the leak site as of the time of writing.

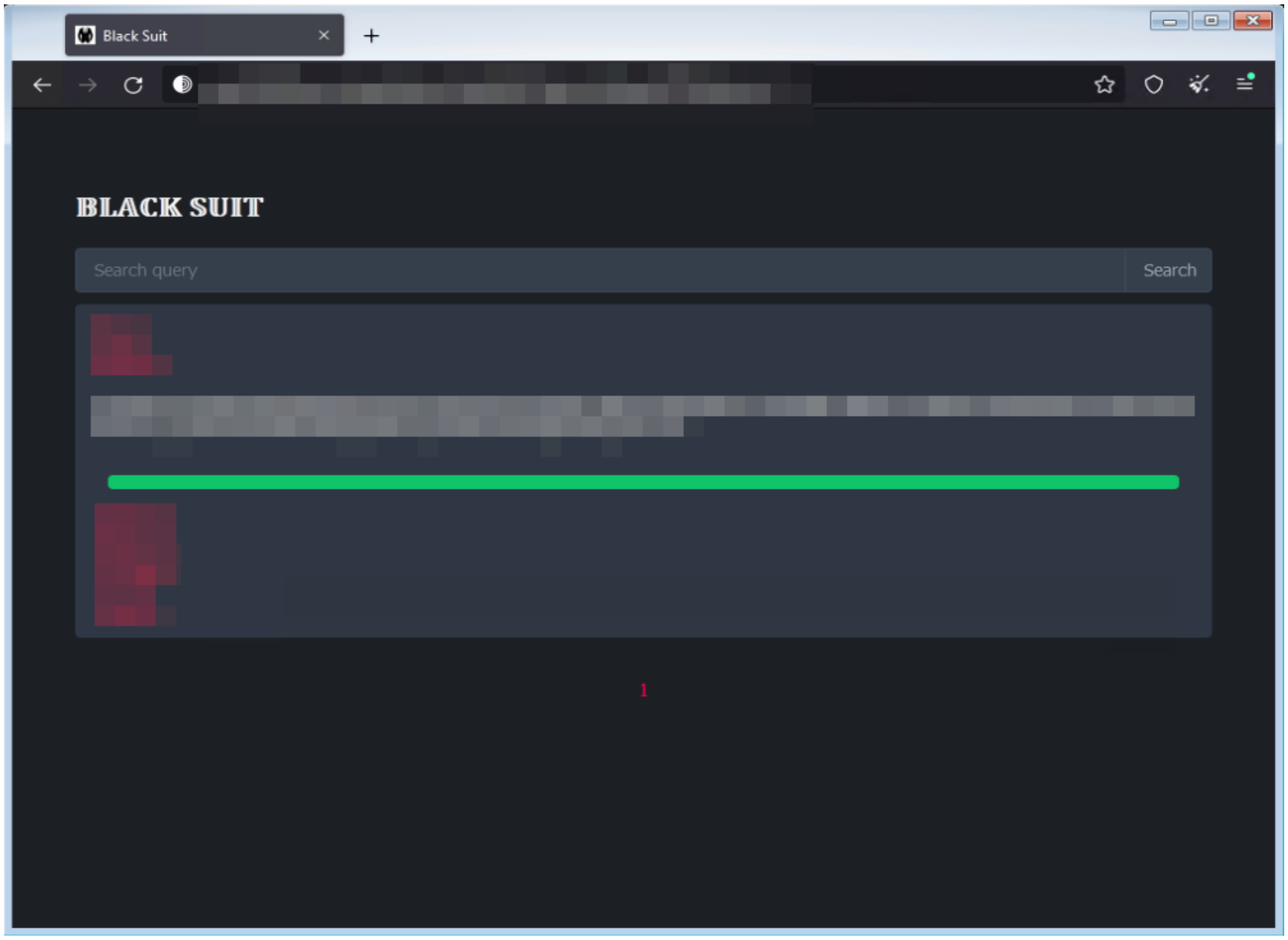
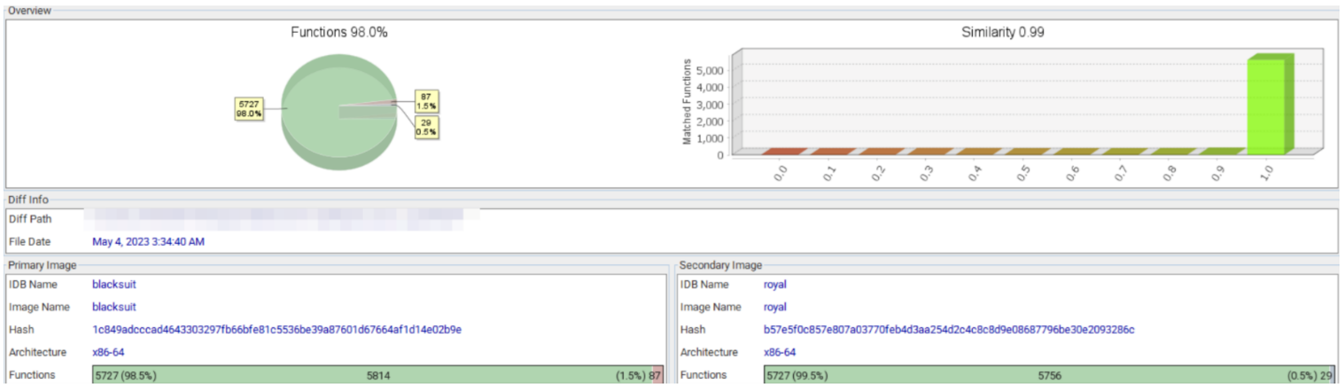


Figure 2. The BlackSuit TOR website

Comparison between Royal ESXi and BlackSuit ESXi variants

One of the BlackSuit ransomware samples we analyzed is an x64 ESXi version targeting Linux machines. An earlier post on Twitter revealed that YARA rules designed for BlackSuit's Linux variant matched samples of the Royal ransomware Linux variant.

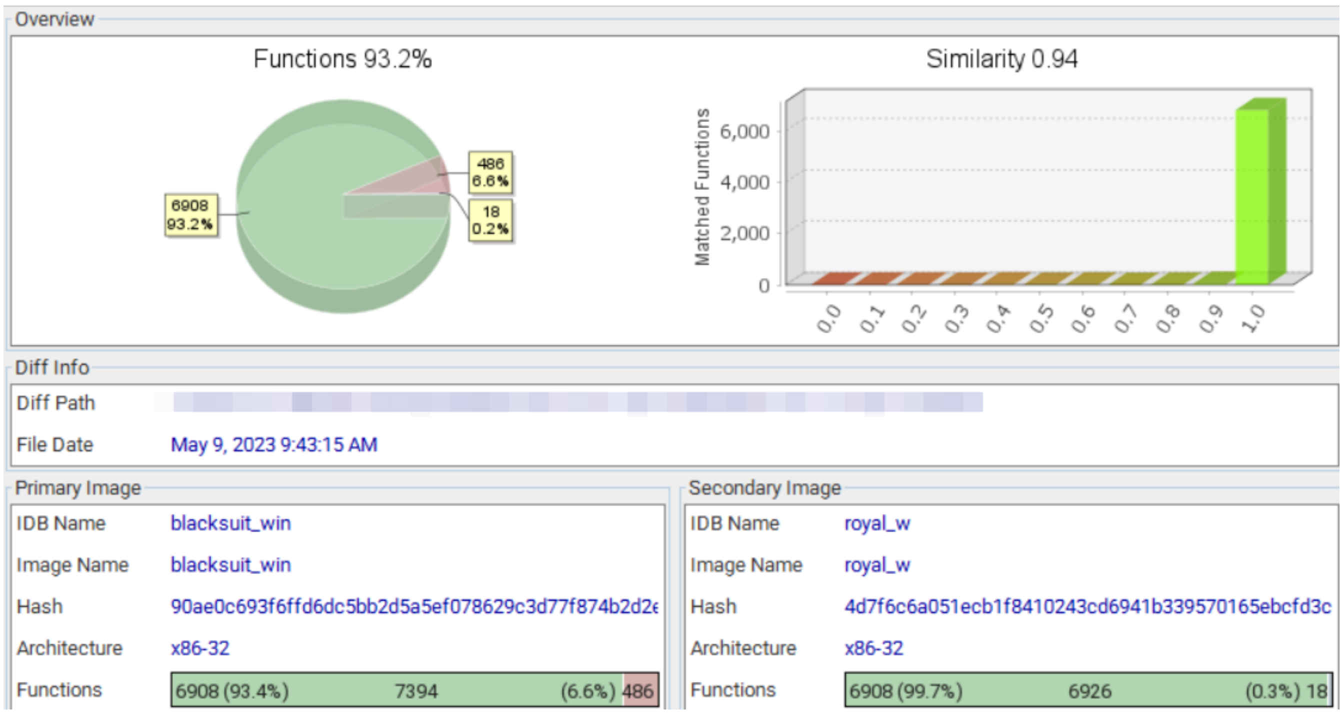
After comparing both samples of the Royal and BlackSuit ransomware, it became apparent to us that they have an extremely high degree of similarity to each other. In fact, they're nearly identical, with 98% similarities in functions, 99.5% similarities in blocks, and 98.9% similarities in jumps based on BinDiff, a comparison tool for binary files.



Diff Info

Diff Path	[redacted]
File Date	May 4, 2023 3:34:40 AM

Primary Image		Secondary Image	
IDB Name	blacksuit	IDB Name	royal
Image Name	blacksuit	Image Name	royal
Hash	1c849adcccad4643303297fb66bfe81c5536be39a87601d67664af1d14e02b9e	Hash	b57e5f0c857e807a03770feb4d3aa254d2c4c8c8d9e08687796be30e2093286c
Architecture	x86-64	Architecture	x86-64
Functions	5727 (98.5%) 5814 (1.5%) 87	Functions	5727 (99.5%) 5756 (0.5%) 29



Diff Info

Diff Path	[redacted]
File Date	May 9, 2023 9:43:15 AM

Primary Image

IDB Name	blacksuit_win
Image Name	blacksuit_win
Hash	90ae0c693f6ffd6dc5bb2d5a5ef078629c3d77f874b2d2
Architecture	x86-32
Functions	6908 (93.4%) 7394 (6.6%) 486

Secondary Image

IDB Name	royal_w
Image Name	royal_w
Hash	4d7f6c6a051ecb1f8410243cd6941b339570165ebcfd3c
Architecture	x86-32
Functions	6908 (99.7%) 6926 (0.3%) 18

Figure 3. Comparison of the Linux variants of BlackSuit and Royal ransomware

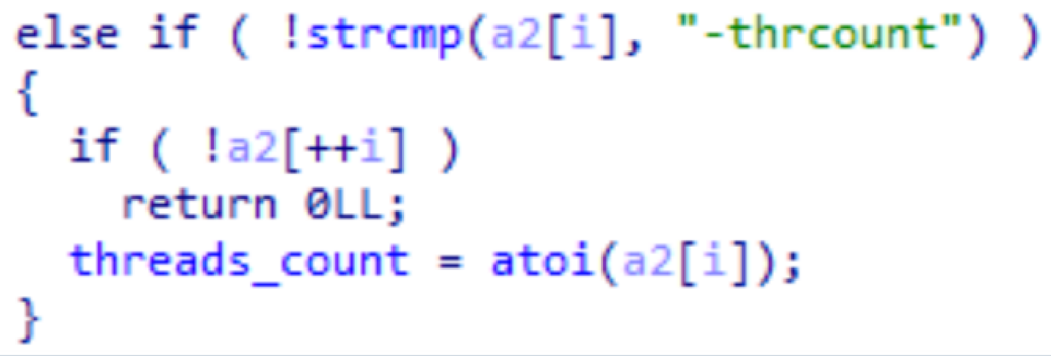
Further analysis found that BlackSuit employs command-line arguments that have a similar function to those used by Royal. However, there are some differences: The strings used in the arguments are different, with BlackSuit also including additional arguments not found in Royal.

Royal Argument	BlackSuit Argument	Description
-id {32-byte characters}	-name {32-byte characters}	Used as the victim's ID, which will be appended to the TOR link found in the dropped ransom note. The process exits if the argument is not provided, or if the provided characters do not have a length of 32 bytes
-ep	-percent {0-100}	Used to define the encryption parameter
-path {target path}	-p {target path}	Used to specify a target directory to encrypt
(Not in Royal)	-thrcount	Used to create a specified number of threads depending on infected machine's processor count
(Not in Royal)	-skip {text file}	Used to specify a text file containing folders to skip
-stopvm	-killvm	Used to terminate VM-linked processes via the EXSCLi command
-vmonly	-allfiles	Encrypt all files
(Not in Royal)	-noprotect	CheckProcStarted

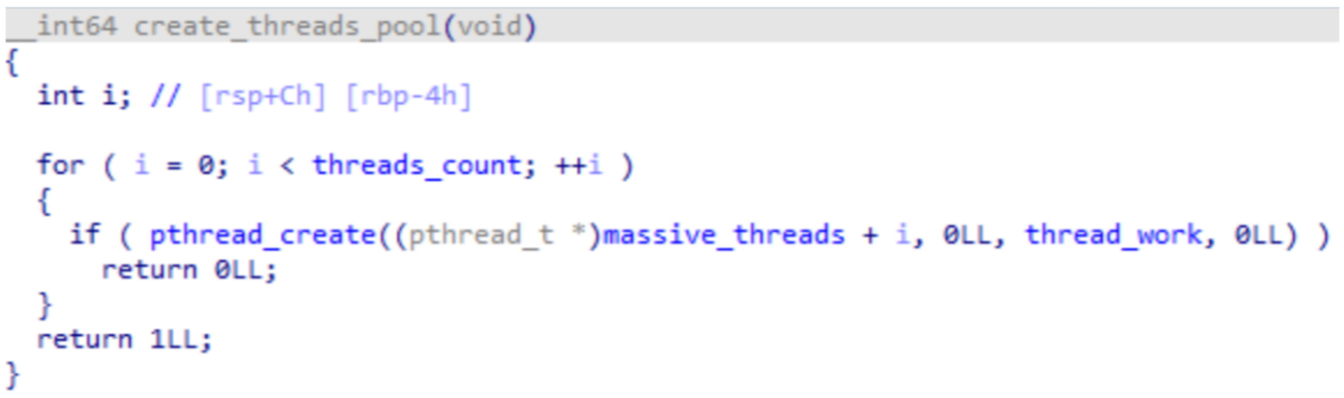
/bin/sh -c ps > PS_list
does not drop the file PID

does not check if the process has already been started		
-fork	-vmsyslog	Used to create fork processes and terminate watchdog timers
Does not terminate processes with the string vmsyslogd in its name		
-logs	-demonoff	Used to display terminal logs

Table 1. A comparison of arguments for the Linux versions of BlackSuit and Royal



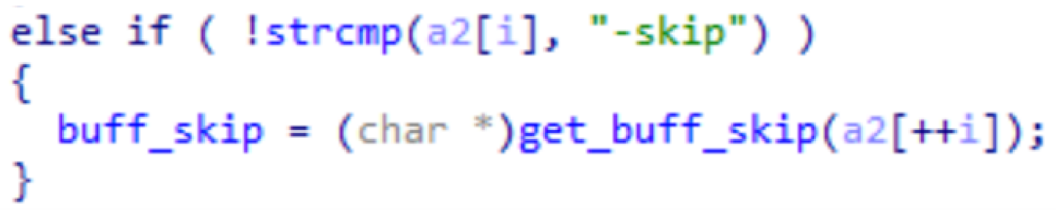
```
else if ( !strcmp(a2[i], "-thrcount") )
{
  if ( !a2[++i] )
    return 0LL;
  threads_count = atoi(a2[i]);
}
```



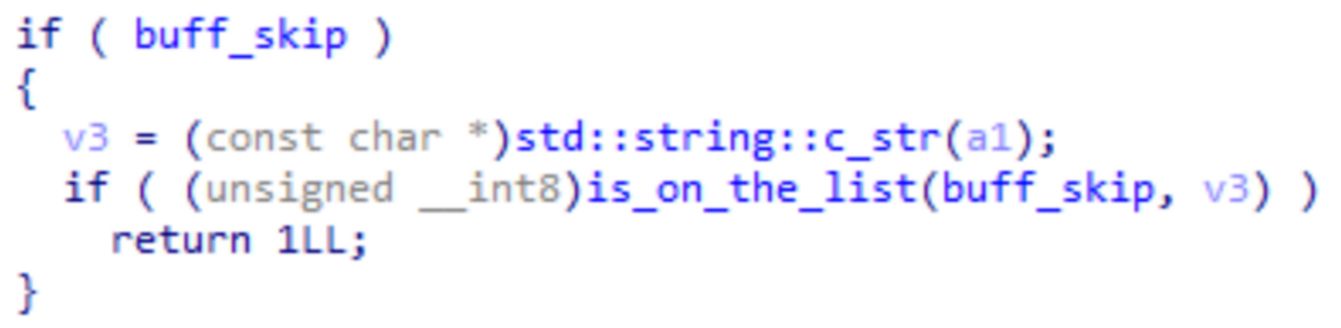
```
__int64 create_threads_pool(void)
{
  int i; // [rsp+Ch] [rbp-4h]

  for ( i = 0; i < threads_count; ++i )
  {
    if ( pthread_create((pthread_t *)massive_threads + i, 0LL, thread_work, 0LL) )
      return 0LL;
  }
  return 1LL;
}
```

Figure 4. Creation of threads based on the value of “-thrcount”

Meanwhile, the *skip* argument is used to indicate a text file that contains a list of folders to be skipped.



```
else if ( !strcmp(a2[i], "-skip") )
{
  buff_skip = (char *)get_buff_skip(a2[++i]);
}
```



```
if ( buff_skip )
{
  v3 = (const char *)std::string::c_str(a1);
  if ( (unsigned __int8)is_on_the_list(buff_skip, v3) )
    return 1LL;
}
```

Figure 5. The “skip” argument used to enumerate folders to skip
During file enumeration and encryption, each respective ransomware family avoids files with the following extensions and filenames:

Royal	BlackSuit
• .royal_u • .royal_w • .sf • .v00 • .b00 • royal_log_ • readme	• .blacksuit • .BlackSuit • .blacksuit_log_ • .list_ • .PID_ • .PS_list • .PID_list_ • .CID_list_ • .sf • .v00 • .b00 • .README.BlackSuit.txt • .README.blacksuit.txt

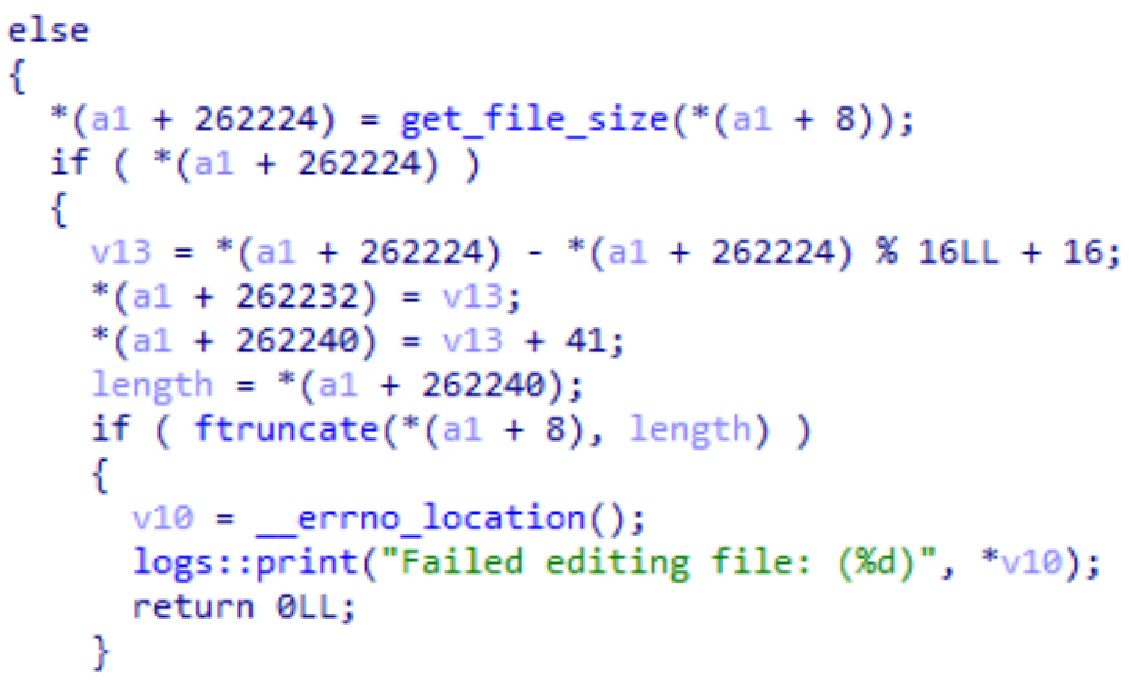
Table 2. List of extensions and filenames skipped by both BlackSuit and Royal

BlackSuit ransomware targets the following extensions if the *–allfiles* argument is not provided:

- .vmem
- .vmdk
- .nvram
- .vmsd
- .vmsn
- .vmss
- .vmtm
- .vmxf
- .vmxf
- .vmx

Intermittent encryption process

The binaries for both BlackSuit and Royal use OpenSSL’s AES for encryption and employ similar intermittent encryption techniques to accelerate the encryption of the victim’s files.



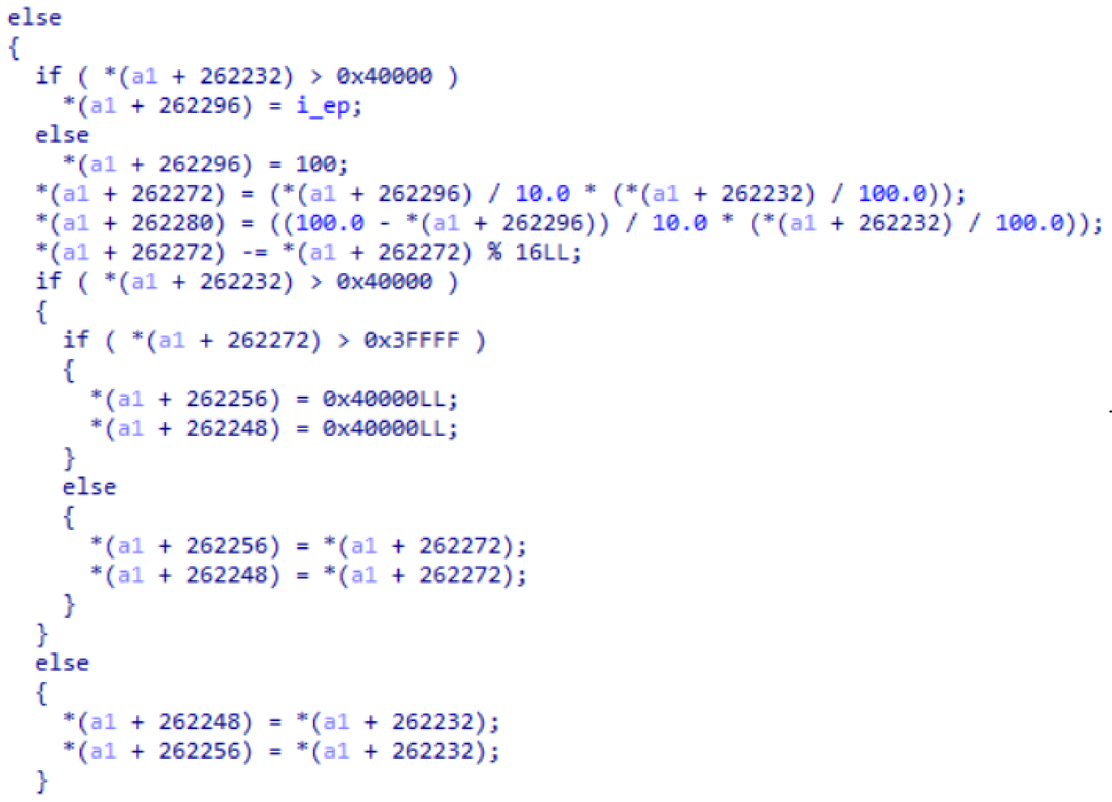
```
else
{
  *(a1 + 262224) = get_file_size(*(a1 + 8));
  if ( *(a1 + 262224) )
  {
    v13 = *(a1 + 262224) - *(a1 + 262224) % 16LL + 16;
    *(a1 + 262232) = v13;
    *(a1 + 262240) = v13 + 41;
    length = *(a1 + 262240);
    if ( ftruncate(*(a1 + 8), length) )
    {
      v10 = __errno_location();
      logs::print("Failed editing file: (%d)", *v10);
      return 0LL;
    }
```

Figure 6.

Preparing the file for encryption

Both BlackSuit and Royal prepare the files for encryption by rounding up the file size to the nearest multiple of 16, after which 41 bytes are added, possibly to account for the encryption header and other metadata.

Next, a check is performed for the file being encrypted to determine if it has a size that is greater than 0x40000h (approximately 262KB). If this condition is met, it will use the value set using *-percent*, which is represented here by the *i_ep* variable. If not, it will use the default, which is 100.



```
else
{
  if ( *(a1 + 262232) > 0x40000 )
    *(a1 + 262296) = i_ep;
  else
    *(a1 + 262296) = 100;
  *(a1 + 262272) = (*(a1 + 262296) / 10.0 * (*(a1 + 262232) / 100.0));
  *(a1 + 262280) = ((100.0 - *(a1 + 262296)) / 10.0 * (*(a1 + 262232) / 100.0));
  *(a1 + 262272) -= *(a1 + 262272) % 16LL;
  if ( *(a1 + 262232) > 0x40000 )
  {
    if ( *(a1 + 262272) > 0x3FFFF )
    {
      *(a1 + 262256) = 0x40000LL;
      *(a1 + 262248) = 0x40000LL;
    }
    else
    {
      *(a1 + 262256) = *(a1 + 262272);
      *(a1 + 262248) = *(a1 + 262272);
    }
  }
  else
  {
    *(a1 + 262248) = *(a1 + 262232);
    *(a1 + 262256) = *(a1 + 262232);
  }
```

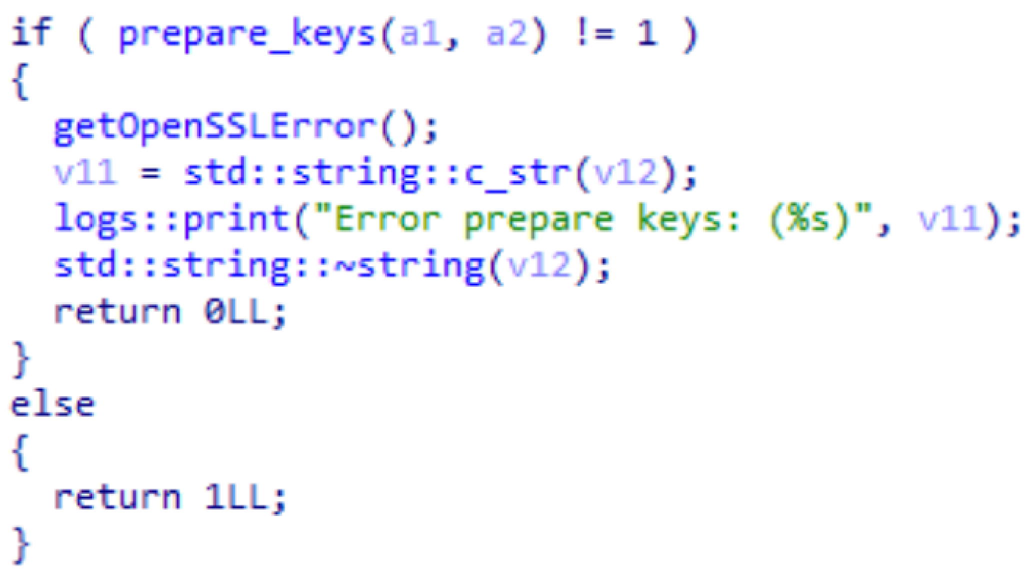
Figure 7.

Calculation of bytes to be used for intermittent encryption

The number of bytes to be used for intermittent encryption is then calculated using the same formula found in the Linux version of Royal ransomware:

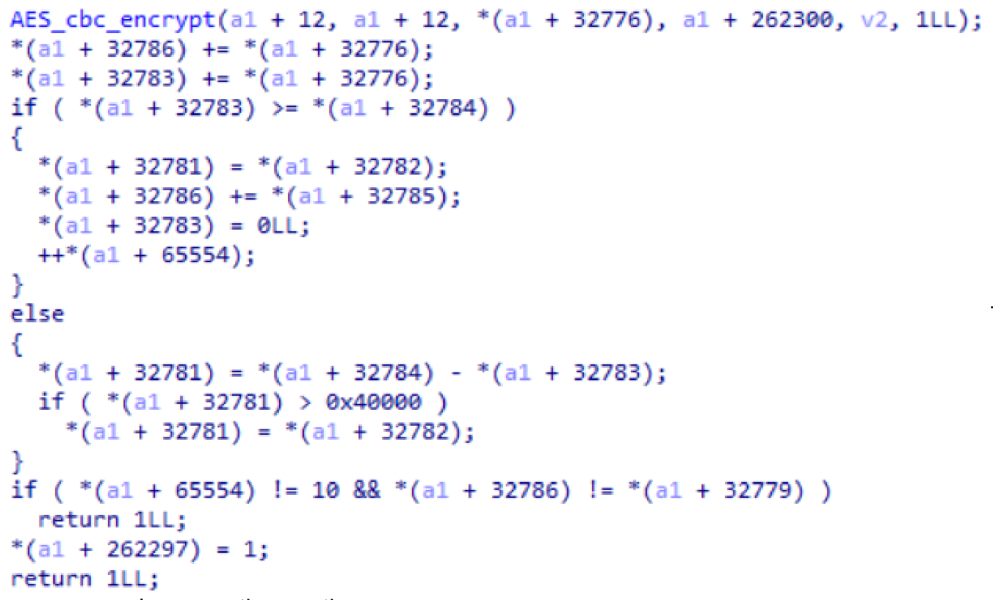
N = (X/10)*(Original File Size / 100) then round down to multiples of 16
Where X is the value of “-percent”

The file size is again checked to calculate the amount of space to be allocated for the data and metadata. Finally, the keys to be used for encryption are prepared.



```
if ( prepare_keys(a1, a2) != 1 )
{
  getOpenSSLError();
  v11 = std::string::c_str(v12);
  logs::print("Error prepare keys: (%s)", v11);
  std::string::~string(v12);
  return 0LL;
}
else
{
  return 1LL;
}
```

Figure 8. Preparation of the keys to be used for AES encyption



```
AES_cbc_encrypt(a1 + 12, a1 + 12, *(a1 + 32776), a1 + 262300, v2, 1LL);
*(a1 + 32786) += *(a1 + 32776);
*(a1 + 32783) += *(a1 + 32776);
if ( *(a1 + 32783) >= *(a1 + 32784) )
{
  *(a1 + 32781) = *(a1 + 32782);
  *(a1 + 32786) += *(a1 + 32785);
  *(a1 + 32783) = 0LL;
  ++*(a1 + 65554);
}
else
{
  *(a1 + 32781) = *(a1 + 32784) - *(a1 + 32783);
  if ( *(a1 + 32781) > 0x40000 )
    *(a1 + 32781) = *(a1 + 32782);
}
if ( *(a1 + 65554) != 10 && *(a1 + 32786) != *(a1 + 32779) )
  return 1LL;
*(a1 + 262297) = 1;
return 1LL;
```

Figure 9. The BlackSuit ransomware’s encryption routine

In the case of BlackSuit, as we previously mentioned, it appends the extension “.blacksuit” to encrypted files and drops a ransom note in the directory where the files are located.

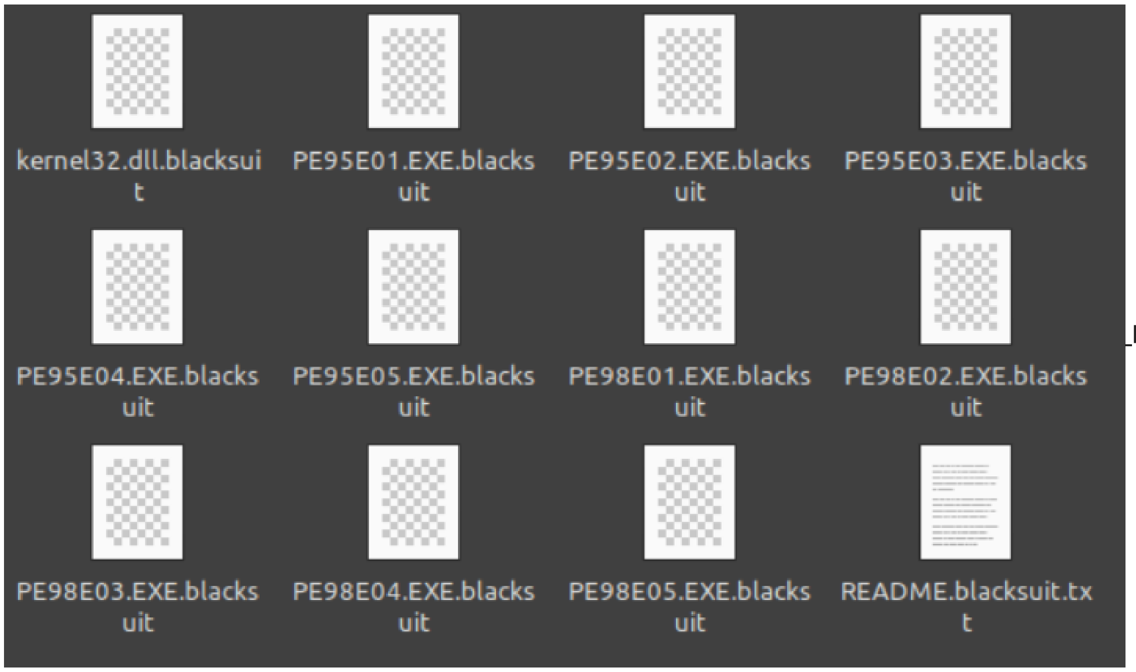
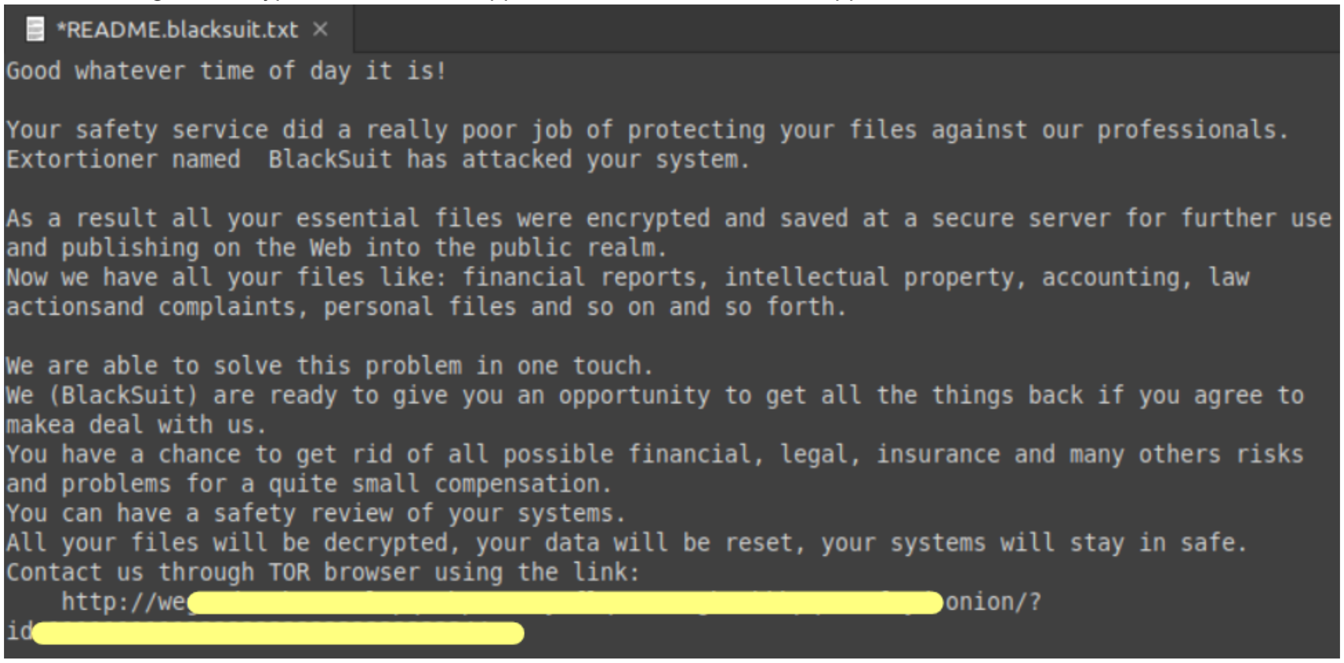


_Figure 10. The folder showing the encrypted files with the appended extension and the dropped ransom note

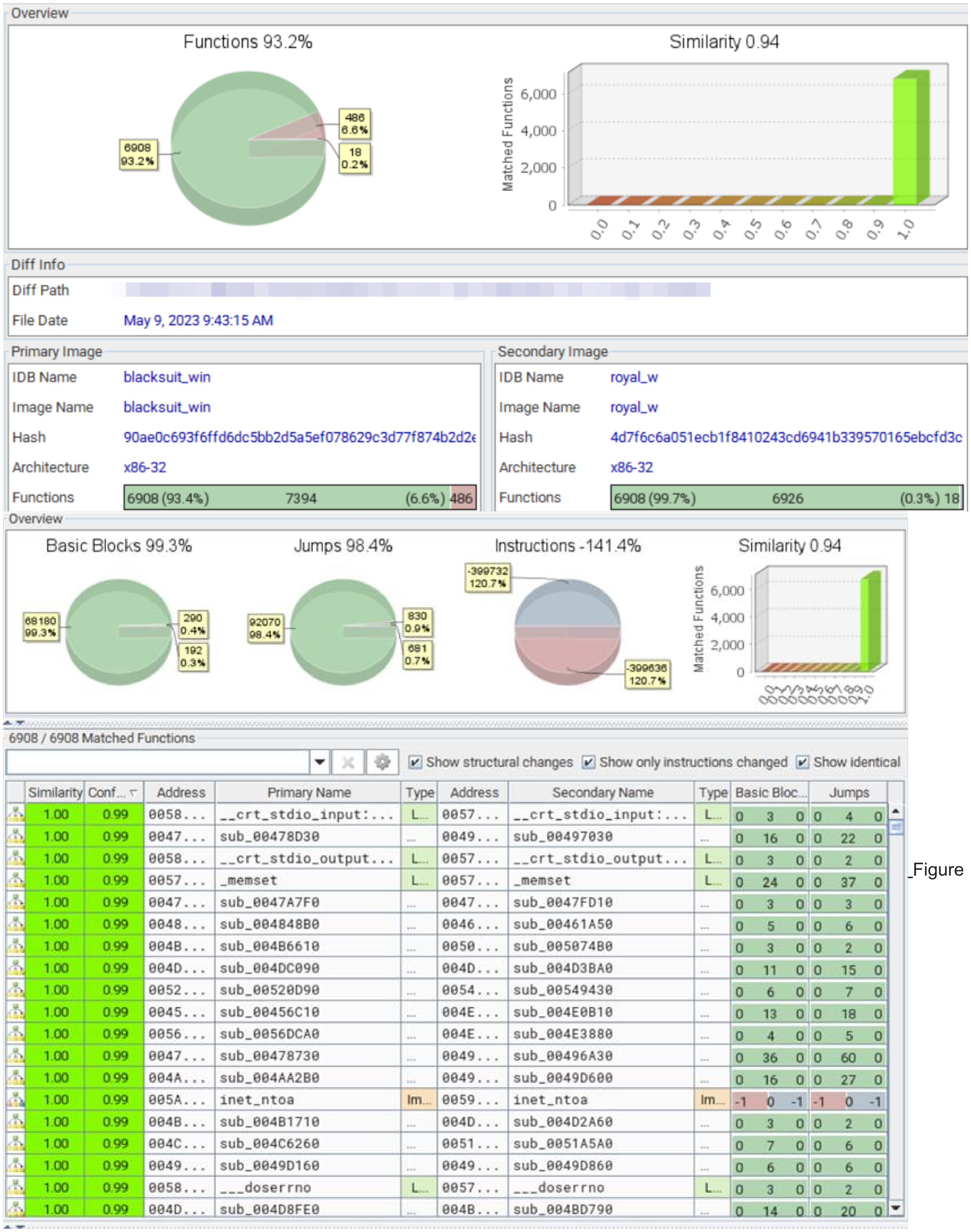


```
*README.blacksuit.txt ×
Good whatever time of day it is!

Your safety service did a really poor job of protecting your files against our professionals.
Extortioner named  BlackSuit has attacked your system.

As a result all your essential files were encrypted and saved at a secure server for further use
and publishing on the Web into the public realm.
Now we have all your files like: financial reports, intellectual property, accounting, law
actionsand complaints, personal files and so on and so forth.

We are able to solve this problem in one touch.
We (BlackSuit) are ready to give you an opportunity to get all the things back if you agree to
makea deal with us.
You have a chance to get rid of all possible financial, legal, insurance and many others risks
and problems for a quite small compensation.
You can have a safety review of your systems.
All your files will be decrypted, your data will be reset, your systems will stay in safe.
Contact us through TOR browser using the link:
    http://we[redacted]onion/?
id[redacted]
```

Figure 11. The content of the ransom note

Comparison between Royal Win32 and BlackSuit Win32 variants

In addition to the Linux-based sample, we also analyzed a Windows 32-bit version of BlackSuit, which also exhibits significant similarities with its Royal ransomware counterpart (93.2% similarity in functions, 99.3% in basic blocks, and 98.4% in jumps based on BinDiff).



Overview

Diff Info

Diff Path	[redacted]
File Date	May 9, 2023 9:43:15 AM

Primary Image

IDB Name	blacksuit_win
Image Name	blacksuit_win
Hash	90ae0c693f6ffd6dc5bb2d5a5ef078629c3d77f874b2d2e
Architecture	x86-32
Functions	6908 (93.4%) 7394 (6.6%) 486

Secondary Image

IDB Name	royal_w
Image Name	royal_w
Hash	4d7f6c6a051ecb1f8410243cd6941b339570165ebcfd3c
Architecture	x86-32
Functions	6908 (99.7%) 6926 (0.3%) 18

6908 / 6908 Matched Functions

Similarity	Conf...	Address	Primary Name	Type	Address	Secondary Name	Type	Basic Bloc...			Jumps		
1.00	0.99	0058...	__crt_stdio_input:...	L...	0057...	__crt_stdio_input:...	L...	0	3	0	0	4	0
1.00	0.99	0047...	sub_00478D30	...	0049...	sub_00497030	...	0	16	0	0	22	0
1.00	0.99	0058...	__crt_stdio_output...	L...	0057...	__crt_stdio_output...	L...	0	3	0	0	2	0
1.00	0.99	0057...	_memset	L...	0057...	_memset	L...	0	24	0	0	37	0
1.00	0.99	0047...	sub_0047A7F0	...	0047...	sub_0047FD10	...	0	3	0	0	3	0
1.00	0.99	0048...	sub_004848B0	...	0046...	sub_00461A50	...	0	5	0	0	6	0
1.00	0.99	004B...	sub_004B6610	...	0050...	sub_005074B0	...	0	3	0	0	2	0
1.00	0.99	004D...	sub_004DC090	...	004D...	sub_004D3BA0	...	0	11	0	0	15	0
1.00	0.99	0052...	sub_00520D90	...	0054...	sub_00549430	...	0	6	0	0	7	0
1.00	0.99	0045...	sub_00456C10	...	004E...	sub_004E0B10	...	0	13	0	0	18	0
1.00	0.99	0056...	sub_0056DCA0	...	004E...	sub_004E3880	...	0	4	0	0	5	0
1.00	0.99	0047...	sub_00478730	...	0049...	sub_00496A30	...	0	36	0	0	60	0
1.00	0.99	004A...	sub_004AA2B0	...	0049...	sub_0049D600	...	0	16	0	0	27	0
1.00	0.99	005A...	inet_ntoa	Im...	0059...	inet_ntoa	Im...	-1	0	-1	-1	0	-1
1.00	0.99	004B...	sub_004B1710	...	004D...	sub_004D2A60	...	0	3	0	0	2	0
1.00	0.99	004C...	sub_004C6260	...	0051...	sub_0051A5A0	...	0	7	0	0	6	0
1.00	0.99	0049...	sub_0049D160	...	0049...	sub_0049D860	...	0	6	0	0	6	0
1.00	0.99	0058...	___doserrno	L...	0057...	___doserrno	L...	0	3	0	0	2	0
1.00	0.99	004D...	sub_004D8FE0	...	004B...	sub_004BD790	...	0	14	0	0	20	0

_Figure

12. Comparison of the Linux variants of BlackSuit and Royal ransomware

Our analysis found that BlackSuit accepts the following command-line arguments:

Royal Arguments	**BlackSuit Arguments**	**Description**

-path {target path}	-p {target path}	If provided, will only encrypt the contents of the target path
-id {32-byte characters}	-name {32-byte characters}	Used as the victim's ID, which will be appended to the TOR link found in the dropped ransom note. The process exits if the argument is not provided, or if the provided characters do not have a length of 32 bytes
-ep	-percent {0 to 100}	Used to define encyption parameters
(Not in Royal)	-list {text files}	Used to specify a text file containing the target directories to encrypt
(Not in Royal)	-delete	Used to delete itself
-networkonly	-network	Used to encrypt file shares connected to the system
-localonly	-local	Used to encrypt the local system only
-disablesafeboot	-disablesafeboot	Used to disable safeboot
-noprotect	-noprotect	Used to disable mutex creation

Table 3. A comparison of arguments for the Win32 versions of BlackSuit and Royal

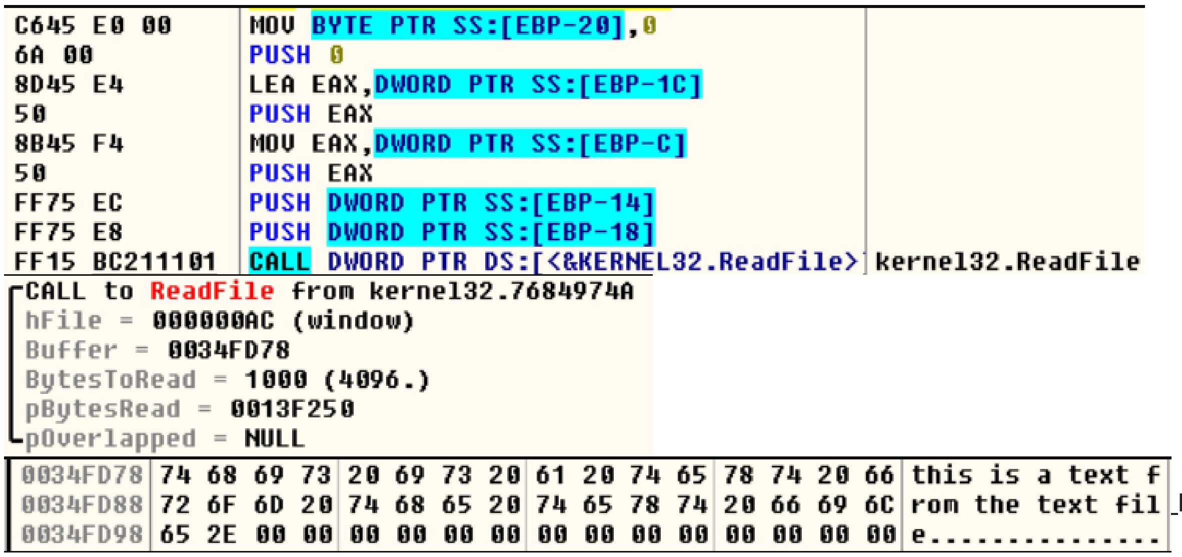
While BlackSuit introduces different argument strings compared to Royal, their purpose remains similar. BlackSuit combines arguments from various Windows versions of Royal Ransomware, while also introducing new arguments such as "-delete" and "-list" that are specific to itself.

The *-delete* argument uses the following command to continuously check for the existence of its file by looking for the filename:

```
cmd /v/c "set f={Malware File Name}&for /l %l in () do if exist !f! (del /f/a "!f!") else (exit)"
```

If the file is found, it is immediately deleted. The command keeps running indefinitely until the file is deleted, at which point the loop will exit.

The *-list* argument is used to specify a text file containing target directories to encrypt. It loads the file using *ReadFileFAPI* then places the contents of the text file in a buffer. Note that the loaded text file is a sample text file we used for testing and not the format of the text file that will be loaded in an actual attack.



```
C645 E0 00     MOV BYTE PTR SS:[EBP-20],0
6A 00          PUSH 0
8D45 E4        LEA EAX,DWORD PTR SS:[EBP-1C]
50             PUSH EAX
8B45 F4        MOV EAX,DWORD PTR SS:[EBP-C]
50             PUSH EAX
FF75 EC        PUSH DWORD PTR SS:[EBP-14]
FF75 E8        PUSH DWORD PTR SS:[EBP-18]
FF15 BC211101  CALL DWORD PTR DS:[<&KERNEL32.ReadFile>]   kernel32.ReadFile
CALL to ReadFile from kernel32.7684974A
 hFile = 000000AC (window)
 Buffer = 0034FD78
 BytesToRead = 1000 (4096.)
 pBytesRead = 0013F250
 pOverlapped = NULL
0034FD78 74 68 69 73 20 69 73 20 61 20 74 65 78 74 20 66 this is a text f
0034FD88 72 6F 6D 20 74 68 65 20 74 65 78 74 20 66 69 6C rom the text fil
0034FD98 65 2E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 e...............
```

Figure 13. Loading the text file. Note that we loaded the sample text file to show that it loads the file when using the -list argument.

if –*disablesafeboot* is passed as an argument, it removes the "safeboot" value from the current boot entry in the Boot Configuration Data (BCD) and performs an immediate system restart via the following command:

```
“%System%\bcdedit.exe" /deletevalue {current} safeboot
shutdown.exe /r /t 0
```

When encrypting network shares using the *-network* argument, BlackSuit will check if the IP address begins with the following numbers to ensure that it is encrypting local systems:

- 192.168.
- 10.
- 100.
- 172.

It avoids encrypting files with the following strings in their file path:

Royal	BlackSuit
• $recycle.bin • $windows.~bt • $windows.~ws • boot • google • mozilla • perflogs • tor browser • windows • windows.old • royal	• Windows • ADMIN$ • IPC$

Table 4. Royal and BlackSuit avoid encrypting files that have these strings

Royal	BlackSuit
• .exe • .dll • .bat • .lnk • .royal_u • .royal_w	• .exe • .dll • .BlackSuit • .blacksuit • README.BlackSuit.txt

Table 5. Royal and BlackSuit avoid encrypting files that contain these extensions

BlackSuit ransomware also deletes shadow copies using the following command:

```
"%System%\vssadmin.exe" Delete Shadows /All /Quiet
```

Conclusion and insights

The emergence of BlackSuit ransomware (with its similarities to Royal) indicates that it is either a new variant developed by the same authors, a copycat using similar code, or an affiliate of the Royal ransomware gang that has implemented modifications to the original family.

One possibility for BlackSuit’s creation is that, since the threat actors behind Royal (and Conti before it) are one of the most active ransomware groups in operation today, this may have led to increased attention from other cybercriminals, who were then inspired to develop a similar ransomware in BlackSuit. Another option is that BlackSuit emerged from a splinter group within the original Royal ransomware gang.

Whatever the case may be, the emergence of another ransomware like BlackSuit provides further evidence that threat actors will always try to look for more effective tools for their attacks, from modifying existing code to developing unique ransomware families, to profit from their victims. As such, both organizations and individual users should remain vigilant when it comes to protecting their files and data from ransomware attacks.

Recommendations and solutions

Organizations can defend against ransomware attacks by implementing a comprehensive security framework that directs resources towards establishing a strong defense strategy. Here are some recommendations:

- Create an inventory of assets and data
- Identify authorized and unauthorized devices and software
- Conduct audits of event and incident logs
- Manage hardware and software configurations
- Grant administrative privileges and access only when necessary
- Monitor network ports, protocols, and services
- Establish a whitelist of approved software applications
- Implement measures for data protection, backup, and recovery
- Enable multifactor authentication (MFA)
- Deploy up-to-date security solutions across all system layers
- Remain vigilant for early indications of an attack

By adopting a multi-pronged approach to securing potential entry points, such as endpoints, emails, websites, and networks, organizations can detect and defend against malicious elements and suspicious activities, effectively safeguarding themselves from ransomware attacks.

A multilayered approach can help organizations guard possible entry points into their system (endpoint, email, web, and network). Security solutions can detect malicious components and suspicious behavior, which can help protect enterprises.

Indicators of Compromise (IOCs)

SHA256	Detection name
90ae0c693f6ffd6dc5bb2d5a5ef078629c3d77f874b2d2ebd9e109d8ca049f2c	Ransom.Win32.BLACKSUIT.THEODBC
1c849adcccad4643303297fb66bfe81c5536be39a87601d67664af1d14e02b9e	Ransom.Linux.BLACKSUIT.THEODBC
6ac8e7384767d1cb6792e62e09efc31a07398ca2043652ab11c090e6a585b310	Ransom.Win32.ROYAL.AA
4d7f6c6a051ecb1f8410243cd6941b339570165ebcfd3cc7db48d2a924874e99	Ransom.Win32.ROYAL.SMYECJYT
b57e5f0c857e807a03770feb4d3aa254d2c4c8c8d9e08687796be30e2093286c	Ransom.Linux.ROYAL.THBOBBC