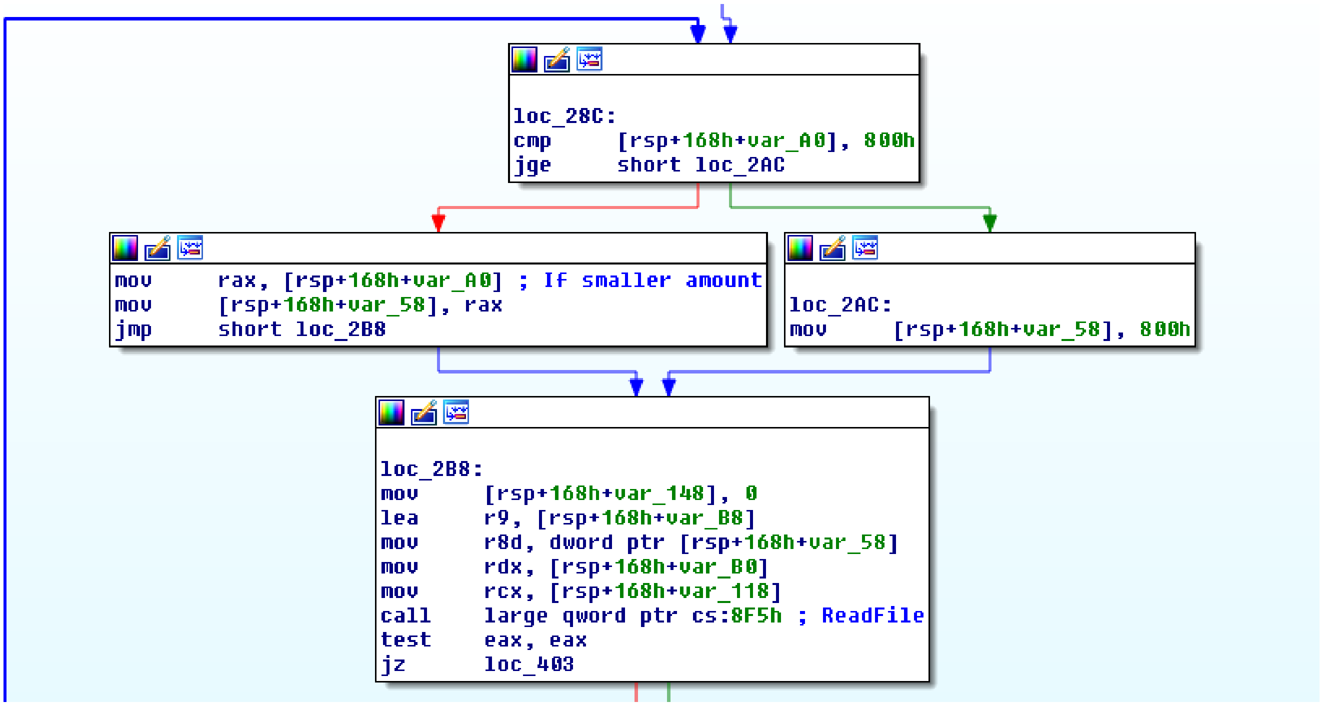
Diavol resurfaces

medium.com/walmartglobaltech/diavol-resurfaces-91dd93c7d922

Jason Reaves September 30, 2022



```
loc_28C:
cmp     [rsp+168h+var_A0], 800h
jge     short loc_2AC
```

```
mov     rax, [rsp+168h+var_A0] ; If smaller amount
mov     [rsp+168h+var_58], rax
jmp     short loc_2B8
```

```
loc_2AC:
mov     [rsp+168h+var_58], 800h
```

```
loc_2B8:
mov     [rsp+168h+var_148], 0
lea     r9, [rsp+168h+var_B8]
mov     r8d, dword ptr [rsp+168h+var_58]
mov     rdx, [rsp+168h+var_B0]
mov     rcx, [rsp+168h+var_118]
call    large qword ptr cs:8F5h ; ReadFile
test    eax, eax
jz      loc_403
```

Jason Reaves

Sep 30

·

5 min read

By: Jason Reaves and Jonathan McCay

We previously walked through the Diavol ransomware variants file encryption[1] which has been linked to the TrickBot group[2]. After the recent breakup[3,4], Diavol all but seemed to have disappeared. Curiously, we began to notice an uptick in samples submitted to VirusTotal. While investigating the more recent samples, we were able to determine that it uses a mix of RSA encryption and XOR encoding for files. In some instances, file recovery is still possible.

The following samples were identified on VirusTotal:

```
SHA256: aac969e36686f8f8517c111d30f8fb3b527988ebd31b3b762aec8d46e860eb9d
Creation Time 2022-09-05 20:01:56 UTC
First Submission 2022-09-09 21:06:06 UTC
Last Submission 2022-09-13 15:50:00 UTC
Last Analysis 2022-09-13 15:50:00 UTC
```

```
SHA256: fb5ee29b98446d34520bf04a82996eefec3b5692710c5631458da63ef7e44fe4
Creation Time 2022-09-05 20:04:30 UTC
First Submission 2022-09-11 20:30:20 UTC
Last Submission 2022-09-11 20:30:20 UTC
Last Analysis 2022-09-11 20:30:20 UTC
```

```
SHA256: 708806f5e2e8bfa3d1e911e391ff2ccf1edcac05cc1df80439b8b867253423dfCreation Time
2022-08-25 16:12:58 UTCFirst Submission 2022-08-29 19:49:08 UTC Last Submission 2022-
09-03 15:40:44 UTC Last Analysis 2022-09-03 15:40:44 UTC
```

The samples are now 64 bit but function similarly. For the purposes of this report we will be going through the 7088 sample above. For the purposes of this report, we will be going through the 7088 sample above.

```
group=testfile_ext=.bullynote_filename=WARNING.txt
```

File encryption still involves the use of a 2048 byte XOR key which is randomly generated in the GENBOTID piece of the main bot. The key is then stored in the main bot and reused later in the file encryption code. Then a loop will sit reading chunks of 2048 bytes unless the amount of data to be encoded is less than 2048:

The first part of the file encryption is the aforementioned usage of the 2048 byte XOR key. For most files, the amount of bytes that will be XOR encoded is based on the overall file size divided by 10. Then a loop will sit reading chunks of 2048 bytes unless the amount of data to be encoded is less than 2048:

A similar XOR loop has been implemented, which can be seen in the previous version of Diavol[1]. The loop will handle XOR encoding the chunk of data that was read before writing it back to the file:

After XOR encoding the file, the RSA encrypted XOR key is written to the end of the file followed by the number of encoded bytes:

Next the bot single XOR encodes the number of encoded bytes and writes that to the end of the file:

After XOR encoding and writing the appropriate data to the end of the file, the bot goes back to the beginning of the file and begins reading in chunks of 0x75 bytes. It will RSA encrypt them and the encrypted bytes are then written back to the file but without the padding bytes.

In this way, 0x75 * 10 or 1170 bytes at the beginning of the file will be RSA encrypted after getting XOR encoded.

A quick test can be performed to validate our findings, using a file of NULLs and a large MSI file. First, we validate the end data that was added to the file, which should be 110+0x900+16 bytes from the end:

```
>>> data = open('test_data.txt.bully', 'rb').read()>>> 110+0x900+16 2430>>> end =
data[-2430:]
```

Skipping over the RSA encrypted XOR key should show the two 8 byte values with the second being XOR encoded with 0xFF

```
>>>
end[0x900:]'\x88\x13\x00\x00\x00\x00\x00\x00w\xec\xff\xff\xff\xff\xff\xffk\xa8\x0f/6o\
\xf4\xbd\x03\xf9H\x01\x99\xa6\xd7\x9ae\xee\xf3\xa7\xe9\xc6\xb1\xf8\x81\xe0\xb6\xc4\xba
 0x13885000>>>
end[0x900+8:]'w\xec\xff\xff\xff\xff\xff\xffk\xa8\x0f/6o\x12\x08\xd6\xbe\xaaw\xf1\x1b0\
\xf4\xbd\x03\xf9H\x01\x99\xa6\xd7\x9ae\xee\xf3\xa7\xe9\xc6\xb1\xf8\x81\xe0\xb6\xc4\xba
 l = bytearray('w\xec')>>> l[0] ^= 0xff>>> l[1] ^= 0xff>>> lbytearray(b'\x88\x13')
```

Since the file is NULLs, the clear XOR key should be the first 2048 bytes after we skip over the 1170 RSA encrypted bytes at the beginning:

```
>>> key = bytearray(data[1170:1170+2048])
```

We can test this against another file, in this case an MSI:

```
>>> data2 = open('powerpointmui.msi.bully', 'rb').read()>>> test_block =
bytearray(data2[1170:])>>> >>> for i in range(len(test_block)):...   test_block[i] ^=
key[i%2048]...>>>
test_block[:10000]bytearray(b'\xa4A(H\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\
 Database\x00\x00\x00\x1e\x00\x00\x00;\x00\x00\x00Microsoft Office PowerPoint MUI
(Portuguese (Brazil)) 2007\x00\x00\x1e\x00\x00\x00\x16\x00\x00\x00Microsoft
Corporation\x00\x00\x00\x1e\x00\x00\x00"\x00\x00\x00Installer, MSI, Database,
Release\x00\x00\x00\x1e\x00\x00\x00\x84\x00\x00\x00This Installer database contains
the logic and data required to install Microsoft Office PowerPoint MUI (Portuguese
(Brazil)) 2007....snip...
```

So it is possible to recover most of each file trivially, after you recover the XOR key. The next step is to just rebuild the first 1170 bytes.

IOCs

Endpoint:

```
WARNING.txtwarning.txt.bully4eb5bea255c0308b296f5aa259f6862688b41ba2d6b7cca40118de9007
```

Ransom Note:

```
You've been hacked. All your corporate network servers and workstations are
encrypted.Your company is a victim of double extortion ransomware attack.\rhat is it?
Basically it means that not only your data is encrypted, but it's also have been
exfiltrated from your network.Double Extortion attack explained in details
:https://www.zscaler.com/resources/security-terms-glossary/what-is-double-extortion-
ransomware\r==== What now? =====If you want your network to be fully operational
again and if you want us not to publish all files we've taken :1. Download Tor
Browser from original site : https://torproject.org\r. Open this url in Tor Browser
and visit this website :
https://7ypnbv3snejqmgce4kbewwvym4cm5j6lkzf2hra2hyhtsvwjaxwipkyd.onion/\r. Enter this
key : 57C0E-4C543-DCABB-EBF0C-2EDCA-A9FC4If you've done everything correctly - now
you are able to contact us and take a chance to leave this all behind for a
reasonable fee.\rOTE : If TOR network is unavailable by any reason - you can use any
VPN service to solve it.
```

Network:

```
hxxps://7ypnbv3snejqmgce4kbewwvym4cm5j6lkzf2hra2hyhtsvwjaxwipkyd[.]onion173.232[.]146[
```

References

1: https://medium.com/walmartglobaltech/diavol-the-enigma-of-ransomware-1fd78ffda648

2: https://www.bleepingcomputer.com/news/security/fbi-links-diavol-ransomware-to-the-trickbot-cybercrime-group/

3: https://www.advintel.io/post/the-trickbot-saga-s-finale-has-aired-but-a-spinoff-is-already-in-the-works

4: https://www.bleepingcomputer.com/news/security/conti-ransomware-shuts-down-operation-rebrands-into-smaller-units/