BlackMatter - The New Star Of Ransomware

blog.minerva-labs.com/blackmatter

- Tweet
-

After the demise of the DarkSide ransomware affiliate program, a vacuum was left in the market. This space was promptly filled by new groups such as Lockbit and BlackMatter. BlackMatter, the up-and-coming star of the ransomware scene, is thought to be DarkSide's direct heir, from which it got some of its code. While no high-profile breach of BlackMatter have been published yet, BleepingComputer claims the group already netted a 4 million ransom payment.

Minerva's research team obtained a recent BlackMatter sample, and in this blog, we will give some crucial details of the ransomware's inner workings and evasion techniques.

Evasion Techniques:

Like most modern malware, BlackMatter employs string encryption and dynamic API function resolving, that is done to hinder static analysis. The ransomware's API resolving routine is quite unique, as it will save resolved function pointers in an encoded form and decode them only when used.

A de-compilation of the API resolving function:

```
if ( module_handle )
{
  functions_array = (int *)(a1 + 4);
  while ( 1 )
  {
    module_handle = *v4++;
    if ( module_handle == 0xCCCCCCCC )
      break;
    function_from_checksum = get_function_from_checksum(module_handle ^ 0x1002BFFF);
    stub_heap_allocation = RtlAllocateHeap(a3, 0, 16);
    if ( *(_DWORD *)(stub_heap_allocation + 16) != 0xABABABAB )
      *functions_array++ = stub_heap_allocation;
    *(_BYTE *)stub_heap_allocation = -72;
    random_int = w_gen_random(0, 4u);
    if ( random_int )
    {
      switch ( random_int )
      {
        case 1u:
          v13 = w_gen_random(1u, 9u);
          *(_DWORD *)(v14 + 1) = __ROR4__(function_from_checksum, v13);
          *(_WORD *)(v14 + 5) = -16191;
          *(_BYTE *)(v14 + 7) = v13;
          *(_WORD *)(v14 + 8) = -7937;
          break;
```

To further complicate the analysis, if a heap allocation ends with the integer 0xABABABAB the malware will not save the stub's address in the reconstructed import address table. This will effectively crash the ransomware if it is running under a debugger. More information on why this technique works can be found here under "Heap Protection".

Another anti-debugging technique used by BlackMatter is hiding threads using the NtSetInformationThread function with the parameter ThreadHideFromDebugger (0x11). This will cause a thread to "run away" from a debugger, resulting in an encrypted machine. It is

worth mentioning that this exact technique is used by Lockbit 2.0, further strengthening the evidence that the two are linked.

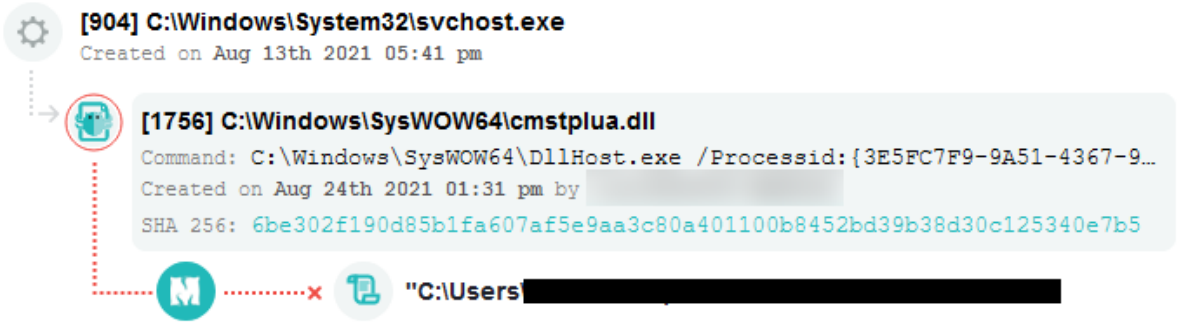
Thread hiding code from BlackMatter:

```
signed int v1; // eax

if ( thread_id )
  v1 = thread_id;
else
  v1 = -2;
return NtSetInformationThread_stub(v1, 0x11, 0, 0);
```

Unlike other ransomware, BlackMatter will encrypt Russian devices. The sample we analyzed did not bother to check the computer's locale before encrypting it. The malware does not care where it executes and will even encrypt a commercial sandbox.

Ransomware is seldom the initial payload of an attack, which must be resistant to security products, hence environmental awareness must be part of its repertoire. To evade security tools it will usually employ evasion techniques such as memory injection and 'living off the land'. Minerva Labs prevents the initial payload from executing, thereby stopping threats like BlackMatter from gaining a foothold on the network in the first place.



IOCs:

78826cac6f30f62cb6499cebc97d6a9bed22b0a2adef6e0cad14742a4b593e68

References:

- https://news.sophos.com/en-us/2021/08/09/blackmatter-ransomware-emerges-from-the-shadow-of-darkside/
- https://www.tesorion.nl/en/posts/analysis-of-the-blackmatter-ransomware/.
- https://www.bleepingcomputer.com/news/security/blackmatter-ransomware-gang-rises-from-the-ashes-of-darkside-revil/

Talk To Minerva Labs