MuddyWater Back with DarkBit

Published: 2023-04-21 · Archived: 2026-04-05 22:51:14 UTC

Recently, we came across a tweet about DarkBit ransomware. An Iranian APT group, named MuddyWater, is reportedly behind the DarkBit ransomware. In this blog we will explore the ransomware's initial access method, the use of Cobalt Strike and the final ransomware payload.

Initial Access Method

The initial lure was delivered as an ISO file.

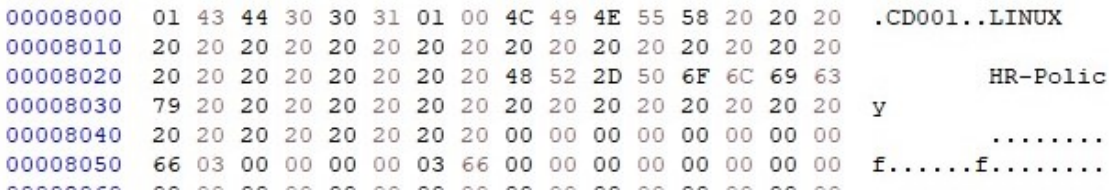


Figure 1 – ISOFile

The payload included a shortcut file (with a .doc extension) and a zip file.

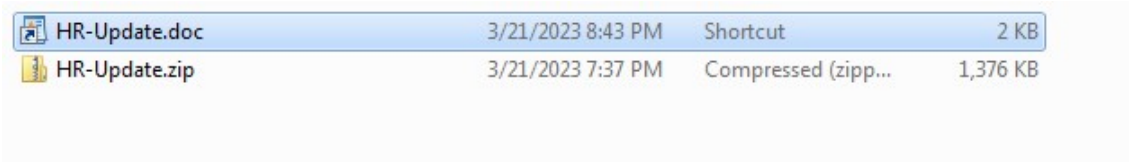


Figure 2 – Contents Inside ISO File

The shortcut was using PrintBrm.exe to unpack the HR-Update.zip and run it as shown below. PrintBrm.exe is a windows inbuilt command line tool.

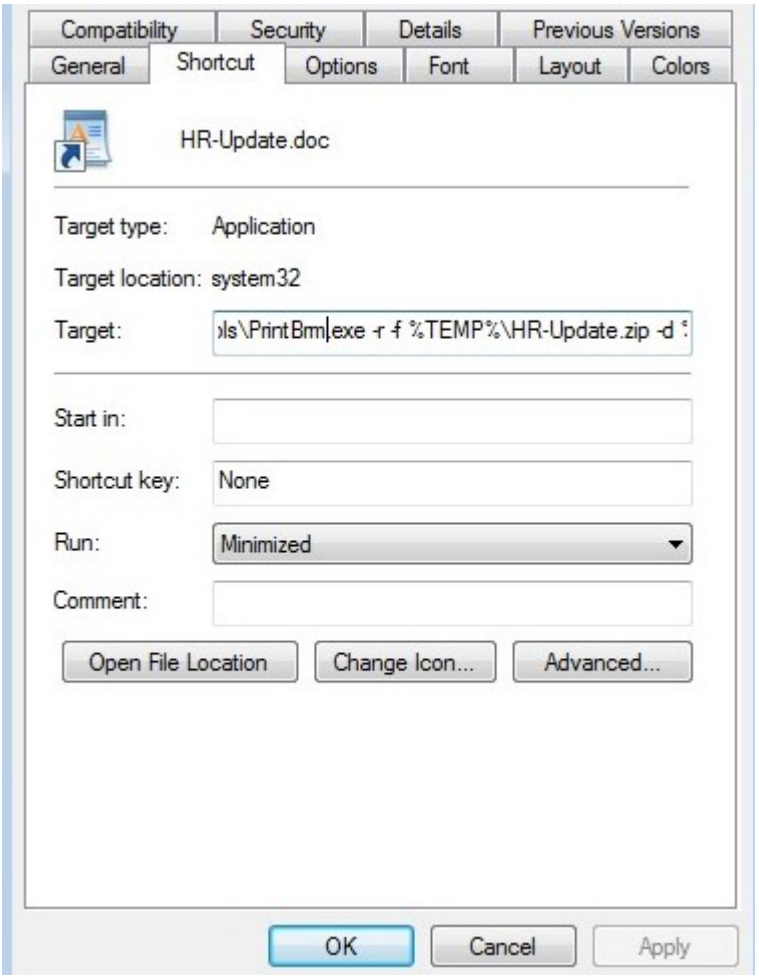


Figure 3 – Shortcut File

```
cmd.exe /c xcopy .\HR-Update.zip %TEMP% /h /y && PrintBrm.exe -r -f %TEMP%\HR-Update.zip -d %TEMP%\unzip & %TEM
```

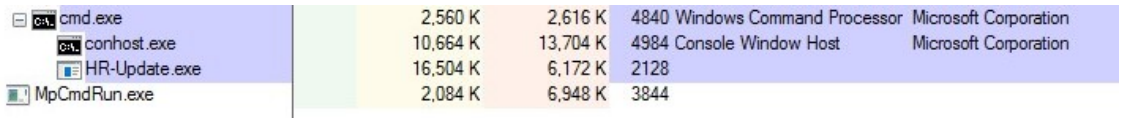


Figure 4 – HR-Update.exe Running

HR-Update.exe was a Cobalt Strike beacon. Cobalt Strike, a penetration testing tool, can also be used by attackers for gaining a foothold in the system. The final ransomware payload is downloaded with the help of Cobalt Strike.

At the time of writing the blog, we were unable to get the exact DarkBit ransomware payload. So we are using another available sample that belonged to the same campaign.

Analysis of Ransomware

This DarkBit ransomware sample is written in GoLang. It contains command line arguments.

Figure 5 – Ransomware Features

It also contains an inbuilt configuration file as shown in Figure 6.

Figure 6 – InBuilt Config

Further analysis revealed that they had obfuscated some dll names like advapi32.dll and functions like SystemFunction036.

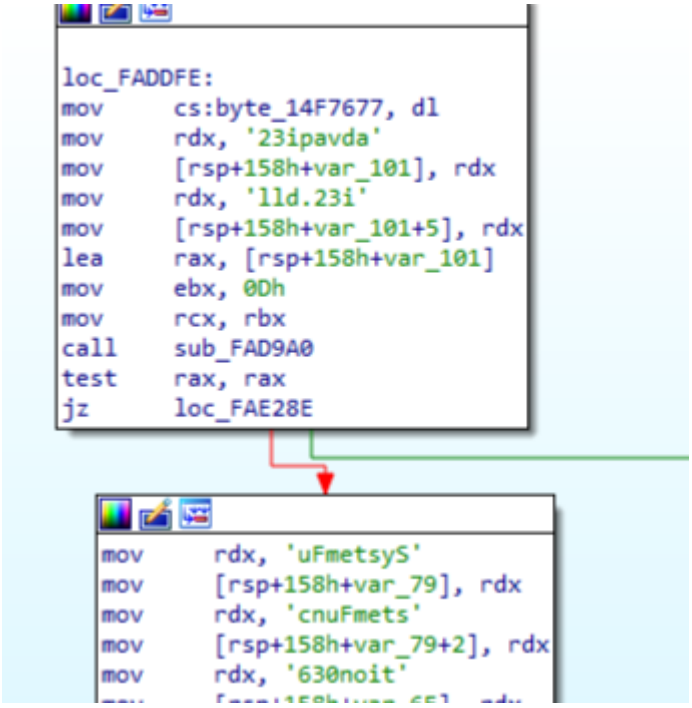


Figure 7- Obfuscation

Its dynamically resolving API at this address. Malware authors tend to dynamically resolve API to avoid static detections.

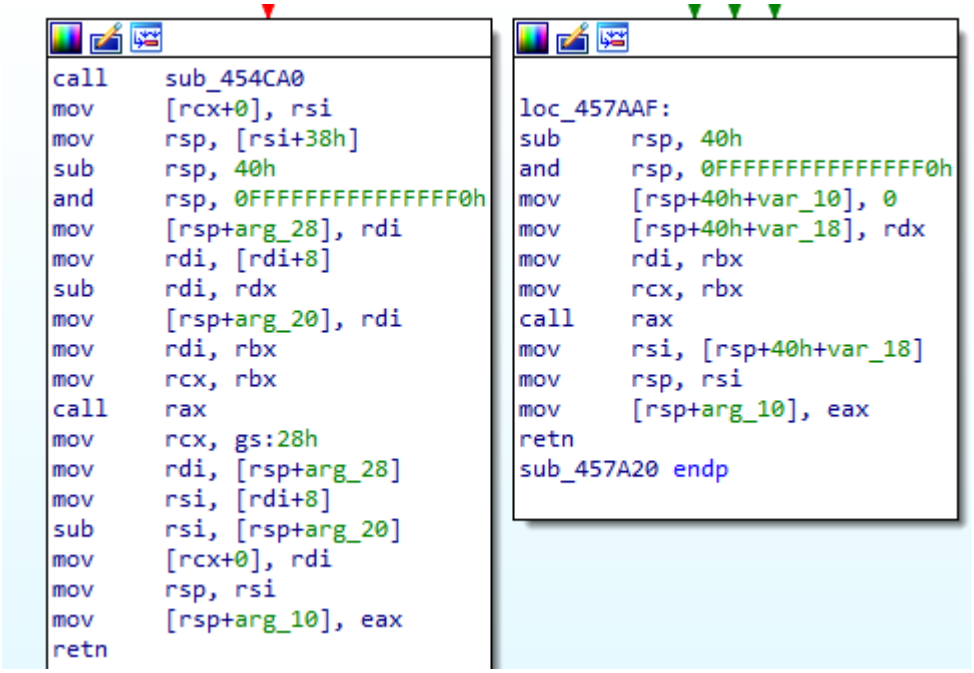


Figure 8 – Dynamically Resolving API

CreateMutexW API is being used to check if an instance of the malware is already running. As can be seen in Figure 5 previously, they are also using multithreading.

As the customary prelude to file encryption, they are using vssadmin.exe to delete all the shadow copies.

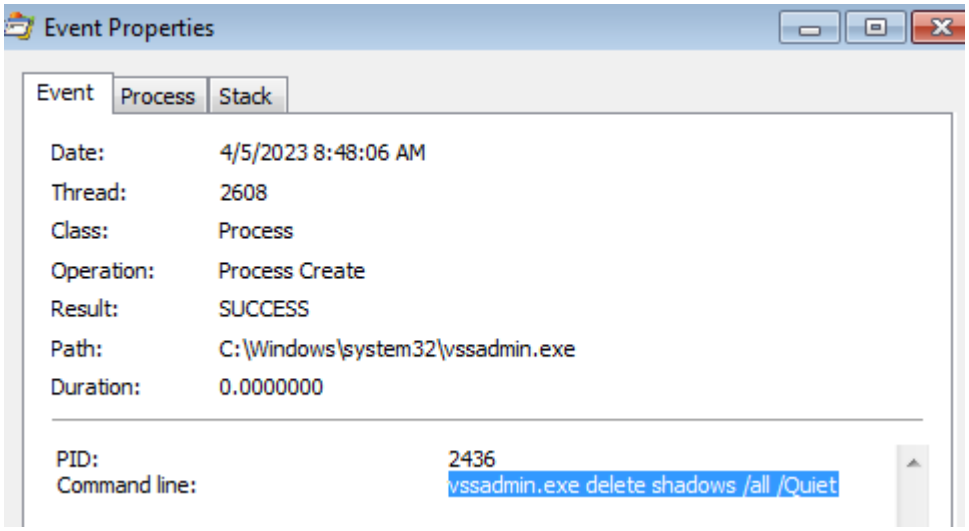


Figure 9 – Delete Shadow all

Here they are using SystemFunction036 (documented in MSDN as RtlGenRandom) to generate a random key as shown in Figure 10.

Figure 10 – SystemFunction036

SystemFunction036 is accessed multiple times in the code with varying buffer sizes passed to it.

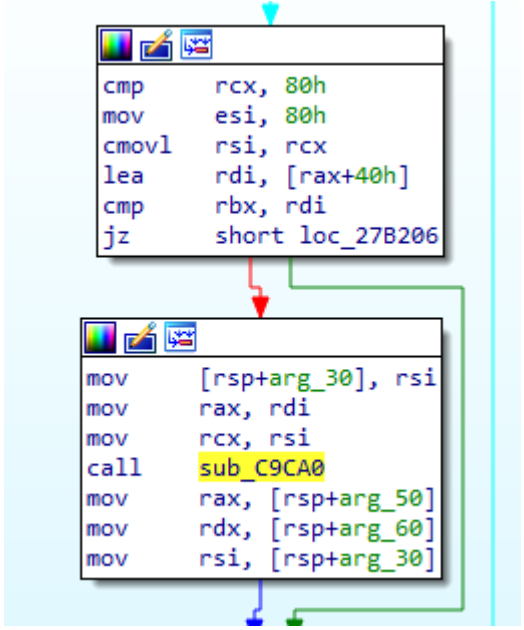


Figure 11 – Call To Dynamically Resolve API and Then SystemFunction036

Here we can see that the buffer size of 80 is made available for SystemFunction036.

```
mov     rsi, [rdx+38h]
mov     rax, [rsp+50h+arg_20]
mov     rbx, [rsp+50h+arg_28]
mov     rcx, [rsp+50h+arg_30]
mov     rdi, [rsp+50h+arg_38]
call    rsi
mov     rdx, [rsp+50h+arg_18]
mov     rsi, [rdx+38h]
mov     rax, [rsp+50h+arg_20]
mov     rbx, [rsp+50h+var_10]
mov     ecx, 4
mov     rdi, rcx
call    rsi
mov     rdx, [rsp+50h+arg_18]
mov     rsi, [rdx+30h]
mov     rax, [rsp+50h+arg_20]
mov     rbx, [rsp+50h+var_18]
xor     ecx, ecx
mov     rdi, [rsp+50h+var_20]
call    rsi
mov     [rsp+50h+var_18], rax
mov     [rsp+50h+var_28], rbx
mov     [rsp+50h+var_20], rcx
mov     rdx, [rsp+50h+arg_18]
```

Figure 12 – Encrypting Key

Later-on, the key used for encrypting the files is itself encrypted and attached to the encrypted files.

```
loc_BDAB2E:
mov     rbp, rsp
mov     rax, [rsi]
bswap   rax
mov     [rbp+0], rax
mov     rdx, 428A2F98D728AE22h
add     r15, rax
mov     rax, r12
add     r15, rdx
mov     rcx, r12
ror     rax, 0Eh
mov     rdx, r12
ror     rcx, 12h
xor     rax, rcx
mov     rcx, r12
ror     rdx, 29h
and     rcx, r13
xor     rdx, rax
mov     rax, r12
not     rax
add     r15, rdx
and     rax, r14
xor     rax, rcx
```

Figure 13 – EncryptionAlgorithm

From Figure 13, we can see that It's encrypting. It is likely using AES to encrypt the files, as strings related to the same functions can be found elsewhere in this same sample.

FindFirstFileW, FindNextFileW are used to iterate through the file system, to find the appropriate file and then encrypt it.

At that point it was observed that its writing file in chunks and not as a whole. For doing the same it's using SetFilePointerEx API to move the file pointer to a specific address.

Figure 14 – SetFilePointerEx

It's then using the WriteFile API.

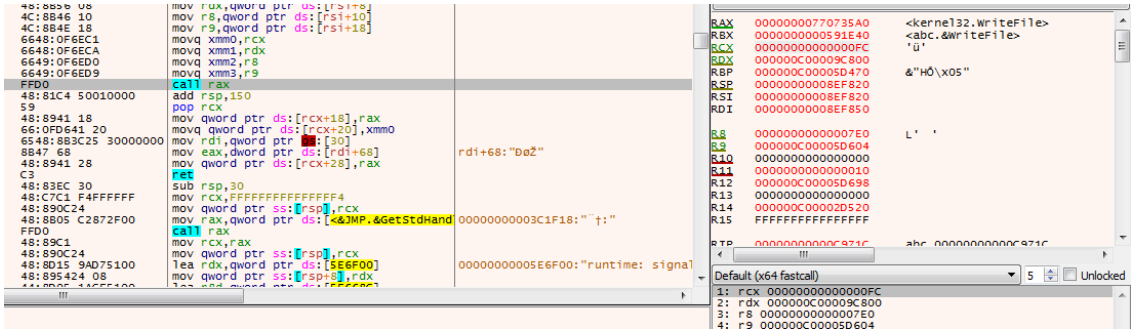
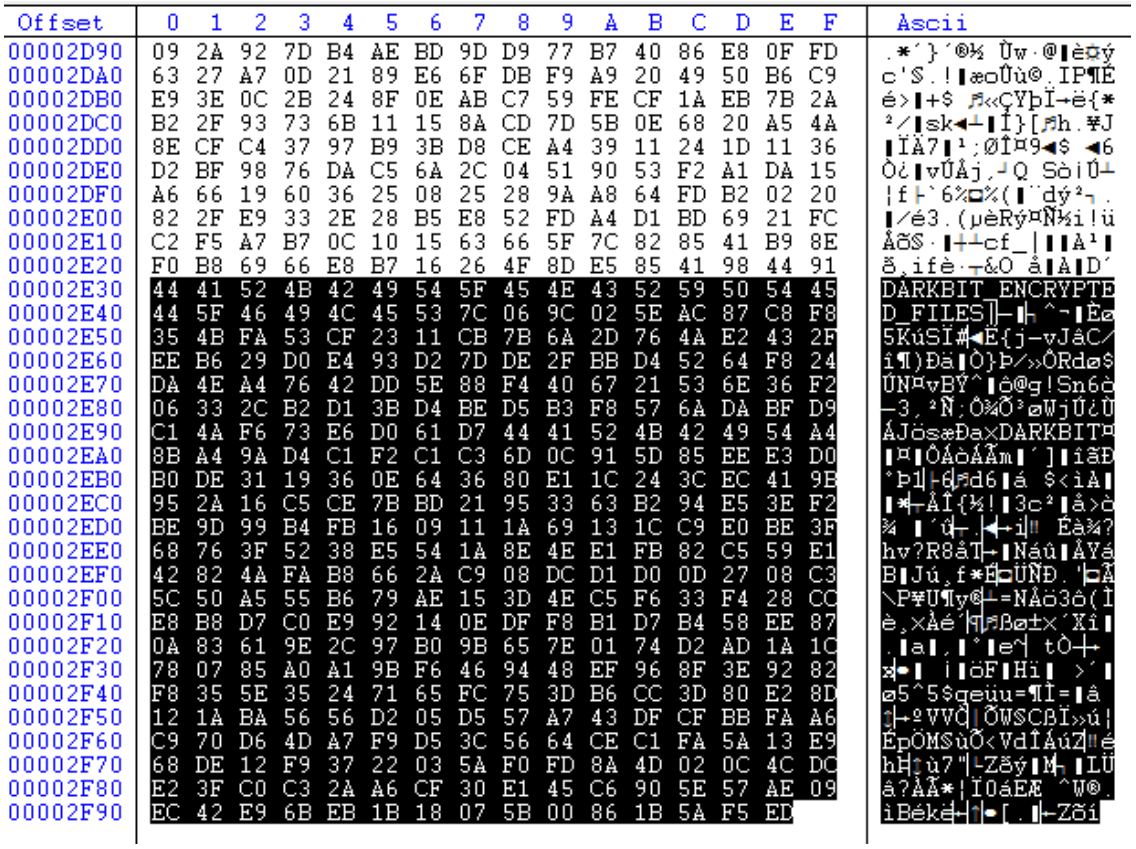


Figure 15 – WriteFile

All these functions are called one after another, till all the files are encrypted.

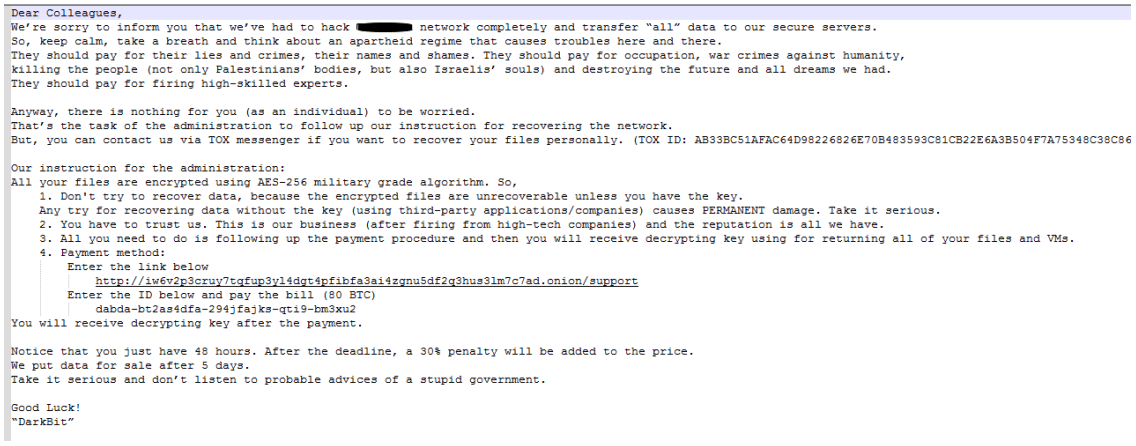
After encrypting the file, the key is stored at the end of the file.



Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Ascii
00002D90	09	2A	92	7D	B4	AE	BD	9D	D9	77	B7	40	86	E8	0F	FD	.*'}´®½ Ùw·@†è¤ý
00002DA0	63	27	A7	0D	21	89	E6	6F	DB	F9	A9	20	49	50	B6	C9	c'§.!‰æoÛù© IP¶É
00002DB0	E9	3E	0C	2B	24	8F	0E	AB	C7	59	FE	CF	1A	EB	7B	2A	é>†+$ ♬«ÇYþÏ→ë{*
00002DC0	B2	2F	93	73	6B	11	15	8A	CD	7D	5B	0E	68	20	A5	4A	²/†sk◄┴†Í}[♬h.¥J
00002DD0	8E	CF	C4	37	97	B9	3B	D8	CE	A4	39	11	24	1D	11	36	†ÏÄ7†¹;ØÎ¤9◄$ ◄6
00002DE0	D2	BF	98	76	DA	C5	6A	2C	04	51	90	53	F2	A1	DA	15	Ò¿†vÚÅj,♩Q SòiÚ┴
00002DF0	A6	66	19	60	36	25	08	25	28	9A	A8	64	FD	B2	02	20	¦f↓`6%◘%(†¨dý²┐.
00002E00	82	2F	E9	33	2E	28	B5	E8	52	FD	A4	D1	BD	69	21	FC	†/é3.(µèRý¤Ñ½i!ü
00002E10	C2	F5	A7	B7	0C	10	15	63	66	5F	7C	82	85	41	B9	8E	Âõ§·†┼┴cf_|†‚…A¹†
00002E20	F0	B8	69	66	E8	B7	16	26	4F	8D	E5	85	41	98	44	91	ð¸ifè·┬&O å†A†D'
00002E30	44	41	52	4B	42	49	54	5F	45	4E	43	52	59	50	54	45	DARKBIT_ENCRYPTE
00002E40	44	5F	46	49	4C	45	53	7C	06	9C	02	5E	AC	87	C8	F8	D_FILES]┼╢ ^¬†Èø
00002E50	35	4B	FA	53	CF	23	11	CB	7B	6A	2D	76	4A	E2	43	2F	5KúSÏ#◄Ë{j-vJâC/
00002E60	EE	B6	29	D0	E4	93	D2	7D	DE	2F	BB	D4	52	64	F8	24	î¶)Ðä†Ò}Þ/»ÔRdø$
00002E70	DA	4E	A4	76	42	DD	5E	88	F4	40	67	21	53	6E	36	F2	ÚN¤vBÝ^†ô@g!Sn6ò
00002E80	06	33	2C	B2	D1	3B	D4	BE	D5	B3	F8	57	6A	DA	BF	D9	◘3,²Ñ;Ô¾Õ³øWjÚ¿Ù
00002E90	C1	4A	F6	73	E6	D0	61	D7	44	41	52	4B	42	49	54	A4	ÁJösæÐa×DARKBIT¤
00002EA0	8B	A4	9A	D4	C1	F2	C1	C3	6D	0C	91	5D	85	EE	E3	D0	‹¤šÔÁòÁÃm◙'…]†îãÐ
00002EB0	B0	DE	31	19	36	0E	64	36	80	E1	1C	24	3C	EC	41	9B	°Þ1↓6♫d6†á ∟$<ìA†
00002EC0	95	2A	16	C5	CE	7B	BD	21	95	33	63	B2	94	E5	3E	F2	•*▬ÅÎ{½!•3c²"å>ò
00002ED0	BE	9D	99	B4	FB	16	09	11	1A	69	13	1C	C9	E0	BE	3F	¾ †´ù▬ ◙◄→i‼ Éà¾?
00002EE0	68	76	3F	52	38	E5	54	1A	8E	4E	E1	FB	82	C5	59	E1	hv?R8åT→†Náû‚ÅYá
00002EF0	42	82	4A	FA	B8	66	2A	C9	08	DC	D1	D0	0D	27	08	C3	B‚Jú¸f*É◘ÜÑÐ♪'◘Ã
00002F00	5C	50	A5	55	B6	79	AE	15	3D	4E	C5	F6	33	F4	28	CC	\P¥U¶y®§=NÅö3ô(Ì
00002F10	E8	B8	D7	C0	E9	92	14	0E	DF	F8	B1	D7	B4	58	EE	87	è¸×Àé'¶♫ßø±×´Xî‡
00002F20	0A	83	61	9E	2C	97	B0	9B	65	7E	01	74	D2	AD	1A	1C	◙ƒaž,—°›e~☺tÒ-→∟
00002F30	78	07	85	A0	A1	9B	F6	46	94	48	EF	96	8F	3E	92	82	x•† …† ¡›öF”Hï– >'‚
00002F40	F8	35	5E	35	24	71	65	FC	75	3D	B6	CC	3D	80	E2	8D	ø5^5$qeüu=¶Ì=€â◙
00002F50	12	1A	BA	56	56	D2	05	D5	57	A7	43	DF	CF	BB	FA	A6	↕→ºVVÒ♣ÕW§CßÏ»ú¦
00002F60	C9	70	D6	4D	A7	F9	D5	3C	56	64	CE	C1	FA	5A	13	E9	ÉpÖM§ùÕ<VdÎÁúZ‼é
00002F70	68	DE	12	F9	37	22	03	5A	F0	FD	8A	4D	02	0C	4C	DC	hÞ↕ù7"♥Zðý Š♫ ☙LÜ
00002F80	E2	3F	C0	C3	2A	A6	CF	30	E1	45	C6	90	5E	57	AE	09	â?ÀÃ*¦ÏOáEÆ ^W®○
00002F90	EC	42	E9	6B	EB	1B	18	07	5B	00	86	1B	5A	F5	ED		ìBékë←↑•[.†←Zõí

Figure 16 – Ransomware key

The encrypted files are given '.darkbit' extension and also a ransom note is dropped in the respective folders.



```
Dear Colleagues,
We're sorry to inform you that we've had to hack [redacted] network completely and transfer "all" data to our secure servers.
So, keep calm, take a breath and think about an apartheid regime that causes troubles here and there.
They should pay for their lies and crimes, their names and shames. They should pay for occupation, war crimes against humanity,
killing the people (not only Palestinians' bodies, but also Israelis' souls) and destroying the future and all dreams we had.
They should pay for firing high-skilled experts.

Anyway, there is nothing for you (as an individual) to be worried.
That's the task of the administration to follow up our instruction for recovering the network.
But, you can contact us via TOX messenger if you want to recover your files personally. (TOX ID: AB33BC51AFAC64D98226826E70B483593C81CB22E6A3B504F7A75348C38C86

Our instruction for the administration:
All your files are encrypted using AES-256 military grade algorithm. So,
    1. Don't try to recover data, because the encrypted files are unrecoverable unless you have the key.
    Any try for recovering data without the key (using third-party applications/companies) causes PERMANENT damage. Take it serious.
    2. You have to trust us. This is our business (after firing from high-tech companies) and the reputation is all we have.
    3. All you need to do is following up the payment procedure and then you will receive decrypting key using for returning all of your files and VMs.
    4. Payment method:
        Enter the link below
            http://iw6v2p3cruy7tqfup3yl4dgt4pfibfa3ai4zgnu5df2q3hus3lm7c7ad.onion/support
        Enter the ID below and pay the bill (80 BTC)
            dabda-bt2as4dfa-294jfajks-qti9-bm3xu2
You will receive decrypting key after the payment.

Notice that you just have 48 hours. After the deadline, a 30% penalty will be added to the price.
We put data for sale after 5 days.
Take it serious and don't listen to probable advices of a stupid government.

Good Luck!
"DarkBit"
```

Figure 17 – Ransomware Note

Tweet from a self-identifying DarkBit twitter handle, associating itself with [MuddyWater]

Figure 18 – Tweet on DarkBit

MuddyWaters uses different types of attacks for initial access like phishing email campaigns, using tools like MimiKatz to break into the system, etc. In this case, MuddyWaters made use of Cobalt Strike to get initial access into the system. From the ransomware note, we figured out that it was a politically motivated attack.

We at K7 Labs provide detection for DarkBit ransomware and all the latest threats. Users are advised to use a reliable security product such as “K7 Total Security” and keep it up-to-date to safeguard their devices.

Indicators of Compromise (IOCs)

File Name	Hash	Detection Name
DarkBit.exe	9880FAE6551D1E9EE921F39751A6F3C0	Trojan (0058e3dd1)
hr-update.iso	1219A8880DEBDD10D081195E27A2A016	Trojan (0001140e1)

References

https://attack.mitre.org/groups/G0069/
