

BiBi Wiper: A Malware Analysis Amidst the Israel-Hamas-ISIS Conflict

 idanmalihi.com/bibi-wiper-a-malware-analysis-amidst-the-israel-hamas-isis-conflict/

June 3, 2024



Executive Summary

On October 7th, 2023, a war broke out between Israel and Hamas-ISIS. During the war, a pro-Hamas hacker group attacked Israeli companies with the 'BiBi' wiper, corrupting and disrupting their operating systems without demanding a ransom and stealing files or information from the systems.

Malware known as "Wiper" is designed to erase and corrupt data on targeted systems. Unlike other types of malware that aim to steal or spy on data, wiper malware is specifically created to cause damage, often making the affected data or system impossible to recover.

In this report, I will analyze the BiBi-Windows-based wiper and examine its characteristics, attack vectors, and methods for detecting it.

Technical Details

File Name: bibi.exe

File Type: Executable

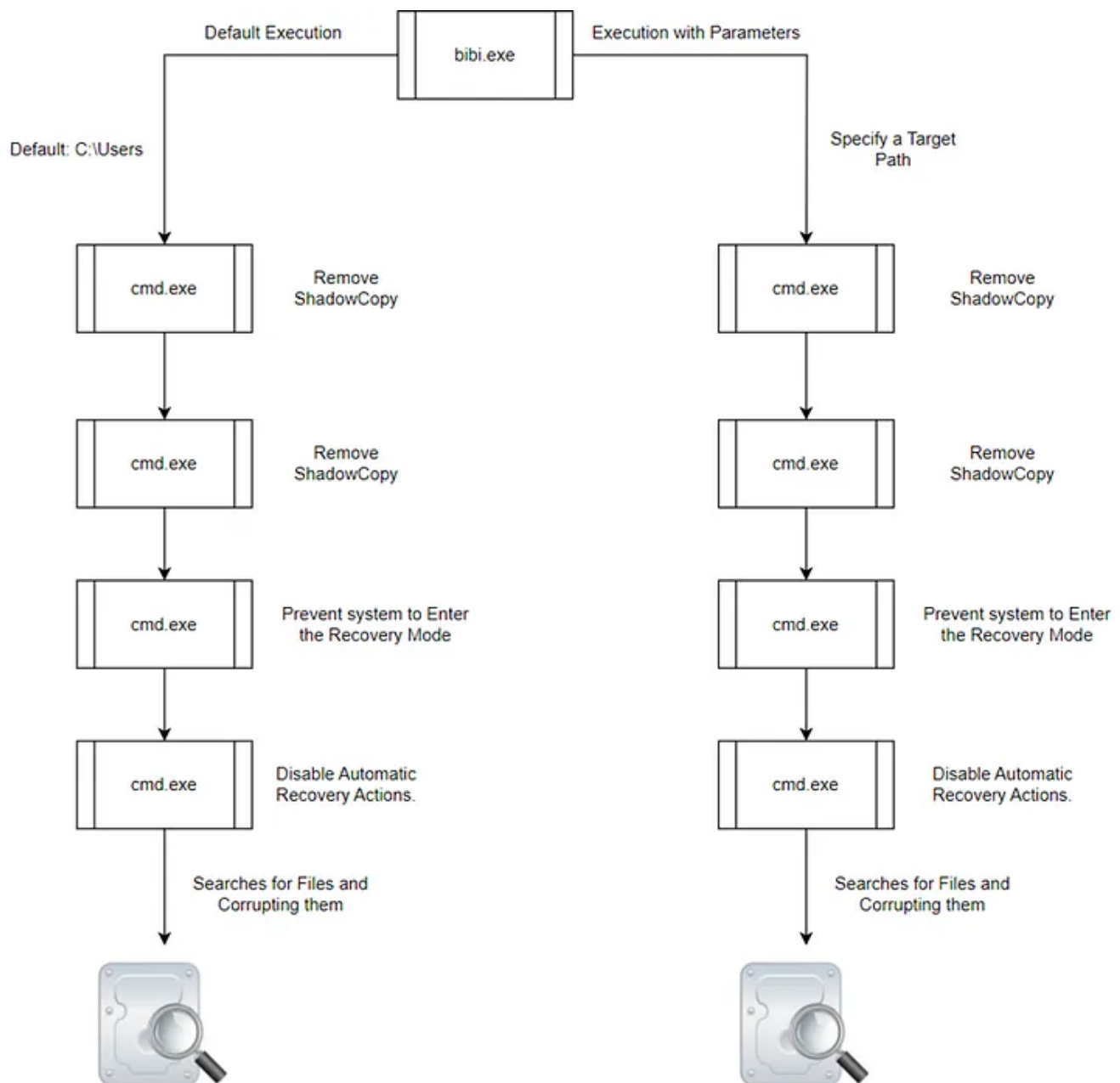
Architecture: PE32+ (64-bit)

Size: 203KB

MD5: e26bba0304f14ef96beb60376791d32c

SHA256: 40417e937cd244b2f928150cae6fa0eff5551fdb401ea072f6ecdda67a747e17

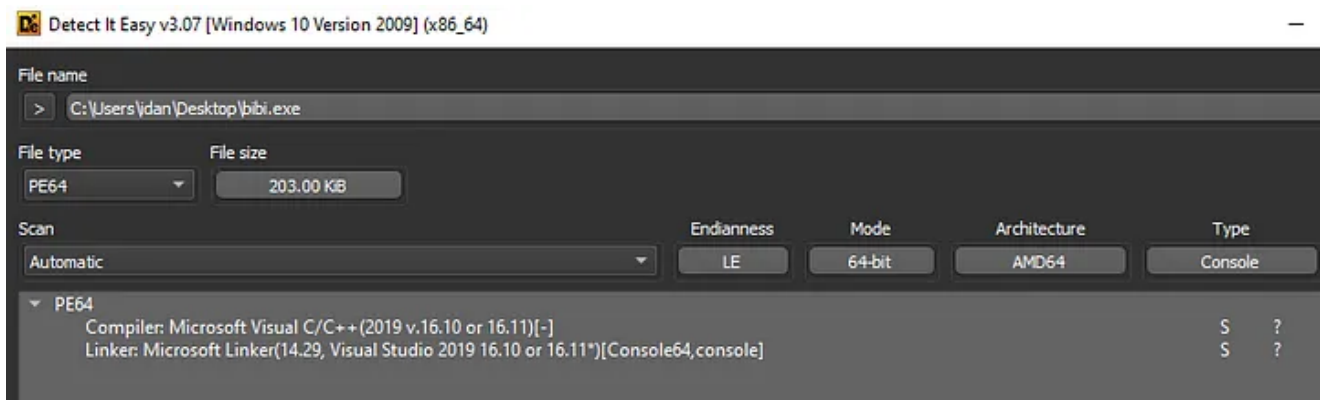
Wiper Execution Diagram



Static Analysis

In this analysis section, I will analyze and provide valuable insights into the characteristics and behavior of the wiper without executing the code.

The wiper was compiled with Microsoft Visual Studio and was likely coded in either C or C++ programming languages. Additionally, the wiper is a 64-bit executable.



Compilation Time

The wiper was compiled on October 22nd, 2023.

```
Compile date:          2023-10-22 00:24:41
Debug:
  Date.Debug.Pogo:     2023-10-22 00:24:41
  Date.Debug.Iltcg:    2023-10-22 00:24:41
```

Strings

The following strings seem to be related to the C++ programming language. These strings are often associated with the internals of C++ classes and their runtime representations.

```
'Type Descriptor'
'Base Class Descriptor at (
'Base Class Array'
'Class Hierarchy Descriptor'
'Complete Object Locator'
```

The wiper collects information about the date and time of the system.

Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
January
February
March
April
June
July
August
September
October
November
December
MM/dd/yy
dddd, MMMM dd, yyyy
HH:mm:ss

The wiper execution produces system information related to system paths, processor cores, threads, rounds, and stats.

```
[+] Stats: %d | %d
[!] Waiting For Queue
[+] Round %d
```

```
[+] Path: %s
[+] CPU cores: %d, Threads: %d
```

During its operation, the wiper carries out four commands in the operating system.

The malware author executes specific commands to remove shadow copies, which can make it difficult to restore data from backups.

```
cmd.exe /c vssadmin delete shadows /quiet /all
cmd.exe /c wmic shadowcopy delete
```

The malware author executes the following command to prevent the system from entering recovery mode.

```
cmd.exe /c bcdedit /set {default} bootstatuspolicy ignoreallfailures
```

The malware author then executes a particular command in order to disable any automatic recovery actions.

```
cmd.exe /c bcdedit /set {default} recoveryenabled no
```

```
lla/ teIuq/ swodahs eteled nimdassv c/ exe.dmc
eteled ypocwodahs cimw c/ exe.dmc
seruliafllaerongi ycilopsutatstoob }tluafed{ tes / tidedcb c / exe.dmc|
on delbaneyrevocer }tluafed{ tes/ tidedcb c/ exe.dmc
```

While executing the wiper, it renames the files to .BiBi extension and disrupts the file system.

.BiBi

The following string appears that it could be indicative of spyware or a keylogger.

```
0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
```

Wiper DLL Usage

The wiper runs a DLL file during execution:

KERNEL32.dll	99	0002F9C0	00000000	00000000	0002FD80	00022000
--------------	----	----------	----------	----------	----------	----------

Malware Resources

The malware author uses a legitimate Microsoft manifest file to evade detection by security products.

```
<?xml version='1.0' encoding='UTF-8' standalone='yes'?>
<assembly xmlns='urn:schemas-microsoft-com:asm.v1' manifestVersion='1.0'>
  <trustInfo xmlns="urn:schemas-microsoft-com:asm.v3">
    <security>
      <requestedPrivileges>
        <requestedExecutionLevel level='asInvoker' uiAccess='false' />
      </requestedPrivileges>
    </security>
  </trustInfo>
</assembly>
```

Dissecting the Wiper

In its initial execution, the malware collects information about the system's disk drives using the 'GetLogicalDrives' API function.

Malware uses this function to detect disk drives on systems and attempt to spread itself across these drives.

```
loc_140005C37:
call    cs:GetLogicalDrives
mov     dword ptr [rbp+57h+var_78], eax
mov     ebx, 1Ah
```

After gathering information about the disk drives on the system, the malware uses the 'GetDriveTypeA' API function to determine the type of each drive, such as removable, CD-ROM, network, etc.

```
lea     rcx, [rbp+57h+lpRootPathName]
cmp     qword ptr [rbp+57h+var_88+8], 10h
cmovnb  rcx, [rbp+57h+lpRootPathName] ; lpRootPathName
call    cs:GetDriveTypeA
```

The malware uses the 'GetNativeSystemInfo' API function to determine the processor architecture of the system and whether it is a 32-bit or 64-bit processor.

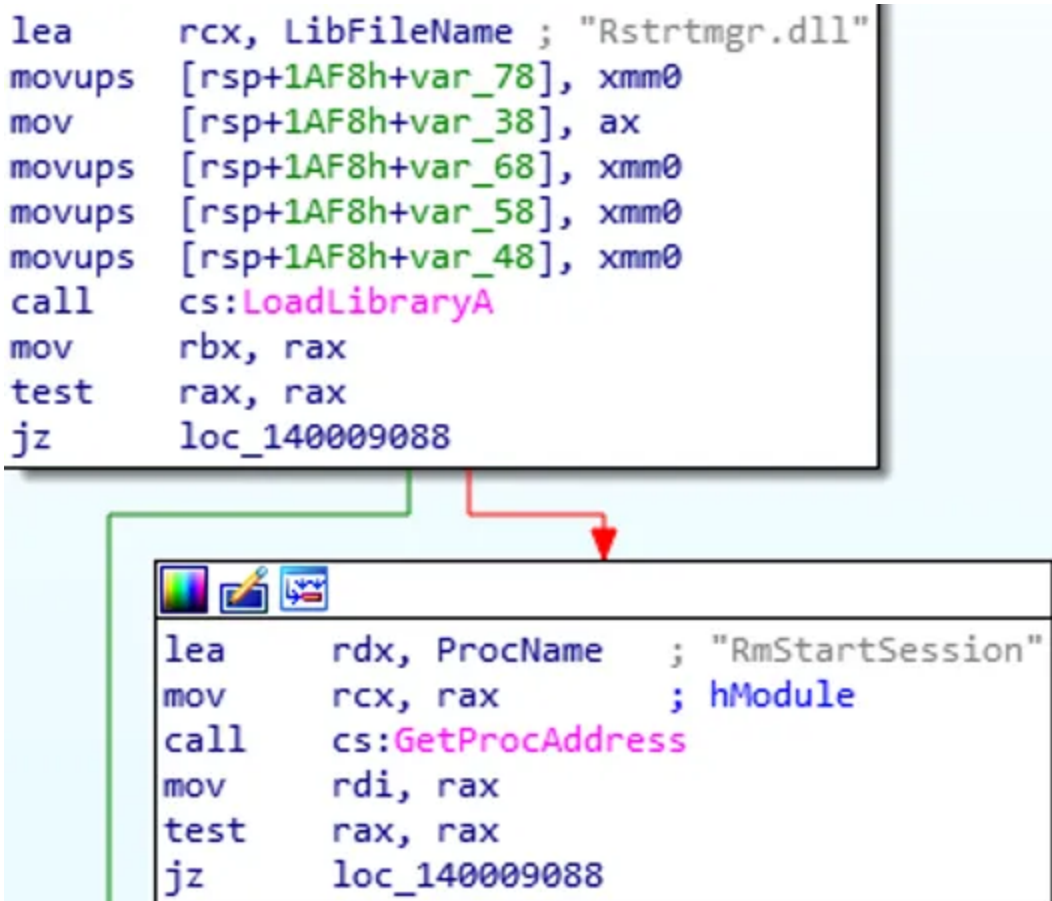
Then, the wiper takes the number of processors of the system, moves the value to the eax register, and displays the value in the command prompt in the initial execution.

```

sub     rsp, 58h
lea     rcx, [rsp+58h+SystemInfo] ; lpSystemInfo
call    cs:GetNativeSystemInfo
mov     eax, [rsp+58h+SystemInfo.dwNumberOfProcessors]
add     rsp, 58h
retn
sub_1400090B8 endp

```

The malware loads the rstrtmgr.dll DLL file using the LoadLibraryA API function. If the DLL loads successfully, the GetProcAddress function will get the DLL's RmStartSession address.



The screenshot below demonstrates how the operation in Figure 15 is executed dynamically:

#	Time of Day	Thread	Module	API
50	10:17:21.878 AM	3	bibi.exe	LoadLibraryA ("Rstrtmgr.dll")
51	10:17:21.878 AM	3	bibi.exe	GetProcAddress (0x00007ffc70e0000, "RmStartSession")

When the malware finishes gathering information, it corrupts files in the C:\Users directory and collects CPU and thread data from the system.

A threat actor can execute the BiBi wiper by double-clicking on the executable, and by default, the executable will start corrupting the files inside of the C:\Users path.

On the other hand, a threat actor can execute the wiper via the command line and specify a specific path.

0000000140005812	48:8BCB	mov rcx,rbx	rbx:&L"C:\Users"
0000000140005815	E8 76100000	call bibi.140006B90	
000000014000581A	48:8D55 87	lea rdx,qword ptr ss:[rbp-79]	
000000014000581E	48:837D 9F 10	cmp qword ptr ss:[rbp-61],10	
0000000140005823	48:0F4355 87	cmovae rdx,qword ptr ss:[rbp-79]	
RIP 0000000140005828	48:8D0D E16502	lea rcx,qword ptr ds:[14002C110]	000000014002C110: "[+] Path: %s\n"
0000000140005DEA	45:88C6	mov r8d,r14d	
0000000140005DED	41:8AD1	mov edx,r9d	
RIP 0000000140005DF0	48:8D0D 296302	lea rcx,qword ptr ds:[14002C120]	000000014002C120: "[+] CPU cores: %d, Threads: %d\n"
0000000140005DF7	E8 34B2FFFF	call bibi.140001030	
0000000140005DFC	48:C745 DF 881	mov qword ptr ss:[rbp-21],1388	
0000000140005E04	48:8D4D DF	lea rcx,qword ptr ss:[rbp-21]	

 C:\Users\idan\Desktop\bibi.exe

```
[+] Path: C:\Users
[+] CPU cores: 4, Threads: 12
```

Wiper execution via the command line and specifying a specific path:

```
PS C:\Users\idan\Desktop> .\bibi.exe 'C:\Users\idan\Desktop\New folder'
[+] Path: C:\Users\idan\Desktop\New folder
[+] CPU cores: 4, Threads: 12
```

	1KDF7FNx2A.BiBi5602	11/21/2023 11:56 AM	BIBI5602 File
	CUH67bRfrM.BiBi5603	11/21/2023 11:56 AM	BIBI5603 File
	GgLDO9JhZy.BiBi5603	11/21/2023 11:56 AM	BIBI5603 File
	kBskslJOza.BiBi5603	11/21/2023 11:56 AM	BIBI5603 File
	qWKwdWYG9n.BiBi5603	11/21/2023 11:56 AM	BIBI5603 File
	r9IGsxBsge.BiBi5603	11/21/2023 11:56 AM	BIBI5603 File
	Rw1R7yg5Us.BiBi5603	11/21/2023 11:56 AM	BIBI5603 File
	RwPvl5hEQR.BiBi5603	11/21/2023 11:56 AM	BIBI5603 File
	woL6VZDv42.BiBi5603	11/21/2023 11:56 AM	BIBI5603 File

The malware uses the 'Sleep' API function to delay thread execution. This function is often employed for time-based evasion by adding delays in the code.

```

loc_140009167:
lea     rdx, [rsp+48h+var_28]
mov     rcx, rbx
call    sub_1400098D0
mov     ecx, eax          ; dwMilliseconds
call    cs:Sleep
mov     edx, 1
lea     rcx, [rsp+48h+var_28]
call    sub_1400099D4
mov     rax, [rsp+48h+var_28]
cmp     rax, [rbx]
jl      short loc_140009167

```

Using the 'CreateProcessA' API function, four command prompts are opened to execute specific commands.

cmd.exe /c vssadmin delete shadows /quiet /all

```

mov     [rsp+8+arg_0], rbx
push    rbp
lea     rbp, [rsp-60h]
sub     rsp, 160h
mov     rax, cs:__security_cookie
xor     rax, rsp
mov     [rbp+60h+var_10], rax
xor     ebx, ebx
mov     [rsp+160h+var_F8], 0Fh
lea     rdx, allaTeiuqSwodah ; "lla/ teIuq/ swodahs  eteled nimdassv ..."
mov     [rsp+160h+var_110], rbx
lea     rcx, [rsp+160h+var_110]
mov     [rsp+160h+var_100], rbx
lea     r8d, [rbx+31h]
call    sub_140006990

```

allaTeiuqSwodah db 'lla/ teIuq/ swodahs eteled nimdassv c/ exe.dmc',0
; DATA XREF: sub_140005530+2B↑to

cmd.exe /c wmic shadowcopy delete

```

loc_14000566F:
mov     r8d, 21h
mov     [rsp+160h+var_110], rbx
lea     rdx, aEteledYpocwoda ; "eteled ypocwodahs cimw c/ exe.dmc"
mov     [rsp+160h+var_100], rbx
lea     rcx, [rsp+160h+var_110]
mov     [rsp+160h+var_F8], 0Fh
call    sub_140006990
cmp     [rsp+160h+var_F8], 10h
lea     rcx, [rsp+160h+var_110]
mov     rdx, [rsp+160h+var_100]
cmovnb  rcx, [rsp+160h+var_110]
add     rdx, rcx
lea     rcx, [rsp+160h+var_110]
cmp     [rsp+160h+var_F8], 10h

```

aEteledYpocwoda db 'eteled ypocwodahs cimw c/ exe.dmc',0
; DATA XREF: sub_140005530+14A↑to

cmd.exe /c bcdedit /set {default} bootstatuspolicy ignoreallfailures

```

loc_1400578E:
mov     r8d, 46h
mov     [rsp+160h+var_110], rbx
lea     rdx, aSeruliafllaero ; "seruliafllaerongi ycilopsutatstoob }tlu"...
mov     [rsp+160h+var_100], rbx
lea     rcx, [rsp+160h+var_110]
mov     [rsp+160h+var_F8], 0Fh
call    sub_140006990
cmp     [rsp+160h+var_F8], 10h
lea     rcx, [rsp+160h+var_110]
mov     rdx, [rsp+160h+var_100]
cmovnb rcx, [rsp+160h+var_110]
add     rdx, rcx
lea     rcx, [rsp+160h+var_110]
cmp     [rsp+160h+var_F8], 10h
cmovnb rcx, [rsp+160h+var_110]
call    sub_140009E34
cmp     [rsp+160h+var_F8], 10h
lea     rcx, [rsp+160h+var_110]
lea     rdx, [rbp+60h+CommandLine]
cmovnb rcx, [rsp+160h+var_110]
sub     rdx, rcx
nop

```

```

aSeruliafllaero db 'seruliafllaerongi ycilopsutatstoob }tluafed{ tes / tidedcb c / ex'
; DATA XREF: sub_140005530+269fo
db 'e.dmc',0

```

cmd.exe /c bcdedit /set {default} recoveryenabled no

```

loc_140058AE:
mov     r8d, 34h
mov     [rsp+160h+var_110], rbx
lea     rdx, aOnDelbaneyrevo ; "on delbaneyrevocer }tluafed{ tes/ tided"...
mov     [rsp+160h+var_100], rbx
lea     rcx, [rsp+160h+var_110]
mov     [rsp+160h+var_F8], 0Fh
call    sub_140006990
cmp     [rsp+160h+var_F8], 10h
lea     rcx, [rsp+160h+var_110]
mov     rdx, [rsp+160h+var_100]
cmovnb rcx, [rsp+160h+var_110]
add     rdx, rcx
lea     rcx, [rsp+160h+var_110]
cmp     [rsp+160h+var_F8], 10h
cmovnb rcx, [rsp+160h+var_110]
call    sub_140009E34
cmp     [rsp+160h+var_F8], 10h
lea     rcx, [rsp+160h+var_110]
lea     rdx, [rbp+60h+CommandLine]
cmovnb rcx, [rsp+160h+var_110]
sub     rdx, rcx
nop

```

```

aOnDelbaneyrevo db 'on delbaneyrevocer }tluafed{ tes/ tidedcb c/ exe.dmc',0
; DATA XREF: sub_140005530+389fo

```

```

lea     rax, [rbp+60h+ProcessInformation]
xor     r9d, r9d          ; lpThreadAttributes
mov     [rsp+160h+lpProcessInformation], rax ; lpProcessInformation
lea     rdx, [rbp+60h+CommandLine] ; lpCommandLine
lea     rax, [rsp+160h+StartupInfo]
xor     r8d, r8d          ; lpProcessAttributes
mov     [rsp+160h+lpStartupInfo], rax ; lpStartupInfo
xor     ecx, ecx          ; lpApplicationName
mov     [rsp+160h+lpCurrentDirectory], rbx ; lpCurrentDirectory
mov     [rsp+160h+lpEnvironment], rbx ; lpEnvironment
movups  xmmword ptr [rbp+60h+StartupInfo.dwFillAttribute], xmm0
mov     [rsp+160h+dwCreationFlags], 8000001h ; dwCreationFlags
mov     [rsp+160h+bInheritHandles], ebx ; bInheritHandles
movups  xmmword ptr [rsp+160h+StartupInfo.lpReserved], xmm0
mov     [rbp+60h+StartupInfo.wShowWindow], bx
movups  xmmword ptr [rbp+60h+StartupInfo.lpTitle], xmm0
movups  xmmword ptr [rbp+60h+StartupInfo.dwXSize], xmm0
movups  xmmword ptr [rbp+60h+StartupInfo.lpReserved2], xmm0
movups  xmmword ptr [rbp+60h+StartupInfo.hStdOutput], xmm0
call    cs:CreateProcessA

```

The following commands executed can be viewed from the created cmd processes.

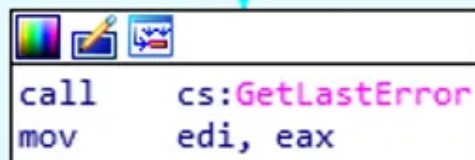
bibl.exe		< 0.01	2,484 K	5,512 K	10368		
cs%	conhost.exe		7,036 K	17,228 K	9496	Console Window Host	Microsoft Corporation
cs%	cmd.exe		1,532 K	1,760 K	2780	Windows Command Processor	Microsoft Corporation
cs%	cmd.exe		1,536 K	1,760 K	10088	Windows Command Processor	Microsoft Corporation
cs%	cmd.exe		1,532 K	1,760 K	7320	Windows Command Processor	Microsoft Corporation
cs%	cmd.exe	Susp...	1,524 K	1,748 K	10252	Windows Command Processor	Microsoft Corporation
Process Name	PID	Condition	Path	Result	Detail		
bibl.exe	10368	Process Create	C:\Windows\SYSTEM32\conhost.exe	SUCCESS	PID: 2780, Command line: cmd.exe /c wmic diskdrive delete, shadowrun /q /r /all		
bibl.exe	10368	Process Create	C:\Windows\SYSTEM32\cmd.exe	SUCCESS	PID: 10088, Command line: cmd.exe /c wmic shadowcopy delete		
bibl.exe	10368	Process Create	C:\Windows\SYSTEM32\cmd.exe	SUCCESS	PID: 7320, Command line: cmd.exe /c brdelt /set (default) bootman,spkkey,spnrepl,wine		
bibl.exe	10368	Process Create	C:\Windows\SYSTEM32\cmd.exe	SUCCESS	PID: 10252, Command line: cmd.exe /c bootdel /set (default) recovery,atd no		

After the execution of the commands, the wiper malware begins the corruption action in the file system. The malware begins by utilizing the 'CreateFileW' API function to open files. It sets the 'dwCreationDisposition' parameter to the value of '3', which corresponds to 'OPEN_EXISTING.' This action allows the malware to determine whether the file exists in the file system or not.

```

mov     [rsp+48h+hTemplateFile], rdi ; hTemplateFile
mov     edx, eax                     ; dwDesiredAccess
mov     [rsp+48h+dwFlagsAndAttributes], r9d ; dwFlagsAndAttributes
mov     rcx, r10                     ; lpFileName
xor     r9d, r9d                     ; lpSecurityAttributes
mov     [rsp+48h+dwCreationDisposition], 3 ; dwCreationDisposition
lea     r8d, [rdi+7]                 ; dwShareMode
call    cs:CreateFileW
mov     [rbx], rax
cmp     rax, 0FFFFFFFFFFFFFFFFh
jnz     short loc_140009852

```



```

call    cs:GetLastError
mov     edi, eax

```

The malware uses the 'WriteFile' API function to rewrite data to files in the file system. It handles the files using the 'hFile' handle. If the write operation fails, the malware takes action at the 'loc_140017C5E' memory location. The data written in the 'lpBuffer' pointer is what the malware writes when it handles a file.

```

mov     rcx, [rbp+57h+hFile] ; hFile
lea     r9, [rsp+130h+NumberOfBytesWritten] ; lpNumberOfBytesWritten
mov     r8d, eax                 ; nNumberOfBytesToWrite
mov     [rsp+130h+lpOverlapped], rbx ; lpOverlapped
lea     rdx, [rbp+57h+Buffer] ; lpBuffer
call    cs:WriteFile
xor     r11d, r11d
test    eax, eax
jz      loc_140017C5E

```

The Wiper malware uses the 'FindFirstFileExA' API function to search for files, directories, and sub-directories. The starting point for the search is the path that the threat actor specified during the execution of the Wiper, or it defaults to the 'C:\Users' path inside the 'lpFileName' pointer.

```

loc_1400094EF:
and     [rsp+38h+var_10], 0
xor     r9d, r9d          ; fSearchOp
and     [rsp+38h+var_18], 0
mov     r8, rdi           ; lpFindFileData
mov     rcx, rsi          ; lpFileName
lea     edx, [r9+1]       ; fInfoLevelId
call    cs:FindFirstFileExW
mov     [rbx], rax
cmp     rax, 0FFFFFFFFFFFFFFFFh
jnz     short loc_140009553

```

Then, the malware continues the search progress with the 'FindNextFileW' function.

```

sub_140009484 proc near
sub     rsp, 28h
call    cs:FindNextFileW
test    eax, eax
jz      short loc_140009499

```

During the corruption of the files in the file system, the wiper changes their extension to '.BiBi1'.

```

.C:\Users\idan\AppData\Local\Google\Chrome\User Data\Autofill States\2020.11.2.164946\e2IjVFEQG.BiBi1
.C:\Users\idan\AppData\Local\Google\Chrome\User Data\Autofill States\2020.11.2.164946\eJ57jLQiUM.BiBi1
.C:\Users\idan\AppData\Local\Google\Chrome\User Data\Autofill States\2020.11.2.164946\EndvADzIbf.BiBi1
.C:\Users\idan\AppData\Local\Google\Chrome\User Data\Autofill States\2020.11.2.164946\FSi5PQjTwN.BiBi1
.C:\Users\idan\AppData\Local\Google\Chrome\User Data\Autofill States\2020.11.2.164946\gV5KCWqj2s.BiBi1
.C:\Users\idan\AppData\Local\Google\Chrome\User Data\Autofill States\2020.11.2.164946\gYVRP1Ms6M.BiBi1
.C:\Users\idan\AppData\Local\Google\Chrome\User Data\Autofill States\2020.11.2.164946\gylmT2OVYA.BiBi1
.C:\Users\idan\AppData\Local\Google\Chrome\User Data\Autofill States\2020.11.2.164946\H7HKK5pIZ0.BiBi1
.C:\Users\idan\AppData\Local\Google\Chrome\User Data\Autofill States\2020.11.2.164946\HpQfYfXlqv.BiBi1
.C:\Users\idan\AppData\Local\Google\Chrome\User Data\Autofill States\2020.11.2.164946\hR7HfwWIHo.BiBi1
.C:\Users\idan\AppData\Local\Google\Chrome\User Data\Autofill States\2020.11.2.164946\HVBzZB8hZS.BiBi1
.C:\Users\idan\AppData\Local\Google\Chrome\User Data\Autofill States\2020.11.2.164946\iCe3LBINf.BiBi1
.C:\Users\idan\AppData\Local\Google\Chrome\User Data\Autofill States\2020.11.2.164946\jasiJ2q3pV.BiBi1
.C:\Users\idan\AppData\Local\Google\Chrome\User Data\Autofill States\2020.11.2.164946\JAuzbcQqR5.BiBi1
.C:\Users\idan\AppData\Local\Google\Chrome\User Data\Autofill States\2020.11.2.164946\jBYi82XM8o.BiBi1
.C:\Users\idan\AppData\Local\Google\Chrome\User Data\Autofill States\2020.11.2.164946\jZZJP9g8N9.BiBi1

```

Conclusion

A pro-Hamas hacker group attacked Israeli companies during the war between Israel and Hamas-ISIS.

The malware's purpose was to destroy and corrupt the files in the operating system of the target without demanding any ransom.

The Wiper was moving between files in the file system of the target and writing gibberish content to corrupt the files, change their name to a random name, and extension to 'BiBi' to know the files were corrupted and unrecoverable.

The behind-the-scenes wiper attack is more political and less technical because the malware is very basic, poorly written, and very easy to detect. The pro-Hamas hacker group only turned to Israeli companies because of the war. This can be understood with the name of the malware and the name of the extension BiBi (Prime Minister of Israel).

Indicator of Compromise (IoCs)

1. BiBi.exe
2. e26bba0304f14ef96beb60376791d32c
3. [Random].BiBi[Random]

MITRE ATT&CK

Execution	Defense Evasion	Discovery	Impact
Command and Scripting Interpreter	Masquerading	System Information Discovery	Data Destruction
	File Deletion	File and Directory Discovery	Inhibit System Recovery
	Virtualization/Sandbox Evasion		

Yara Rule

```
rule BiBi_wiper
{
  meta:
    author = "Idan Malihi, CyFox XDR"
    created = "21/11/2023"
    description = "YARA Rule for BiBi Wiper Detection"
    MD5 = "72e3a0160cbbd3d8825694176c5d4333"

  strings:
    $mz = { 4D 5A }
    $command1 = { 73 65 72 75 6C 69 61 66 6C 6C 61 65 72 6F 6E 67 69 20 79 63 69 6C 6F 70 73 75 74 61 74 73 74 6F 6F 62 20 7D 74 6C 75 61 66 65 64 78 20 74 65 73 20 2F 20 74 69 64 65 64 63 62 20 63 20 2F 20 65 78 65 2E 64 6D 63 }
    $command2 = { 6F 6E 20 64 65 6C 62 61 6E 65 79 72 65 76 6F 63 65 72 20 7D 74 6C 75 61 66 65 64 78 20 74 65 73 2F 20 74 69 64 65 64 63 62 20 63 2F 20 65 78 65 2E 64 6D 63 }
    $command3 = { 6C 6C 61 2F 20 74 65 49 75 71 2F 20 73 77 6F 64 61 68 73 20 20 20 65 74 65 6C 65 64 20 6E 69 6D 64 61 73 73 76 20 20 63 2F 20 65 78 65 2E 64 6D 63 }
    $BiBi = { 2E 00 42 00 69 00 42 00 69 00 }
    $Stats_String = { 5B 28 5D 20 53 74 61 74 73 3A 20 25 64 20 7C 20 25 64 }
    $Queue_String = { 5B 21 5D 20 57 61 69 74 69 6E 67 20 46 6F 72 20 51 75 65 75 65 }
    $Round_String = { 5B 28 5D 20 52 6F 75 6E 64 20 25 64 }

  condition:
    all of them
}
```

Detection

```
λ yara64.exe BiBi.yara malware\  
BiBi_Wiper malware\\bibi.exe
```

Keep Posted with the Latest Research Articles
