Malware Targeting Israelis during "Iron Swords" war

Contents

Executive Summary

During the Iron Swords conflict between Israel and Hamas, the Israel National Cyber Directorate (INCD) received complaints regarding a malicious email campaign targeting Israeli citizens. The attackers pretended to be the INCD while sending emails from various trusted domains. On February 11, between 13:00 and 18:00 (IST), multiple complaints were reported to the INCD regarding these deceptive emails. Additionally, after receiving the complaints, the INCD issued warnings about the malicious email campaign.

Figure 1 - Image showing the warning issued by the INCD in response to the malicious email campaign.

The extent of the damage caused by these emails is currently unknown. However, several news outlets, including 0404 or Hamal, mentioned the attack.

Further investigation into the incident revealed the operational methods of the malware involved, enabling the creation of a Yara Rule for detection purposes.

In a nutshell

1. Phishing email campaign: The malware was distributed via emails from trusted domains, disguising itself as a software update, a tactic aimed at deceiving users into downloading and executing it.
2. Loader functionality: The malware functions as a loader and requires the presence of Hebrew language settings to execute its payload, suggesting a targeted approach towards Israeli citizens.
3. Dual malware drops: Upon execution, the malware drops two distinct payloads if Hebrew language settings are detected.
4. Active Directory targeting: The first payload is deployed specifically if the user is within an Active Directory (AD) environment. It targets other computers within the AD using LDAP, enabling lateral movement and broader compromise.
5. Wiper functionality: The second payload serves as a wiper, indiscriminately corrupting files on infected systems, potentially causing significant damage without specific targeting.

Malicious Loader Analysis

The malware spread through an attachment, embedded in deceptive emails targeted at Israeli entities and linked directly to the ongoing "Iron Swords" war. It disguised itself as a "Software Update" to trick users into downloading and running it. This analysis examines the Windows version of the malware, one of few variations deployed. The binary lacks obfuscation, packing, or any protective measures, and it contains several strings that could provide valuable information to analysts right from the beginning of the static analysis.

The malware is flagged by 53 anti-viruses. Most of them flag the malware as Torjan.

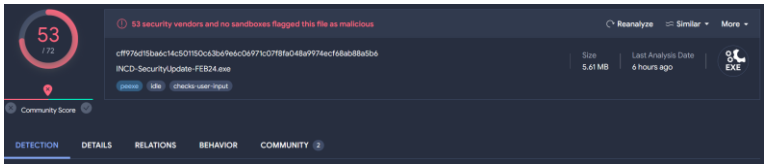

Figure 2 - Screenshot from VirusTotal; Detections tab.

We begin by loading the binary into CFF Explorer. Upon inspecting the stamp date of the binary, an intriguing discovery was made: the date of stamp corresponds to October 7th, coinciding with the commencement of the war.

INCD-SecurityUpdate-FEB24.exe

Member	Offset	Size	Value	Meaning
Machine	00000104	Word	014C	Intel 386
NumberOfSections	00000106	Word	0005	
TimeDateStamp	00000108	Dword	65215D8F	
PointerToSymbolTa...	0000010C	Dword	00000000	
NumberOfSymbols	00000110	Dword	00000000	
SizeOfOptionalHea...	00000114	Word	00E0	
Characteristics	00000116	Word	0102	Click here

Figure 3 - Screenshot from CFF Explorer displaying the date of stamp in Unix time format.

Further examination of the binary revealed notable section sizes, particularly in the .data and .rsrc sections.

INCD-SecurityUpdate-FEB24.exe

Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations N...	Linenumbers ...	Characteristics
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word	Word	Dword
.text	0000FF93	00001000	00010000	00000400	00000000	00000000	0000	0000	60000020
.rdata	0000657E	00011000	00006600	00010400	00000000	00000000	0000	0000	40000040
.data	00057CC4	00018000	00057200	00016A00	00000000	00000000	0000	0000	C0000040
.rsrc	0052CE40	00070000	0052D000	0006DC00	00000000	00000000	0000	0000	40000040
.reloc	000010A8	0059D000	00001200	0059AC00	00000000	00000000	0000	0000	42000040

Figure 4 - Screenshot from CFF Explorer highlighting the sections and their respective sizes.

Using *strings* tool, I found strings within the malware that explicitly mention .NET. However, when using a *file* tool on the malware, it appears that the malware is not .NET. This discrepancy could suggest that the file is a loader or downloader. Additionally, it's worth noting that this malicious loader is a console application.

The malware's strings mention access to anything related to keyboard layout, as well as references to video or media in general:

```
C:\Users\Public\Microsoft Connection Agent.jpg
C:\Users\Public\Video.mp4
C:\Users\Public\Microsoft System Agent.exe
Microsoft System Manager.exe
C:\Users\Public\Microsoft System Manager.exe
Windows Defender Agent.exe
C:\Users\Public\Windows Defender Agent.exe
KERNEL32.DLL
CreateProcessA
Keyboard Layout\Preload
0000040d
System\Keyboard Layout\Preload
```

Figure 5 – Strange strings found in the malware's loader.

I then used DIE (Detected it easy) to calculate the entropy of the malware. However, it's important to note that even if the entropy is high, it doesn't necessarily mean the malware is packed. In this case, the high entropy was likely due to the video and picture embedded within the malware executable:

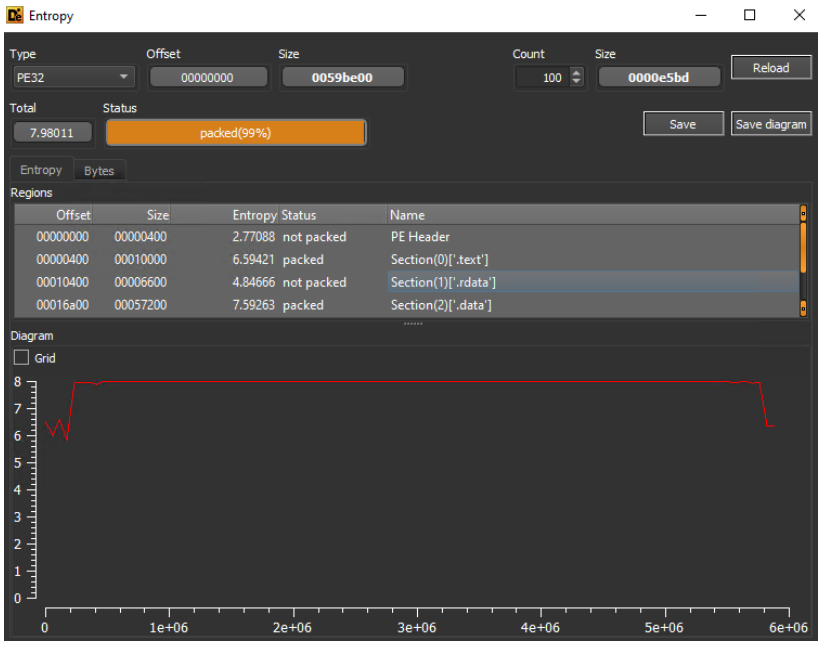

Figure 6 - Screenshot of entropy window within DIE

Using IDA to disassemble the loader, I searched for references to that keyboard registry path. I found something interesting: The malware won't load the malicious payload if a specific keyboard language isn't present, which is 0000040d. A quick Google search helped me determine that the language is Hebrew.

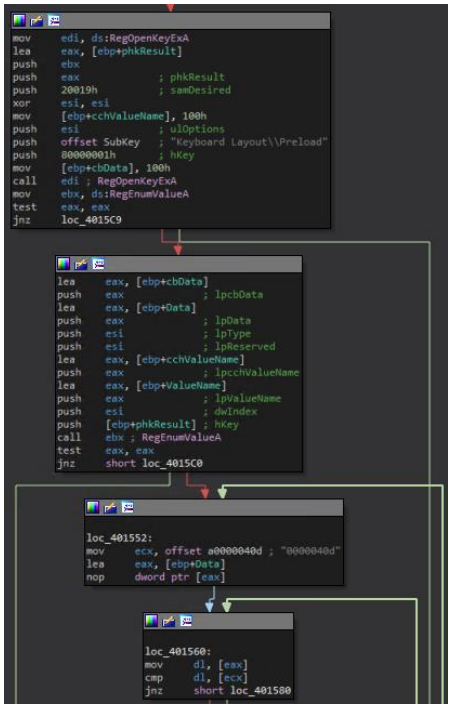

Figure 7 - IDA View of conditioned malware loading.

If Hebrew is indeed one of the languages that Windows uses, the loader will copy itself to a path under the public user directory, disguised as "Microsoft System Agent" to deceive the victim into believing that the loader is part of Microsoft's applications:

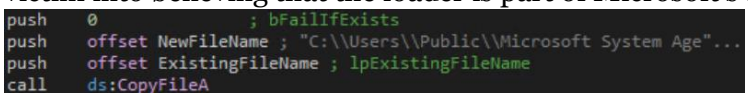

Figure 8 - The loader copies itself.

As we delve deeper into the code, we uncover that the malware's loader executes a shell command. Specifically, it attempts to execute the "runas" command to run the copied loader with elevated permissions.

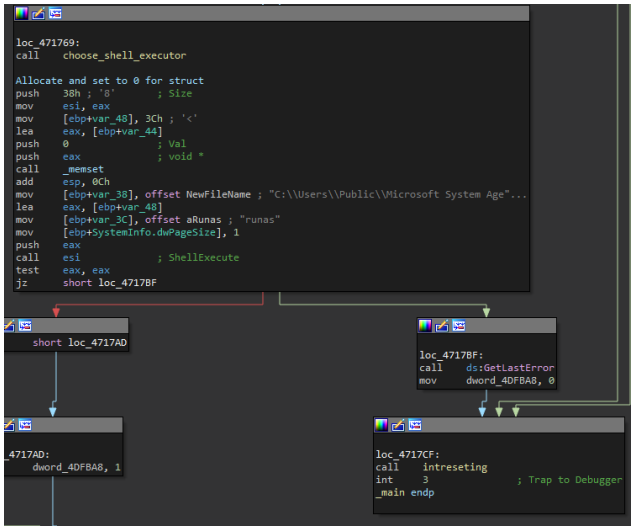

Figure 9 - IDA view of the malware's loader try to run with elevated permissions.

We encountered a significant function that handles EXE files along with media files like MP4 and JPG. This function manages the loading of malicious binaries and extraction of the video and picture content. The video and picture are contained within the executable file.

```
bufptr = 0;
if ( !NetGetDCName(0, 0, &bufptr) )
{
  NetApiBufferFree(bufptr);
  v0 = fopen("C:\\Users\\Public\\Windows Defender Agent.exe", "wb");
  v1 = v0;
  if ( v0 )
  {
    sub_404F8F(&unk_436930, 1, 11776, v0);
    sub_404C9A(v1);
  }
}
v2 = fopen("C:\\Users\\Public\\Microsoft Connection Agent.jpg", "wb");
v3 = v2;
if ( v2 )
{
  sub_404F8F(&unk_439730, 1, 219589, v2);
  sub_404C9A(v3);
}
ResourceA = FindResourceA(0, (LPCSTR)0xD000, (LPCSTR)0xA);
Resource = LoadResource(0, ResourceA);
v6 = LockResource(Resource);
v7 = SizeofResource(0, ResourceA);
v8 = fopen("C:\\Users\\Public\\Video.mp4", "wb");
v9 = v8;
if ( v8 )
{
  sub_404F8F(v6, 1, v7, v8);
  sub_404C9A(v9);
}
v10 = fopen("C:\\Users\\Public\\Microsoft System Manager.exe", "wb");
```

Figure 10 - Pseudocode of the main function which dropping the binaries and media files.

The video contains anti-Netanyahu content aimed at instilling fear in Israeli citizens, while the picture contains propaganda. I won't share the video. Furthermore, I'll cover the analysis of the two malicious binaries in the next section.

Malicious Binaries Dropped

First Malware Dropped

For the first malware, the loader checks for a domain controller; if none is found, it will not deploy it. The malware is saved once again under the public user path, labeled as "Windows Defender Agent", aiming to further confuse the victim.

```
if ( !NetGetDCName(0, 0, &bufptr) )
{
  NetApiBufferFree(bufptr);
  v0 = fopen("C:\\Users\\Public\\Windows Defender Agent.exe", "wb");
  v1 = v0;
  if ( v0 )
  {
    sub_404F8F(&unk_436930, 1, 11776, v0);
    sub_404C9A(v1);
  }
}
```

Figure 11 - Pseudocode of the conditional execution of the first binary.

The dropped malware is in .NET format, confirming initial suspicions. Analysis of the malware's strings and the loader's conditions suggests that the malware may target Windows PCs within active directory.

```
foreach (object obj2 in new DirectorySearcher(new DirectoryEntry("LDAP://" +
  domain.PdcRoleOwner.Name + "/DC=" + domain.Name.Replace(".", ",DC=")))
{
    Filter = "(objectClass=computer)"
}.FindAll())
{
    SearchResult searchResult = (SearchResult)obj2;
    if (searchResult.Properties.Contains("name"))
    {
        string text3 = searchResult.Properties["name"][0].ToString();
        if (!string.Equals(text3, machineName,
         StringComparison.OrdinalIgnoreCase))
        {
            try
            {
                string text4 = "\\\\" + text3 + "\\C$\\Users\\Public\\Microsoft
         System Agent.exe";
                File.Copy(text, text4, true);
```

Figure 12 - The malware searching for computer objects in an Active Directory domain and then coping the loader.

Second Malware Dropped

For the second malware dropped, the loader runs it unconditionally. The malware is saved once again under the public user path, labeled as "Microsoft System Manager", aiming to further confuse the victim. This one is not .NET. This malware is a wiper, it will write random bytes into all files but the malware files.

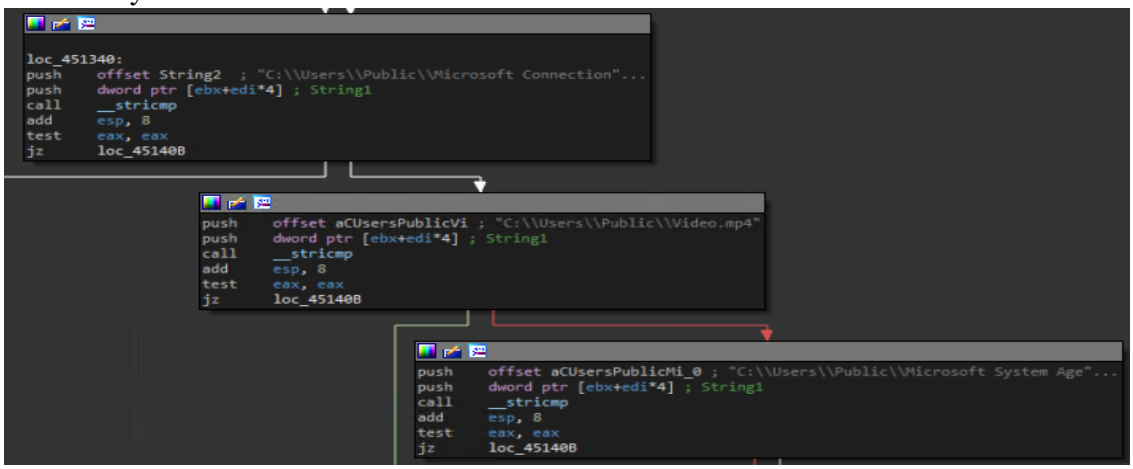

Figure 13 - The malware compares the path given with its file paths.

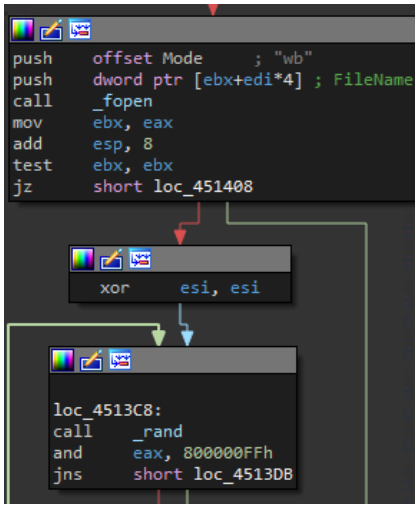

Figure 14 - The malware writes random bytes into all files.

Dynamic analysis

Running the malware did indeed meet our expectations. The video runs repeatedly, and the picture we found is now the wallpaper. Our files were wiped with random bytes.

Using Procmon, we can now observe the process creation of the wiper by the loader:

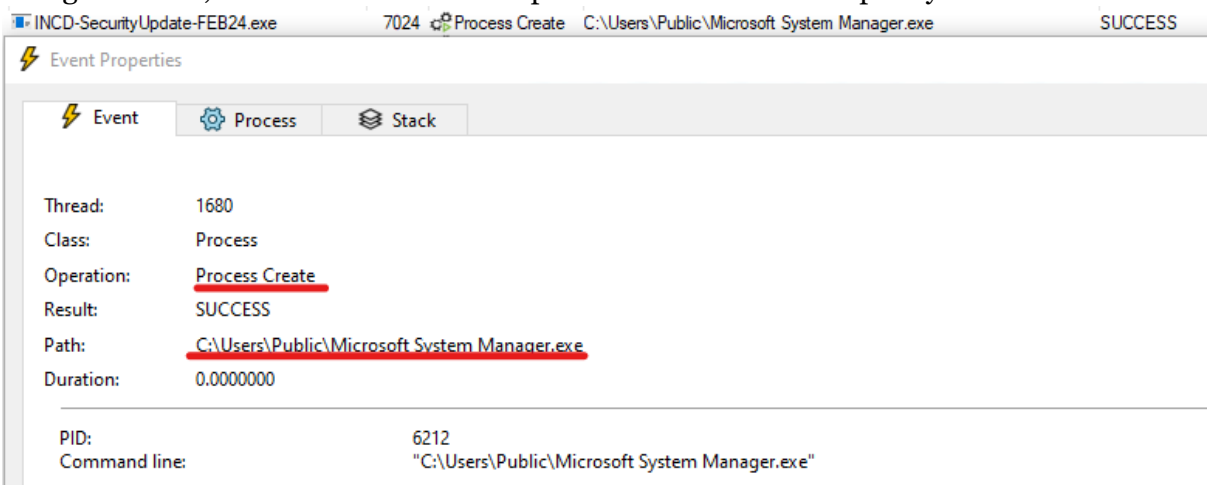

Figure 15 - View of Procmon window showing the process creation of the wiper.

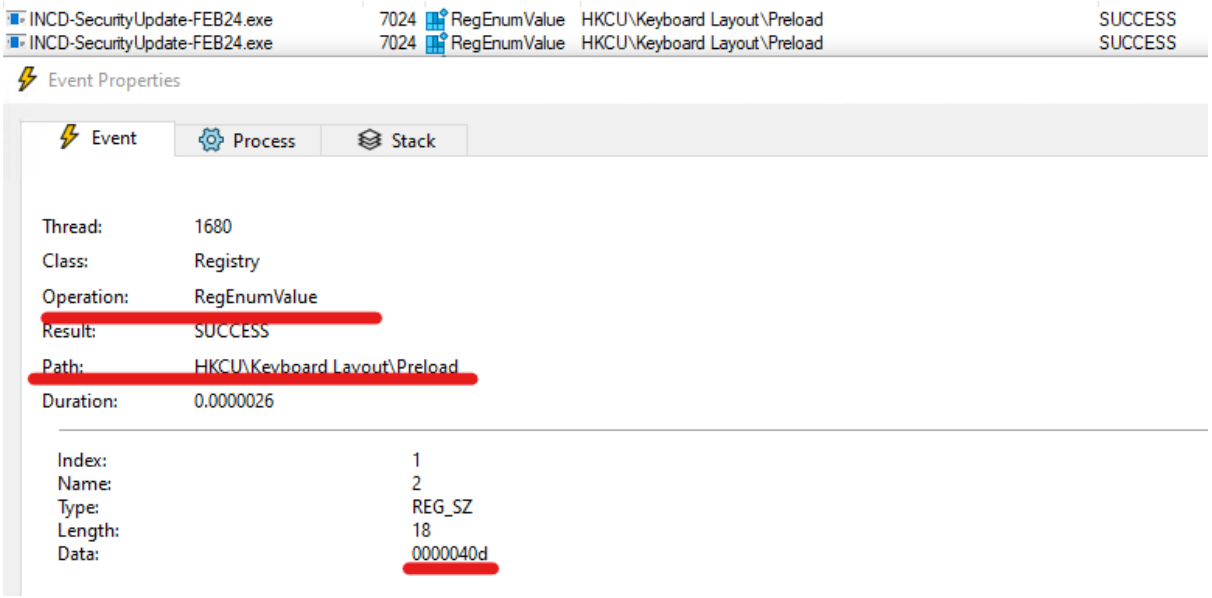

Figure16 - View of Procmon window showing the "RegEnumValue" operation.

Attacker Description

The attacker appears to be a cybercriminal or group of cybercriminals targeting Israeli citizens during the Iron Swords conflict between Israel and Hamas. They demonstrate a certain level of technical capability to impersonate the Israel National Cyber Directorate (INCD) and conduct a malicious email campaign. However, their malware exhibits poor coding practices, lacking protective measures such as packing, obfuscation, or any sophisticated evasion techniques. This suggests a lower level of sophistication or expertise in malware development and indicates a dependence on simple tactics to achieve their objectives.

IOCs

File name	Size	SHA256
INCD-SecurityUpdate-FEB24.exe / Microsoft System Agent.exe	5.61 MB	cff976d15ba6c14c501150c63b69e6c06971c07f8fa048a9974ecf68ab88a5b6
Windows Defender Agent.exe	11.50 KB	b447ba4370d9becef9ad084e7cdf8e1395bafde1d15e82e23ca1b9808fef13a7
Microsoft System Manager.exe	120.00 KB	e6d2f43622e3ecdce80939eec9fffb47e6eb7fc0b9aa036e9e4e07d7360f2b89

Yara rule

```
rule Iron_Swords_Malware {
    meta:
        description = "YARA rule for Iron Swords conflict malware"
        author = "Harel Tsfoni"
        reference = "SHA256: cff976d15ba6c14c501150c63b69e6c06971c07f8fa048a9974ecf68ab88a5b6"
    strings:
        $string1 = "C:\\Users\\Public\\Microsoft System Agent.exe"
        $string2 = "C:\\Users\\Public\\Microsoft System Manager.exe"
        $string3 = "C:\\Users\\Public\\Windows Defender Agent.exe"
        $string4 = "0000040d"
        $string5 = "Keyboard Layout\\Preload"
    condition:
        all of them
}

```

Conclusion

In conclusion, the analysis of the malware targeting Israeli entities in the context of the "Iron Swords" conflict reveals a campaign lacking in sophistication, yet nevertheless, deceptive in its approach to infiltrating Windows systems. The malware is published through deceptive emails, masquerading as a "Software Update."

Employing basic tools such as IDA, CFF Explorer, and strings, we delved into the malware's code structure and behavior. Despite its simplicity, further investigation uncovered basic functionalities designed to evade detection and escalate privileges. The malware's basic check for specific keyboard language settings before executing its payload suggests a somewhat targeted approach towards Israeli citizens.

Furthermore, the analysis revealed the presence of anti-Netanyahu propaganda embedded in the malware, alongside additional malicious payloads within the loader. The first dropped malware, posing as "Windows Defender Agent," indicates an attempt to target Windows PCs within active directory environments. Meanwhile, the second dropped malware, labeled as "Microsoft System Manager," functions as a wiper, corrupting files on infected systems.

In essence, while lacking in complexity, this malware campaign underscores the ongoing need for vigilance and robust cybersecurity measures to counter even basic threats.