ESET Wiper: Iranian APT Group Toufan's Politically Motivated Attack on Israeli Firms | Idan Malihi

Archived: 2026-04-05 13:22:03 UTC

Wiper Diagram Execution

The hacktivist group expressed a political opinion through the ESET wiper that the hacktivist group is attacking ESET customers due to the war between Israel and Hamas that started on October 07th, and Hamas started their attack at 06:29 AM.

The hacktivist group delivers a threat through the wiper to the ESET company: "`Hey ESET, wait for the leak.. Doing business with the occupiers puts you in scope!`"

During its operation, the ESET wiper executes a file with administrator privileges using the `runas` command. It appears that the wiper uses the `ShellExecute` API function to execute the file specified in the `open` string. Additionally, the wiper will execute a file located in the `Users\Public` path.

Additionally, the wiper generates a `conf.conf` file in the `Users\Public` directory and drops three files: `MicrosoftEdge.exe`, `csrs.exe`, and `SecurityHealthSystray.exe`.

Regarding the following string, it appears the wiper will play an MP4 video during its execution.

The wiper is connected to the URL `www.oref.org.il/alerts/RemainderConfig_eng.json`, which is Israel's civil defense alert system. This indicates that the wiper might be trying to access or misuse information from this URL.

The ESET wiper loaded the winhttp.dll file to import several API functions related to HTTP/S connections to the `www.oref.org.il/alerts/RemainderConfig_eng.json`, Israel's civil defense alert system. This communication purpose is to check the returned status code to determine whether the system's IP address is related to Israel. If the IP address is not related to Israel, the response will be 403, forbidden, and the wiper will not continue with the infection process.

Otherwise, it continues with the infection process.

Then, the ESET wiper creates a new file named `conf.conf` using the `CreateFileW` function, located in the `C:\Users\Public` directory.

The wiper employs the `fwrite()` C function to write the buffer in the `EDI` register to the `conf.conf` file.

The content of `conf.conf` appears to consist of the first 7 bytes from `www.oref.org.il/alerts/RemainderConfig_eng.json`. If the wiper successfully extracts these bytes, it indicates that the system is related to Israel.

Then, the wiper replicates itself using the `CopyFileA` function to the `C:\Users\Public` directory, naming it `SecurityHealthsSystray.exe`.

After the wiper replicates itself, it loads the `Shell32.dll` file into the process's address space using the `LoadLibraryA` function. This allows the use of the `ShellExecuteExA` function to execute the `SecurityHealthSystray.exe` file with administrator privileges via the `runas` system command.

The wiper initially executes the `SecurityHealthSystray.exe` file, but it checks the specific path from which the file is run. If it runs from the Desktop path, it will execute one part of the code. However, if it runs from the `C:\Users\Public path`, it will execute a different part of the code, which contains the next malicious code. When the wiper is executed from the `C:\Users\Public` path, it suspends the execution of the process's thread for a specified number of milliseconds.

The wiper utilizes several API functions to capture a snapshot of the endpoint's running processes. It also checks for the presence of known monitoring, debugging, or disassembling tools on the operating system. The functions used for this purpose include `CreateToolhelp32Snapshot`, `Process32First`, and `Process32Next`.

If the wiper detects known tools like `procexp.exe`, `procmon.exe`, and `xdbg32.exe`, it prompts error messages related to the `MicrosoftEdge.exe` and `csrs.exe` files.

Otherwise, the wiper fails to detect the tools and uses the `CreateProcessA` function to execute the `MicrosoftEdge.exe` file.

After running several malicious processes in the background, the wiper utilizes the `SystemParametersInfoA` API to modify the desktop wallpaper. The operation `push dword ptr ds:[D59398]` corresponds to the `pvParam` parameter, which points to the wallpaper file path. During execution, this is a pointer to a memory location that contains the string `C:\Users\Public\image.jpg`. The operation `push 14` pertains to the `uiAction` parameter. The decimal value `14` represents the `SPI_SETDESKWALLPAPER` parameter, which is responsible for setting the desktop wallpaper to the file specified in the `pvParam` parameter.

The `SendInput` function clicks the Volume Up button on the keyboard to set the speaker volume to 100 percent for the video displayed on the victim's screen.

Then, the wiper uses the `ShellExecuteA` function to execute the `video.mp4` file, playing a video related to the Hamas-ISIS Israel war, which began on October 7^{th,} on the victim's screen.

Once the wiper detects the connected system drives, it generates multiple threads to expedite the wiping process.

Content of `private.txt` prior to wiping:

Content of `private.txt` after wiping:

Upon execution, the program first hides the console window using `Program.FreeConsole()`, effectively concealing its presence from the user. It then attempts to access the Windows registry to check for the presence of Microsoft Outlook by locating the `OUTLOOK.EXE` path. If this search is successful, the malware proceeds to download a malicious ZIP file from a remote URL `https://share-center.com/files/Attachment.zip` and saves it to the public user directory. Next, the malware constructs an email with the subject line "`The Files You`

`Requested`" including the infected ZIP file as an attachment. This action is intended to spread the ZIP file to other users. Additionally, if the system is part of an Active Directory environment, the malware checks for domain membership and attempts to execute a function called `InfectAD` to propagate throughout the network, highlighting its intent to spread laterally within enterprise environments.

The `InfectOutlook` function is designed to automate Microsoft Outlook's sending of phishing emails with attachments to a large number of recipients. It starts by creating an instance of the Outlook application using a `CreateInstance` call with a COM object. Once the Outlook instance is generated, it accesses the MAPI namespace to retrieve the global address list. The function then iterates through all the entries in the address list, extracting email addresses. These addresses are stored in the `uniqueEmails` list, ensuring that no duplicates are added by checking for the presence of each email address before including it. Additionally, the function appends a predefined email address, `brunomartin@tutamail.com`, to this list, likely to send copies of the email to the attacker's email.

MITRE ATT&CK

Yara Rule
