Malicious Macros and Zone Identifier Alternate Data Stream Information Bypass

cloudsek.com/malicious-macros-and-zone-identifier-alternate-data-stream-information-bypass/

Anandeshwar Unnikrishnan March 28, 2022

VBA macro is one of the most extensively abused features of Microsoft Excel, used by adversaries to gain an initial foothold in the target network. The sophistication of the technique leveraged to deploy the payload determines the detection of the macro code. However, the main reason for the risk is still the inherent nature of the Office applications to offer execution of macros.

Recently Microsoft took a bold move and disabled the macros from running in documents downloaded from the internet. This blog covers a few techniques employed by adversaries to bypass such restrictions and execute malicious macros in documents downloaded from the internet.

What are Macros and how are they used maliciously?

Macros are special-use programmes that are used to automate operations within a larger application or piece of software. Macros consist of a set of instructions and operations expressed in a Macro Language (such as Visual Basic for Applications or VBA) or a conventional programming language. When a certain trigger is fired, the programme will automatically execute these commands. Threat actors write malicious code in the same Macro Language and hide it in documents and spreadsheets, distributed over the internet. Followed by which the code is activated as soon as the file is opened.

Zone Identifier and Security

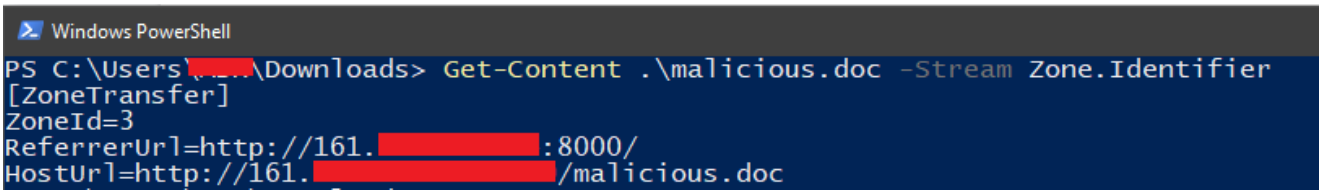
How Does Windows Identify the Source of the Files?

In a phishing attack, data downloaded from the internet is handled by a browser application running on Windows. Browsers create an Alternate Data Stream named `"Zone.Identifier,"` whenever such data is downloaded and a *"ZoneId"* is addedto this stream, representing the zone from which the file originated. Zone IDs are listed in the table below. For more information on URL security zones refer to this article.

Zone	ZoneId
Local machine	0
Local intranet	1
Trusted sites	2

Internet	3
Restricted sites	4

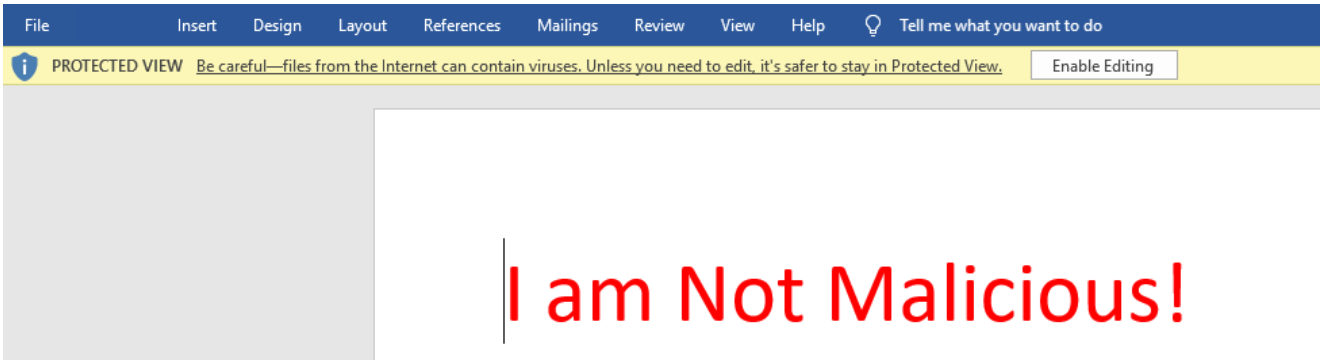
We can use PowerShell to query the Zone.Identifier stream data to obtain the assigned ZoneId. As shown in the image below, a lot of information on the file is retrievable, especially its source information such as URL and host details. The file shown in the image has a ZoneId of 3. And with reference to the aforementioned table, ID 3 stands for 'Internet'. This indicates that the malicious.doc is downloaded from the internet.



This indicates that the malicious.doc is downloaded from the internet.
Security implementations on Windows make use of the zone identifier data and utilize the "Mark of the Web" feature, to identify local files from downloaded ones. Here are a few mechanisms that leverage the "Mark of the Web" feature to take action:

- Windows Smart Screen
- Protected View in Office Suite
- Application Guard
- Visual Studio untrusted data protection

In the above instance, when malicious.doc with ZoneId 3 is opened on Word, it opens in Protected View as shown below:



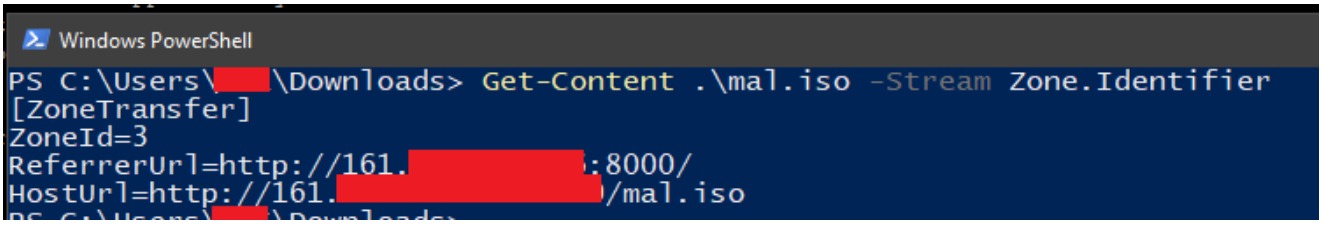
In the above instance, when malicious.doc with ZoneId 3 is opened on Word, it opens in Protected View as shown below:

Executing Internet Macros

When non-NTFS (New Technology File System) formats are used as containers to hold a malicious file, the browser will be incapable of creating ADS in files inside the container. In such cases, the browser fails to assign a ZoneId for the attacker files, enabling macro execution.

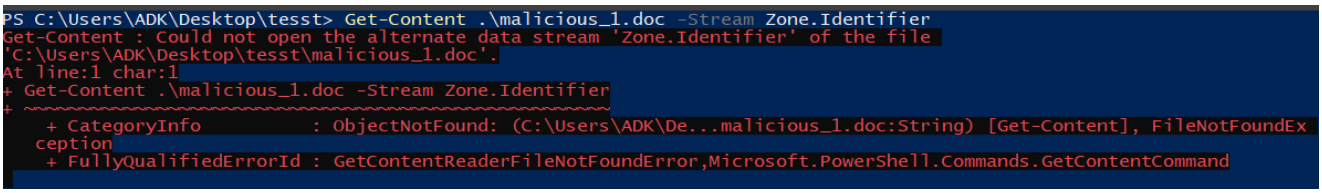
The “.iso” format is a popular container choice for malicious files. And such files extracted from ISO don’t have a Zone. Identifier stream, which causes Windows to treat it as a local file.

The image below shows that a downloaded ISO file uses the Mark of the Web and is assigned ZoneId 3.



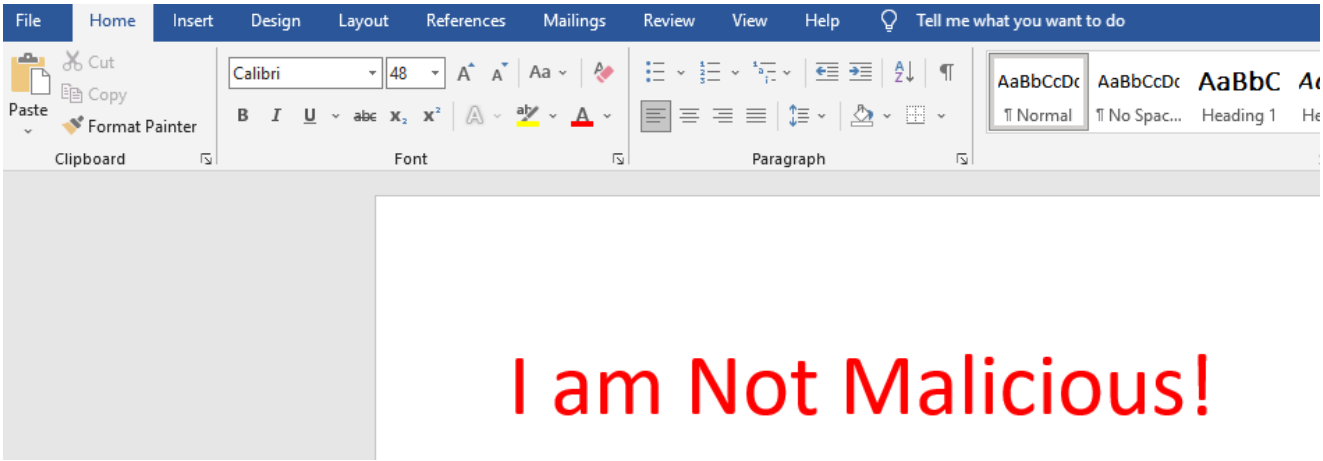
downloaded ISO file uses the Mark of the Web and is assigned ZoneId 3

When files are extracted from ISO to the disk, they do not have any zone identification data as shown in the following image. Hence when stream data is queried, PowerShell throws an error stating that the stream object cannot be found. Verifying that the malicious file is categorized as a local file by the system.



Hence when stream data is queried, PowerShell throws an error stating that the stream object cannot be found.

When a malicious document extracted from an ISO is opened in Word, the ProtectedView feature doesn’t alert the user. Thus no-sandbox protection will be offered by Protected View, allowing the malware to have an unrestricted execution on the system.



When a malicious document extracted from an ISO is opened in Word, the ProtectedView feature doesn’t alert the user

Conclusion

Adversaries exploit the features of VBA Macros to bypass Zone Identifier techniques employed by Office applications. They execute malicious macros in documents that the browser fails to identify as files from the internet, only to jump security measures and infect target networks. Observing the following mitigation measures could allow users to prevent such attacks:

- A Defense in Depth approach should always be preferred instead of relying on a single defense.
- It is ideal to enforce Microsoft's Attack Surface Reduction rules (ASR rules) in an enterprise environment.
- Do not execute files shipped in suspicious container formats.

Author Details

Anandeshwar Unnikrishnan
Threat Intelligence Researcher , CloudSEK
Anandeshwar is a Threat Intelligence Researcher at CloudSEK. He is a strong advocate of offensive cybersecurity. He is fuelled by his passion for cyber threats in a global context. He dedicates much of his time on Try Hack Me/ Hack The Box/ Offensive Security Playground. He believes that "a strong mind starts with a strong body." When he is not gymming, he finds time to nurture his passion for teaching. He also likes to travel and experience new cultures.

Gursehaj Singh
Total Posts: 0
Gursehaj is a Threat Intelligence Editor Intern at CloudSEK. In hi free time you can find him reading and writing Medium articles :partying_face:
×

Anandeshwar Unnikrishnan
Threat Intelligence Researcher , CloudSEK
Anandeshwar is a Threat Intelligence Researcher at CloudSEK. He is a strong advocate of offensive cybersecurity. He is fuelled by his passion for cyber threats in a global context. He dedicates much of his time on Try Hack Me/ Hack The Box/ Offensive Security Playground. He believes that “a strong mind starts with a strong body.” When he is not gymming, he finds time to nurture his passion for teaching. He also likes to travel and experience new cultures.

-
-

Latest Posts

- 
- 

- 
-