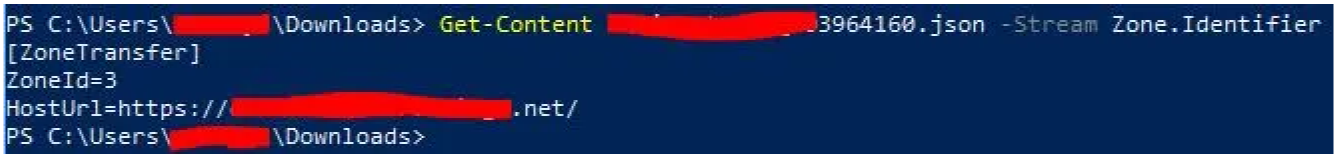
KYLE CUCCI'S CYBER RAMBLINGS

Home Whoami Projects



```
PS C:\Users\[redacted]\Downloads> Get-Content [redacted]3964160.json -Stream Zone.Identifier
[ZoneTransfer]
ZoneId=3
HostUrl=https://[redacted].net/
PS C:\Users\[redacted]\Downloads>
```

HOW MALWARE ABUSES THE ZONE IDENTIFIER TO CIRCUMVENT DETECTION AND ANALYSIS

By D4rksystem | June 13, 2023

I was investigating a malware sample that uses an interesting trick to circumvent sandboxes and endpoint defenses by simply deleting its zone identifier attribute. This led me on a tangent where I began to research more about zone identifiers (which, embarrassingly enough, I had little knowledge of prior). Here are the results of my research.

The Zone.Identifier is a file metadata attribute in Windows that indicates the security zone where a file originated. It is used to indicate a level of trustworthiness for a file when it is accessed, and helps

Windows determine the security restrictions that may apply to the file. For example, if a file was downloaded from the Internet, the zone identifier will indicate this, and extra security restrictions will be applied to this file in comparison to a file that originated locally on the host.

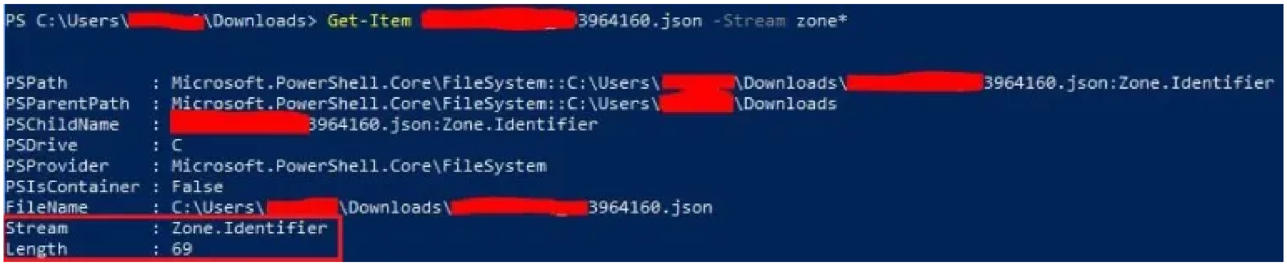
The zone identifier is stored as an alternate data stream (ADS) file, which resides in the file's metadata. There are five possible zone identifier values that can be applied to a file, represented as numerical values of 0 to 4:

- Zone identifier "0": Indicates that the file originates on the local machine. This file will have the least security restrictions.
- Zone identifier "1": Indicates that the file originated on the local Intranet (local network). Both zone identifier 0 and 1 indicate a high level of trust.
- Zone identifier "2": Indicates that the file was downloaded from a trusted site, such as an organization's internal website.
- Zone identifier "3": Indicates that the file was downloaded from the Internet and that the file is generally untrusted.
- Zone identifier "4" – Indicates that the file came from a likely unsafe source. This zone is reserved for files that must be treated with extra caution as they may contain malicious content or pose a security risk.

You can use the following PowerShell command to check if a file has a zone identifier ADS:

```
Get-Item <file_path> -Stream zone*
```

An example of this output can be seen below. Notice the highlighted area that denotes the ADS stream ("Zone.Identifier") and its length. Also note that if no data is returned after running this command, the file likely does not have a zone identifier stream.



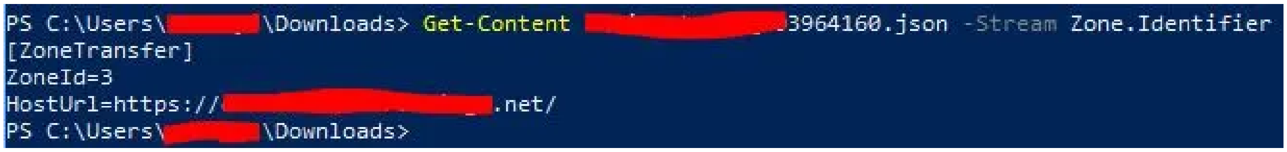
```
PS C:\Users\[redacted]\Downloads> Get-Item [redacted]3964160.json -Stream zone*


PSPath        : Microsoft.PowerShell.Core\FileSystem::C:\Users\[redacted]\Downloads\[redacted]3964160.json:Zone.Identifier
PSParentPath  : Microsoft.PowerShell.Core\FileSystem::C:\Users\[redacted]\Downloads
PSChildName   : [redacted]3964160.json:Zone.Identifier
PSDrive       : C
PSProvider    : Microsoft.PowerShell.Core\FileSystem
PSIsContainer : False
FileName      : C:\Users\[redacted]\Downloads\[redacted]3964160.json
Stream        : Zone.Identifier
Length        : 69
```

To view this file's zone identifier stream, you can use the following PowerShell one-liner:

```
Get-Content <file_path> -Stream Zone.Identifier
```

An example of this can be seen below:



```
PS C:\Users\[redacted]\Downloads> Get-Content [redacted]3964160.json -Stream Zone.Identifier
[ZoneTransfer]
ZoneId=3
HostUrl=https://[redacted].net/
PS C:\Users\[redacted]\Downloads>
```

A zone identifier file will look like something like this:

```
[ZoneTransfer]
ZoneId=3
ReferrerUrl=https://www.evil.com
HostUrl=https://download.evil.com/malware.doc
```

In this example, this Zone.Identifier indicates that the associated file originates from “zone 3”, which typically corresponds to the Internet zone. The ReferrerUrl denotes the domain of the webpage where the file was downloaded from or potentially the referrer domain, and the HostUrl specifies the precise location where the file was downloaded from.

These zones are also referred to as the Mark of the Web (MoTW). Any file that originates from Zone 3 or Zone 4, for example, are said to have the mark of the web.

Malware can abuse the zone identifier in a few different ways, with a couple different goals:

Defense Evasion

Malware can manipulate the zone identifier value to spoof the trust level of a file. By assigning a lower security zone to a malicious file, the malware can trick Windows and defense controls into treating the file as if it came from a trusted source.

To accomplish this, malware can simply modify its files’ zone identifiers. Here is how this can be accomplished via PowerShell:

```
Set-Content file.exe -Stream Zone.Identifier -Value "[ZoneTransfer]`nZoneId=1"
```

This PowerShell one-liner modifies a file’s zone identifier to be a certain value (in this case, setting the zone ID to “1”). This may help the malware slip past certain defensive controls like anti-malware and EDR, and may make the malware look less suspicious to an end user.

Or, the zone identifier stream can simply be deleted, which may trick some defense controls. In order to attempt to bypass defenses, a variant of the malware family SmokeLoader does exactly this. SmokeLoader calls the Windows API function DeleteFile (see code below) to delete its file's zone identifier stream. You can investigate this for yourself in a SmokeLoader analysis report from JoeSandbox (SHA256: 86533589ed7705b7bb28f85f19e45d9519023bcc53422f33d13b6023bab7ab21).

```
DeleteFileW (C:\Users\user\AppData\Roaming\ichffhi:Zone.Identifier)
```

Alternatively, malware authors can wrap their malware in a container such as a IMG or ISO file, which do not typically have zone identifier attributes. Red Canary has a great example in this report.

Anti-Analysis and Sandbox Evasion

Malware may inspect the zone identifier of a file to circumvent analysis. Malicious files that are submitted to an analysis sandbox or are being analysed by a reverse engineer may have a different zone identifier than the original identifier the malware author intended. When the malware file is submitted to a sandbox, the zone identifier may be erroneously set to 0, when the original value is 3. If malware detects an anomalous zone identifier, it may cease to execute correctly in the sandbox or lab environment.

The pseudo-code below demonstrates the logic of how malware may check its file's zone identifier:

```
zone_identifier_path = current_file_path + ":Zone.Identifier"

with open(zone_identifier_path, 'r') as f:
     zone_info = f.read()

     # Check if the zone is Internet zone (zone ID 3 or higher)
     if "ZoneId > 2" in zone_info:

          # File is from the Internet zone (as expected), continue running
          return()

     else:
          # File may be running in a sandbox or analysis lab!
          TerminateProcess()
```

If you are craving more information on this topic, other good resources are here and here.

— Kyle Cucci (d4rksystem)

Share this:

Like this:

Loading...

Forensics, *Malware Analysis*

Forensics, *Malware*, *Malware Analysis*

Comments are closed.

PREVIOUS

Book Summary – “Evasive Malware: Understanding Deceptive And Self-Defending Threats”

NEXT

Analysis Of The NATO Summit 2023 Lure: A Step-By-Step Approach

SEARCH POSTS

Search …

RECENT POSTS

- “Beeeeeeeeep!”. How Malware Uses the Beep WinAPI Function for Anti-Analysis
- Unpacking StrelaStealer
- Creating Quick and Effective Yara Rules: Working with Strings
- Analysis of the NATO Summit 2023 Lure: A Step-by-Step Approach
- How Malware Abuses the Zone Identifier to Circumvent Detection and Analysis

CATEGORIES

- Evasive Malware Book
- Forensics
- Malware Analysis
- Recon

ARCHIVES

- May 2024
- March 2024
- January 2024
- July 2023
- June 2023
- May 2023
- August 2021
- January 2021

- December 2020
- August 2020
- April 2020
- January 2020
- October 2019
- June 2019
- February 2018
- September 2017
- August 2017

Search Posts

Search …

Recent Posts

- “Beeeeeeeeep!”. How Malware Uses the Beep WinAPI Function for Anti-Analysis
- Unpacking StrelaStealer
- Creating Quick and Effective Yara Rules: Working with Strings
- Analysis of the NATO Summit 2023 Lure: A Step-by-Step Approach
- How Malware Abuses the Zone Identifier to Circumvent Detection and Analysis

Recent Comments

Fileless attacks: How attackers evade traditional AV and how to stop them – Computer Security Articles on Malware Analysis in 5-Minutes: Deobfuscating PowerShell Scripts

Where do entries from regsvr32-registered DLLs get stored - Boot Panic on Chantay's Resume: Investigating a CV-Themed ZLoader Malware Campaign

Archives

- May 2024
- March 2024
- January 2024
- July 2023
- June 2023
- May 2023
- August 2021
- January 2021
- December 2020
- August 2020
- April 2020
- January 2020
- October 2019
- June 2019
- February 2018
- September 2017
- August 2017

Categories

- Evasive Malware Book
- Forensics
- Malware Analysis

Meta

Log in

Entries feed

Comments feed

WordPress.org