Mark-of-the-Web from a Red Team's Perspective | Outflank

By Stan

Published: 2020-03-30 · Archived: 2026-04-05 19:40:02 UTC

Zone Identifier Alternate Data Stream information, commonly referred to as Mark-of-the-Web (abbreviated MOTW), can be a significant hurdle for red teamers and penetration testers, especially when attempting to gain an initial foothold.

Your payload in the format of an executable, MS Office file or CHM file is likely to receive extra scrutiny from the Windows OS and security products when that file is marked as downloaded from the internet. In this blog post we will explain how this mechanism works and we will explore offensive techniques that can help evade or get rid of MOTW.

Note that the techniques described in this blog post are not new. We have witnessed all of them being abused in the wild. Hence, this blog post serves to raise awareness on these techniques for both red teamers (for more realistic adversary simulations) and blue teamers (for better countermeasures and understanding of attacker techniques).

Introduction to MOTW

Mark-of-the-Web (MOTW) is a security feature originally introduced by Internet Explorer to force saved webpages to run in the security zone of the location the page was saved from. Back in the days, this was achieved by adding an HTML comment in the form of `<!--saved from url=>` at the beginning of a saved web page.

This mechanism was later extended to other file types than HTML. This was achieved by creating an alternate data stream (ADS) for downloaded files. ADS is an NTFS file system feature that was added as early as Windows 3.1. This feature allows for more than one data stream to be associated with a filename, using the format “filename:streamname”.

When downloading a file, Internet Explorer creates an ADS named `Zone.Identifier` and adds a ZoneId to this stream in order to indicate from which zone the file originates. Although it is not an official name, many people still refer to this functionality as Mark-of-the-Web.

Listing and viewing alternate data streams is trivial using PowerShell: both the `Get-Item` and `Get-Content` cmdlets take a “Stream” parameter, as can be seen in the following screenshot.

The following ZoneId values may be used in a Zone.Identifier ADS:

- 0. Local computer
- 1. Local intranet
- 2. Trusted sites
- 3. Internet
- 4. Restricted sites

Nowadays all major software on the Windows platform that deals with attachments or downloaded files generates a Zone.Identifier ADS, including Internet Explorer, Edge, Outlook, Chrome, FireFox, etc. How do these programs write this ADS? Either by creating the ADS directly or via the system's implementation of the IAttachmentExecute interface. The behavior of the latter can be controlled via the SaveZoneInformation property in the Attachment Manager.

Note that Windows 10's implementation of the IAttachmentExecute interface will also add URL information to the Zone.Identifier ADS:

For red teamers, it's probably good to realize that MOTW will also get set when using the HTML smuggling technique (note the "blob" keyword in the screenshot above, which is an indicator of potential HTML smuggling).

The role of MOTW in security measures

The information from the Zone Identifier Alternate Data Stream is used by Windows, MS Office and various other programs to trigger security features on downloaded files. The following are the most notable ones from a red teamer's perspective (but there are more – this list is far from complete).

Windows Defender SmartScreen

This feature works by checking downloaded executable files (based on Zone Identifier ADS) against a whitelist of files that are well known and downloaded by many Windows users. If the file is not on that list, Windows

Defender SmartScreen shows the following warning:

MS Office protected view

The Protected View sandbox attempts to protect MS Office users against potential risks in files originating from the internet or other dangerous zones. By default, most MS Office file types flagged with MOTW will be opened in this sandbox. Many users know this feature as MS Office's famous yellow bar with the "Enable Editing" button.

MWR (now F-Secure labs) has published a great technical write-up on this sandbox some years ago. Note that some MS Office file types cannot be loaded in the Protected View sandbox. SYLK is a famous example of this.

MS Office block macros downloaded from the internet

This feature was introduced in Office 2016 and later back-ported to Office 2013. If this setting is enabled, macros in MS Office files flagged with MOTW are disabled and a message is displayed to the user.

This warning message cannot be ignored by the end user, which makes it a very effective measure against mass-scale macro-based malware.

Visual Studio project files

Opening untrusted Visual Studio project files can be dangerous (see my presentation at Nullcon Goa 2020 for the reasons why). By default, Visual Studio will display a warning message for any project file which has the MOTW attribute set.

Application Guard for Office

This newly announced feature runs potentially malicious macros embedded in MS Office files in a small virtual machine (based on Application Guard technology) in order to protect the OS.

From the limited documentation available, the decision to run a document in a VM is based on MOTW. Unfortunately, I don't have access to this technology yet, so I cannot confirm this statement through testing.

Strategies to get rid of MOTW

From a red teamer's perspective, there are two strategies we can employ to evade MOTW. All of the techniques that we have witnessed in the wild can be categorized under the following two strategies:

1. **Abusing software that does not set MOTW** – delivering your payload in a file format which is handled by software that does not set or propagate Zone Identifier information.
2. **Abusing container formats** – delivering your payload in a container format which does not support NTFS' alternate data stream feature.

Of course there is a third strategy: social engineering the user into removing the MOTW attribute (right click file -> properties -> unblock). But since this is a technical blog post, this strategy is out of scope for this write-up. And

for the blue team: you can technically prevent your end-users from doing this by setting HideZoneInfoOnProperties via group policy.

Let's explore the two technical strategies for getting rid of MOTW in more depth…

Strategy 1: abusing software that does not set MOTW

The first strategy is to deliver your payload via software that does not set (or propagate) the MOTW attribute.

A good example of this is the Git client. The following picture shows that a file cloned from GitHub with the Git client does not have a Zone.Identifier ADS.

For red teamers targeting developers, delivering your payloads via Git might be a good option to evade MOTW. This is especially relevant for payloads targeting Visual Studio, but that is material for a future blog post. 🙂

Another famous example of software that does not set a Zone.Identifier ADS is 7Zip. This archiving client only sets a MOTW flag when a file is double-clicked from the GUI, which means the file is extracted to the temp directory and opened from there. However, upon manual extraction of files to other locations (i.e. clicking the extract button instead of double-clicking), 7Zip does *not* propagate a Zone.Identifier ADS for extracted files. Note that this works regardless of the archiving file format: any extension handled by 7zip (7z, zip, rar, etc) will demonstrate this behavior.

This appears to be a conscious design decision by the 7Zip lead developer, as can be seen in the following excerpt from a discussion on SourceForge. More information can be found here.

As a side note, I wouldn't recommend using 7Zip for extracting potentially dangerous files anyway, since it is a product known for making "odd" security decisions (such as the lack of ASLR…).

Strategy 2: abusing container formats

Remember that alternate data streams are an NTFS feature? This means that Zone Identifier ADS cannot be created on other file systems, such as FAT32. From a red teamer's perspective we can exploit this behavior by embedding our payload in a file system container such as ISO or VHD(X).

When opening such a container with Windows Explorer, MOTW on the outside container will not be propagated to files inside the container. This is demonstrated in the screenshot below: the downloaded ISO is flagged with MOTW, but the payload inside the ISO is not.

Note that payload delivery via the ISO format is an evasion technique commonly observed in the wild. For example, TA505 is a prominent actor known to abuse this technique.

Message to the Blue Team

So, what does all of this mean when you are trying to defend your network?

First of all, the fact that a security measure can be circumvented does not render such a measure useless. There will be plenty of attackers that do *not* use the techniques described in this blog post. In particular, I am a big fan of the measure to block macros in files downloaded from the internet which is available in MS Office 2013 and subsequent versions.

Second, the techniques described in this blog post acknowledge a very important security paradigm: *defense in depth*. Do not engineer an environment in which your security depends on a single preventive measure (in this example MOTW).

Start thinking about which other measures you can take in case attackers are trying to evade MOTW. For example, if feasible for your organization, block container formats in your mail filter and proxy. Also, limit the impact of any malicious files that may have bypassed measures relying on MOTW, for example using Attack Surface Reduction rules.

I think you get the idea: *don't do coconut security* – a single hard layer, but all soft when it's cracked.

In order to help other red teams easily implement these techniques and more, we've developed Outflank Security Tooling (OST), a broad set of evasive tools that allow users to safely and easily perform complex tasks. If you're interested in seeing the diverse offerings in OST, we recommend scheduling an expert led demo.
