Eleven Paths

SmartScreen Report

Telefónica CYBER SECURITY UNIT

elevenpaths.com

Content

What is SmartScreen?

SmartScreen is a component of Windows Defender aimed at protecting users against potentially harmful attacks, whether in the form of links or files. It is included under this name from Internet Explorer 8 in Windows 7, and protection against executables was added in Internet Explorer 9. In Windows 8 this binary reputation protection was introduced to the entire operating system.

When a user is browsing the Internet, the filter or SmartScreen component analyses the sites they are visiting and, if they access a site considered suspicious, it displays a warning message so that the user can decide whether or not to continue. To do this, it uses a dynamic list of phishing sites and domains using malicious code, and if the sites have already been reported, access is blocked.

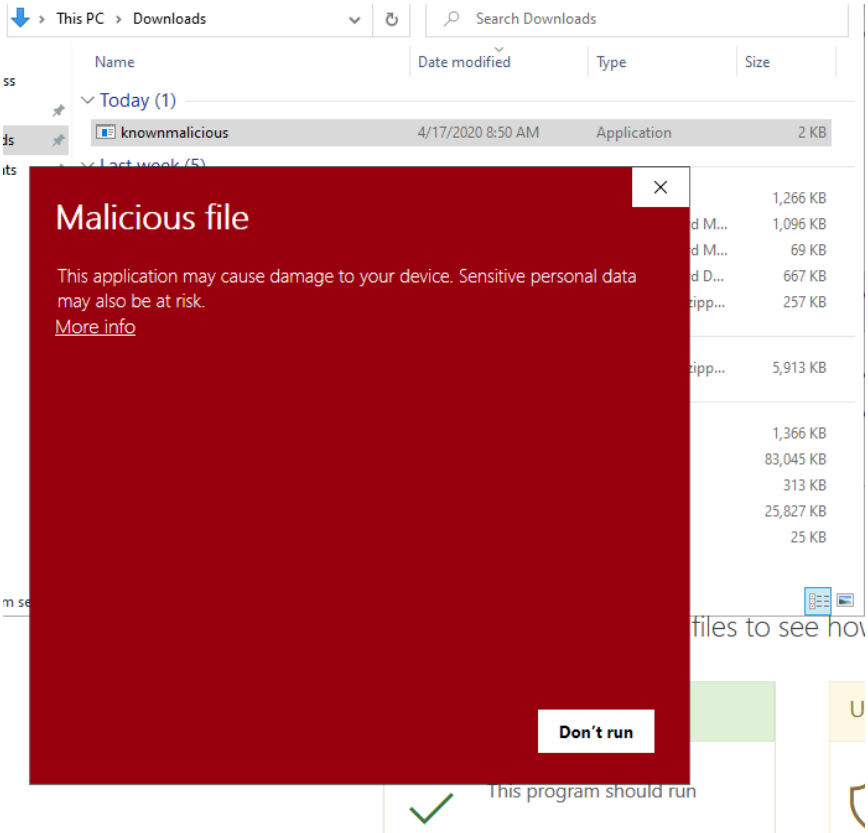
As for downloaded files, SmartScreen checks them to verify their reputation on the basis of the information collected from other users of Microsoft products.

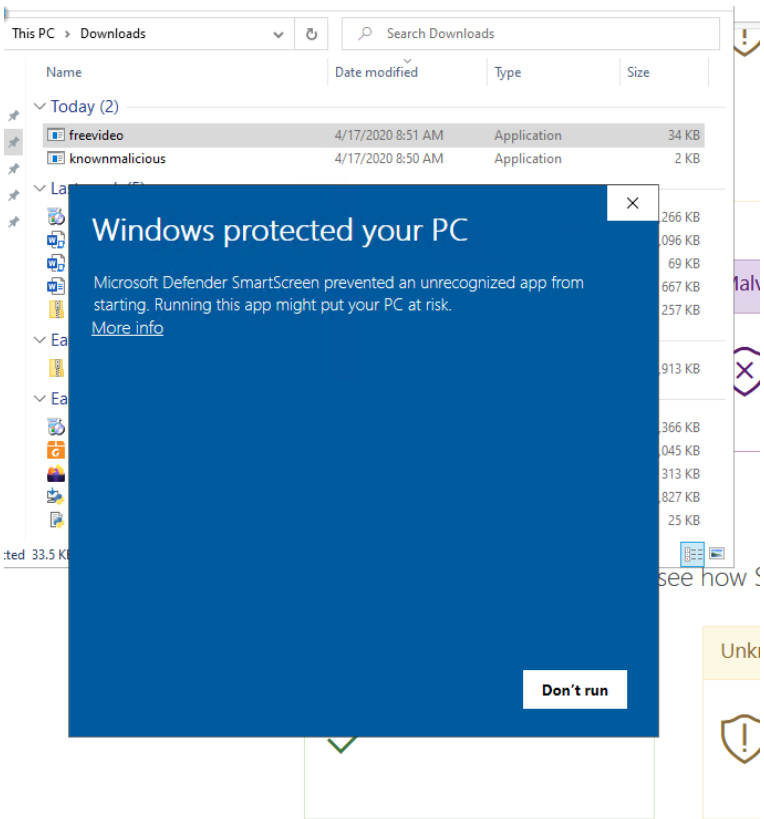
In short, according to Microsoft's website[1], SmartScreen offers the following advantages:

- Anti-phishing and Anti-malware.
- Reputation-based URL and app protection.
- Operating system integration.
- Improved heuristics and diagnostic data.
- Management through Group Policy and Microsoft Intune.
- Blocking URLs associated with potentially unwanted applications.

The goal of SmartScreen is quite clear and has been analysed in several sites. However, for this report we focus on understanding how SmartScreen works, particularly on files downloaded to the system. **We have tried to understand what triggers this protection component developed by Microsoft to better understand its effectiveness.**

[1] Windows Defender SmartScreen. [Online]. Available: https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-defender-smartscreen/windows-defender-smartscreen-overview.





What is Alternate Data Stream?

To understand how SmartScreen works with downloaded files, we must first talk about ADSs. Alternate Data Streams or **ADS** is a feature of the NTFS file system developed by Microsoft and included since the days of Windows NT (1993). ADS makes it possible to store metadata in a file, either through a stream directly or through another file. The command line defines the most common use, typing:

```
echo hidden text >archivoOriginal.txt:archivoAEsconder.txt
```

and to display it, instead of executing a ***type archivoOriginal.txt***, since they would see the information *archivoOriginal.txt*, they executed:

```
type archivoOriginal.txt:archivoAEsconder.txt
hidden text
```

In the 1990s, many used **ADS** to hide information within a file because of the poor documentation existing at the time. Some software also hid malicious code using these techniques. However, it was not only used for malicious purposes, but the other way round. Some other products have used it for different actions, such as Kaspersky Antivirus, that used it to save the results of scanned files in the same file as an **ADS**.

Currently **ADS** are also used by different products to tag files in the *":Zone.Identifier"* stream in order to know when a file is external (i.e. not created on the computer itself) and should therefore be analysed by **SmartScreen**. Because of this feature, Microsoft began tagging all files downloaded from Internet Explorer (at the time), other browser developers began doing the same to take advantage of **SmartScreen**'s protection.

The value written to the Stream, i.e. the ZoneId, can have any value. However, **SmartScreen**'s behaviour is governed by the values shown in Table 1:

Identifier	Value
URLZONE_LOCAL_MACHINE	0
URLZONE_INTRANET	1
URLZONE_TRUSTED	2
URLZONE_INTERNET	3
URLZONE_UNTRUSTED	4

Table 1. Source: https://docs.microsoft.com/en-us/dotnet/api/microsoft.visualstudio.ole.interop.urlzone.

The most interesting point to note from the table is that the **SmartScreen** component will only act if ZoneId value = 3.

It is known that the programs that take into account ZoneId in order to transmit information about the origin of files are Microsoft Edge, Internet Explorer and Outlook but, do other popular programs that download files from the Internet take advantage of it?

Performance analysis

We have raised a series of questions and analyse each case to better understand the effectiveness of the protection provided by the operating system itself against potential malicious files downloaded to the computer. The first question, and perhaps the most obvious one, assuming that the solution was designed to protect us when downloading a file via the native browser, is:

What other browser should have that behaviour? That is, which ones tag the downloaded file with ZoneID = 3?

We analysed the 10 most used browsers in desktop operating systems, according to NetMarketShare[2]. To do this, we install the latest version of each by January 2020 and download a file from a website. **Is the ZoneId added to the downloaded file?**

As can be seen in Table 2, the vast majority do tag downloaded files with ZoneID = 3. Although regarding Internet Explorer (currently unsupported), the original implementation only covered the tagging of executable files but not of the rest of the files. The only browser on the market today that completely ignores this feature is Baidu. Only 10% do not implement ZoneID = 3 for downloaded files, required for SmartScreen to analyse them.

Browser	ZoneId Added
Chrome	Yes
Firefox	Yes
Internet Explorer	For executable files only
Edge	Yes
Sogou Explorer	Yes
Opera	Yes
QQ	Yes
Yandex	Yes
UC Browser	Yes
Baidu	No

Table 2. Use of ZoneId by the most popular browsers on the market

What about Email Clients?

To answer this question, we searched forums and portals to find out the most popular email clients during 2019, as there is no official list of the most used browsers. We sent a document by email and determined if the attachment was tagged with ZoneID = 3 when saved to disk.

The result is shown in Table 3. Less than 50% of email clients tagged files with ZoneId = 3.

[2] https://netmarketshare.com/

Email Client	ZoneId Added
Outlook	Yes
Windows Live Mail	Yes
Thunderbird	Yes
eM Client	No
Mailbird	No
Mailspring	No
Sylpheed	No

Table 3. Use of ZoneId by the most popular email clients on the market

It seems that this protection is less implemented in email clients than in browser clients. Given that there are hardly any contraindications to its use, **is it possible that software developers may not be aware of this feature?** Bearing this in mind, the next question was a bit wider: **What other software could be used regularly by users to download files from the Internet?**

That's how we decided to analyse the most common type of software that current users would have installed on their computers, software that would allow them to download potentially malicious software and SmartScreen wouldn't even analyse for not having ZoneID = 3:

- Instant Messaging Clients
- FTP Clients
- Code Versioning Software
- Cloud Synchronisers

Instant Messaging Clients

As for the instant messaging clients studied, 42% do not implement ZoneID = 3 for downloaded files – required for SmartScreen to analyse them.

Instant Messaging Client	ZoneId Added
Telegram Desktop	No
Pidgin	No
Signal	Yes
Riot	Uses Internet Explorer
Slack	Yes
Wire	Yes
Discord	Uses default browser
Viber	No
WhatsApp	Yes
Line	No
Skype	No
Microsoft Teams	Yes

Table 4. Use of ZoneId by the most popular instant messaging clients on the market

FTP Clients

With regard to the FTP clients studied, 86% do not implement ZoneID = 3 for downloaded files – required for SmartScreen to analyse them.

FTP Client	ZoneId Added
FileZilla	No
WinSCP	No
Cyberduck	No
Coffee Cup Direct FTP	No
Solar Putty	No
MonstaFTP	Uses browser

Table 5. Use of ZoneId by the most popular FTP clients on the market

Code Versioning Clients

As for the code versioning tools studied, 100% do not implement ZoneID = 3 for transferred files – required for SmartScreen to analyse them.

Code Versioning Tool	ZoneId Added
Github Client	No
Git-scm	No
TortoiseHG	No
TortoiseSVN	No

Table 6. Use of ZoneId by the most popular code versioning clients on the market

Cloud Synchronisation Clients

Regarding the cloud synchronisation clients studied, 100% do not implement ZoneID = 3 for downloaded files – required for SmartScreen to analyse them.

Cloud Sync Client	ZoneId Added
Insync	No
Dropbox	No
Google Sync	No
OneDrive	No
GoodSync	No
Odrive	No

Table 7. Use of ZoneId by the most popular cloud sync clients on the market

File Transfer

We also decided to test what happens with the most traditional file transfer mechanisms when a file is copied through a network share, transferred via Bluetooth, or when using PowerShell WebClient with Microsoft's own utilities. The result can be found below:

File Transfer Mechanism	ZoneId Added
Microsoft Bluetooth	No
Shared Folder	No
Powershell WebClient	No

Table 8. Use of ZoneId by other file transfer mechanisms

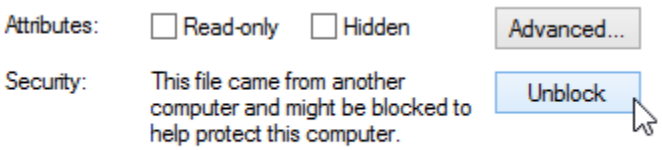
Potential evasions

After understanding how and when the file is tagged, the research led us to reflect on **which process is responsible for running SmartScreen and whether there are ways to bypass that process.** To conduct the test, we mostly tagged files that were interpreted and known by SmartScreen as malicious to find out whether or not the file executed in this way was bypassing SmartScreen. We took a series of files in different interpreted languages and set the bit.

It is simple:

```
echo [ZoneTransfer] > knownmalicious.py:Zone.Identifier
echo ZoneId=3 >> knownmalicious.py:Zone.Identifier
```

To find out if it has been activated, the following message will appear in the file properties:



If you delete the attribute or check the "Unblock" option, that security warning will disappear from properties.

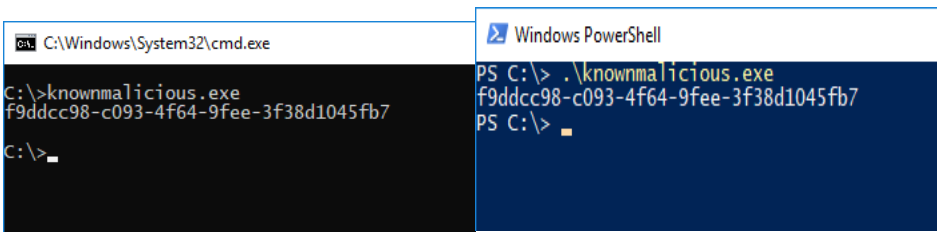
The result can be seen in the following table:

Action	Result	Comments
Double click on malicious .exe	Do not bypass SmartScreen	
Double click on malicious .vbs	No SmartScreen action	Warning displayed (see illustration below)
Double click on malicious .vbe	Do not bypass SmartScreen	

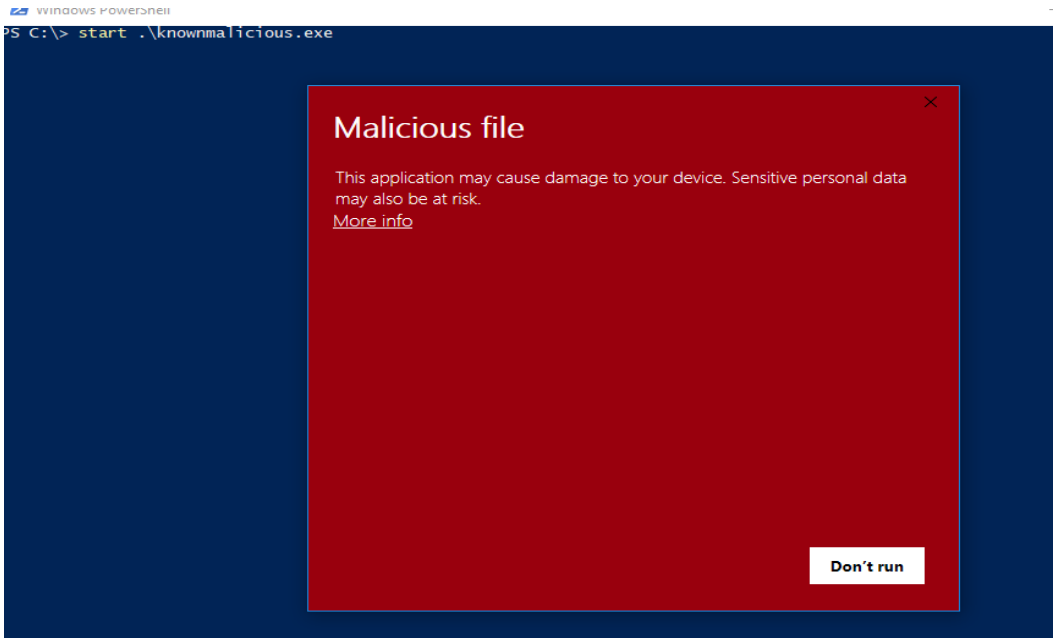
Double click on malicious .bat	Do not bypass SmartScreen	
Double click on malicious .msi	Do not bypass SmartScreen	
Double click on malicious .com	Do not bypass SmartScreen	
Double click on malicious .cpl	Do not bypass SmartScreen	
Double click on malicious .jse	Do not bypass SmartScreen	
Double click on malicious .scr	Do not bypass SmartScreen	
Double click .wsf	No SmartScreen action	Warning displayed
Double click .py	No SmartScreen action	
Launched with *start* from cmd console	No SmartScreen action	
Launched with *start* from powershell console	Do not bypass SmartScreen	
Directly executed in cmd	No SmartScreen action	
Directly executed in powershell	No SmartScreen action	
Double click on Shared Folder	No SmartScreen action	Warning displayed
Running from context menu	No SmartScreen action	
Running from Task Scheduler	No SmartScreen action	

Table 9

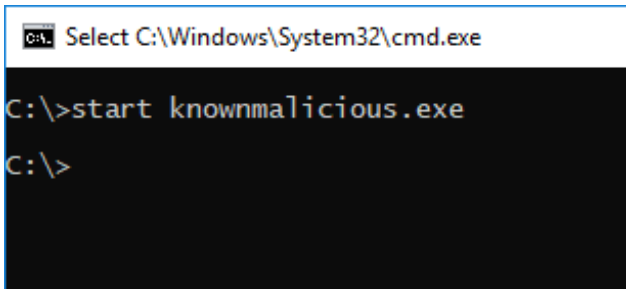
Some examples of how we launched the executables (the code indicates that the malware that should have stopped SmartScreen has been successfully executed):



Perhaps the most interesting point is the difference when launching them by using the *start* command:

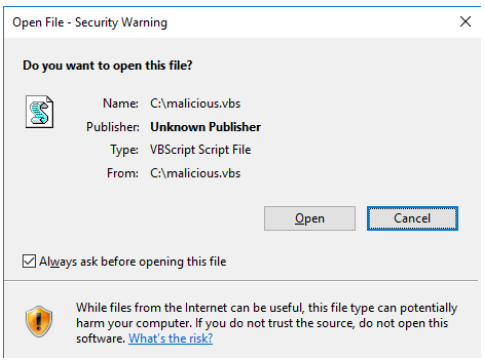


Where SmartScreen gets in the way of PowerShell, but not in the way of CMD.



Note that tagging the executable that interprets programs (python.exe, cmd.exe, cscript.exe, powershell.exe) has no influence. The table would be the same.

In some cases, the following alert is displayed:



Finally, we tested a type of software commonly used to transfer a file: file compressors. Do they respect tagging? We wished to check if the tag is maintained after downloading, compression, and decompression. Is the ZoneID=3 maintained to be further scanned by SmartScreen after downloading? The conclusions are shown in Table 10.

Of the file compressors and decompressors studied, 75% do not implement ZoneID=3 – required for SmartScreen to scan the files once they are decompressed.

Decompressor	ZoneId Maintained
7zip	No
Windows Decompressor	Yes
PeaZip	No
Ashampoo Zip	No
Zipware	No
Hamster Zip Archiver	No
Winrar	No
Winzip	Yes

Table 10. ZoneId maintained after operation of some compressors/decompressors

Countermeasures & Conclusions

This report has focused on two different approaches. Software developers who respect (or not) the tag that allows Windows to scan files with SmartScreen and the software that, when the file is tagged, calls (or not) this protection.

As for the first approach, in principle there are no particular countermeasures beyond that programmers use more frequently this feature. Consequently, the solution seems to lie in greater dissemination and awareness of the issue so that they begin to implement it. It is important for file downloading programs in general to be aware of the file source table, particularly ZoneId. This allow them to work with SmartScreen in order to improve user security.

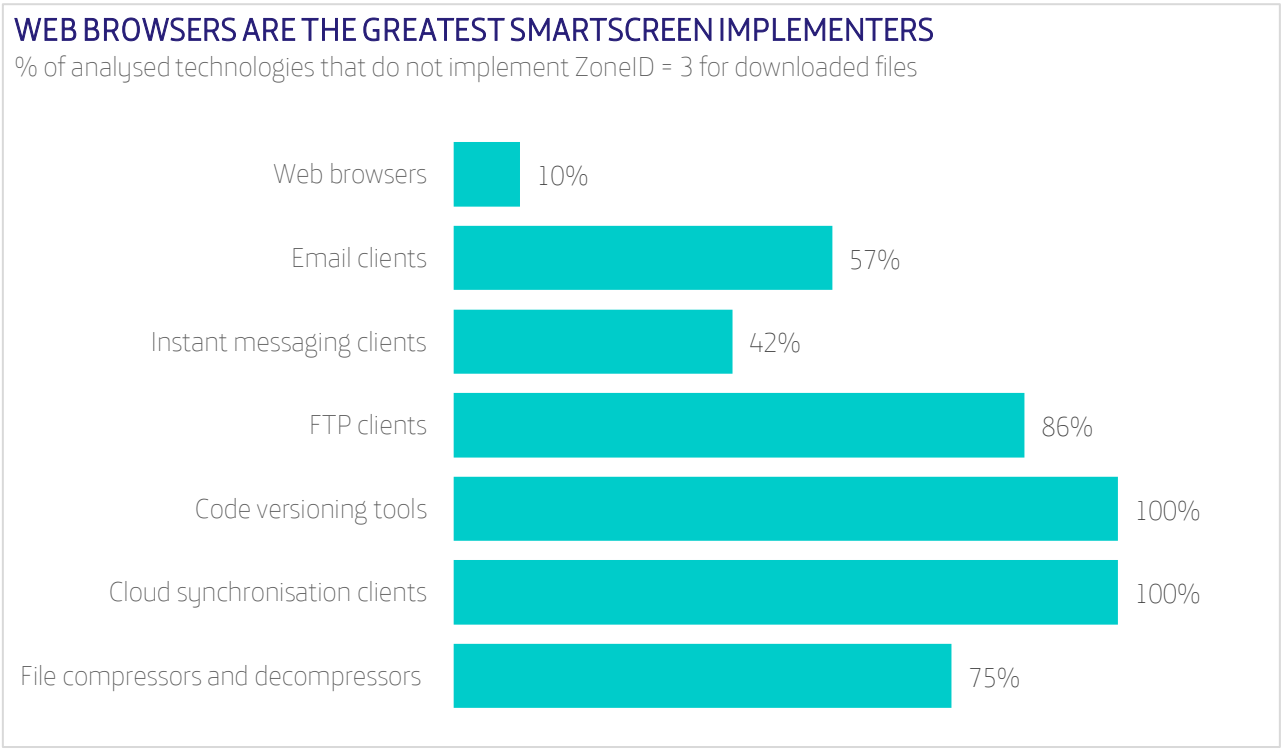
In the image below there is a code snippet implemented by Mozilla Firefox where the ZoneId is embedded in a downloaded file. This is not something too hard to implement. Regarding the second scenario, the program that runs interpreted code could be the one that calls SmartScreen if it finds the tag.

```
try {
  let zoneId = "[ZoneTransfer]\r\nZoneId=" + zone + "\r\n";
  let { url, isPrivate, referrerInfo } = aDownload.source;
  if (!isPrivate) {
    let referrer = referrerInfo
      ? referrerInfo.computedReferrerSpec
      : "";
    zoneId +=
      this._zoneIdKey("ReferrerUrl", referrer) +
      this._zoneIdKey("HostUrl", url, "about:internet");
  }
  await stream.write(new TextEncoder().encode(zoneId));
} finally {
  await stream.close();
}
```

In general, the following conclusions can be drawn from the analysis carried out:

- As for **web browsers, only 10 % do not implement ZoneID = 3** for downloaded files - required for SmartScreen to analyse them.
- Of the **email clients analysed, 57 % do not implement ZoneID = 3** for downloaded files - required for SmartScreen to analyse them.
- Regarding **instant messaging clients, 42 % do not implement ZoneID = 3** for downloaded files - required for SmartScreen to analyse them.
- With regard to **FTP clients analysed, 86 % do not implement ZoneID = 3** for downloaded files - required for SmartScreen to analyse them.
- As for **code versioning tools analysed, 100 % do not implement ZoneID = 3** for transferred files - required for SmartScreen to analyse them.
- Regarding **the cloud synchronization clients analysed, 100 % do not implement ZoneID = 3** for downloaded files - required for SmartScreen to analyse them.

- Of the **file compressors and decompressors analysed, 75 % do not implement ZoneID = 3** – required for SmartScreen to scan the files once they are decompressed.



In this way, it can be concluded that a potential attacker would have several ways of getting a malicious file onto a machine with greater guarantees of not being discovered by **SmartScreen**: by trusting users to download executables through certain programs or by forcing them to do so through programs that do not implement it at all.

Developers need to be aware of how **SmartScreen** works to take advantage of its detection capabilities and better protect the user.

About ElevenPaths

At ElevenPaths, Telefónica's Cybersecurity Company, we believe in the idea of challenging the current state of security, an attribute that must always be present in technology. We are always redefining the relationship between security and people, with the aim of creating innovative security products which can transform the concept of security, thus keeping us one step ahead of attackers, who are increasingly present in our digital life.

We combine the freshness and energy of a start-up with the power, experience, and robustness of Telefónica to provide solutions that enable prevention, detection, and response against everyday threats in our digital world.

We build strategic alliances to provide a strengthened security to our clients. Moreover, we work jointly with organizations and entities such as the European Commission, Cyber Threat Alliance, ECSO, EuroPol, Incibe, and the Organization of American States (OAS).