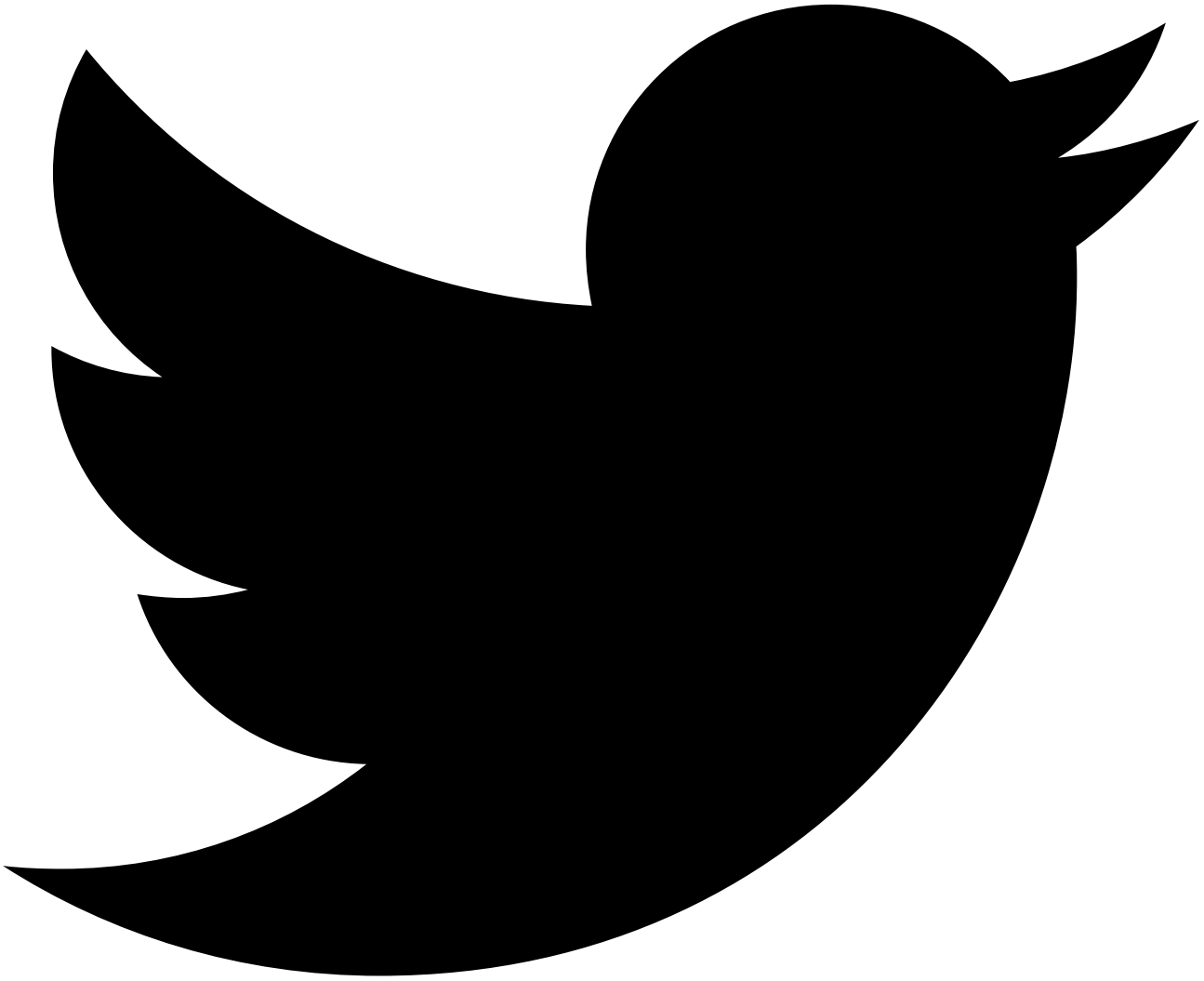
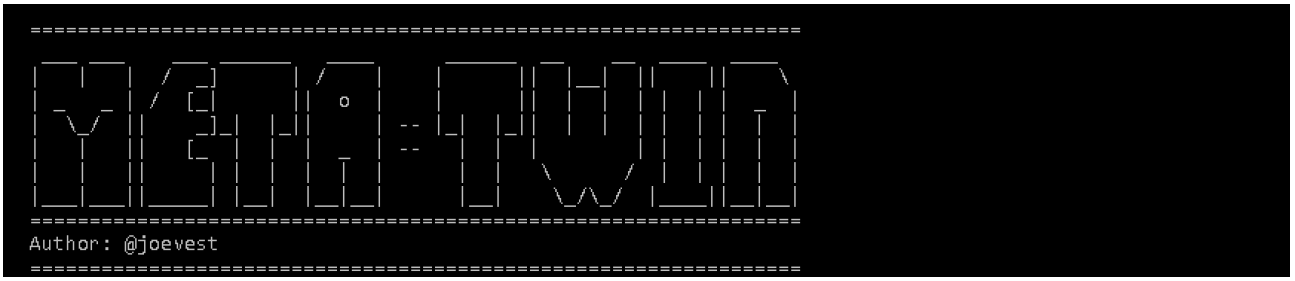
Borrowing Microsoft MetaData and Signatures to Hide Binary Payloads

Joe Vest | October 9, 2017 | Tweet This Post:





Overview¶

A twitter post by Casey Smith (@subtee) inspired me to update a tool written by Andrew Chiles (@andrewchiles) and I a few years ago.

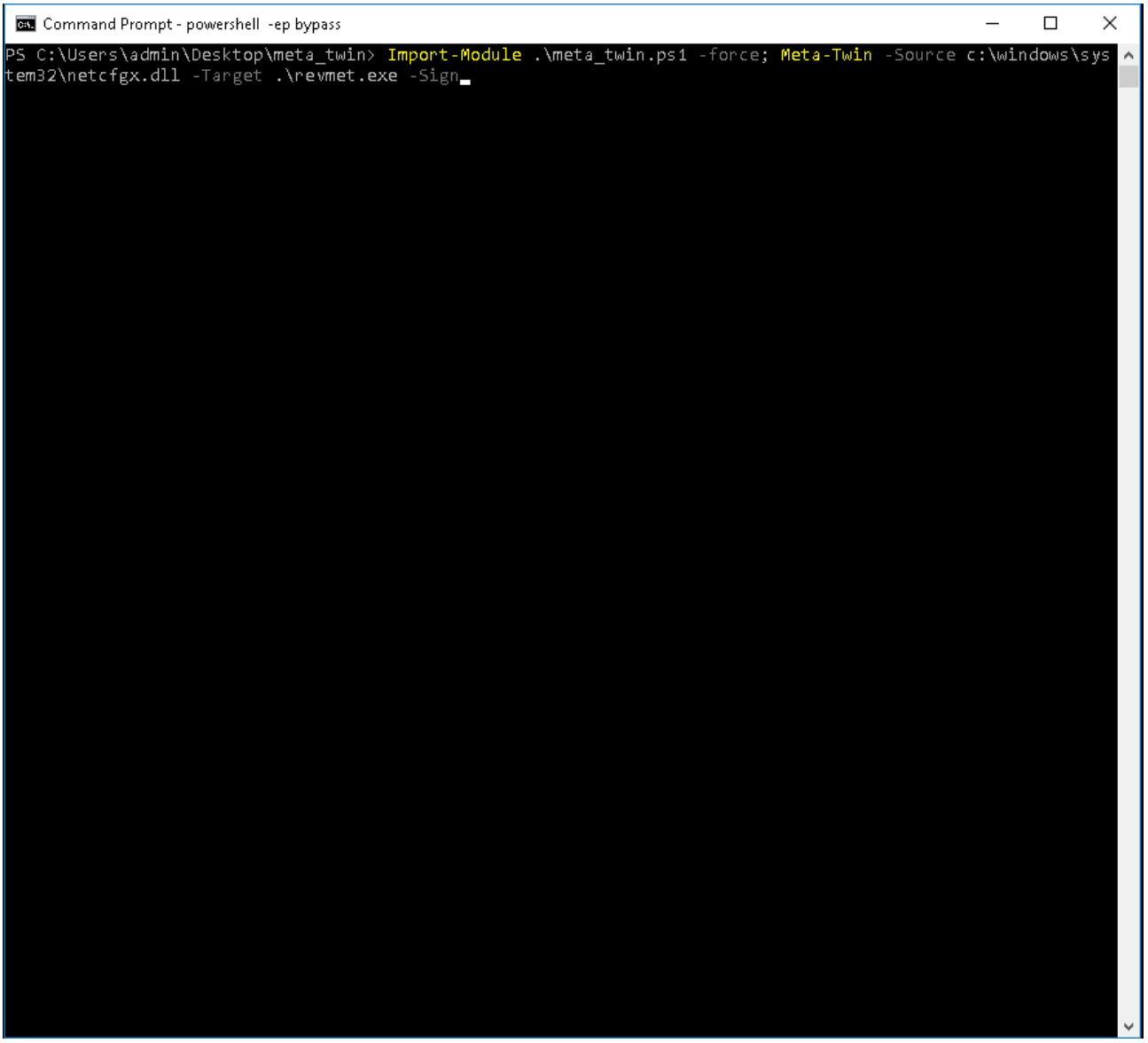
During a Red Team engagement, it can be helpful to blend in with the environment as best as possible when forced to operate from disk. Operating in memory is great, but in many situations or scenarios, you must resort to binaries on disk. A technique I've used with great success is to modify a binary's resource information (metadata). This includes fields such as file icons, version, description, product name, copyright, etc. When defeating security defenses or managing IOCs (See my SANS Breaking Red webcast series for more on IOC management), a threat will often attempt to trick or deceive an analyst. Making files blend into the environment can cause an analyst to treat malicious behavior as trusted. If a binary says is it from Microsoft, it must be…

This is where MetaTwin comes into play. This is rewritten to not only modify a binary's metadata, but also add a digital signature as recently described by @subtee and @mattifestation.

1. MetaTwin starts with a legitimate signed source binary, such as explorer.exe
2. Extracts the resources (via ResourceHacker) and digital signature information (via SigThief)
3. Writes the captured data to a target binary

Demo¶

In this example, I'm simply using a default meterpreter reverse_tcp binary. Nothing special here, use any binary (.exe or .dll). Personally, we're huge fans of Cobalt Strike during real engagements.



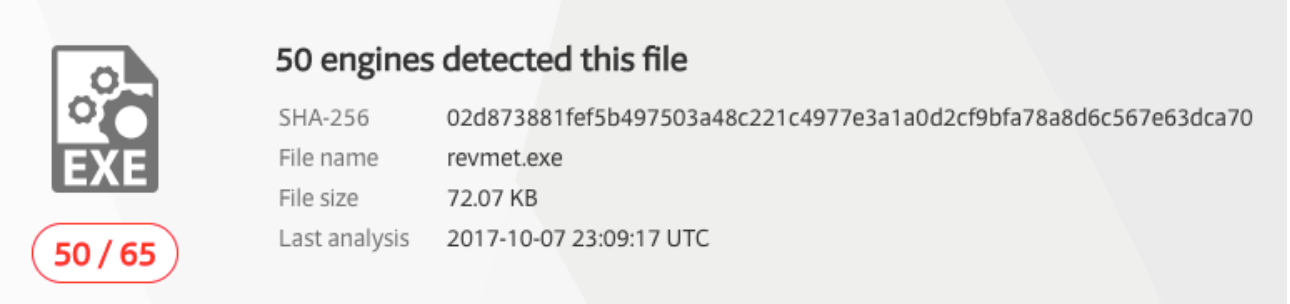
As you can see, the file looks and feels like it could belong there. Storing this in a location such as c:ProgramData... with a modified time stamp, **could** buy a Red Team operator a bit of time and support long(er) term persistence.

Interesting Observations¶

AntiVirus¶

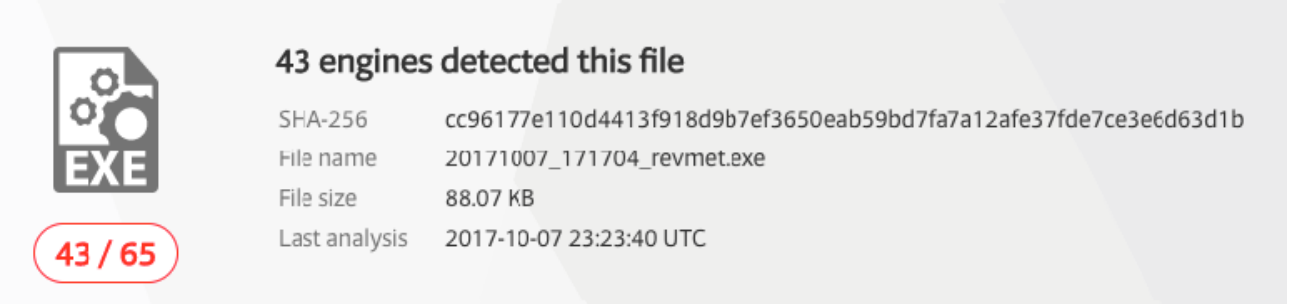
Often simple modifications can cause defensive tools to react in different ways. Of course AV is often not a show stopping defensive tool, but we were curious as to how AV handled a default Metasploit meterpreter binary when modified with MetaTwin. No obfuscation other than the addition of metadata and digital signatures. The results were interesting…

Default Reverse TCP Meterpreter Binary¶



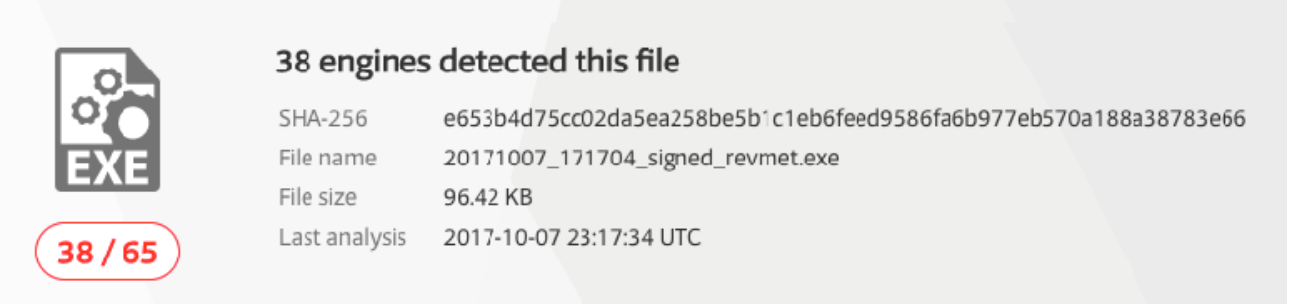
As expected, VirusTotal reported several hits

Metadata added to Reverse TCP Meterpreter Binary¶



Interestingly, adding metadata alone reduced the AV detection rate.

Metadata and Digital Signature added to Reverse TCP Meterpreter Binary¶



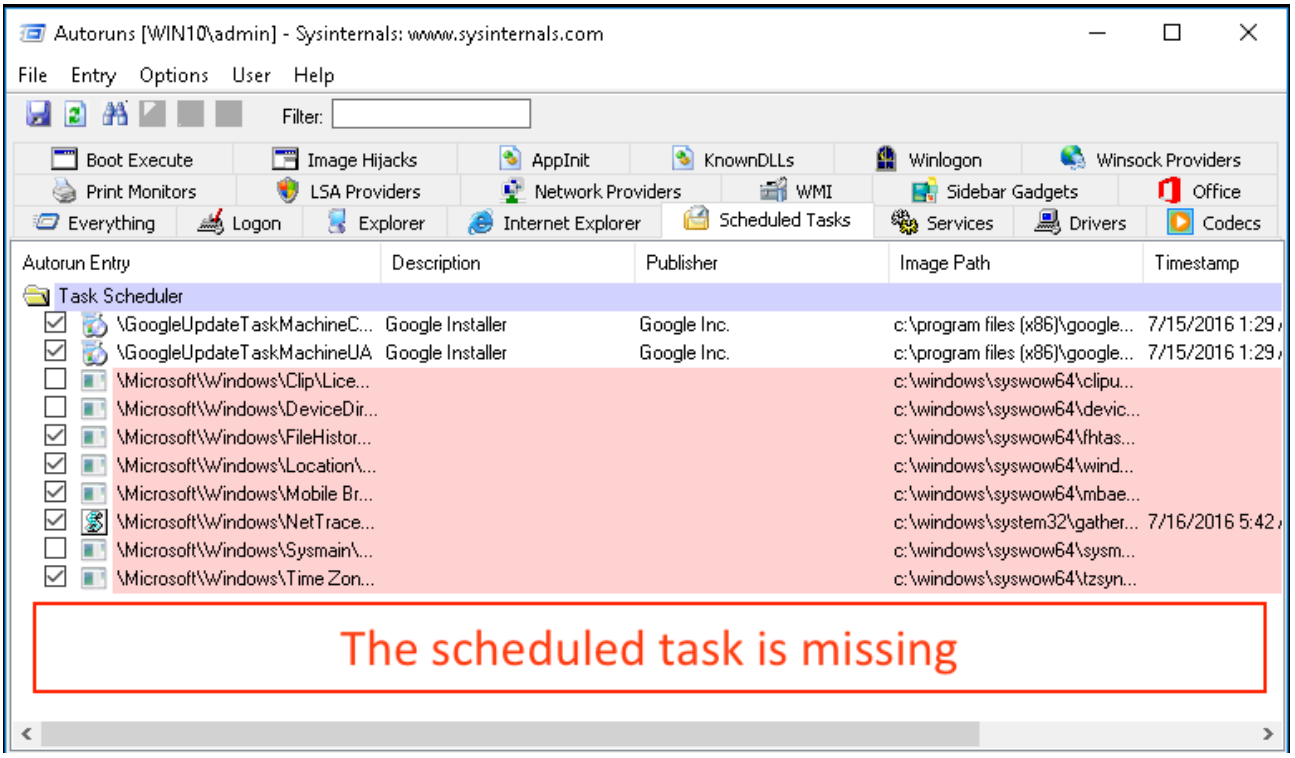
After adding a digital signature and the metadata, exposure dropped from 76% to 58%. This is important because we're not even trying to evade AV!

SysInternals AutoRuns¶

In additions to Antivirus, you can see how default tool behavior responds to these modifications using SysInternals AutoRuns.

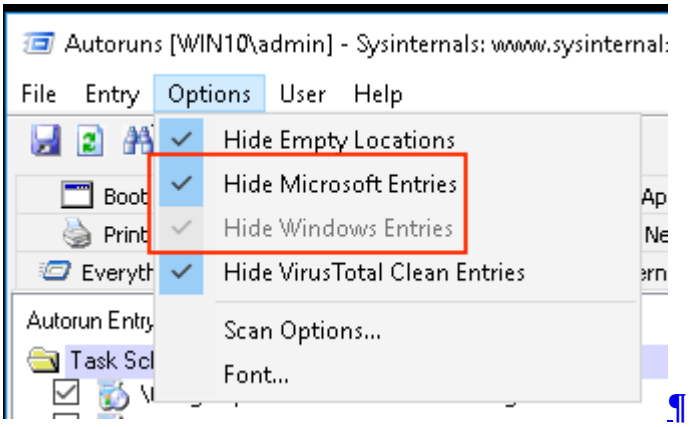
Using the modified binary, we created simple persistence mechanism using a scheduled task. AutoRuns can be used to display this type of Windows persistence. But… the modified binary is hidden by default. Take a look…

AutoRuns Default Settings Hide the "Microsoft" scheduled task¶



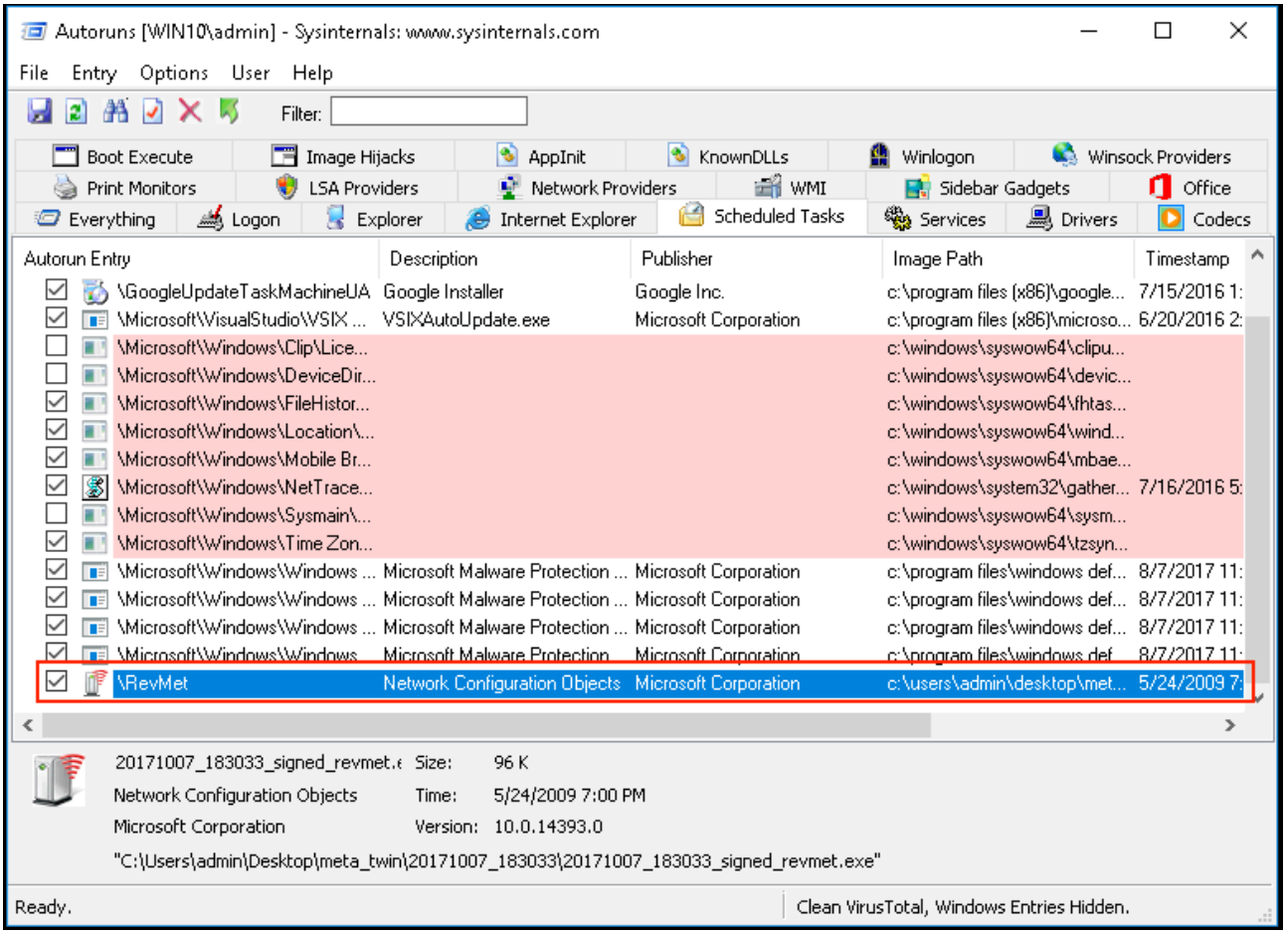
¶

AutoRuns Default Options



¶

Changing the Default Reveals the "Microsoft" scheduled task



¶

Takeaway¶

Based on these observations, it's clear that some AV and EDR tools make poor assumptions based on file metadata and digital signatures that can make them less effective or confuse an inexperienced Blue Team member. Red Team operators can use this to their advantage if forced to operate from disk in future engagements.

Source: https://threatexpress.com/blogs/2017/metatwin-borrowing-microsoft-metadata-and-digital-signatures-to-hide-binaries/