HowTo: Determine/Detect the use of Anti-Forensics Techniques

Archived: 2026-04-06 00:31:20 UTC

The use of anti-forensics techniques to hide malicious activity (malware installation, intrusion, data theft, etc.) can be something of a concern during an examination; in fact, in some cases, it's simply assumed when particular data or artifacts can't be found. It's easy to assume that these techniques were used when we look at a very limited range of artifacts; however, as we begin to incorporate additional and more comprehensive data sources into our analysis processes, we begin to be able to separate out the anti-forensics signal from the noise.

The term "anti-forensics" can refer to a lot of different things. When someone asks me about this topic, I generally try to get them to describe to me what they're referring to, and to be more specific. As with anything else, nomenclature can be important, and messages get scrambled when the use of terms becomes too loose. Rather than address this as broad topic, I thought we'd take a look at some of the common techniques used to hide evidence on or remove it from a system...

TimeStomp

One of perhaps the most publicly discussed anti-forensic techniques is referred to as *time stomping*, in part due the name of the tool used to demonstrate this capability. While this initially threw a monkey wrench into our analysis processes, it was quickly realized that the use of this sort of technique (and tool) could be detected. Then, as things tend to go in any eco-system, there was an adaptation to the technique...rather than modifying a 64-bit time stamp with a 32-bit value, the technique was adapted to copy the file times from kernel32.dll onto the target file, preserving 64-bit granularity. Once again, analysis techniques were updated. For example, in 2009, Lance Mueller talked about detecting the use of time changing utilities in his blog. There's been discussion regarding techniques for changing the $FILE_NAME attribute time stamps, as well as those within the $STANDARD_INFORMATION attribute, just as there have been techniques for detecting the use of this technique. Direct and thorough analysis of the MFT (the analysis of which is predicated by having a thorough understanding of the MFT records themselves) can be revealing, whereas techniques such as detecting program execution and David Cowen's NTFS TriForce can prove valuable insight, as well.

Tools: MFT parser, knowledge of MFT records

Changing the System Time

Okay, let's say that rather than changing the times of specific files, and intruder changes the system time itself. This would mean that, after that change, the times recorded by the system would be different...so how could we

detect this? One way to do this is to list available Event Log records by sequence number and generated time...if the system time were rolled back, this activity would become evident as the sequence numbers increased but at one point, the time generated was earlier than the time for the previous record. Lance Mueller's first forensic practical exercise provided a great example of how to detect system time changes using this technique.

Tools: evtparse.pl ('-s' switch)

Zapping Event Records

I've heard analysts state the there were gaps in the available Event Logs, so an intruder must have been able to remove specific event records from the log. Again, I've heard this claimed, but I've never seen the data to support this sort of thing. Writing a tool to do this is hazardous to the intruder...it may not work, and may instead crash the system. Why not just do something much simpler, such as (given the appropriate privileges) clear the Event Log and disable auditing all together.

I've had to analyze a number of Windows systems where the Event Logs have been cleared, and with Windows XP and 2003 systems in particular, it's been pretty trivial to recover a good deal of those deleted event records.

Checking the LastWrite time of a Registry key within the Security hive file (see the *auditpol.pl* RegRipper plugin) will help you determine when the audit policy of the system was last modified.

Multiple Techniques

What we've discussed thus far was not intended to be a comprehensive listing of all anti-forensics techniques; rather, I wanted to look at a couple and point out analysis processes that you could employ to detect the use of such techniques. The thing about using anti-forensics techniques is that less is better; the fewer and more simple the techniques used, the harder they are to address. For example, simply deleting a file...downloader, executable file, etc...after us is perhaps the simplest technique to use, as it prevents an analyst from obtaining a copy of the file for analysis. Say a script downloads and executes a file, then deletes it...the analyst may still find artifacts to indicate that the file was executed (i.e., Prefetch file, AppCompatCache artifacts, etc.) but not be able to determine explicitly what the file was designed to do.

However, to use multiple techniques requires additional planning and effort. If this is done automatically, then either a larger application, or multiple applications will need to be downloaded to the system. The problem that the intruder then runs up against is that the applications have to be tested specifically against the version of Windows that has been compromised...different versions of Windows may have different functionality behind the API, and the applications may not work correctly, or may even crash the system. The more "sophisticated" the technique used, the more planning and effort is required. If multiple applications are used, it's more likely that indications of program execution will be found. If a more manual approach is used, then the intruder must spend more time and engage with the system more, again leaving more tracks and artifacts as they interact with the environment.

Summary

The key things to remember with respect to determining or detecting the use of anti-forensics techniques are:

1. If you suspect it, provide it. Find the evidence. If you suspect that a particular technique has been used, gather the data that supports, or ultimately disproves, your theory. Don't just wave your hand and suggest that "anti-

forensics techniques were used." If you suspect that one or more techniques were used, identify them explicitly. Then, you can pursue demonstrating or disproving your theory.

2. Remember that you're not only on the same battlefield as the bad guy, but you actually have an advantage. You're examining an acquired image, which is the "scene of the crime", frozen in time and unchanging. You can go back and start your analysis over again, without the fear of loosing any of your previous artifacts.

3. Document your analysis; if you didn't document it, it didn't happen. Once you've documented your analysis, including both what worked and what didn't, you can then incorporate your findings into future analysis, as well as share your finding with other analysts.

Source: http://windowsir.blogspot.com/2013/07/howto-determinedetect-use-of-anti.html