SPEARTIP®

CYBER COUNTERINTELLIGENCE

Outmaneuver Your Adversary™

OBSERVATIONAL MALWARE ANALYSIS

Dynamic Analysis - Memory Analysis – Trace Analysis

Observational Malware Analysis (OMA) provides a better understanding of malware capabilities, the mission of the attacker, and the effects on the company being targeted.

Kris Bleich, CISSP, EnCE, C|EH

Observational Malware Analysis provides a better understanding of malware capabilities, the mission of the attacker, and the effects on the company being targeted. An analyst armed with this methodology and skillset is a valuable resource to defend against today's most advanced threats.

This paper serves as a high-level summary of a fully integrated forensic approach to identifying today's advanced malware threats with higher confidence, better understanding, and in a more time efficient manner. Much of the information mentioned in the article should be fairly well known to a reader who oversees incident response or forensics teams, and to those who perform such tasks as part of their work experience. This overall approach encompasses known information with a new approach enabled by a technology provided within the HBGary Responder Pro solution. Together, these pieces embody an approach referred to as Observational Malware Analysis (OMA).

OMA allows for the elimination of the need to use scripting tools and other coding skillsets to identify malware and its behaviors. This inherently cuts down on the skills required and time spent on identifying key capabilities of malware that will be described later in this article. In order to fully understand the noted approach, an explanation of the current threat landscape and techniques used by today's advanced threat actors must be noted.

> **"So in a war, the way is to avoid what is strong and to strike what is weak."**
> -Sun Tzu, The Art of War

As new techniques for securing network systems evolve, so does the threat landscape. Those who wish to exploit weaknesses in network systems using malware employ techniques and strategies which were codified by Sun Tzu, specifically in the above passage found in The Art of War, written around 500BC.

Malware is developed much like the software applications used today. Sophisticated teams have the development cycles to create malware packages that seek to persist and perform malicious tasks as instructed by their authors. Malware continues to advance as the security techniques and solutions they look to subvert mature. The authors of malware have adapted to modern security approaches allowing for quicker and more successful attacks.

Malware authors look to develop malicious code based on functionality that can be broken down into the following four general categories, each which describe specific characteristics of the operation of malware.

- Infection
- Propagation
- Mission
- Defense

Infection generally describes the beginning phases of a malware compromise. This describes how a particular piece of malware gains access to target systems for initial execution. Often, infection methods are the most trivial from the attacker's point of view but the most complicated to identify from an analyst's standpoint. In order to truly identify infection methods, an analyst often has to utilize several different tools, conduct detailed analysis of infected systems, and retrieve sets of logs in order to make this determination. Such a task requires the cooperation of multiple branches of the company's security team.

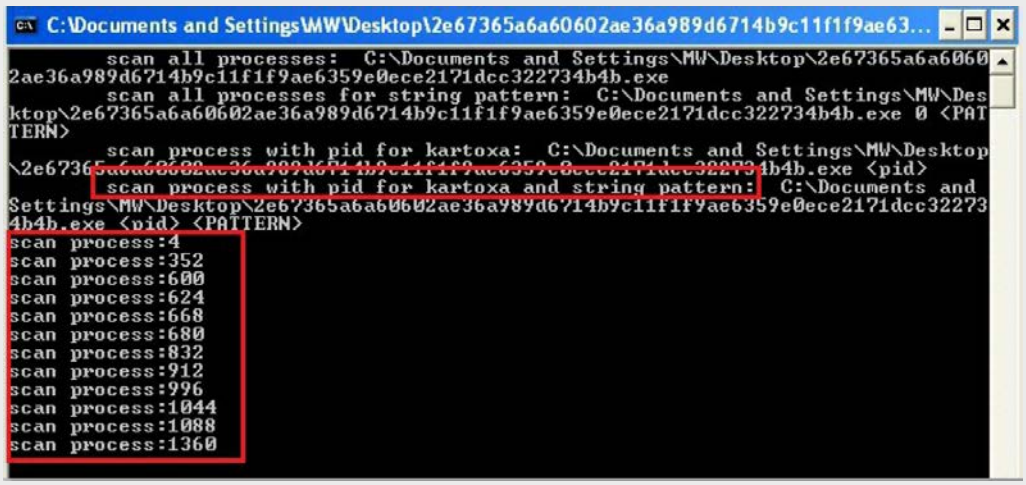
Persistence, or the method by which a piece of malware continues operating following a reboot of the target system, is also contained in this category. Malware must be able to survive long enough to propagate and complete its mission.

Propagation describes the method by which malware moves or replicates from one system to another. Malware may propagate via email, trojanizing files on an infected system or replicating onto external drives or network shares. For many attackers, propagation is key. If propagation of malware across the network is successful, the attacker has increased the amount of time it will take for an analyst or a team of incident responders to identify the true scope of the breach. A method to identify and correlate these methods allows for the most accurate information to be used in identifying the true scope of a breach.

Mission describes the functionality of a piece of malware which focuses on the end result desired by the attacker. The purpose of the malware may be to simply add the infected system to an army of compromised systems or it may involve the compromise of data on the target system, to include financial data, intellectual property or user credentials.

Defense embodies a wide array of malware operation and capability and is arguably the most important aspect to the lifespan of malware. The longer malware is able to persist undetected on an infected system, the more likely it is to complete its mission and propagate to other targets. Therefore, concealment is often a prime concern for malware authors and may take the form of attempting to blend in with legitimate system processes or disabling or circumventing host-based antivirus solutions. In addition, "anti-forensic" techniques are more commonly being employed by malware authors to inhibit "static analysis" of malicious binaries once they are detected on compromised systems. This last category acts as a barrier for the malware, to allow it to carry out the other three categories of functionality: **Infection**, **Propagation**, and **Mission**.

How does OMA apply to the ongoing conflict involving the use of increasingly sophisticated malware? The attackers in this conflict have recognized the value of targeting "Data In Execution" versus "Data At Rest". Attackers have found that targeting sensitive data before it reaches a state of rest increases the chance of obtaining useful data that has not yet been encrypted or protected. A prime and well publicized example of this type of attack was the attack on Target by malware designed to compromise credit card track information in active memory (Data In Execution), rather than attempt to compromise encrypted data on the hard drive (Data At Rest). An analysis of the malware used in the breach of Target systems quickly found that active memory was specifically targeted and "scraped" for information of interest (shown below) and later transmitted outbound.



Attacks like the breach of Target systems continue to damage the reputations of today's largest organizations because threat groups capitalize on the current security practices used. Attackers have knowledge of the skill and time required to identify and categorize data in execution. Malware tools have evolved to ensure sound functionality with a high rate of success.

The analysis of malware must also evolve and begin focusing on "Data In Execution". Malware analysis has traditionally been comprised of two general categories, Dynamic Analysis and Static Analysis. Dynamic Analysis involves the execution of malware and the observations of behavior while Static Analysis typically involves the analysis of a piece of malware using a disassembler and debugger. Traditional Static Analysis relies almost entirely on the analysis of data at rest.

Once a threat is discovered, malware analysts attempt to analyze the threat to identify its capabilities. Having knowledge and understanding of Static Analysis techniques allows malware authors to develop obfuscation techniques and complicate their coding methods, increasing the difficulty of analysis. Defeating static analysis has become the focus of malware authors as they implement malware capability within the category, in an attempt to increase the lifespan of malware attacks.

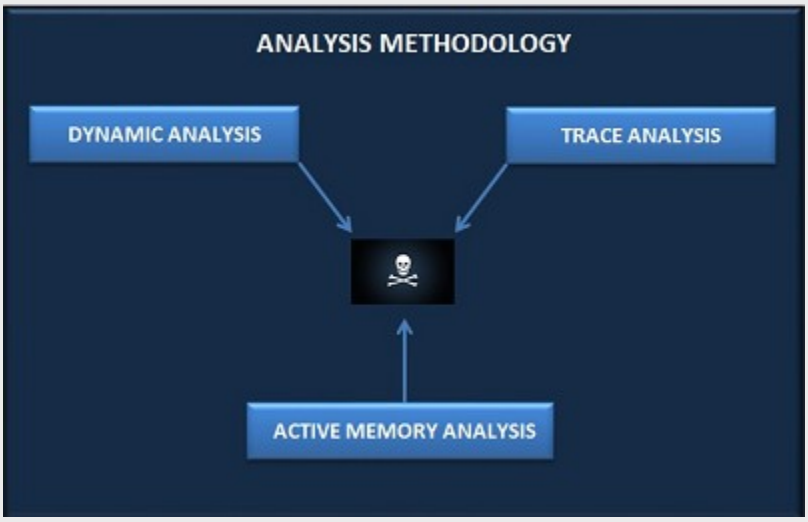
These are some of the more prevalent developer techniques used to counter Static Analysis as a part of the defense strategy:

Custom Packers
Encryption
Code Overwriting
Indirect Jumps
TLS Callbacks (hiding malicious code within the PE Header)

In response to such techniques, a more dynamic and multifaceted malware analysis approach reduces the amount of time required for analysis, while increasing the effectiveness of the analyst. The OMA approach, utilizing traditional Dynamic Analysis, Active Memory Analysis and cutting edge "Trace Analysis", allows for a more holistic analysis of malware by utilizing three differing perspectives. All three "prongs" of this methodology involve the analysis of "Data In Execution" which significantly reduces, and in some cases negates, the effectiveness of the anti-forensic techniques employed by malware authors attempting to implement countermeasures.

In the past, the analysis of data in execution has been difficult, solely relying on an exceptional amount of expertise and known indicators. These two methods have a significant amount of limitations, including the necessity to train an analyst in the craft of understanding memory and its intricate data structures, to having a "bag" of known indicators that will likely be unsuccessful at detecting modern and zero-day threats. In the past, these limitations have often left the analyst without the proper methods to perform analysis of data in execution. OMA successfully alleviates many of the difficulties that come along with analyzing data in execution.

OMA provides a complete "big picture" of a particular malware incident, from specific malware operation and execution to communication and mission. In addition, this multi-layered analysis methodology allows for the quick analysis of malware samples as they operate within active memory, as well as how they interact with the file system on a compromised host. Most importantly, this approach to malware analysis bypasses and negates some of the most prevalent anti-forensic techniques employed by malware authors.



Dynamic Analysis of a malware sample's execution and communication capability, coupled with an analysis of the hard drive of an infected system allows for a more detailed "historical" analysis of the malware, possibly including a determination of infection vector, initial date of infection, and whether sensitive data was compromised during an incident. This process includes the monitoring of a malware sample within a virtual environment or other isolated malware analysis environment as well as the monitoring and analysis of network activity, both within an isolated analysis environment as well as within a compromised network environment. This approach can yield a significant amount of valuable data identifying the capability, communications and sophistication of a malware sample, which can be used to direct the Active Memory and Trace Analysis phases.

The utilization of Active Memory analysis allows for a detailed analysis of a malware sample as it operates within active memory. This analysis of "Data In Execution" focuses on malware's behavior within active memory, including the compromising of other processes, creation of mutexes and network communication activity. In addition, the analysis of active memory activity on a system within a compromised environment allows for the detailed analysis of the malware's capabilities without obtaining or downloading malicious content for static analysis, which may be unavailable when responding to advanced threats. This approach can also detect advanced malware that may evade or compromise security software, such as kernel mode rootkits.

A "Trace Analysis", while similar to traditional Static Analysis, is a dynamic approach that monitors and graphs program behavior, including new threads and processes created as a result of malware execution. This approach focuses on observing and recording "Data In Execution" and unlike Static Analysis, allows for the historical examination of processes spawned or compromised by the original malicious binary. It allows for the quicker identification of what triggers malware behaviors and rapid visualization of malware execution.

The "Trace Analysis" also allows the visualization and analysis of encryption and decryption activity conducted by a malware sample, including the manipulation of decrypted strings in active memory. In the event that a custom packer or advanced encryption algorithm is used to obfuscate a malicious binary, it may take a significant amount of time or be impossible to de-obfuscate the binary for Static Analysis. Because this quasi-dynamic approach focuses on "Data In Execution", it allows for a malware sample to be analyzed despite the use of custom packers, encryption or other anti-forensic techniques employed to foil Static Analysis.

This multi-layered methodology has the benefit of targeting the weaker and less protected aspect of malware, its execution. It provides an analyst with multiple avenues for analysis which complement each other and provide a means to analyze advanced malware which may employ defenses designed to obfuscate the binary and foil Static Analysis, which come in the form of packers, obfuscation or other encryption techniques. With the de-obfuscation of such techniques, the playing field has been leveled allowing an analyst to become more effective.