Advanced malware continues to play a leading role in network breaches. Despite millions being spent on security products, organizations are unable to prevent advanced malware from compromising their networks.

The first step to safeguarding your organization’s data from breaches is to understand how advanced malware operates. This white paper will describe how malware outsmarts many of the most popular ‘next-gen’ malware detection tools in the industry, and what you can do about it.

What is Advanced Malware?

Most of us are familiar with the concept of malware—a broad term applied to many related types of unwanted or harmful objects that can compromise systems, harvest data, and otherwise damage corporate networks. But just what is advanced malware, and how is it different from the older viruses, Trojans, bots, and worms that have been around for decades?

The term advanced malware refers to sophisticated malicious code that has been designed with superior evasion and infection capabilities. Advanced malware avoids being detected and can remain hidden for extended periods of time as it conducts complex cyberattacks. Advanced malware does not necessarily refer to a specific type of malware (such as ransomware). Instead, it describes malware that exhibits sophisticated behavior.

Who is Behind Advanced Malware?

There is no one group or organization behind advanced malware. Instead, there are several types of organizations behind the creation and distribution of advanced malware, with a range of motivations and objectives.

These bad actors include state-sponsored teams, organized crime rings, and military groups. Their objectives include harvesting data they can easily monetize, conducting espionage to steal intellectual property, or targeting users who will pay a ransom quickly.

How Advanced Malware Avoids Detection

Most organizations rely on the malware detection capabilities included in their firewalls, web and email gateways, intrusion prevention systems (IPS) or endpoint products to protect their networks against breaches.

Unfortunately, most of these tools depend on outdated technologies that prevent them from detecting breaches caused by advanced malware, such as:

- Signature-based detection
- VM (virtual machine) environments
- Behavior-based detection
Why Signature-Based Malware Detection Is Ineffective

Conventional malware detection products function by examining each object (such as an email attachment) and calculating its digital signature. If that signature appears in a database of known malware, the product flags the object as malicious. This is an effective method of detecting malware when the signature exists.

Unfortunately, today’s malware authors can alter the signature of their code to avoid detection. Because security tools examine the internal components of an object to generate a signature, modifying even a single bit in any of the malware’s components changes the object’s signature.

There are multiple transformation techniques used by malware authors, and applying any of them can alter a signature:
- Code permutation
- Register renaming
- Expanding and shrinking code
- Insertion of garbage code or other constructs

According to Trend Micro, bad actors create a million new malicious objects every day. Some of these are truly new threats, but most are variations on existing malware. Unfortunately, it can be several days after a new malicious object appears in the wild before security vendors update their signatures (although it’s not unusual for two weeks to pass before a security vendor makes a signature available). Until the new signature arrives, conventional security controls will not detect the malware and organizations are vulnerable to a network breach and data exfiltration during that time.

Perhaps more concerning is that vendors may never add signatures to their databases for many of the advanced malware objects. Malware that uses less-sophisticated techniques (and is therefore relatively easy to detect) and targets large numbers of victims has a much higher chance of being detected and a signature added to a malware database.

Advanced malware, on the other hand, uses more sophisticated techniques and often targets very few victims (such as just one organization or even a few people within one organization). This narrow focus greatly reduces the odds that its signature will ever appear in a database of malicious objects.

Advanced Malware Knows When It’s in a Sandbox

Given the failures of traditional signature-based technologies to detect advanced malware, security vendors began to embrace sandbox technology several years ago. By creating an isolated environment that is separate from an organization’s production network, a sandbox provides additional detection capabilities. Sandboxes simulate a network environment to fool the object into demonstrating malicious behavior, thus allowing the sandbox to identify and block the object before it can compromise systems or applications. Because this method used observed behaviors and not signatures to detect malware, it was very effective in the past in detecting new strains of malware.

However, today’s advanced malware is engineered specifically to detect when it is running in a sandbox. The malware avoids taking any malicious actions to evade detection while in the sandbox, allowing it to enter the network and initiate its malicious behavior.
Sandbox technologies typically utilize VM (virtual machine) environments like VMware, Xen, KVM, Parallels/Odin and VDI. This allows a user or an administrator to run one or more “guest” operating systems on top of another “host” operating system. Each guest operating system executes within a virtual environment and allows managed access to both virtual and actual hardware. In theory, the environment provided by the VM is self-contained, isolated, and indistinguishable from a “real” machine.

Unfortunately, VM technologies insert artifacts, which allow advanced malware to discover that it is running in a virtual environment. These artifacts include additional operating system files and processes, supplementary CPU features, and other components necessary for the virtualization to work.

Advanced malware looks for these artifacts to detect the presence of a sandbox. For example, some of the techniques used by malware to recognize a VMware based environment include:

- Examining registry keys for values that are unique to virtual systems. In VMware, there are over 300 references in the registry to “VMware”.
- Looking to see if VM tools are installed. In a VMware Windows Workstation, there are over 50 references in the file system to “VMware” or “VMX”.
- Checking for certain processes and services that are specific to VM environments such as VMwareService.exe, VMwareTray.exe, etc.
- Identifying the BIOS serial number or MAC address of the virtual network adapter to reveal the vendor. For example, MAC addresses beginning with 00-05-69, 00-0c-29, 00-1c-14 or 00-50-56 are associated with VMware.
- Analyzing specific structures within system memory, such as the Interrupt Descriptor Table (IDT). This table is located in different areas for VM environments compared to physical machines.
- Examining specific hardware parameters that are unique to either VM or real physical environments. Advanced malware may query various attributes like serial numbers or other values belonging to the motherboard, processor, SCSI controller, etc.

It is not only highly skilled hackers who can implement sophisticated techniques like these. Today there are numerous toolkits available that allow novice cybercriminals to create malware that can detect the presence of a VM.
Advanced Evasion Tactics

One of the key characteristics of advanced malware is its level of stealth and ability to evade detection. In addition to defeating signature-based detection products and behavior-based detection tools like sandboxes, there are other evasion techniques advanced malware uses to avoid detection. Table 1 below lists some of the more common tactics:

<table>
<thead>
<tr>
<th>MALWARE EVASION TACTIC</th>
<th>DESCRIPTION</th>
<th>MALWARE RESPONSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stalling Delays</td>
<td>The malware remains idle for an extended period, avoiding all malicious activity.</td>
<td>Ten minutes is usually sufficient for most sandboxes to timeout and assume the object is benign, providing an opportunity for malware to infect a system. Note that most legacy sandboxes can detect if the malware calls the operating system’s sleep function, but if the malware performs the delay internally without calling the O/S, a conventional sandbox will not see the evasive behavior.</td>
</tr>
<tr>
<td>User Action Required Delays</td>
<td>Some malware avoids doing anything malicious until a user performs a specific action (e.g. a mouse click, pressing a key, opening or closing a file, exiting the program).</td>
<td>Malware avoids malicious activity until it sees user action, thereby avoiding detection by a conventional sandbox.</td>
</tr>
<tr>
<td>Intelligent Suspension of Malicious Activity</td>
<td>Unlike simple stalling techniques, this category includes sophisticated evasion techniques that discover the presence of a sandbox and suspend malicious actions to avoid detection.</td>
<td>Malware generally avoid these behaviors until it has penetrated an actual host or machine: • The injection or modification of code within other applications • Attempts to establish persistence and download additional code • Decryption of files • Attempts to move laterally across the network • Connections to its command and control servers</td>
</tr>
<tr>
<td>Fragmentation</td>
<td>A technology that splits malware into several components that only execute when it is reassembled by the targeted system.</td>
<td>When fragments are evaluated separately, (which is typically the case with conventional sandbox technology) each fragment remains dormant, so that the malware appears harmless.</td>
</tr>
<tr>
<td>Return-Oriented Programming (ROP) Evasion</td>
<td>A technique where malware injects functionality into another process without altering the code of that process by modifying the contents of the stack (the set of memory addresses that tells the system which segment of code to execute next).</td>
<td>Malware authors replace the correct return contents of the stack with a specifically-crafted sequence of addresses that changes which code is executed, thus altering the functionality of the program. Malware using ROP evasion avoid discovery by delegating the execution of its malicious code to other programs. Since the malicious activities are not performed directly by the malware program itself, the chances of being detected by a conventional sandbox are greatly reduced.</td>
</tr>
<tr>
<td>Rootkits</td>
<td>A Rootkit is an application (or set of applications) that hides malicious code in the lower layers of the operating system.</td>
<td>Because a conventional sandbox can only monitor calls to the operating system and not what the operating system does with those calls, the malicious actions performed by a rootkit will generally go undetected by a sandbox.</td>
</tr>
</tbody>
</table>

Lastline – A Unique Approach to Detecting Advanced Malware

It is critical that any detection technology remain hidden from the malware to be able to effectively detect advanced malware. Equally important, the technology must be able to detect malicious objects that don’t have signatures, and identify malicious capabilities—even if the malicious code has not yet executed.

Lastline® developed Deep Content Inspection™ specifically to provide complete visibility into malware behavior that other technologies miss, while remaining hidden from the malware itself. Lastline created a unique isolation and inspection environment that simulates an entire host including the CPU, system memory, and all input/output devices.
This approach to sandboxing is unique in the industry and allows Lastline Breach Defender to observe all the malicious actions engineered into a piece of malware, without being visible to the malware. Figure 1 illustrates how Deep Content Inspection provides unmatched visibility down to the hardware level.

Other sandboxes, including those described as "next-generation", only have visibility into the communication that occurs between the malware and the operating system. They can't inspect the actual malware's execution, nor interact with it like Lastline Breach Defender. As a result, sandboxes have significantly lower detection rates and higher false positives than Lastline.

**Yesterday’s Technology Cannot Detect Today’s Advanced Malware**

Advanced malware attacks and data breaches are occurring at an unprecedented rate, despite organizations spending vast amounts of money on security tools that claim to detect advanced threats.

The only way to defeat this sophisticated type of malware is to implement tools that have been designed specifically to detect known evasion techniques and easily adapt to new ones. Unfortunately, today's VM-based sandboxes, web and email gateways, IPS, and firewalls (even next-generation firewalls) are not up to that task.

Organizations that rely on these products for detecting advanced malware run a significant risk of being victims to cyberattacks. Fortunately, Lastline Breach Defender is uniquely suited to detect today's advanced malware.

Experience the Lastline Advantage
For more information please visit www.lastline.com