THE STATE OF SECURITY
News. Trends. Insights.

4 Elements of a Stealthy Sandbox

TRIPWIRE GUEST AUTHORS

APR 7, 2015 | INCIDENT DETECTION

Sandboxes (or automated, dynamic malware analysis systems) are one of the most advanced threat detection tools available to security professionals, and are quickly being adopted by forward-thinking enterprise and mid-market organizations globally.

These systems use behavioral analysis methods to monitor unknown malware programs in a simulated environment for tell-tale signs of foul play. The advantage of this approach is clear — the sandbox has the ability to identify previously unseen or zero-day threats that other systems miss.

But not all sandbox solutions can claim the same capabilities and effectiveness.

Here are 4 things you should consider when building or implementing a sandbox in your environment:

1. INTEGRATION WITH EXISTING SECURITY SOLUTIONS

Any security-savvy person knows that a strong defense strategy employs multiple layers of protection to defend against new threats. Network and host-based controls span all phases of threat protection — from prevention, to detection, to response. When incorporating a sandbox solution into your environment, it is important to consider the ease (or level of difficulty) of integration with solutions from other vendors.

“If you can find a solution that has good network detection and tight integration with endpoint forensics capabilities, that’s a very powerful combination,” explains Gartner industry analyst Lawrence Orans.

Strong, effective and open integrations between solutions can enhance the capabilities of each technology, beyond what they can provide on their own. Beware of sandbox-based solutions with closed architectures that force you to rip and replace other security investments in order to add sandboxing capabilities.

2. HIGH LEVEL OF VISIBILITY INTO MALWARE BEHAVIOR

A sandbox has to see as much as possible of the execution of a program, otherwise, it can miss important activity and cannot accurately detect the presence of malware in an environment.

When monitoring the behavior of a user mode process, almost all sandboxes look at the system call interface or the Windows API. System calls are functions that the operating system exposes to user mode processes, so that they can interact with their environment (e.g. read files, send packets over the network, read a registry entry on Windows, etc.). System calls and Windows API function calls need to be monitored but this is only one piece of the puzzle.

A sandbox that only monitors system calls is blind to everything that happens in between.

Many recently-discovered sophisticated attacks (like Equation, Regin, Dark Hotel, and Turla/Uroburos) execute in the kernel of the operating system – a behavior that lies outside the scope of what can be monitored using traditional hooking mechanisms. These threats are particularly pernicious because their kernel components are running with the highest level of permissions available on a computer system.

Some sandboxes are able to go one step further than just hooking function calls to also monitor the instructions that a program executes between these.

3. RESISTANCE TO EVASION

In the past year, we’ve seen a massive increase in evasive malware behavior designed specifically to bypass first-generation sandboxing systems. If a sandbox does not perform monitoring in a stealthy way, there is a good chance malware will recognize it is under surveillance and alter its behavior to "play nice" and (at least appear to) follow the rules.

Many sandboxes reveal weakness here.

"Most of the current sandboxes on the market use a virtualization approach, where they're running a version of Windows on top of a hypervisor, for example," explains Lastline's VP of Products Brian Laing.

"They're then installing code and files inside the operating system for the detection. This really minimizes what they can see, because they can only see what the operating system will allow them to see. It also means that they have files and processes that are in there that make them detectable."

4. SCALABILITY OF ANALYSIS AND MANAGEMENT

Most people focus very heavily on the detection capabilities of the sandbox system (and rightfully so), but we need to spend just as much time on the scalability aspects of the system, as well.

Not only must it scale to handle any amount of network traffic present but also process any number of files that must be analyzed and manage the number of events generated. Many appliance-based approaches fail to scale, as new boxes must be purchased to handle high volumes of traffic and objects analyzed.

"If your system can handle the files your network sees in a 24-hour period, but all of those files come in during the first eight hours of the day or even less, you may be waiting eight plus hours before you have the information or even knowledge that there's a breach in your network," said Laing.

"And that means the attacker is spending much longer exfiltrating data and spreading laterally around your network."

WHEN IN DOUBT, TEST IT OUT

Some of the architectural concepts and differentiation points between sandboxes can be complicated. When determining the right fit for your environment, make sure to request a proof of concept and spend time thoroughly testing each product. Put two or more sandboxes into your production environment and take note of malware detection rates, manageability and workflow capabilities.

About the Author: *In addition to being co-founder and chief architect at Lastline (@LastlineInc),* ***Engin Kirda*** *is a Professor at the Northeastern University in Boston, and the director of the Northeastern Information Assurance Institute. Before that, he has held faculty positions at Institute Eurecom in the French Riviera and the Technical University of Vienna where he co-founded the Secure Systems Lab that is now distributed over five institutions in Europe and US. Engin's recent research has focused on malware analysis (e.g., Anubis, Exposure, Fire) and detection, web application security, and practical aspects of social networking security. His recent work on the deanonymization of social network users received wide media coverage.*

Editor's Note: *The opinions expressed in this and other guest author articles are solely those of the contributor, and do not necessarily reflect those of Tripwire, Inc.*