Advanced Threats, Advanced Solutions: Integrating a Sandbox Into Your Infrastructure

In computers, the term sandboxing has long been used to represent a safe, isolated environment in which to run malicious code so researchers can analyze it. Network security appliances are now using this concept: they can execute and inspect network traffic and uncover malicious code that would previously slip past traditional security measures. Capable of virtually emulating entire end-user operating environments, a sandbox safely executes suspicious code so its output activity can be observed. Malicious activities—including file operations, network connections, registry or system configuration changes and others—are exposed so threats can be neutralized. Early security sandboxes could only scan executable files (think Windows .exe and .dll files), but advanced platforms are now able to scan many other additional file types, such as Adobe Flash and JavaScript, and Microsoft Office files, among others. Cutting-edge sandboxing solutions today now provide tight integration into the rest of your security infrastructure, allowing you to submit objects, receive results and take protective actions from your established inspection points.

Fighting today's advanced threats requires a multilayered approach: Fortinet FortiSandbox offers you the ultimate combination of proactive mitigation, visibility and rich reporting. It also delivers the power of the Fortinet award-winning antivirus and threat-scanning technology, dual-level sandboxing, and the option of additional integration with the FortiGuard cloud-based Community to deliver state-of-the-art threat protection.

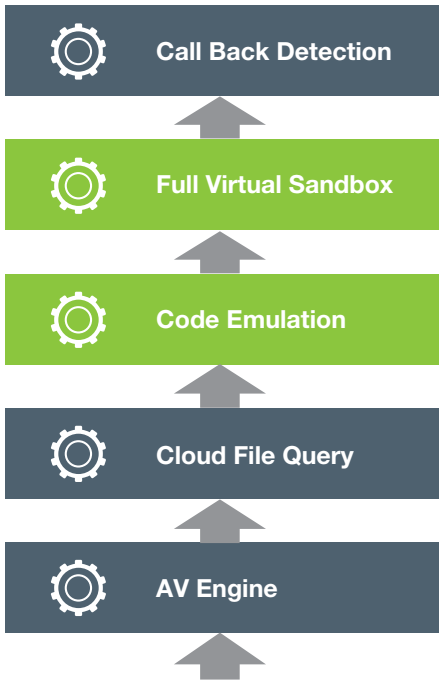

FIGURE 1: FORTISANDBOX TECHNOLOGIES

Why Is Sandboxing Important?

If sandboxing technology has existed for so long, why only now has it been considered an important piece of your security infrastructure? The answer is simple: cybercriminals do not rest on their laurels. They continue to hone their abilities and invest in developing new tools and techniques to deliver malicious wares. Cybercriminals continually discover novel ways of exploiting software and users to spread infections and carry out their goals. Today's advanced sandboxing solutions can help you quickly discover new threats and rapidly mitigate or minimize threats.

Sandbox and Proactive Signature Detection

Sandboxing is a rigorous inspection that can be incredibly resource intensive, especially when following multistage attacks: code needs to fully execute in the sandbox before it can be analyzed, and exploring all code execution paths—possibly including additional modules that malicious code tries to download—takes time. Fortinet combines sandboxing with proactive signature detection to filter traffic before it hits the full sandbox, which is much faster than full sandboxing alone.

Traditional signature detection is reactive, as the signatures are merely fingerprints of threats that have already been seen. Fortinet patented *Compact Pattern Recognition Language* (CPRL) is a deep-inspection, proactive signature-detection technology developed through years of research by FortiGuard Labs. A single CPRL signature can catch 50,000 or more variants of a family of malware. It includes decryption, unpacking and emulation of code for powerful static analysis, which reduces the volume of code that needs full sandboxing. CPRL proactive signature detection helps cast a wider net over the attacks and methods of modern Advanced Persistent Threats (APT) and Advanced Evasion Techniques (AET), preserving full sandbox analysis for the most sophisticated of threats.

APTs and AETs

APTs are custom-developed, targeted attacks. They can evade straightforward detection, using previously unseen (or *zero-day*) malware, exploit vulnerabilities (unpatched security holes), and come from brand-new or seemingly innocent hosting URLs and IPs. Their goal is to compromise their target system with advanced code techniques that attempt to circumvent security barriers and stay under the radar as long as possible. They also include a high degree of social engineering to fool even the most security-conscious end users.

And there are even many ways APTs seek to evade security barriers like sandboxing.

Logic Bombs

Logic bombs are code that remains dormant after installation until a specific trigger occurs. The most common logic bombs are time bombs, and logic bombs have been used in some high-profile attacks in the past. In a time bomb, the malicious part of the code remains hidden until a specified time. The attacker can plant malware on multiple systems, hopefully unnoticed until a certain time, when all the bombs virtually "detonate." Other logic bombs use other human interaction like mouse clicks and reboots to indicate that the bomb is located in a user's computer, and not a sandbox inspection appliance, before running. Logic bombs can be difficult to detect, since the logic conditions are unlikely to be met in the sandbox without heavy instrumentation. In addition to such instrumentation, FortiSandbox can expose logic bombs with its CPRL and code emulation prefilters prior to actually running the code. Real-time analysis of actual operating instructions uncover the logic conditions that will trigger the bomb without waiting for execution.

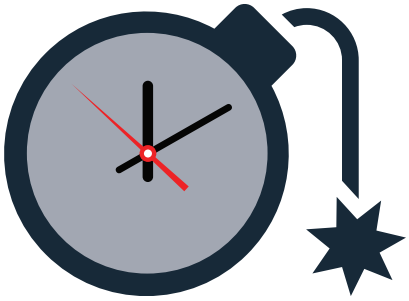

FIGURE 2: LOGIC BOMBS

Rootkits and Bootkits

Advanced malware often contains a rootkit component that subverts the operating system with kernel-level code to take full control of the system. Sandboxes are potentially vulnerable to this evasion technique, since output behavior monitors may also be subverted. Further, rootkits infect the system with malware during system boot-up—something that is typically not observed by a sandbox. FortiSandbox addresses this problem again with CPRL detection—to find advanced rootkit/bootkit routines before they run and hide.

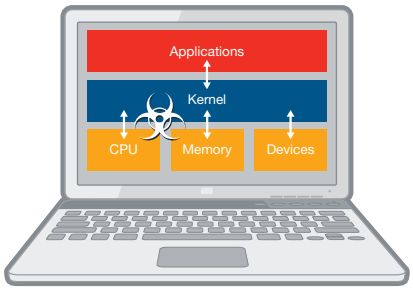

FIGURE 3: ROOTKIT

Sandbox Detection

Another advanced evasion technique is awareness of the sandbox environment itself. APT code may contain routines that attempt to determine if it is running in a virtual (often

sandbox) environment, or may check for fingerprints of specific vendors' sandbox environments. If the code detects that it is in a sandbox, it will not run its malicious execution path. Again, CPRL is able to deeply inspect, detect and capture code that probes for a sandbox.

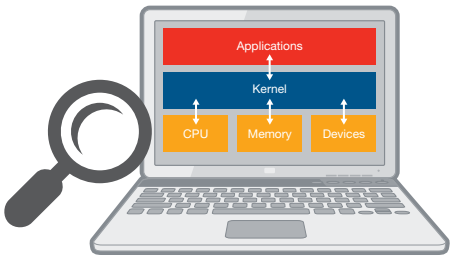

FIGURE 4: SANDBOX DETECTION

Botnet Command and Control Window

Botnet command and control activity typically begins with a *dropper*. The dropper is completely clean code other than a routine that connects to a specific URL or IP address to download a file on command. The command can come from an attacker hours, days or weeks from the initial run time. If the server that the dropper connected to is unavailable or dormant during sandbox analysis, no malicious activity will be observed. CPRL helps catch anomalous code techniques that identify malware without relying on reaching and seeing that window of operation, and the global FortiGuard intelligence network includes botnet monitoring that reveals botnet activity in the field as it happens.

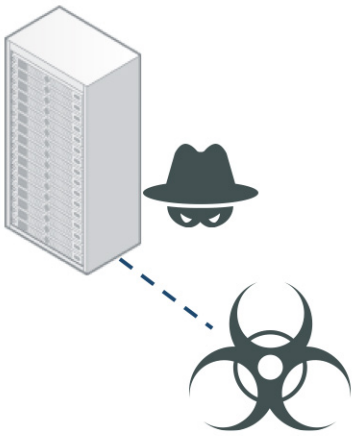

FIGURE 5: C&C WINDOW

Network Fast Flux

Advanced malware may employ *fast flux* or domain generation algorithm (DGA) techniques to change the URL or IP address that an infected host will attempt to connect to in order to avoid reputation-based identification of command and control servers. During sandbox observation, the infection looks for one address, but over time on the infected endpoint, the malicious code will attempt to contact a different address that is brought online at specific times in order to deliver malicious traffic. FortiGuard tracks fast flux networks and feeds the gathered threat intelligence back to the sandbox for use during pre-scans. Fortinet looks at the DNS level, rather than just at IP blacklists, which is more typical with solutions from other vendors.

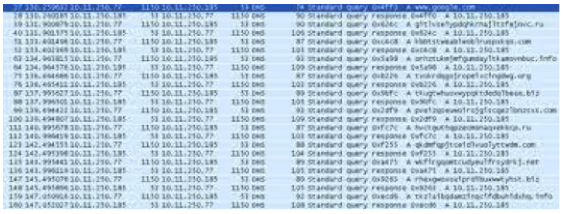

FIGURE 6: DGA

Encrypted Archives

An older but still useful trick is hiding malware inside an encrypted archive, which isn't extractable without the password needed to open the file. With a bit of social engineering, attackers convince the target to open the archive and launch the malware by entering the password. A sandbox can't automatically enter the password, which means the malware won't run during observation and analysis. Fortinet's patented compressed archive header inspection enables detection of malware fingerprints that have been disguised by encryption.

FIGURE 7: ENCRYPTED ARCHIVES

Binary Packers

Binary packers cloak malware by encrypting it in garbled portions that are difficult for traditional antivirus solutions to analyze. The malicious code gets unpacked upon execution and infects the host. Similar techniques are used to embed malicious code in languages such as JavaScript and Adobe ActionScript for Flash. Historically, this technology was used to compress executable code when memory was at a premium. Memory capacity today is no longer an issue, but binary packers are frequently used to circumvent antivirus inspection. In the case of JavaScript and ActionScript, this methodology can legitimately be used for copy protection. The Fortinet antivirus engine supports script de-obfuscation and detection of many binary packers, and also unpacks malware into native form for deep CPRL-based analysis, allowing real-time detection and mitigation or further execution in the sandbox.

FIGURE 8: BINARY PACKERS

Sandbox Replication

Once evasions are addressed, the value of a strong sandbox shines. The goal of sandboxing is to completely replicate the behavior of malicious code seeking entry to the organization. Ideally, the output in the sandbox should be identical to the output of the code like it was run in an end user's environment. In practice, producing identical results is difficult because of the number of variables involved. It's similar to trying to grow two identical plants from seeds; even slight variations in the amount of water, light, temperature and soil composition will produce different results.

Exploits and Applications

Advanced threats can be disguised in document files that trick their application (like Microsoft Word, Microsoft Excel or Adobe Reader) into running malicious code. To reliably replicate this behavior, the sandbox must run through an array of operating systems, each running multiple versions of applications. It's a trial-and-error process that takes time and money. Many sandbox solutions try to address this problem by adding horsepower—bigger CPUs, more virtual machines and more RAM—but those actions are incredibly inefficient and expensive. The better path is to balance the operating systems and applications based on how often they're used and to pick the most fertile environment to trigger malicious behavior.

32- vs. 64-bit, Windows XP vs. Windows 7/8/10

32-bit code can run in both 32-bit and 64-bit environments, so malware authors continue to use 32-bit to maximize their number of infections. The majority of malware today is still in executable files, specifically in Portable Executable 32-bit format (PE32). PE32 files will execute in both Windows XP and 7/8/10 environments, so most malicious behavior can be observed in Windows XP (which doesn't support 64-bit code) without further testing in 7/8/10.

Also, except in very limited cases, Microsoft no longer supports Windows XP. This means Windows XP will stop receiving security patch updates, meaning more and more security holes will open up. Windows XP environments have become even more fertile for infection. The good news is that even more malware will trigger properly in XP in the sandbox. The bad news? People using Windows XP have become massive targets for attackers. You can bet that malware will be developed to go after the low-hanging fruit of end users that haven't switched to 7/8/10. Based on available data, though, there are a lot of Windows XP systems still out there today.

Windows 7/8 Security Mechanisms

Windows introduced security technology in Windows 7/8 that helps stop malicious code and document exploits from executing. Since Windows XP doesn't have the same technology, running the code in XP in the sandbox increases detection, even if the threat is written specifically for 7/8.

Android

Mobile malware is fast-growing, especially in the relatively open and fragmented Android environment. Although security incidents stemming from mobile have been of limited impact so far, the risk is growing—especially in certain regions of the world. The ability to uncover threats seeking entry via smartphones and tablets that are increasingly part of our environments will increase in importance over time.

Beyond Windows and Android

Further, Fortinet's antivirus engine running on FortiSandbox with CPRL fully supports 32-bit and 64-bit code and multiple platforms: Windows, Mac OS X, Linux, Android, Windows Mobile, iOS, Blackberry and legacy Symbian.

Advanced Integration

Fortinet FortiSandbox provides customers the ability to fully and completely integrate their sandbox appliance into their existing security infrastructure with minimal effort. Solutions from multiple vendors can be difficult to integrate, and in some cases integration is not possible at all. This can lead to greater management overhead, more devices for your security staff to manage and monitor, and will increase the complexity of your network. Your security staff is likely already overburdened—asking them to monitor and respond to yet another unique device with its own unique User Interface and methods of operation increase the chances that a threat will be overlooked or missed entirely.

With minimal configuration, FortiSandbox can fully integrate with your FortiGate, FortiMail, FortiWeb and FortiClient devices. You can configure these devices to send files (or a selected subset of files) to the FortiSandbox for detection, and if the sandbox discovers a file deemed to be malicious or a risk, that rating will be returned to the submitting device for policy-based response. Further, FortiSandbox will maintain ratings from its analysis that can be queried by integrated products for faster response and even create a signature to send back to all integrated products designed (FortiGate and FortiClient today) to received them in order to protect against any further

attack attempts via additional entry points as well as lateral movement. FortiSandbox can also send intelligence back when it discovers a malicious URL used to deliver or execute attacks: once a threat has been discovered, FortiSandbox will send that information to prevent any additional content from being downloaded from that domain.

This degree of information sharing and automated response is not limited to Fortinet devices, however. Default connectors as well as an open, standards-based API allow for information sharing to third-party security products also in use within the environment.

Conclusion

The reality is that malware creators are well aware of all forms of security technology, and those attackers will build disguises and use advanced evasion techniques in the hopes of bypassing security mitigations and tools in order to successfully deliver their malware. Detection comes down to inspecting as many layers as possible through all potential angles of attack. The best approach is a combination of proactive threat prevention (such as Fortinet CPRL) to stop as many threats as possible, including those designed with AETs to fool even the latest advanced detection technologies like sandboxing, while still leveraging those advanced technologies to uncover the most sophisticated custom attacks. Further, tying prevention to advanced detection as a seamless solution to cover all potential attack vectors and facilitate incident response is key. Finally, when such components are delivered by a common threat research group like FortiGuard Labs, response and intelligence updates are ensured, even as the threat landscape continues to evolve. At a minimum, look for solutions and labs that routinely share information and collaborate to ensure that even heterogeneous security infrastructures can be strengthened without leaving gaps.

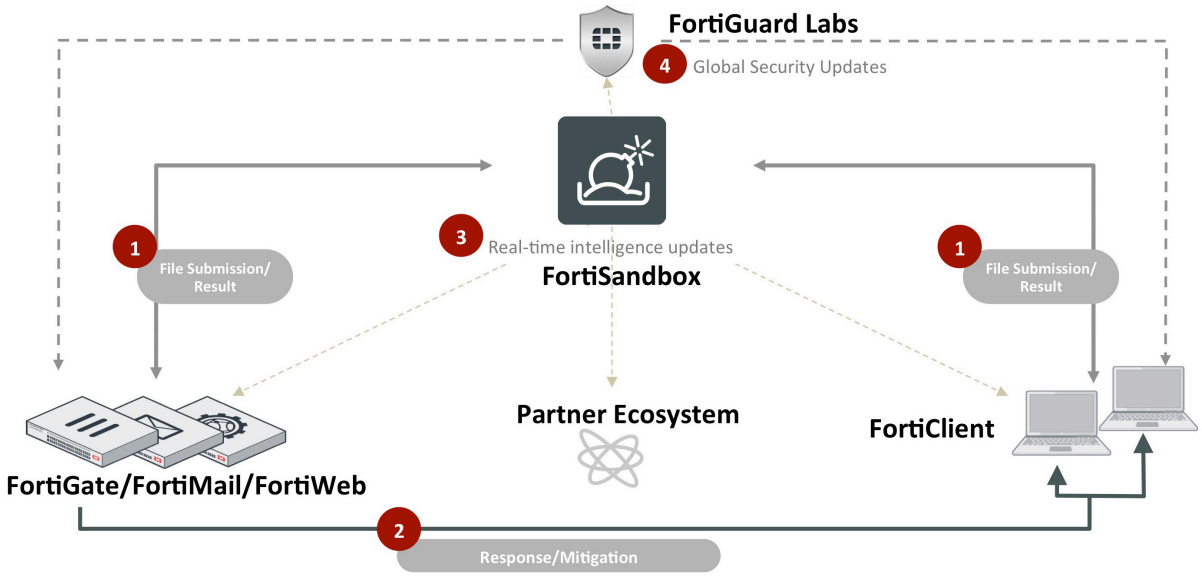

FIGURE 9: INTEGRATED SANDBOX

GLOBAL HEADQUARTERS
Fortinet Inc.
899 Kifer Road
Sunnyvale, CA 94086
United States
Tel: +1.408.235.7700
www.fortinet.com/sales

EMEA SALES OFFICE
905 rue Albert Einstein
Valbonne
06560, Alpes-Maritimes,
France
Tel +33 4 8987 0500

APAC SALES OFFICE
300 Beach Road 20-01
The Concourse
Singapore 199555
Tel: +65.6513.3730

LATIN AMERICA SALES OFFICE
Paseo de la Reforma 412 piso 16
Col. Juarez
C.P. 06600
México D.F.
Tel: 011-52-(55) 5524-8428

Dec 5, 2016