# Head-First into the Sandbox

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. The same concept is now being applied by network security appliances to execute and inspect network traffic, uncovering malicious code that would previously slip past traditional security measures.

Capable of virtually emulating entire operating systems, a sandbox safely executes suspicious code so its output activity can be observed. Malicious activities including file/disc operations, network connections, registry/system configuration changes and others are exposed so threats can be neutralized.

Sandboxes were traditionally used for executable files, but now they're also used to run application data that can contain hidden malicious code, such as Adobe Flash and JavaScript, among others. Some applications, like Adobe Reader X, have their own built-in sandbox for the same purpose. For example, if Reader X encounters malicious code when it opens a PDF, the code is contained in a sandbox and not allowed to infect the operating system.

## Why Sandboxing Now?

If sandbox technology is so old, why is it suddenly so important? Because as cyber criminals figure out more about common methods of security detection, they tend to invest more of their research and development efforts in security evasion. Sandbox technology today can help discover evasive new threats or old threats in new disguises.

## Sandbox and Proactive Signature Detection

But sandboxing is resource-intensive. 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 sandbox, since it's much faster than simply sandboxing alone.

Traditional signature detection is reactive, as the signatures are merely fingerprints of threats that have been spotted "in the wild". Fortinet's 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

> "... as cyber criminals figure out more about common methods of security detection, they tend to invest more of their research and development efforts in security evasion."

50,000 or more disguises a piece of malware can be wrapped in. Combined with 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).

## Advanced Persistent Threats

Advanced Persistent Threats 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.

## Advanced Evasion Techniques

There are multiple ways threats can evade security barriers. These are some common sandbox evasion techniques.

### Logic bombs

The most common logic bombs are time bombs, and they've been seen in high-profile attacks. In a time bomb, the malicious part of the code remains hidden until a specified time. The attacker can plant malware on multiple systems, unnoticed until the appointed time, when all the bombs detonate. Other logic bombs trigger at the evidence of human interaction (mouse clicks, system reboots, etc.), indicating that the bomb is in a user's computer, not a sandbox inspection appliance. Logic bombs are difficult to detect, since the logic conditions are unlikely to be met in the sandbox, so the code doesn't follow the malicious execution path during the inspection. Fortinet crafts the proper environment to expose logic bombs with CPRL and code emulation analysis prior to actually running the code. CPRL performs real-time analysis of actual operating instructions, so the logic conditions that will trigger the bomb can be observed.

### 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, bootkits infect the system with malware during system boot-up — something that is typically not observed by a sandbox. Fortinet addresses this problem again with CPRL detection — to find advanced rootkit/bootkit routines before they run.

### Sandbox detection

Another advanced evasion technique is environment awareness. APT code may contain routines that attempt to determine if it's running in a virtual environment, indicating it might be in a sandbox, or may check for fingerprints of specific vendors' sandbox environment. If the code detects that it's in a sandbox, it won't run its malicious execution path. CPRL is able to deeply inspect, detect and capture code that probes for a sandbox.

### 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 URL/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 dormant during the window of opportunity while the sandbox runs the code, no malicious activity will be observed. CPRL helps catch anomalous code techniques that identify malware without relying on hitting that window of opportunity, and the global FortiGuard intelligence network includes botnet monitoring that reveals botnet activity in the field as it happens.

### Network fast flux

Advanced malware may employ fast flux or domain generation algorithm techniques to change a URL/IP address that an infection will connect to. During sandbox observation, the infection looks for one address, but over time in the end-user's machine, the code will take a different path to a second address that serves up 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 as is common among other vendors.

### Encrypted archives

An age-old trick, attackers can simply encrypt malware in an archive. Then, through a bit of social engineering, they get the end-user to open the infection by entering the password. A sandbox can't automatically enter the password, so the malware won't run during observation. Fortinet's patented compressed archive header inspection enables detection of malware fingerprints that have been disguised by encryption.

### Binary packers

Binary packers cloak malware by encrypting it in garbled portions that can't easily be analyzed by traditional antivirus security. The code gets unpacked when it's executed and infects the host. Similar techniques are used to embed malicious code in languages such as JavaScript and Adobe's ActionScript for Flash. Historically, this technology was used to compress executable code when memory was at a premium. Today memory isn't 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 FortiGate antivirus engine supports script de-obfuscation and detection of many binary packers, and unpacks malware into native form for deep CPRL-based analysis, allowing real-time detection and mitigation or further execution in the sandbox.

### Polymorphic malware

Polymorphic malware changes each time it's run, adding bits of garbage code in an effort to foil pattern and checksum-based inspection. When unpacked by an operating system, the malicious code will execute. Polymorphic code poses a challenge to traditional reactive inspection, which Fortinet tackles through a combination of its proactive antivirus inspection engine technology, CPRL and the sandbox. The engine can unpack the malware into a native format to filter as much traffic as possible with the lowest processing resources prior to running what's left in the sandbox for deeper inspection.

## Sandbox Replication

The goal of sandboxing is to completely replicate the behavior of malicious code. Ideally, the output in the sandbox should be identical to the output of the code if 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.

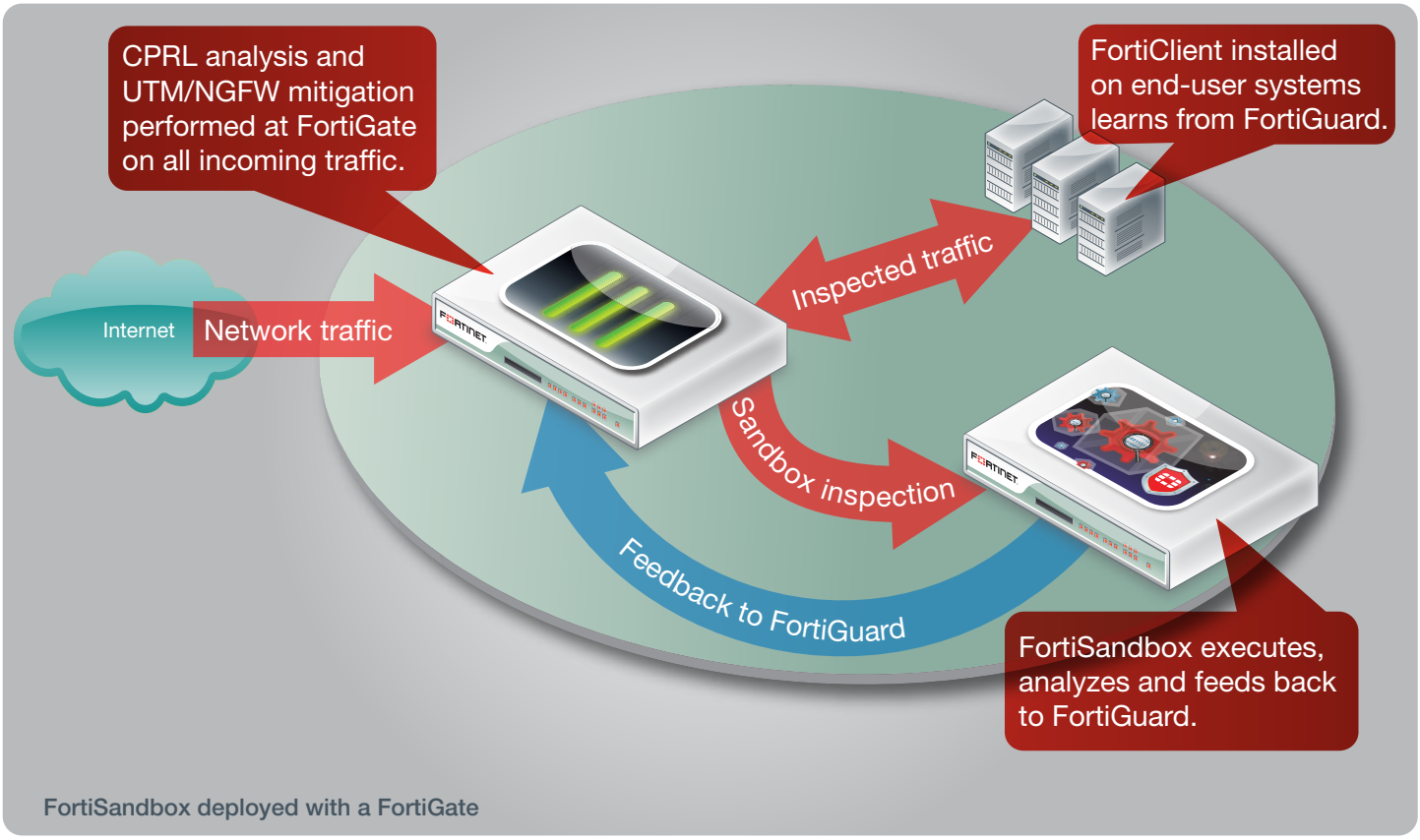
### Exploits and applications

Advanced threats can be disguised in document files that trick their application (like Word, 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, more RAM. But that's 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.

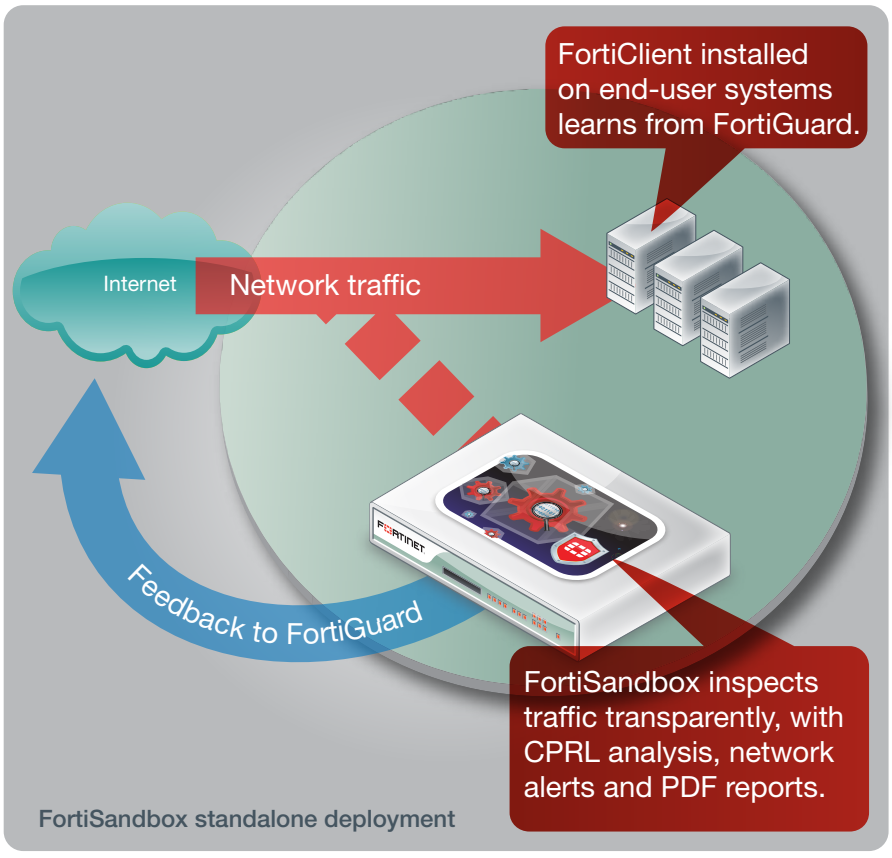
> "FortiGate and other Fortinet products are designed to mitigate advanced threats as the first line of defence at the perimeter and core, with proactive technology to detect attacks in real time and thwart them before damage is done."

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

32-bit code can run in both 32-bit and 64-bit environments, so malware authors prefer 32-bit to maximize infections. Most 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 environments, so most malicious behaviour can be observed in XP (which doesn't support 64-bit code) without further testing in 7/8. But Fortinet's antivirus engine running on FortiSandbox




FortiSandbox deployed with a FortiGate




FortiSandbox standalone deployment

with CPRL fully supports 32-bit and 64-bit code and multiple platforms: Windows, Mac, Linux, Android, Windows Mobile, iOS, Blackberry and legacy Symbian.

### 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.

### Windows XP end of life

Windows XP is fertile ground for infection, and the best operating system for reliably replicating threats in sandbox inspection. But after April 8, 2014, XP will no longer be supported by Microsoft. This means no more security patch updates, so more and more security holes will open up in XP environments. XP environments will be even

more fertile for infection. The good news is that even more malware will trigger properly in XP in the sandbox. The bad news is that it will be juicy target for attackers. You can bet that malware will be developed to go after the low-hanging fruit of end-users who don't switch to 7/8. Judging by past trends, the migration will not happen overnight.

## Threat Landscape Summary

Most of the threats observed by FortiGuard Labs are 32-bit and written to execute in Windows XP environments. Windows XP is still an active market, and an easy target. As long as developers can make 32-bit malware that will function on XP today and cross-function on Windows 7/8 when users migrate, there's no motive to craft custom Windows 7/8 malware. While FortiGuard Labs doesn't predict an immediate onslaught of 64-bit threats, Fortinet is already positioned to capture both with the combination of its CPRL, antivirus engine and FortiSandbox.

## FortiSandbox Operating System Support

To efficiently and effectively catch threats, FortiSandbox allocates resources to Windows XP and Windows 7/8 virtual environments based on the current threat landscape. As the landscape changes, supported environments will as well. Enhancements to detection of new evasion technologies and targeted platforms are integrated with FortiGate and FortiSandbox as they emerge. FortiSandbox also supports O/S-independent detection with code emulation and antivirus engine pre-filtering.

## Proactive Mitigation and Endpoint Protection

Sandbox technology can find threats, but stopping them is more efficiently performed by hardened network threat management or firewall appliances. If malicious activity is discovered, Unified Threat Management (UTM) appliances and Next Generation Firewall (NGFW) Advanced Threat Protection (ATP) can block it.

FortiGate and other Fortinet products are designed to mitigate advanced threats as the first line of defence at the perimeter and core, with proactive technology to detect attacks in real time and thwart them before damage is done. FortiSandbox is an extension of Fortinet's leading UTM/NGFW solution, feeding intelligence back to FortiGate and/or FortiGuard. New instances of attack from threats uncovered by FortiSandbox can be blocked at the first line of defence as the lifecycle continues.

## Conclusion

The reality is that all forms of security technology are known to malware creators, some of whom will craft disguises and use advanced evasion techniques. Detection comes down to inspecting as many layers as possible through all potential angles of attack. The best approach is a combination of gateway-based and sandbox inspection. Sandboxing provides a useful additional layer of defence in today's threat landscape. Used properly, it's a learning device, ultimately tied into gateway security so it can quickly identify new threat activity on the network and facilitate incident response. Integration capability between such appliances is key. FortiGuard Labs frequently discovers and monitors emerging evasion techniques so that quick counter-updates and intelligence may be sent to Fortinet solutions. Fortinet currently supports FortiSandbox integration with FortiGate, FortiManager and FortiMail security appliances.

**GLOBAL HEADQUARTERS**
Fortinet Inc.
899 Kifer Road
Sunnyvale, CA 94086
United States
Tel: +1.408.235.7700
Fax: +1.408.235.7737
www.fortinet.com/sales

**EMEA SALES OFFICE**
120 rue Albert Caquot
06560, Sophia Antipolis,
France
Tel: +33.4.8987.0510
Fax: +33.4.8987.0501

**APAC SALES OFFICE**
300 Beach Road 20-01
The Concourse
Singapore 199555
Tel: +65.6513.3730
Fax: +65.6223.6784

**LATIN AMERICA SALES OFFICE**
Prol. Paseo de la Reforma 115 Int. 702
Col. Lomas de Santa Fe,
C.P. 01219
Del. Alvaro Obregón
México D.F.
Tel: 011-52-(55) 5524-8480