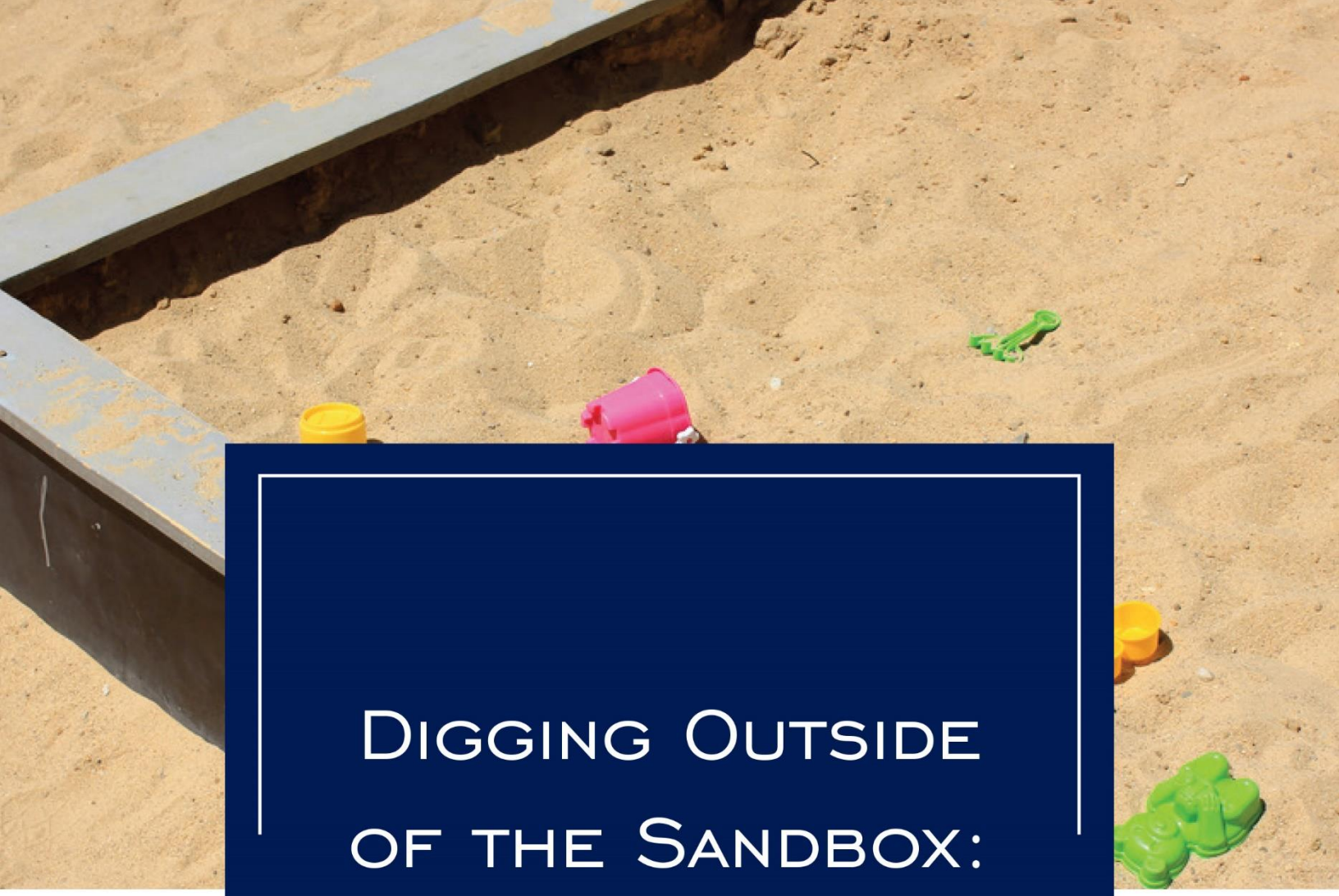


Digging Outside of the Sandbox:

Why Browser-Borne Malware's Too Tough for the Sandbox

ERICOM
Be Connected, Be Secure

Table of Contents

ERICOM

Introduction: The sandbox story

When you think about common anti-malware security techniques, 'sandboxing' undoubtedly comes to mind. This technique was first applied in the early 1990s and has proven so highly effective that it is still used widely today, over twenty years later. In recent years, sandboxes with built-in analytics have been leveraged to add an extra layer of security to anti-virus, firewall and other defensive solutions, providing a safe place to "detonate" files that might carry zero-day exploits, ransomware and other threats without risking endpoints or the networks to which they're connected.

For over two decades, sandboxes have been a safe place to detonate files that might be malicious

Today, with browser-borne threats becoming an increasingly grave concern for organizational endpoints and networks, does the sandboxing approach still provide adequate protection against browser-executable malicious code? Let's dig into the issue and see where it leads us.

Are users gritty enough to let sandboxing work?

What exactly is a 'sandbox'? Just like sandboxes found in a playground, digital sandboxes are enclosed areas designed for experimentation and "make-believe" – in this case, not for play but rather as walled-off spaces on a user device or server where incoming files are executed to see if they're safe for general usage. Within the sandbox, files cannot access external resources (although some implementations may permit configurations that allow it). While software may run indefinitely within sandboxes, in the cybersecurity context, sandboxes generally serve as an interim waystation to full access, where files and programs are quarantined for testing until proven safe.

Premature or faulty user decisions can set malware free

Sandboxes are a valuable security technology, with a few key limitations. First, because *all* file activities must be executed within the sandbox to ensure they are safe, sandboxing is s-l-o-w. In addition, basic sandboxing depends on user judgment to determine when it's safe to crack open the box and allow access to external resources, or to fully release files from the sandbox. A premature or faulty decision might expose the sandbox as a Pandora's Box, resulting in malware infecting the endpoint and from there, entire organizational networks.

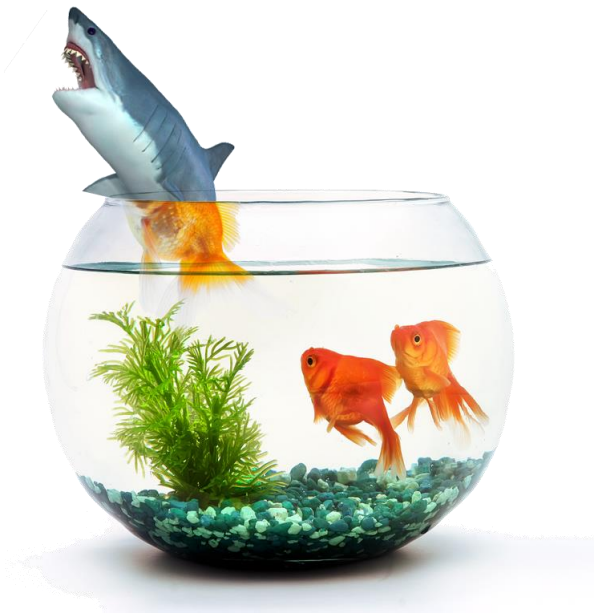
The sandbox gets smarter - but is it enough?

In the past decade, sandboxes have emerged from the playground and grown more sophisticated. Some of the most prominent players in cyberdefense have beefed up legacy sandboxes with behavioral analytics, heuristics and artificial intelligence (AI) to both speed up analysis and determine when it's safe to lift the lid from the box. These solutions can be delivered as local sandboxes on endpoints or networks, or based in the cloud. Leading firewall vendors, such as Check Point and Sophos, are integrating these smart sandboxes with their firewall solutions as well.

Malicious hackers have developed innovative malware that circumvents even the smartest sandboxing solutions

While intelligent sandbox technologies go a long way toward eliminating human-factor errors and expediting the process of examining file behavior, they also introduce new weaknesses. Some of these are designed in, such as processes that speed up sandboxing by releasing code in batches, and depend on technology to put the genii back in the bottle if malware is discovered in a subsequent batch. Others are vulnerabilities that are quickly exploited by malicious hackers, who have developed innovative evasion techniques to circumvent even the most rigorous sandboxing solutions.

Extended 'sleep' is an example of how hackers have exploited one such vulnerability, in the ongoing cat-and-mouse game of malware detection and evasion. Sophisticated malware files slumber as long as they're inside the sandbox, behaving harmlessly so as not to raise any alarms. Malicious "tells" remain hidden for the duration of the test period, until the files are deemed safe by the sandboxing system. Once the benign-seeming files are released from the sandbox to the end-user computer, however, they spring to life, attacking vulnerable networks and systems, and potentially resulting in security breaches with heavy costs.



Recent reports indicate that ransomware including Karmen, Locky and Cerber all include successful anti-sandbox features, tricks or tools. Ironically, what was intended as an anti-sandbox "kill switch" in WannaCry was poorly designed and instead is believed to have stopped the spread of one of the most vicious ransomware attacks in memory. Next time, organizations may not be so lucky.

Control vs. access: A delicate balance

Sandboxes most definitely have their place in the cybersecurity hall of fame, alongside anti-virus solutions, firewalls, secure web gateways and other solutions, as an important part of a layered defense strategy. And without doubt, leading security vendors will continue upping their sandbox technology to stay ahead of malicious and innovative hackers.

Browser-borne code executes without being downloaded, before it even gets to the sandbox

However, when it comes to protecting networks from attacks originating via browsers, currently the most prevalent threat vector, today's widely used defensive security products -- firewalls, secure web gateways, sandboxes, URL filters and others – are essential, but *still* not enough. Here's why:

Today's browsers are sophisticated environments where millions of tiny programs execute *without being downloaded*, and over which neither users nor system administrators have much control. In essence, browser-executable code was *created* to usher outside activity from the Internet onto endpoint computers. This is exactly what makes browsers so powerful and essential, but also so exceedingly dangerous.

Browser-executable code runs constantly, at virtually every moment that a user browses a website, in real-time. Because it is never actually downloaded, these tiny apps simply can't be sealed in a sandbox to cool their heels while being observed. However, they *can* introduce malware and fileless exploit kits that quickly spread from endpoint to server, and throughout the whole organizational network. The deal is done before the malware touches the sandbox. And no amount of behavioral analysis, heuristics or AI can stop it.

Browsing is such an integral part of business activity that walling off Internet use on virtual machines or in no-access sandboxes is out of the question. Users access out-of-the-sandbox endpoint functions such as printing, downloading or email applications via their browser constantly and without thought. And when they do, the cat is out of the bag, the deal is done, and no number of clichés can stop the malware, ransomware and other threats from spreading through the endpoint and onto the system.

If sandboxes are just part of the cybersecurity answer and, like anti-virus and firewalls, provide only partial protection from browser-borne threats, how can businesses secure endpoints and networks from browser-executable malware and the browsers that run it blindly?

Lock applications in, toss out the key

The answer lies in software containers that are purpose-built to isolate browsing.

Containers are in many ways similar to sandboxes, with a crucial conceptual twist. While sandboxes generally serve as a quarantined waystation for files en route to the endpoint or network, containers are the end of the road for applications.

> In the world of software containers, applications can be isolated with no chance of reprieve

In the world of containers, all applications are assumed to be deadly. Not guilty until proven innocent – just guilty, period, with no chance of reprieve. There's no testing, no checking, no "what if" or "so far." What happens in the container stays in the container, so it can *never* spread to the endpoint operating system or from there to the network. Each container can be designed as an isolated environment tailor-made for just a single application, with only the elements required for that purpose. As a result, it is inherently more streamlined and easily managed than physical or even virtual machines and offers a smaller attack surface, making it an optimal technology for cybersecurity applications.

Browsing, contained

As executable code, not just files, browser-borne content cannot be successfully sandboxed. But it can be contained, along with the browser that runs it. Remote browsing entails creating virtual browsers within containers, where browser-executable code runs without risk. These processes, both good and malicious, stay permanently locked in the container, fully isolated, until it's destroyed. By eliminating the container completely, we also eliminate opportunities for malware to persist.

> The trick is accessing application outcomes, while the app remains sealed off

While irrevocably double-sealing the browser, along with attendant executable code, is a great security move, we're left with the conundrum of how users can get full, integrated and seamless access to the websites they need. If a browser executes within a container and remains sealed within, how can users interact with and benefit from the Internet from their regular workplace devices? How can they flip between work applications and internet sites, clicking, copying, downloading and printing as they proceed?

The solution within

The answer lies in what resides in the container, alongside the virtual browser. Remote Browser Isolation (RBI), one of the hottest and most effective new cybersecurity solutions, leverages container-based virtual browsers to render websites as safe content and stream it to endpoint browsers in real time. RBI provides a seamless, interactive browsing experience that is transparent to users – while keeping all browser-executable code locked safely within the container.

Remote Browser Isolation provides a natural browsing experience while keeping risky browser-executable code locked safely away

Like sandboxes, containers may be created on endpoints or organizational networks. The most secure solutions, such as Ericom Shield, locate the containers off organizational networks and away from all endpoints, in the cloud or network DMZ, so that no active code ever reaches the endpoint or organizational network.

To prevent possible cross-site contamination, each browser tab executes in a new virtual browser, in its very own container. And in a further safeguard, each browser or tab is destroyed, along with its container and any code that persists, shortly after going inactive, although sessions are quickly restored if the user returns. Containers are never reused: Each browsing session is opened in a fresh new container. Together, these measures ensure that chances of accidental leakage and contamination drop to virtually nil.

The bottom line

If sandboxing is the 'original' isolation technique, then virtual containers are definitely the 'new, tougher and much improved' version. Taking the same core principles that made sandboxing so effective for so long – quarantining files that may contain malware threats – virtual containers have gone many steps further by applying the method to applications, not just files, and leveraging full browser isolation and container destruction to guard organizational systems against even the shrewdest of modern malware, ransomware and other threats.

Virtual containers are the new, tougher and much improved isolation technique, that works for applications as well as for files

Ericom Shield is an advanced remote browser isolation solution that adds a powerful layer to organizational defense-in-depth strategy by isolating malware, ransomware and other threats where they can't harm corporate network or user devices. It transparently secures Internet use, including file downloads, while reducing risk, costs and operational burden to IT staff responsible for browsing operations. Ericom Shield harnesses the power of isolation to deliver secure browsing and protect the corporate network and endpoints.

Contact us now for more information about how Ericom Shield Remote Browser Isolation can protect your organization from browser-borne ransomware, malware and other threats

ERICOM
BE CONNECTED, BE SECURE

www.ericom.com/solutions/browser-isolation | shield@ericom.com
US: (201)767-2210 Europe: +44 (0)1905 777970 ROW: +972-2-591-1700