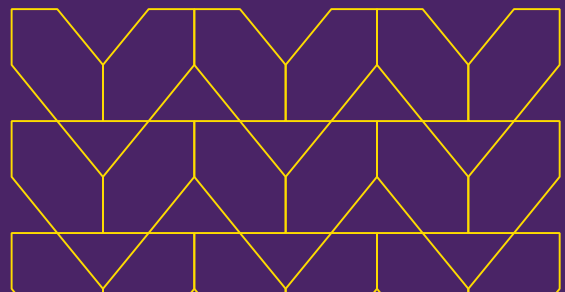
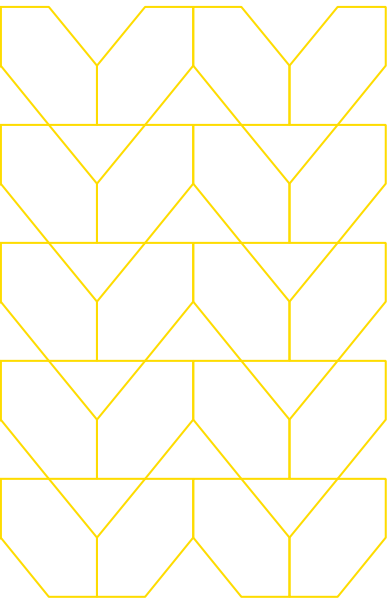


Adaptive Clientless Rendering™

The innovation powering the
Menlo Security Isolation Core™



Web browsers are among the most important applications in our business lives, yet they're also the most vulnerable to attack.



The simple act of loading a malicious web page suffices to compromise the user's endpoint, leading to malware installation, data theft, and penetration of corporate networks. Downloading malicious content is easier than most people think. Email has become the attack vehicle of choice, because threat actors can now spin up a seemingly legitimate email from a trusted individual or brand that plays on our trusting human nature. Often, users are all too willing to click on a link that leads to an infected site or give away their credentials to bogus web forms that look like the real thing. At the same time, an ever-increasing set of browser features ensure that attackers will continue to have an unlimited supply of vulnerabilities to exploit and links will continue to look more real.

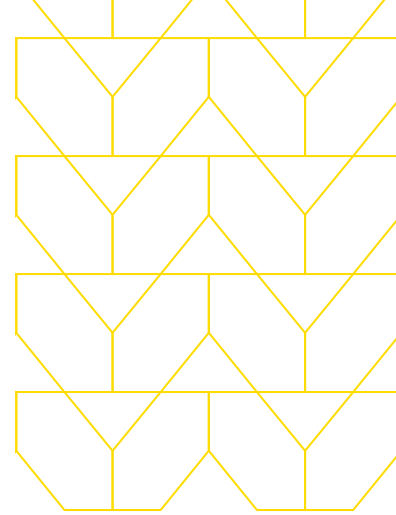
A critical ingredient in today's browser exploits is active content. In the modern web, active content largely comes in the form of JavaScript. Active content executes in the context of the user's browser and enables significant attacker control and visibility into the browser's workings and vulnerabilities. For instance, active content enables the attacker to discern memory locations (address space disclosure), influence data layout (heap spray), and dictate code generation (JIT spray)—all of which are key techniques in crafting a successful exploit.

Modern endpoints have built-in defenses against simple browser exploits, but active content execution enables determined adversaries to bypass these defenses with sophisticated, multi-stage attacks. In particular, two pervasive defenses—Data Execution Prevention (DEP/NX) and Address Space Layout Randomization (ASLR)—thwart simple code injection and Return-Oriented Programming (ROP) exploits, respectively. However, with the aid of active content, an exploit can bypass both DEP and ASLR, typically by triggering a secondary vulnerability—one that, for instance, reveals the memory location of native code. The exploit can then use that code to craft ROP code sequences that execute the attacker's bidding.

25%

Through 2022, 25 percent of organizations will adopt browser isolation

Verizon, 2018 Data Breach Investigation Report



Isolation Is the Future

Remote browser isolation is a technology that offers a solution to the security challenge posed by executing active content on the endpoint. It centers around the notion of an isolated browser—a web browser that loads and runs pages, including all embedded active content, inside a contained environment in the cloud with the goal of isolating any potential browser infection away from the endpoint. In most incarnations, the isolated browser and the endpoint are separated by a secure channel using a minimal, highly restrictive protocol designed to carry rendering updates to the endpoint and user input to the isolated browser, and nothing else. This “air-gapped” browsing mode enables browser isolation to defend against today’s sophisticated zero-day exploits. Specifically, by running untrusted active content on the isolated browser and preventing it from probing and exfiltrating data from the endpoint, browser isolation precludes exploitation of secondary vulnerabilities essential to bypassing standard endpoint defenses.

The Isolation Challenge: Making It Practical

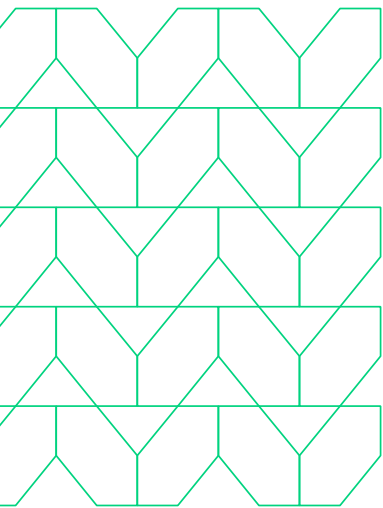
Despite its compelling security benefits, remote browser isolation must meet IT and end-user needs if it is to become a widely adopted security technology. To that end, we have identified five key requirements for a practical browser isolation solution.

The first requirement, **clientless deployment**, reduces IT burden by altogether avoiding the need for endpoint software installs, and with it, the risk of destabilizing the endpoint. The clientless nature also enables easy enterprise-wide deployment via in-network proxy configuration, as well as fully centralized management of browsing policies and security updates across all devices—including personal devices—within the enterprise network.

Equally important, a **native user experience**, in which users do not perceive a difference from a native experience, is critical for ensuring end-user productivity and buy-in. In particular, users should not have to alter the way they browse the web or be distracted by changes to their browser's behavior. Moreover, rendering speed and quality should be identical to native for a broad range of media types (text and video), and day-to-day operations such as printing and copy-paste should work as usual. Geography shouldn't factor into the user experience either. Users need to be able to browse the web and access Software-as-a-Service (SaaS) platforms no matter where business takes them—whether it's across oceans or down the street in the local coffee shop.

Scalability is also important. As IT organizations undergo cloud transformation, the number of web requests will only increase. Scaling to meet this demand needs to be effortless without the need for lengthy capacity planning.

Remote browser isolation should also be **integrated** with the rest of the security stack to ensure that all security policies are being met. This includes extending the broader architectural role of an isolation core to additional services, including cloud access security broker (CASB) and data loss prevention (DLP).



And finally, remote browser isolation should reduce operational costs through better malware containment, fewer false positives, lower help-desk costs, and a decrease in the need to reimage damaged machines—because every expenditure in today's hyper-competitive business environment needs to pay for itself.

Meeting these requirements means solving the challenging problem of transparently remoting the isolated browser's rendered output to the existing endpoint browser without requiring additional endpoint modifications (no agents or special plug-ins).



Unfortunately, the traditional and most prevalent remoting technique—pixel mirroring, also known as virtual desktop infrastructure (VDI)—based video streaming—falls short, mainly because it treats the isolated web page as a bag of pixels to be mirrored with a client that has little understanding of what those pixels represent. The result is a one-size-fits-all approach that not only precludes adapting the remoting technique to the kind of content being displayed (text vs. video), but also slows down page load time and responsiveness by eliminating opportunities to harness the browser’s hardware-accelerated rendering features, and hinders everyday workflow operations such as printing and copy-paste.

Recognizing the challenges posed by pixel-mirroring technology, several browser isolation solutions have traded off the clientless form factor for specialized endpoint browsers, plug-ins, and virtualization. While these trade-offs are acceptable in some environments, the resulting IT burden caused by trouble tickets and endpoint disruption has proved to be a significant barrier to broader deployment.

Adaptive Clientless Rendering

Menlo Security’s patented Adaptive Clientless Rendering™ (ACR) architecture is the core technology behind the Menlo Security Isolation-Powered Cloud Platform. In a clear departure from traditional VDI-based video streaming technology, ACR combines a web-based delivery vehicle with a greater understanding of the isolated page to simultaneously enable clientless deployment and a native user experience.

Depicted in Figure 1, the ACR architecture involves two major components: the Safe Page and the isolated browser.

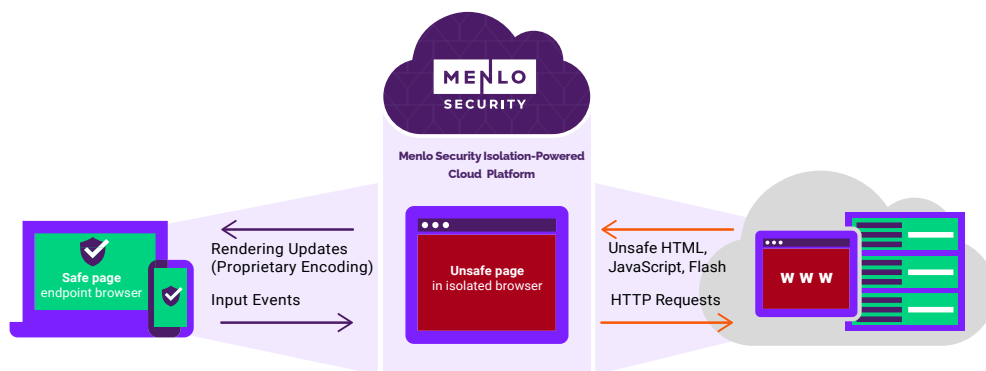


Figure 1: The ACR clientless architecture. The existing endpoint browser loads a safe, transcoded version of the original page (Safe Page) that interprets rendering updates coming from the isolated browser and relays input events back to it—all over a secure HTTPS channel.



The Safe Page is a safe, transcoded version of the target web page that is loaded by the existing endpoint browser in lieu of the original page. Served via a secure web proxy, the Safe Page establishes an SSL-encrypted communication channel to a freshly allocated isolated browser upon loading, and then applies rendering updates coming from the isolated browser and relays user inputs back to it. The Safe Page capitalizes on rapidly converging web standards and advances in browser engines to work accurately and efficiently regardless of which major browser or device is used on the endpoint.

Running on the Menlo Security Cloud Platform, the isolated browser loads web pages on the endpoint's behalf. It sends rendering updates to the Safe Page in response to dynamic page changes and injects user inputs coming from the Safe Page. Based on an up-to-date version of the Chromium browser engine, the isolated browser inherits its security, stability, and feature set. However, since no browser engine is immune to infection, the Menlo Security Cloud Platform operates under the assumption that its isolated browsers will eventually be infected as well. Thus, as a key step in securing the isolation platform as well as end users, the Menlo Security Cloud Platform employs frequent disposal of isolated browsers along with multi-level container isolation to avoid both persistent and lateral infection.

Multiple Modes to Deliver a Seamless User Experience

The key to the ACR transparent user experience is its ability to reconstruct an equivalent version of the isolated browser's Document Object Model (DOM) on the endpoint browser. The DOM is the dynamic, browser-internal representation of the user-visible rendering. Reconstructing a DOM enables most of the rendering work—including interactive animations such as scrolling—to be performed solely on the endpoint browser using its built-in GPU-optimized machinery. Moreover, a reconstructed DOM exposes semantics of the content being rendered so that the full browser feature set—including copy-paste, find in page, printing, and password managers—continues to function.

ACR employs two distinct reconstruction modes that are dynamically selected at page granularity based on a variety of factors, including the endpoint device used and the content of the page.

Mode 1: DOM Mirroring

Intuitively, the goal of DOM Mirroring is to mirror only the benign DOM information to the endpoint browser. To reflect DOM changes on the client, each isolated browser tab actively monitors the currently loaded page's DOM tree. These updates are applied to the local DOM using the standard DOM API available in all browsers.

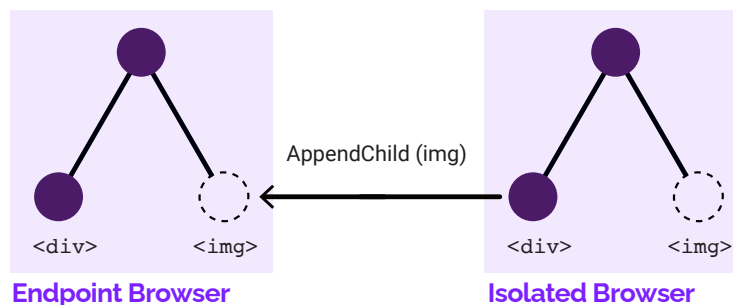



Figure 2: DOM Mirroring allows the mirroring of the DOM tree and the resources to the endpoint browser.

Adaptive Transcoding: DOM Mirroring confers distinct advantages over VDI-based video streaming approaches, primarily owing to the selective exposure of DOM elements to the client. A key benefit is that it enables the selection of remoting strategy at DOM element granularity, so nonactive safe elements are left as is and active unsafe elements are either dropped altogether or are replaced with a safe, transcoded variant that is best suited for the element's media type.

Rendering and Workflow Offloading: Central to providing a truly transparent user experience, DOM Mirroring leverages the capabilities of the endpoint browser—resulting in visible benefits such as fast page loads, smooth scrolling and animations, and crisp, high-quality HTML5 video playback. The semantically aware rendering of DOM Mirroring also enables the client browser to render a truly native look and feel, regardless of endpoint browser or platform.

Finally, DOM Mirroring avoids disruption to workflow operations such as copy-paste, find-replace, and printing. Copy-paste, for example, is difficult to emulate in a truly native fashion using VDI-based video streaming because of browser-enforced security restrictions on asynchronous clipboard manipulation. Printing, too, is encumbered by the endpoint browser's view of the page as a block of pixels, as opposed to a document that can be reflowed to accommodate any output device. In contrast, DOM Mirroring provides the endpoint browser's existing workflow mechanisms with all the information it needs to do its job, so emulation is not needed.



Applicability to Document Isolation: Like web browsers, document applications such as Microsoft Office and PDF viewers are also susceptible to malicious content downloaded off the web or sent via email attachments. Here, too, active content embedded within a malicious document plays an enabling role in exploiting the host application's vulnerabilities. It is no surprise that the core techniques behind ACR apply equally well to the problem of isolating documents. In particular, upon downloading a malicious document through the browser, the Menlo Security Cloud Platform uses ACR to transcode the document into a layout-preserving HTML5 page and then loads the transcoded content into the isolated browser. DOM Mirroring then ensures safe, clientless, and transparent mirroring of the document to the endpoint.

Mode 2: Smart DOM

As business moves to the cloud, and as user access to SaaS platforms and web apps through mobile platforms becomes mission critical, the rendering engine in an Internet isolation cloud needs to adapt to these trends. New, powerful browsing features in mobile browsers make a new generation of isolation engine increasingly necessary. The proliferation of mobile devices introduces the need to be both power and network efficient when remoting content to the endpoint. Menlo Security's second-generation option—called Smart DOM—takes these new technical requirements into account.

Rather than mirror the isolated browser's DOM tree, Smart DOM generates a different yet equivalent DOM tree on the endpoint browser using low-level rendering data structures provided by the isolated browser's compositor subsystem. The unique approach to DOM reconstruction used by Smart DOM confers several benefits:

Smart DOM ACR renders pages accurately across a wide range of browsers, thus offering a more consistent user experience across browsers old and new, as well as across desktop and mobile devices. Smart DOM avoids cross-browser incompatibilities using patent-pending techniques.

Smart DOM is bandwidth efficient. Unlike VDI-based video streaming, Smart DOM avoids duplicate network transfers.

As with DOM, **Smart DOM preserves the native user experience.** Smart DOM renders quickly on the endpoint, naturally harnessing the browser's GPU acceleration to enable 60FPS scrolling, pinch-zoom, and animations across desktop and mobile devices alike. Moreover, Smart DOM facilitates the accurate emulation of native-supporting functionality such as copy-paste, find-replace, printing, mobile text input, endpoint local fonts, and native page widgets.



Smart DOM renders on the endpoint. Smart DOM mirrors no active content, thus exposing minimal attack surface.

This new way of safely mirroring content on users' browsers ensures that Menlo's ACR technology is adaptable in the truest sense. New modes can be added as browsers and web development tools evolve. **Menlo intelligently optimizes which mode to use and sets Smart DOM objects as the default for mobile.** Together, the two rendering modes are delivered by a cloud service that can continuously evolve without impacting service uptime.

This allows the Menlo Security Cloud Platform to seamlessly and safely render web content to users' devices while increasing the accuracy of the native browser experience, scaling to a wider array of different media, and decreasing the amount of bandwidth that the Internet isolation cloud requires. Most importantly, the user's browsing experience is not changed at all. Users can continue to browse the web, access web apps, and work online as they have always done, with no impact on performance.

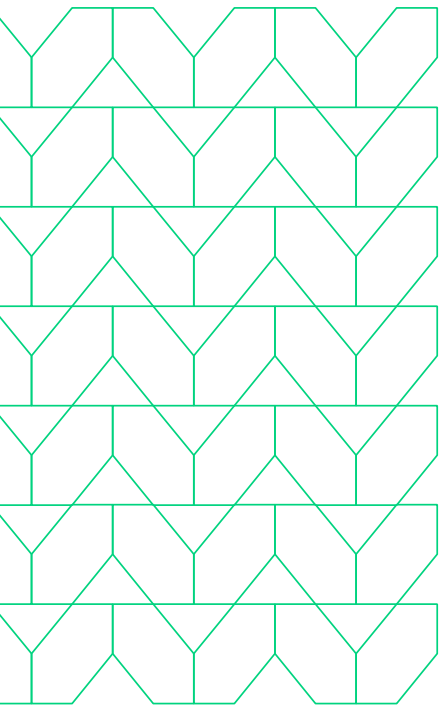
Security

The guiding principle behind the ACR security model is that active content execution is the key to successful evasion of endpoint defenses; without it, an exploit has little hope of bypassing existing defenses, regardless of what else an infected isolated browser sends down to the client. With this principle in mind, ACR employs two security mechanisms that, in conjunction, offer a strong defense against even the most determined adversaries.

The first mechanism, **active content blocking and transcoding**, ensures that active content never gets sent to the endpoint. In DOM Mirroring mode, this involves filtering all incoming DOM elements, attributes, and CSS against a whitelist. For instance, **<script>** elements and onclick attributes are dropped, while **<object>** elements are replaced with a safe remoting widget that displays the transcoded, real-time output of the plug-in window as it is rendered on the isolated browser.

Smart DOM requires no special filtering mechanism as it sends only compositor-level structures that are guaranteed by design not to contain DOM or CSS state. As an added layer of protection, Menlo Security also employs a Content Security Policy at its strictest setting (no inline script, no plug-in) to ensure that the endpoint browser blocks all active content executions.

A second security mechanism, **protocol checking and enforcement**, ensures that the isolation platform is not fooled into executing active content by malformed updates coming from an infected isolated browser, and that it does not inadvertently leak exploit-aiding information to an infected isolated browser. In particular, all rendering updates coming from the isolated browser are expected to be in a canonical format. For example, in DOM Mirroring mode, DIV element updates must have the “div” tag; the endpoint does not accept any other string variant, even those including strange character codes that may be interpreted unexpectedly by the endpoint browser. Second, the endpoint verifies that outgoing messages adhere to a simple user-input protocol; e.g., “click button 1,” “keypress code 45,” or “scroll to 45.” This leaves an infected isolated browser without a channel to probe the endpoint for vulnerabilities or to exfiltrate information useful for bypassing standard endpoint defenses.



From:
Browser isolation promised to safeguard users from future zero-day threats by running active content away from the endpoint.

To:
Isolation provides the most secure Zero Trust approach to preventing malicious attacks—by preventing malware from reaching end users while they work online, and by removing the operational burden for security teams.

Applicability to Document Isolation

Like web browsers, document applications such as Microsoft Office and PDF viewers are also susceptible to malicious content downloaded off the web or sent via email attachments. Here, too, active content embedded within a malicious document plays an enabling role in exploiting the host application’s vulnerabilities. It is no surprise that the core techniques behind ACR apply equally well to the problem of isolating documents.



In particular, upon downloading a malicious document through the browser, the Menlo Security Cloud Platform uses ACR to strip out the active content and transcode the document into an HTML5 page that preserves the original layout. Menlo then loads the transcoded content into the isolated browser, providing mirroring of the document to the endpoint.

Conclusion

The browser feature set continues to expand with new JavaScript-accessible APIs. Going forward, we can anticipate novel exploits against this newly exposed attack surface, with active content firmly remaining the predominant vector for exploitation. Browser isolation promises to shield users from these future threats by running active content away from the endpoint. However, to reach its true potential, browser isolation faces the challenge of providing both clientless deployment and a fully transparent user experience. Adaptive Clientless Rendering™—the novel remoting technology at the core of the Menlo Security Isolation-Powered Cloud Platform—meets this challenge by adaptively and safely reconstructing a full-fledged DOM on the endpoint browser, thus enabling native rendering and the full scope of native browser functionality.

With mechanisms to ensure that active content never executes on the endpoint browser, ACR defends against zero-day threats while providing a clientless and native browsing experience.

To learn more about securing the ways people work, visit menlosecurity.com or email us at ask@menlosecurity.com.



To find out more, contact us:

menlosecurity.com

(650) 695-0695

ask@menlosecurity.com



About Menlo Security

Menlo Security enables organizations to eliminate threats and fully protect productivity with a one-of-a-kind, isolation-powered cloud security platform. It's the only solution to deliver on the promise of cloud security—by providing the most secure Zero Trust approach to preventing malicious attacks; by making security invisible to end users while they work online; and by removing the operational burden for security teams. Now organizations can offer a safe online experience, empowering users to work without worry while they keep the business moving forward.

© 2021 Menlo Security, All Rights Reserved.