Island

Delivering Browser Isolation Outcomes Natively with the Island Enterprise Browser

White paper | June 2022

Overview

By their very nature, web browsers are built to run 3rd party code directly on the endpoint. The majority of these engagements come without verification, creating fertile ground for attackers wishing to exploit unsuspecting end-users. While most enterprises perform continuous efforts focused on educating users to minimize risky behaviors, not even the best education can prevent all of today's sophisticated threats. Phishing, malware, ransomware, and many other threats often begin with a web-based engagement. The consumer browser is an unwilling participant in these engagements, so enterprises need to layer control after control around these web browsers to insulate them from danger. The traditional answer? Stand up a stack of controls in front of these consumer browsers to protect the browsing experience.

> These attacks and defenses leave the user's consumer browser (which cannot defend itself from such techniques) subject to exploitation

Protecting web usage typically began with a standard web gateway (proxy) infrastructure for many organizations. While these were practical approaches in years past, the growth of encrypted traffic (SSL) and sophisticated threats such as browser code injection leave existing proxies and SASE solutions unable to protect end-users adequately. These attacks and defenses leave the user's consumer browser (which cannot defend itself from such techniques) subject to exploitation.

In an attempt to combat these attacks, many organizations explored using Remote Browser Isolation (RBI) technologies to augment existing proxy resources. The concept behind RBI is to force uncategorized (or other conditional) web traffic into a virtualized cloud environment for remote content execution. As the user engages web content in this way, the site is rendered over a video stream (often HTML5) back to the user's consumer browser. In principle, the user is protected from any harmful content.

Technological Challenges of Remote Browsers

On the surface, a remote vehicle to execute potentially dangerous web content for the user seems like a viable protection strategy. However, this approach is fraught with its own set of challenges. To begin with, it is not palatable to force all users' traffic through RBI. Why? Because the user experience and performance simply are not acceptable for everyday use. Rendering the content remotely and streaming it back to the user adds noticeable lag and visual imperfections. Thus, because the experience is generally poor, RBI technologies are usually invoked only in specific situations. For example, RBI is often used where content must be isolated for potentially malicious web content on untrusted sites. This means that only a fractional subset of traffic (usually 1-2%) is passed through RBI technologies in the first place. Thus by reducing the footprint of where RBI is engaged, the organization can attempt to remove the concern over end-user friction.

> Rendering the content remotely and streaming it back to the user adds noticeable lag and visual imperfections. Thus, because the experience is generally poor, RBI technologies are usually invoked only in specific situations.

Yet this leaves a significant gap for sites categorized as collaboration, file sharing, social media, and others. In these cases, the web traffic is never passed through an RBI solution, yet risky content still exists. Further, Single Page Applications (SPA) and HTML5 canvas rendering are meant to be executed locally and would not be candidates for passing through RBI solutions. Put simply; the attack surface is much larger than the exploitation footprint protected by RBI. These limitations call into question the value of the investment.

RBI technologies are also quite limited where other types of common browser-centric attack techniques might be employed, such as:

- **Modern Phishing Attacks**
- **Exfiltration Of Data**
- **Man-in-the-Middle Attacks**
- **Malicious Extension Exploitation**
- **Embedded Malicious Document Content**
- **Localized Browser Tampering**
- **Man-in-the-Browser Attacks**

In each of these cases above, RBI either has no role in protecting against the attack or cannot be instrumented to protect because it only sees uncategorized or untrusted traffic.

Use Case Limitations

As previously mentioned, RBI is most often invoked for web traffic destined for suspicious sites that might cause remote browser code injection or attempts to phish users leveraging fake sites. However, this limited usage of RBI means that it cannot fulfill more valuable browser-based use cases that may be important to the organization, such as:

- **SaaS and Internal Web Application Protection**
- **Contractor and Third-Party Provisioning/Protection**
- **Call-Center Worker Governance**
- **Bring-Your-Own-Device Policies**
- **Privileged User Protection**

A web browsing experience is often central to the needs of such use-cases. Yet it is essential to note that RBI can play no effective role in these scenarios. To begin with, the necessary traffic for these needs usually isn't routed to RBI. Further, RBI just isn't built to solve these challenges and lacks the mechanics required to add value to these core browsing use-cases.

Rethinking Browser Isolation Outcomes

The proliferation of threats leveraging the web has piqued interest in RBI technologies. However, RBI provides limited solutions solving only the symptoms (user phishing, remote code injection). This pattern is all too frequent in cybersecurity, where solutions are built to address a handful of symptoms without addressing the core problem. In the cases mentioned above, the core problem is that consumer browsers were never developed to accommodate the needs of the Enterprise.

> As the inventor of Remote Browser Isolation, Island co-founder and CTO Dan Amiga has almost ten years of experience innovating browser technologies and a deep understanding of its pitfalls.

Yet what if the browser was built for the enterprise? This is precisely what Island considered as we created the industry's first Enterprise Browser. As users engage with all corners of the web, Island has innovated built-in browser protection to ensure that all engagements are safe. These capabilities deliver far more effective outcomes than clunky RBI solutions while doing so in a native browsing experience. This ensures that the user has exceptional protection without the negative impacts on their experience. Let's dive in more.

As the inventor of Remote Browser Isolation, Island co-founder and CTO Dan Amiga has almost ten years of experience innovating browser technologies and a deep understanding of its pitfalls. Thus, at the core, Island put significant expertise and effort into delivering built-in Browser Isolation into the Enterprise Browser without the need for the "remote" part.

At its foundation, Island was built on the Chromium project. This project is the basis of modern browsers like Google Chrome, Microsoft Edge, and many others. Using Chromium ensures a web browsing experience that end-users are familiar with and the snappy performance they expect. Yet, Chromium's Just-In-Time (JIT) compiler has been at the core of many recent zero-day vulnerabilities across all browsers which leverage Chromium. Rather than attempting unnatural techniques such as Remote Browser Isolation, which ruins the user experience, Island took a different approach by going straight to the source of the problem. While the JIT was originally designed to improve user performance, the tiny performance improvements it delivers come with a tradeoff of significant vulnerability. Thus, Island pioneered disabling the JIT as a default configuration resulting in no noticeable performance impact while eliminating the most vulnerable part of any Chromium-based browser. Disabling the JIT also disables WebAssembly, further reducing the attack surface. Yet Island created policy-driven capabilities to selectively enable the JIT and WebAssembly in the very rare situations that require it.

> **By delivering Browser Isolation directly into the Enterprise Browser, Island removed the most significant areas of browser vulnerability and added capabilities to protect against exploits**

In addition to rethinking the use of the JIT, Island also leverages several additional protective capabilities by enabling Arbitrary Code Guard, Control Flow Enforcement, and Control Flow Guard. Each of these capabilities ensures that arbitrary code cannot be injected directly in an attempt to manipulate the memory or execution flow of the Enterprise Browser.

By delivering Browser Isolation directly into the Enterprise Browser, Island removed the most significant areas of browser vulnerability and added capabilities to protect against exploits. As previously mentioned, this solves the core problem of advanced web threats rather than the symptoms. These alone negate the need for Remote Browser Isolation solutions by preventing malicious code execution directly within the browser.

Ensuring Complete Browser Protection

While Island has embedded Browser Isolation capabilities directly into the browser, delivering a safe browsing experience must go deeper than RBI solutions can provide. Thus, Island emphasized protecting the browsing experience beyond Browser Isolation outcomes alone. Below are a few additional capabilities that Island uniquely delivers directly within the browser:

Man-in-the-Middle Protection

Island is the first browser to provide policy-driven capabilities to recognize when an untrusted man-in-the-middle technique is employed. This allows the organization to completely prevent data theft by a man-in-the-middle technique.

Document Isolation

With a built-in document viewer, Island provides a facility to allow interaction with a document without the risk of malicious embedded code being executed on the desktop. This is controllable by policy, enabling great flexibility over its use.

Malicious Extension Protection

By controlling the entire browsing experience, Island also includes oversight of extension usage. This gives the organization the power to control by audience which extensions are allowed and which are not. In addition, Island's Extension Protection can ensure that critical application or user data can be protected from extensions where required.

Localized Tamper Prevention

Island protects its own installed software footprint with anti-tampering capabilities. This ensures that any attempts to manipulate core processes or memory footprint will completely disable the browser if detected.disable the browser if detected.

Man-in-the-Browser Protection

By leveraging many of the core technological capabilities within Island's Browser Isolation, the Enterprise Browser delivers native man-in-the-browser protection. This ensures that when attempts are made to insert code impersonating a legitimate site, the rendering of the code can entirely be eliminated.

Anti-Phishing

Island built a unique facility directly within the browser to protect users' credentials against phishing attempts. In users' everyday browsing engagements, attackers may attempt to trick the end-user into keying their corporate credentials into a fake site. Island's Anti-Phishing capabilities ensure that corporate credentials can only be input into legitimate, trusted destinations.

Web Categorization

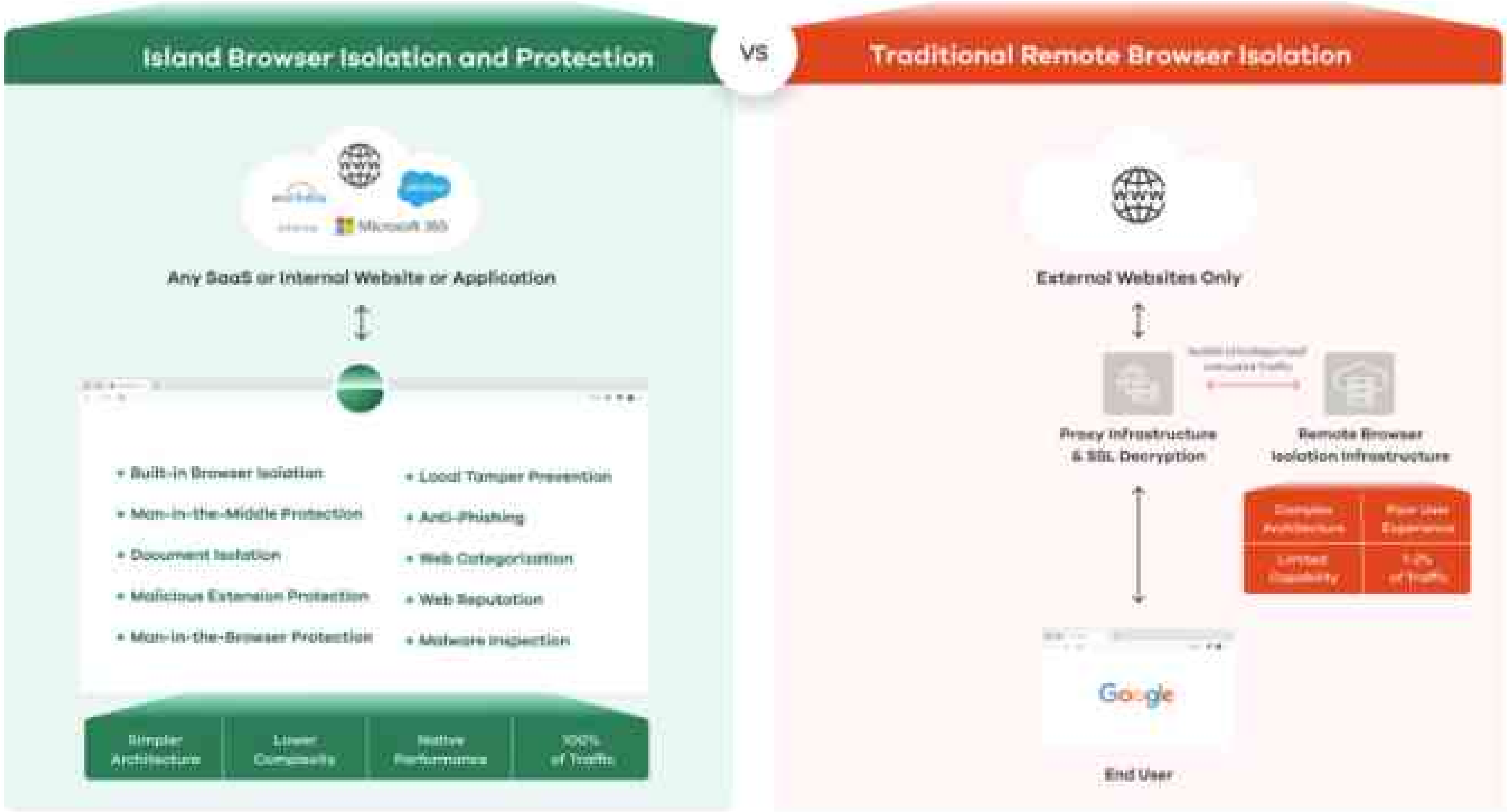
Categorizing web content for safe browsing has been a hallmark of web proxy technologies for years. Island simply embeds web categorization directly into the browser for safe browsing and compliance needs.

Web Reputation

Dynamic protection over end-user engagements is often performed within proxies and other technologies leveraging reputational capabilities. Island simplifies these capabilities by building web reputation directly into the browser for real-time user protection.

Malware Inspection

At its basic level, it is essential to inspect content that users may download or upload to ensure the safety of the files in question. Island Enterprise Browser has built-in malware inspection to ensure that all files uploaded or downloaded from any web destination can be inspected as policy requires anti-tampering capabilities. This ensures that any attempts to manipulate core processes.


Island Browser Isolation and Protection
VS
Traditional Remote Browser Isolation
Microsoft 365
Any SaaS or Internal Website or Application
Built-in Browser Isolation
Man-in-the-Middle Protection
Document Isolation
Malicious Extension Protection
Man-in-the-Browser Protection
Local Tamper Prevention
Anti-Phishing
Web Categorization
Web Reputation
Malware Inspection
Simpler Architecture
Lower Complexity
Native Performance
100% of Traffic
WWW
External Websites Only
Proxy Infrastructure & SSL Decryption
Remote Browser Isolation Infrastructure
Complex Architecture
Poor User Experience
Limited Capability
1-2% of Traffic
Google
End User

The Takeaway

Protecting users and the organization's most valuable application resources is vitally important. While Remote Browser Isolation technologies were an interesting concept many years ago, their adoption never made it to the mainstream. The implementation of RBI is complex, limited in outcomes, and disrupts the end-user experience. They are designed to address symptoms of web usage yet do not address the core problems.

Island has taken the outcomes promised by RBI technologies and brought them natively into the browser experience. Instead of solving symptoms, Island went to the core of the problem by building a browser specifically for the Enterprise. Yet Island didn't stop with Browser Isolation alone; the Enterprise Browser delivers a full spectrum of capabilities to ensure user protection from an exceptional range of threats. It boils down to a simple question: why have a subpar remote protection experience if you can deliver an easier, more complete, and natural experience locally?

	Island Enterprise Browser	Remote Browser Isolation
Performance	Native Browser Performance	Poor Performance
Impact on UX	Natural User Experience	Unpleasant User Experience
Traffic Coverage	All Traffic	1-2% of Traffic
Anti Exploitation	Proactive Built-In Exploit Prevention	Remote Execution of Content
Phishing Protection	Domain Misuse Prevention	Render Site Remotely as Read-Only for Uncategorized Traffic
Man-in-the-Middle Protection	Complete Man-in-the-Middle Protection	None
Man-in-the-Browser Protection	Complete Man-in-the-Browser Protection	None
Malware & Ransomware Protection	File scanning for upload and downloads to block malicious payloads	Limited
Extension Protection	Full Extension Control and Protection	None
Device Posture Support	Full Device Posture Assessment for Policy Driven Decisions	None
Document Isolation	Full Localized Document Isolation with Complete File Engagement	Rendering of Content in Cloud with No Engagement
Secure Storage	Built-in Secure Storage For Full File Engagement	No Secure Storage
Last Mile Controls	Full-Last Mile Control for Natural Application Protection and Interaction	No Last Mile Controls
Enterprise Modules	Complete Capability of Engaging Browser-based RPAs for Unique Application Needs	No Browser-based RPA
Industry Trend	The Future	The Past