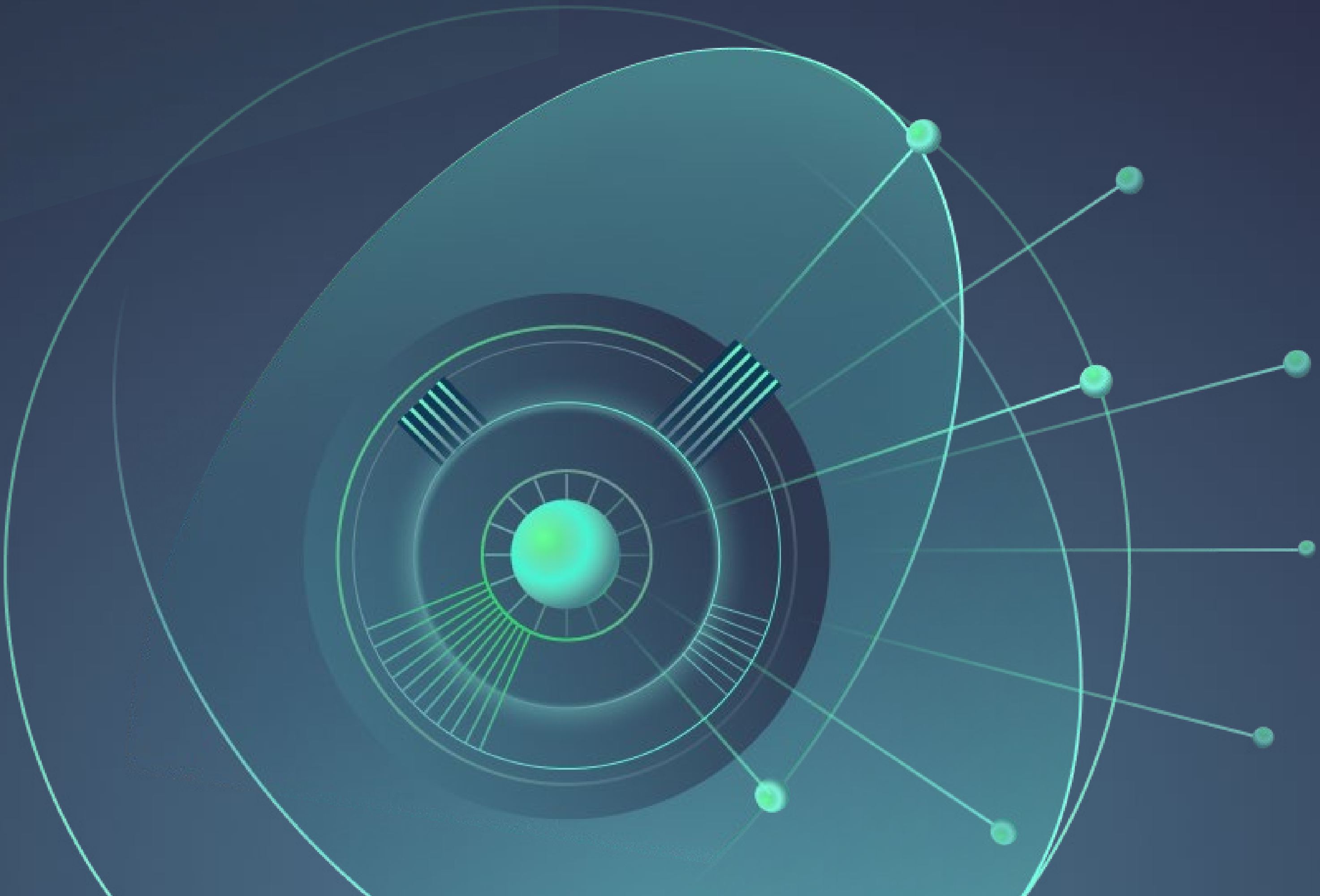
GARRISON

Browser Isolation Buyer’s Guide

What you need to know about investigating the use of Browser Isolation to secure users as they browse the web



If you’re one of the growing number of professionals reviewing the use of Browser Isolation to protect your organisation from threats like ransomware and phishing, you’re far from alone. But the technology you choose can have implications for security, usability, management overheads, and cost.

It’s easy to see why Browser Isolation (or Remote Browser Isolation) is such a critical technology. Threats on public web pages are growing and, while firewalls, proxies, user training, web filters, and other tools can help, there is always a risk that malicious code can make its way onto your endpoints.

Contents

What is Browser Isolation?

Browser Isolation solutions protect users from ransomware, phishing and other web-based threats while they browse the web or click links in emails. While it is useful for all users, it's especially used to secure vulnerable or high-risk users, such as senior management, system administrators, or anyone whose job might involve visiting untrusted sites, while having privileged access to your systems and assets. Unlike, many other security technologies, browser isolation does not rely upon detection, instead it assumes that all web content which flows through it is risky.

For the time that a user needs to gain access to the risky web, the Browser Isolation solutions effectively remove the browsing session from the user's device, isolating that user from any risks on the web. Different solutions then use different approaches to relay the browsing session back to the user.

Applied correctly, Browser Isolation can potentially remove a whole class of cyber threat – which is why so many organisations are using or exploring it, whether for high-risk users and sites or across the enterprise. However, not all Browser Isolation tools use the same techniques and technologies, nor do they all offer truly robust security.

Full isolation technologies use **pixel-pushing techniques** with a two-system isolation platform to offer robust security and high levels of compatibility. To qualify as full isolation the solution must have a verifiable pixel gap.

Partial isolation through transcoding and DOM (Document Object Model) mirroring or remodelling, offer a lesser level of security to maintain an acceptable user experience and overall cost.

This Buyers Guide explores the differences between these competing Browser Isolation technologies and gives you the information you need to evaluate them and decide which is the best fit for your organisation.

What is a Verifiable Pixel Gap?

"Pixel-pushing" creates an interactive video stream to the users' browser.

New advances in hardware-based pixel-pushing can now enable full isolation with an easily verifiable pixel gap - a physically enforced separation between the user and the web - that delivers a powerful combination of security and usability alongside lower costs and management overheads. The hardware-based verifiable pixel gap as part of a cloud service eliminates the need for you to deploy any specialised hardware.

Why are organisations adopting Browser Isolation?

Web browsing poses a significant risk to any security-conscious enterprise. And as phishing scams and ransomware become increasingly sophisticated, the threat to users only grows.

For instance, Verizon's Data Breach Investigations Report found that 36% of cyber security breaches involve phishing attacks; 11% more than the previous year[1]. And today, Google Safe Browsing lists just under 2.1 million websites as dangerous[2]. Crucially, Google's list only includes the dangerous websites we know about. The unknown threats could be far greater in number.

Protecting users from the growing ranks of malicious pages is notoriously difficult, as user behaviour always includes elements of unpredictable human error. Trying to train users not to click on malicious links or visit dangerous web sites will not be effective in stopping today's sophisticated attacks. It only takes one user to make a mistake and the enterprise could become compromised. You cannot realistically expect all your users to never make a mistake.

There are only a handful of ways to truly secure a user that browses the public internet. You can use a carousel of separate sacrificial devices (and deal with the costs and administrative burden of constantly replacing malware-infested hardware). Alternatively, you can adopt a Browser Isolation solution. Any other method, whether it be firewalls, secure web gateways, web filter lists, endpoint protection, or training to educate users, still involves your users directly accessing web pages with their devices – which could present a risk.

Browser Isolation avoids these issues entirely by ensuring your users' endpoints never connect to a web page at all. Instead, a remote machine accesses web pages on your users' behalf and delivers a separate, clean version of the web pages.

[1] https://www.verizon.com/business/en-gb/resources/reports/dbir/

[2] https://transparencyreport.google.com/safe-browsing/overview?hl=en_GB&unsafe=dataset:1;series:malwareDetected,phishingDetected;start:1148194800000;end:1612080000000&lu=unsafe

Considerations if not using a Browser Isolation solution

While your organisation may be considering a Browser Isolation solution, you might run into resistance to the need to implement one. The table below has some questions and considerations that you can use to help explore and justify the need for Browser Isolation.

Question	Consideration
Have you identified high risk users? Consider: payment teams, privileged access users, executives, developers, investigations teams, legal teams, operational teams, field engineers, etc.	High risk users should include those who are administrators, security staff or have access to sensitive systems (like payment systems) or data. At the very least, these users should be using a Browser Isolation solution. Carrying out an exercise to identify high risk users can surprise a lot of organisations as to the risks they are potentially exposed to by their staffs' roles and the access they have.
What risk is incurred by any users or websites not using a Browser Isolation solution?	If you are not comfortable with this remaining risk, consider implementing browser isolation for these as well.
Have you determined the risk to the enterprise if any user's endpoint is compromised?	Attackers may be able to escalate their privileges or spread malware once they get access to a single endpoint in your enterprise. Even the compromise of a low-risk user can be exploited as an entry point to travel laterally within your organisation.
Is the enterprise heavily relying on user training to stop phishing attacks?	If so, the enterprise is likely incurring an unacceptable risk. Phishing remains a top vector for successful attacks, including ransomware. Even with the best training, you can't expect all users, to detect all attacks, all the time. Browser Isolation allows users to click links without concern.
Is the enterprise relying on endpoint protection and browser security to protect against phishing and browsing dangerous websites?	If so, the enterprise is likely incurring an unacceptable risk. Attackers regularly test their malicious code against the endpoint protection systems and browsers on the market. Browser isolation does not rely on detection and protects you against both known and unknown malicious web content.
How are you protecting users from browsing dangerous websites?	If you overly restrict users from browsing, they may not be able to do their jobs. Conversely, allowing users to browse potentially dangerous sites without Browser Isolation opens the enterprise up to the risk of attack, including ransomware.
Out of caution, do you block websites that users would like to access (like Reddit or YouTube)?	If you are blocking websites that would let users better do their jobs, Browser Isolation can let you safely re-enable access to such sites.
How do you ensure that users who employ their personal devices to access your enterprise applications can safely browse and click on URLs and not become infected?	If you allow users to connect to enterprise applications from their personal devices, Browser Isolation can ensure that those devices cannot become infected from malicious URLs or websites.

Full or partial isolation: similar names, very different results

There are two broad schools of thought around how to isolate users' web browsing:

Partial Browser Isolation strips the website code down to a smaller subset of information to remove malicious code or other parts of a website that could be compromised. That data is then reconstructed to better resemble the original website before being sent to the user. This type of process is enabled by transcoding technologies such as DOM mirroring or remodelling and network vector rendering.

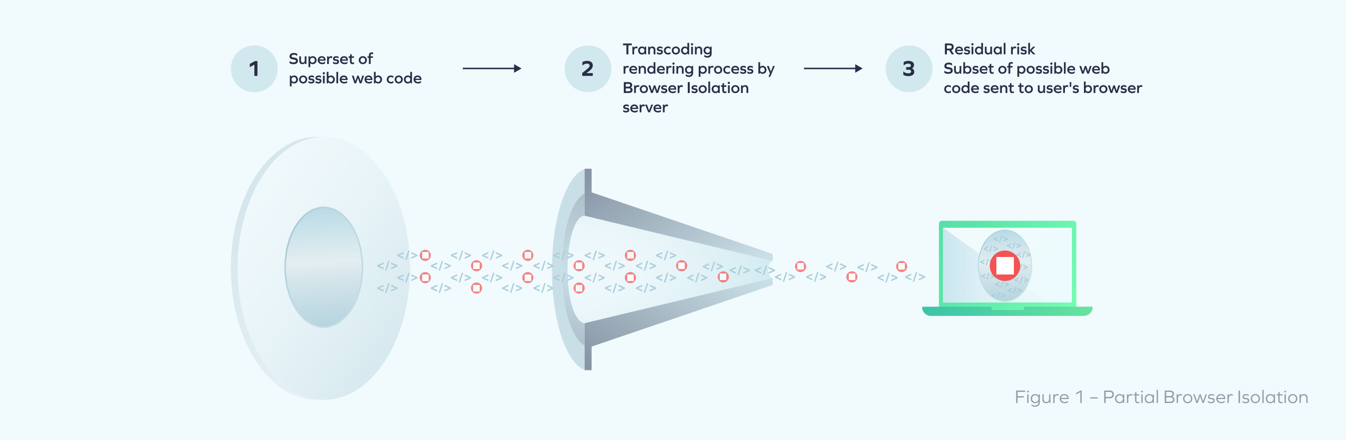

1 Superset of possible web code
2 Transcoding rendering process by Browser Isolation server
3 Residual risk Subset of possible web code sent to user's browser
</>

Figure 1 – Partial Browser Isolation

Full Browser Isolation involves completely separating users from the websites they browse. The Browser Isolation solution handles the browsing in its entirety and delivers the information to users as an interactive video stream of pixels that includes none of the website's original code. This is enabled by a technology called pixel-pushing. Generally video streams are encoded and delivered via software, but more modern solutions use dedicated hardware to ensure robust security, improve the user experience and reduce the cost. For full browser isolation there must be a verifiable pixel gap.

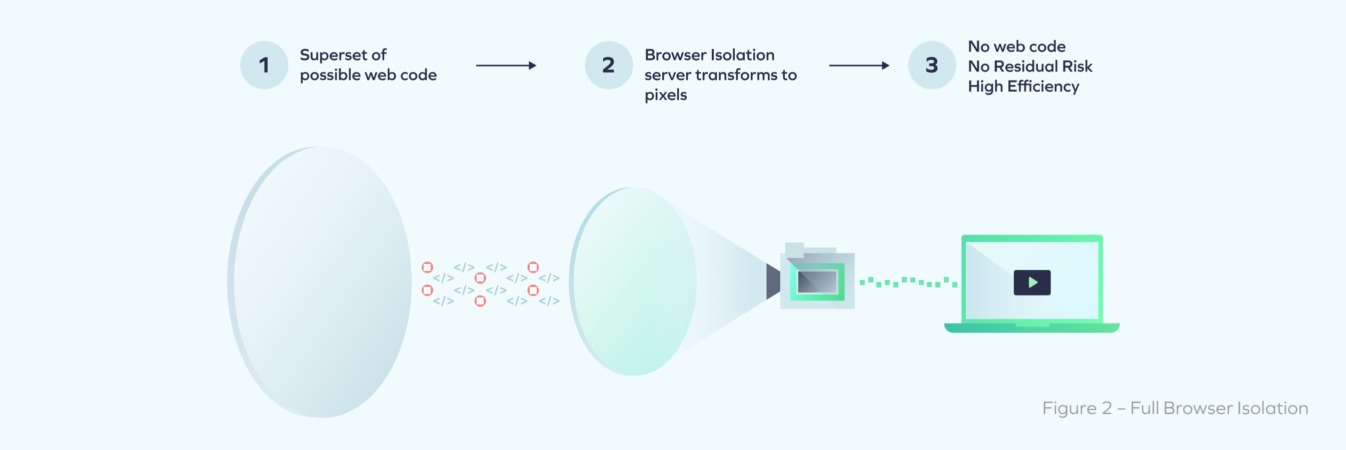

1 Superset of possible web code
2 Browser Isolation server transforms to pixels
3 No web code No Residual Risk High Efficiency
</>

Figure 2 – Full Browser Isolation

Of course, both approaches have the same end goal: delivering a secure web browsing experience. But they offer different results in terms of the web experience users receive, IT management and costs, and – most critically – the level of security provided. Moreover, even with a pixel-pushing method, an improperly architected or implemented Browser Isolation Platform may still result in just partial browser isolation if it lacks a verifiable pixel gap. This will be explained in greater detail in a later section.

Browser Isolation as Part of SWG or SSE

Sometimes browser isolation is part of a Secure Web Gateway (SWG) or Secure Service Edge (SSE) solution. It is important to evaluate the browser isolation component itself, not just in combination with the SWG or SSE solution, Otherwise, the other capabilities of the SWG or SSE solution may mask weaknesses in the browser isolation component.

For instance, the SWG or SSE may only send traffic it considers risky to the browser isolation component so that most traffic that it considers safe bypasses the browser isolation and is sent directly to the user's browser. While this will improve performance and usability, it does raise some critical security questions:

- How does the solution decide what is a risky website or content? Likely this will be through some type of URL filtering or malware detection mechanism.
- How is this different than just relying on malicious website and malware detection for defence? It isn't much different since anything these detection methods miss will bypass browser isolation anyway. In essence, such a solution violates the principle of Browser Isolation not to rely on detection.
- What if the risky website decision is incorrect? In such cases, the risky content will be sent directly to the user's browser potentially infecting that endpoint.

The point of full browser isolation is to transform unknown, risky web content into something that is safe. In other words, because it does not rely on detection, it can handle all malicious web content that traditional detection methods miss. In fact, you should assume that attackers have access to leading SWG, SSE and other security solutions and have already tested their attacks against them to ensure they will work. Today's sophisticated attackers (particularly those backed by organised crime and nation states) often have labs with copies of leading security solutions against which to test. A better security approach is to send all web content through Browser Isolation or, at the very least, all web content for high-risk users.

A solid Browser Isolation solution will still provide the desired security even if attackers have complete knowledge of how it works and test their attacks against it.

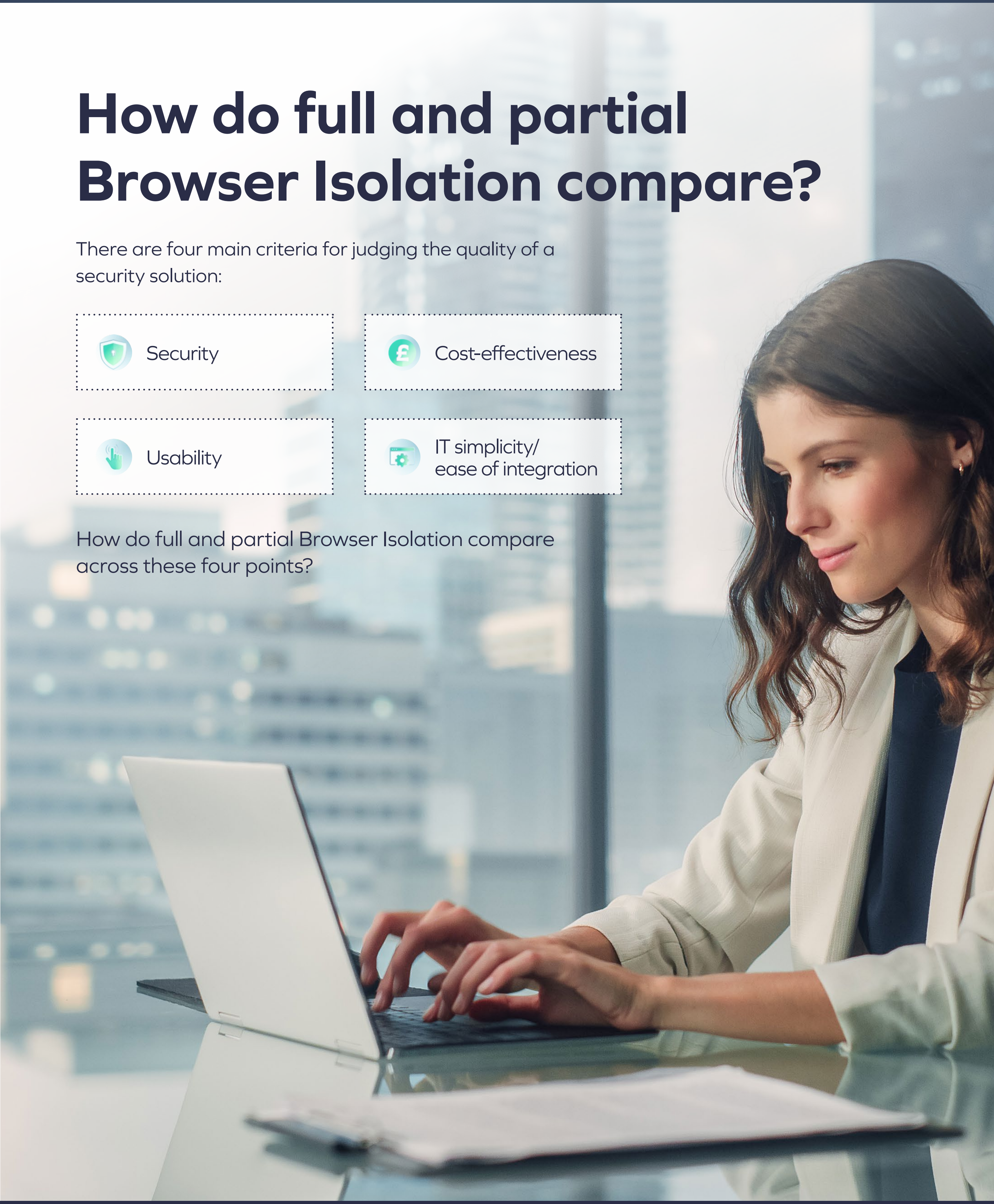
In addition, it will be hard to evaluate the performance and usability of browser isolation if you aren't sure if the web content went through the browser isolation or was sent directly to the user's browser. It is important to evaluate the true strength of the browser isolation on a standalone basis.

How do full and partial Browser Isolation compare?

There are four main criteria for judging the quality of a security solution:

- Security
- Cost-effectiveness
- Usability
- IT simplicity/ ease of integration

How do full and partial Browser Isolation compare across these four points?



Security

Nobody considers Browser Isolation technology unless they’re serious about securing their key users. The first criteria, then, is what protection the solution offers.

Browser Isolation solutions work by keeping the user away from potentially harmful website code. So, when you’re considering how effective they are, the main question is what code – if any – still makes it through to the user’s device.

Partial Browser Isolation

Because transcoding presents a subset of the original code to users, it’s inherently porous. The effectiveness of the security depends on which parts reach the user, and what gets stripped out. You’re likely to have questions about these decisions, as they determine the potential for malicious code to slip through the net or for attackers to exploit the site in a new way.

Unfortunately, transcoding is generally a black box: solution providers rarely explain exactly what subset of website code gets used. Security then becomes a matter of trusting the vendor without being able to verify how it works.

Moreover, even a pixel-pushing method may still only provide partial browser isolation if the browser isolation platform is not robustly architected and implemented. If the browser isolation platform consists of only a single system that translates the web traffic to pixels, then should that system become compromised, it can be used by an attacker to send something other than pixels to the end users’ browser - such as malicious code - and thereby compromise that endpoint.

In addition, even if two systems are used in the browser isolation platform, if any web content other than raw pixels (and raw Pulse-Code Modulation audio) can be sent between them, there is still the potential for the system that connects to the web site to send malicious code to the trusted system that connects to the end user’s endpoint. The correct way to implement a browser isolation platform for Full Browser Isolation is described below.

Full Browser Isolation

Full isolation technologies like pixel-pushing are inherently non-porous because they prevent users from interacting directly with any website code. All web content is transformed into a harmless video stream of pixels. These Browser Isolation technologies therefore offer far more comprehensive security.

What to look out for:

- Security-conscious enterprises will want to look for the improved protection and transparency that Full Browser Isolation offers.
- Partial Browser Isolation can be secure, but it depends on how such a solution transcodes websites.
- If you are interested in partial isolation solutions, make sure the provider can offer transparency around how their transcoding works and what website code it lets through to users. Otherwise, the enterprise is incurring an unknown level of risk despite the use of a Browser Isolation solution.
- Even if the browser isolation solution claims to use pixel-pushing, it still needs to have a properly architected and implemented Browser Isolation Platform to provide Full Browser Isolation. Otherwise, it only provides Partial Browser Isolation.

Security of the Browser Isolation Platform

For full browser isolation, the Browser Isolation Platform needs to be implemented as shown below, with two systems: System A that is assumed to be compromised and System B that is always trusted. Moreover, the transfer format between the two should create a "pixel gap" analogous to air gap security techniques to ensure that only raw pixels—and therefore no code—can be transferred from system A to system B.

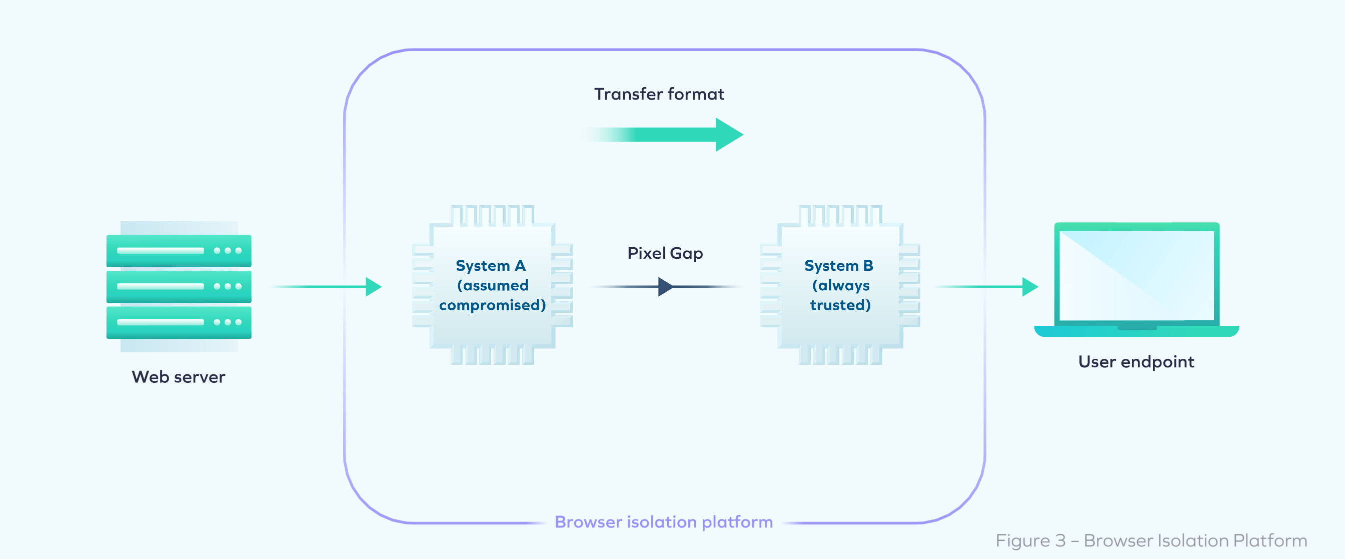

Transfer format
Web server
System A (assumed compromised)
Pixel Gap
System B (always trusted)
User endpoint
Browser isolation platform

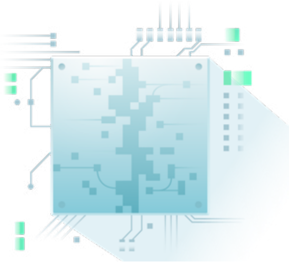
Figure 3 – Browser Isolation Platform

Hardsec uses FPGAs to provide a video display on system B and the video camera on system A that captures the displayed pixels. Even if System A is completely compromised, this pixel gap ensures that no malicious code can reach system B or the user endpoint.

Remember: Full Browser isolation must have a verifiable pixel gap.

Some providers may tout a single system Browser Isolation platform as implementing a sandbox. However, history has shown that attackers have regularly found ways to escape sandboxes and infect systems. For example, see CVE-2020-65724 that describes a vulnerability that permitted an attacker to escape a Chrome sandbox.

Note that audio content from a website can be handled in an analogous fashion as the pixel gap for visual data. Only raw Pulse-Code Modulation audio should be transmitted across the gap between System A and B. Audio player and recorder FPGAs on system A and B can assist with implementing this audio gap.



What is hardsec?

Hardsec is a security architecture described at http://hardsec.com. Instead of CPUs, hardsec uses lower-complexity (non-Turing-machine) digital logic to implement security, avoiding the inherent vulnerability that lies in the flexibility of software. By making use of FPGA silicon, hardsec can deliver security while maintaining the flexibility to address real-world cybersecurity problems in a cost-effective manner.

Usability

Browser Isolation can have two main impacts on usability:

1. **Incompatibility with websites can break the user experience.**
2. **Sending traffic between the client and the Remote Browser Isolation solution can add latency.**

These are both key issues. If your users don't enjoy their browsing experience, they may try to find a way around your Browser Isolation solution – creating new security risks.

Partial Browser Isolation

In some cases, partial isolation technologies offer acceptable latency levels. But this is situation-specific: in many cases latency can be poor and variable, particularly where technologies use protocols that aren't optimised for real-time communication. Transcoding can also create significant compatibility issues. Some kinds of content – like video playback, for example – may not work at all, or only function with a limited set of features.

While website and plugin developers constantly update their code, transcoding solution providers must continually update their systems to keep pace. When they fall behind, website features (and even entire websites) can stop working, significantly degrading the user experience. Some vendors will be tempted to fix such incompatibilities by allowing the unsupported code to simply pass through—increasing the security risk.

Finally, for some websites, transcoding solutions can be bandwidth-intensive, meaning they don't work well under poor network conditions. This is particularly noticeable on sites where transcoding tricks don't work well. In these cases, such providers have to either pass the web traffic through (a security risk) or fall back to pixel-pushing – for which their technology is typically not optimised, unlike true pixel-pushing solutions.

Full Browser Isolation

Pixel-pushing technologies avoid compatibility issues as they don't interact with website code – they instead turn the entire content into an interactive video stream that's sent to the user in real time.

Historically, these streams demanded high bandwidth resulting in significant latency and a degraded browsing experience. This remains true of many software-based pixel-pushing solutions, but new, advanced hardware-based solutions mitigate much of these bandwidth requirements.

Such solutions use specialised hardware to compress and stream video feeds more efficiently to help reduce latency and deliver a seamless, acceptable browsing experience. And by hosting the solution in the cloud, enterprises can get the security and usability benefits of modern pixel-pushing without worrying about deploying and maintaining hardware.

What to look out for:

- Hardware-based pixel-pushing solutions can offer the best balance between latency and compatibility – and the most consistent user experience.
- Many vendors will lock you to pre-set web addresses for their trial period – limiting your ability to test the service in normal browsing conditions. So, when assessing solutions for usability, make sure any demos or trials let you test the service on all websites.
- Ultimately, usability is subjective. The only way to decide which experience your users will enjoy is by testing different solutions.

Cost-effectiveness

Cost will always be a vital concern when assessing security solutions. If a Browser Isolation solution's upfront or ongoing costs are too high, it could limit your ability to scale. And if licensing models are inflexible, it can affect how you decide to roll out and deploy solutions across different user groups. And between the technology licence itself, the computing resources, and the bandwidth connectivity costs, there can be a lot to consider here.

Partial Browser Isolation

Different partial isolation solutions will use different transcoding approaches to protect users, so ongoing costs can vary between vendors. While some partial isolation solutions may keep bandwidth requirements down, many rely on transcoding approaches that can be compute-intensive – leading to significant infrastructure requirements and costs.

Vendors will also approach licensing and scalability differently, so it's worth calculating the potential costs if you decide to roll the service out to more users than you initially planned.

Full Browser Isolation

Traditional, software-based pixel-pushing isolation moves significant data volumes, which can be compute and bandwidth-intensive and lead to high operating costs.

But new, hardware-based full isolation solutions significantly reduce those ongoing costs. And cloud solutions running on purpose-built hardware can offer the same benefits without the need to pay for isolation devices upfront.

What to look out for:

- Hardware-based pixel-pushing solutions can offer lower ongoing costs compared to software-based alternatives or partial Browser Isolation.
- Where possible, use a vendor that will offer licensing based on concurrent active browsing sessions instead of per user. This type of usage-based licensing allows more flexibility in how the solution is deployed (for example, the enterprise could have a large group using the solution less frequently or a smaller group of intensive users for a similar cost).

IT simplicity/Ease of Integration

Whether it's through initial deployment requirements, or ongoing manageability and integration issues, you'll want to be sure the Browser Isolation tool keeps things simple for technical teams – and doesn't divert IT resources from other essential work.

For instance, having an 'Allow' list of less risky sites (O365, ServiceNow, Salesforce etc) and pushing everything else through a browser isolation solution can also relieve the burden on your helpdesk having to continually service requests to sites that users require and investigate the risk of that access.

Partial Browser Isolation

The key issue with transcoding solutions is that many aren't designed to work alongside existing proxies and secure web gateways. And those that claim interoperability with such security tools may still need extensive configuration to ensure everything integrates and works together correctly.

Even when a solution is properly configured and integrated, the low compatibility of transcoding-based solutions can put pressure on IT to answer a greater volume of support tickets, as users encounter websites that don't work. This may put pressure on IT to bypass the Browser Isolation solution causing security risks.

Full Browser Isolation

Unlike transcoding approaches, full Browser Isolation doesn't need to modify entire chunks of website code to deliver pages to users. While there's still a risk of incompatibility, there's a much lower chance of new updates to websites breaking the underlying method of Browser Isolation. And that means there's less need to constantly install update patches.

And depending on the solution vendor, the upfront deployment requirements can also be easier than with partial isolation solutions.

Hardware-based alternatives vary in IT complexity, but on-premises options require upfront installation and deployment. By comparison, hardware-based solutions hosted in the cloud don't have this need – although, like any solution, some configuration is still required to ensure interoperability with proxies and other security tools.

What to look out for:

- In general, full isolation solutions are more consistently compatible with websites and more readily designed to integrate with other security tools – so they demand less of the IT department.
- Some organisations will want the additional control of deploying their Browser Isolation tools on premises.
- Those that don't have such stringent requirements can further reduce IT management burden by opting for a hardware solution hosted in the cloud.
- But think carefully about how the Browser Isolation solution will integrate with other security solutions, such as the proxy and secure web gateway.

Full Browser Isolation and full security – without the drawbacks

While partial and full Browser Isolation technologies each have their pros and cons, there's no doubt that organisations that put security first are likely to consider the latter.

Some firms may be willing to explore a less secure solution if they believe it will offer other usability, cost and management overhead benefits. However, this often tends to be a box-checking exercise rather than truly adding security.

However, for most security-conscious enterprises, hardware-accelerated, full isolation – delivered through the cloud – offers the best combination of security, user experience, IT management, and cost.

Browser Isolation Solution Requirements

After reading the previous sections, you should now have a good understanding of the different Browser Isolation methodologies and their pros and cons. The requirements and response considerations in the four sections below can assist you in evaluating various Browser Isolation solutions against:

- Security
- Usability
- Cost-effectiveness
- IT simplicity/ ease of integration

Security

Reminder:
Full isolation: pixel-pushing with two-system isolation platform with a verified pixel gap
Partial isolation: rendering, transcoding or a single system platform

Security Requirement	Response Consideration
Is this a full or partial Browser Isolation solution?	A full Browser Isolation solution ensures that no web code will reach the endpoint. The next two sections have additional specific requirements depending on the answer to this question.
Describe how the pixel gap (if one exists) can be easily verified to only pass pixels and PCM audio bits.	If the solution does not include a verifiable pixel gap, then you cannot be confident in the browser isolation it provides and must consider it only partial browser isolation.

Security Requirement	Response Consideration
Partial Browser Isolation solution section (Transcoding or Rendering)	**Likely not 100% Browser Isolation.**
Describe exactly how the solution transcodes website code into a subset of website code.	Many vendors will not share this information.
Share the list of the subset of website code.	Many vendors will not share this information. If you don't have this, you have no way to verify how secure the solution is.
Share the report from a pentester, or a trusted 3rd party expert that has verified the subset of website code will ensure that no malicious code could leak through.	Validating this will require skill, time and an understanding of how malicious code could be transmitted via a website.
Explain how you assure that the subset of website code will never allow potentially dangerous code to be transmitted.	Need to understand how the vendor keeps the solution updated and compatible. What assurances do you have that no shortcuts will be taken in the future that could compromise security?
Full Browser Isolation solution section (Pixel-pushing with properly architected Browser Isolation Platform)	**Should provide 100% Browser Isolation.**
Does the solution use specialised Web-Isolation Hardware?	Offers additional security and performance advantages. See hardware section.
If a software-only architecture, describe how the solution ensures that only a video stream of pixels will be transmitted to the user's endpoint.	Even the most sophisticated attacker should not have a way to send code through the Browser Isolation solution to the endpoint. Ensure you can verify the robustness of any claimed pixel gap.
Is audio sent only as raw Pulse-Code Modulation (PCM) or similar format?	Just as only pixels should be sent for the display, only raw PCM or similar format should be used for audio—just the raw audio bits—and no content that could be executed as code.
Is the browser isolation platform split into two systems – one that is assumed compromised and one that is always trusted? There should be a "pixel gap" between the two which only allows a video stream of pixels to flow.	If the browser isolation platform is only a single system, it may become compromised and send malicious content to the end user's system. By having two systems and only sending an interactive stream of pixels between them, even if the system connected to the website becomes compromised, nothing can be sent across the gap.
If split between two systems, can only pixels flow between them?	There should be a "pixel gap" between the two which only allows a video stream of pixels to flow. If any other content, even rendered or transcoded content, flows between the two systems the potential for malicious code to compromise the "trusted" system, and thereby the end user, exists.
Is the pixel gap implemented with hardware or software?	The strength of the pixel gap is easier to evaluate if implemented with hardware. Solutions that claim to have a pixel gap implemented with just software are hard to verify and have a greater risk of compromise that destroys the claimed gap.
Full Browser Isolation solution section (Pixel-pushing with properly architected Browser Isolation Platform)	**Should provide 100% Browser Isolation.**
Describe how the specialised hardware uses Hardsec principles . See https://hardsec.com/.	Hardsec provides additional trust in the security of the hardware architecture and implementation.
Provide an easy-to-understand description of the hardware architecture.	Reliable vendors will describe in sufficient detail how the hardware architecture works.
Describe how the hardware architecture ensures that only a video stream of pixels will be transmitted to the user's endpoint	Even the most sophisticated attacker should not have a way to send code through the Browser Isolation solution to the endpoint.
Does the hardware use FPGAs?	If the hardware uses chips with firmware, a dedicated attacker may find a way to have the firmware reprogrammed. FPGAs cannot be reprogrammed.
List any government security agencies or experts that have reviewed the architecture	Robust security solutions should be able to withstand the scrutiny and challenges of demanding security environments

Security Requirement	Response Consideration
Cloud-based Browser Isolation solutions	
Describe the control plane architecture of the solution	The cloud architecture of the solution should readily demonstrate robust security.
Existence of strong multi-tenant architecture and controls	A tenant must never be able to see another tenant's data or even the presence of another tenant. Moreover, a tenant must not be able to leverage the architecture as a pivot point to attack another tenant.
SOC2 certification for solution	Just piggy backing on the cloud provider's SOC2 certification is not good enough if the vendor has a cloud-based solution. The vendor needs their own to cover their specific solution implementation.
Browser Isolation policy	
Ability to specify a policy of what goes through the Browser Isolation solution	Enterprises should be able to define what does and doesn't go through Browser Isolation
Ability to specify which users' browsing sessions will use the Browser Isolation solution	Could be used to ensure that all high-risk users' browsing goes through the Browser Isolation solution.
Ability to define websites for which the solution applies	Some organisations may enable "allow-listed" sites to not go through Browser Isolation. This can save bandwidth and processing.
Policy and configuration changes logged	Standard "must have" security requirement.
Describe how the audit trail is secured	Should be protected from modification and deletion.
Provider access to logs/audit trails	The Browser Isolation solution provider should not have access to logs or audit trails, even for a cloud deployment.
2FA or MFA be required for admin access	Multi-factor authentication must be supported. Your organisation may require this.
General Security Questions	
Describe how the Browser Isolation administration accounts are managed and secured	Look for use of multi-factor authentication, no shared admin accounts, etc.
The Browser Isolation server starts new clean sessions	New browsing sessions should not have artifacts or potentially malicious code left over from previous sessions.
How is session separation maintained? Can a session be accidentally or deliberately hijacked?	Look for a clear and architecturally sound explanation of session separation and security.
Provider access to sessions	The Browser Isolation solution provider should not have any access to user sessions, even in a cloud-based deployment.
Describe how a man-in-the-middle attack is prevented.	Explanation should be clear and architecturally sound.
Users can download files and store them in an isolated location and display them through the Browser Isolation solution?	Allows a user to securely read and interact with downloaded files such as PDFs without posing a risk to the user's endpoint.
Granular content control (cut/paste, print)	Ensure that malicious code cannot reach the user's endpoint through cutting, pasting, or printing website content.
The free trial of the solution must not restrict which web sites can be accessed through the trial	If the trial restricts the web sites that can be tested, this indicates that the solution vendor is nervous about potential customers or even attackers testing their solution against malicious websites—a sign that the underlying architecture has known security drawbacks.
List any aspect of the Browser Isolation solution architecture and methodology that are treated as a black box with no available explanation as to how it works	If any are listed, this indicates that the solution vendor is likely relying on security by obscurity and that the architecture and methodology will not withstand scrutiny. The result is that the enterprise is likely incurring risk it doesn't know about.

Usability

Usability Requirement	Response Consideration
What latency does the user experience when browsing a website with the Browser Isolation solution compared to without one?	Latency should be low enough that the user experience is acceptable for regular use.
What latency does the user experience when watching a video via the browser?	The video experience should be acceptable for regular use.
Describe how the Browser Isolation solution lets the user to provide input to the web site.	The method should allow a comparable user experience to when the user is not using the Browser Isolation solution.
Consistency of user experience	Pixel-pushing solutions are more consistent since they don't have to transcode or render a website into something different.
No to low end user training needed	End user training should be unnecessary or minimal (although training of administrators of Browser Isolation will be needed).
Users can use the Browser Isolation from both personal and corporate devices	Users should have an easy way to safely browse the web from a personal device.
List known popular websites that are not compatible with the Browser Isolation solution	Ideally there are no such sites that are needed by business users.
Support for common browsers	Should support at least Edge and Chrome.
Support of basic browser functionality (history, favourites, auto complete)	Deficiencies in this area will negatively affect user acceptance.
Support for browser session settings like cookies	Deficiencies in this area will negatively affect user acceptance.
Support for common end user devices	Look for at least PC and Mac support. Mobile support is a plus.
Does the solution work with VDI	It should work with VDI.
Support for safe file sharing or downloading	Should support this capability. Since downloaded files could contain malicious code, this might require use of an e-mail security solution to scan files
Administration Requirement	
Rate the ease of configuring	Configuration should be straightforward and easy to ensure that it remains secure.
Rate the ease of defining policies	Policy definition should be straightforward.
Availability of out of the box, pre-defined policies	Pre-defined polices can ease and speed up deployment
Rate the flexibility and robustness of policy definition	There should be enough flexibility in the way polices are defined to implement the solution the way you want

Cost-Effectiveness

Cost-Effectiveness Requirement	Response Consideration
Understand the cost for a typical deployment or your deployment	Look for flexible licensing and pricing options so that you don't have to pay for more than you need.
Consumption cost model	Licensing based on consumption should be possible.
Availability as a stand-alone solution	If the Browser Isolation solution must be purchased as part of a suite or bundle, it will cost more, both to purchase and deploy.

Cost-Effectiveness Requirement	Response Consideration
If deploying on-premise, what hardware must be purchased and what is the likely cost	Rendering, transcoding, and software-based pixel-pushing solutions are likely CPU intensive and will require sufficient hardware to ensure low enough latency for a good user experience. Hardware-based pixel-pushing will require buying a certain number of specialised appliances.
How much IT administrative time must be allocated to managing the solution	Need to understand the personnel requirements to support the solution.
Timeframe to set up a PoC	Should be very quick.
Time needed to deploy in production	A cloud deployment should be relatively fast. An on-premise deployment may take longer due to the need to purchase equipment.
Proven scalability	Solution should have been deployed at other large customers, to ensure confidence that PoC or pilot can scale.
Affordable Total Cost of Ownership (TCO)	Calculate TCO from license cost, maintenance cost, systems cost, deployment cost, infrastructure cost, and people cost. Must be an affordable amount and reasonable per user.

IT Simplicity/Ease of Integration

IT Simplicity/Ease of Integration Requirement	Response Consideration
Available as a cloud solution	Cloud is easier to deploy, especially if the solution uses specialised hardware.
Available as an on-premise solution	Allows the enterprise more flexibility if both on-premise and cloud deployments are available.
Zero-endpoint deployment	Deployment should not require installing any code on the endpoint (no agent, special browser, or browser plug-on needed).
Does not require additional security solutions to function	Should not require additional solutions to provide security like Endpoint Detection or Extended Detection and Response (EDR and XDR) for core functionality
Lightweight integration	Straightforward and easy to integrate with existing network infrastructure
Force desired websites to always go through Browser Isolation	Using proxy redirect or other techniques, the Browser Isolation solution should be able to enforce that enterprise users always have Browser Isolation enabled for browsing selected web sites. Optionally, notify the user when going to such a site.
Allow desired web sites to not go through Browser Isolation	Ability to configure so that selected allow-listed (or non-blocklisted) web sites do not use Browser Isolation when browsed .
Likelihood of website updates being incompatible with the Browser Isolation solutions	Transcoding or rendering solutions are more likely to be incompatible with an updated or new website.
Time it typically takes to fix an incompatibility issue	Should be only a few hours once reported.
Easy updating	Updates to the Browser Isolation solution are easy to obtain and deploy. This may be unnecessary for a cloud deployment, but will be necessary if on-premise.
How does the solution identify and authenticate users	Should work with ADFS, Azure AD, and other SAML-compliant IdPs.
Integration and interoperability with other solutions like firewall, proxy, and secure web gateway	Understand how easy it is do integrate and what is available out of the box.
Provide SLA (Service Level Availability)	Should have at least 99.9% availability for cloud solutions.

GARRISON

Email info@garrison.com
UK Telephone +44 (0) 203 890 4504
US Telephone +1 (646) 690-8824