All Roads Lead to the Browser: A SANS Buyer's Guide to Browser Isolation

Matt Bromiley

A SANS Buyer's Guide

All Roads Lead to the Browser:
A SANS Buyer's Guide to Browser Isolation

Written by **Matt Bromiley**

May 2020

Sponsored by:

Cyberinc Corporation

Defending modern enterprises often requires advanced thinking and out-of-the-box solutions. In our experience, legacy security policies and implementations, such as outdated security tools, lack of visibility and/or a failure to protect the end user, are equally failing to adequately equip the information security team to perform their duties. This leads to failure to mitigate or prevent intrusions that, by many accounts, could have been neutralized with a simple implementation. Think back over the past 24 months—multiple high-profile enterprise breaches have been traced to a single entry vector that could have been easily protected.

We want to reverse this trend and change the way you see one of the most integral parts of your environment: the web browser. By far one of the most commonly used applications within any organization, browsers are often your users' go-to tool for accessing resources such as webmail, the corporate intranet and external sites. However, browsers are also one of the most common applications that threat actors use in the early stages of an intrusion. Furthermore, as organizations move various operations to the cloud, dependency on the browser is even more prevalent than before—thus, we are labeling the browser as the **new endpoint**.

In this guide, we examine the concept of browser isolation and its potential role within modern organizations. A relatively young concept within the information security world, browser isolation focuses on limiting the impact that a browser can have on a victim system. However, isolation itself is not new: Consider air-gapped networks, which have simply disallowed a connection as a security measure. This, however, is not sustainable in modern computing, which is connectivity- and cloud-dependent. Isolation has been somewhat cumbersome or computationally heavy, requiring significant resources per user to achieve any form of isolation success.

Luckily, we believe current implementations are utilizing technology in a way that makes isolation seamless to the user, but extremely cost- *and* time-effective for the security team. Instead of heavy endpoint agents monitoring and intercepting browser activity, technology has allowed these solutions to be more seamless within the organization and provide native browser support. This has created new possibilities to assess whether you should secure perhaps one of the most vulnerable and far-reaching attack surfaces within your network: the browser!

How to Use This Guide

This buyer's guide is meant to help you determine whether browser isolation is a good fit for your organization and your security posture. As we explore various concepts around browser integration, we focus on questions about how browser isolation may impact the environment. This guide is broken into the following sections:

- **Alone, yet surrounded—**How does browser isolation work and what are the key factors to look for in various solutions? Furthermore, how does or should browser isolation impact the user experience?
- **Separate, yet integrated—**How does browser isolation integrate with the rest of your environment and current security posture? Is isolation a standalone product or should it complement other features?
- **Isolated, yet managed—**Once browser isolation has been implemented, how can your security team manage it?

We also include multiple "Stress Test" points throughout this paper. As you evaluate browser isolation solutions, look to these Stress Test points to understand critical attributes that a solution should provide. We provide these in a way to help guide your team as it gets hands-on with a solution.

As you work your way through this guide, we encourage you to consider where browser isolation may fit within your current security model. For many organizations, inserting a layer of control and/or mitigation between the user and the browser seems like a no-brainer. However, each organization's use cases are different: Does this technology deploy a new agent or require a new appliance? What's the overhead, and does the IT team benefit from this implementation? To help answer these questions and more, use this guide to identify what features may or may not work for you.

Takeaway

Browsers pose an interesting security concern: They tend to be one of the most frequently used applications for your users, but also are commonly involved in early stages of corporate intrusions. Thus, their security must be simultaneously seamless and highly effective.

Alone, Yet Surrounded

To understand how browser isolation can be effective within your organization, it helps to again think of the browser as your new endpoint. Many endpoint security solutions focus on detecting activities such as security bypasses, exploits and commonly used attacker commands. However, many of these activities originate with user actions such as clicking, often in a browser or an email client, a link that opens—you guessed it!—a browser.

User Experience

The concept of securing the endpoint is not new at all; instead, with browser isolation, you are focusing on one of the most prolific applications in your enterprise. By adding a layer of controlled security between the user and the browser—one that remains completely unknown to the user—you lower the attack surface through which the organization can be compromised. We're not changing the concept of endpoint security; we're changing what the endpoint is.

Consider the ways users utilize browsers within your environment:

- **Credential access—**SSO services and web forms collect credentials and allow users to sign in to perform the daily duties.
- **Email—**Users access webmail sites (internal or external), click links and access sites (such as office or productivity suites) to conduct business.
- **Intranet access—**Corporate intranets often host and serve up sensitive, internal-only data necessary for day-to-day operations.
- **Cloud-based services—**Many third-party providers and cloud services are accessed by users in the browser. Thus, the new endpoint is also your primary connection into third-party services.
- **General web browsing—**Employees often browse the internet day-to-day, whether job-related or not. Have internet, will travel?

You'll notice most of the preceding items are essential for daily business operations. Impacting the user experience for any of these uses will certainly change users' ability to effectively perform their duties and create more headaches for your security team. Our first recommendation is to determine the extent to which browser isolation impacts the user experience.

Expert Advice

Browser isolation should be transparent to users. It should not inhibit day-to-day activities or business operations, such as web browser or cloud resource access. It should also be transparent for the actions users are used to performing: copy/paste, right-click, printing, etc. Keep this mantra in mind: The user experience should not be impacted!

Threat Mitigation

Aside from protecting the user browsing experience, the second-highest priority for browser isolation is actually defending against threats. Remember, browsers are complex applications with millions of lines of code that often result in vulnerabilities. Furthermore, they are integral to your daily business operations. A security tool

Stress Test #1: Browser isolation should barely impact the user experience.

Ask your evaluation team to spend time focusing on how they interact with various websites. Choose well-known internet and intranet sites that could be utilized during normal business operations.

Objective	Ensure that browser isolation allows users to interact with sites as they normally would. Examples include ensuring encrypted sites are truly encrypted, session maintenance, storing of cookies and preserving of credentials and access between common resources.
Test	Open a standard, IT-approved browser and access well-known, reputable internet/intranet sites. How well does the page load? Can users log in and are sessions maintained?
Expected Outcome	The browsing experience should feel similar to users, as if there was no isolation in effect. Note that the user experience may differ *slightly*, but it should hardly be distinguishable.

is hardly effective if it allows these threats to propagate at the same rate as before implementation. By ensuring all elements of a page are rendered and/or executed away from the user's workstation and on the browser isolation platform, we would expect to mitigate browser-borne threats. Figure 1 examines some of the most common browser-borne threats and how browser isolation should mitigate them.

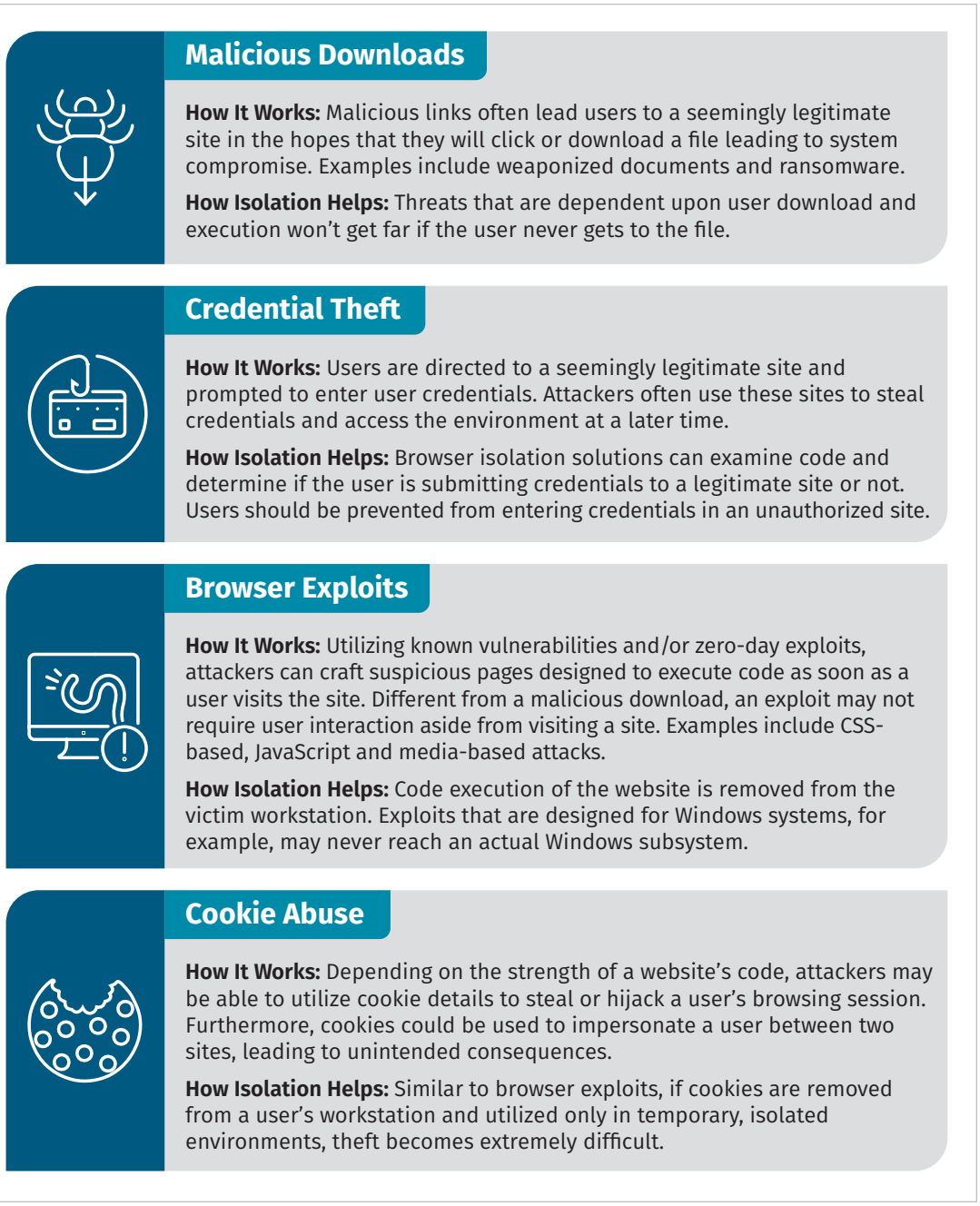

Figure 1. Using Isolation to Mitigate Common Browser-borne Threats

Expert Advice

Browser-borne threats are often dependent on vulnerable versions or fooling users into thinking they are somewhere safe. Browser isolation inserts a layer of security that can foil and/or detect these dependencies, respectively.

Of course, there is no guarantee that isolation will mitigate all browser-borne threats. However, isolation addresses a much wider range of threats than typical endpoint monitoring does by simply removing and rendering the browser code away from the workstation. Figure 2 begins with a typical flow of events—as users browse the internet, their content is passed down directly to their system.

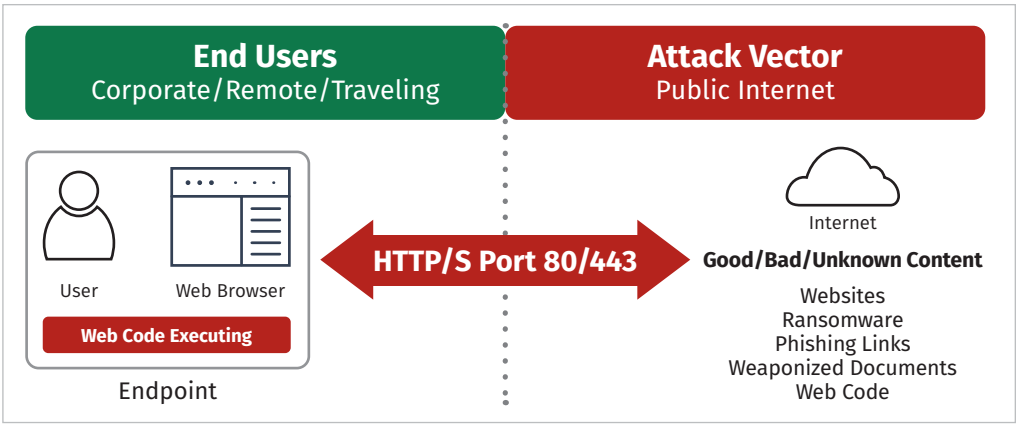

Figure 2. Browser Activity Workflow for a Typical User

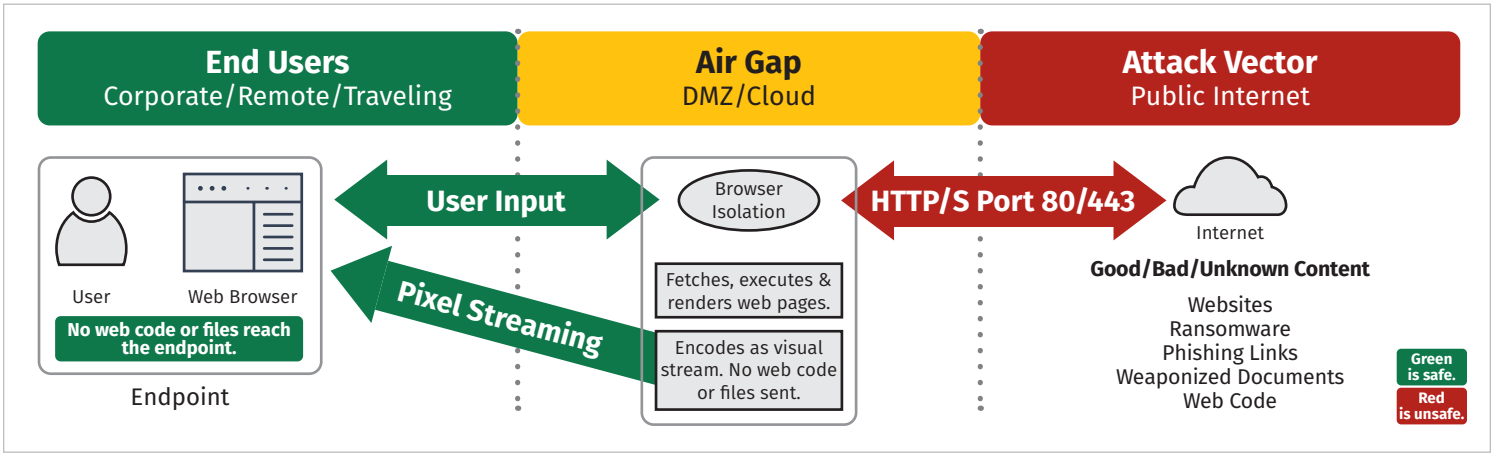

Figure 3. Browser Activity Workflow for Typical User with Isolation Implemented

The key difference highlighted in Figure 3 is that code is executed within the browser isolation platform, not on the user's system or within the user's web browser. Only safe content should be returned to the user, thus preventing exploitation of well-known vulnerabilities.

Thus, we arrive at our second stress test: defending against common attacks. A quick note: You may need to employ the assistance of a security vendor or a red team to ensure that your browser isolation defends against common threats. We don't want you to plan, build or stage an attack that simply may run amok within your organization. **Do this inside of a controlled environment.**

Stress Test #2: Browser isolation should defend against common attacks.

Because browser isolation performs code analysis and page loading away from the victim system, common attacks such as browser exploits should not impact the victim workstations.

Objective	Confirm that browser code execution occurs on the isolation platform, not on the victim system.
Test	Open a vulnerable browser version or one that is not approved by IT policy. Navigate to a site known for exploit dropping. (Note: You may need to work with a security vendor or the solution provider to identify a test website/exploit. You can also prop up your own website using open source security tools.) Allow, click and run all downloads.
Expected Outcome	Despite the use of a vulnerable browser, code execution should never occur on the victim system. The browser isolation platform should remove code execution and furthermore should inform the user of the activities that occurred.

We would also expect to see browser isolation success in the identification of credential-harvesting websites. Business email compromise (BEC) cases are perhaps some of the most prolific cases where simple credential theft—again, via the browser—has led to billions of dollars in damage in recent years.[1]

BEC and similar campaigns attribute much of their success to intricate phishing campaigns and credential-harvesting techniques. Similar to code execution detection, browser isolation platforms can also identify fake credential submission sites and warn users before they provide their data.

Stress Test #3:
Browser isolation should defend against credential theft.

While they are inspecting code of websites, browser isolation platforms are adept at detecting credential submission fields and determining if sites are legitimate or not—before users interact with them.

Objective	Ensure that users do not enter credentials into a fake, credential-harvesting website.
Test	Open a fake credential submission site. (Again, you may need to work with a vendor or red team to prop one up.) See if fake credentials can be submitted and harvested from the web page.
Expected Outcome	Users should not be allowed to insert their credentials into a fake, credential-harvesting website. Furthermore, users should be notified that

Defending against various browser threats while barely impacting the user experience, we feel, is a baseline must-have for any browser isolation solution. However, these solutions are meant to complement a security program, not replace it.

Separate, Yet Integrated

As mentioned, browser isolation should be an extension of your current security capabilities. With an extra layer of security added into user browsing sessions, there's a very good chance that your team will not only be able to mitigate threats earlier, but also be freed up to deal with higher severity threats to the organization. However, unless the security team can observe these data points, the data is useless.

Our next recommendation involves testing the integration capabilities of any isolation platform within the current environment. Simply preventing users from accessing malicious data or sharing credentials is half the battle; the security team needs to be aware of when and where these types of events occur. Similar to endpoint monitoring alerts, it's helpful for the team to have data points it can react to. Integrations may include:

- **Proxy—**Can browser isolation integrate with a proxy, perhaps blacklisting malicious links as they are observed, ensuring stronger security in tandem with isolation?
- **Threat intelligence—**Can the browser isolation receive and/or send malicious sites/code to a threat intelligence service?
- **Malware intelligence/sandboxing—**Can browser isolation relay its suspicious-identified scripts and code to a malware engine that can further test and/or provide feedback for the team?

Expert Advice

Ingest browser isolation data into your central logging platform or SIEM to give your security team another angle for observation and detection within the environment.

[1] "2019 Internet Crime Report," https://pdf.ic3.gov/2019_IC3Report.pdf

Simultaneously, browser isolation offers a unique viewpoint that organizations may not have had previously. Many organizations find themselves detecting threat actor incidents either in the early stages of command-execution or lateral movement. In more dire situations, attackers are detected during data exfiltration or even worse, *after* exfiltration and as a part of third-party notification.

Stress Test #4: Browser isolation solutions should integrate with current capabilities.

Browser isolation is yet another tool to prevent users from causing impact to the environment through the aforementioned attack vectors. However, if these cannot integrate and be monitored like other areas of the environment, the capabilities can be left unchecked.

Objective	Ensure that browser isolation statistics, such as alerts, hits and changes, are formatted and provided in a way that can be ingested into your current monitoring platform.
Test	Speak with the browser isolation provider and ask for guidance on integration with a SIEM or logging platform. During a test or proof-of-concept (PoC) run, actually connect platforms and ensure that your team can model off of data provided from browser isolation.
Expected Outcome	Data should be sent from the browser isolation platform to your SIEM or monitoring platform. Ensure that your team can ingest and model around

Similarly, browser isolation also helps neutralize multiple threats from MITRE's ATT&CK® Matrix.[2] The ATT&CK Matrix, which helps identify various tactics, techniques and procedures (TTPs) used by attackers, can also be thought of as an attack flow. For example, an attacker would need to gain access to an environment in order to exfiltrate data from it. An attacker would need to escalate privileges in order to run commands as an administrator, leading to additional lateral movement, etc. If these techniques can be neutralized earlier, the latter stages are barely a consideration for a mitigated environment.

However, as shown in Figure 4, there are multiple TTPs that are mitigated or neutralized when threats are prevented earlier via a lack of browser exploitation.

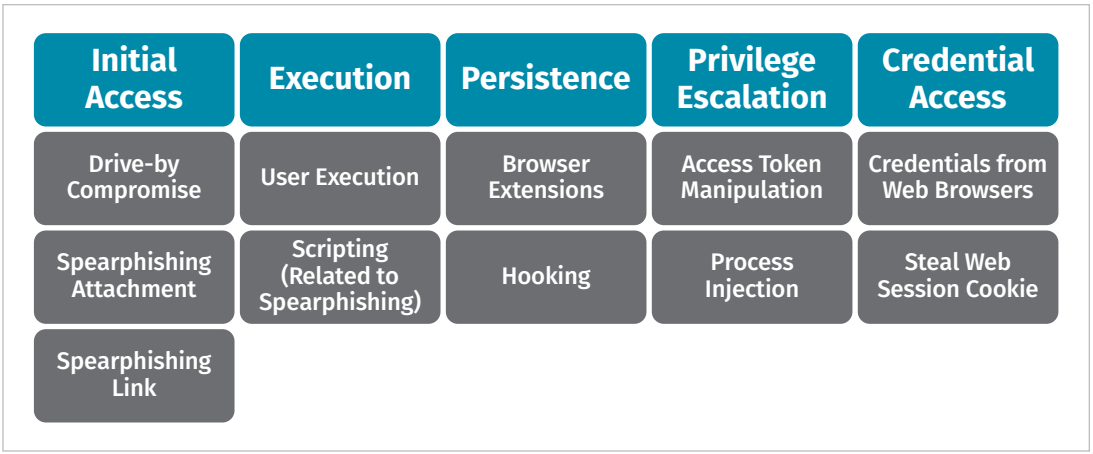

Figure 4. Techniques Mitigated or Neutralized by Browser Isolation

Browser isolation can prevent more techniques than what we've called out here. For example, an attack like ransomware may start with the browser, but end with data encryption or destruction, as well as other impacts to the organization. However, as you assess security impact, keep in mind that if we take away the browser as the entry vector, attackers may never reach the latter stages of an intrusion.

[2] MITRE ATT&CK Matrix is a trademark of The MITRE Corp.

Isolated, Yet Managed

Browser isolation should be a technology that the security team can rely on to detect and mitigate threats. Simultaneously, while being invisible to the end user, it should be highly operable and detailed for the information security team. It should be a near-seamless implementation, while remaining seamless to the users (as previously discussed). Furthermore, we recommend assessing whether you need browser isolation as a standalone or integrated component; either should be successful. This allows for customization and tailoring of the tool according to an organization's needs. After all, many information security teams know that their environments are really collectives of teams with different requirements and operations. Security solutions should operate the same way.

Expert Advice

When evaluating any security product, know your data regulatory requirements ahead of time. Always have them in hand during any meeting or PoC, and ensure that the product can guarantee compliance. If it doesn't, walk out—this isn't the tool for you.

When evaluating browser isolation solutions, look and ask for the ability to enforce custom policies or customize the tool to the needs of your organization (e.g., data retention requirements). In far too many situations, organizations try to fit their needs within the requirements of a tool, not the other way around. Remember, browser isolation impacts the day-to-day of your employees and one of their most common applications. This tool needs to be seamless to users, and it needs to work.

Stress Test #5:
Browser isolation solutions should be manageable.

Browser isolation is not a one-size-fits-all solution and may require management or tuning from your security team.

Objective	Ensure that the solution offers centralized administration and management options to customize to your environment. This should include policies targeting specific users, user groups, business units or the entire organization.
Test	Ask for access to a management platform and encourage your information security team to walk through each possible step and outcome. Enact various policies and test on specific users or user groups.
Expected Outcome	Policies should be adhered to as quickly as possible. Users should be allowed to access the resources they need for business operations, but

Finally, when implementing browser isolation, you should be complementing the rest of your pre-existing environment—regardless of security maturing. Different from integration with security, we want you to keep in mind the other architecture concerns that may be in place. Consider deployment options such as on premises, in the cloud, or physical versus virtual. How global is your workforce? Do you need to ensure redundancy and/or high availability during increased loads?

Another important consideration involves regulatory requirements that often dictate network traffic encryption and/or decryption and data retention standards. Browser isolation, while meant to increase the capabilities of the security team, should not break or invalidate any of these architectural requirements.

Stress Test #6: Browser isolation solutions should not break regulations.

Depending on where in the world your business is or what type of data you handle, your organization is likely subject to certain rules and regulations about data retention, encryption, etc. Browser isolation should keep you compliant.

Objective	Ensure that the solution is aware of and adheres to any regulatory requirements in place at your organization.
Test	Speak with the vendor team and ask about the features they have in place to ensure adherence to data standards such as HIPAA, PCI and GDPR. This may require policy implementation by the security team. Test it.
Expected Outcome	Again, policies should be adhered to. If data retention is a requirement, ensure that data is retained where specified and/or accessible by the information security team. If the tool does not adhere to a requirement your organization must adhere to, don't go any further. Regulations should not be broken, and the ability to audit organizational risk should be present.

Expert Advice

Ask your security solution providers about their future plans with respect to all of the points discussed here. A product may not satisfy all of your requirements, but purchase orders are hardly crafted overnight. Allow flexibility on both sides, and your solution may end up being everything you need.

Stress Test Checklist

Throughout this guide, we included multiple Stress Test points, which we're labeling as the must-haves as you evaluate browser isolation solutions. Figure 5 includes a summarized checklist of those points.

While not the only metrics for the success, this checklist is a good starting point as you test browser isolation solutions within your organization. At an absolute minimum, though, we encourage you to ensure that an isolation solution would empower your organization, not hinder security progress.

✔	#	Test	Notes
☑	1	Limit impact to the user experience.	Analyze the impact to the user experience by inserting the browser isolation layer.
☑	2	Defend against common attacks.	Validate that the browser isolation platform can defend against common/well-known vulnerabilities and attacks.
☑	3	Defend against credential theft.	Validate that the solution can mitigate and/or prevent credential theft.
☑	4	Integrate with current security capabilities.	Confirm that the solution will integrate with and/or empower current capabilities.
☑	5	Make it manageable for the security team.	Confirm that the security team can manage and tune the solution as needed.
☑	6	Ensure regulations and requirements are followed.	If your business has certain regulations/requirements (think HIPAA, GDPR, etc.), ensure that browser isolation keeps these in place.

Figure 5. Stress Test Checklist

Closing Thoughts

In this guide, we put forth multiple considerations for any organization considering browser isolation for their environment. Because browsers are one of the most common applications in any enterprise, inserting an extra layer of security allows the security team to lean on an extra layer of control, potentially mitigating a wide range of browser-borne attacks.

However, while isolation is a novel concept, it cannot slow or inhibit daily business operations. As stated earlier, many browser isolation concepts were such an inconvenience to the user base that they found alternatives, bypassing security controls. In this guide, we explored attributes that you and your organization should consider before implementing and during testing. These included monitoring user impact, ensuring integration with current security setup, ease of management and others.

Finally, any addition to a suite of tools is only as useful as its ability to empower the information security team to perform its required duties with more confidence or efficiency. Browser isolation should be a manageable and integrated approach to security. Your team should not have to deal with more cumbersome work with security controls—quite the opposite. Integrating controls such as browser isolation may actually have a reverse effect, providing a chance for your other tools to deal with fewer false positives and operate more efficiently. Your security team will thank you, as will your protected user base. The browser is your new endpoint; act accordingly.

About the Author

Matt Bromiley is a SANS Digital Forensics and Incident Response instructor, teaching FOR508: Advanced Incident Response, Threat Hunting, and Digital Forensics and FOR572: Advanced Network Forensics: Threat Hunting, Analysis, and Incident Response. He is also an IR consultant at a global incident response and forensic analysis company, combining his experience in digital forensics, log analytics, and incident response and management. His skills include disk, database, memory, and network forensics, incident management, threat intelligence, and network security monitoring. Matt has worked with organizations of all shapes and sizes, from multinational conglomerates to small, regional shops. He is passionate about learning, teaching and working on open source tools.

Sponsor

SANS would like to thank this paper's sponsor:

Cyberinc

Last Updated: May 6th, 2020

Upcoming SANS Training

Click here to view a list of all SANS Courses

SANS OnDemand	**OnlineUS**	**Anytime**	**Self Paced**
SANS SelfStudy	**Books & MP3s OnlyUS**	**Anytime**	**Self Paced**