Innovation Insight for Remote Browser Isolation

Published 8 March 2018 - ID G00350577 - 25 min read

By Analysts Neil MacDonald

Almost all successful attacks on users originate from the public internet and many involve web-based attacks. Security and risk management leaders can contain damage by using remote browser isolation to separate end-user internet browsing sessions from enterprise endpoints and networks.

Overview

Key Findings

- Perfect prevention of breaches isn't possible; you must assume compromise. Thus, part of your strategy must be the isolation and containment of an attacker's ability to do damage.
- The impact of web-based attacks can be dramatically reduced through browser isolation, a straightforward approach that can be implemented locally or delivered remotely from a server. Of the two approaches, remote browser isolation provides stronger isolation.
- Resetting the remote browser sessions back to a known good state after every use virtually eliminates the ability of undetected and stealthy attacks to persist beyond a single session.
- Complete isolation of users from the internet isn't practical. Some internet content will need to move from the internet into enterprise systems for interaction, editing and collaboration.

Recommendations

Security and risk management leaders operating and evolving cloud security:

- Evaluate and pilot a remote browser solution in 2018 for specific high-risk users, such as finance, or use cases such as rendering email-based URLs, particularly if your organization is risk-averse.
- Favor remote browser solutions that don't require a local agent or application to be installed, and instead use HTML5 to deliver remote sessions to the user's local modern browser for access.
- Reset browser sessions back to a known good state after every use, but favor solutions that preserve some element of user personalization such as bookmarks across sessions.
- Plan for web applications that can't be remotely presented, such as web conferencing.

- Design and implement a capability for content movement from the public internet into enterprise systems, but only after intensive scanning using multilayered threat detection techniques.

Strategic Planning Assumptions

Through 2022, organizations that isolate high-risk internet browsing and access to URLs in email will experience a 70% reduction in attacks that compromise end-user systems.

By 2022, 25% of enterprises will adopt browser isolation techniques for some high-risk users and use cases, up from less than 1% in 2017.

Analysis

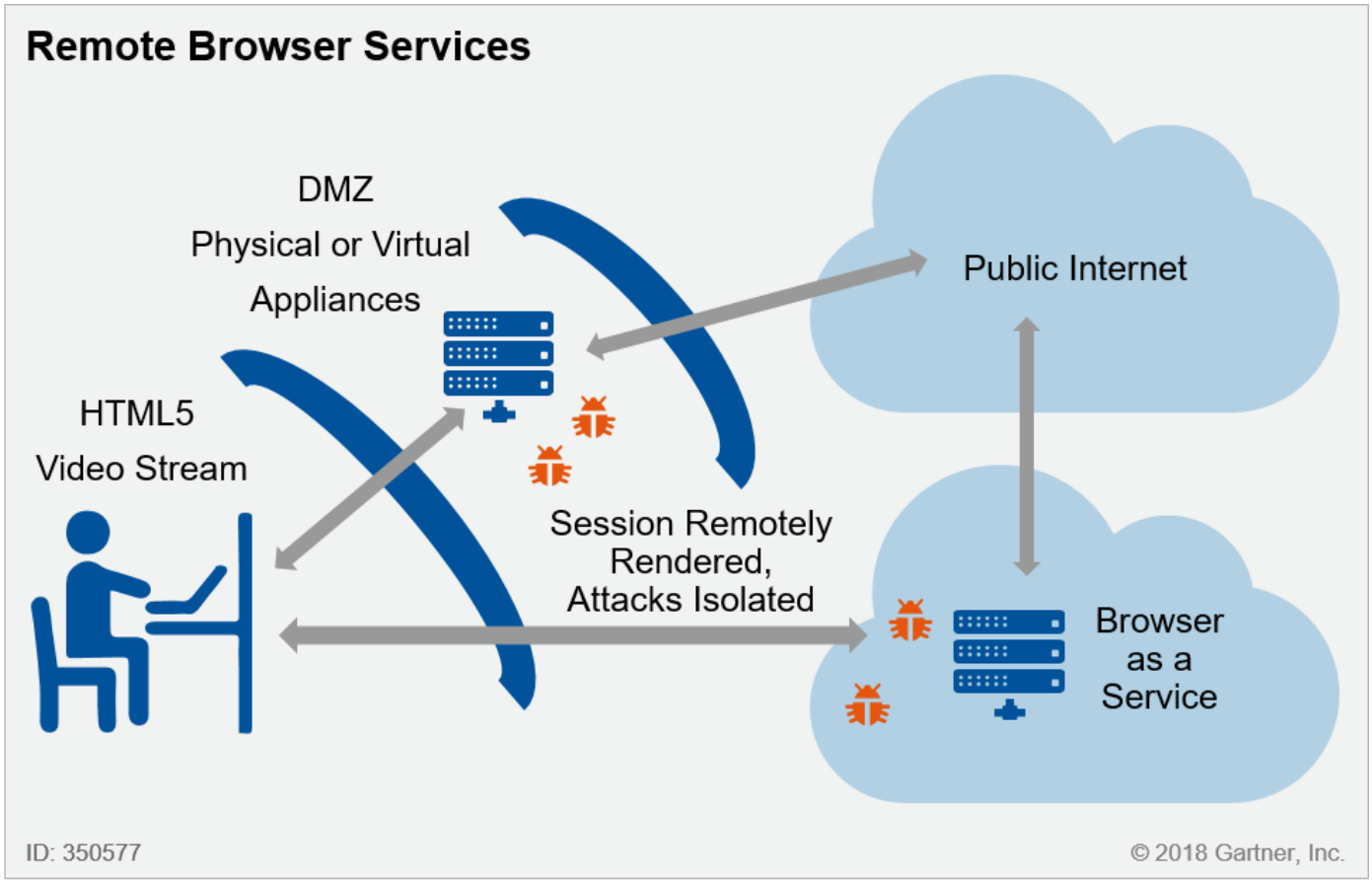
The public internet is a cesspool of attacks, many of which are delivered to enterprise users through the everyday act of browsing the web or clicking on a URL in email. Attackers are easily bypassing preventative controls, such as signature-based malware scanning, firewalls and secure web gateways (SWGs). Browser-based attacks are a primary threat vector for attackers to target users, and vulnerable web browsers and plug-ins are an easy target. No matter how good we think we are at patching and blocking attacks, we can never be good enough, as evidenced by the WannaCry ransomware attacks in 2017. New approaches to protect systems and data against breaches are needed. Gartner believes the time has come to isolate browser access from the dangers of the public internet for at least a portion of their traffic dealing with high-risk users and use cases.

Rather than delude ourselves that we can block all attacks, let's acknowledge and accept that some attacks will succeed no matter what we do. Instead, we must focus on containing the ability of the attacker to cause damage and reduce the surface area for attack. A remote browser isolates the user's internet browsing activity from the end user's device and from the rest of the enterprise's networks and systems. This effectively creates an "air gap" between inevitable attacks and the enterprise network, restricting the ability of an attacker to establish a foothold, move laterally, breach other enterprise systems and exfiltrate data. Notably, remote browser isolation can thwart ransomware attacks, blocking their ability to encrypt the users' files on their devices or in enterprise file shares, neither of which are directly accessible from the remote browser session.

Definition

Remote browser offerings are a subset of browser isolation technologies (the other category being local browser isolation; see Note 1) that remove the browsing process from the end user's desktop and move it to a browser server or cloud-based browser service. Remote browser servers then render the browser content remotely and send a bidirectional stream representing the rendered session to the end user's local browser with audio/video sent to the user, and keyboard and mouse interactions sent back to the session (see Figure 1).

Figure 1. Remote Browser Services



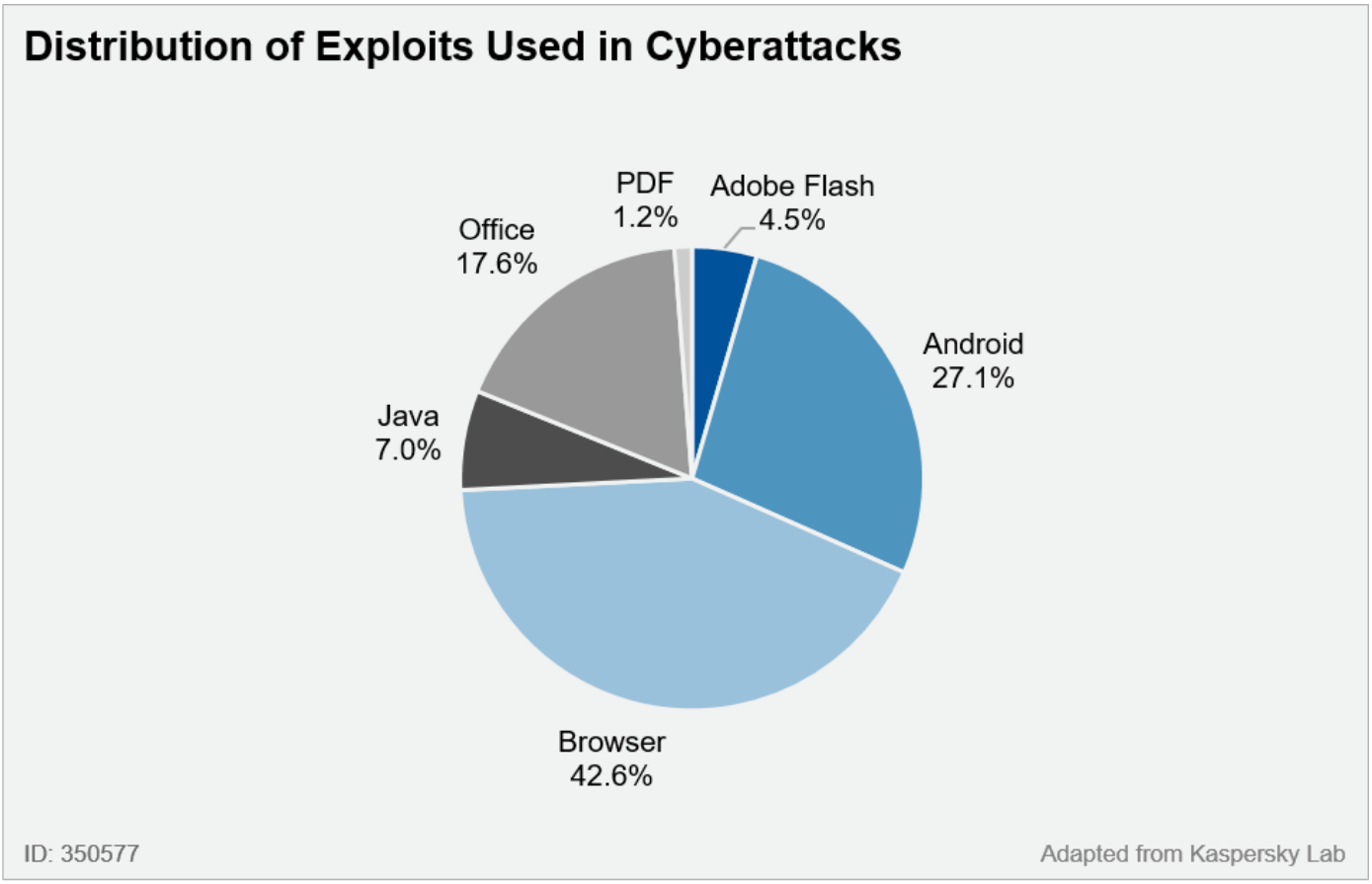
Source: Gartner (March 2018)

Description

Remote browser offerings remove the act of internet browsing off of the end user's device and into a remote server, typically on-premises in the demilitarized zone (DMZ), or delivered as a cloud-based service. In the spirit of people-centric security, the user is still empowered and enabled to browse the public internet, but because the remote browser is essentially "air gapped" from the user's physical desktop and enterprise network, any attacks on the remote browser session are constrained in their ability to cause damage. Every browser session is isolated and treated as if it might have been compromised and, ideally, every session is reset back to a known good state from immutable templates when completed.

The very act of users browsing the internet and clicking on URL links opens the enterprise to significant risk. Symantec's 2017 Internet Threat Report (https://www.symantec.com/security-center/threat-report) found that an average of 2.4 new browser vulnerabilities are discovered per day, and its labs detected an average of 229,000 web-based attacks per day. In the Kaspersky Security Bulletin: Overall Statistics for 2017 report, [1] browser-based exploits still represented the bulk of exploits used in cyberattacks (see Figure 2).

Figure 2. Distribution of Exploits Used in Cyberattacks, by Type of Application Attacked, November 2016 to October 2017



Source: Adapted from Kaspersky Lab

Attacking through the browser is too easy, and the targets are too rich. Depending on the nature of the underlying vulnerability exploited, compromising a system can be as easy as getting a user to visit a compromised website (aka drive-by attacks). Even ostensibly "good" websites are easily compromised and can be used to attack visitors. The Symantec 2017 Internet Threat Report indicates that 76% of all websites contain a critical vulnerability that, if exploited, may allow malicious code to be run without user interaction.

Further, modern browsers are a significant source of vulnerabilities, as are common browser plug-ins, such as Adobe. The surface area for attack represented by vulnerable browsers and browser plug-ins, as well as unpatched browsers and plug-ins, is significant. Gartner estimates that through 2020, 99% of vulnerabilities exploited will continue to be ones known by security and IT professionals for at least one year.

Benefits and Uses

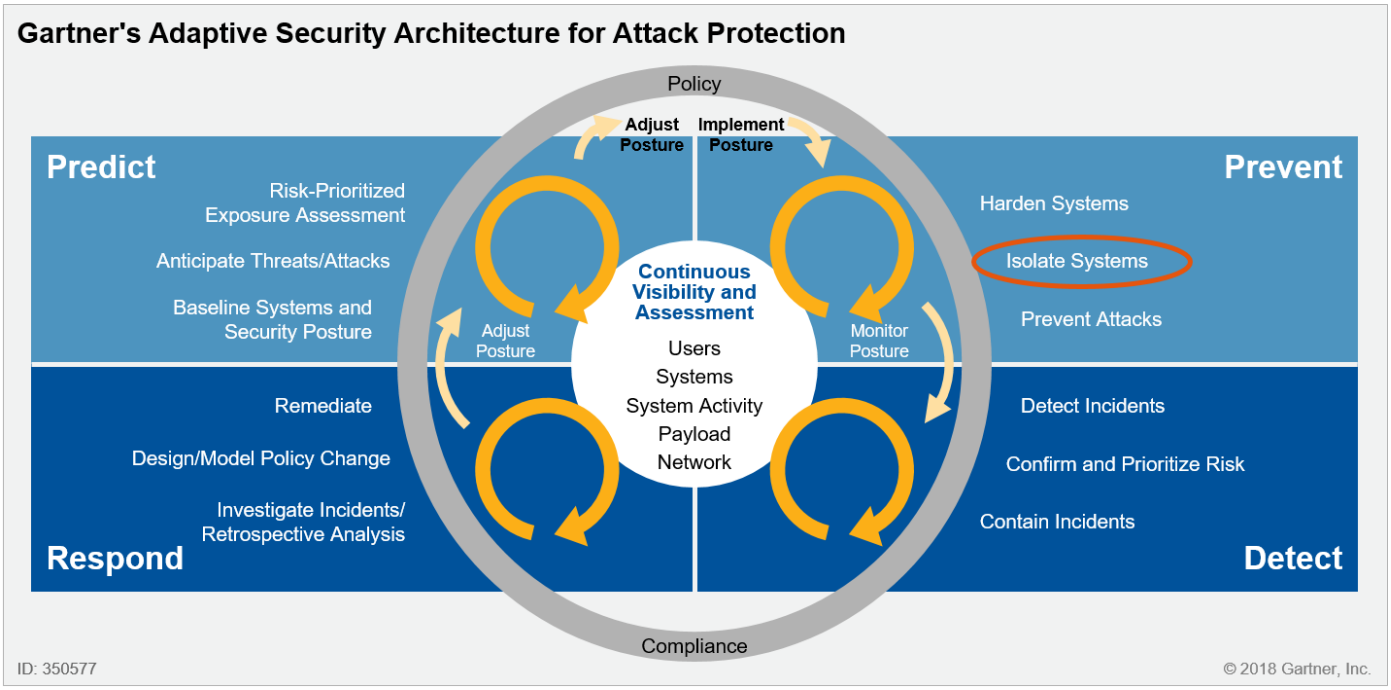
The benefits of using a remote browser service are immediate. End-user devices and enterprise systems and networks are kept isolated from the cesspool of internet-borne attacks.

Remote browsing can protect the organization from:

- Unpatched browsers and plug-ins
- Browser and browser plug-in zero days
- Targeted attacks carried in web content

Attacks will happen, but their ability to cause damage are kept isolated in the remote browser service with no direct connectivity in internal enterprise systems or data. Isolation is a key preventative strategy in Gartner's Adaptive Security Architecture for attack protection (see Figure 3).

Figure 3. The Critical Role of Isolation in Gartner's Adaptive Security Architecture for Attack Protection



Source: Gartner (March 2018)

An attack might temporarily infect the remote browser, but can't reach other enterprise systems and can't persist as attacks are removed as sessions are restored to a known good state after each use. This may also provide the enterprise with privacy benefits, as tracking cookies can be deleted when the session is deleted.

Since the vast majority of attacks on enterprises are carried over the public internet, simply moving the browsing process directly from the end-user device and getting it off of the enterprise network will reduce the impact of an attack. We estimate that organizations that isolate internet web browsing will experience a 70% reduction in attacks that compromise end-user systems. An analogy would be a person using a remotely controlled robot to open suspicious packages. If the package explodes, the damage is contained while the real user is remote, isolated and unharmed. With remote browser services, every internet website and its content is assumed to be untrusted and capable of causing damage.

Protection from hostile content carried via the web must also be a part of the solution. Even if patching was perfect, the end user can be tricked into downloading weaponized content, such as a document or PDF. Therefore, a complete remote browser service must also include remote viewer capabilities, document-flattening capabilities and integration with other enterprise malicious content scanning systems for scenarios where internet-based file objects need to be viewed and, in some cases, brought down to the user's local machine.

Protecting trusted web applications is another use case for remote browsers. The arrows in Figure 1 are bidirectional, meaning protection flows in both ways. Specifically, a remote browser solution could be used to protect a trusted web application from untrusted users, inverting the flow of protection. In this way, a potentially compromised user's device is kept isolated from directly accessing the intranet web application or other systems, helping to protect against sensitive data exfiltration to unmanaged and untrusted devices. Some emerging vendors target primarily this use case, protecting internal applications form attacks by external and untrusted users.

Adoption Rate

The remote browser category is embryonic, with enterprise adoption of less than 1% today. However, in discussions with clients, interest in this type of approach is growing. Enterprises are growing frustrated with the ongoing stream of compromises of end-user systems (and subsequent attacks on enterprise back-end systems), where the root cause was traced to some type of web-based attack. In some cases, government and regulatory bodies are moving to require remote browsers (for example, the Singapore government moved to require remote browsers for "internet surfing separation" for its employees in 2016 [2]). A "default deny" (zero-trust) posture is being adopted where possible for systems and network connectivity. Browser isolation is a key part of this strategy, and both local and remote browser isolation approaches will be used.

We believe browser isolation solutions for internet access strike the right balance of enabling users to access the public internet (people-centric security, a "default allow" posture) with reducing the risk of web-based attacks by isolating these interactions from direct interaction with end-user systems and enterprise networks (a default-deny posture). Windows 10 will have an impact here in raising awareness and adoption as Microsoft includes local browser isolation in 2018 in Windows 10. We believe that, over the next five years, local and remote browser isolation of at least some user browsing (for example, high-risk URLs, emailed URLs and high-risk users) to the public internet will be adopted by 25% of enterprises.

Risks

The biggest risk in adopting a remote browser offering is that a remote browser breaks the user experience. Since most of the remote browser services use Linux because of licensing issues, they cannot run Internet Explorer (IE) or Safari browsers. Therefore, any web applications that are IE-specific or Safari-specific won't work. However, browser-specific applications are rare on the public internet. Widespread adoption of HTML5 has reduced this risk, and all modern browsers, including IE, support it. Intranet applications can be rendered locally using IE or Safari for Macs.

Performance is potentially an issue, since sessions are remotely presented. This is most often an issue with heavy users of internet-based video sessions, such as via YouTube. Some remote browser vendors will render the video remotely and stream it (typically using HTML5's built-in codecs) to the end user. Some vendors may "pass through" the video stream for local browser rendering, but this is not recommended, as it opens the slight opportunity for a media-based attack. Others will convert the stream to essentially a bitmap, and send it through frame by frame.

A related issue is that the remote browsing session is more CPU-intensive as compared to traditional web proxies, and thus can be more expensive to implement and scale.

Latency is a risk. Regardless of where the browser session is rendered, there will be similar bandwidth consumed (the YouTube content, in this example), so there should not be a net increase in bandwidth requirements. The difference is where the active content is processed. Cloud-based browser-as-a-service providers will require multiple geographically dispersed points of presence with adequate bandwidth to reduce latency by keeping content processing as close to the user as possible.

Another area of risk is dealing with web-based content. Browser-borne content and file objects, such as documents, represent a risk if the content needs to move to the user's local device. This is discussed in detail in the Evaluation Factors section.

Applications that need access to local browser context and richness will create issues. For example, web applications that use geolocation will return the location of the browser server, not the user. This can create a significantly negative user experience if, for example, the user is in France, but the browser service is based in Germany and search results default to German. Applications that need access to the user's local microphone or camera can't be remotely presented (for example, WebEx or Live Meeting). None of the remote browser services currently support these. Videoconferencing websites must be whitelisted so they render locally.

A loss of browser isolation is also a risk. The browser servers will themselves become a target for attack. If containers are used, a compromise of the host OS could lead to a loss of isolation. If virtual machines (VMs) are used, a compromise of the hypervisor would result in a similar situation. If the browser service is delivered as a service from the cloud, then similar issues with cloud-based multitenancy are present — and complicated, as now sessions must be kept sufficiently isolated from other tenants. Choosing a solution built on a leading infrastructure as a service (IaaS) platform such as Amazon Web Services (AWS) or Microsoft Azure helps to reduce risk. However, if the browser servers use containers, customers should look for architectures where single VMs are not used to host containers from different tenants. Hardware-based solutions may offer the strongest isolation; however, hardware costs are a consideration, which is why many vendors favor Linux containers to achieve higher densities.

Finally, the browser service becomes a single point of failure for user access to the internet if the remote browser server or service is down. High-availability architectures for the service are imperative.

Evaluation Factors

When evaluating remote browsers, the key questions to ask are:

1. Is a local client/agent required? Ideally, the answer is no. No agent or application on the user's local device should be needed, other than a single, modern browser that supports HTML5 (most often for enterprises, this will be the built-in IE or Safari browser).

2. What web rendering engine does the browser service use, and how is it kept up-to-date with changes in HTML5? Ideally, the vendor has chosen an industry-standard rendering engine that is kept up-to-date with modern web standards.

3. How does the browser service support plug-ins? Which plug-ins does it support? At a bare minimum, support for PDFs and Flash is needed. Some organizations may require support for client-side Java with a local Java Virtual Machine (JVM) running in the browser server (where the application is then rendered to the user's local system).

4. How well does the remote browser support cloud SaaS applications such as Office 365 or G Suite? Is the latency acceptable? Does the vendor recommend whitelisting these sites (see Note 2)?

5. When the end user encounters file objects on the public internet, does the browser service provide a remote viewer? For what document formats? Another option is to flatten the content coming in from the public internet to remove potentially malicious code that might be embedded within it, such as taking a Word document and converting it to a text file, PDF or HTML5 before moving. [3] Another example is a technology Gartner refers to as "content disarm and reconstruction" (CDR), based on standards (see "Market Guide for Secure Email Gateways").

6. When end users encounter file objects on the public internet and they need to take the content natively to their local device (for example, they need a PowerPoint file from the public internet), does the browser service offer options to scan the content for malware? If the content is a document, it could be disarmed by using CDR approaches. [3] Or if the content is executable code, before the content can be moved to the local device, it would be scanned for viruses using multiple techniques: first, using multiple antivirus signature engines like VirusTotal, then scanned using machine-learning-based malware detection engine, and then detonated in a network sandbox. If the content passes all layers of inspection, it can then be moved locally. Test the performance of this mechanism — lengthy delays beyond a few minutes could encourage users to seek ways to circumvent the system.

7. When the user wants to cut/paste data from the internet, what happens? Some organizations will want to disable cut and paste entirely. Ideally, the text is flattened, and rich objects, such as embedded Excel, Word or PowerPoint objects, are removed and cannot be cut and pasted. Rich objects may be allowed, however, if embedded executable code such as macros are removed.

8. Does the vendor use Windows or Linux servers to provide its browser service? Who is responsible for licensing the OS? Patching the OS? Most vendors will likely use Linux because of the licensing issues associated with Windows. This provides an additional security benefit, because most web-based attacks are designed for Windows.

9. Does the vendor use full VMs or containers for the browser sessions? VMs provide stronger isolation, but at a higher cost in terms of resources and startup times for new sessions.

Containers boot more rapidly, support higher densities and could be used for each tab opened, but with a lighter-weight isolation between them.

10. Is the browser session set back to a known good state for each new user session? How about for each tab opened within a given user session? This should be a mandatory requirement. If and when an end user's browsing session is compromised, there should be no ability for the malware to persist once the session is complete or the tab is closed.

11. Since sessions should be reset back to a known good state, how are user personal preferences and settings (such as home page, bookmarks and font size) saved and persisted from day to day for a given user? Some elements should be persisted to improve usability. Some cookies for common sites (for example, a personal banking site) should be persisted, but unknown and unwanted third-party cookies should be removed, based on policy.

12. How is web video content, such as YouTube, handled? Specifically, is the video rendered remotely with a compressed stream sent over HTML5, or is it passed through the media stream to the user's local browser and rendered locally? Ideally, the content is rendered remotely to reduce the small threat of media-based attacks on an underlying vulnerability in the user's local audio/video codecs.

13. Bandwidth-related questions should be asked, but typically are no different from when the session data is carried directly to the end-user system. In both cases, the session information is sent, with remote browsing, and the session is flattened to keep out attacks. Beyond HTML5, determine what protocol or codec is used to carry session information to the user's device (for example, H.264, PC over Internet Protocol [IP], Remote Desktop Protocol [RDP], Independent Computing Architecture [ICA] or similar).

14. How are web-conferencing applications such as WebEx handled? These applications need local microphone and camera access. No remote browser service yet supports this, and the answer will likely require that these specific internet-based services be whitelisted (see Note 2) and run natively on their local browsers, opening a small surface area for attack.

15. If the browser service is cloud-based, what cloud infrastructure does it use? What geographic distribution does it have for the workloads? Are the browser service's workloads located next to internet points of presence with dedicated bandwidth? Is regional affinity an option for your users? Is the architecture multitenant, or do you get browser servers dedicated to your company?

16. How will mobile users be handled? Will they be directed to the nearest cloud-based browser service? How?

17. How strong is the isolation architecture? There is a trade-off between isolation and performance. Document Object Model (DOM) mirroring solutions still render some content locally, opening up avenues for attack. Solutions that use HTML5 could still be attacked over HTML5 if the browser server is compromised.

18. What is the architecture for high availability to avoid creating a single point of failure?

19. How is the decision made to render locally using the local browser, versus rendering remotely? Typically, intranet sites are rendered locally and internet sites are rendered remotely, and this is transparent to the user, using either the IP address, URL or integration with the SWG to make the decision.

20. Does the vendor have a SWG offering or partnership for traffic that isn't remotely presented? Alternatively, the buyer may want remote browser to replace other services from existing SWGs (see "Magic Quadrant for Secure Web Gateways"), so the vendor's SWG-like capabilities should be evaluated (see Note 3).

21. How are embedded links in email messages captured and rendered? Many email-borne attacks also use URLs, so these should be rendered using the remote browser service, not rendered locally. This needs to be tested with email vendors that rewrite URLs as a security precaution.

22. Does the vendor have a roadmap item for remotely rendered email integration?

23. If the isolation is being used in the reverse direction to isolate enterprise apps and data from unmanaged devices, are enterprise mobile applications supported? Windows applications? Or only web-based applications?

Remote Browser Alternatives

It is critical that enterprises understand the value of actively isolating public internet browsing. However, a remote browser solution is not the only way to achieve browser isolation.

Full virtual desktop infrastructure (VDI) is an alternative. Windows might be used, but the licensing required to equip each user with a Windows desktop (or even a single Windows IE icon) is significant. Although Microsoft has delivered containers for Windows, no vendors yet offer a solution based on this architecture.

Local browser isolation is an alternative using Windows containment approaches (see Note 1). The local browser is used, but kept isolated from the rest of the desktop using several approaches:

- A full VM could be run locally using a hosted OS model, but like the VDI alternative, this is a costly approach if Windows is used. It has significant hardware requirements and represents a significant surface area for attacks. [4]

- Bromium uses a lightweight hypervisor ("microvisor") approach to isolate all processes (each browser tab gets its own isolated process) on the Windows system. This approach still has significant hardware requirements, but is more manageable than using separate VMs.

- Lighter-weight containment approaches are available from Avecto. For example, Avecto lowers the privileges of the browser session and the applications handling internet content.

- On Linux desktops, the browser could be run deprivileged in a container. Multiple Docker images area available for this. [5]

- On Windows 10 enterprise systems, enterprises can run the Edge browser in a virtual session that is isolated from the OS services. Microsoft has announced it is bringing Windows Defender Application Guard to Windows 10 Pro edition users in the next feature update of Windows 10 in 1H18. [6] The appeal of this is limited: isolation works only on Windows 10, only supports the Edge browser (not Internet Explorer), requires virtualization-based security be activated on hardware that supports chip-level virtualization and requires the properly licensed version of Windows.

A hardened browser could be run locally to reduce risk [7] or on a hardened, dedicated browser-based system, such as a Google Chromebook (see "A Secure Introduction to Chrome OS in the Enterprise"). Some have caused compatibility issues, and any code written by humans, no matter how hardened, will contain embedded vulnerabilities that will be targeted. Likewise, to be functional, these browsers require plug-ins to handle content such as PDFs, audio/visual files and similar content, and these plug-ins themselves also contain vulnerabilities that will be targeted.

Alternatively, an enterprise could harden its standard browser image, standardize a set of plug-ins, prevent arbitrary plug-ins, supplement with an SWG and focus on best-in-class patching to keep the residual risk as low as possible. However, even with hardened browsers, content-based attacks carried over the browsing session remain a risk — including the growing threat of ransomware. To address this, all active content from the internet could be flattened, [3] but this complicates the deployment.

If an enterprise simply wants to reverse the isolation model for web-based applications, it typically uses a web application firewall (WAF). To isolate nonweb mobile enterprise applications from unmanaged mobile devices, solutions such as Avast's Virtual Mobile Platform (https://www.avast.com/virtual-mobile-platform) or Hypori by Intelligent Waves (http://www.intelligentwaves.com/hypori.html) could be used.

Recommendations

The public internet represents significant risk, and compromising end-user systems to gain a foothold into enterprise systems is an ever-present possibility. Therefore, now is the time for enterprises to consider isolating end-user systems from direct internet access for high-risk users and use cases such as email-based URLs. If your enterprise is looking for new approaches to this pervasive security problem, Gartner recommends using the evaluation factors listed above as the starting point for the evaluation and piloting of at least two competing pilot solutions in 2018.

Specific recommendations include:

- Don't start with all users.

  - Focus on higher-risk individuals more likely to be targeted, such as in the executive office, research and development, or finance (for example, payment processing).

    - Start with fixed desktops. Laptop-based users should be migrated in future phases after fixed desktops are completed.

- Don't remotely present all internet traffic initially. Most organizations start with a subset of the riskiest URLs to remotely present:
    - URLs that have not been categorized by their SWG, or that carry a low reputation score
    - URLs of nonwhitelisted sites (assuming the organization has developed a list of commonly used sites)
    - URLs embedded in email as part of a broader anti-phishing strategy
- Favor vendors that don't require a local agent to be installed. Insist on standard HTML5 support for the rendering of the sessions within the user's local browser.
- Require vendors to use an industry-standard web-rendering engine, such as WebKit, and not a proprietary implementation, to avoid rendering incompatibilities.
- Test email client integration so that when potentially malicious internet-based URLs are sent within emails, they are rendered remotely rather than locally. Ensure compatibility with URL rewriting software if this is used to protect email. Note that Proofpoint acquired remote browser isolation vendor Weblife for this integration.
- Favor vendors that offer a choice of either an on-premises-based browser server deployment or delivery as a cloud-based SaaS offering, and, ideally, support hybrid deployments using both approaches for different types of users and locations.
- Ensure that you (or the service provider, if the browser is provided as a service) patch the systems well, as hackers will target the base OS or hypervisor used in the browser server. Require whitelisting-based lockdown of the core Linux OS to protect from attack.
- Require browser sessions be set back to a known good state on every new session. Ideally, require a new container to be created for each tab opened.
- Pressure SWG vendors to provide this capability as a logical adjacency to their existing URL and malware-filtering protection. Symantec has already done this with its acquisition of Fireglass.
- Sign short-term contracts only for periods of 12 to 24 months; the market is embryonic.

Representative Remote Browser Providers

- Authentic8
- Citrix

- Cyberinc (an Aurionpro company; acquired Spikes Security)
- Ericom Software
- Garrison (ARM-based hardware appliance)
- Light Point Security
- Menlo Security
- Oodrive
- Proofpoint (acquired Weblife)
- Randed
- Symantec (acquired Fireglass)
- WEBGAP

Evidence

[1] Insider attacks are not considered external attacks. External attacks are those attacks initiated by an entity external to the enterprise. The vast majority of external attacks are network-based, using the internet to carry the attack, including attacks involving weaponized content. Approximately 2% of attacks are executed via direct access, typically carried on removable media, external hard drives or devices directly synced to the user's system (see the 2016 IBM X-Force Threat Intelligence Report (https://www-01.ibm.com/marketing/iwm/dre/signup?source=ibm-WW_Security_Organic&S_PKG=ov44518&S_TACT=102PW24W&dynform=21349)).

[2] See "Singapore Hit by 16 Waves of Online Attacks Since April Last Year," (https://www.gov.sg/news/content/today-online-spore-hit-by-16-waves-of-online-attacks-since-april-last-year) Singapore Government, 10 June 2016.

[3] Examples of content disarm and reconstruction (also referred to as "content sanitization") solutions:

- Check Point Software Technologies SandBlast Agent (https://www.checkpoint.com/products/endpoint-sandblast-agent/)
- Clearswift Advanced Protection Threat (https://www.clearswift.com/solutions/protecting-critical-information/advanced-threat-protection)
- Glasswall Solutions (https://www.glasswallsolutions.com/)
- OPSWAT (https://www.opswat.com/products/metadefender/core/data-sanitization)
- ReSec (https://resec.co/)

- Sasa Software GateScanner (http://www.sasa-software.com/content-disarm-and-reconstruction/product-3/)
- Symantec (https://www.symantec.com/connect/blogs/activate-symantec-s-disarm-feature-sanitize-infected-powerpoint-attachments)
- Tresys XD Air (http://www.tresys.com/products/xd-air)
- Votiro (http://www.votiro.com/)

[4] See "10 New VM Escape Vulnerabilities Discovered in VirtualBox," (https://www.techrepublic.com/article/10-new-vm-escape-vulnerabilities-discovered-in-virtualbox/) TechRepublic, 25 January 2018.

[5] See "Dockerizing Desktop Applications," (https://linuxacademy.com/blog/linux/dockerizing-desktop-applications/) Linux Academy blog, 21 November 2016.

[6] See "Announcing Windows 10 Insider Preview Build 17063 for PC," (https://blogs.windows.com/windowsexperience/2017/12/19/announcing-windows-10-insider-preview-build-17063-pc/#LvvLYFzwQXrivWEk.97) Microsoft Windows Blog, 19 December 2017.

[7] For examples of hardened browsers, see "The Best Secure Browsers 2018," (https://www.techworld.com/security/best-8-secure-browsers-3246550/) Techworld, 2 February 2018.

Note 1
Local Browser Isolation

Browser isolation is an umbrella term that includes two primary approaches. The first category is the primary focus of this research note — remote browser isolation. Here, the movement of the actual browsing process off of the user's local system to a remote service (on-premises server, on-premises hardware or delivered as a cloud-based service). The second category is local browser isolation. Here, the browsing process is kept local to the user's system and isolate from the rest of the machine using software-based isolation capabilities, typically virtualization-based.

Example vendors include Apozy, Bromium, Hysolate, Ntrepid and Microsoft.

Note 2
Whitelisting of Sites

Nearly every vendor requires the whitelisting of sites that require local system richness such as microphone and camera access. Others will recommend the whitelisting of additional sites such as Office 365. Note the number of websites that the vendor recommends that bypass the remote browser service is a good indication of the vendor's ability to handle more complex client/server-type applications.

Note 3

SWG-like Capabilities

For some users and use cases, the remote browser may entirely replace traditional SWGs. The remote browser service should offer SWG capabilities such as URL categorization blocking, high-fidelity reporting, role-based access, group and user level policies and reporting, granular web application control, bandwidth rate shaping, quota-based controls, and file and object sandboxing. These can be offered from the vendor or through a SWG partnership.