Securing the Published Browser

Endpoint and hosted browser security guidance

June 2016

CITRIX®

Contents

Scope and Audience

This document is intended to provide both summary and detailed guidance for architecting, deploying, administering and auditing browsers in XenApp and XenDesktop hosted environments. Detailed guidance includes configurations, registry settings, Group Policy Objects (GPOs) and PowerShell automations.

The Resources and References section links to guidance for additional Citrix technologies including the Citrix Secure Browser server and service.

Introduction

Browsers are the most common published application in virtualized environments...*and also the most exposed to security issues*

Browsers are the most published application in virtualized environments – and the most exposed to security issues, as they've historically been one of the most vulnerable pieces of software on any endpoint. The security concerns with browsers are legendary and involve complex settings, third-party plugins, active content, Flash, Java, browser extensions and other components that must be kept under strict control. Virtualization through XenApp and XenDesktop provides unique methods to fine-tune browser security and protect sensitive data across web applications, compliance environments, administrative portals, email and the cloud.

Web browsers are ubiquitous - used by most everyone, every day, as a core technology for doing their jobs online. The browser provides a platform to deliver web applications to users and the richness of functionality rivals traditionally installed applications. Web applications can be written in the common language of HTML5, but are often written specific to web browser platforms that have a unique set of controls and capabilities. Unfortunately, there is a lot of bad web code that does not adhere to standards, yet still functions within specific versions of browsers (and therefore requires those specific versions). While the web browser is seemingly a standard for access – the reality is that web applications and browsers comprise an often fragile and specific set of technologies that must be carefully managed to assure functionality, availability, security and privacy.

Supportability and Sustainability

Enterprises set web standards and select target browser platforms to provide consistency to their application development and support teams, as well as to deliver desired endpoint functionality. Web applications are highly dependent on specific technical capabilities and settings - and may be tied to specific versions, plugins, support and sustainability models that may not be available in updated target platforms. Due to factors that are often outside the control of application and platform teams, support for older, less secure browsers may be required for a prolonged period of time, often due to the web application not being sufficiently updated over time. Of course, users and administrators do not want to remain on older platforms and may additionally seek to move to newer or different operating systems, such as Mac users that require Internet Explorer to use specific web applications.

One way to address these issues is by leveraging browsers deployed in non-traditional ways, such as virtualizing the browser and publishing it via XenApp. The virtualization in this case is accomplished by abstracting the browser from the operating system of the user and endpoint, removing platform dependencies and allowing the user to utilize any platform to access the web application.

Don't leave browsing to chance – *Publish specific to the use case!*

Some browser versions are dependent on a specific operating system, but need to be utilized across platforms (such as providing Internet Explorer to Linux or Apple operating systems). Additionally, older browsers may be required that are not available in currently supported operating systems. By leveraging virtualization and a published browser, multiple browsers and a variety of conflicting versions and capabilities can be delivered to support the extensive variety of browser requirements within the enterprise. Some web applications may also require specific plugins, add-on components or scripting languages to be made available, which may be considered too risky to be provided on a standard endpoint or too complicated to control and manage on a Bring Your Own Device endpoint.

Upon a merger and acquisition scenario, the combined company web platform and browser support may be in conflict, but there will be a period of time where certain employees need to leverage web applications with competing requirements. By leveraging published browsers, multiple browsers can be deployed to support diverse users and usage models, allowing use of web applications from both companies to support the acquisition. For example, a company that was just acquired has applications that only work with Internet Explorer 8, yet Internet Explorer 11 has already been deployed to the acquiring enterprise's desktops. This is not an ideal state, as Internet Explorer 8 is no longer supported, but by using an Internet Explorer 8 published browser, careful sandboxing controls the risk of using out of date components that are required to support business processes.

The evolution of browser hosting

While in the past it was very common to publish browsers for compatibility reasons, security is becoming the most important aspect driving published browsers.

As browser support complexities have increased, browsers have been hosted in order to provide the required plugins or capabilities, typically specific versions of Java or ActiveX controls required by Intranet websites. This trend reversed with the popularity of alternative browsers. Browsers such as Firefox or Chrome made extensions very popular. As a result, browsers are often published to avoid using extensions that might potentially track the content of the internal websites, or otherwise degrade security and privacy.

In addition, many organizations desire to prohibit direct endpoint access to the Internet – especially for the browser. An increasingly popular solution is secured remote browsing – delivering the entire Internet, intranet and security experience through published browsers for all employees.

Outline of this technical whitepaper

This technical whitepaper will address the following considerations for securing the published browser:

- When bad things happen to good browsers – illustrations of common security problems and their virtualized solutions
- Browser security basics – what you need to know to choose security options
- How to lock down browsers at the endpoint for accessing virtualized resources, as well as datacenter and cloud-hosted browsers

- Guidance for hardening published browsers, including group policy and PowerShell configuration of security policies
- How to tune and maintain the browser components to be application-specific and further minimize the attack surface
- The configuration tradeoffs between security and functionality, along with those tradeoffs between anonymity and auditing
- Resources and references

The guidance provided is appropriate for enterprise, home-based and third-party browser usage.

Which browser?

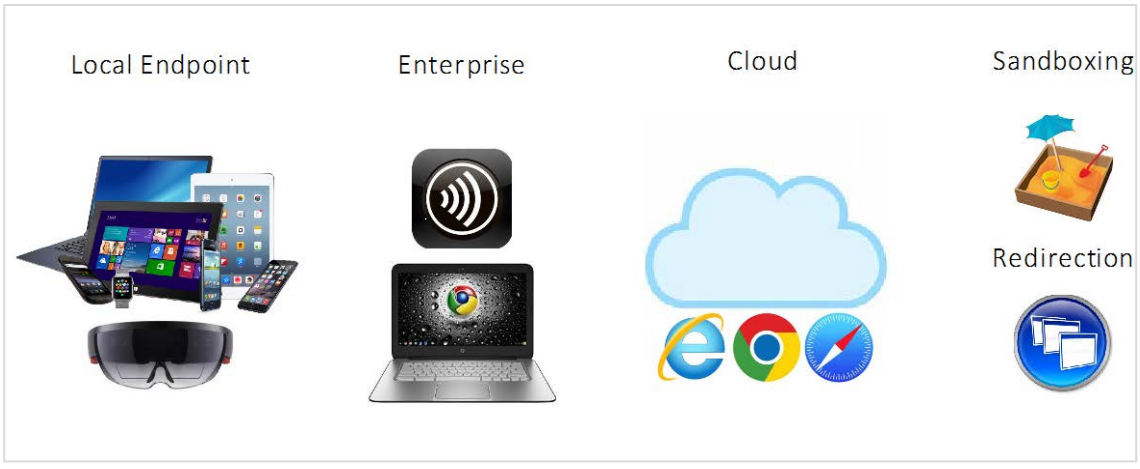
When we consider the context of the browser, it's important to understand which browser? Are we using a browser integrated with the general-purpose OS of the endpoint? A separately installed browser? A purpose-built browser appliance? A containerized or enclaved browser on the endpoint? Are we using a published browser hosted on XenApp? XenDesktop? Nested? General purpose, or application-specific and tuned to the unique needs of individual web/SaaS/Cloud apps?

As you can see, the context of the browser is critical to understanding its role in security and in making the appropriate configuration choices.

For the purposes of this whitepaper, we will separate our guidance between the unique aspects of the endpoint browser and that of the published browser, noting that there are several additional permutations to these categories.

- Integrated browsers, installable browsers, browser appliances
- Local browser access with URL redirection
- Browser sandboxing and enclaving
- Persistent and non-persistent features, including history, caching and logging

Browsers exist in many forms – both local and hosted.



When Bad Things Happen to Good Browsers

There are plenty of scenarios whereby we are presented with a difficult choice of clicking something on the web or not, with minimal or no visual indicators it is a bad choice. Some examples of this include the homograph attack where a website's certificate uses Cyrillic letters that look the same as the legitimate site, replacing the favicon with a lock symbol, displaying a different link than the actual target, overlaying buttons with invisible controls, or other clickjacking techniques where the user performs actions they were not intending.

-Eric Beiers

Browsers are the "front end" for modern applications with architectures that range across web, cloud, mobile and embedded platforms. Compromising the confidentiality, integrity and/or availability of the "front end" has a significant effect on security – and the browser is thus an integral component of application and endpoint security. And it's easy for a trusted browser to become tainted and untrusted – attacks that target the browser can deliver their payload and infect even just by visiting a site that hosts malware as a download, plugin update, coded in active content or even silently embedded in an image or video.

The following are some of the more common and interesting attacks against browsers, along with misconfigurations and vulnerabilities that enable these attacks. Additional information beyond these examples is provided in the References section of this document.

Common browser security concerns

- Ransomware
- Phishing
- Browser vulnerabilities
- Plugin vulnerabilities
- Active content
- Excessive access
- Data loss
- Session hijacking
- TLS support
- Bad certificates
- Proxies and MITM
- Flash remoting
- Privacy erosion
- Outdated browsers

Phishing

Among all browser-based attacks, one of the most popular is phishing. Phishing attacks lure their victims to click on a link that could be in an email, on a social-media page, embedded in the metadata of a graphic, or delivered through an advertisement, with increasingly creative methods being developed. Variants of phishing attacks include spear phishing (a highly-targeted personal attack) and whaling (targeting high-value victims).

The top traditional mitigation for phishing attacks has been user education via mantras such as: Don't click on untrusted links! With the popularity of URL obfuscation and shorteners, it is getting much more difficult to casually detect phishing URLs, although plugins are available that provide a preview of the final destination, by just hovering over the link. Additional mitigations for phishing include sandboxing/containerizing/enclaving/virtualizing the browser, hardening, URL redirection, web content inspection, whitelisting and blacklisting.

Ransomware

Ransomware attacks encrypt data and demand payment for the decryption key. And, if the attackers use strong cryptography and good programming techniques, recovery will require either a restore from timely backups or ransomware payment.

Mitigations for ransomware are similar to phishing, with the notable addition of including a strong file backup and synchronization regimen, as well as restricting access to filesystems to only what is absolutely necessary for point-in-time specific web application usage.

OWASP Top 10

- A1 Injection
- A2 Broken Authentication and Session Management (XSS)
- A3 Cross Site Scripting (XSS)
- A4 Insecure Direct Object References
- A5 Security Misconfiguration
- A6 Sensitive Data Exposure
- A7 Missing Function Level Access Control
- A8 Cross Site Request Forgery (CSRF)
- A9 Using Components with Known Vulnerabilities
- A10 Unvalidated Redirects and Forwards

Source: www.owasp.org

Vulnerabilities

The list of top browser and web vulnerabilities is best summarized through the OWASP Top 10, which is included here for reference. Mitigations across the OWASP Top 10 will be specified in detail throughout this paper. Remember that in web applications, code can be run at the endpoint browser, as well as in many web applications and services that constitute the application backend.

Active content

Active content frameworks including Java, Flash, Silverlight, Adobe Reader and others extend the functionality of the browser – and also the attack surface. Plugin-dependent apps cause the user to install high risk components in the quest for additional functionality. These high-risk components can be old and vulnerable versions of active content frameworks, overly-permissive settings, malware masquerading as a required update or patch, or custom plugins that directly compromise the browser and its host endpoint.

Mitigations for active content based vulnerabilities include installing only the necessary active content frameworks, carefully controlling patches and updates, hardening of the framework components and publishing application-specific browsers.

Session hijacking

Hijacking a session enables the attacker to perform actions on behalf of the browser – and therefore on behalf of the user. Browser sessions can be hijacked in many ways, including redirections, Man-In-The-Middle (MITM) attacks and via malware. The caching of credentials in the browser and access to filesystems greatly increases the risks to users and the value to attackers, making hijacking a particularly dangerous exploit. The reliance of the browser on services including DNS, network security, certificate management and the integrity of cryptographic algorithms further expose sensitive content.

Mitigations to session hijacking include the use of strong authentication to further protect credentials, application-specific networking, restricting and specifying certificate and cryptographic algorithm usage, as well as end-to-end encryption across the network for all content. Threat actor position must be factored into mitigations to thwart MITM on the network, MITM on the proxy, a different web site with connectors to the app and malware. Delivering a virtualized and sandboxed one-time-use browser substantially reduces the effectiveness of an MITM attack by prescribing and therefore limiting access to resources.

Data loss

Browser security concerns aren't only about attacks and vulnerabilities – access to filesystems, other sites, email, credentials and passwords enable both inadvertent and intentional data loss on behalf of the end-user. For example, an end-user with access to web-based email from home could copy sensitive email text and attachments to their home system with little to no restriction. Publishing a virtual browser ensures that enterprise controls and policy are enforced across any usage scenario – which is especially essential for home and third-party usage.

Privacy is also eroded as the browser retains logs and personal information, including Favorites and the History of sites visited, which can show up unintendedly during meetings and web conferences. Various cross site tracking tools exist that allow advertisers to track users without them even providing their identification to a website the first time they visit that particular website. Many browsers now report information back to the browser provider to assist with enhancing the browser experience, but also erode the potential privacy for the user. Some of this information seems harmless (consider spell check services, prediction services and usage statistics), but can be used to identify the user and their behaviors. Information shared behind the scenes may include sensitive information that the user had no intention of sharing outside their regular enterprise or personal usage.

In addition, browser diversity can cause people to overlook that they're using different browsers at home, at work, on tablets and mobile phones. Consider the situation of a desktop at home, one in the office, a personal tablet, a family shared tablet and a smartphone, there are many browsers in use – Chrome on Android, Chrome + Safari on the tablet, IE + Edge + Chrome on Windows machines and an older version of published IE to access Intranet. All of these situations potentially use different frameworks and different patch levels. Some of them may have include insecure storage of passwords and information. In some virtualized environments, administrators will simply deploy virtualized browsers and allow users to select their favorite. While this is a very user friendly approach, the permutations also mean that updates likely break functionality – which can keep these general-purpose browsers in an unpatched and unsupported state. This must, of course, be avoided by publishing browsers specific to each use case.

Browser security use cases

Understanding the intended use of the browser is critical to security-specific configuration. The following are specialized use cases that require unique security considerations:

- Browser-based email
- Location-based usage restrictions
- Non-employee apps / access
- App-specific security requirements
- Disposable and one-time use browser
- Kiosk mode
- Apps that require old browser versions
- Lockdown of cloud and SaaS apps
- Enforcement of Security Zones
- Administrative and management portals
- Enforcing mutual authentication
- Mitigating phishing
- Access virtualization
- OneURL access simplification
- IPSec deprecation for casual endpoints
- Regulatory compliance
- Implementing context-specific security

Browser-based email
Using Outlook Web Access (OWA), Office 365, Gmail and consumer email services.

Location-based usage restrictions
Different configurations and policies are required to support office workers, mobile workers, home users and geographical locations

Non-employee apps and access
Contractors, partners, suppliers and other third-parties

App-specific security requirements
Lock-down the browser with settings and plugins unique to app requirements

Disposable and one-time-use browser
One-time usage, purge everything!!!

Kiosk mode
Similar to the disposable browser and runs in an open-area kiosk intended for use by multiple parties

Apps that require old browser versions
Support for multiple browser versions often including those that have known vulnerabilities but are required for supporting specific apps and must therefore be carefully sandboxed

Lockdown of cloud and SaaS apps
Use of external apps securely

Enforcement of security zones
Publishing browsers specific to the security zone context (PCI, etc.)

Regulatory compliance
Publish the browser with settings specific to regulations such as PCI DSS, HIPAA and/or EU Data Privacy

Administrative and management portals
Enforcing virtualized access for all administrative usage

Enforcing mutual authentication
Cryptographically binding the trust between endpoint and application

Mitigating phishing
Publishing the browser to more carefully control what happens when a link is clicked or otherwise launched

OneURL access simplification
Publishing a single URL to facilitate access

Access virtualization
No direct access to sensitive resources combined with the reduction of direct network socket connectivity (e.g. eliminate IPSec as a default)

Implementing context-specific security
Placement of the browser in a specific controlled security location (implicitly allowing or disallowing access and how far an attacker can pivot) by using the 5W's of Access (who, what, when, where, why) for contextual access determination.

Browser security levels

Browser security zones stratify configurations to meet the specific needs of locations, networks, technologies, regulations, data sensitivity and use cases.

Zone	Security Level
IE Zones	• Trusted sites • Restricted sites • Internet • Intranet
Data Sensitivity	• Public • Confidential • Restricted • Classified • Material • etc.
Technology-Specific	• No plugins • Flash support • Java support • JavaScript support • No popups • etc.
Use Case-Specific	• Compliance • Social media • Third-party • General • App-specific • etc.

- The level of security is determined and implemented considering the tradeoffs between user experience, active management and realized security.
- Policy tiers are developed for browser security considering access to social media, mission-critical apps, sensitive data, internal/external, SSO, etc.
- Clipboard considerations determine how the copy/paste buffer is utilized within the application and between applications
- Security models can also be inverted, with potentially vulnerable content run on the hosted browser and the endpoint thus protected.
- Redirections and rewrites direct and transform URLs, facilitating mapping to security zones.
- With virtualization, the browser can be published according to the mission of the web application – without the undue relaxations that would otherwise be necessary to standardize on the lowest common denominator of browser configuration that is often a major enterprise security tradeoff.

Hardening

Hardening is the process of reducing the attack surface and must include in scope the web browser, published application, operating system (OS) and network.

The hosted OS will be specifically deployed for the purposes of supporting secured web browsing, and can therefore have additional specific controls and restrictions that would not be appropriate for a general purpose operating system used for other applications. Any unnecessary components and services that are not required for the purposes of browsing the web should be disabled to reduce the attack surface and resultant risk. Tools such as AppLocker are available to restrict executables, scripts and installers to only those required for web browsing. Mandatory profiles or selective profile redirection can be used to allow the least amount of persistence of any attack or cache. Since we know exactly what is required to publish only a browser, we can be much more restrictive in our operating system lockdown, compared to publishing a generalized workspace.

Running a web browser as a local administrator or with the privileges of an authenticated domain user (when it is not explicitly required by the web application), introduces unnecessary risk and requires adequate sandboxing of the native web browser. If the user is a privileged user, and their browser becomes compromised, the browser would have the privilege of the authenticated user, in which an attacker may be able to pivot from the compromised host into additional components of the network.

To create dynamic websites, developers have used a variety of features and add-ons to provide a better experience for users. Many of these features have been the source of vulnerabilities of the past, and should be restricted or disabled if not required. These components include, but are not limited to: ActiveX Controls, Silverlight, JavaScript, Java Plugins, Flash and PDF readers. Many of these strive to become more secure, and some browsers now offer locally provided sandboxed versions of these add-ons to assist with mitigation of attacks. If the user or application never requires a specific high risk component, why provide it to the user? If the user or application absolutely requires a known risky component, then utilize a published browser in a secured enclave to provide the capability, while still restricting the damage that could be done if that component was compromised. We want to enable the users to be able to complete their workflows, but anytime we introduce a restriction to the browser, we are reducing the 'availability' of certain functionality - and this may impede user productivity. We must be aware that once users are annoyed sufficiently, they will work around security policies, so we must provide ways for users to complete their work safely with the least risk to the enterprise. We also must be prepared for blocking legitimate capabilities, and have an exception process to allow the capability to be available when warranted and risk-appropriate.

The ideal way to perform hardening on browsers in the enterprise is to use Group Policy controls: either those built-in the box (in the case of Internet Explorer), from the browser vendor (Chrome) or through third party methods which extend Group Policy to manage one specific browser or all browsers.

Browser Security Policy Elements

The following are considerations and tradeoffs to consider when designing and implementing realistic policies:

Culture: Organizational policy acceptance and enforcement – what's realistic?

Managing plugins: Otherwise known as "breaking stuff"

Compatibility: Can you use IE 11's Enterprise and Compatibility modes to render all or most webpages, or do you need separate browsers?

Controlling content: Restrictions on downloads, cut/copy/paste/save/print, email attachments?

Management reality: How many configurations to manage? Can you maintain whitelisting and blacklisting?

Automation: How's your scripting abilities? Can you keep up with the pace of app change? Can you afford not to?

How much for privacy?: Track locations, block ads, trash URL history? This directly affects user experience!

Authentication: Single sign on, pass-through authentication, cached credentials and password tools?

Encryption: Control over certificates, chains, root certs, configuring for end-to-end encryption. Introspection?

Performance considerations: Ad blocking, image refactoring , ICA vs. HTML, video playback, Skype, and launch times

Bumps in the wire: The impact of forward proxies and upstream content filtering – especially third-party networks and services

Issue: Ads are coming from advertising networks where the content provider has no association with the content – who do you trust? Forbes, or the customers of the numerous ad networks that serve content onto the forbes.com page?

Published Browser Security Solutions Overview

Guidance for hardening published browsers

- Publish browsers unique to security zone needs
- Publish an app-specific browser for unique needs
- Publish disposable browsers for personal and social use
- Specifically restrict/allow use of plugins and extensions
- Specifically restrict/allow encryption and certificates
- Control USB and peripheral usage per application
- Control access to filesystems and fileshares
- Perform endpoint inspection and jailbreak detection
- Enforce contextual access policies
- Configure for app-specific logging and reporting
- Control the clipboard and consider one-way clipboard

Browser security policies

This section contains browser-specific policies that should be considered when performing hardening.

Internet Explorer policies

Web Browser Security Revisited

- Restrict process spawning
- MIME handling
- Object caching protection
- Scripted windows security restrictions
- Protection from Zone Elevation
- Information bar
- Restrict ActiveX install prompts
- Restrict active scripting
- Unsigned controls
- Java applets
- Submit non-encrypted form data
- Disallow font downloads
- Restrict file download
- Add-on management (deny/allow/list)
- Security zones
- Disable user persistence
- Display mixed content
- Pop-up blocker
- Prompt for client certificate
- Authenticode settings
- Open files based on content, not file extension
- Disallow active content over restricted protocols to access my computer
- User data persistence
- Enhanced Protected Mode
- ForceASLR
- HTM5 Sandbox Attribute
- Force 64-bit IE

Chrome policies

- Leveraging ADMX templates (windows) (http://dl.google.com/dl/edgedl/chrome/policy/policy_templates.zip)
- Pre-installing Chrome on servers via DSC: https://github.com/PowerShell/xChrome (not recommended for endpoints.)
- Leveraging policies on Linux and Mac (using JSON or Server Admin Tools)
- Chrome for Business vs standard
- Use a web service to help resolve navigation errors (enabled by default)
- Use a prediction service to help complete searches (enabled by default)
- Predict network actions to improve page load performance (enabled by default)
- Send a 'Do not track' request with your browser traffic
- Content settings
- Manage exceptions
- Cookies and site data
- Pop-up blocker
- Block sites from accessing microphone and camera
- Block sites from multiple automatic file download
- Clearing history
- Syncing using your Google ID
- RemoteAccessHostFirewallTraversal, RemoteAccessHostDomain, RemoteAccessHostRequireTwoFactor, RemoteAccessHostRequireCurtain
- DefaultCookiesSetting, DefaultImagesSetting, DefaultPluginsSetting, DefaultPopupsSetting, DefaultGeolocationSetting, DefaultJavaScriptSetting
- CookiesAllowedForUrls, CookiesBlockedForUrls
- ExtensionInstallBlacklist, ExtensionInstallWhitelist, ExtensionInstallForcelist
- SupervisedUsersEnabled
- Chrome password management

Firefox policies

- Pre-Installing Firefox on servers (recommended only for servers and not for endpoints): https://github.com/PowerShell/xFirefox
- Levering the ADM template (https://addons.mozilla.org/en-US/firefox/addon/gpo-for-firefox/). Warning: Last update was 2014 and admins report hit-or-miss results.
- Instant Website ID
- Setting the master password
- Clearing history
- Extensions.getAddons.cache.enabled = false
- Network.prefetch-next = false

Flash Policies

- Flash Player Admin Guide:
  http://www.adobe.com/devnet/flashplayer/articles/flash_player_admin_guide.html

Java policies

- Java Deployment Guide:
  https://docs.oracle.com/javase/8/docs/technotes/guides/deploy/properties.html

Policies for managing multiple browsers on the same endpoint:

- Interaction between Internet Explorer 11 and Edge:
  https://blogs.windows.com/msedgedev/2015/08/26/how-microsoft-edge-and-internet-explorer-11-on-windows-10-work-better-together-in-the-enterprise/

Securing XenApp browser sessions using group policy

Running a browser hosted provides an excellent defense. But, even with execution of the browser running remotely, the endpoint must still be protected from items that can travel across the remoting system. For example, we want the hosted system to have no vision to files via the Citrix Receiver on the endpoint, and by default, the ability to map the end user file system into host system view is enabled. The user would have to explicitly accept the mapping event, but may not understand the implications. For more prescribed security, we want to disable Client Drive Mapping and manage a number of other settings.

Citrix has group policy recommendations which are part of the Common Criteria evaluated configuration guide and many of these are applicable to hosted browsing. The full set appropriate for a given configuration are an item that administrators and security representatives must evaluate for each given set of applications.

To set the policies, first import the ADMX files onto the domain controller. Instructions for this are included in the common criteria download zip, filename

- Evalulated_Configuration_Guide_XenAppXenDesktop_7.6.pdf

Notable items (subset) from the Common Criteria guidance are listed below. Some setting guidance is different than the Common Criteria recommendation and these are noted in the below chart with footnotes.

The following should be considered for hardening browsers published in XenApp:

Policy Name	Settings	Enable or Disable	Importance
Disable Client Reconnect	Auto client reconnect: Prohibited Multi-Stream computer setting: Disabled Session reliability connections: Prohibited Disconnected session timer: Enabled Disconnected session timer interval: 0	Enabled	Low
Disable Windows Media Redirection	Multimedia conferencing: Prohibited Windows Media redirection: Prohibited	Enabled	Medium
Disable Audio over UDP	Audio over UDP: Prohibited	Enabled	Low
Disable Flash Redirection	Flash acceleration: Disabled Flash backwards compatibility: Disabled Flash default behavior: Disable Flash acceleration	Enabled	High
Enable USB Device Redirection	Client USB device redirection: Prohibited	Enabled	Medium
Disable Client Drive Redirection	Auto connect client drives: Disabled Client drive redirection: Prohibited Client fixed drives: Prohibited Client floppy drives: Prohibited Client network drives: Prohibited Client optical drives: Prohibited Client removable drives: Prohibited	Enabled	High
Disable Clipboard Redirection	Client clipboard redirection: Prohibited	Disabled	High. Do not restrict all clipboard, instead configure one-way (below)
Disable Audio Redirection	Audio over UDP real-time transport: Disabled Audio Plug N Play: Prohibited Client audio redirection: Prohibited Client microphone redirection: Prohibited	Enabled	Low
Disable Printer Redirection	Auto-create client printers: Do not auto create client printers Auto-create generic universal printer: Disabled Automatic installation of in-box printer drivers: Disabled Client printer redirection: Prohibited Universal Print Server enabled: Disabled	Enabled	High
Disable Multimedia Redirection	Multimedia conferencing: Prohibited Windows Media redirection: Prohibited	Enabled	Medium
Disable Plug and Play USB Redirection	Client USB Plug and Play device redirection: Prohibited[1]	Enabled	High

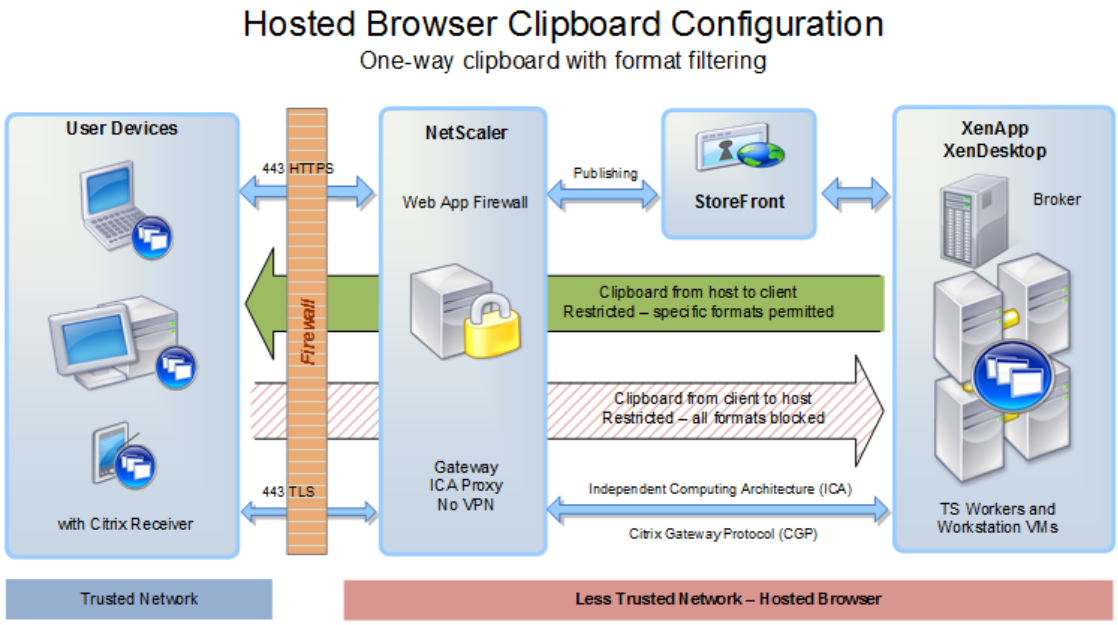
[1] XenApp and XenDesktop 7.6 and beyond permit a more granular control of clipboard function, allowing data to restricted to "inbound" to the client machine and filtering the clipboard data formats to a set of formats approved by the administrator.

Clipboard one-way with format filtering

In the 2009 release of Citrix XenApp 5 Feature Pack 2, Citrix introduced the ability to restrict the clipboard to one-way operation. This "one-way" feature, however, always operated from the perspective of the hosted execution being the trusted space and the endpoint being highly suspicious. The clipboard could be disabled completely, or permitted from the client into the host, but there was no means to permit clipboard functionality from the host into the client while blocking in the other direction. In the case of hosted web browsers, this security model is the reverse of what is desired. The XenApp 5 Feature Pack 2 one-way case also provided no means for restricting clipboard formats to only an administrator defined approved set.

In response to this need, enhanced clipboard filtering was implemented in XenApp and XenDesktop 7.6 (and beyond) to allow clipboard restriction in either direction, each with the ability to specify specific clipboard formats that are allowed to travel the remoting protocol. Policy is controlled server side on the XenApp and XenDesktop servers. While they are trusted "less", they are still items maintained by the administrators who manage the endpoint computers.

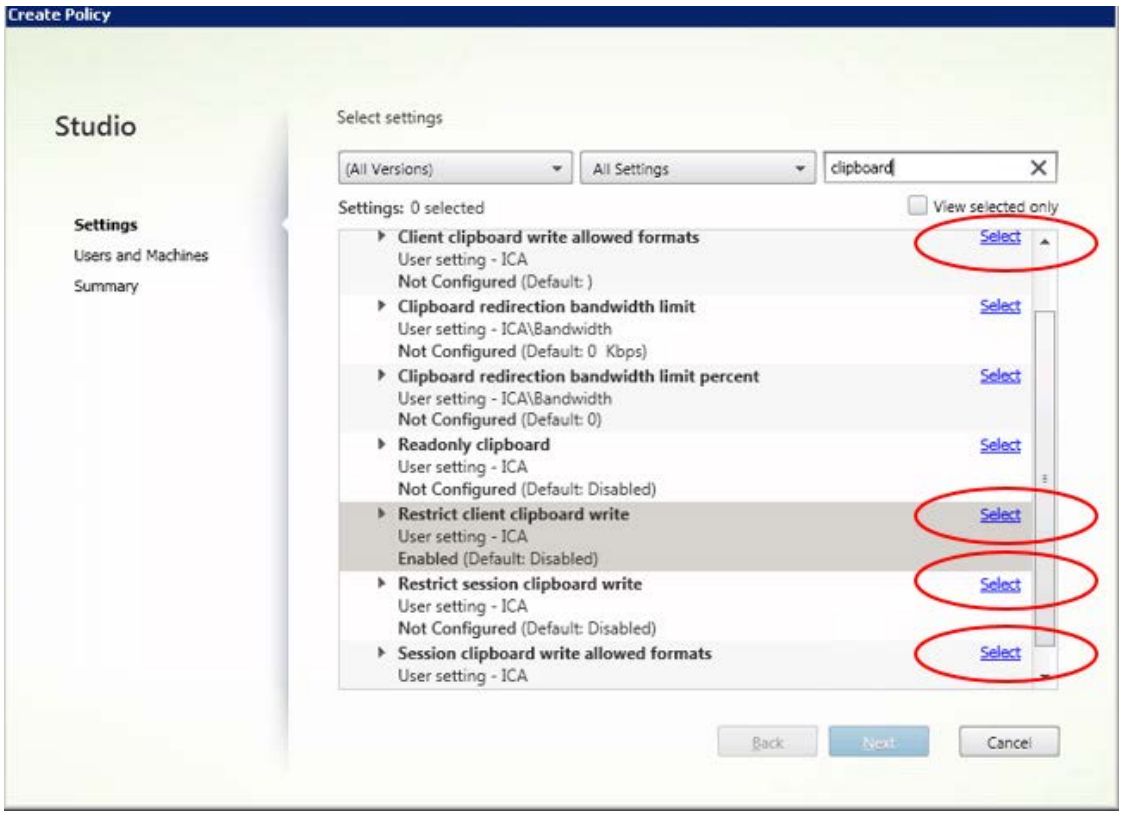
Note that the desired clipboard policies can be assigned to individual apps, enclaved groups of apps, or consistently across all applications, as required.



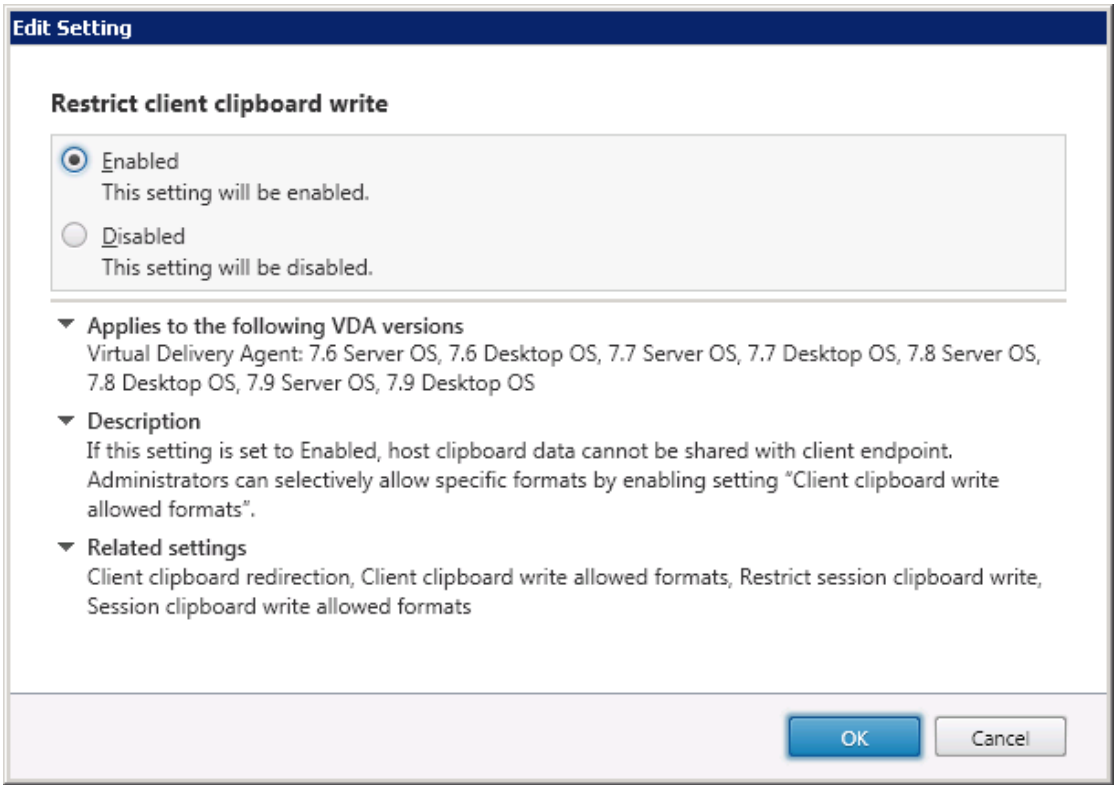
The one-way clipboard model is defined as:

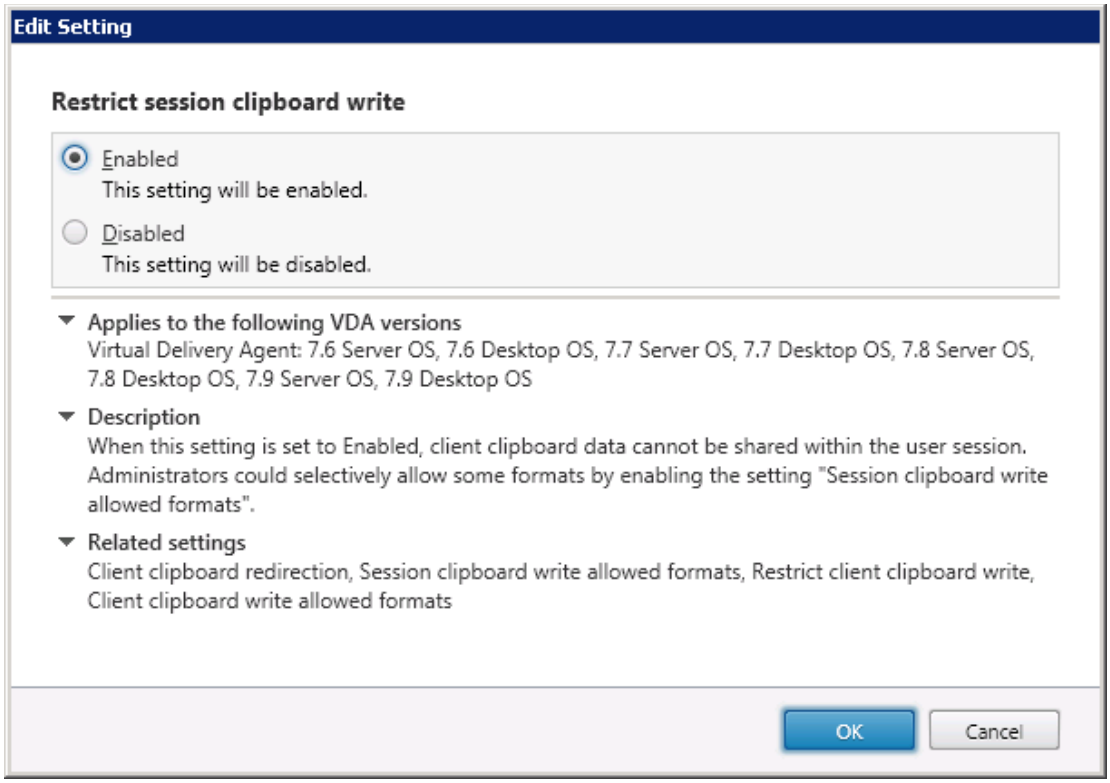
1. Restricting the clipboard redirection (e.g. preventing clipboard redirection)
2. Permitting specific formats for copy and paste

Clipboard policies are configured with Citrix Studio.



Notice that the clipboard activity is "restricted" in both directions, and then permitted for specific formats only inbound to the trusted endpoint.

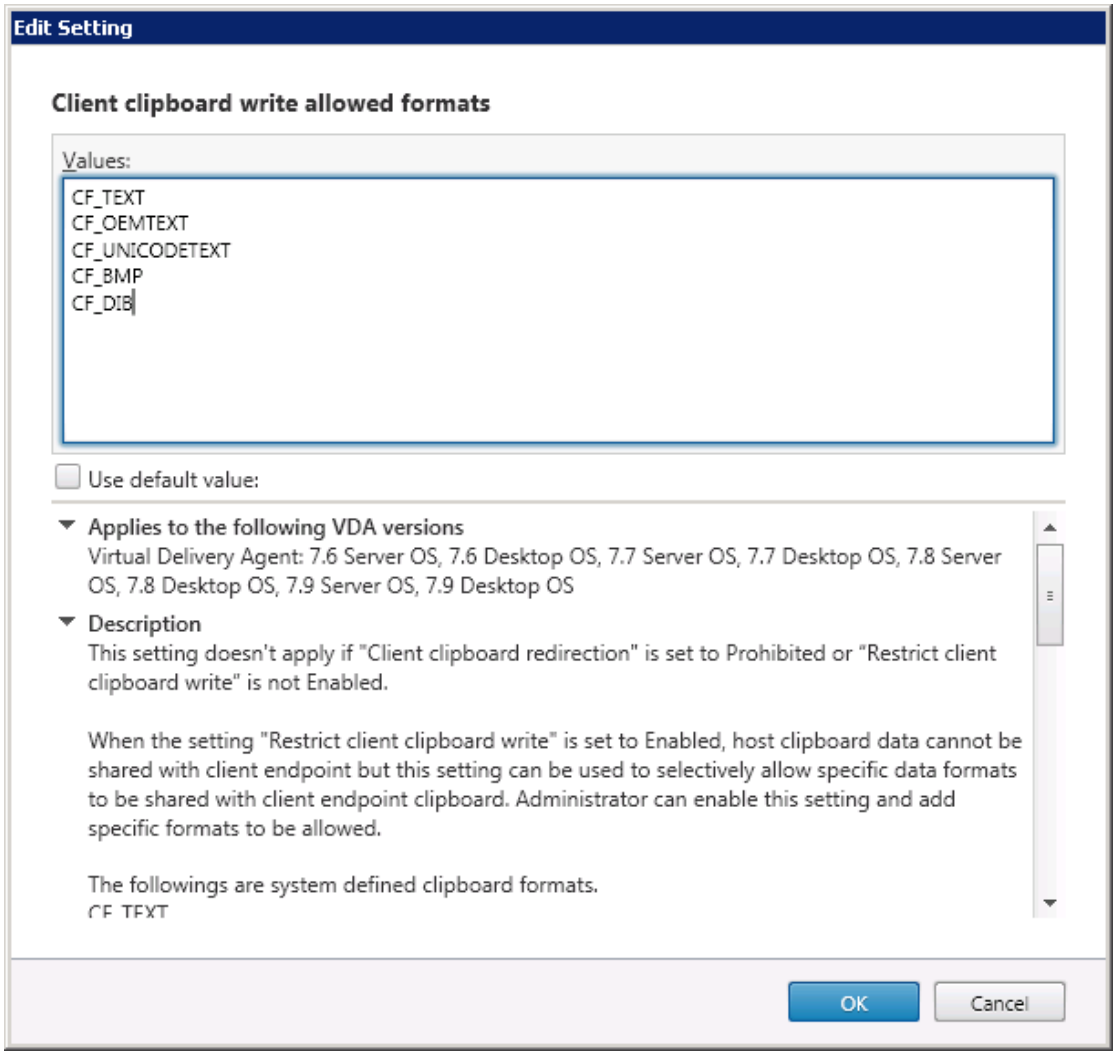




Clipboard formats to permit

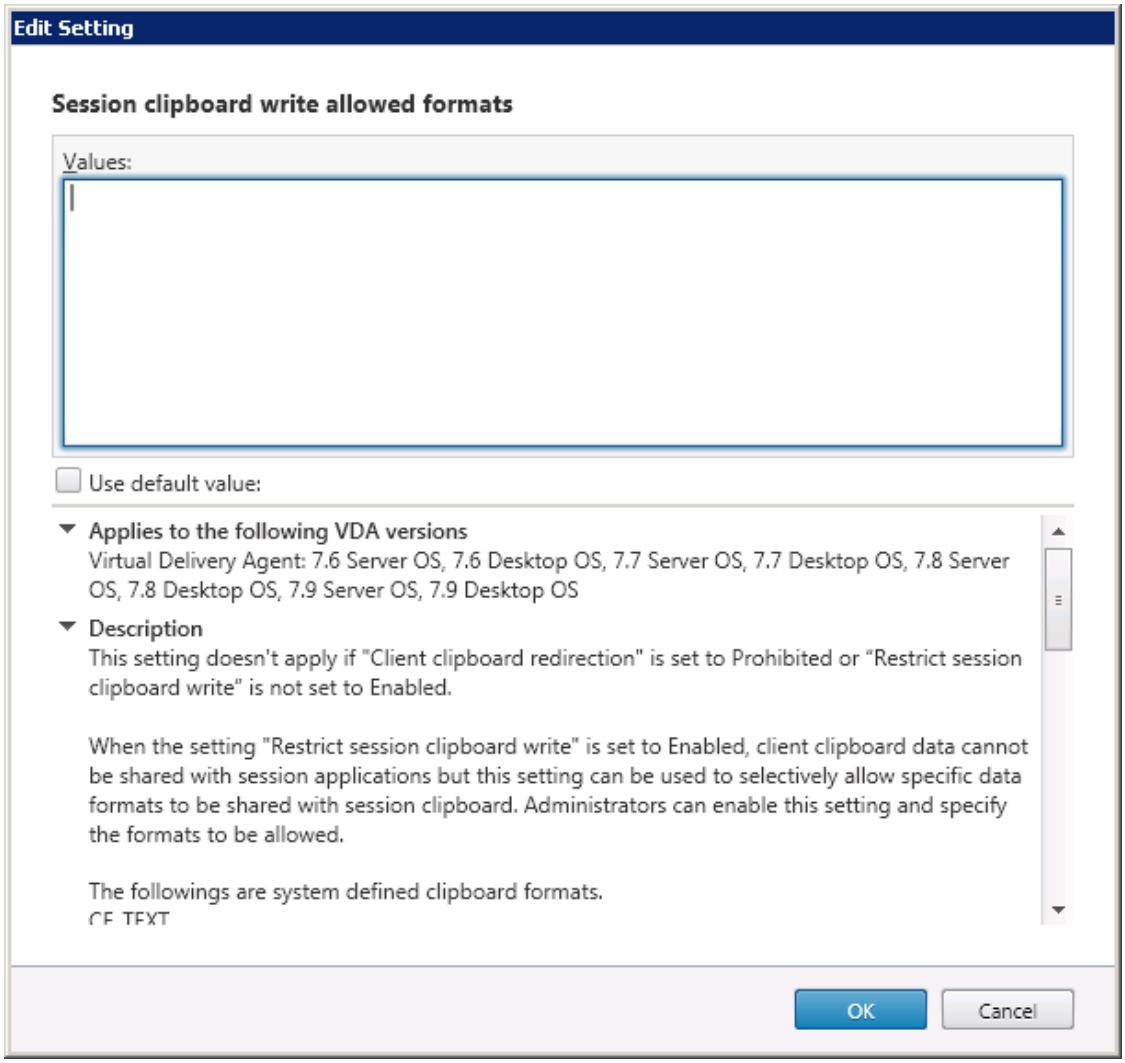
Microsoft defines a large set of standard clipboard formats. The MSDN site currently lists 26 different formats, though many are variations of the same. For text information, CF_TEXT, CF_OEMTEXT and CF_UNICODETEXT are all textual and the operating system will convert from one format placed into the clipboard and a similar format read from the clipboard. Similar examples for images include, CF_BITMAP, CF_DIB, CF_DIBV5 and CF_DIF.

Some administrators indicate that they wish to restrict clipboard to "text and image" and others indicate "text, just text". However, the terms "text" and "image" are not clipboard formats. To provide flexibility to the administrator when restricting clipboard formats, the policy system requires the administrator to name the specific formats that they wish to permit. Formats are identified "by name" using the MSDN standard constants as textual values (e.g. CF_TEXT).



Notice that the clipboard transfer from host to client is restricted, but specific formats are defined by the administrator that they SHOULD be transferred. All data outside the administrator defined set of clipboard formats will not be transferred.

The set of formats defined above is a potential set, you may have specific requirements. Simple clipboard formats (text) should be preferred to more elegant formats.



Notice that no clipboard formats are listed as permitted for transfer from the trusted client to the less trusted session (host).

Program defined clipboard formats

Windows permits applications to name their own clipboard formats. These are assigned a "name" by the application when the custom clipboard format is created. It is possible to define any name as a clipboard content type (e.g. "Foo"). Without clipboard restrictions in place, the Citrix remoting protocol will transfer application-defined clipboard formats. Choose your formats carefully, as rich formats can contain active content, scripts and malware!

In the case of hosted browser execution, the use of custom clipboard formats across ICA remoting would be rare and it is normal for these to not be included in the set of clipboard formats to transfer, though they could be listed. Notice that the ability to support application-named custom formats is the reason that these configuration panels require the admin to enter textual data rather than select from a set of radio buttons. Application-defined clipboard formats and system defined clipboard formats are processed identically – they all have names and if that name is included in the approved set, it will be transferred. System defined names always begin with "CF_", e.g. "CF_TEXT".

Citrix Secure Browser lockdown document

Items recommended for Citrix Secure Browser are similar to those for securing a hosted browser. Here is a link to the Secure Browser publishing document https://docs.citrix.com/content/dam/docs/en-us/workspace-cloud/downloads/Secure%20Browser%20-%20Deployment%20Guide.pdf

Hiding the address bar

Depending on the nature of the web application, it may or may not be appropriate to give the user the ability to select or see the address bar. To remove:

- Google Chrome:
  - By adjusting run parameter to be "--kiosk --no-default-browsercheck --no-first-run URL"
- Microsoft Internet Explorer:
  - By adjusting run parameter to be "–k URL"
  - Perform same task with Group Policy. Enable User or Computer | Policies | Admin Templates | Windows Components | Internet Explorer | Enable Full Screen mode

Often, a hosted browser use will be unconstrained, so the limiting of the address bar will not be needed.

Group Policy Restrictions to consider:

Client disk drives
In a hosted browser scenario, client disks should not be accessible from the host system.

Prevent Client Drive Mapping

Client drive redirection

Allowed
Client files and drives can be mapped, if specified elsewhere

Prohibited
No client files or drives will be mapped

Auto connect client drives: Disabled
Client drive redirection: Prohibited
Client fixed drives: Prohibited
Client floppy drives: Prohibited
Client network drives: Prohibited
Client optical drives: Prohibited
Client removable drives: Prohibited

Client printers
In a hosted browser scenario, client printers should not be accessible from the host system.

Client printer redirection

Allowed
Client printers can be mapped, if specified elsewhere

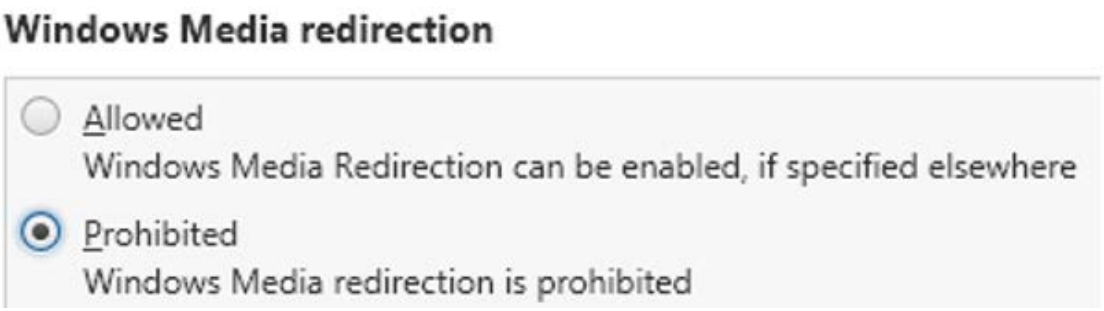
Prohibited
No client printers will be mapped

Auto-create client printers: Do not auto create client printers
Auto-create generic universal printer: Disabled
Automatic installation of in-box printer drivers: Disabled
Client printer redirection: Prohibited
Universal Print Server enabled: Disabled

Windows Media redirection

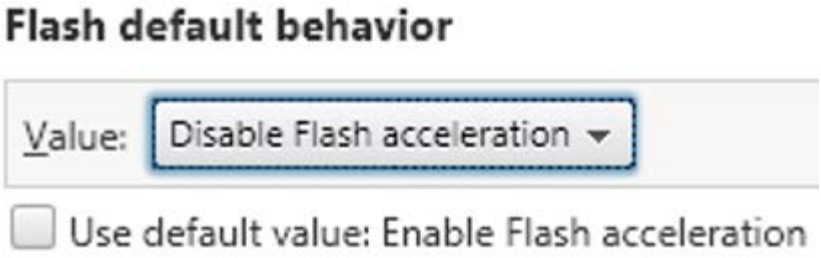
In a hosted browser scenario, remoting of Microsoft defined media types normally should not be enabled. Administrators may have specific cases where remoting of Windows Media is required.

Prohibiting Windows media redirection prevents client side rendering of DirectShow, DirectX Media Objects and Media Foundation.



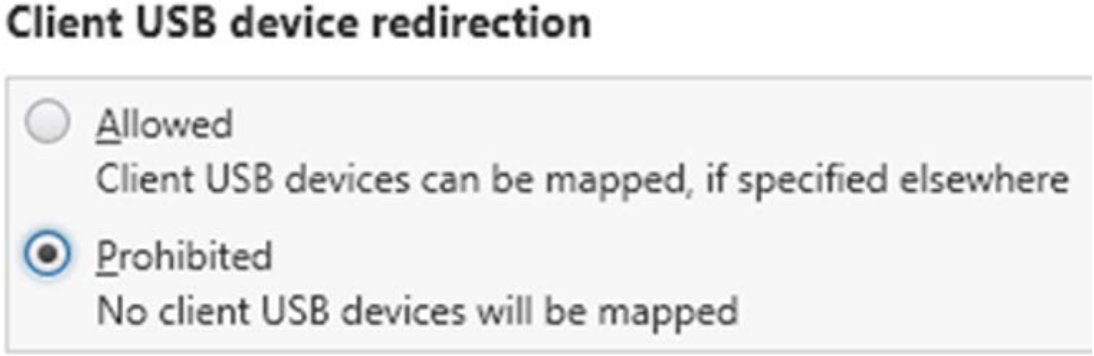
Flash

In a security-focused hosted browser scenario, Flash should be rendered on the server and then remoted to the client as graphics. Enterprise requirements for specific web applications may require flash remoting be enabled and where this is required, the browser should be restricted to access only the specific websites required. Settings exist for Flash remoting to specifically whitelist the sites where flash remoting should be performed.



Client USB device redirection

In a hosted browser scenario, the use of client side USB devices should be very rare. This usage is enabled by the default installation, and should be disabled in a hosted browser configuration.



Hardening the Network to support secured browsing

With control of the web client placement, network based protections can be implemented to increase the security posture via control of the network and the egress point of the browser. The placement of network based security appliances, such as IDS and IPS servers to monitor the network, can be combined with web proxy servers that have integrated AV and security subscriptions enabled. Further restrictions enable only TCP port 443 on the edge firewall, and only allow the proxy server to communicate to the Internet, so even if another application tries to egress traffic, it would have to go through the sanitizing proxy.

DNS is another service that may be restricted at the edge firewall, disallowing TCP/UDP port 53 from reaching the Internet by anything but the corporate recursive DNS server. If enterprises are concerned about hackers leveraging DNS queries to exfiltrate data from the enterprise by querying domains that they own, configure the enterprise DNS server to be the only server that can communicate on to the Internet on port 53. This both reduces DNS-based methods of data exfiltration and simplifies the identification of the data loss associated with an incident. Many DNS servers can be configured with enhanced security capabilities and security subscriptions to blacklist bad IP addresses and names and restrict any recently created domains to protect against fast flux botnet attacks. Included in the security subscriptions are specific IP addresses attributed to known attackers, state-sponsored attack regions, botnets, countries, business, pornography, TOR exit nodes, spammers, malicious agents, and ad networks that may be useful to block.

Proxy servers have matured along with the Internet, and often have browser-specific security capabilities built in. By using a published browser, we can have the browser automatically configured to use corporate proxy servers, instead of using a direct connection from the user's endpoint. There are many security features of proxy servers, including content filtering, authentication, anti-virus scanning, anti-malware scanning, object whitelisting and blacklisting, code analysis and auditing of activities. Proxy servers and other sensors such as intrusion protection systems leverage a network based global threat intelligence, which keeps the protections as current as possible.

Privacy

By providing a published browser, we introduce additional privacy measures for the user. A website may not have the ability to collect personally identifiable information about the user, since the browser may not contain any previous sessions and will be coming from the same system as other users. This has its advantages, as tracking technologies are often outpacing anti-tracking technologies and methods. This additional privacy is a good thing for some users, but many end users trade privacy for the convenience of better search results and a customized experience based on the user.

Users of hosted browsers will no longer be coming from their endpoint IP address, but rather from the published browser which may be in the context of the corporate data center. Many sites customize the experience based on identifying where the user is coming from. This varies from making the initial web site have a better starting point, to blocking or allowing content based on the source address. Some customers have had success by deploying proxy servers or by leveraging proxy services in specific geographic locations, so that simple IP based geolocation services respond appropriately, and provide the user the experience they require.

Education has helped users understand the basics of what not to click on the Internet, but the attacks are becoming more sophisticated, often having no visual cues that the site is being mischievous. Education is extremely important for our users, but many users will be fooled. Users treat their work computer as if it was not their own, and often will follow written policies, so technical controls and containment is suggested to assist in mitigating user-mediated risk.

Information technology and security professionals need to protect our users, while still allowing them to do their work. If security gets in their way, they will often circumvent security and find alternate insecure unsanctioned ways. We need to provide the users a platform that is easy and effective for them to complete their tasks.

Many of us interact with our local browser daily, and we have found ways to protect ourselves. Many of us use pop up blockers built in to our browsers, we add plugins to increase our security, such as ad blockers, SSL redirection tools, and secure single sign on tools. We avoid installing insecure add-ons and try to avoid unknown software from running on our computers. We are smart about what we click, and we stay on the 'safe' places on the Internet. When a site doesn't load properly, we may have a second or third browser installed on our local computer, so that when the first one didn't work with a particular site, we have an alternative browser that may work. We have a preference of browser application and we like that the browser remembers our history and can maintain our bookmarks. Providing the training required to accomplish each of these tasks to all users to maintain their own local browser is not possible at scale without technical controls, or additional technologies

Anytime you restrict a web component or technology, you have the chance to break a website for users, which would not allow them to accomplish their work tasks. If the restriction is required by the business, and it is a high risk component, some method of containment or control could be leveraged to mitigate the risk. It is important to determine the business requirements for a user's workflow. If a user needs a specific version of a component, such as JavaScript, Java Plugin, ActiveX control, Silverlight, Flash, or PDF viewer, that component can be enabled in a controlled published browser, that is under the watchful eye of the information technology team. This allows a reduction of the attack surface of the endpoint, not requiring what might be considered an insecure component to be deployed on a managed or unmanaged endpoint. Tools such as Citrix AppDNA can spider a website and determine some website dependencies, which may be helpful in identifying user platform requirements.

Controlling Browser Placement

When you run a standard browser from your personal computer, the browser is running from the context of that computer. If you are at home and you need access to a web component that is on the company's internal network, the browser must be able to interact with the internal network through a full or partial VPN - which may expose more than required. An alternative would be to place a published browser in the proper location to have sufficient access to the web application. This is similar to a traditional XenApp deployment, where running the client portion of the application close to the data portion of the application would have been instituted primarily for performance reasons.

Since the published browser can run from anywhere XenApp virtual instance is placed, we now have control of the placement of the source of web client communication. We now have to ask ourselves, where is the best place to run the web client from? Possible options include from an external cloud provider, demilitarized zone, internal corporate network or a secure enclave within the corporate network. By placing the web client within an external cloud provider, it may be closer to the web application, but if the browser was compromised in any way, the attacker would not be within the walls of the enterprise network.

The secure enclave idea has become quite popular recently as a secured area of the network that has a very restrictive firewall. Optional proxy and DNS servers that are security aware and have security subscriptions provide live threat intelligence to mitigate attacks. The firewall would only allow the proxy and DNS servers to communicate outbound, and could be used to clean and monitor the traffic as required. Users would enter this secure environment through the use of a NetScaler, providing for multifactor authentication and encrypted communication. Optionally session recording could be enabled, satisfying additional audit requirements.

For a long time, Citrix XenApp has been utilized to move the client and data components closer to each other for increased performance due to serial communication. Some AJAX web applications developed and tested under ideal network conditions are extremely chatty, and when running the web browser on the endpoint over a slow or degraded connection, the application does not perform well, due to the chatty nature of the application. The average website is over 2 megabytes currently, and it is steadily increasing. By providing a view of the rendered website, users on satellite connections on oil rigs, cruise ships or even space stations can be provided a usable experience: one that that would not be possible with a locally installed browser.

Managing the browser on the endpoint is challenging for enterprise IT – especially when the device is employee owned and not domain joined. By moving the web browser to be under the control of corporate IT within the datacenter, there are a variety of controls and polices that can be enforced on the published browser that benefit security and compliance.

Resources and references

- OWASP Top 10 (https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project)
- Securing Your Virtual Desktop Environment whitepaper (https://support.citrix.com/servlet/KbServlet/download/32029-102-686963/)
- Citrix Ready solutions https/www.citrix.com/ready
- Web Application Security Consortium www.webappsec.org
- Synergy 2016 video – Securing the Published Browser (https://www.youtube.com/watch?v=zy3sb_0m8jI)
- What most IT admins don't know about FireFox management whitepaper (http://tinyurl.com/jxb575m)
- What most IT admins don't know about Internet Explorer management whitepaper (http://tinyurl.com/o49aobk)
- Citrix Common Criteria documentation (https://www.citrix.com/about/legal/security-compliance/common-criteria.html)

Acknowledgements

The primary authors of this document are Eric Beiers, Joseph Nord and Kurt Roemer. We would also like to acknowledge Steven Krueger, Chris Mayers, Jeremy Moskowitz and Martin Zugec for their valued contributions.

Learn more

- www.citrix.com/secure
- more..citrix.com/secure-browser
- www.citrix.com/xendesktop

About Citrix

Citrix (NASDAQ:CTXS) aims to power a world where people, organizations and things are securely connected and accessible to make the extraordinary possible. Its technology makes the world's apps and data secure and easy to access, empowering people to work anywhere and at any time. Citrix provides a complete and integrated portfolio of Workspace-as-a-Service, application delivery, virtualization, mobility, network delivery and file sharing solutions that enables IT to ensure critical systems are securely available to users via the cloud or on-premises and across any device or platform. With annual revenue in 2015 of $3.28 billion, Citrix solutions are in use by more than 400,000 organizations and over 100 million users globally. Learn more at www.citrix.com.