
**UNIVERSIDAD DON BOSCO**
**VICERRECTORÍA DE ESTUDIOS DE POSTGRADO**

**TRABAJO DE GRADUACIÓN**

# Endurecimiento (Hardening) de navegadores web más utilizados. Caso práctico: Implementación de navegadores endurecidos (Microsoft Internet Explorer, Mozilla Firefox y Google Chrome) en un paquete integrado para Microsoft Windows.

**PARA OPTAR AL GRADO DE:**
**MAESTRO EN SEGURIDAD Y GESTION DEL RIESGO INFORMATICO**

**ASESOR:**
**Mg. JOSÉ MAURICIO FLORES AVILÉS**

**PRESENTADO POR:**
**ERICK ALFREDO FLORES AGUILAR**

**Antiguo Cuscatlán, La Libertad, El Salvador, Centroamérica**

**Febrero de 2015**

**AGRADECIMIENTOS**

A Dios Todopoderoso, por regalarme vida, salud y determinación para alcanzar un objetivo más en mi vida.

A mi amada esposa, que me acompaño durante toda mi carrera, apoyó y comprendió mi dedicación de tiempo y esfuerzo a este proyecto y nunca dudó que lo concluiría con bien.

A mis padres y hermana, que siempre han sido mis pilares y me enseñaron que lo mejor que te pueden regalar en la vida es una buena educación.

A mis compañeros de trabajo, que mediante su esfuerzo extraordinario me han permitido contar con el tiempo necesario para dedicar mucho más tiempo a la consecución de esta meta.

A mis amigos de los que siempre he tenido una palabra de aliento cuando la he necesitado.

A mi supervisor y compañeros de la Escuela de Computación de la Universidad de Queens por facilitarme espacio, recursos, tiempo e información valiosa para la elaboración de este trabajo.

A mi asesor de tesis, al director del programa de maestría y mis compañeros de la carrera, que durante estos dos años me han ayudado a lograr esta meta tan importante.

Erick Alfredo Flores Aguilar

**INDICE**

**RESUMEN**

Una herramienta fundamental para desarrollar actividades en línea hoy en día es el Navegador web, independientemente de cuál se utilice. Por tanto, esta herramienta se convierten en un punto susceptible para la seguridad y privacidad, tanto por las vulnerabilidades inherentes como por los agregados de software que pueda incluir y que a su vez son vulnerables. El presente trabajo trata sobre la evolución de los navegadores más utilizados, sus vulnerabilidades, técnicas de mitigación y el desarrollo de un paquete integrado para Microsoft Windows (7 y 8) que incluye 3 navegadores (Explorer, Firefox y Chrome), plugins y extensiones seguras preinstaladas y características de seguridad y privacidad pre configuradas.

*Índice de términos – Internet, Navegadores, Privacidad de datos, Seguridad informática*

## I. INTRODUCCIÓN

En la actualidad, un usuario de un equipo de cómputo pasa la mayoría del tiempo de utilización trabajando con un navegador web, ya sea para simplemente buscar información, comunicarse con otros usuarios, realizar transacciones que involucran dinero, realizar el trabajo de la oficina a través de una aplicación web, entre otras. Estas actividades solo son algunas entre una amplia variedad que pueden ser realizadas en un entorno web.

Al considerar lo anterior, es obvio llegar la conclusión que el navegador web es una de las aplicaciones más utilizadas y a su vez más importantes de las que se encuentran instaladas en los sistemas de los usuarios, y por tanto, el usuario debería tomarse el tiempo para asegurarse que se utiliza un navegador funcional, actualizado, y sobre todo seguro.

Para lograr que el usuario tenga una mejor experiencia con el o los navegadores web de su preferencia, se propone en este trabajo primero recopilar y presentar información importante acerca de los tres navegadores web más utilizados a nivel global y en El Salvador (Microsoft Internet Explorer desde su versión 8 con 22.07% y 12.8%, Mozilla Firefox con 19.09% y 23.92%, y Google Chrome con 50.18% y 59.14% respectivamente). Los porcentajes reflejan las estadísticas obtenidas entre agosto y octubre de 2014 en el servicio de análisis web StatCounter, que ofrece servicios a individuos o instituciones que desean conocer los detalles de visitas a sus sitios, permitiendo registrar los datos de visitas entre 0 y 250,000 de manera gratuita, y con un pago recurrente hasta 60 millones de visitas mensuales. La información recopilada además incluye la evolución de dichos navegadores, las vulnerabilidades principales de cada uno y los ataques a los que los usuarios se pueden ver expuestos por el uso de estas herramientas, y los métodos de mitigación de riesgo aplicables a estos navegadores.

Una vez completada la porción documental, se procederá a generar un paquete de software compatible con Microsoft Windows 7 y 8, que incorpore versiones actualizadas de los tres navegadores que remplacen a las versiones ya instaladas en el equipo al que se aplique, y que posiblemente estén comprometidas por plugins o extensiones maliciosas, o simplemente han quedado desfasadas. Los navegadores instalados además contarán con plugins y extensiones que mejoren la experiencia del usuario y configuraciones que mejoren su seguridad y privacidad. La aplicación a entregar será generada de manera que pueda ser modificada y crear versiones subsecuentes de la misma, que aseguren la instalación de los navegadores más actualizados al menos dos veces al año, mientras estos no cambien radicalmente su funcionamiento.

La aplicación podrá ser utilizada por usuarios en general sin restricciones, tanto en equipos institucionales como en computadoras propias y personales, y además podrá ser fácilmente descargada de repositorios de acceso público y gratuito.

## II. MARCO TEORICO

### 2.1 Navegadores Web: Evolución y situación actual

#### 2.1.1 Arquitectura de un navegador web

Antes de describir la funcionalidad y aspectos de seguridad y privacidad de los navegadores web, es conveniente conocer cómo funcionan internamente, ya que así, cuando se trabaje documenten las vulnerabilidades, ataques y métodos de mitigación existentes, se comprenda a que componente o subsistema del navegador podríamos estar afectando y su interrelación con otros elementos del aplicativo.

La mayoría de navegadores web modernos [1][2], y entre ellos los que constituyen el objeto principal de estudio de este trabajo, comprenden ocho subsistemas principales, más las dependencias entre estos subsistemas, así:

a) Interface de usuario: constituye la capa entre el usuario y el motor de navegación. Provee características tales como barras de herramientas, manejo de descargas, muestra del progreso de carga de una página y capacidades de impresión. Provee además interacción con el sistema operativo y con otras aplicaciones.

b) Motor de navegación: es un componente integrable que provee una interface de alto nivel al motor de renderizado. Soporta las acciones primitivas de navegación tales como “adelante”, “atrás” y “actualizar”

c) Motor de renderizado: se encarga de producir una representación visual para un identificador uniforme de recursos (URI). Es capaz de desplegar contenido HTML y XML, inclusive estilizados con CSS. Además es responsable de calcular la disposición exacta de la página y utiliza algoritmos para ajustar incrementalmente la posición de los elementos desplegados.

d) Subsistema de red: implementa los protocolos de transferencia de archivos tales como HTTP y FTP. Además puede implementar funciones de caché de elementos y recursos recuperados recientemente.

e) Intérprete de JavaScript: evalúa el código escrito en JavaScript incluido en la página.

f) Analizador de XML: analiza los documentos XML y los transforma en un árbol DOM (Document Object Mode).

g) Subsistema de despliegue: provee las funciones primitivas de manejo de ventanas, widgets y fuentes del sistema.

h) Subsistema de datos persistentes: almacena datos asociados a la sesión de navegación. Pueden ser datos de alto nivel como favoritos, o de bajo nivel como cookies, certificados de seguridad o elementos de caché.

En la figura 1, pueden apreciarse los componentes enumerados que forman parte de la arquitectura genérica de los navegadores generalmente aceptada y sus interrelaciones y dependencias.

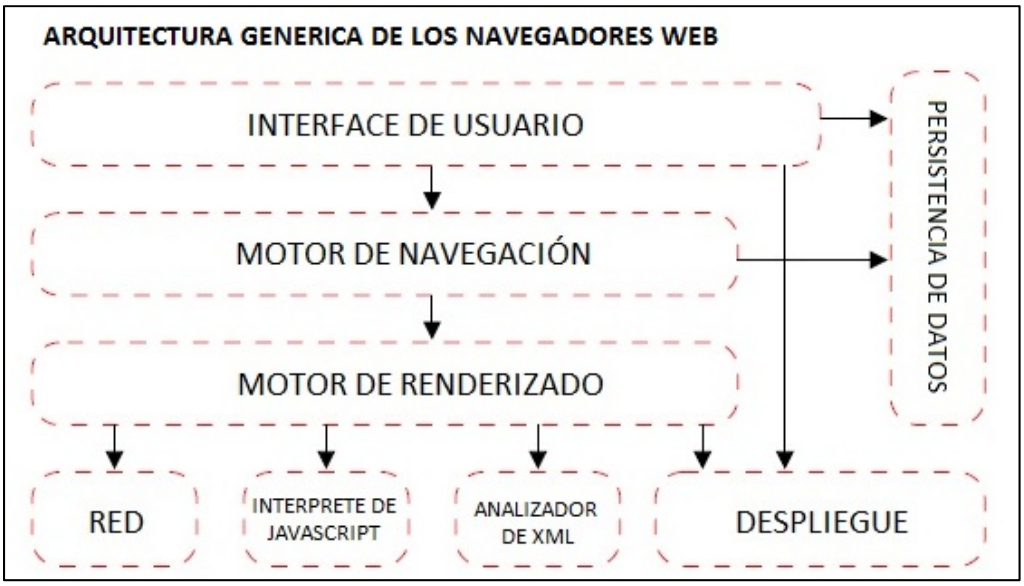


Fig. 1. Arquitectura genérica de los navegadores web[1]

### 2.1.2 Microsoft Internet Explorer[1]

Microsoft Internet Explorer, conocido comúnmente como IE, es un navegador web que fue desarrollado por Microsoft para incluirlo en su sistema operativo Windows 95, y ha sido uno de los navegadores más utilizados desde entonces, con un máximo de utilización que se calculó aproximadamente en 95% en el período entre 2002 y 2003. Esta cuota de mercado ha ido disminuyendo de forma gradual debido a la competencia de otros navegadores.

El proyecto de Internet Explorer dio inicio en 1994, liderado por Thomas Reardon y por Benjamin Slivka, que se basaron en el código fuente de Spyglass, uno de los primeros navegadores web disponibles. Paulatinamente dicho código fuente dejó de utilizarse en versiones posteriores a la 1.0

**Versiones anteriores**

- IE 1.0: aparece en agosto de 1995 y se ofrece como un complemento incluido en Microsoft Plus! para Windows 95

- IE 2.0: liberado en noviembre de 1995, compatible con Windows NT 3.5 y 4.0. Incluye soporte para SSL, cookies, y RSA. Primera versión multiplataforma disponible para Macintosh System 7

[1] Descarga o actualización de IE en: http://windows.microsoft.com/es-es/internet-explorer/download-ie

- IE 3.0: lanzado en agosto de 1996. Primera versión sin el código fuente Spyglass. Incluye soporte para controles ActiveX, y applets de Java. Era posible mantener versiones anteriores del navegador al instalar la 3.0

- IE 4.0: disponible en 1997, se enfocó en la integración con el sistema operativo y aplicaciones como Outlook Express, Microsoft Chat y NetMeeting, además de proveer cifrado de 128 bits.

- IE 5.0: disponible desde 1999, sufrió cambios como el paso de los canales a los favoritos, y además fue la última versión disponible de manera formal para Mac OS.

- IE 6.0: lanzado en agosto de 2001, poco después de Windows XP. Incluye mejoras en DHTML, CSS y el motor MSXML. Se mejora el aspecto de seguridad con IE 6.0 SP1 y subsiguientes service packs.

- IE 7.0: disponible desde octubre de 2006, incluye navegación en pestañas y un filtro de suplantación de identidad. Se elimina la integración con el escritorio de Windows. Compatible con Windows XP SP2 y superiores, Windows Vista y Windows Server 2003 SP1 y superiores.

- IE 8.0: lanzado en marzo de 2009, incluye soporte para plataformas Windows de 32 y 64 bits. Presenta mejoras en manejo de contenidos RSS, CSS y AJAX. Aprueba el test Acid2 e incluye de forma invocable el motor de renderizado de IE7. Incluye mejoras en el soporte de JavaScript y refuerza las características de privacidad y seguridad con InPrivate y SmartScreen.

- IE 9.0: lanzado en marzo de 2011, obteniendo un puntaje de 95/100 en Acid3, con soporte casi completo para CSS3, soporta HTML5, y cuenta con la capacidad de utilizar aceleración por hardware para el renderizado de páginas usando Direct2D y DirectWrite.

- IE 10.0: disponible también desde abril 2011 (en una versión platform preview). Se aceleró su aparición debido a los constantes cambios dentro de lo que se denominó la “nueva guerra de los navegadores” y se incluyeron nuevas mejoras al soporte de CSS3 y HTML5.

**Versión actualizada estable:**

Internet Explorer 11[3] constituye la última versión estable disponible al momento de escribir este trabajo. Lanzada en octubre de 2013 para Windows 8.1 y en noviembre del mismo año para Windows 7.

**Características interesantes y novedades en IE 11.0:**

- Pestañas: se soportan 100 pestañas por sesión en contraste con la versión anterior que soportaba originalmente 10 por sesión).

- Navegación lado a lado: soportada en la experiencia de IE en modo aplicación de Windows 8, al abrir un enlace en una nueva ventana, esta se coloca al lado de la original, usando cada una la mitad de la pantalla.

- Adobe Flash incluido como una característica de la plataforma y actualizado automáticamente a través de Windows Update.

- DTN (Dot Not Track): habilitado como parte de la configuración express. Esta característica envía una señal DTN=1 a cada página que se visita, permitiendo así a los usuarios decidir si su actividad de navegación puede ser registrada y utilizada para fines publicitarios por ejemplo.

- Modo protegido mejorado: Provee seguridad adicional comparada con versiones anteriores de esta característica introducidas en IE7. Refuerza y agrega restricciones adicionales a las capacidades del navegador, previniendo muchos de los escenarios comunes que pueden ser aprovechados por hackers y además limitando la información que el mismo navegador puede proveer a sitios no confiables.

- Mejoras al soporte de CSS3 y HTML5 tales como Web Workers, WebSockets, Gráficos Vectoriales Escalables (SVG), ejecución asincrónica de scripts, y varias APIs como AppCache y Drag-and-Drop.

- Soporte para la Librería de Gráficos Web (WebGL) para el despliegue de animaciones y gráficos 3D directamente en el navegador.

- Mejoras en la barra de herramientas del desarrollador, siempre desplegable con la tecla F12.

- Utiliza la versión más reciente del motor de renderizado Trident 7.0 que básicamente es el responsable de las mejoras de compatibilidad y desempeño del navegador.

- Es importante mencionar que en Windows 8, no puede actualizarse de manera independiente el navegador a la versión 11, sino que debe aplicarse la actualización a Windows 8.1 para disponer del navegador actualizado.

### 2.1.3 Mozilla Firefox[2]

Mozilla Firefox es un navegador web libre, gratuito y de código abierto, desarrollado desde sus inicios como una aplicación multiplataforma, con compatibilidad para los sistemas Microsoft Windows, Mac OS X y GNU/Linux.

Este esfuerzo ha sido coordinado por la Corporación y la Fundación Mozilla, y la idea detrás de este navegador ha sido implementar los estándares actuales en la medida de lo posible para mantener compatibilidad y mostrar de manera eficiente y exacta el contenido web, y por supuesto, adoptar futuros estándares cuando sea necesario.

[2] Descargas y actualización de Firefox en: https://www.mozilla.org/es-MX/firefox/new/

**Orígenes**:

Firefox [4] tuvo sus inicios en un experimento del Proyecto Mozilla liderado por Dave Hyatt, Joe Hewitt y Blake Ross. La idea se centraba en los problemas causados por un lado por el modelo de apoyo por parte de Netscape, y por otro, en lo que denominaban una suite demasiado saturada (Mozilla Application Suite). En abril 2003 se creó un navegador que fuera independiente de la suite, y se centraron los esfuerzos del grupo en dicho navegador y en el componente de correo Mozilla Thunderbird.

**Versiones anteriores:**

- Phoenix: nombre anterior con el que inicio el proyecto. Ya incluía la capacidad de personalizar la barra de navegación, el uso de tabuladores o pestañas, un gestor de complementos y múltiples páginas de inicio (cubre las versiones 0.1 a 0.5 en el año 2002)
- Firebird: con este nombre se tuvieron problemas con la base de datos homónima. Mejoras en los marcadores, en opciones de privacidad y la inclusión de un gestor de contraseñas (cubre las versiones 0.6 a 0.7 en el año 2003)
- Firefox original: se crean los instaladores formales para Windows y Linux (cubre las versiones 0.8 a 0.10.1 en el año 2004)
- Firefox 1: se agrega soporte para SVG y Canvas, y se mejora la compatibilidad con JavaScript 1.5 y CSS 2/3 (cubre las versiones 1.0 a 1.5 entre 2004 y 2007)
- Firefox 2: primera versión con protección anti-phishing. Botones independientes para cada pestaña y soporte para JavaScript 1.7 (cubre las versiones 2.0a a 2.0.0.20 entre 2006 y 2008)
- Firefox 3: nueva interface de usuario para FTP. Alcance cumplimiento de test Acid2. Nuevo motor de JavaScript Tracemonkey, se agrega soporte nativo para JSON. Se incluye el modo de navegación privada (cubre las versiones 3.0 a 3.6 entre 2006 y 2012)
- Firefox 4: se lanzan versiones para 64 bits (Linux/Mac). Se incorpora característica Do Not Track (DTN). Se corrigen más de 7000 errores desde la primera beta (cubre las versiones 4.0b1 a 4.0.1 entre 2010 y 2011)
- Firefox 5: se estandariza la versión del motor de renderizado (Gecko) a la misma versión del navegador (5.0) (cubre las versiones 5.0a2 a 5.0.1 en el año 2011)
- Firefox 6 en adelante: se inicia el esquema de actualizaciones Nightly, Aurora, Beta y luego publicación oficial.
- Firefox 7: mejora radical en el uso de la memoria
- Firefox 8: carga más rápida (20%) que Firefox 5. Se eliminan pestañas animadas.
- Firefox 9: mejora en soporte para HTML5 y CSS

- Firefox 10: solución a problema de alto riesgo de hacking mediante imágenes
- Firefox 11: se incluyen herramientas para desarrolladores Page Inspector y Style Editor
- Firefox 15: pruebas de nueva interface de usuario Australis
- Firefox 18: se añade compatibilidad con Windows 8 (año 2012)
- Firefox 20: se incluye soporte para códec H.264
- Firefox 32: cierra el ciclo de desarrollo el 14 de octubre de 2014.

**Versión actualizada estable:**

La nueva versión de Mozilla Firefox ya liberada (14 de octubre de 2014) es la 33.0 que termina su ciclo de actualización el 25 de noviembre de este mismo año. La versión siguiente (34.0) continúa con el esquema Nightly, Aurora, Beta, Publicación oficial.

**Características interesantes y novedades en Firefox 33:**

- Interface: Australis mejorada y con amplias capacidades de personalización
- Navegación privada, georreferenciación y aceleración vía GPU mejoradas.
- Acceso a miles de complementos compatibles y seguros en addons.mozilla.org
- Herramientas para desarrolladores clásicas como la Consola de errores, Scratchpad para probar código JavaScript, Editor HTML, Inspector DOM
- Soporte para OpenH.264
- Sincronización mejorada a través de Firefox Sync
- Cumplimiento 100/100 en test Acid3
- Soporte optimizado para el uso de SSL/TLS, anti-phishing, antimalware e integración con una variedad de antivirus gratuitos y comerciales.
- Mejora en la seguridad con la capacidad de conexión a un proxy HTTP utilizando HTTPS
- Utiliza la última versión de motor de renderizado Gecko alineado a la versión de Firefox (33.0)
- Inclusión (Nightly) de WebIDE para pruebas de compatibilidad y funcionalidad de código en Firefox OS

### 2.1.4 Google Chrome[3]

Google Chrome [5] es un navegador web propiedad de Google Inc. y desarrollado en base a diversos frameworks de código abierto tales como el motor de renderizado Blink (derivado a su vez de WebKit).

Chrome fue liberado en su primera versión (beta) en septiembre de 2008, lanzándose una versión estable en diciembre del mismo año, y accesible al público en general. Chrome fue desarrollado para ser un navegador multiplataforma y actualmente está disponible para Microsoft Windows, Mac OS X, múltiples distribuciones de GNU/Linux, Chrome OS y además para los sistemas operativos móviles Android e iOS.

**Versiones anteriores:**

- Versiones 0.2 a 0.4: versiones beta iniciales. Se incluyen las características básicas del navegador como las pestañas, administración de marcadores, corrección ortográfica y bloqueo de ventanas emergentes. Desde el inicio estuvo basado en el motor de renderizado WebKit (entre septiembre y noviembre de 2008)

- Versión 1: primera versión estable y de uso generalizado (diciembre de 2008)

- Versión 2: mejora de un 35% de rendimiento ejecutando JavaScript. Se agrega el modo de pantalla completa, el auto llenado de formularios y el soporte básico para GreaseMonkey (mayo de 2009)

- Versión 3: mejora de un 25% en ejecución de JavaScript. Se agrega soporte para video y audio con HTML5 (octubre de 2009)

- Versión 4: se agrega el soporte para extensiones, y se mejora la compatibilidad con HTML5. Esta versión logra pasar completamente el test Acid3 (enero de 2010)

- Versión 5: se agrega la capacidad de sincronización de preferencias vías cuentas de Google. Se integra la funcionalidad de Adobe Flash Player con el navegador (mayo de 2010)

- Versión 8: se inaugura la Chrome Web Store. El visor de PDFs se ejecuta en sandbox para mejorar la seguridad. Se implementa “about:flags” para poder modificar los valores de una gran variedad de opciones de Chrome (diciembre de 2010)

- Versión 14: se incluye la validación de sitios HTTPS (septiembre de 2011)

- Versiones 22 y 23: se agrega soporte para TLS 1.1. Además se hace disponible la preferencia DTN (Dot Not Track) al usuario (noviembre 2012)

- Versión 28: se reemplaza el motor de renderizado WebKit por Blink (julio de 2013)

- Versión 32: se integran indicadores a nivel de pestaña para audio y webcam (mejora la privacidad). Se mejora la interface en modo Windows 8 Metro (enero 2014)

[3] Descarga de Google Chrome en: http://www.google.com/chrome/

- Versión 37: como en la mayoría de las versiones se añaden mejoras en el desempeño y estabilidad, y se corrigen múltiples errores. Se añade soporte a sistemas Windows de 64 bits (agosto de 2014).

**Versión actualizada estable**:

La última versión para sistemas desktop que se considera estable y ha sido lanzada de forma generalizada es Google Chrome versión 38, y se encuentran en desarrollo y estado beta las versiones 39 y 40.

**Características interesantes y novedades en Chrome 38:**

- Soporte a varias modalidades de autenticación de dos factores, incluyendo uso de smartcards y llaves USB.
- Consola de JavaScript y Herramientas de desarrollador
- Aislamiento de procesos (sandboxing)
- Modo incógnito de navegación
- Flash player y visor de PDFs integrado
- Integración muy cercana a aplicaciones de Google (Gmail, Drive, Docs, Traductor, Plus, Calendario, Mapas)
- Soporte completo a CSS 2/3, HTML5, WebGL, WebM, sin comprometer el rendimiento de la aplicación.
- Native Client, para poder ejecutar directamente código C y C++

### 2.1.5 Otros navegadores

**Opera**[4]

Opera es un navegador web propiedad de Opera Software en Noruega, que lanzo una primera versión estable y de uso general en 1996. Utiliza el motor de renderizado Blink y es multiplataforma incluyendo soporte para Microsoft Windows, Mac OS X y GNU/Linux (para Linux el motor de renderizado es Presto).

La última y actualizada versión estable para Windows es la versión 25 (liberada en octubre de 2014)

[4] Descarga de Opera en: http://www.opera.com/es

**Características principales:**

- Administrador de contraseñas y administrador de descargas
- Administrador de tareas que permite cerrar independientemente cada proceso
- Navegación por pestañas con capacidades de hibernación
- Administración de marcadores
- Bloqueo de ventanas emergentes
- Extensiones (algunas diseñadas para Chrome pueden ser compatibles)
- Navegación privada
- Protección anti fraude, detección de sitios inseguros y phishing

**Safari**[5]

Safari es el navegador web propietario desarrollado por Apple Inc. pensado para sus sistemas OS X e iOS. Originalmente también estaba disponible para Microsoft Windows pero el soporte fue removido en el 2012.

Originalmente fue liberado en su versión 1.0 en junio de 2003. Fue hasta la versión 3 (en octubre de 2007) en que se implementó la compatibilidad con Windows.

La última versión para Windows liberada y estable es la 5.1.7 (de mayo de 2012) y utiliza el motor de renderizado WebKit.

**Características principales:**

- Cumplimiento 100/100 con test Acid3
- Navegación por pestañas
- Inspectores CSS, DOM y JavaScript
- Soporte completo para HTML5
- Extensiones
- Motor de JavaScript mejorado (Sunspider) comparable al rendimiento de Chrome

[5] Descarga de Safari en: https://www.apple.com/safari

### 2.1.6 Motores de renderizado

El motor de renderizado de un navegador web es un componente que procesa contenido marcado en lenguajes tales como HTML, XML, formatos de imagen, y otros, y además interpreta información de formato como en CSS o XSL. Luego este motor despliega el contenido formateado en una pantalla o impresor.

Si bien es cierto es un componente principalmente usado en los navegadores, también podemos encontrarlos en clientes de correo electrónico, lectores de libros electrónicos, sistemas de ayuda en línea, y otras aplicaciones similares.

Las versiones modernas de los motores de renderizado tienen la capacidad de comenzar a mostrar una página aunque la información de contenido o formato aún no haya sido completamente recibida. Esto causa cambios en la visualización de la página a medida que más datos son recibidos.

Los navegadores web modernos, han optado por un enfoque modular para la representación de contenido web separando las funciones entre dos componentes bien definidos:

El motor de renderizado: este realiza la mayor parte del trabajo, tomando una URL y un conjunto de coordenadas para el área de despliegue de una ventana. Luego recupera el contenido de la URL correspondiente y básicamente dibuja una representación visual de la página dentro de la ventana. Se encarga además del manejo de enlaces, formularios, cookies, scripts, plugins, y otros elementos.

La aplicación cliente: provee elementos tales como barras de menú, barras de direcciones, barras de estado, administración de marcadores, e historial. Sirve como interface entre el motor de renderizado y el usuario y también con el sistema operativo en el que el aplicativo se ejecuta.

**Principales motores de renderizado**

**Trident**

También conocido como MSHTML es el motor de renderizado propietario utilizado en las versiones Windows del navegador Internet Explorer.

Introducido en la versión 4 del navegador en 1997, se ha actualizado periódicamente, teniendo principal importancia la actualización llevada a cabo para las versiones 7 y 8 del navegador, que le permitieron cumplir con los estándares web principales y dar soporte a nuevas tecnologías emergentes. Actualmente se utiliza la versión de Trident 8.0 incluido en Internet Explorer 11.

Trident[6] fue diseñado como un componente incrustable que permite a los desarrolladores agregar funcionalidades de navegación web a sus aplicaciones, usando C++ o .NET por ejemplo. Esta funcionalidad se agrega enlazando la librería mshtml.dll al proyecto en desarrollo.Otras aplicaciones que utilizan Trident son: Microsoft Outlook, Skype, y Windows Media Player.

**Gecko**

Gecko[7] es un motor de renderizado gratuito y de código abierto, utilizado en varias aplicaciones desarrollado por Mozilla Foundation y Mozilla Corporation, entre ellas las emblemáticas Mozilla Firefox y Mozilla Thunderbird.

Ha sido desarrollado para soportar estándares web y es utilizado por las aplicaciones para desplegar desde páginas web, hasta interfaces de usuario completas utilizando XUL. Este motor ofrece una API que permite su uso en navegadores web, aplicaciones que presentan contenido web y aplicaciones cliente-servidor. El motor está escrito en C++ y es multiplataforma. Soporta estándares web como CSS 2/3, DOM nivel 1, 2 y 3, HTML5, JavaScript, XHTML, y XML.

**Blink**

Blink[8] es un motor de renderizado desarrollado como parte del Proyecto Chromium por Google con contribuciones de Opera Software, Intel, Samsung, entre otros. Fue liberado en abril de 2013 y se considera un derivado de WebKit. Se utiliza actualmente en Google Chrome (a partir de la versión 28), Opera (a partir de la versión 15), y Amazon Silk.

Las razones de Google para deriver a Blink de WebKit es la falta de progreso (o más bien lentitud) de este último. Se estaba dando demasiada importancia y desarrollando características y detalles importantes para una sola plataforma, lo que ralentizaba el proceso de mejora.

Por lo tanto Google decidió remover estas características específicas de la plataforma del código de WebKit, prácticamente reinventar el modelo de seguridad y hacer algunos cambios que permitieran un progreso mucho más rápido en el soporte a CSS y JavaScript.

**WebKit**

WebKit[8] es un motor de renderizado para páginas y contenido web, previamente utilizado por Google Chrome y que en la actualidad potencia el funcionamiento del navegador Safari de Apple Inc.

WebKit también forma parte de los productos Amazon Kindle y BlackBerry Browser. WebKit ha sido desarrollado en C++ y provee un conjunto de clases para el despliegue de contenido web y manejo de enlaces.

WebKit está disponible bajo una licencia tipo BSD (mucho más permisiva que GPL), con excepción de los componentes WebCore y JavaScriptCore que se rigen por una licencia tipo GNU/GPL.

### 2.1.7 Principales tecnologías relacionadas

**HTML5**

En 1989 Tim Berners-Lee desarrollo HTML. Si bien es cierto el concepto de Internet ya tenía varios años de rondar los círculos científicos y técnicos, el simple hecho de encontrar los recursos y conectarse a ellos era demasiado difícil. Berners-Lee trabajó sobre este problema desarrollando dos tecnologías:

- HTTP: protocolo que permite a los servidores web funcionar
- HTML: lenguaje de script que permite la presentación de texto con enlaces incrustados en el documento haciendo referencia al mismo servidor o a uno diferente.

Una de las principales razones de la popularidad del desarrollo de Berners-Lee fue la facilidad en el aprendizaje y uso de HTML, que usa conceptos simples de etiquetas al inicio y final de una sección. El reto en la actualidad para HTML es que los usuarios no esperan páginas web tan simples como las creadas a principios de los años 90's, por lo que el W3C desarrolló un estándar actualizado al que denominó XHTML 2.0, pero que no tuvo buena aceptación.

Un estándar subsiguiente, desarrollado por el Web Hypertext Application Technology Working Group (WHATWG), es HTML5[9], que prácticamente ha descartado a XHTML 2, y que la mayoría de compañías y desarrolladores están tratando de cumplir y lo reconocen (por sobre HTML4 de 1997, inclusive) como el único nuevo estándar con el que están de acuerdo y deben soportarlo en sus productos, por ejemplo, los navegadores web.

Si HTML4 ya incluía CSS (en su primera versión CSS1), el uso de gráficos PNG, la adopción de DOM para ejecutar consistentemente JavaScript independientemente del navegador, y además la primera versión de XML, el nuevo estándar ofrece:

- Etiquetas con codecs para mostrar contenidos multimedia
- Etiquetas para manejar grandes conjuntos de datos
- Mejoras en formularios (nuevos tipos de datos)
- Visores MathML y SVG
- Características de Drag & Drop
- Código con semántica mejorada
- Nuevas APIs: drag & drop, trabajo off-line, geolocalización, almacenamiento y consultas SQL, WebWorkers
- Capacidades para interactuar a bajo nivel con la red, puertos USB, archivos, cámaras, micrófonos, y otro hardware especializado.

**JavaScript**

JavaScript[10] es un lenguaje de programación interpretado, y se considera una variante o dialecto del estándar ECMAScript. Es orientado a objetos, imperativo, y dinámico.

JavaScript es utilizado sobre todo en su forma client-side (ejecución en el lado del cliente) como parte del navegador web, permitiendo así hacer mejoras en la interface del usuario y presentar páginas web dinámicas. También puede ser ejecutado del lado del servidor (SSJS).

Es importante mencionar que aunque utiliza nombres y convenciones del lenguaje de programación Java, no está relacionado directamente con este y su semántica es diferente, siendo más similar al lenguaje C.

**Características principales de JavaScript:**

- Estructurado y relativamente seguro
- Tipos de datos automáticos (dependen del valor y no de la variable)
- Orientado a objetos (casi en su totalidad)
- Funciones de primera clase y funciones anidadas
- Se vale de prototipos en lugar de clases para implementar herencia
- Dependiente del entorno de ejecución (por ejemplo en un navegador)
- Las funciones pueden ser invocadas como métodos.
- Soporte a expresiones regulares

**CSS**

Cascade Style Sheets[11] (más conocido como CSS) es un lenguaje utilizado para controlar y definir la presentación de un documento escrito en HTML, XML o XHTML. La principal función de CSS es separar la información del documento de la forma en que debe ser presentada. CSS puede ser utilizado por:

- El creador del documento: escribiendo la información de estilo como parte de las etiquetas HTML, como una hoja de estilo interna marcada por la etiqueta <style> (para formatear un documento individual), o en una hoja externa (un archivo ".css") de modo que la misma hoja de estilo pueda ser aplicada a múltiples documentos. Esta última es la forma preferida ya que garantiza consistencia en el formateo de documentos y mejora el uso del ancho de banda al descargar menos información.
- Por el usuario que visualiza la página, definiendo una hoja de estilos que sobre escribe los estilos definidos por el creador del documento.

Las especificaciones CSS1 y CSS2 datan de 1996 y 1998 respectivamente y se abandonó su desarrollo en 2008. Actualmente se trabaja sobre la especificación CSS3[6], pero no como una especificación única, sino más bien como documentos separados denominados módulos, compatibles con las especificaciones anteriores, pero cada módulo presenta su propio estado de desarrollo. Algunos ya han alcanzado el estado de recomendación oficial, y otros son considerados solamente candidatos.

**2.2 Elección de Navegadores incluidos en la aplicación integrada a desarrollar**

Con la información presentada en los numerales anteriores, puede inferirse que actualmente la cantidad de opciones disponibles para los usuarios para navegar en la web e inclusive para acceder a aplicaciones web locales o institucionales es muy amplia y variada. En el marco teórico se presentan 5 navegadores populares pero la cantidad de navegadores independientes, más las variantes de los más utilizados (forks) es mucho más grande.

Además los sistemas y condiciones en que los usuarios navegan también varían, desde Microsoft Windows, pasando por Mac Os y distribuciones de Linux, y además entornos móviles basados en iOS y Android.

Por esta razón se considera adecuado delimitar el ámbito de este trabajo en base a tres aspectos fundamentales:

- Navegadores web a incluir en el estudio y aplicativo orientado al endurecimiento de los mismos.
- Sistemas operativos en que los navegadores operan.
- Tipo de dispositivo en que se ejecuta el sistema.

Para el desarrollo de la investigación y posterior desarrollo de una aplicación que implemente navegadores endurecidos que sean más seguros y ofrezcan mejores características de privacidad se ha determinado el siguiente conjunto de condiciones:

- Dispositivos: Computadoras de escritorio y portátiles (laptops). No se incluyen tablets, teléfonos celulares inteligentes ni consolas de videojuegos.
- Sistema operativo: Microsoft Windows 7/8 de 32 y 64 bits
- Navegadores[7]: Exclusivamente Microsoft Internet Explorer, Mozilla Firefox y Google Chrome.

[6] Estado de las especificaciones de CSS: http://www.w3.org/Style/CSS/current-work incluyendo los módulos en los que se trabaja actualmente y su compatibilidad con especificaciones anteriores

[7] Las versiones de los navegadores partirán de Internet Explorer 10/11 (según el sistema operativo Windows 7/8), Firefox 33 y Chrome 38

Para determinar el sistema operativo objetivo y los navegadores a incluir en el estudio, se utilizaron estadísticas proporcionadas por el sitio/herramienta StatCounter[8]. Las estadísticas arrojadas por este sitio en diferentes formatos, períodos de tiempo y geolocalización, se basan en el análisis de más de 3 millones de sitios a nivel global, y con un conteo de aproximadamente 15 billones de visitas a páginas de estos sitios mensualmente.

En las figuras y tablas presentadas a continuación podemos ver la tendencia de uso de sistemas operativos (para desktops y portátiles), y además los navegadores más utilizados (siempre en entorno de desktops y portátiles).

**Tendencia de uso de sistemas operativos a nivel global y en El Salvador (entre Agosto y Octubre de 2014)**

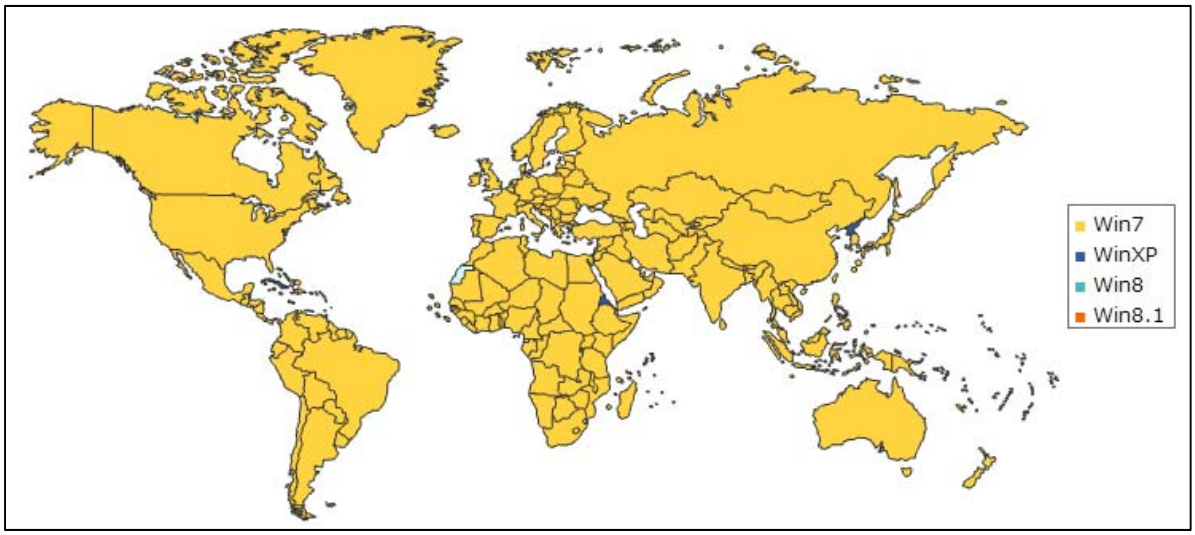


Fig. 2. Utilización global de sistemas operativos según estadísticas de StatCounter

Como puede notarse en la Fig. 2, el predominio de Windows 7 es evidente como sistema operativo más utilizado globalmente en las estadísticas recogidas por el sitio respecto al acceso a los sitios monitoreados. Pero es necesario visualizar estos datos en un gráfico que muestre porcentajes y que permita determinar el grado de utilización de sistemas “No-Microsoft” (Fig. 3)

[8] En el sitio http://gs.statcounter.com se pueden generar las estadísticas presentadas en este trabajo y además incluir otros factores de interés sobre plataformas de hardware, sistemas operativos y software utilizado para accedera los sitios monitoreados globalmente

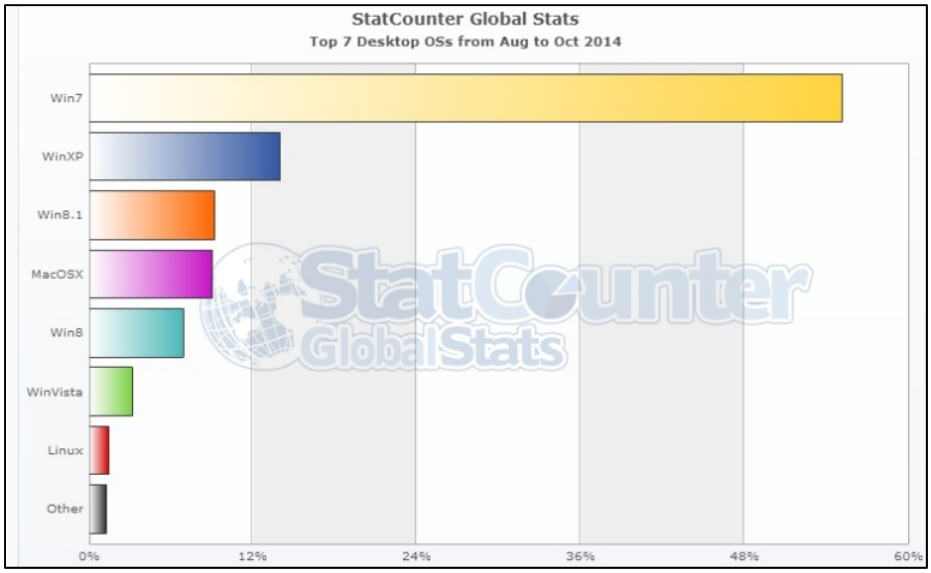


Fig. 3. Porcentaje de utilización global de sistemas operativos – gráfico de barras (según estadísticas de sitio StatCounter)

En la Fig. 3 se reflejan datos de sistemas como Mac OS (en color violeta) y de un compilado de distribuciones de Linux (en color rojo), así como otros sistemas Windows y un porcentaje bastante pequeño de otros sistemas operativos que incluyen a Firefox OS, Chrome OS y BeOS por ejemplo.

Para mostrar los datos de porcentaje exactos se incluye la Tabla 1.

| OS | Market Share Perc. (Aug to Oct 2014) |
|---|---|
| Win7 | 55.14% |
| WinXP[9] | 13.93% |
| Win8.1 | 9.22% |
| MacOSX | 8.99% |
| Win8 | 6.90% |
| WinVista | 3.14% |
| Linux | 1.40% |
| Other | 1.28% |

Tabla 1. Porcentaje de utilización global de sistemas operativos – valores (según estadísticas de sitio StatCounter)

[9] Se ha descartado para este trabajo la inclusión de Windows XP debido a la terminación del soporte oficial de parte de Microsoft desde Abril de 2014.

Los porcentajes de utilización de sistemas operativos para el acceso a los sitios web monitoreados por StatCounter también pueden encontrarse para El Salvador en el mismo período de tiempo (agosto a octubre de 2014). Los datos se presentan en la Fig. 4 y Tabla 2 respectivamente.

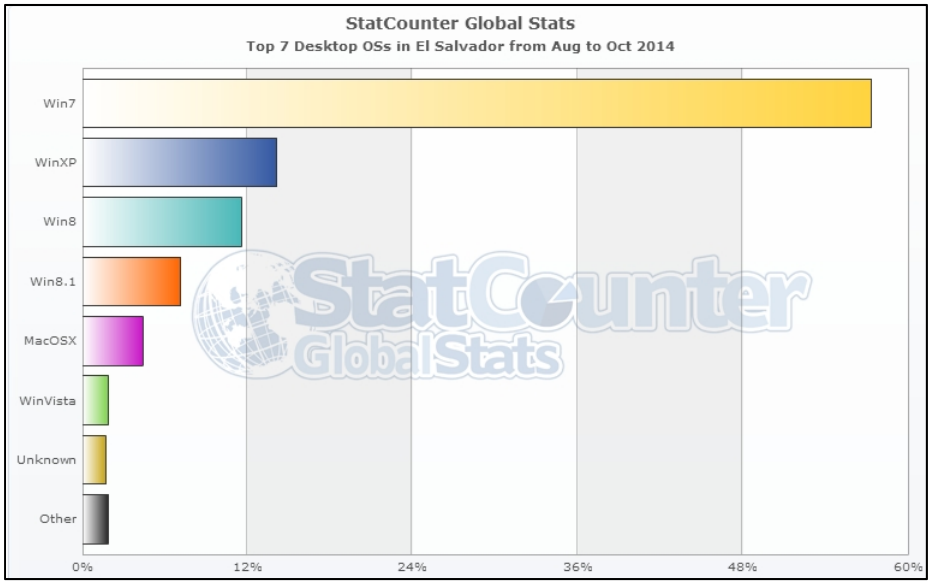


Fig. 4. Porcentaje de utilización en El Salvador de sistemas operativos – gráfico de barras (según estadísticas de sitio StatCounter)

Como puede verse en estas estadísticas, la utilización cambia ligeramente al considerar las visitas a los sitios realizadas desde ubicaciones en El Salvador, pero se mantienen las tendencias del más utilizado.

| OS | Market Share Perc. (Aug to Oct 2014) |
|---|---|
| Win7 | 57.29% |
| WinXP | 14.13% |
| Win8 | 11.55% |
| Win8.1 | 7.14% |
| MacOSX | 4.40% |
| WinVista | 1.90% |
| Linux | 1.58% |
| Other | 2.01% |

Tabla 2. Porcentaje de utilización en El Salvador de sistemas operativos – valores (según estadísticas de sitio StatCounter)

**Tendencia de uso de navegadores web a nivel global y en El Salvador (entre Agosto y Octubre de 2014)**

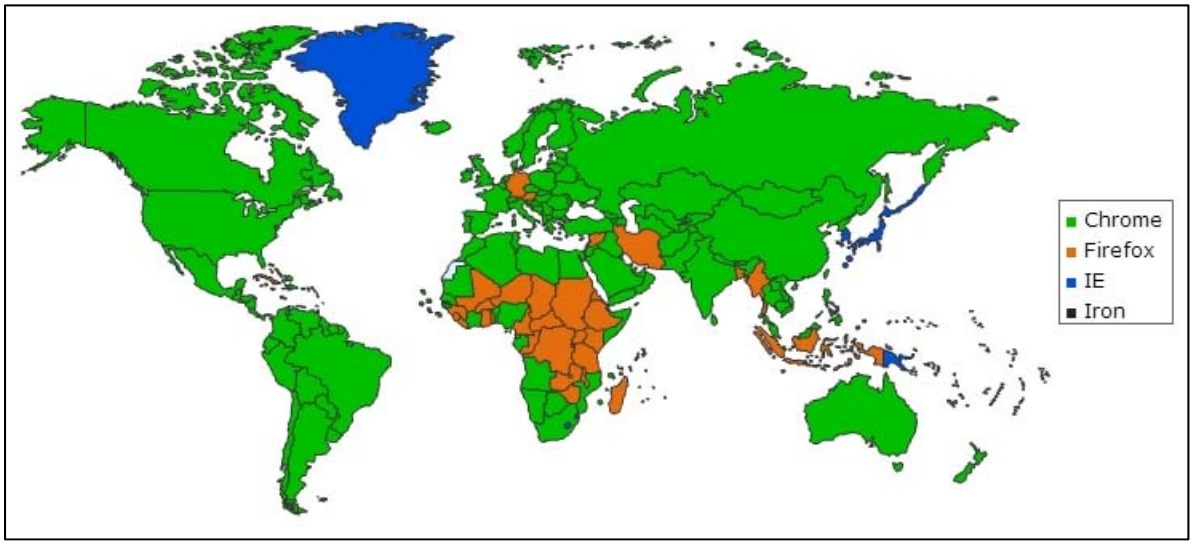


Fig. 5. Utilización global de navegadores web según estadísticas de StatCounter[10]

La Fig. 6 y Tabla 3 muestran los detalles de los porcentajes de uso de navegadores a nivel global, resultando estos valores por supuesto de los incluidos en los sistemas operativos de escritorio considerados.

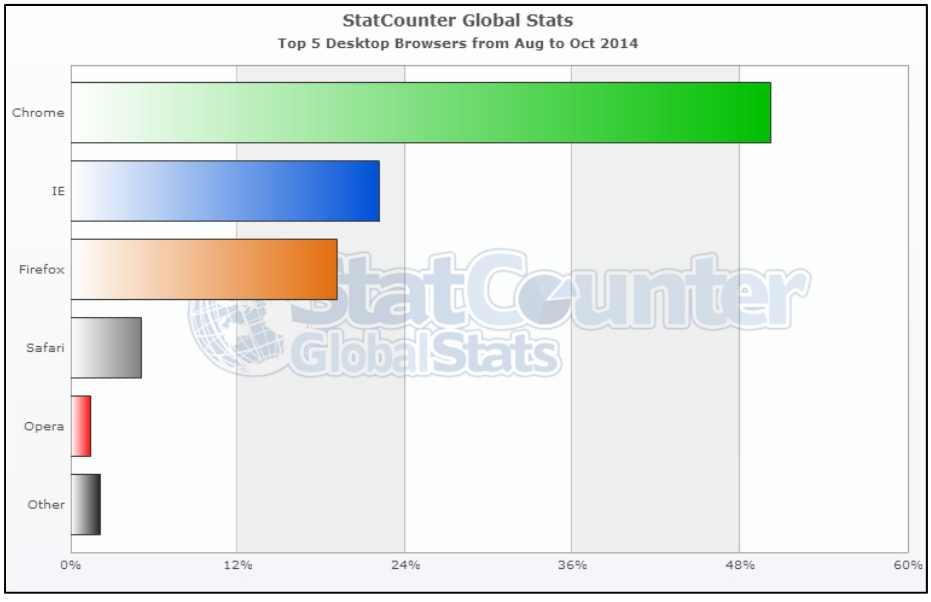


Fig. 6. Porcentaje de utilización global de navegadores web – gráfico de barras (según estadísticas de sitio StatCounter)

[10] SRW Iron es un navegador web basado en Chromium y aparece solo en las estadísticas con mapas pero no en barras ni en tablas de datos

| Browser | Market Share Perc. (Aug to Oct 2014) |
|---|---|
| Chrome | 50.18% |
| IE | 22.07% |
| Firefox | 19.09% |
| Safari | 5.06% |
| Opera | 1.46% |
| Other | 2.14% |

Tabla 3. Porcentaje de utilización global de navegadores web – valores (según estadísticas de sitio StatCounter)

Las mismas estadísticas se incluyen en la Fig 7 y Tabla 4 para el caso específico de uso de navegadores para El Salvador.

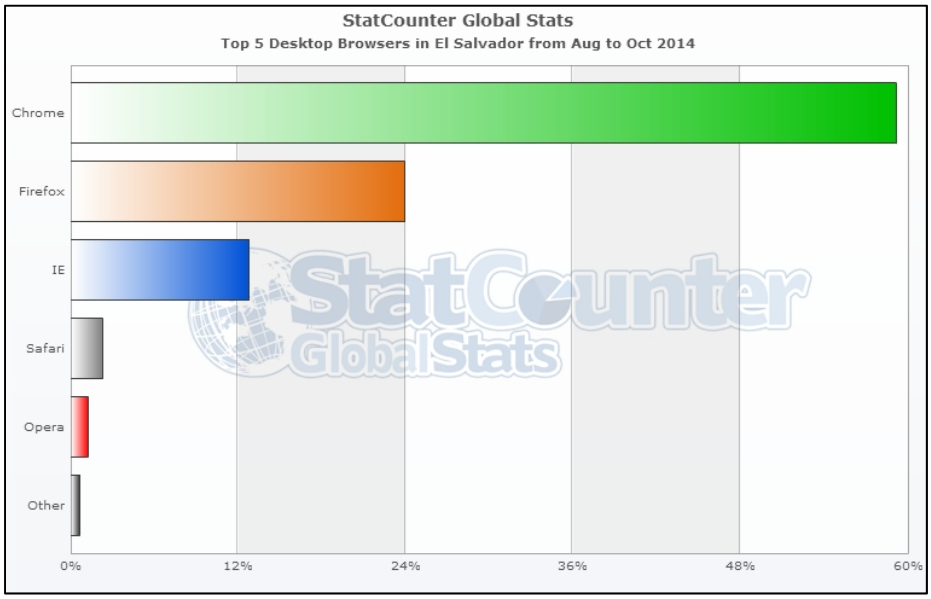


Fig. 7. Porcentaje de utilización en El Salvador de navegadores web – gráfico de barras (según estadísticas de sitio StatCounter)

| Browser | Market Share Perc. (Aug to Oct 2014) |
|---|---|
| Chrome | 59.14% |
| Firefox | 23.92% |
| IE | 12.80% |
| Safari | 2.26% |
| Opera | 1.21% |
| Other | 0.67% |

Tabla 4. Porcentaje de utilización en El Salvador de navegadores web – valores (según estadísticas de sitio StatCounter)

## III. VULNERABILIDADES, AMENAZAS Y ATAQUES EN LOS NAVEGADORES WEB

### 3.1 Conceptos importantes:

**Vulnerabilidad**

Este término se refiere a una debilidad en un sistema informático, que permite a un atacante modificar o ignorar los controles de acceso del sistema y por tanto violar la confidencialidad y privacidad de datos y aplicaciones.

**Amenaza**

Se puede definir una amenaza como un elemento o acción con capacidad para atentar contra la seguridad de la información.

La amenaza surge a partir de la existencia de una vulnerabilidad, es decir que una amenaza solo puede existir si existe una vulnerabilidad que pueda ser aprovechada (explotada).

**Ingeniería Social**

Se denomina ingeniería social[12] a la práctica de obtener información privada o confidencial a través del engaño o manipulación de usuarios con legitimo acceso a dicha información, aprovechando el principio o afirmación que los usuarios son un eslabón débil en la cadena de custodia de la información.

**Ataque**

Un ataque[13] (informático) consiste en conocer y aprovechar alguna o varias vulnerabilidades en el software, en el hardware o inclusive en los usuarios (recabados vía ingeniería social) que forman parte de un sistema informático, con la finalidad de obtener algún beneficio (reputación,

económico, político, entre otros), causando un efecto negativo en la seguridad del sistema, y que repercute en los activos de información de un individuo u organización. Básicamente es la materialización de una amenaza.

**Seguridad de la Información**

Consiste en asegurar que los activos de información de una organización, sus contenidos, capacidades y características, son utilizados justo en la manera en que fueron concebidos, garantizando que el acceso a la información en el momento en que se requiera, por parte de usuarios autorizados, y con la certeza que la información accedida es integra y fiable. En otras palabras, que se protegen los tres pilares de la seguridad: Disponibilidad, Confidencialidad e Integridad.

**Plugin**

Componente o pieza de software que añade una funcionalidad específica a los navegadores (por ejemplo la visualización de un tipo de archivo o formato en particular).

El plugin regularmente se instala en el sistema e inclusive puede ser accedido por varios navegadores, de modo que no es específico ni exclusivo para un navegador. Los plugins en algunos casos pueden ejecutarse fuera del entorno del navegador.

**Extensión**

Una extensión es una aplicación o conjunto de componentes de software que agregan o remueven funcionalidades al navegador web. Por lo regular no son programas aislados o independientes como los plugins, y son diseñados para un navegador en particular y no pueden ser ejecutados sin antes realizar un proceso de instalación dentro del navegador.

**Complemento**

Clasificación que reúne a las extensiones, plugins y otros componentes de software que agregan funcionalidad a otro programa (por ejemplo temas, skins, y barras).

**Aplicación Web**

Aplicaciones web son las herramientas de software que los usuarios pueden utilizar accediendo al servidor web que la aloja, mediante el uso de un navegador web. Ventajas de estas aplicaciones son: fácilmente actualizables de forma centralizada, no son dependientes del sistema operativo y no necesitan la instalación de un cliente específico en cada equipo de usuario final que deba tener acceso a la aplicación.

**Script**

Programa regularmente almacenado en un archivo de texto que por lo general es procesado por un intérprete para realizar tareas diversas tales como interactuar con el sistema operativo o con el usuario.

## 3.2 Los navegadores web como entidades vulnerables

### 3.2.1 Evolución bajo presión y sin preocupaciones fundamentales de seguridad

Los navegadores web han tenido una de las historias evolutivas más competitivas y dramáticas en lo que a tecnología se refiere, y así sigue siendo ya que un navegador que no evoluciona para llenar las expectativas de sus potenciales usuarios es abandonado y olvidado en ese entorno tan agresivo y competitivo.

Los navegadores en sus inicios eran aplicativos mucho menos complicados y sofisticados con un propósito bastante simple: debían mostrar páginas relativamente sencillas y permitir el seguimiento a los vínculos contenidos en las mismas, en una época en que la web contenía solamente un pequeña fracción de la información que hoy en día tenemos disponible.

En la actualidad los navegadores deben seguir cumpliendo esa tarea básica, pero al mismo tiempo deben brindar soporte a extensiones que agregan características, deben tener la capacidad de incluir plugins para permitir el acceso a información en formatos para los que no se incluyó soporte originalmente en el programa, deben permitir el acceso a recursos de hardware como las cámaras o micrófonos de una computadora para brindar una experiencia de comunicación completa al usuario, inclusive, deben permitir a servicios de terceros geolocalizar la ubicación del sistema y equipo en que el navegador se ejecuta para ofrecer al usuario final una mayor sensación de personalización.

Entonces, la necesidad de ofrecer estas capacidades con la rapidez suficiente para no perder en la carrera con otros navegadores, y a ser posible superarlos, en muchos casos deja a la seguridad y privacidad como las áreas menos atendidas en nuevas versiones, aunque esto no significa que las empresas que los desarrollan olviden por completo la importancia de la seguridad, pero probablemente se concentrarán más en una característica que permita la visualización de cierto tipo de contenido, que en la posibilidad de que un atacante pueda aprovechar una vulnerabilidad no corregida en el navegador, y simplemente se deja para ser parchada en un futuro próximo.

Por ejemplo, en 2013 cuando se liberó la versión 18 de Mozilla Firefox, se le dio mucha publicidad al hecho que en esta versión se había corregido una vulnerabilidad relativa a la prevención de contenido mixto, deshabilitando la carga de contenido HTTP si el origen presentaba un perfil HTTPS.  El problema es que la vulnerabilidad fue reportada en diciembre del año 2000.

### 3.2.2 Superficie de ataque de los navegadores

El término de “superficie de ataque” de un activo informático (hardware o software) se refiere a la región o componentes que son vulnerables y por tanto potenciales puntos de interés para posibles atacantes.

En el caso de los navegadores web, la superficie de ataque básicamente se refiere a los componentes y funcionalidades vulnerables a la influencia de fuentes de datos e instrucciones no confiables.

Eso significa que la superficie de ataque de un navegador web se extiende a su motor de renderizado, sus intérpretes, sus analizadores, y sus interfaces con el sistema operativo y con la red.

La superficie de ataque se ve aumentada por elementos que no son nativos al navegador pero que indudablemente serán requeridos por el usuario final o definidos por una política institucional. Las extensiones por ejemplo pueden verse influenciadas por páginas que el navegador carga y esta influencia puede ser el resultado de código malicioso, añadiendo vulnerabilidades que posiblemente ya han sido corregidas en el núcleo del navegador, pero no en la extensión.

Lo mismo ocurre con los plugins, que son ejecutados cuando una aplicación web o tipo de archivo propietario lo requiere. Los plugins especialmente (muchos de ellos) tienen un historial de vulnerabilidades de seguridad (Flash, visores PDF, Java y visores de imágenes y vídeo).

Además no cuentan con un mecanismo o capacidad de instalación o actualización centralizada por lo que es difícil y complejo asegurar que todos los usuarios cuentan con plugins correctos, actualizados y relativamente seguros, ya que la actualización en el peor de los casos debe hacerse manualmente. Esto por supuesto hace más grande la superficie de ataque.

### 3.2.3 Malas prácticas de seguridad que afectan directamente a los navegadores

**Principio de robustez en el diseño de Internet**

También conocido como la Ley de Postel[14], induce a los diseñadores y desarrolladores de software para Internet a “ser conservadores en lo que hacen, pero ser liberales en lo que aceptan de otros” .

Este principio por supuesto no se apega a técnicas ni políticas de seguridad funcionales, pero es un problema directo que afecta a los navegadores ya que desde su concepción fueron pensados en permitir la ejecución de instrucciones en muy variadas formas y desde ubicaciones potencialmente peligrosas.

De la mano con la seguridad implementada en los navegadores, también debería animarse a los desarrolladores de aplicaciones web en cambiar su adhesión al principio de robustez, de modo que “sean conservadores con lo que hacen y sean mucho más conservadores con lo que aceptan de otros”. Si este pensamiento se generalizara entre desarrolladores, los posibles atacantes definitivamente se encontrarían con entornos mucho más seguros y difíciles de atacar.

**Seguridad solamente implementada en el perímetro**

Hoy en día, la mayoría de las organizaciones y empresas consideran que la seguridad aplicada a sus fronteras o bordes (firewalls por ejemplo) es suficiente para proteger todos sus activos de información, y de hecho la mayor parte del presupuesto a este tipo de seguridad.

Claro que la seguridad del perímetro exterior es importantísima, pero si se dedica todo el esfuerzo y recursos a potenciarla, se está asumiendo que los atacantes solamente intentarán

irrumpir desde la capa más externa y luego intentarán romper las defensas en el orden en que han sido establecidas. Esto, por supuesto no refleja el mundo real, ni las técnicas actuales de los potenciales atacantes. Debe considerarse por tanto la existencia de un micro perímetro alrededor de recursos como el navegador web, el cual también necesita ser protegido, muy probablemente de atacantes internos.

### 3.2.4 Metodología generalizada de ataque a los navegadores web

Como se puede observar en los tópicos anteriores, la tarea de proteger una aplicación como los navegadores web no es fácil. Es más, puede considerarse bastante compleja, ya que la variedad de ataques posibles es muy extensa, y ni siquiera requieren la participación activa del atacante en tiempo real, si no que el ataque pudo haber sido implementado en una página que forma parte de un sitio que se considera seguro, y cuando el navegador accede a este recurso en línea, si no está suficientemente protegido o endurecido, será presa fácil del atacante.

De lo anterior podemos deducir además, que algunas técnicas de protección que pudieron resultar prácticas en el pasado, como por ejemplo el establecimiento de listas negras, no son muy efectivas cuando se trata de proteger una aplicación que utilizamos para acceder a cantidad de sitios con contenido muy diverso, y no es factible determinar a ciencia cierta cuales son los sitios y recursos maliciosos y bloquearlos.

Si bien es cierto, los ataques pueden tomar muchas formas y seguir diferentes metodologías, es necesario definir algunos elementos en común que permitan determinar fases importantes en los ataques, y que a su vez, permitan categorizar las acciones, procedimientos y tecnologías orientadas a impedir o detener los ataques, y a reducir los riesgos de los mismos.

**Etapas en el ataque a un navegador web**

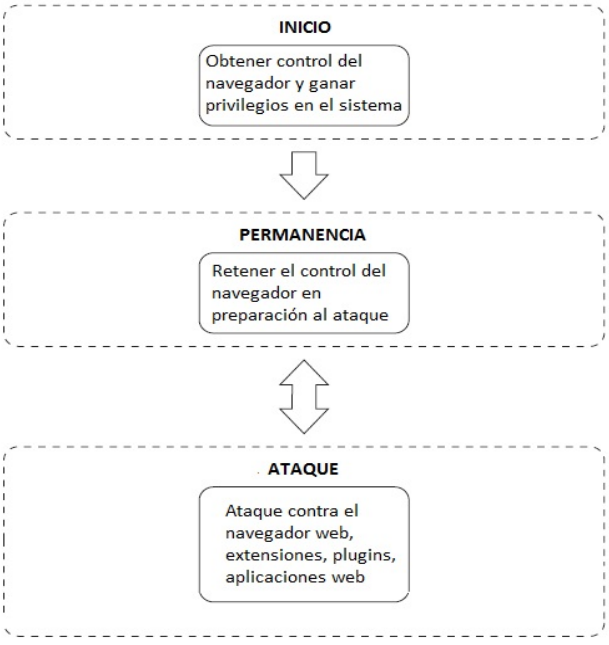


Fig. 8 Etapas generales del ataque a un navegador web[15]

**Etapa de inicio**

Esta etapa probablemente a nivel técnico pueda parecer relativamente sencilla y que no se causa daño alguno, pero realmente es la fase más importante ya que si no es completada, no es posible ejecutar el ataque manera completa ya que el navegador estaría fuera del alcance del potencial atacante. En esta fase puede probarse una gran cantidad métodos para obtener control, incluso variantes de los ya conocidos, esperando encontrar vulnerabilidades no atendidas, y si se consigue la ejecución de código propio del atacante puedo forzarse al navegador a pasar a otras etapas.

**Etapa de permanencia**

El control obtenido en la fase anterior debe ser mantenido de modo que puedan iniciarse ataques a partir de este punto. Esto puede lograrse incluyendo en el código malicioso original, instrucciones que fuercen al navegador a consultar por más instrucciones al atacante, de modo recurrente, y esas instrucciones por supuesto constituyen la siguiente fase del ataque.

Claro que esta situación posiblemente no pueda ser mantenida de manera indefinida. Por ejemplo, si las fases de inicio y permanencia tienen influencia sobre una pestaña en particular, y el usuario decide cerrar esa pestaña, el ataque podría terminar antes de lo deseado por el atacante. Debido a esto, regularmente luego de comprometer un elemento en particular del navegador, se ensayan métodos de control que tengan un efecto más extenso en el mismo.

**Etapa de ataque**

En esta etapa se evalúa el nivel de control obtenido sobre el navegador para poder lanzar los verdaderos ataques, que pueden presentar diferentes formas, por ejemplo, ataques locales, comprometiendo al sistema operativo en el que se ejecuta el navegador, hasta ataques remotos que involucren a otros sistemas en la red local o incluso en redes externas al sistema atacado.

Uno de los ataques típicos de esta fase consiste en saltar o evitar las políticas de origen (SOP – Same Origin Policy, más adelante se incluye información detallada al respecto) ya que al lograr esto, prácticamente se está pasando sobre un nivel de seguridad interno del navegador, permitiendo que otros tipos de ataque sean factibles.

Una vez se ha violentado este nivel de seguridad, el atacante y su código malicioso, pueden dedicarse a atacar al navegador en sí, a sus extensiones, a sus plugins, a otros equipos en la red (local o amplia), inclusive pueden lanzarse ataques que tengan como objetivo al usuario que trabaja con el navegador, usando técnicas de ingeniería social potenciadas por software.

## 3.3 Técnicas para iniciar y mantener el control de un navegador web

### 3.3.1 Técnicas basadas en Cross-Site Scripting (XSS)

Antes de 1995 y de la introducción de JavaScript en el navegador Netscape, básicamente el contenido web era entregado vía HTML de forma estática.

Cuando esto cambió con el uso de un lenguaje dinámico, no pasó mucho tiempo antes que se dieran a conocer las primeras formas de código malicioso

Una de las principales formas en que se comenzó a violentar la seguridad fue a través de técnicas en las que contenido no confiable o inseguro es procesado y posteriormente se convierte en confiable para el navegador que recibe las instrucciones. Si este código contiene HTML, JavaScript, VBScript o cualquier otro tipo de contenido dinámico, el navegador podría llegar a ejecutar instrucciones potencialmente peligrosas. A esto se le denominó Cross-site Scripting (al inicio conocido como CSS). Debido a la coincidencia con Cascade Style Sheets y para evitar confusiones, se cambió la denominación a XSS. En la actualidad se conocen y explotan diversos tipos de XSS entre los cuales se cuentan:

**Reflected Cross-site Scripting**

Es una de las formas de XSS más comunes. En este caso, contenido no confiable es enviado por el usuario a una página o aplicación web, y este mismo contenido es devuelto como parte de una respuesta. El navegador al evaluar que las instrucciones proceden de un servidor web con el que ya tuvo contacto y considera seguro, procede a ejecutar código arbitrario. Es interesante mencionar que este tipo de XSS se mantiene dentro de las políticas de SOP, es decir no las violenta.

**Stored Cross-site Scripting**

También conocido como Persistent XSS, es similar a la variante Reflected, excepto que el código XSS de hecho ya reside en el almacenamiento de la aplicación web (por tanto no debe ser introducido por el usuario). Esto implica que cualquier usuario que visite una página en el sitio comprometido puede ejecutar el código malicioso. El código puede estar almacenado en una base de datos, pero también existen otros recursos de almacenamiento en que puede encontrarse, tal como archivos log.

**DOM Cross-site Scripting**

En las variantes anteriores de XSS se nota que la vulnerabilidad y el código maliciosos deben estar presentes en el servidor web.

En la variante DOM XSS la vulnerabilidad existe exclusivamente en el código del lado cliente (JavaScript por ejemplo).

**Universal Cross-site Scripting**

Variante de XSS que no está restringida por los principios de SOP. Aprovecha vulnerabilidades propias del navegador o de sus extensiones o plugins.

**Virus XSS**

Variante de Stored XSS. En este caso la vulnerabilidad, almacenada en el código directamente o en una base de datos.

“Infecta” al navegador que visita la página, y luego no solamente afecta a los recursos de ese sitio o aplicación en particular, si no que trata de infectar o atacar a otras aplicaciones web fuera del dominio de la aplicación original. Estas vulnerabilidades también tienen capacidades de auto replicación y propagación.

### 3.3.2 Técnicas basadas en ingeniería social

**Phishing**

Comprende una variedad de componentes entre los cuales se puede mencionar sitios web falsos, mensajes de correo electrónico falsos y también mensajes instantáneos falsos.

La intención principal de estos recursos maliciosos es atraer a las potenciales víctimas hacia sitios falsos y comprometidos para obtener credenciales válidas y así acceder a los recursos verdaderos del usuario (portales de banca en línea, sitios de compras en línea, servicios gubernamentales, entre otros).

Las variantes más utilizadas de phishing son:

- E-mail phishing: correo electrónico enviado a múltiples destinatarios (masivamente por lo regular), solicitando que se responda a ese mensaje con información privada del usuario y que el atacante considera de valor. En ocasiones también se incluyen adjuntos o enlaces maliciosos que al ser abiertos pueden enviar información al atacante sin conocimiento del usuario.

- Phishing en sitios web: se prepara un sitio web falso (construido desde cero, o clonado del sitio real) y se intenta que los usuarios se validen en el sitio falso con sus credenciales verdaderas de modo que el atacante puede recolectarlas y usarlas posteriormente en el sitio verdadero.

- Spear phishing: utiliza también sitios web falsos, pero está orientado a usuarios bien definidos, no es masivo.

- Whaling: phishing orientado a usuarios de alto nivel en una organización (gerentes, directores, y posiciones similares).

Si bien es cierto la finalidad principal del phishing es la obtención de información privada del usuario, también puede ser utilizado para atacar a navegadores web, ofreciendo enlaces a sitios web comprometidos y luego lanzando ataques basados en otras técnicas (como XSS). Esto se logra utilizando técnicas adicionales de ingeniería social como el denominado “baiting”, en el que se ofrece un beneficio o recompensa al usuario por visitar un sitio en particular.

Para hacer más efectivas estas técnicas, se acompañan por métodos de ofuscación de las URLs (como los sistemas para acortarlas) o bien recursos visuales como los códigos QR (sobre todo orientado a móviles).

### 3.3.3 Técnicas tipo Man-in-the-Middle
### Man-in-the-Browser

A diferencia de los tradicionales ataques tipo Man-in-the-Middle, que ocurren en capas inferiores del modelo de referencia OSI, MitB[16] se desarrolla completamente dentro del navegador, pero mantiene atributos de MitM, por ejemplo:

- No son notados por el usuario
- Están ocultos también al servidor
- Son capaces de modificar contenido solicitado
- Son capaces de leer el contenido solicitado
- No requieren la participación del atacado

Este tipo de intrusión se ha visto incrementada últimamente en software y código malicioso orientados a ataques contra sistemas de banca en línea, e implementado a gran escala utilizando botnets.

**Envenenamiento de DNS**

Método para implementar ataques MitM que apunta a los varios niveles de funcionalidad del servicio DNS para obligar a un usuario final a visualizar documentos y contenido comprometido con otro tipo de ataques. Por ejemplo, se puede insertar registros maliciosos en el archivo "hosts" del equipo local, de modo que al intentar resolver un nombre a dirección, se entregue el valor de la dirección ya preparada por el atacante.

En caso de no poder acceder directamente al archivo local de un equipo, o necesitar que el ataque aplique por ejemplo a toda una subred se pueden insertar registros maliciosos en el servidor DNS local, de modo que ante cualquier solicitud de resolución, se entreguen las direcciones preparadas por el atacante. Una tarea un tanto más difícil es tratar de modificar un servidor DNS de nivel superior, que por lo regular están bajo la supervisión de proveedores de servicio u organismos que dedican mucho esfuerzo a mantener la seguridad de sus recursos.

Una variante de estos ataques de envenenamiento de DNS puede usarse en caso de no poder insertar los registros maliciosos en servidores DNS auténticos. Este tipo de ataque se combina con ataques de envenenamiento de ARP de modo que se fuerza a los equipos cliente en una red a dirigir sus peticiones DNS a una dirección que parece válida (a nivel de IP) pero que corresponde a un equipo del atacante (a nivel de MAC).

### 3.3.4 Uso de IFrames

La etiqueta <iframe> es usada intensivamente en el diseño de páginas HTML como una forma sencilla de incrustar otro documento en la página actual (por ejemplo para mostrar contenido publicitario incrustado en todo tipo de sitios web).

Por su forma de funcionamiento, esta etiqueta puede ser usada para dar características de persistencia a ataques contra navegadores, esto es debido a que un atacante tiene control sobre el contenido completo DOM del IFrame, incluyendo CSS, HTML y JavaScript.

Mediante funciones relativamente sencillas pueden crearse ya sea IFrames ocultos o IFrames superpuestos o traslapados. Los IFrames ocultos pueden ser tan pequeños como 1 pixel y sin bordes. Los superpuestos pueden ajustarse exactamente al tamaño completo de la ventana del navegador sin bordes, de modo que al cargarse el IFrame solo unos cuantos milisegundos después se carga el contenido solicitado por el usuario, causando que toda la interacción con el navegador sea con IFrame malicioso y no con la página auténtica (por ejemplo una página de login para ingresar a un sistema).

### 3.3.5 Uso de características del navegador, motores e intérpretes

**Eventos del navegador**

Este tipo de técnica orientada a mantener el control sobre el navegador está basada en tiempo.

Pareciera una técnica sencilla y no muy eficaz, pero su objetivo es tratar de mantener una ventana o pestaña del navegador abierta por la mayor cantidad de tiempo posible, permitiendo la ejecución de más código malicioso.

Por ejemplo, un usuario intenta cerrar una pestaña que presenta contenido que no le interesa (y que ha sido comprometida). El navegador al intentar cerrar la pestaña puede solicitar una confirmación de cierre que probablemente no sea atendida por el usuario, dejando así la pestaña abierta hasta que un tiempo después, el usuario al notar que no fue cerrada, efectivamente hace clic sobre el diálogo de confirmación de cierre.

Esos segundos o minutos adicionales pueden hacer la diferencia en la cantidad de código malicioso ejecutado.

Inclusive en navegadores como Firefox puede utilizarse esta técnica dos veces seguidas: la primera haciendo uso de un diálogo propio del navegador solicitando confirmación de cierre y una vez el usuario hace click en el botón correspondiente puede lanzarse un segundo diálogo controlado por JavaScript que vuelve a solicitar la confirmación.

**Ventanas tipo Pop-under**

La mayoría de usuarios está familiarizada con los molestos pop-ups que son abiertos ante un evento como un click en un enlace. Ante la aparición de estas ventanas adicionales, se tiene la protección anti pop-ups integrada en los navegadores (debe estar activa) y si no funciona completamente, al ser evidentes las ventanas son cerradas una a una por el mismo usuario.

Pero también, mediante código malicioso, puede abrirse una nueva ventana con la característica pop-under, y por tanto la ventana comprometida, no se mostrará al usuario más que en la barra de tareas del sistema operativo, pasando en muchos casos inadvertida por mucho tiempo. Los navegadores modernos, también incluyen protección anti pop-under cuando analizan el código y de hecho encuentran la propiedad “popunder” y al considerar que una ventana será abierta sin participación directa del usuario. Pero existen variantes, en las cuales se manipulan los eventos del mouse por ejemplo, mediante JavaScript, para saltar estas protecciones incluidas en el navegador.

**Uso de texto codificado**

Los navegadores modernos tienen incluidos filtros que permiten encontrar código potencialmente malicioso al buscar por ejemplo apariciones de “eval” en el código.

Una forma relativamente fácil de esconder estos fragmentos de código es codificándolos, por ejemplo usando codificación base64 para formar el equivalente de la cadena de texto, o bien formando la misma cadena de caracteres a través de la transformación de códigos no alfanuméricos[17].

## 3.4 Técnicas utilizadas en la fase de Ataque

### 3.4.1 Ataques a las Políticas del Mismo Origen (Same Origin Policy – SOP)

**Importancia de SOP**

La política del mismo origen (SOP) probablemente es una de las medidas de control más importantes en la actualidad respecto al contenido web. Pero, al mismo tiempo que es muy importante, también es una de las especificaciones implementadas de manera más inconsistente e potencialmente insegura.

La razón de la importancia se debe al hecho que las páginas y los recursos que incluyen se asume que tienen el mismo nombre de host, esquema y puerto, y por tanto provienen del mismo origen. Si alguno de estos atributos cambia, entonces el recurso puede tener un origen diferente y ser potencialmente malicioso.

La intención de SOP entonces, es en principio evitar la interacción entre recursos de orígenes distintos, y esto puede parecer muy simple al inicio, pero cuando se considera cual navegador se está utilizando, y cuáles son las extensiones y plugins involucrados puede volverse realmente complejo cumplir las políticas.

Además, en un inicio SOP aplicaba sobre todo para recursos externos, pero ha sido extendido a otros tipos de orígenes, por ejemplo el esquema “file” que es local al equipo en el que ejecuta el navegador, sus extensiones y plugins.

**El papel de CORS**

CORS (Cross-origin resource sharing) es un mecanismo que permite que un recurso de una página web (JavaScript por ejemplo) pueda ser solicitado por una página en otro dominio fuera del que se considera su origen. Por defecto, el compartir estos recursos estaría prohibido en los navegadores (por SOP), pero CORS define una forma válida para permitir estos recursos compartidos entre orígenes. Debe configurarse en el servidor web la respuesta que será enviada en los encabezados al solicitante, determinando así los dominios válidos desde los cuales pueden accederse recursos compartidos.

Este mecanismo, si no es correctamente configurado puede convertirse en una puerta para muchos ataques a SOP.

### 3.4.2 Entornos en los que se puede atacar a SOP

A continuación se listan algunos ejemplos relevantes de los entornos en los que se pueden observar ataques a la política de mismo origen, pero es necesario aclarar que la cantidad de entornos, aplicativos y complementos que pueden verse expuestos a un ataque es mucho más extenso.

**Ataques a SOP en Java**

Las versiones de Java 1.7u17 y 1.6u45 por ejemplo, no obligan a cumplir SOP si dos dominios son resueltos vía DNS a la misma dirección IP.

Por lo tanto, se permitirán peticiones con orígenes distintos (cross-origin) y se recibirán las respuestas a dichas peticiones. Esta vulnerabilidad puede convertirse en crítica, si es explotada en un entorno virtual en donde varios dominios son administrados por el mismo servidor y son resueltos con la misma dirección IP.

**Ataques a SOP en Internet Explorer**

En las versiones previas a IE 8b2 (por ejemplo IE 6 y 7) existían diversas vulnerabilidades relacionadas a SOP, sobre todo en la implementación de la propiedad “document.domain”, la cual podía ser simplemente reescrita y con esto se obviaba el valor original de la propiedad.

**Ataques a SOP en Firefox**

En las versiones 15 y 16 de Firefox se descubrieron al menos dos vulnerabilidades importantes relativas a SOP. La primera estaba relacionada al uso la propiedad “window.location” que debería ser por defecto de solo lectura, y la vulnerabilidad permitía el acceso no autorizado al valor de la propiedad. El impacto fue tal, que se restringió la descarga de Firefox 16 desde los sitios oficiales, hasta que la vulnerabilidad fue corregida.

La otra vulnerabilidad se relacionaba al soporte de IFrames en sandbox. Si se colocaba el valor “allow-scripts” al atributo “sandbox” de un IFrame, se permitía la ejecución de código JavaScript

que podía modificar la propiedad “window.top”, y así poder modificar la propiedad “location” de la página padre.

**Ataques a SOP en CORS**

Los anteriores tipos de ataques y las vulnerabilidades que los permitían residían en los aplicativos y complementos en el lado cliente. Pero también se pueden potenciar los ataques a SOP desde el lado servidor.

La mala configuración de las funcionalidades de CORS puede convertirse en una de las formas más explotadas y peligrosas de ataques a SOP. Basta con incluir en un servidor un valor de wildcard en las respuestas al encabezado “Access-control-allow-origin”, es decir un asterisco (*) para permitir peticiones entre dominios y efectivamente poder acceder a las respuestas a dichas peticiones.

Si se desea usar este valor en el encabezado para implementar una política laxa de SOP, solo debería implementarse para servir contenido que no se considere sensitivo en términos de seguridad y privacidad.

### 3.4.3 Ataques dirigidos al usuario del navegador

Si bien es cierto, las vulnerabilidades y los ataques que las explotan están dirigidas en contra del navegador, en muchas ocasiones se ven potenciadas por la participación del usuario final del software, el cuál posiblemente engañado por técnicas de ingeniería social, confía en los contenidos que se le ofrecen, incluso cuando estos son obviamente maliciosos.

**Desfiguración del contenido (Defacing)**

Uno de los métodos preferidos por los atacantes y que resulta relativamente más fácil de implementar para ellos es hacer Defacing de una página individual o hasta de un sitio completo. Esta desfiguración del contenido original lleva al usuario a realizar acciones no deseadas, como por ejemplo, entregar al atacante información sobre sus credenciales de ingreso a un sistema o información privilegiada y confidencial como por ejemplo información financiera en un sistema de banca en línea.

Si el atacante ha sido capaz de ejecutar JavaScript en un origen diferente (el ataque comienza como un caso de XSS), puede efectivamente obtener partes de la información del usuario y al mismo tiempo insertar sus propios datos.

Algunos de los métodos más utilizados en los ataques de defacing son los siguientes:

- Captura de las entradas producidas por el usuario: las capturas pueden ser implementadas de diversas maneras, inclusive cuando el usuario aún no ingresa datos en un formulario, puede registrarse su actividad dentro del navegador, por ejemplo:

- Mediante eventos de “foco” y de mouse: el atacante puede utilizar los eventos “focus” y “blur” para registrar la ubicación precisa en que el usuario ha hecho click. Esto facilita al atacante poder insertar posteriormente elementos en los que el usuario puede hacer click y ejecutar código malicioso adicional. El registro de estos eventos es especialmente útil al

atacante cuando se interactúa con teclados virtuales. Algunos de los eventos monitoreados son “mouseover”, “mousemove”, “mousedown” y “mouseup”

- Usando eventos de teclado: se puede añadir al contenido rutinas para registrar los eventos “keydown”, “keypress” y “keyup”, que permitirán al atacante conocer por ejemplo cuando y como se está registrando datos en un formulario.

- A través del uso de eventos de formularios: algunos atacantes pueden concentrarse en la actividad realizada por el usuario en los campos de un formulario, y así no deben analizar largos listados de eventos de mouse y teclado.

- Mediante la superposición y registro de actividad en un IFrame: como ya se había mencionado anteriormente, un IFrame podría ser colocado sobre la verdadera página web que contiene al formulario, y entonces el usuario no estaría llenando los verdaderos campos si no escribiendo o haciendo click en las regiones preparadas por el atacante. Para hacer transparente este tipo de ataque al usuario, los valores introducidos, serán pasados al formulario posteriormente.

**Ataques a las pestañas del navegador (TabNabbing)**

Este tipo de ataque (tabnabbing[18]) se basa en la cantidad de tiempo en que una pestaña abierta pasa “inactiva”, es decir que ha perdido el foco (usando el evento “blur” para determinar cuánto tiempo ha pasado desde que se le dio atención por última vez a dicha pestaña).

Si ha pasado un tiempo considerado por el atacante como suficiente (de modo que el usuario no está atento a lo que sucede con esa pestaña), se puede reemplazar su contenido con una página maliciosa (por supuesto con una URL distinta), previamente clonada por el atacante, inclusive en detalles tan mínimos como el favicon.

El tabnabbing es especialmente efectivo para los atacantes de sistemas de banca en línea, que confían en que los usuarios abrirán la página del servicio del banco y la dejarán desatendida por un tiempo, entonces se aprovecha para cambiar la página, y al regresar a esta pestaña, el usuario no notará el cambio en la URL y volverá a interactuar con la página (falsa), entregando información sensible al atacante. Por ejemplo, se puede reemplazar la página original con una en que se indique al usuario que su sesión ha expirado y que necesita validarse nuevamente, con lo que el atacante conseguirá las credenciales de ingreso del usuario.

**Ataques basados en las notificaciones del navegador**

En la actualidad la mayoría de los navegadores han cambiado la forma en la que notifican al usuario de actividades tales como la descarga de archivos o la ejecución o solicitud de permisos por parte de un plugin, de una forma “modal” a una forma “modeless”. Esto quiere decir que se ha cambiado de un tipo de notificaciones que abrían una ventana adicional, y si bien el usuario podía cambiarse entre la ventana “modal” a otras aplicaciones, no era posible interactuar con la ventana original del navegador que la había creado. Las notificaciones “modeless” permiten que el usuario siga interactuando con la ventana original sin interrumpir la

navegación normal, lo que se considera una mejora al uso de la interface de usuario. Pero esto también representa una posibilidad para el ataque.

Por ejemplo, en Internet Explorer (en las versiones que ya utilizan intensivamente las notificaciones de tipo “modeless”, IE 9, 10 y 11), es posible crear una ventana de navegador tipo pop-under (usando una herramienta de evasión como jQuery), y en esta ventana comprometida, iniciar la descarga de un ejecutable que contiene un elemento de malware (por ejemplo virus y troyanos). El usuario posiblemente no se percate de esta ventana nueva en la que la descarga se está realizando. Cuando la descarga concluye, se lanza la notificación “modeless” y se fuerza a cambiar el foco a la ventana pop-under, de modo que las acciones posteriores del usuario no vayan dirigidas a la ventana original si no a la comprometida. En este momento se puede solicitar al usuario que presione una tecla como [Enter] o al barra espaciadora, y así iniciar la ejecución del malware.

### 3.4.4 Ataques contra la Privacidad

En los numerales anteriores, se ha tratado sobre todo ataques que causan problemas en contra de la seguridad, pero es necesario mencionar también que existen ataques específicos contra aspectos relacionados con la privacidad del usuario y del contenido.

En teoría, la mayoría de los navegadores modernos, implementan modos en lo cuales no se registran datos de la actividad del usuario, es decir, no se almacenan archivos temporales ni cookies y no se guarda el historial de la sesión de navegación una vez que la sesión concluye con el cierre del navegador (para nuestro caso particular hablamos del modo Incógnito de Chrome, del modo InPrivate de Internet Explorer y de la Navegación Privada de Firefox).

Pues aún con el uso de los modos privados de navegación, existen métodos y ataques que pueden reunir información sensible relativa a la privacidad del usuario por ejemplo:

**Ataques dirigidos a las cookies**

Aunque el usuario utilice navegación privada, o simplemente haya deshabilitado las cookies completamente o para ciertos sitios, se han creado métodos para tomar la información de una serie de tipos de almacenamiento utilizados por el navegador para generar lo que serían cookies casi indestructibles.

Uno de estos elementos es la API denominada Evercookie[19] que utiliza los siguientes tipos de almacenamiento para la creación de estas super cookies:

- HTTP cookies
- Flash cookies
- Almacenamiento de HTML5
- Almacenamiento de Silverlight

- Historial de navegación
- Información en caché

A la persistencia de estas cookies se le denomina “cookie respawning”

## 3.5 Ataques dirigidos a componentes específicos

### 3.5.1 Ataques directos al navegador (y motor de renderizado)
### Concepto de Fingerprinting

Si el atacante desea enfocar sus acciones dependiendo de ciertas vulnerabilidades conocidas, y así evitar perder tiempo lanzando ataques que no serán efectivos para un navegador, versión o plataforma específica, puede intentar determinar con un cierto grado de exactitud el entorno completo del navegador al cual desea comprometer.

A este proceso se le denomina fingerprinting e intenta como su nombre lo indica, identificar de la manera más cercana posible al navegador, su versión, el motor de renderizado que utiliza, en que plataforma se ejecuta e inclusive el nivel de parchado del mismo, para determinar cuáles ataques serán más efectivos dada la combinación de factores. El término fingerprinting también aplica para la detección de características únicas de la ubicación, historial y acciones de individuos que utilizan servicios web, pero en estos casos el ataque está más relacionado con invasiones a la privacidad, como el caso de las super cookies.

El proceso de identificación del navegador y sus capacidades mediante fingerprinting se puede llevar a cabo mediante una serie de técnicas que involucran al menos tres factores:

- Encabezados de HTTP
- Propiedades del DOM
- Elementos únicos del navegador

### Fingerprinting mediante encabezados de HTTP

Como parte de una solicitud web, un navegador siempre envía información al servidor de modo que al identificarse ante el servidor, pueda entregarse la información en una forma comprensible para el navegador o incluso enviar un mensaje de incompatibilidad, indicando que parte o todo el contenido no se visualizará o accederá de manera correcta (de modo que el usuario posiblemente tenga que utilizar otro navegador para el que si se haya incluido compatiblidad)

La información de estos encabezados de hecho es relativamente fácil obtenerla de servicios en línea que no son maliciosos y que nos permiten visualizar las diferencias entre estos encabezados, por ejemplo el servicio de "Echo" de Opera Soft[11]

Al analizar los encabezados enviados por el navegador al servidor, uno de los datos más importantes para determinar la "identidad" del navegador es el encabezado "User-Agent". Contiene información acerca del navegador en sí (Firefox, IE, Chrome), del sistema en el que se ejecuta (Windows, Linux, MacOs), si la plataforma es de 32/64 bits, del motor de renderizado utilizado, y de características especiales.

*User-Agent: Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.2; WOW64; Trident/6.0; Touch)*

Si bien este encabezado puede ofrecer mucha información útil (tanto para desarrolladores válidos como para atacantes), no es confiable en un 100% ya que puede ser modificado de forma válida, por ejemplo para implementar un modo de compatibilidad del navegador (e incluida como una función dentro del mismo), o falsificado con fines maliciosos, tan sencillo como instalar una extensión[12].

**Fingerprinting mediante propiedades de DOM**

Usando los encabezados del navegador se podría tener una idea bastante cercana del navegador que se atacará, pero si se desea más exactitud, inclusive en navegadores que tienen un largo historial de versiones mayores (Firefox y Chrome) y entre las cuales hay diferencias muy sutiles, el análisis de las propiedades específicas de DOM puede ser de mucha utilidad para los potenciales atacantes.

Existen listas de compatibilidad que informan sobre la existencia o falta de una propiedad específica de DOM dependiendo de la versión del navegador[13]. Pero no solo esto es útil, sino también los valores que pueden tomar estas propiedades. Por ejemplo, aunque se haya falsificado o cambiado el valor del encabezado "User-Agent" la propiedad "window.navigator.userAgent" aún conserva el valor real que permitirá identificar correctamente al navegador.

**Fingerprinting mediante características únicas del navegador**

Si aún después de verificar las propiedades de DOM específicas para un navegador y versión, se desea más exactitud, se pueden evaluar detalles como nombres de variables (son diferentes dependiendo del navegador aunque la funcionalidad es la misma) o características agregadas en cierta versión, por ejemplo características de visibilidad[14].

---

[11] El servicio permite la fácil visualización de encabezados intercambiados entre navegador y sitio web, accesible en: http://echo.opera.com

[12] Complemento para Chrome para poder cambiar a voluntad los encabezados enviados al servidor para identificar al navegador: User-Agent Switcher for Chrome, https://chrome.google.com/webstore/detail/user-agent-switcher-for-c/djflhoibgkdhkhhcedjiklpkjnoahfmg

[13] Compendio de todas las capacidades DOM por navegador y versión http://webbrowsercompatibility.com/dom/desktop/

[14] Comprobación de características de visibilidad de JavaScript por navegador y versión http://caniuse.com/#feat=pagevisibility

### 3.5.2 Ataques a HTTPS

La mayoría de usuarios consideran que si en su navegador aparece un ícono de un candado, el sitio y la comunicación es segura y prácticamente inviolable. Si bien es cierto que el ícono indica que los datos están siendo transmitidos usando HTTPS en lugar de HTTP, no significa que un atacante no pueda implementar un ataque que le permita acceder a información que debería estar siendo cifrada.

Los tipos de ataque detallados a continuación son variados y tratan de subvertir de diferentes maneras los elementos que fundamentan las comunicaciones seguras usando HTTPS.

**Cambio de HTTPS a HTTP**

En teoría, los datos cifrados no pueden ser visualizados en su forma de texto claro a menos que se disponga de la respectiva llave para descifrarlos. Debido a esta protección que resultaría difícil de romper por un atacante, se prefiere tratar de cambiar el tráfico entre el cliente y el servidor de modo que no se llegue a establecer el canal seguro y no haya necesidad de descifrar los datos.

Generalmente se usan dos técnicas para intentar que el usuario utilice una versión del sitio sin HTTPS.

El atacante puede interceptar el tráfico entre el navegador y el servidor (ayudándose de técnicas de Man-in-the-Middle) de modo que él se convierte en un punto de cifrado valido (una especie de proxy) entre ambas entidades. El ataque básicamente utiliza ARP Spoofing para obligar al navegador en el cliente a no contactar directamente al sitio, y cuando se recibe una respuesta del servidor que indica que debe hacerse el cambio de HTTP a HTTPS (regularmente una respuesta 302), el atacante si hace el cambio y continúa la comunicación mediante un canal seguro, pero la comunicación con el cliente sigue haciéndose únicamente con HTTP.

Este ataque es particularmente efectivo cuando en el servidor se han implementado defensas que no permiten que ciertos recursos sean servidos sobre conexiones inseguras o simplemente no existen en páginas sin HTTPS. El servidor en este caso entrega el contenido tal como ha sido programado, y el atacante confía que en el lado cliente, el usuario no note que el indicador del “candado” no apareció y por tanto sus datos están siendo enviados usando texto claro.

Ahora bien, si en el servidor existen versiones de las páginas y otros recursos que puedan ser entregados tanto por HTTPS como por HTTP, el atacante puede optar por cambiar los enlaces directamente dentro del navegador y así no debe implementar tecnologías de cifrado en su estación “proxy”.

Este tipo de ataque es efectivo como ya se mencionó, cuando el servidor cuenta con elementos no seguros (una versión no https del sitio) y además padece de vulnerabilidades tipo XSS, ya que debe permitir inyección de código (regularmente JavaScript) que cuando se ejecute en el navegador comprometido, cambie las etiquetas HREF de los enlaces de “https://” a simplemente “http://”

Una variante más compleja de este ataque puede ser implementada si hay porciones del sitio que no cuentan con un equivalente HTTP, y consiste en clonar el sitio original y cambiar los enlaces de modo que sean HTTP y en realidad redirijan a recursos del sitio del atacante.

En este como en los ataques anteriores, el atacante confía en que el usuario final no se dé cuenta de la falta del indicador de comunicación segura en su navegador y mantenga abierta la sesión.

**Ataques usando Certificados**

En caso que el atacante desee mantener la ilusión para el usuario final de estar comunicado mediante un canal seguro (obtener en su navegador el indicador de uso de HTTPS) puede optar por lanzar un ataque utilizando certificados falsos. Este tipo de ataque funciona tanto de forma conjunta con el ataque de Man-in-the-Middle implementado con ARP Spoofing y convirtiéndose en un proxy entre el servidor y el navegador, como al reescribir los enlaces originales y redirigir las comunicaciones hacia un sitio clonado que el controle.

En ambos casos, se ofrece al usuario final un certificado forjado por el mismo atacante, por tanto falso, pero que permite mostrar en el navegador el indicador de seguridad. En muchos casos, el navegador presentará un mensaje indicando que el certificado no es válido, y en este caso el atacante confía en que el usuario no leerá completa la advertencia y simplemente aceptará el certificado.

Las comunicaciones en este caso, en efecto están cifradas, pero el atacante no se preocupa de esto ya que al haber entregado su propio certificado al usuario final, se asegura de contar con la llave que le permite descifrar los datos en su equipo intermedio.

### 3.5.3 Ataques a JavaScript

En las secciones anteriores, uno de los recursos casi siempre presente en las técnicas de inicio y retención del control, así como en la fase de ataque como tal, es JavaScript. Muchos de los ataques lo utilizan como base para por ejemplo tomar ventaja de vulnerabilidades XSS.

Pero también existen ataques que van dirigidos a subvertir el funcionamiento de JavaScript de manera directa. Estos ataques se consideran de bajo nivel ya que para ser implementados debe afectarse por ejemplo la forma en que un script administra la memoria disponible para ejecución, como se reserva, utiliza y luego libera.

Los ataques más sencillos van orientados a simplemente hacer que el navegador deje de funcionar (crash), y de ser posible también afecte al mismo sistema operativo, ya que aunque la asignación de memoria del navegador se hace mediante un administrador propio (como el caso de Firefox con jemalloc), este administrador de memoria también se comunica con la tradicional función de asignación del SO (malloc) y dependiendo de cómo se implemente el ataque puede causar efectos no deseados en el sistema.

Los ataques más sofisticados intentan crear espacios de memoria de tamaños fijos, justo al lado de espacios utilizados (formando una especie de agujeros de ubicación bien conocida por el atacante en el heap de la memoria – heap spray[20]), aprovechando vulnerabilidades que causan que el administrador de memoria considere estos espacios como libres aunque en realidad aún mantienen datos (código ejecutable, imágenes y etiquetas html) que no son realmente eliminados hasta que se hace una llamada explicita al proceso de recolección de basura del administrador de memoria.

### 3.5.4 Ataques a extensiones en el navegador

Como se ha mencionado anteriormente en la sección 3.1 (Conceptos importantes), una extensión es un componente de software que de manera opcional (solo si está instalada en el navegador) agrega funcionalidad al mismo. Regularmente los equipos de desarrollo de los principales navegadores se concentran en el desarrollo del núcleo del aplicativo permitiendo que se asignen muchos recursos a mejorar sus características de seguridad, dejando muchas funcionalidades sin cubrir y a la espera que terceros las agreguen con sus propios desarrollos (tanto empresas como individuos).

Para funcionar, una extensión debe tener acceso a un conjunto de privilegios que le permitan interactuar con las páginas y el contenido de estas, y de este modo se convierte en un elemento útil para los usuarios. Pero este mismo conjunto de privilegios también las convierte en un punto vulnerable deseable para los atacantes.

Para lograr una mejor comprensión de como las extensiones de hecho incrementan la superficie de ataque del navegador, en las figuras 9 y 10 se muestra la arquitectura de las extensiones en los navegadores más populares (solamente se incluyen las arquitecturas de Google Chrome y Mozilla Firefox ya que cuentan entre ambos con un porcentaje muy superior al de IE en lo que respecta a las extensiones realmente utilizadas por los usuarios. En el caso de Internet Explorer, en su mayoría son extensiones desarrolladas por Microsoft con un modelo de seguridad e integración que reduce la posibilidad de ataques a estos componentes).

**Google Chrome**

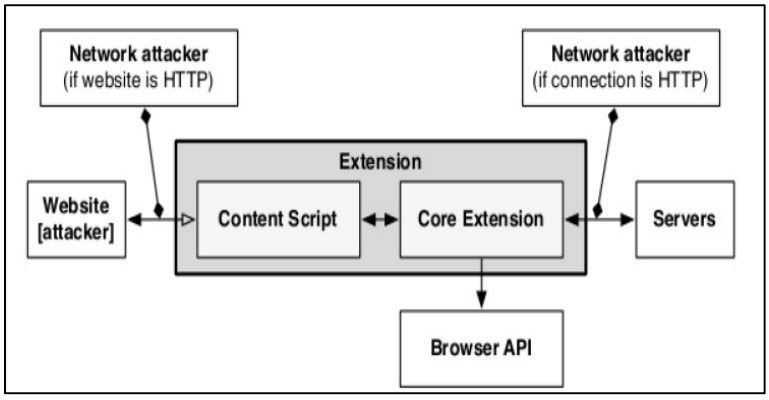


Fig. 9. Arquitectura y seguridad de extensiones en Google Chrome[21]

**Mozilla Firefox**

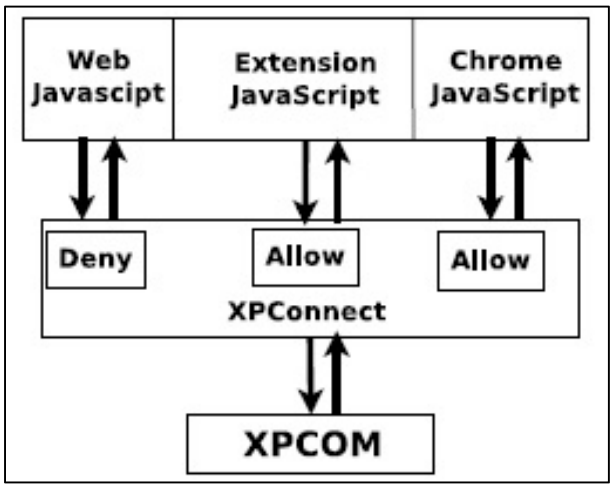


Fig. 10. Arquitectura y seguridad de extensiones en Mozilla Firefox[22]

**Fingerprinting relativo a las extensiones**

Como ya se presentó en el numeral 3.5.1 los procesos y técnicas de fingerprinting pueden ser de gran utilidad para los potenciales atacantes, ya que permiten obtener un conocimiento mucho más acertado de la verdadera superficie de ataque con la que se cuenta.

Lo mismo ocurre con las extensiones, que pueden verse expuestas a fingerprinting mediante los encabezados HTTP, a través de las propiedades de DOM, y además mediante el manifiesto (versión 1 o 2) de la extensión. El atacante en este caso debe construir su propia base de datos para identificar las extensiones, versión y capacidades, pero esto se ve facilitado por el hecho que tanto en Google Chrome como en Mozilla Firefox, las extensiones simplemente están comprimidas en forma ZIP (cada una con su extensión propia .xpi o .crx) lo que permite descomprimirlas con herramientas comunes (7zip por ejemplo) y analizar el código fuente con total libertad.

**UXSS**

Universal Cross-site Scripting (mencionado en el numeral 3.3.1 como una de las variantes de este tipo de ataques) puede ser potenciado en el caso de las extensiones de los navegadores web. Esto puede ocurrir cuando la extensión incluye en su archivo de manifiesto instrucciones muy liberales que permiten todos los orígenes como por ejemplo:

*"permissions": [*

*"<all_urls>"*

*]*

**Inyección y ejecución de comandos del Sistema Operativo**

Este tipo de ataque depende de los permisos y libertades concedidos dentro de la zona privilegiada “chrome://” respecto a la menos privilegiada zona de Internet en relación al navegador. Si la extensión acepta dentro de sus entradas la inclusión de código JavaScript por ejemplo, al ser procesado por la extensión y dentro de la zona con mayores privilegios se puede insertar comandos de sistema operativo que pueden resultar potencialmente perniciosos (en este punto es necesario recordar que algunas acciones dentro del SO también requerirán privilegios del sistema, así que dependiendo del tipo de cuenta con la que se está ejecutando el navegador, los daños al sistema pueden ser inclusive catastróficos.

### 3.5.5 Ataques a plugins en el navegador

Los plugins también representan un aumento a la superficie de ataque del navageador ya que pueden contener sus propias vulnerabilidades. Por lo regular los plugins son llamados mediante las etiquetas <embed> u <object> que al forzar la carga de un archivo, tras determinar el tipo MIME al que corresponde e inclusive pudiendo pasarle parámetros al objeto.

**Ataque a la funcionalidad “Click to Play”**

La características Click to Play está presente (con variadas denominaciones) en muchos de los navegadores modernos, y pretende agregar protección contra plugins maliciosos, o contra la ejecución de plugins válidos, pero que han sido detectados como vulnerables. Agrega la capacidad al navegador de notificar sobre la carga inminente del plugin y entonces el usuario puede decidir si efectivamente permite o no la ejecución.

A pesar de esta protección algunos plugins pueden simplemente saltar el requerimiento de aprobación del usuario o mantenerse ocultos y terminar ejecutándose.

En la mayoría de casos los ataques a Click to Play son implementados utilizando combinaciones de ingeniería social con ataques tipo “clickjacking”[23], de modo que cuando se solicita la ejecución de un plugin malicioso o desactualizado, se puede mostrar una ventana emergente que “oculte” o distraiga al usuario de la notificación verdadera del navegador, consultando si se desea ejecutar o no el plugin. Al hacer click en la ventana emergente comprometida, se está de hecho aceptando proceder a la ejecución.

## IV. MITIGACIÓN DE RIESGOS EN EL USO DE NAVEGADORES WEB

### 4.1. Mitigación a través de configuraciones propias del navegador

Como es lógico cada navegador cuenta con un método propio para la configuración de sus características y funcionalidades, entre ellas por supuesto, las configuraciones de seguridad y privacidad.

Por supuesto, es posible configurar estas opciones, banderas e interruptores (swtiches) a través de la interface de configuración del mismo navegador, pero por lo regular, estas interfaces han sido diseñadas para dar acceso a las opciones más comunes y no a la totalidad de configuraciones posibles.

Además, se pretende entregar al usuario final un navegador que ya cuente con las opciones configuradas con valores seguros, por lo que no es factible trabajar directamente con estas interfaces (ventanas, cuadros de diálogo y pestañas).

Por tanto, dependiendo del navegador, se configurarán las opciones durante el proceso de instalación de la misma aplicación, de manera transparente al usuario y sin la posibilidad de cambiar los valores durante el proceso de instalación.

Se ha definido trabajar en 6 áreas específicas de opciones de configuración, para los 3 navegadores, de manera que se puedan agrupar dichas opciones y de esta forma se hace más fácil posteriormente agregar configuraciones que se consideren necesarias o que hayan experimentado cambios en cada plataforma. Las áreas son:

- Configuraciones generales del navegador: página de inicio, capacidades y frecuencia de actualizaciones.
- Configuraciones de controles, extensiones y plugins: listas blancas y negras, obsolescencia, controles firmados y no firmados.
- Configuraciones de cifrado: validez y vigencia de certificados, métodos de autenticación y funciones de hashing.
- Configuraciones de Java, JavaScript y protecciones contra XSS: opciones para evitar la ejecución de código malicioso.
- Configuraciones de descargas: protección contra ejecutables y binarios potencialmente maliciosos que no solo afectarían al navegador si no al sistema operativo y otras aplicaciones (virus, troyanos, gusanos entre otros).
- Configuraciones de privacidad: opciones de cookies, geolocalización, y rastreo

La configuración de las opciones se realiza modificando directamente los valores en el registro de Windows o a través de archivos que contienen los valores que se aplicarán a cada opción. En las secciones siguientes se detalla el método de configuración para cada navegador en particular.

### 4.1.1. Internet Explorer 10/11

La configuración de las opciones de seguridad y privacidad de Internet Explorer en Windows 7 y Windows 8/8.1 se realiza modificando el registro de configuración del sistema operativo, el cual

es una base de datos jerárquica donde se almacenan los valores de configuración del mismo sistema, de los controladores de hardware, servicios, interface de usuario y aplicaciones de Microsoft y de terceros.

Lastimosamente Internet Explorer no cuenta con un método propio para verificar el estado de las configuraciones de seguridad y privacidad en su estado actual o por defecto, pero si se desea comprobar los valores usando la herramienta Regedit, se puede navegar hasta la llave de Políticas (policies) de la configuración del equipo local y exportar a un archivo ".reg" el contenido de dicha llave. Luego será necesario revisar las subllaves relacionadas (se mencionan a continuación).

Los valores de registro de Windows 7/8 a modificar corresponden principalmente a la llave:

• HKEY_LOCAL_MACHINE (HKLM)

Las subllaves específicas son:

•HKLM\Software\Policies\Microsoft\Windows\CurrentVersion\Internet Settings

•HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\Ext

•HKLM\Software\Policies\Microsoft\InternetExplorer\Restrictions

•HKLM\Software\Policies\Microsoft\InternetExplorer\Control Panel

•HKLM\Software\Policies\Microsoft\InternetExplorer\SQM

•HKLM\Software\Policies\Microsoft\InternetExplorer\Security

•HKLM\Software\Policies\Microsoft\InternetExplorer\Main\

•HKLM\Software\Policies\Microsoft\InternetExplorer\Download

•HKLM\Software\Policies\Microsoft\InternetExplorer\PhishingFilter

•HKLM\Software\Policies\Microsoft\InternetExplorer\Privacy

•HKLM\Software\Policies\Microsoft\InternetExplorer\Geolocation

•HKLM\Software\Policies\Microsoft\InternetExplorer\Suggested Sites

•HKLM\Software\Policies\Microsoft\InternetExplorer\DomainSuggestion

La configuración de los respectivos valores no se realizará directamente al registro (por ejemplo utilizando las herramientas Regedit[15] o Regedt32 disponibles en el sistema) si no, que se hará

[15] Más información sobre el uso del editor de registro y estructura del mismo en http://support.microsoft.com/kb/256986

de forma segura a través del manejo de políticas de grupo (GPO) con la herramienta Gpedit[16], aunque no de manera manual si no con comandos de Windows Powershell.

En el caso de Internet Explorer, debido a su alta integración con el sistema Windows, no es necesario instalar plantillas administrativas adicionales a través del editor de políticas de grupo. Las políticas de grupo a modificar con los comandos de Powershell apropiados están ubicadas en:

- Configuración del Equipo -> Plantillas Administrativas -> Componentes de Windows -> Internet Explorer

Los valores de configuración a modificar se encuentran en los apartados siguientes de la plantilla administrativa seleccionada:

- Directo en la configuración de Internet Explorer
- Aceleradores
- Características de seguridad
- Compatibilidad de aplicaciones
- Configuración de Internet
- Eliminar el historial de exploración
- Panel de control de internet
- Privacidad
- Vista de compatibilidad

Para un detalle completo de las opciones de seguridad y privacidad a configurar favor remitirse al Apéndice 1 “Plantilla de configuraciones de seguridad Internet Explorer 10/11”, el cual incluye los valores específicos a configurar en la columna de configuración segura.

### 4.1.2 Google Chrome

De la misma manera que el navegador propietario de Microsoft, en el caso de Google Chrome, la forma en la que se modificarán las opciones de seguridad y privacidad del navegador es a través de la interacción con el registro del sistema. Google Chrome si cuenta con un método de verificación propio de las políticas aplicadas, el cual puede ser accedido escribiendo directamente en la barra de direcciones del navegador “chrome://policy”. Como resultado se obtiene la lista de las políticas con valores establecidos de manera explícita. Para la mayoría

[16] El editor de políticas de grupo cargable a través de consolas MMC, más información al respecto en: https://technet.microsoft.com/en-us/library/cc758588(v=ws.10).aspx

de usuarios, esta lista debería aparecer vacía, indicando que no se ha establecido ningún valor para ninguna política en particular.

Para Google Chrome los valores del registro a modificar corresponden a la llave:

•HKEY_LOCAL_MACHINE (HKLM)

Las subllaves específicas son:

•HKLM\Software\Policies\Google\Chrome\

•HKLM\Software\Policies\Google\Chrome\ExtensionInstallBlacklist

•HKLM\Software\Policies\Google\Chrome\ExtensionInstallWhitelist

•HKLM\Software\Policies\Google\Chrome\URLBlacklist

•HKLM\Software\Policies\Google\Chrome\DisabledPlugins

•HKLM\Software\Policies\Google\Chrome\EnabledPlugins

•HKLM\Software\Policies\Google\Chrome\DefaultPluginsSetting

•HKLM\Software\Policies\Google\Chrome\CookiesSessionOnlyForUrls

Al ser Google Chrome un producto de terceros, no se dispone de una plantilla administrativa por defecto en el sistema para realizar las configuraciones directamente en el editor de políticas de grupo o utilizando comandos de Powershell. Por tanto, antes de iniciar la configuración es necesario descargar e instalar la plantilla apropiada.

La descarga se realiza directamente desde servidores de Google en la forma de un archivo comprimido denominado “policy_templates.zip”[17] que contiene las plantillas para el producto de software en los idiomas en que el navegador está disponible, para nuestro caso en particular en español (es). La instalación se completa a través del archivo “chrome.adm” y una vez que la plantilla está instalada puede procederse a realizar los cambios de las políticas con el método preferido (utilizando comandos de Powershell por ejemplo).

Las políticas de grupo a modificar se encuentran en:

- Configuración del Equipo -> Plantillas Administrativas -> Plantillas Administrativas Clásicas (ADM) -> Google -> Google Chrome

Los valores de configuración a modificar se encuentran en los apartados siguientes de la plantilla administrativa seleccionada:

- Directo en la configuración de Google Chrome

[17] Descarga oficial de plantillas para GPO de navegador Chrome: http://dl.google.com/dl/edgedl/chrome/policy/policy_templates.zip

- Administrador de contraseñas
- Configuración de contenido
- Configurar opciones de acceso remoto
- Extensiones
- Políticas de autenticación HTTP
- Proveedor de búsquedas predeterminadas
- Páginas de inicio
- Servidor Proxy

Para un detalle completo de las opciones de seguridad y privacidad a configurar favor remitirse al Apéndice 2 “Plantilla de configuraciones de seguridad Google Chrome” que incluye los valores del registro específicos a afectar en la columna de configuraciones seguras.

#### 4.1.3. Mozilla Firefox

El método de configuración del navegador Mozilla Firefox difiere radicalmente de los otros dos navegadores ya que será realizada a través de archivos de configuración colocados en directorios específicos de la aplicación y leídos en cada ejecución.

Los archivos utilizados para establecer los valores de configuración y sus respectivas ubicaciones son las siguientes:

• Archivo: “defaults.js”

Ubicado en el directorio “defaults/pref” del navegador. Utilizado para referenciar el nombre y ubicación del archivo de configuración que efectivamente contendrá las políticas de seguridad y privacidad cargadas en el navegador.

• Archivo: “mozilla.cfg”

Ubicado en el mismo directorio donde se encuentra el ejecutable de Firefox. Este archivo contiene una política definida en cada línea de configuración. El orden en que sean ingresadas las políticas no es relevante para su lectura y aplicación.

Firefox cuenta con un mecanismo propio para la verificación de las políticas de configuración y se accede ingresando directamente en la barra de direcciones del navegador “about:config”. A diferencia de Google Chrome que permite el acceso sin advertencias a esta modalidad (ya que solo se pueden verificar los valores pero no modificarlos), Firefox si advierte al usuario que la zona a la que se está ingresando (preferencias), permite el cambio de opciones que pueden

causar funcionamiento no deseado en el navegador, y por lo tanto debe ser utilizada con precaución.

También a diferencia de Chrome que no muestra por defecto las políticas no definidas de manera explícita, Firefox si muestra todas las opciones de configuración disponibles, resaltando en negritas cuales han sido definidas de forma explícita a través de los archivos de configuración.

Por supuesto, para que estos archivos no sean modificados por atacantes o inclusive de manera inadvertida y accidental por los mismos usuarios, es necesario definir mediante permisos de archivo que no pueden ser modificados por cuentas de usuario, si no solo por cuentas administrativas o una sola cuenta de administrador en particular.

Además cada opción configurada en el archivo mozilla.cfg, debe incluirse usando la modalidad "lockpref" de modo que los usuarios no puedan modificarlas desde el mismo navegador.

Para un detalle completo de las opciones de seguridad y privacidad a configurar favor remitirse al Apéndice 3 "Plantilla de configuraciones de seguridad Mozilla Firefox" que incluye los valores configurados y bloqueados en la columna de configuraciones seguras.

### 4.2 Mitigación mediante la inclusión de extensiones y plugins seguros y aprobados

La configuración de opciones directamente en los navegadores es una primera línea de defensa contra posibles ataques iniciados a través de código malicioso contenido en sitios potencialmente inseguros. Pero no es la única manera de ayudar a reducir la superficie de ataque al navegador.

También la inclusión de extensiones y plugins seguros y autorizados tiene un impacto positivo en la mejora de la seguridad y privacidad que los usuarios requieren.

A continuación se detalla la lista de extensiones y plugins (a las que en adelante se hará referencia de manera común como complementos) a incluir en los navegadores endurecidos (nótese que los agregados de software no están disponibles para todos los navegadores por lo que su inclusión depende de la disponibilidad para una determinada plataforma):

**Para todos los navegadores:**

**Web of Trust (WOT)[18]**

Con este complemento basado en valoraciones o ratings, que ayuda a prevenir a los usuarios de sitios maliciosos, sitios de phishing, sitios de fraudes en línea (scams) y contenidos similares. Cuando el usuario abre un sitio potencialmente peligroso WOT cubre la pantalla con una advertencia y espera a que el usuario decida si desea continuar en el sitio o lo cerrará.

[18] Url oficial https://www.mywot.com/en/download Para el caso de IE se ofrece un ejecutable. Para Chrome y Firefox se realiza la instalación usando sus respectivos repositorios de aplicaciones.

La advertencia es bastante obvia y explicativa de modo que el usuario conozca los riesgos a los que se expone si continua en navegado en el sitio detectado.

**AdBlock Plus**[19]

Este complento permite estar suscrito a una variedad de listas de filtrado que ayudan a bloquear contenido no deseado o hasta malicioso, y como su nombre lo indica tiene especial efectividad contra sitios de publicidad no deseada (Ads). La lista de filtros a la que un usuario puede suscribirse es muy extensa, por lo que si se seleccionan todas, se corre el peligro de ralentizar excesivamente la navegación. Por este motivo se incluirán las listas EasyPrivacy+EasyList y además Malware Domains únicamente.

**Para Google Chrome y Mozilla Firefox:**

**BitDefender Traffic Light**[20]

Complemento muy similar a WOT pero con listas negras muy extensas que pueden ayudar a ampliar el espectro de sitios maliciosos, de phishing y fradulentos que pueden ser accedidos por el usuario. El complemento bloquea la carga del sitio si se detecta este dentro de la lista negra.

**HTTPS Everywhere**[21]

Este complemento refuerza la protección contra sitios con contenido mixto, es decir, que si se dispone de la opción para cifrar la conexión, el complemento fuerza a que se cifre, y evita el contenido que se intenta servir a través de simple HTTP.

**Ghostery**[22]

El uso de este complemento mejora las funcionalidades de los navegadores que evitan que terceros como motores de búsqueda y agencias de anuncios y publicidad, rastreen las páginas que han sido visitadas utilizando estos navegadores.

**Solo para Google Chrome:**

**ScriptSafe**[23]

Complemento para bloquear la ejecución de código malicioso. Incluye la capacidad de creación y mantenimiento de listas blancas para agregar sitios considerados seguros y desde los cuales se desea ejecutar los scripts sin restricciones.

---

[19] Url oficial de descarga del complemento https://adblockplus.org/es/ Mismas condiciones de descarga e instalación que para WOT

[20] En esta página se encuentran las urls oficiales para los repositorios de software propios de cada navegador http://www.bitdefender.es/solutions/trafficlight.html

[21] En esta página se encuentran las urls oficiales para los repositorios de software propios de cada navegador https://www.eff.org/HTTPS-EVERYWHERE

[22] En esta página se encuentran las urls oficiales para los repositorios de software propios de cada navegador https://www.ghostery.com/es/

[23] Url oficial de instalación en Chrome Store https://chrome.google.com/webstore/detail/scriptsafe/oiigbmnaadbkfbmpbfijlflahbdbdgdf

**Vanilla**[24]

Complemento que mejora el bloqueo de cookies e inclusive permite la configuración de opciones mejoradas de auto destrucción de cookies no deseadas después de un tiempo definido por el usuario, evitando así la persistencia de estos elementos.

**Solo para Mozilla Firefox:**

**NoScript**[25]

Complemento sugerido para Firefox para bloquear la ejecución de código maliciosos en forma de scripts (similar a ScriptSafe).

**Cookie Monster**[26]

Complemento similar a Vanilla para Mozilla Firefox. Permite el bloqueo y destrucción de las cookies no deseadas, inclusive por pestaña.

**Secret Agent**[27]

Complemento que ayuda a evitar los procesos de fingerprinting. Una vez instalado, constantemente cambia el perfil del navegador de manera aleatoria evitando por ejemplo que después de hacer una búsqueda en un motor, se muestre publicidad no deseada relativa a esa búsqueda en un sitio de redes sociales.

## V. DESARROLLO DE APLICATIVO INTEGRADO CON LOS NAVEGADORES MÁS UTILIZADOS

### 5.1 Herramienta de desarrollo del instalador

La primera tarea en el desarrollo del instalador que integrara de manera coherente los tres navegadores seleccionados para este trabajo, en conjunto con sus opciones de seguridad y privacidad preconfiguradas y los complementos incluidos, fue encontrar una herramienta que permitiera la creación de un paquete de instalación que al ser iniciado por el usuario en los sistemas operativos objetivos permitiera la ejecución de las tareas detalladas a continuación:

- Inicio y ejecución del instalador en la modalidad de un asistente o wizard de instalación
- Informar al usuario final de las capacidades y condiciones de la instalación

[24] Url oficial de instalación en Chrome Store https://chrome.google.com/webstore/detail/vanilla-cookie-manager/gieohaicffldbmiilohhggbidhephnjj

[25] Url oficial de instalación en repositorio de Mozilla https://addons.mozilla.org/es/firefox/addon/noscript/

[26] Url oficial de instalación en repositorio de Mozilla https://addons.mozilla.org/es/firefox/addon/cookie-monster/

[27] Página de terceros con vínculo al complemento ttps://www.dephormation.org.uk/index.php?page=81

- Advertir al usuario final de las acciones que debería tomar previo a la instalación y si no las ha realizado cancelar dicho proceso para reiniciarlo cuando se cumplan las sugerencias y requisitos

- Permitir la detección de la versión y particularidades del sistema operativo

- Permitir la detección de versiones previas (probablemente comprometidas) de algunos de los navegadores a instalar y consultar que acciones se tomarán con estas instancias previas

- Desempaquetar los componentes requeridos para cada navegador en una ubicación temporal.

- Aplicar las rutinas de instalación requeridas, incluyendo aplicación de opciones e instalación de complementos. Crear los accesos necesarios a las aplicaciones instaladas si es necesario.

- Finalización del proceso de instalación y limpieza de elementos temporales.

Con base en estos requerimientos se evaluaron tres posibles herramientas de desarrollo para el instalador deseado.

### 5.1.1 Tipo de instalación de la solución

Se consideraron dos tipos de instalación: MSI (Microsoft Installer) y un Instalador No-MSI (Setup.exe).

Brevemente se mencionan las características y ventajas de cada tipo de instalación disponible para sistemas Windows:

**Instalador MSI**

Este tipo de instalación usa un archivo principal (.msi) que contiene una base de datos legible por el servicio Windows Installer. Las instrucciones almacenadas en la base de datos, son seguidas por el servicio para realizar operaciones tales como copia de archivos, establecimiento de valores de registro y funcionalidades similares.

La instalación tipo MSI, devuelve las funcionalidades de instalación al sistema operativo y en algunos casos facilita el despliegue al simplificar por ejemplo los procesos que tienen relación con una variedad de librerías del sistema (por ejemplo .dll).

Al basar su funcionamiento en el servicio Windows Installer, los paquetes MSI cuentan con:

- Manejo de versiones

- Manejo de recursos: archivos y llaves de registro

- Manejo de componentes: librerías por ejemplo

- Manejo de características: opciones de configuración

- Manejo de estados de los productos: instalado, ejecución desde la fuente, instalación bajo demanda o al primer uso, y no instalado

De hecho un instalador tipo MSI podría simplemente ser un conjunto de instrucciones de configuración que altere el funcionamiento de una aplicación preinstalada sin aportar recursos nuevos. En caso de necesitar archivos adicionales regularmente estos se proporcionan en contenedores tipo Windows Cabinet (archivos .cab)

**Instalador Setup.exe**

Una instalación que utiliza un paquete tipo Setup.exe, por lo regular extrae los recursos de instalación de sí mismo, creando una estructura de directorios temporal utilizada en el momento de la ejecución del instalador. Un instalador tipo Setup.exe puede contener a su vez un instalador tipo MSI y hacer una llamada al servicio Windows Installer si es necesario.

Este tipo de instalación permite la instalación del producto principal, y además la adición de logos, plantillas y complementos que serán aplicados a la instalación central.

Mientras que Windows Installer solo permite un paquete tipo MSI ejecutándose en un momento determinado, el cual debe concluir sus procesos antes de iniciar otra ejecución. Un paquete tipo Setup.exe puede por ejemplo ejecutar otros paquetes (por ejemplo instalar dependencias o ejecutables individuales) al mismo tiempo, incluyendo paquetes MSI en secuencia.

En general, los paquetes tipo Setup.exe ofrecen mayor flexibilidad que su contraparte MSI, ya que estos últimos cuentan con reglas específicas de cómo administrar las instalaciones, actualizaciones y la desinstalación de productos. Al usar un Setup.exe se tiene un mayor control, se pueden hacer llamados a funciones del sistema operativo, utilizar ejecutables desarrollados en lenguajes como C++ por ejemplo, para ampliar las capacidades del instalador.

**Elección del tipo de instalación**

Se utilizará para el desarrollo del instalador producto de este trabajo, un paquete tipo Setup.exe, ya que como se mencionó anteriormente, esta modalidad de instaladores permite un control total sobre las necesidades de instalación requeridas, no dependiendo de manera tan estricta de las capacidades de un servicio ya aportado por el sistema, pero que puede aprovechar las capacidades del sistema operativo para realizar tareas de configuración de las opciones instaladas (por ejemplo la ejecución de comandos de Powershell)

### 5.1.2 Alternativas de desarrollo evaluadas

Se inició la evaluación de alternativas de desarrollo paralelamente a la determinación del tipo de instalación necesaria, por lo que en las alternativas aparece inclusive una opción dedicada a la creación de instalaciones tipo MSI. Una vez definido el tipo de instalación requerido para el

despliegue de los navegadores endurecidos, incluyendo sus opciones de seguridad y privacidad y los complementos seguros, la elección se redujo a dos alternativas.

Las tres alternativas evaluadas inicialmente (una gratuita y dos con costos por su utilización) se detallan a continuación:

**WiX**[28]

Windows Installer XML Toolset, es una herramienta que permite a desarrolladores crear instaladores basados en el servicio Windows Installer. Esta herramienta está disponible a través del tipo de licencia MS-RL (Microsoft Reciprocal License), sin costo, pero haciendo énfasis en que esta licencia no se traslada a los componentes instalados con un paquete creado con WiX.

Tiene la capacidad de crear paquetes de instalación (.msi), módulos de mezclado de componentes (.msm) y paquetes de actualización o parches (.msp). La herramienta cuenta con sus propias opciones de línea de comando para actuar de manera autónoma, o puede integrarse con entornos de desarrollo más completos como MSBuild, Team Build o Visual Studio.

Dentro de sus capacidades extendidas están las funcionalidades de instalación de sitios IIS, creación de bases de datos SQL Server, registro de excepciones en Microsoft Firewall, entre otras.

Si bien es cierto, el objetivo de WiX es la creación de paquetes tipo MSI, también cuenta con funcionalidades básicas para la creación de wrappers o bootstrapers que permiten crear un instalador contenido en un archivo .exe, pudiendo inclusive usar código C# o C++ en el paquete, pero que siempre debe cumplir con los requisitos y limitaciones de Windows Installer.

**Caphyon Advanced Installer**[29]

Esta herramienta de desarrollo de instaladores ayuda a simplificar la construcción de instaladores tipo MSI y No-MSI (setup.exe), a través de una interface de usuario de alto nivel que libera al desarrollador de las complicaciones de la tecnología en la que se basan los instaladores. La licencia tipo Architec para un usuario tiene un costo de $2999.00 con seis meses de soporte incluidos. Se probó usando una versión Free Trial de 30 días (período de prueba entre el 1 y el 30 de diciembre de 2014 para la versión 11.6.3 del producto).

Los proyectos creados con esta herramienta se almacenan como archivos XML, lo que facilita por ejemplo manejar un sistema de control de versiones de los instaladores creados. Cuenta con una interface de línea de comando propia y puede integrarse de ser necesario con Make o Ant.

Con Advanced Installer se pueden incluir en el proyecto de manera sencilla archivos individuales, carpetas completas, accesos directos, llaves y valores de registro de Windows,

[28] Descarga oficial de herramienta WIX en: http://wixtoolset.org

[29] Descarga oficial de herramienta Advanced Instaler en su modalidad trial http://www.advancedinstaller.com/

variables de entorno nuevas o agregadas a variables ya existentes, modificación de archivos .INI, controladores tipo ODBC, y JDBC.

Si es necesario se pueden mezclar componentes preexistentes o reutilizar desarrollos previos dependientes de .NET, con interacción con servicios del sistema, se pueden crear asociaciones de tipos de archivos a aplicaciones predeterminadas y definición de tipos MIME. Además se pueden alterar permisos y derechos de archivos y carpetas.

Los instaladores creados con esta herramienta pueden ser firmados digitalmente para garantizar a los usuarios finales que el paquete proviene de una fuente segura y que no ha sido alterado o manipulado, y evitar sospechas de software malicioso.

La funcionalidad de la propia herramienta puede ser extendida a través de código C, C++, VBScript o JavaScript.

Lastimosamente no cuenta con un lenguaje de script propio que permita tener un control total del instalador desarrollado desde la propia herramienta, si no que se confía esta tarea a código externo.

**InstallShield**[30]

Flexera Software InstallShield Premier, es una suite de software para desarrollo de instaladores que lleva muchos años en el mercado y se constituye en una de las mejores opciones. Tiene la capacidad de generar instaladores tipo MSI por supuesto, pero su mayor ventaja reside en la creación de complejos instaladores tipo Setup.exe. La licencia de la versión Premier en modo locked station tiene un costo de $4799.00 y se cobra extra por contratos de soporte. Esta herramienta se probó utilizando una versión Free Trial de 21 días (período de prueba entre el 8 y el 29 de diciembre de 2014 para la versión 2014 del producto).

Simplifica las tareas de creación de instaladores básicamente para todas las plataformas de escritorio posibles, instaladores stand-alone, despliegue de aplicaciones web, cliente-servidor e inclusive aplicaciones virtualizadas y listas para la nube. En los sistemas operativos soportados, puede instalar automáticamente roles y características que serán necesarios para la ejecución de la aplicación instalada (en Windows Server 2008/2012 por ejemplo). Ejecuta scripts de Powershell para instalaciones complejas que necesitan mucha interacción con el sistema operativo.

Proporciona extensas capacidades de reportería y logs que permiten analizar el estado de las instalaciones y así determinar la existencia de posibles errores en el paquete o de prerrequisitos que no se han tomado en cuenta e incluirlos en versiones futuras del paquete.

Permite la creación de instaladores exclusivos para sistemas de 64 bits, lo cual es beneficioso cuando en un sistema se ha desactivado WoW64. Se integra con la suite de desarrollo Visual Studio 2013, permite la firma digital de los instaladores para generar confianza en el usuario final y ayuda en la generación de instalaciones multicapa.

[30] Descarga official de las versiones de Installshield como parte de Flexera Software
http://www.flexerasoftware.com/producer/products/software-installation/installshield-software-installer/

Además Installshield, facilita la instalación de actualizaciones ya sea auto contenidas o a través de repositorios en red para la descarga de componentes nuevos o modificados desde instalaciones anteriores. Si es necesario, el tamaño del ejecutable que los usuarios finales descargan se puede dejar al mínimo y usar un modo de instalación web para completar las descargas y tareas de instalación.

Una de las características más importantes de esta suite es la disponibilidad de un lenguaje de script propietario denominado InstallScript. Con este lenguaje se puede controlar toda la lógica y comportamiento del proyecto de instalación, obteniendo así un grado de flexibilidad, control e integración con el sistema operativo detectado superior a las otras soluciones probadas.

Debido a la extensa gama de características y funcionalidad de esta suite, se ha elegido para el desarrollo del instalador que contendrá los tres navegadores considerados en el estudio, sus archivos de configuración preestablecidos, complementos seguros, y los prerrequisitos en caso que sea necesario actualizar el sistema para permitir la instalación (en Windows 7 por ejemplo). Y como ventaja adicional, InstallShield también cuenta con una versión compatible con Linux lo que facilitaría la creación de un instalador similar para sistemas operativos no-Windows a futuro.

### 5.1.3 Lenguaje propietario para el desarrollo del instalador

Como se mencionó en el apartado anterior, una de las características más ventajosas de la suite de desarrollo de instaladores seleccionada es su lenguaje de programación InstallScript, ya que permite el control total y granular de los procesos de instalación.

Un proceso de instalación es una colección o conjunto de eventos, manejadores o handlers, llamadas a funciones, y datos usados por los eventos y funciones. Todos estos elementos pueden expresarse en términos del lenguaje InstallScript, que es bastante simple, con una curva de aprendizaje muy suave, pero poderoso si se usa para el objetivo correcto: crear instaladores.

La estructura del lenguaje es similar a la del lenguaje C, con un formato y sintaxis bien definidas. Implementa ciertos tipos de datos, con sus respectivas propiedades, lo que permite la creación de funciones personalizadas.

**Características importantes del lenguaje InstallScript**[24]

- Palabras reservadas del lenguaje: son básicamente los comandos utilizados para formar la estructura y control del script (ej. abort, exit, export, set, return)

- Constantes predefinidas: identificadores reservados del lenguaje que son utilizados para acceder a valores literales y que no pueden ser alterados manualmente si no que son resultado de una o varias funciones (ej. askpath, backbutton)

- Variables predefinidas: variables utilizadas en tiempo de compilación del script (ej. _FILE_, _LINE_)

- Tipos de datos: tipos predefinidos con los que se puede definir una variable (ej. binary, bool, char, int, list)

- Directivas de preprocesador: Inician con "#" y son instrucciones ejecutadas al momento de compilar el script y que pueden instruir al compilador a incluir otros archivos de script o inclusive a detener la compilación (ej. #define, #error)

- Control de flujo: InstallScript cuenta con estructuras de control que pueden ser utilizadas para agregar capacidades de decisión e iteración al script (ej. for..endfor, repeat..until)

- Manejadores de eventos: Estados y funciones bien definidas en determinados momentos de la ejecución de la instalación. Pueden ser globales, por componente, misceláneos o avanzados (ej. OnBegin, OnEnd)

- Funciones: InstallScript incluye funciones predefinidas del lenguaje, funciones creadas por el usuario e inclusive funciones residentes en elementos externos tales como archivos DLL.

- Operadores: símbolos predefinidos que realizan una determinada acción sobre uno o más operandos (ej. +, &, =)

- Objetos: Nombres de los objetos y manejadores de dichos objetos disponibles en InstallScript (ej. Err, Reboot)

### 5.2 Componentes de software incluidos en el instalador

A continuación se incluye el listado y detalles pertinentes de los componentes de software incluidos en el instalador integrado de navegadores endurecidos con complementos preinstalados:

**Microsoft Internet Explorer**[31]

- Instalador de Internet Explorer 11 en español 32 bits (IE11-Windows6.1-x86-es-es.exe, 30.5 Mbytes)

- Instalador de Internet Explorer 11 en español 64 bits (IE11-Windows6.1-x64-es-es.exe, 55.5 Mbytes)

- Actualizaciones necesarias para la instalación de IE11 en sistemas x86 y x64 de Windows 7 en español (44 Mbytes para 32 bits, 75.2 Mbytes para 64 bits)

- Windows6.1-KB2533623, corrige problemas relacionados a ejecución de código remotamente a través de la carga incorrecta de librerías DLL

- Windows6.1-KB2670838, mejoras y correcciones en el tratamiento de gráficos a través de los componentes Direct2D, Direct3D, DirectWrite, H.264, entre otros

[31] Descarga oficial de instalador / actualización de Internet Explorer 11: http://www.microsoft.com/es-es/download/internet-explorer-11-for-windows-7-details.aspx

- Windows6.1-KB2729094-v2, actualización de la fuente de sistema Segoe UI Symbol
- Windows6.1-KB2731771, corrige problema y agrega nuevas APIs relacionadas a la conversión entre el tiempo local y su equivalente en UTC
- Windows6.1-KB2786081, corrección al problema de IE10 que no permite almacenar credenciales de un sitio al cerrar sesión de Windows o reiniciar completamente el sistema
- Windows6.1-KB2834140-v2, corrección al error "0x0000050" al tener instalada la actualización KB2670838
- IE11-Windows6.1-KB3008923, corrige errores en la plantilla administrativa de GPO de IE luego de actualizar desde versiones 8, 9 o 10
- Instalador de complemento WOT para IE en español (WOT-latest-es-x64.msi, 1.7 Mbytes)
- Instalador de complemento AdBlock Plus para IE (adblockplusie-1.3, 5.7 Mbytes)

**Google Chrome**

• Versión de Google Chrome 38.0.2125.104 portable[32] (directorio completo con complementos preinstalados, aproximadamente 200 Mbytes)

• Plantilla administrativa de políticas de grupo para Google Chrome (chrome.admx, 207 Kbytes y archivo específico de lenguaje chrome.adml (251 Kbytes)

**Mozilla Firefox**

• Versión de Mozilla Firefox 33.0.2 portable[33] (directorio completo con complementos preinstalados y configuraciones incluidas, aproximadamente 100 Mbytes)

• Archivos de configuración defaults.js y mozilla.cfg (menos de 10 Kbytes)

### 5.2.1 Condiciones especiales y requerimientos derivados del sistema operativo

**Si el sistema operativo es Microsoft Windows 8.1**

• En Windows 8.1 el navegador predeterminado de Microsoft ya es Internet Explorer 11 por lo que no se ofrecerá la opción para instalar este navegador

[32] Sitio de descarga de Chrome Portable http://portableapps.com/apps/internet/google_chrome_portable
[33] Sitio de descarga de Firefox Portable http://portableapps.com/apps/internet/firefox_portable

- Se ofrecerá la opción de aplicar la configuración segura incluida en el instalador y los complementos WOT y Adblock Plus (el usuario puede decidir si aplica la configuración segura y los complementos)

- En este sistema operativo IE11 no puede ser desinstalado, de modo que sí el usuario desea removerlo del sistema, se ofrecerá la opción de desactivar la característica de sistema que lo habilita (accesible a través del Panel de Control en la configuración de Programas y Características). Es necesario notar que aunque el navegador puede ser "desactivado" algunas aplicaciones de Microsoft o de terceros podrán seguir utilizando el motor de navegación en caso que dependan de él aun cuando como característica ya no esté disponible.

**Si el sistema operativo es Microsoft Windows 8**

- En Microsoft Windows 8, no se dispone de compatibilidad con Internet Explorer 11 (el sistema incluye la versión IE10 y no puede ser actualizada individualmente)

- Al momento de la instalación, si se detecta este sistema se ofrecerá al usuario aplicar las configuraciones de seguridad y privacidad y los complementos sobre IE10 y se le informará que la única manera de poder utilizar IE11 es a través de la actualización del sistema completo a Windows 8.1. Las configuraciones y los complementos mantendrán su validez si el usuario decide hacer esta instalación en algún momento posterior a la instalación de los navegadores endurecidos.

**Si el sistema operativo es Microsoft Windows 7**

- Para Internet Explorer 11 se requiere que el sistema operativo haya sido actualizado al menos con el correspondiente Service Pack 1. En caso que este requisito no se cumpla, se informará al usuario que es crítico realizar dicha actualización y deberá hacerla por cuenta propia. Las opciones en este punto serán: abortar la instalación de los navegadores o continuar sin incluir Internet Explorer. Si se opta por esta última alternativa no se aplicarán configuraciones de seguridad y privacidad y no se instalarán complementos para este navegador.

- Si el requisito de Service Pack 1 se cumple, se detectará la arquitectura del sistema (x86 o x64) y se aplicará la instalación de las actualizaciones necesarias, y del propio navegador correspondiente a la arquitectura detectada.

**Para todos los sistemas operativos considerados en el trabajo**

- Las ediciones de Google Chrome y Mozilla Firefox a instalar son de 32 bits

- Las versiones de Google Chrome y Mozilla Firefox no son las últimas estables disponibles. Estas podrán ser actualizadas una vez instaladas por el usuario a través de los mecanismos de actualización propios de cada navegador, y además dentro de las configuraciones seguras se garantiza que en un período adecuado se detectará la existencia de versiones actualizadas y se ofrecerá al usuario su respectiva instalación para mantener al día estos componentes de software.

- En versiones posteriores (actualizadas) del instalador integrado se irán incluyendo las últimas versiones estables de los tres navegadores. Se planea al menos 3 actualizaciones anuales (cada cuatrimestre como mínimo). La creación de una nueva versión en caso de cambios críticos que afecten las características de seguridad o en caso de detección de fallos considerados críticos en alguno de los navegadores puede dar paso a una actualización menor del instalador fuera de ese período definido.

- La instalación de los navegadores no-Microsoft se hará por usuario, esto facilitará por ejemplo, contar con navegadores protegidos y confiables en caso que para un usuario determinado se hayan producido problemas de seguridad que comprometan a los navegadores, dichos problemas no afectarán las instancias de los navegadores para otros usuarios en el mismo equipo.

### 5.2.2 Capacidades y ejecución del instalador integrado

En esta sección se incluyen los detalles de ejecución, opciones modificables por parte del usuario y resultado del proceso.

**Método de despliegue**

El instalador estará disponible en la modalidad de un único archivo “Setup.exe” de aproximadamente 600 Mbytes, y puede ser colocado en uno de los repositorios en línea con los que cuenta la Universidad Don Bosco.

Debido al tamaño del archivo y para conservar el ancho de banda que implicaría la descarga desde los servidores de la Universidad (ya sea desde su sitio principal, o desde alguno de los servidores que mantienen otros servicios, será colocado en sendas ubicaciones en la nube aprovechando los dos servicios de almacenamiento con que cuenta la institución en la actualidad: Microsoft Office 365, y Google Drive, en los cuales es posible colocar archivos individuales de hasta 5 Gbytes. Se distribuirá el enlace de descarga a los miembros de la comunidad universitaria y también estará disponible al público en general que desee probarlo e instalar los navegadores endurecidos.

**Etapas de la ejecución**

a) Se inicia la ejecución haciendo doble clic sobre el instalador descargado

b) En el primer cuadro de diálogo del instalador se informa al usuario sobre los navegadores a instalar y se le recuerda que en caso de optar por la desinstalación de los navegadores actuales del sistema, debería realizarse una copia de respaldo de los favoritos o marcadores en cada uno ya que esa función no estará disponible como parte del instalador, de modo que el usuario pueda restaurarlos posteriormente. En este punto se puede abortar la instalación en caso que se decida realizar la copia, o simplemente continuar.

c) Si el usuario decide continuar la ejecución, el instalador realiza la detección de navegadores previamente instalados (válido para Google Chrome y Mozilla Firefox), y se ofrece la opción de removerlos del sistema. La opción no está disponible para Internet Explorer debido a su alta integración con el sistema, si se tiene por ejemplo Internet Explorer 9, se degradaría a Internet Explorer 8, y si este último estuviera funcionando, no puede ser eliminado (para Windows 7). Lo mismo ocurre con IE10 en Windows 8. Si se marcan las opciones se procederá a la desinstalación de Firefox y/o Chrome (no se puede revertir estas opciones una vez ejecutadas) utilizando los siguientes métodos:

**Desinstalación de Firefox**

Usando comandos de sistema operativo se detecta si Firefox está instalado (IF EXIST). En caso afirmativo se utiliza el siguiente comando para la desinstalación:

*"%ProgramFiles(x86)%\Mozilla Firefox\uninstall\helper.exe" -ms*

Una ventaja de Firefox es que independientemente de la versión se encuentra instalado en la misma ubicación y dispone del ejecutable helper.exe que al ser invocado con las opciones –ms realiza una desinstalación completa a nivel de sistema del navegador.

**Desinstalación de Chrome**

Utilizando la misma capacidad de comprobación de la existencia del directorio del navegador con IF EXIST, se determinará primero la versión específica del navegador (solo versiones estables, no beta). Una vez determinada a ubicación exacta se ejecuta el comando apropiado:

*"%ProgramFiles(x86)%\Google\Chrome\Application\39.0.2171.95\Installer\setup.exe" --uninstall --force-uninstall -system-level*

El ejemplo anterior aplica específicamente para la versión 39.0.2171.95

Las versiones que el instalador tendrá la capacidad de remover, cubren al menos un período de tres años hacia atrás, considerando que alguna de esas versiones se encuentra en las computadoras modernas. Inclusive, si se hubiera instalado una versión más antigua que las consideradas, con seguridad se debería haber actualizado hace un tiempo.

Las versiones específicas que se podrán remover van desde la 21.0.1180.75 hasta la 39.0.2171.95. Versiones posteriores a esta última podrán ser removidas por implementación actualizadas del instalador integrado.

d) Una vez se haya completado la remoción de las versiones antiguas se procederá a la instalación de las versiones actualizadas y seguras. Se inicia con la copia de las carpetas completas de las versiones portables de Google Chrome y Mozilla Firefox, que ya incluyen los complementos que mejoran la seguridad y privacidad. Se instalan primero estos navegadores ya que este proceso no requiere reiniciar el sistema. Como se ha mencionado anteriormente en el caso de Firefox los archivos de configuración con las opciones pre configuradas también son copiados en este momento. Para Internet Explorer, si no se cumple con el requisito de Service Pack 1 de Windows 7 no se instala el navegador en versión 11.

e) De manera transparente para el usuario, se procede a la instalación de las actualizaciones post SP1 de Windows 7 que sean necesarias (según arquitectura detectada) y luego a la actualización del navegador en si. Si es necesario, el instalador notificará que es necesario reiniciar el sistema para continuar.

f) Una vez reiniciado el sistema (si fuera necesario), se procede a la ejecución de los comandos PowerShell que configuran las opciones de seguridad y privacidad para los navegadores Google Chrome e Internet Explorer 10/11, cambiando los valores de las respectivas plantillas administrativas de las políticas de grupo.

g) Para finalizar se crean los íconos con las etiquetas “Google Chrome Seguro”, “Mozilla Firefox Seguro” e “Internet Explorer Seguro” en el escritorio del usuario.

h) El instalador realiza la remoción de archivos temporales utilizados en el proceso y se informa al usuario que se ha concluido la instalación.

**CONCLUSIONES**

En implementaciones de seguridad centralizadas y con políticas de cumplimiento obligatorio, los administradores pueden forzar el uso de navegadores que incluyan características de seguridad pre configuradas y garantizar así un funcionamiento confiable de los mismos. En equipos personales y no administrados de forma centralizada esta tarea se vuelve mucho más difícil, y depende del usuario final la decisión de aplicar las configuraciones de seguridad. La cantidad y complejidad de las configuraciones se convierte en un motivo para dejar los temas de seguridad y privacidad en un segundo plano, exponiéndolo a una variedad de riesgos. La solución propuesta libera al usuario final de los detalles técnicos y le provee herramientas de trabajo confiables y que minimizan dichos riesgos.

La instalación de los navegadores sanitizados y pre configurados mejora la experiencia del usuario en cuanto a seguridad y privacidad, ayudando a minimizar ataques y problemas que incluyen el bajo rendimiento en la navegación, pérdida o robo de datos personales y confidenciales, suplantación de identidad, fraude electrónico en sistemas sensibles como la banca, e inclusive problemas que afecten a otras aplicaciones y al mismo sistema operativo. Ahora bien, la seguridad no es total y permanente y el grado de esta depende de que el usuario no deshabilite las opciones pre configuradas, no instale complementos de dudosa reputación o claramente maliciosos, y por supuesto que no utilice otros navegadores que no cuentan con la configuración apropiada. Por lo tanto es importante que el uso de estas herramientas se vea acompañado de campañas que concienticen al usuario sobre los riesgos que corre al navegar usando software poco seguro.

La solución presentada puede ser modificada para ampliar su funcionalidad y acomodarse a las preferencias de otros usuarios. Por ejemplo, en una versión posterior, y tras completar el estudio de los métodos y valores de configuración de navegadores como Safari y Opera, estos podrían ser agregados al instalador integrado y ofrecerse como alternativas al usuario.

## REFERENCIAS


[1] Grosskurth, A., & Godfrey, M. W. (2006). Architecture and evolution of the modern web browser. Preprint submitted to Elsevier Science.

[2] Wagner, G., Gal, A., Wimmer, C., Eich, B., & Franz, M. (2011, June). Compartmental memory management in a modern web browser. In ACM SIGPLAN Notices (Vol. 46, No. 11, pp. 119-128). ACM.

[3] Bott, E. (2013). Introducing Windows 8.1 for IT Professionals. Microsoft Press.

[4] Oshri, I., de Vries, H. J., & de Vries, H. (2010). The rise of Firefox in the web browser industry: The role of open source in setting standards. Business History, 52(5), 834-856.

[5] Baysal, O., Davis, I., & Godfrey, M. W. (2011, May). A tale of two browsers. In Proceedings of the 8th Working Conference on Mining Software Repositories (pp. 238-241). ACM.

[6] Crowley, M. (2010). Internet Explorer Architecture. In Pro Internet Explorer 8 & 9 Development (pp. 1-37). Apress.

[7] Yeow, C. C. (2005). Firefox secrets. SitePoint Pty Ltd.

[8] Teixeira, J., & Lin, T. (2014, May). Collaboration in the open-source arena: the webkit case. In Proceedings of the 52nd ACM conference on Computers and people research (pp. 121-129). ACM.

[9] David, M. (2013). HTML5: designing rich Internet applications. Taylor & Francis.

[10] Ullman, L. (2012). Modern JavaScript: Develop and Design. Peachpit Press.

[11] McFarland, D. S. (2012). CSS3: The Missing Manual. " O'Reilly Media, Inc.".

[12] Mitnick, K. D., & Simon, W. L. (2001). The art of deception: Controlling the human element of security. John Wiley & Sons.

[13] Mieres, J. (2009). Ataques informáticos. Debilidades de seguridad comúnmente explotadas). Recuperado http://proton.ucting.udg. mx/tutorial/hackers/hacking. pdf.

[14] Rosenthal, D. S. (2010). Format obsolescence: assessing the threat and the defenses. Library hi tech, 28(2), 195-210.

[15] Villeneuve, N. (2011). Trends in targeted attacks. Trend Micro.

[16] Curran, K., & Dougan, T. (2012). Man in the browser attacks. International Journal of Ambient Computing and Intelligence, 4(1), 29-39.

[17] Xu, W., Zhang, F., & Zhu, S. (2012, October). The power of obfuscation techniques in malicious JavaScript code: A measurement study. In Malicious and Unwanted Software (MALWARE), 2012 7th International Conference on (pp. 9-16). IEEE.

[18] Hashemi, F., & Sadat, H. (2014). A Hybrid Approach to Detect Tabnabbing Attacks.

[19] Sprankel, S. (2011). Online Tracking, Targeted Advertising and User Privacy-The Technical Part.

[20] Ding, Y., Wei, T., Wang, T., Liang, Z., & Zou, W. (2010, December). Heap taichi: exploiting memory allocation granularity in heap-spraying attacks. In Proceedings of the 26th Annual Computer Security Applications Conference (pp. 327-336). ACM.

[21] Rana, A., & Nagda, R. (2014). A Security Analysis Of Browser Extensions. arXiv preprint arXiv:1403.3235.

[22] Saini, A., Gaur, M. S., & Laxmi, V. (2013, November). The darker side of Firefox extension. In Proceedings of the 6th International Conference on Security of Information and Networks (pp. 316-320). ACM.

[23] Huang, L. S., Moshchuk, A., Wang, H. J., Schecter, S., & Jackson, C. (2012, August). Clickjacking: Attacks and Defenses. In USENIX Security Symposium (pp. 413-428).

[24] InstallScript Language Reference. Disponible en línea:

http://helpnet.flexerasoftware.com/installshield21helplib/Subsystems/installshield21langref/helplibrary/LangrefHome.htm Verificado 1 de diciembre de 2014

## APENDICES

APENDICE 1. PLANTILLA DE CONFIGURACIONES DE SEGURIDAD INTERNET EXPLORER 10/11

Configuraciones de seguridad generales del navegador

| Posible Vulnerabilidad | Valor inseguro | Valor seguro configurado y característica modificada |
|---|---|---|
| Los usuarios pueden cambiar los valores de las políticas de seguridad de Internet Explorer | No establecido (permitido) | No permitir a los usuarios cambiar las políticas: Habilitado<br><br>"Security_options_edit" |
| Los usuarios individuales pueden cambiar sus configuraciones de uso de un servidor Proxy | Configuración por usuario (permitido) | Usar configuraciones por usuario: Deshabilitado<br><br>"ProxySettingsPerUser" |
| Página de inicio de Internet Explorer configurada a un valor incorrecto | Página de inicio establecida a un sitio arbitrario | Página de inicio configurada a: www.udb.edu.sv |
| Los usuarios pueden cambiar sus configuraciones de las Zonas de seguridad | No establecido (permitido) | Usar solo configuraciones de equipo: Habilitado<br><br>"Security_HKLM_only" |
| Los usuarios pueden agregar sitios potencialmente inseguros cuando la seguridad avanzada de Internet Explorer ha sido activada | No establecido (permitido) | No permitir a los usuarios añadir/editar sitios: Habilitado<br><br>"Security_zones_map_edit" |
| El navegador puede cambiar automáticamente los valores de configuración de un servidor Proxy | No establecido (permitido) | Impedir el cambio de valores de configuración automático: Habilitado<br><br>"Autoconfig" |
| Los usuarios puede participar en el programa “Mejora de la Experiencia de Usuario” de Microsoft | No establecido (permitido) | Impedir participación del usuario en el programa de mejora: Habilitado<br><br>"DisableCustomerImprovementProgram" |
| Desactivar la comprobación de configuración de seguridad del navegador | Habilitado (no se comprueba) | Configuración segura: Desactivada<br><br>"DisableSecuritySettingsCheck" |
| Los usuarios pueden desactivar el modo protegido del navegador en Zonas Internet, Intranet, Sitios Confiables, Sitios Restringidos | No establecido (el usuario puede elegir) | Configuración segura: Habilitado (el usuario no podrá desactivarlo)<br><br>“2500” |
| Los usuarios pueden activar o desactivar el Modo Protegido Mejorado | No establecido (los usuarios pueden manipularla) | Configuración segura: Habilitado (para todas las zonas se usa el Modo Protegido Mejorado)<br><br>“Isolation” |

| | | |
|---|---|---|
| Uso del bloqueador de elementos emergentes en Zonas Internet, Intranet, Sitios Confiables, Sitios Restringidos | Deshabilitado (se muestran los elementos emergentes sin restricciones) | Configuración segura: Habilitado (se restringen la mayor parte de elementos emergentes)<br><br>“1809” |
| Los sitios web en zonas de menor privilegio pueden desplazarse a la Zona Internet | Habilitada (explícitamente o preguntar al usuario) | Configuración segura: Deshabilitada<br><br>“2101” |
| Se permite la elevación de la zona de seguridad en procesos de Internet Explorer | Deshabilitado | Configuración segura: Habilitado<br><br>“FEATURE_ZONE_ELEVATION” |
| Impedir que los usuarios puedan modificar la URL utilizada para los procesos de actualización de Internet Explorer | No establecido (los usuarios pueden hacer cambios) | Configuración segura: Habilitado (no se permite modificar la URL, con un valor sugerido de about:blank)<br><br>"Update_Check_Page" |
| Impedir que los usuarios puedan modificar el intervalo de tiempo para los procesos de actualización | No establecido (los usuarios pueden determinar su propio intervalo) | Configuración segura: Habilitado (con un valor sugerido de 15 días)<br><br>"Update_Check_Interval" |
| Los usuarios pueden activar y desactivar la característica de Sitios Sugeridos | No establecido (el usuario puede manipularla) | Configuración segura: Deshabilitado (no se registra actividad ni se sugieren sitios)<br><br>“SuggestedSites” |
| Los usuarios pueden recibir sugerencias de URLs en la barra de direcciones | No establecido (se muestran las sugerencias y se pueden manipular) | Configuración segura: Habilitado (no se muestran sugerencias y el usuario no puede activarlas)<br><br>“DomainSuggestion” |
| Se ejecuta un asistente de “Primera ejecución” cuando se instala el navegador (nueva versión) o como parte del sistema operativo | No establecido (se permite la ejecución del asistente) | Configuración segura: Habilitado<br><br>(con un valor sugerido “Ir directamente a página principal”)<br><br>"DisableFirstRunCustomize" |

Configuraciones de controles, extensiones y plugins

| Posible Vulnerabilidad | Valor inseguro | Valor seguro configurado y característica modificada |
|---|---|---|
| Controles ActiveX firmados pueden ser descargados sin confirmación en Zonas Internet, Intranet, Sitios Confiables, Sitios Restringidos | Descarga sin confirmación | Configuración segura: Preguntar al usuario<br><br>“1001” |
| Controles ActiveX no firmados pueden ser descargados sin confirmación en Zonas Internet, Intranet, Sitios Confiables, Sitios Restringidos | Descarga sin confirmación | Configuración segura: Desactivar las descargas<br><br>“1004” |

| | | |
|---|---|---|
| Controles ActiveX determinados como no seguros pueden inicializarse y ejecutarse en Zonas Internet, Intranet, Sitios Confiables | Inicialización y ejecución permitidas | Configuración segura: No permitir ejecución de controles determinados como no seguros<br><br>“1201” |
| Ejecución de controles ActiveX seguros y no seguros habilitada para la Zona Sitios Restringidos | Ejecución no permitida | Configuración segura: Ejecución deshabilitada<br><br>“1201” |
| Los usuarios pueden habilitar y deshabilitar complementos | No establecido (permitido) | No permitir a los usuarios habilitar y deshabilitar complementos: Habilitado<br><br>"Enable Browser Extensions" |
| Los usuarios pueden ejecutar contenido activo desde CDs en sus equipos | No establecido (se pregunta al usuario) | Configuración segura: Deshabilitado<br><br>"LOCALMACHINE_CD_UNLOCK" |
| Se permite la ejecución de software con firmas no válidas | No establecido (se pregunta al usuario) | Configuración segura: Deshabilitado<br><br>"RunInvalidSignatures" |
| Se permiten complementos de terceros | No establecido (se permiten) | Configuración segura: Deshabilitado<br><br>"Enable Browser Extensions" |
| Los usuarios pueden instalar y usar Aceleradores no permitidos de terceros | No establecido (se da acceso a los usuarios a los Aceleradores) | Configuración segura: Habilitado (se desactivan los aceleradores)<br><br>“NoActivities” |
| Carga automática de controles ActiveX ya instalados | Habilitado (no se solicita confirmación al usuario) | Configuración segura: Deshabilitado (se solicita confirmación al usuario para cargar el control)<br><br>“120b” |
| Se permite la ejecución de componentes no firmados con Authenticode de .NET en las Zonas Internet, Intranet, Sitios Confiables, Sitios Restringidos | No establecido (se ejecuta el componente) | Configuración segura: Deshabilitado<br><br>“2004” |

Configuraciones de cifrado

| Posible Vulnerabilidad | Valor inseguro | Valor seguro configurado y característica modificada |
|---|---|---|
| Configuración de TLS/SSL incorrecta | Configuración incluye uso de SSL 2.0 y SSL 3.0 | Configuración solamente permite valores seguros: TLS 1.0 o superior<br><br>“'Use TLS 1.0' or higher” |
| Los usuarios no reciben una notificación de certificado de seguridad inválido o incorrecto | Notificación deshabilitada | Notificación de certificado inválido o incorrecto: Habilitada<br><br>"WarnOnBadCertRecving" |

| | | |
|---|---|---|
| El navegador puede no verificar si los certificados de los servidores han sido revocados | No establecido (no se verifica) | Configuración segura: Habilitado<br><br>"CertificateRevocation" |

Configuraciones de Java, JavaScript, ejecución de código y protecciones contra XSS

| Posible Vulnerabilidad | Valor inseguro | Valor seguro configurado y característica modificada |
|---|---|---|
| Permisos de ejecución de código Java en Zonas Internet y Sitios Restringidos | Ejecución permitida con seguridad baja | Configuración segura: Deshabilitar Java<br><br>“1C00” |
| Permisos de ejecución de código Java en Zonas Intranet, Sitios Confiables, Equipo Local | Ejecución permitida con seguridad baja | Configuración segura: Habilitar Java con seguridad Alta<br><br>“1C00” |
| Permisos de ejecución de código Java en Zonas Bloqueadas (Internet, Intranet, Sitios Confiables, Sitios Restringidos, Equipo Local) | Ejecución permitida con seguridad baja | Configuración segura: Deshabilitar Java<br><br>“1C00” |
| Se permite la ejecución de Scriptlets en Zonas Internet, Intranet, Sitios Restringidos | No establecido (el usuario puede elegir) | Configuración segura: Deshabilitado<br><br>“1209” |
| Se permite la ejecución de Scriptlets en Zona Sitios Confiables | Habilitado (se permite ejecución) | Configuración segura: No establecido (el usuario puede elegir)<br><br>“1209” |
| Se permite la carga de contenido XAML en Zonas Intranet, Sitios Confiables | Habilitado (se permite la carga) | Configuración segura: No establecido (el usuario puede elegir)<br><br>“2402” |
| Se permite la carga de contenido XAML en Zonas Internet, Sitios Restringidos | No establecido (el usuario puede elegir) | Configuración segura: Deshabilitado |
| Los usuarios pueden tener acceso a fuentes de datos en dominios diferentes en Zonas Internet, Intranet, Sitios Confiables, Sitios Restringidos | Acceso a fuentes de datos en dominios diferentes permitido | Acceso a fuentes de datos en dominios diferentes: Deshabilitado<br><br>“1406” |
| Los usuarios pueden navegar en contenido de dominios diferentes | Navegación entre dominios permitida | Navegación entre dominios: Deshabilitada<br><br>“1607” |
| Se permiten operaciones de copiar y pegar desde el portapapeles mediante scripts en Zona Sitios Restringidos | Operaciones de copiar y pegar desde el portapapeles permitidas | Operaciones de copiar y pegar desde el portapapeles: Deshabilitadas<br><br>“1802” |

| | | |
|---|---|---|
| Funciones de Active Scripting permitidas en Zona Sitios Restringidos | Funciones de Active Scripting permitidas | Configuración segura: Active Scripting deshabilitado<br><br>“1400” |
| Aplicaciones y archivos pueden ser cargados en un IFrame sin confirmación en Zonas Internet, Intranet, Sitios Confiables | Creación de IFrames automática | Configuración segura: Preguntar al usuario<br><br>“1804” |
| Aplicaciones y archivos pueden ser cargados en un IFrame sin confirmación en Zona Sitios Restringidos | Creación de IFrames automática | Configuración segura: IFrames deshabilitados<br><br>“1804” |
| Se permite el acceso sin intervención del usuario a Applets de Java en las Zonas Internet, Intranet, Sitios Confiables, Sitios Restringidos | Habilitado (se permite el acceso sin intervención del usuario) | Configuración segura: Deshabilitado<br><br>“1402” |
| Se permiten ventanas iniciadas por scripts sin restriccion de tamaño/posición en Zonas Internet, Intranet, Sitios Confiables, Sitios Restringidos | Habilitado (no se aplican restricciones a estas zonas) | Configuración segura: Deshabilitado<br><br>“2102” |
| Los usuarios pueden arrastrar contenido entre ventanas de diferentes dominios en las Zonas Internet, Sitios Restringidos | Habilitado | Configuración segura: Deshabilitado<br><br>“2709” |
| Permitir el comportamiento de binarios y scripts en Zona Sitios Restringidos | No establecido (se permiten) | Configuración segura: Deshabilitado<br><br>“2000” |
| No se utiliza el filtro de scripts de sitios (XSS) en las Zonas Internet, Intranet, Sitios Confiables, Sitios Restringidos | Deshabilitado (filtro no se utiliza) | Configuración segura: Habilitado<br><br>“1409” |
| Se permiten cambios en la barra de estado vía scripts en Zonas Internet, Sitios Restringidos | Habilitado (se actualiza la barra de estado desde los scripts) | Configuración segura: Deshabilitado (se puede habilitar en las Zonas Intranet, Sitios Confiables)<br><br>“2103” |
| Se permite la ejecución de scripts en controles tipo WebBrowser en las Zonas Internet, Sitios Confiables, Sitios Restringidos | No establecido (el usuario puede elegir) | Configuración segura: Deshabilitado (puede habilitarse en las Zonas Intranet, Equipo Local)<br><br>“1206” |
| Se permite la redirección a otro sitio usando la etiqueta META REFRESH en Zonas Internet, Sitios Restringidos | No establecido (permitido) | Configuración segura: Deshabilitado<br><br>“1608” |

Configuraciones de descarga de archivos y otros elementos

| Posible Vulnerabilidad | Valor inseguro | Valor seguro configurado y característica modificada |
|---|---|---|
| Descarga automática de fuentes en Zonas Internet, Intranet, Sitios Confiables | Descarga sin confirmación | Configuración segura: Preguntar al usuario<br><br>“1604” |
| Descarga automática de fuentes en Zona Sitios Restringidos | Descarga sin confirmación | Configuración segura: Descarga deshabilitada<br><br>“1604” |
| Se permiten descargas de archivos automáticas desde las Zonas, Internet, Intranet, Sitios Confiables, Sitios Restringidos | Descarga de archivos permitida | Configuración segura: Descarga de archivos deshabilitada<br><br>“1803” |
| Se permite la instalación de elementos en el Escritorio desde las Zonas Internet, Intranet, Sitios Confiables, Sitios Restringidos | Instalación de elementos en el Escritorio permitida | Configuración segura: Instalación deshabilitada<br><br>“1800” |
| Requerir datos MIME coherentes para los archivos recibidos en los procesos de Internet Explorer | Deshabilitado | Configuración segura: Habilitado<br><br>“FEATURE_MIME_HANDLING” |
| Habilitar características de seguridad y examen de tipos MIME en procesos de Internet Explorer | Deshabilitado | Configuración segura: Habilitado<br><br>“2100” |
| Se limita la intervención del usuario en las descargas no iniciadas con su participación en los procesos de Internet Explorer | No establecido (se pregunta al usuario) | Configuración segura: Habilitado (no se permite intervención)<br><br>“2200” |
| Los usuarios tienen acceso a funcionalidades de arrastrar y soltar en la Zona de Internet | Funcionalidad de arrastrar y soltar permitida | Funcionalidad de arrastrar y soltar: Deshabilitada<br><br>“1802” |
| Permitir el examen de MIME en Zonas Internet, Intranet, Sitios Confiables, Sitios Restringidos | No establecido (no se aplican restricciones a las zonas) | Configuración segura: Deshabilitada (no permite acciones maliciosas)<br><br>“2100” |
| No se solicita confirmación de primera ejecución en Zonas Internet, Intranet, Sitios Confiables, Sitios Restringidos | No establecido (no se solicita confirmación en estas zonas) | Configuración segura: Deshabilitada (se solicita primera confirmación)<br><br>“1208” |
| El navegador puede no verificar las firmas y su validez en programas descargados | No establecido (no se verifica) | Configuración segura: Habilitado<br><br>“RunInvalidSignatures” |

| | | |
|---|---|---|
| Se envía información sobre rutas locales cuando se suben archivos a un servidor en Zonas Internet, Intranet, Sitios Confiables, Sitios Restringidos | No establecido (por defecto se envía la información de rutas locales) | Configuración segura: Deshabilitado (se elimina la información de rutas locales usando un formulario HTML)<br><br>“UNCAsIntranet” |
| No se pregunta al usuario ni se le advierte sobre la apertura de archivos potencialmente peligrosos en Zonas Internet, Intranet, Sitios Confiables | Habilitado (los archivos se abren sin advertencias de seguridad) | Configuración segura: Habilitado (con un valor Preguntar para que el usuario decida si desea abrir el archivo)<br><br>“2200” |
| No se pregunta al usuario ni se le advierte sobre apertura de archivos potencialmente peligrosos en Zona Sitios Restringidos | Habilitado (los archivos se abren sin advertencia) | Configuración segura: Deshabilitado (los archivos no se abren)<br><br>“2200” |
| No se muestra la barra de notificaciones cuando se detecta un archivo o código restringidos en los procesos de Internet Explorer | No establecido (o Deshabilitado, no se muestra la barra de notificación) | Configuración segura: Habilitado<br><br>“FEATURE_SECURITYBAND” |

Configuraciones de Privacidad

| Posible Vulnerabilidad | Valor inseguro | Valor seguro configurado y característica modificada |
|---|---|---|
| La persistencia de datos del usuario está habilitada en Zonas Internet, Intranet, Sitios Confiables, Sitios Restringidos | Persistencia de datos de usuario habilitada | Persistencia de datos de usuario: Deshabilitada<br><br>“1606” |
| Opciones de inicio de sesión establecidas en automático en Zonas Internet, Intranet, Sitios Confiables | Inicio de sesión automático con credenciales actuales de la sesión | Inicio de sesión: Solicitar siempre nombre de usuario y contraseña<br><br>“1A00” |
| Opciones de inicio de sesión establecidas en automático en Zona Sitios Restringidos | Inicio de sesión automático con credenciales actuales de la sesión | Inicio de sesión: Anónimo para esta zona<br><br>“1A00” |
| Los usuarios pueden alterar el funcionamiento del filtro SmartScreen | Habilitado (y con un valor Activado). Los usuarios pueden modificar el funcionamiento y si se activa se envían todos los sitios a Microsoft sin consultar (afecta la privacidad) | Configuración segura: Habilitado (con un valor Desactivado) para que el usuario elija si desea utilizarlo en la primera ejecución<br><br>“EnabledV9” |
| Los usuarios pueden modificar el comportamiento de la característica Do Not Track del navegador | No establecido (el usuario puede manipularla) | Configuración segura: Habilitado (siempre se envía el encabezado DNT:1)<br><br>“DoNotTrack” |
| Los usuarios pueden usar el botón “Revelar Contraseñas” | No establecido (el botón se muestra en los campos de contraseña) | Configuración segura: Habilitado (no se muestra el botón)<br><br>“DisablePasswordReveal” |

| | | |
|---|---|---|
| Los usuarios pueden activar y desactivar la característica de Geolocalización del navegador | No establecido (el usuario puede manipularla) | Configuración segura: Habilitado (no se habilita la Geolocalización)<br><br>“PolicyDisableGeolocation” |
| Los usuarios pueden eliminar el historial de navegación y modificar la cantidad de días de almacenamiento del mismo | No establecido (se puede borrar y modificar el historial) | Configuración segura: Habilitado (con un valor sugerido de 90 días)<br><br>“History”<br><br>“DaystoKeep” |
| Se puede almacenar en cache información sensitiva mientras está habilitada la funcionalidad de Autocompletar para formularios | No establecido (el usuario puede elegir) | Configuración segura: Habilitado (no se activará la característica)<br><br>“FormSuggest” |

APENDICE 2. PLANTILLA DE CONFIGURACIONES DE SEGURIDAD GOOGLE CHROME

Configuraciones de seguridad generales del navegador

| Posible Vulnerabilidad | Valor inseguro | Valor seguro configurado y característica modificada |
|---|---|---|
| Equipos remotos pueden conectarse al navegador inclusive cuando se dispone de un firewall para filtrado de conexiones | Política no definida (se permiten las conexiones) | Configuración segura: Deshabilitado (se permiten conexiones solo desde la misma red de área local) "RemoteAccessHostFirewallTraversal" |
| Se muestran ventanas pop-up no deseadas | Política no definida (los usuarios pueden manipularla) | Configuración segura: Habilitada (no permitir que ningún sitio muestre pop-ups) "DefaultPopupsSetting" |
| Se puede manipular el proveedor de búsquedas por defecto e incluir una URL comprometida que devolverá resultados maliciosos al usuario | Política no definida 1 (se permiten búsquedas desde la barra de direcciones – omnibox con cualquier proveedor de búsquedas)<br><br>Política no definida 2 (se permite al usuario establecer la lista de proveedores de búsqueda y manipularla) | Configuración segura: Habilitada (se permiten búsquedas desde el omnibox). Se debe configurar además la política del nombre para el proveedor de búsquedas predeterminado y la correspondiente URL con el elemento {searchTerms} y se sugiere que sea Google Cifrado (https://encrypted.google.com) "DefaultSearchProviderEnabled" "DefaultSearchProviderSearchURL" |
| Google Chrome puede seguir ejecutándose en segundo plano inclusive después de cerrar la última ventana del navegador (esto permitiría que código malicioso se siguiera ejecutando aun cuando el usuario ha dejado de utilizar el navegador) | Política no definida (el usuario puede manipularla) | Configuración segura: Deshabilitada (no se continuará ejecutando un proceso de Chrome en segundo plano y el usuario no puede cambiar esta opción) "BackgroundModeEnabled" |
| Google Chrome puede actuar como un proxy entre el servicio de Google Cloud Print y los recursos de impresión locales al equipo del usuario. | Política no definida (la función de proxy de Google Cloud Print está habilitada) | Configuración segura: Deshabilitada (no se puede utilizar ni habilitar la función de proxy de Google Cloud Print) "CloudPrintProxyEnabled" |

| | | |
|---|---|---|
| El navegador puede realizar peticiones previas de DNS y preconexiones de SSL y TCP, e inclusive pre renderizar páginas que el usuario no ha solicitado de manera explicita | Política no definida (la característica es funcional y el usuario puede manipularla) | Configuración segura: Deshabilitada (se evitan todas las acciones que involucren prefetching de servicios de red) "DnsPrefetchingEnabled" |
| Se pueden ofrecer al usuario sugerencias de búsqueda que lo dirijan a resultados potencialmente maliciosos | Política no definida (el usuario puede manipularla) | Configuración segura: Deshabilitada (no se entregarán sugerencias de búsqueda al usuario) "SearchSuggestEnabled" |
| Los usuarios pueden manipular la política de Navegación Segura de Google Chrome | Política no definida (el usuario puede manipularla) | Configuración segura: Habilitada (se utiliza la base de datos de sitios seguros para garantizar que se navega en sitios sin problemas de código o elementos maliciosos) "SafeBrowsingEnabled" |
| Los usuarios pueden establecer o utilizar una página de inicio no autorizada por la institución | Política no definida (el usuario puede manipularla) | Configuración segura: Habilitada (con el valor de URL aprobado http://www.udb.edu.sv) "HomepageLocation" |
| Los usuarios pueden inhabilitar las opciones de actualización automática del navegador de modo que pueden estar trabajando con una versión obsoleta e inclusive comprometida del navegador | Valor del registro establecido en cero ("0" no se permite actividad de actualización) | Configuración segura: Valor del registro establecido en un valor de 1440 (minutos) para establecer al menos una vez al día si existen actualizaciones. "AutoUpdateCheckPeroidMinutes" |

Configuraciones de controles, extensiones y plugins

| Posible Vulnerabilidad | Valor inseguro | Valor seguro configurado y característica modificada |
|---|---|---|
| Los usuarios pueden instalar extensiones no permitidas y potencialmente maliciosas | Política no definida (se permite la instalación de cualquier extensión) | Configuración segura: Habilitada (con un valor de * para indicar que no se permite la instalación de ninguna extensión. Posteriormente se definirá una lista blanca de las extensiones permitidas) ..ExtensionInstallBlacklist] "1"="*" |
| Los usuarios pueden instalar extensiones fuera de la lista blanca definida por la institución | Política no definida (se admiten todas las extensiones) | Configuración segura: Habilitada (se debe especificar las extensiones permitidas mediante una lista blanca usando el ID respectivo de cada extensión) "1"="bhmmomiinigofkjcapegjjndpbikblnp" "2"="cfnpidifppmenkapgihekkeednfoenal" "3"="cfhdojbkjhnklbpkdaibdccddilifddb" "4"="oiigbmnaadbkfbmpbfijlflahbdbdgdf" "5"="gcbommkclmclpchllfjekcdonpmejbdp" "6"="mlomiejdfkolichcflejclcbmpeaniij" "7"="gieohaicffldbmiilohhggbidhephnjj" |

| | | |
|---|---|---|
| Se pueden estar ejecutando plugins desactualizados y obsoletos que sean potencialmente vulnerables | Política no definida (se permite al usuario decidir si se permitirán plugins obsoletos) | Configuración segura: Deshabilitada (no se permite el uso de plugins obsoletos y no se consulta al usuario) “AllowOutdatedPlugins” |
| Se pueden ejecutar automáticamente plugins que si cumplan con el criterio de actualización y no obsolescencia | Habilitada (los plugins se ejecutan automáticamente sin intervención del usuario) | Configuración segura: Deshabilitada (se solicitará confirmación del usuario antes de ejecutar un plugin) "AlwaysAuthorizePlugins" |
| Los usuarios pueden instalar plugins no autorizados que pueden resultar potencialmente maliciosos | Política no definida (el usuario puede utilizar cualquier plugin, excepto los marcados por obsolescencia) | Configuración segura: Habilitada (se define una lista negra en la que se incluyen TODOS los plugins, con un valor de “*”. Luego en una lista blanca se definirán los plugins autorizados y que se consideran seguros)<br><br>..DisabledPlugins] "1"="*" |
| Los usuarios pueden utilizar plugins no aprobados fuera de la lista blanca | Política no definida (el usuario puede manipular los plugins ya instalados) | Configuración segura: Habilitada (se debe definir una lista blanca explicita de los plugins permitidos con su nombre. Estos no podrán ser deshabilitados por el usuario)<br><br>"1"="Shockwave Flash"<br><br>"2"="Chrome PDF Viewer"<br><br>"3"="Silverlight"<br><br>"4"="Java*" |
| El buscador de complementos está activo y podría ofrecer sugerencias sobre plugins potencialmente maliciosos | Política no definida (buscador de complementos activo) | Configuración segura: Habilitada (se impide la ejecución del buscador de complementos)<br>"DisablePluginFinder" |
| Ciertos sitios pueden iniciar la ejecución automática de plugins potencialmente maliciosos sin intervención del usuario | Política no definida (se puede manipular) | Configuración segura: Habilitada (con un valor de Hacer click para ejecutar de modo que el usuario sea consciente de la ejecución del plugin) "DefaultPluginsSetting" |

Configuraciones de cifrado

| Posible Vulnerabilidad | Valor inseguro | Valor seguro configurado y característica modificada |
|---|---|---|
| Se puede hacer uso de autenticación HTTP “básica” implicando esto que las credenciales del usuario son enviadas en texto claro al servidor | Política no definida (se utilizan todos los esquemas posibles: basic, digest, ntlm, negotiate) | Configuración segura: Habilitada (configurando solo los valores digest, ntlm, negotiate) "AuthSchemes" |
| Es posible que el navegador esté aceptando certificados comprometidos o no válidos | Política no definida (no se realizan comprobaciones sobre certificados revocados) | Configuración segura: Habilitada (se realizan verificaciones de revocación OCSP/CRL)<br>"EnableOnlineRevocationChecks" |

Configuraciones de Java, JavaScript, ejecución de código y protecciones contra XSS

| Posible Vulnerabilidad | Valor inseguro | Valor seguro configurado y característica modificada |
|---|---|---|
| El acceso no controlado a URLs en el esquema “JavaScript://” está habilitado | Política no definida (ningún esquema está dentro de la lista negra de Chrome) | Configuración segura: Habilitada (con un valor de esquema “javascript://*”) |
| Los usuarios pueden recibir notificaciones maliciosas directo al escritorio | Política no definida (los usuarios pueden manipularla) | Configuración segura: Habilitada (no permitir que ningún sitio muestre notificaciones) "DefaultNotificationsSetting" |
| El soporte para APIs de gráficos 3D puede estar activo permitiendo a un posible atacante ejecutar instrucciones usando la GPU y así ralentizar innecesariamente el sistema. | Deshabilitada (se permite el acceso a las funciones de las APIs de gráficos 3D como WebGL y Pepper 3D) | Configuración segura: Habilitada (no se permite el acceso a las APIs mencionadas, debe tenerse en cuenta que muy pocos sitios realmente toman ventaja de forma válida de estas APIs) "Disable3DAPIs" |

Configuraciones de privacidad

| Posible Vulnerabilidad | Valor inseguro | Valor seguro configurado y característica modificada |
|---|---|---|
| Sitios maliciosos pueden registrar la ubicación del usuario | Política no definida (los usuarios pueden manipularla) | Configuración segura: Habilitada (no permitir que ningún sitio haga seguimiento de la ubicación física del usuario) "DefaultGeolocationSetting" |
| Los usuarios pueden ingresar contraseñas en cajas de diálogo o formularios en texto claro y pueden ser vistas por terceros (shoulder surfing) | Política no definida (los usuarios pueden ver sus contraseñas en texto claro) | Configuración segura: Deshabilitada (no se muestran las contraseñas en texto claro) "PasswordManagerAllowShowPasswords" |
| El administrador de contraseñas de Google Chrome almacena información que podría ser utilizada por sitios maliciosos para obtener contraseñas de manera automática | Política no definida (se almacenan las contraseñas y el usuario puede manipular esta característica) | Configuración segura: Deshabilitada (no se almacenan o recuerdan contraseñas para ningún sitio) "PasswordManagerEnabled" |
| Se permiten Cookies de terceros (dominios diferentes al que aloja la página originalmente) | Política no definida (el usuario puede manipularla) | Configuración segura: Habilitada (no se permiten cookies de terceros) "BlockThirdPartyCookies" |
| Información privada se está almacenando en servidores de Google mediante el servicio de Google Sync | Política no definida (el usuario puede manipularla) | Configuración segura: Habilitada (no hay sincronización y el usuario no puede cambiar esta opción) "SyncDisabled" |
| Se almacena información sensitiva para ser utilizada en la función de autocompletar formularios | Política no definida (el usuario puede manipularla) | Configuración segura: Deshabilitada (no se utilizará la funcionalidad de autocompletar en formularios) "AutoFillEnabled" |

| | | |
|---|---|---|
| Se pueden enviar informes anónimos de funcionamiento, utilización y fallas de Google Chrome a servidores de Google (incluyendo información sensitiva presente en la memoria en el momento de un fallo por ejemplo) | Política no definida (el usuario puede habilitar el envío de informes cuando se instala y utiliza por primera vez el navegador) | Configuración segura: Deshabilitada (nunca se envían informes anónimos a Google) "MetricsReportingEnabled" |
| Se pueden importar contraseñas no cifradas desde el navegador predeterminado anterior y almacenarlas en forma insegura en el sistema | Política no definida (en el cuadro de diálogo respectivo en la primera ejecución el usuario puede decidir si importa las contraseñas) | Configuración segura: Deshabilitada (no se importan las contraseñas) "ImportSavedPasswords" |
| Los usuarios pueden manipular la política de almacenamiento del historial de manera que no se almacena evidencia de la actividad del usuario que permitiera determinar la causa de posibles problemas iniciados por código malicioso | Política no definida (por defecto el historial se almacena pero el usuario puede manipular esta opción) | Configuración segura: Deshabilitada (se guarda el historial y el usuario no puede cambiar la opción) "SavingBrowserHistoryDisabled" |
| El navegador puede presentar un comportamiento potencial en contra de la privacidad si la política de utilizar cookies exclusivas para una sesión está combinada con la política de restaurar los sitios abiertos la última vez que se utilizó el navegador, haciendo permanente las cookies para estos sitios | Política no definida (se aplica política global de cookies excepto para los sitios restaurados de sesiones anteriores) | Configuración segura: Deshabilitada “CookiesSessionOnlyForUrls” |

APENDICE 3. PLANTILLA DE CONFIGURACIONES DE SEGURIDAD MOZILLA FIREFOX

Configuraciones de seguridad generales del navegador

| Posible Vulnerabilidad | Valor inseguro | Valor seguro configurado y característica modificada |
|---|---|---|
| La página de inicio en el navegador puede ser establecida a un valor no aprobado y potencialmente inseguro | No configurado explícitamente | Preferencia establecida explícitamente (valor www.udb.edu.sv)<br><br>Característica modificada directamente en interface de Firefox |
| El navegador puede haber sido configurado para no permitir actualizaciones, creando la posibilidad de contar con una versión obsoleta del mismo | No configurado explícitamente (manipulable) | Preferencia establecida explícitamente (con un valor “true” para garantizar que las actualizaciones sean permitidas)<br><br>"app.update.enabled" |
| La URL de actualización del navegador puede haber sido manipulada, permitiendo la descarga de actualizaciones potencialmente maliciosas o no funcionales | No configurado explícitamente (manipulable) | Preferencias establecidas explícitamente (Deben ser de servidores bien conocidos de Mozilla)<br><br>“app.update.url” |

| | | |
|---|---|---|
| El navegador puede haber sido configurado para alterar el intervalo de búsqueda de actualizaciones del mismo, en un valor demasiado alto que deje una versión obsoleta operando para el usuario por un tiempo prolongado, o un valor demasiado bajo que incremente la actividad producida por el navegador | No configurado explícitamente (manipulable) | Preferencia establecida explícitamente (en un valor igual o superior a 43200)<br><br>"app.update.interval" |
| El bloqueador de Pop-ups puede ser desactivado por el usuario | No configurado explícitamente (manipulable) | Preferencia establecida explícitamente (valor “1” para mantener el bloqueador activo siempre)<br><br>"privacy.popups.policy" |
| Los usuarios podrían estar navegando sin utilizar la característica de “navegación segura” del navegador | No configurado explícitamente (manipulable) | Preferencia establecida explícitamente (con un valor “true” para garantizar que siempre se utiliza la característica de navegación segura)<br><br>"browser.safebrowsing.enabled" |
| Algunos ataques de malware bien conocidos podrían afectar al navegador al no tener habilitada la opción de navegación segura con bloqueo de malware | No configurada explícitamente (manipulable) | Preferencia establecida explícitamente (con un valor “true” para habilitar la opción de navegación segura con protección contra malware conocido)<br><br>"browser.safebrowsing.malware.enabled" |

Configuraciones de controles, extensiones y plugins

| Posible Vulnerabilidad | Valor inseguro | Valor seguro configurado y característica modificada |
|---|---|---|
| El navegador puede haber sido configurado para no notificar la obsolescencia de plugins instalados dejando versiones potencialmente vulnerables en operación | No configurado explícitamente (manipulable) | Preferencia establecida explícitamente (en un valor “true” para garantizar la notificación de actualización al usuario)<br><br>"plugins.update.notifyUser" |
| Los usuarios podrían inadvertidamente hacer click en los botones de un cuadro de diálogo necesario para la instalación de alguna característica o complemento, sin haber leído completamente las advertencias o recomendaciones de seguridad | Configurado en un valor muy bajo (1000 ms) | Preferencia establecida explícitamente (con un valor suficiente para que el usuario pueda visualizar las advertencias y/o recomendaciones del cuadro de diálogo, por ej. “4000”)<br><br>"security.dialog_enable_delay" |

| | | |
|---|---|---|
| Los usuarios podrían instalar complementos no autorizados y potencialmente maliciosos en el navegador | No configurado explícitamente (manipulable) | Preferencia establecida explícitamente (con un valor “true” para requerir que los complementos a instalarse correspondan a una lista blanca aprobada por la institución)<br><br>"xpinstall.whitelist.required" |
| No se utiliza una lista negra de complementos para bloquear su instalación por parte del usuario | No configurado explícitamente (manipulable) | Preferencia establecida explícitamente (con un valor “true” para requerir la lista negra y así evitar la instalación de complementos no autorizados o maliciosos)<br><br>"extensions.blocklist.enabled" |

Configuraciones de cifrado

| Posible Vulnerabilidad | Valor inseguro | Valor seguro configurado y característica modificada |
|---|---|---|
| Los usuarios pueden estar utilizando versiones obsoletas de SSL/TLS para acceder a sitios comprometiendo así la seguridad del canal de comunicaciones (SSLv3) | No existente | Preferencia creada (con valor “false” para la habilitación de SSLv3)<br><br>"security.enable_ssl3" |
| Los usuarios pueden estar utilizando versiones obsoletas de SSL/TLS para acceder a sitios comprometiendo así la seguridad del canal de comunicaciones (SSLv2) | No existente | Preferencia creada (con valor “false” para la habilitación de SSLv2)<br><br>"security.enable_ssl2" |
| El navegador no ha sido configurado para utilizar de manera predeterminada TLS | No existente | Preferencia creada (con valor “true” para la habilitación de TLS)<br><br>"security.enable_tls" |
| El navegador puede tener valores incorrectos de configuración en las versiones de TLS a soportar (“0” para forzar SSLv3) | No configurados explícitamente | Preferencia establecida explícitamente (valores: “1” para la versión mínima es decir TLS 1.1, y “3” para la versión máxima, para TLS 1.3)<br><br>"security.tls.version.min"<br><br>"security.tls.version.max" |
| Los usuarios pueden estar utilizando un certificado personal de manera automática sin percatarse de su obsolescencia o invalidez | No configurado explícitamente | Preferencia establecida explícitamente: (valor “Ask Every Time” para asegurar que el usuario autorice siempre el uso del certificado personal)<br><br>"security.default_personal_cert" |

| | | |
|---|---|---|
| Los usuarios podrían no recibir una advertencia cuando en un sitio de pasa de una página segura (con ssl/tls habilitado) a una página no segura | No existente | Preferencia creada (con un valor “true” para garantizar que el usuario reciba la notificación siempre)<br><br>"security.warn_leaving_secure" |
| El navegador, de forma inadvertida para el usuario podría estar enviando datos de autenticación usando LM Hash que es conocida por presentar vulnerabilidades | No configurado explícitamente (manipulable) | Preferencia establecida explícitamente (con un valor “false” para evitar que se responda a solicitudes de autenticación que requieran LM Hash)<br><br>"network.ntlm.send-lm-response" |
| Los usuarios pueden agregar inadvertidamente certificados con potencial malicioso cuando el navegador indica que el certificado original ya no es válido y rellena la información para agregarlo de manera automática | Configurado en valor “2” por defecto (permite el auto llenado de la URL del certificado y su adición automática) | Preferencia establecida explícitamente (con un valor “0” para forzar a que el usuario digite la URL del certificado y haga click en el botón para agregarlo)<br><br>"browser.ssl_override_behavior" |

Configuraciones de Java, JavaScript, ejecución de código y protecciones contra XSS

| Posible Vulnerabilidad | Valor inseguro | Valor seguro configurado y característica modificada |
|---|---|---|
| Las preferencias del navegador permiten el acceso al intérprete de comandos (Shell) del sistema, permitiendo la ejecución de comandos potencialmente maliciosos | No configurado explícitamente (manipulable) | Preferencia establecida explícitamente (valor “false” para no permitir el acceso al Shell del sistema)<br><br>"network.protocol-handler.external.shell" |
| La configuración del navegador no bloquea la creación de ventanas emergentes que pudieran ser creadas para ejecutar código malicioso o lanzar ataques | No configurado explícitamente (manipulable) | Preferencia establecida explícitamente (valor “true” para evitar la creación de ventanas emergentes)<br><br>"dom.disable_window_open_feature.status" |
| Código malicioso JavaScript puede ser utilizado para mover o cambiar el tamaño de una ventana de modo que pueda esconderse o que pueda obstaculizar el acceso a otras ventanas | No configurado explícitamente (permitido) | Preferencia establecida explícitamente (valor “true” para evitar la manipulación de las características de las ventanas)<br><br>"dom.disable_window_move_resize" |
| Código malicioso JavaScript puede ser utilizado para intercambiar entre ventanas del navegador pudiendo ocultar un posible ataque. | No configurado explícitamente (permitido) | Preferencia establecida explícitamente (valor “true” para evitar el efecto de flip entre ventanas)<br><br>"dom.disable_window_flip" |

| | | |
|---|---|---|
| Código malicioso JavaScript puede ser utilizado para cambiar o alterar los menús contextuales en ciertos sitios e introducir opciones que inicien ataques u otra actividad maliciosa | No configurado explícitamente (permitido) | Preferencia establecida explícitamente (valor “false” para evitar la modificación de los menús contextuales)<br><br>“dom.event.contextmenu.enabled" |
| Código malicioso JavaScript puede ser utilizado para ocultar o modificar el contenido de la barra de estado del navegador | No configurado explícitamente (permitido) | Preferencia establecida explícitamente (valor “true” para deshabilitar la manipulación de la barra de estado<br><br>“dom.disable_window_open_feature.status" |
| Código malicioso JavaScript puede ser utilizado para modificar una ventana emergente aparentemente válida removiendo información que permitiría detectar un posible ataque, por ejemplo la barra de estado | No configurado explícitamente (permitido) | Preferencia establecida explícitamente (valor “true” para evitar que pueda ocultarse completamente la barra de estado)<br><br>"dom.disable_window_status_change" |
| El navegador podría abrir o ejecutar archivos no seguros tipo JAR | No configurado explícitamente (manipulable) | Preferencia establecida explícitamente (con un valor “false”, para evitar la apertura o ejecución de archivos JAR potencialmente maliciosos)<br><br>"network.jar.open-unsafe-types" |
| Código malicioso existente en algunas páginas locales puede aprovechar el entorno más libre del esquema “File” para acceder a información sensible | No configurado explícitamente (manipulable) | Preferencia establecida explícitamente (con un valor “true” para forzar una política de mismo origen más estricta al esquema File<br><br>"security.fileuri.strict_origin_policy" |
| Código malicioso JavaScript puede iniciar interacción y tomar control sobre plugins instalados en el navegador | No configurado explícitamente (manipulable) | Preferencia establecida explícitamente (con un valor “false” que evita que se exploten vulnerabilidades en los plugins o que se abuse de sus funcionalidades<br><br>"security.xpconnect.plugin.unrestricted" |
| Código malicioso JavaScript puede ocultar la barra de direcciones del navegador | No configurado explícitamente (manipulable) | Preferencia establecida explícitamente (valor “true” no se puede ocultar la barra)<br><br>"app.dom.disable_window_open_feature.location" |
| Los usuarios podrían visualizar URLs JavaScript en el historial de navegación, con destinos potencialmente maliciosos | No configurado explícitamente (manipulable) | Preferencia establecida explícitamente (valor “true” para filtrar las URLs JavaScript y que no sean mostradas en el historial)<br><br>"browser.urlbar.filter.javascript" |

Configuraciones de descarga de archivos y otros elementos

| Posible Vulnerabilidad | Valor inseguro | Valor seguro configurado y característica modificada |
|---|---|---|
| La configuración del navegador puede permitir la apertura automática (y en pantalla completa) de ciertos tipos de archivos potencialmente maliciosos | No configurado explícitamente | Preferencia establecida explícitamente (lista de extensiones sin reproducción automática / en pantalla completa)<br><br>“plugin.disable_full_page_plugin_for_types” |
| El navegador puede haber sido configurado para realizar instalaciones de ítems actualizados de manera oculta al usuario y sin su intervención | No configurado explícitamente (manipulable) | Preferencia establecida explícitamente (valor “false” para no permitir instalaciones silenciosas)<br><br>"plugins.update.notifyUser" |
| Los usuarios podrían descargar archivos de formatos permitidos pero infectados por virus | No existente (no se obliga a un escaneo del archivo cuando se completa la descarga) | Preferencia creada (con un valor de “true” para obligar a un escaneo de los archivos descargados)<br><br>"browser.download.manager.scanWhenDone" |
| El navegador podría permitir el acceso a contenido activo mixto a través de HTTP | No configurado explícitamente (manipulable) | Preferencia establecida explícitamente (con un valor “true” para bloquear el contenido activo mixto)<br><br>"security.mixed_content.block_active_content" |

Configuraciones de privacidad

| Posible Vulnerabilidad | Valor inseguro | Valor seguro configurado y característica modificada |
|---|---|---|
| Sitios mal intencionados pueden tener acceso a información utilizada para llenar formularios anteriores y que fue almacenada como ayuda al proceso de auto llenado | No configurado explícitamente (manipulable) | Preferencia establecida explícitamente (valor “false” para desactivar la característica de auto llenado de formularios)<br><br>"browser.formfill.enable" |
| Potenciales atacantes pueden aprovechar la característica de auto llenado de contraseñas pudiendo utilizarlas para acceder de manera no autorizada a los sitios utilizados por el usuario | No configurado explícitamente (manipulable) | Preferencia establecida explícitamente (valor “false” para no permitir la función de auto llenado de contraseñas)<br><br>"signon.prefillForms" |
| Potenciales atacantes pueden tener acceso a los archivos en que se almacenan las contraseñas y extraerlas para utilizarlas de manera fraudulenta | No configurado explícitamente (manipulable) | Preferencia establecida explícitamente (con un valor “false” para impedir el almacenamiento de contraseñas por parte del navegador)<br><br>"signon.rememberSignons" |

| | | |
|---|---|---|
| Las cookies y otra información que puede violentar la privacidad del usuario puede persistir luego de haber finalizado la sesión y cerrado las ventanas del navegador | No configurado explícitamente (manipulable) | Preferencia establecida explícitamente (con un valor “true” en la opción de sanitizar al cerrar, y con un valor “false” en la opción de preguntar al usuario si desea sanitizar el navegador y evitar la persistencia de datos<br><br>"privacy.sanitize.promptOnSanitize"<br><br>"privacy.sanitize.sanitizeOnShutdown" |
| Algunos sitios podrían rastrear la actividad del usuario sin su consentimiento | No configurado explícitamente (no se envía encabezado DNT) | Preferencia establecida explícitamente (con un valor “true” para habilitar el envío del encabezado y un valor “1” en el encabezado propiamente dicho)<br><br>"privacy.donottrackheader.enabled"<br><br>"privacy.donottrackheader.value" |
| El navegador podría permitir cookies de terceros (dominios diferentes al que se refleja en la barra de direcciones) | No configurado explícitamente (manipulable) | Preferencia establecida explícitamente (con un valor “1” para evitar el uso de cookies de terceros, como parte de la protección de la privacidad del usuario)<br><br>"network.cookie.cookieBehavior" |
| Los usuarios podrían ser redirigidos a una Phishy URL, es decir que incluye en su sintaxis nombres de usuario y contraseñas, sin recibir una advertencia | No existente | Preferencia creada (con un valor de “1” para enviar una advertencia sobre la detección de phishy urls)<br><br>"network.http.phishy-userpass-length" |