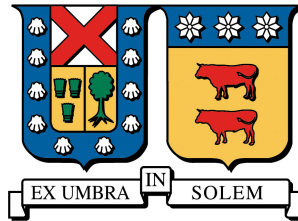


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO, CHILE



?????

Paulina Andrea Silva Ghio

Memoria para optar al título de: Ingeniera Civil Informática

Profesor Guía: Raúl Monges
Profesor Correferente: Javier Cañas

21 de septiembre de 2015

Resumen

El Web Browser es una de las aplicaciones más usadas - *killer app* - y también una de las primeras en aparecer en cuanto se creó el Internet (Década de los 90). Por lo mismo, su nivel de madurez con respecto a otros desarrollos es significativo y permite asegurar ciertos niveles de confianza cuando otros usan un Web Browser como cliente para sus Sistemas.

Actualmente muchos desarrollos de software crean sistemas que están conectados a la Internet, pues permite agregar funcionalidades al sistema y facilidades para sus *Stakeholders*. Esto lleva a depender de un cliente web, cómo un *Web Browser* que permita el acceso a los servicios, datos u operaciones que el sistema entrega. Sin embargo, la Internet influye en la superficie de ataque del nuevo sistema que se implementó, y lamentablemente tanto *Stakeholders* como muchos desarrolladores no están al tanto de los riesgos a los que se enfrentan.

En esta Memoria presentada al Departamento de Informática (DI) de la UTFSM¹ Casa Central, se al incursionará en el ámbito de la seguridad del Web Browser, con el objetivo de obtener documentos formales que servirán como herramientas a personas que Desarrollen Software y hagan un fuerte uso del Navegador para las actividades del sistema desarrollado.

Abstract

The Web Browser is known as one of the most used applications - or *killer app* - and also was the first introduced when the Internet was created (1990). Which is why, it's significant maturity level is above in comparison with other developements and can assure a certain level of *trust* whenever it is used as a client with other systems.

¹Universidad Técnica Federico Santa María

Índice general

Índice general	II
Índice de figuras	III
Índice de cuadros	IV
1. Introducción	1
1.1. Contexto General	1
1.2. El Problema: Desarrollo de Software y Seguridad	2
1.3. Motivación: ¿Por qué estudiar el Browser?	4
1.4. Contribuciones	5
1.5. Metodología	6
1.6. Estructura del Documento	6
Bibliografía	7

Índice de figuras

1.1. Porcentaje de uso de Navegadores [?]	2
---	---

Índice de cuadros

Capítulo 1

Introducción

1.1. Contexto General

Entre 1989 y 1990, Tim Berners-Lee acuñó el concepto de *World Wide Web* y con ésto realizó la construcción del primer *Web Server*, *Web Browser* y las primeras páginas *Web*. Mucho antes que aparecieran los grandes sistemas que ahora conocemos, el *Web Browser* permitía navegar páginas estáticas y realizar una serie de acciones limitadas a las tecnologías de ese tiempo. En la actualidad el *browser* es la herramienta predilecta por todos, desde comprar tickets para una película, realizar reuniones por videoconferencia y muchas otras tareas que invitan a nuevas formas de interactuar y comunicar.

En el último tiempo el mercado de los *Web Browser* ha crecido bastante (Figura 1.1), principalmente debido a la robustez que éstos poseen y a la cantidad de años que llevan desarrollándose en la industria del Desarrollo de Software. Los navegadores más conocidos son: Google Chrome o su versión Open Source Chromium, Firefox, Internet Explorer, Opera y Safari; siendo los primeros 3 el enfoque de este trabajo.

La *Web 2.0* se inició con el uso intensivo de tecnologías como AJAX, y ésto ha permitido una nueva simbiosis entre el usuario, el Web Browser y el Servidor o Web Server que se comunican entre sí. El Navegador Web es una herramienta indispensable para todo tipo de tareas computacionales como comunicacionales, su existencia a penetrado completamente en las labores diarias de todos nosotros. En este mismo instante, la Web a evolucionado nuevamente obteniendo un nuevo nombre: *Web 3.0*, donde se realiza el uso de inteligencia artificial y sistemas de recomendación para generar nuevos tipos de contenido media para el usuario.

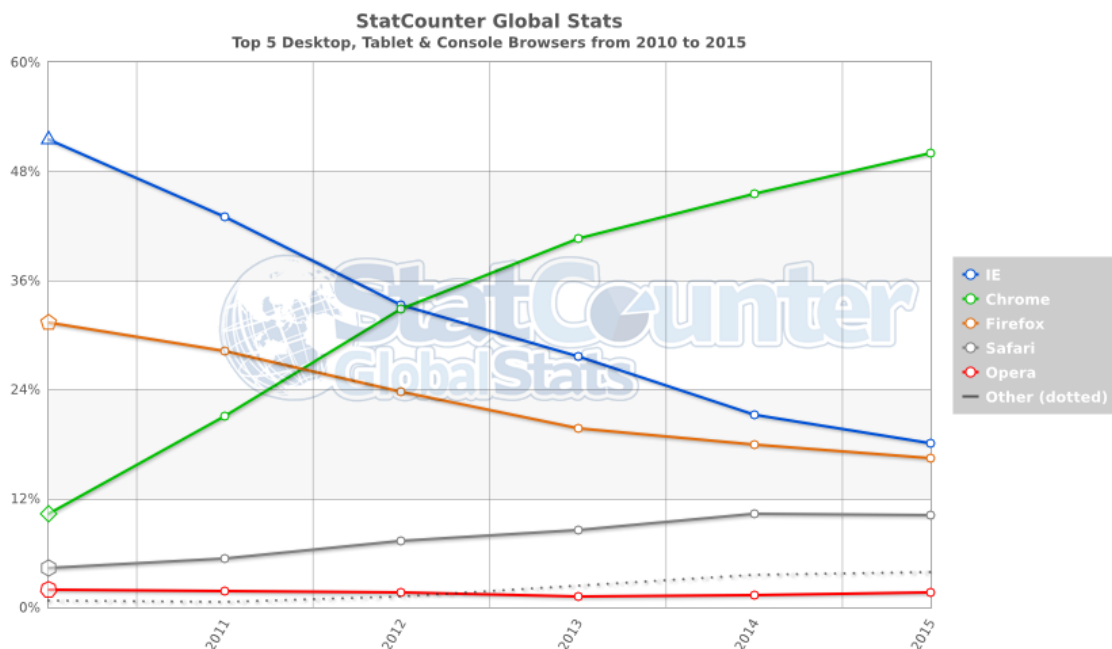


Figura 1.1: Porcentaje de uso de Navegadores [?]

1.2. El Problema: Desarrollo de Software y Seguridad

Ningún Desarrollo de Software es igual al anterior. Por cada nuevo proyecto que surge es necesario ver qué tipo de proceso es el que se usará, qué personas serán parte del grupo de trabajo, qué condiciones económicas estará expuesto, qué *Stakeholder* están pendientes de que el Proyecto salga exitoso y un sin numero de variables, no menos importantes a considerar. Por lo tanto dependiendo de lo anterior, los sistemas podrían llegar a ser simples o muy complejos. En consecuencia, se hace necesario tener ciertas metodologías que aseguren que se cumplan con todos los Requerimientos Funcionales como No-Funcionales del Sistema a construir. Sin embargo un problema que existe recurrentemente, es que la mayoría del Software construido contiene numerosos **defectos** y **errores**, generando así **vulnerabilidades** que son encontradas y explotadas por los atacantes, generando un compromiso parcial o total del sistema [1]. Lo anterior sucede frecuentemente por que los sistemas no son desarrollados teniendo en cuenta la seguridad [2, 3, 4].

Un fenómeno en la literatura llamado *Zero-day attack*, se refiere al momento clave donde un atacante explota una vulnerabilidad - hasta ese momento desconocida - de algún sistema (importante o no), y que si no es parchado lo antes posible puede com-

prometer no solo a sistemas, si no también a los usuarios que hacen uso de éste. Junto con lo anterior muchas veces ocurre que aunque se corrijan estos nuevos ataques, no todos los sistemas que podrían llegar a necesitar del mismo parche para protegerse del ataque, realizan la actualización y su adecuada configuración para así protegerse de una posible amenaza que explote la vulnerabilidad recientemente encontrada. Si bien un **Zero-day Attack** es un evento que podría no ocurrir tan repetidamente, dado que se produce por el largo estudio llevado por el atacante, sobre el sistema a vulnerar, existen otras formas de comprometer a un sistema. Muchas veces al desarrollar sistemas, se prefiere utilizar API's¹ de otros sistemas para poder incluir funcionalidades ya implementadas, fomentando así el Reuso de piezas de Software. Si bien lo anterior es una buena práctica, si el sistema no cuenta con las medidas de seguridad necesarias, estas piezas podrían ser causa de amenazas de seguridad que terminarían por corromper el sistema y en consecuencia podría causar una pérdida monetaria a los *Stakeholders*.

En general lo expuesto anteriormente ejemplifica perfectamente lo que tienen que lidiar los equipos de trabajo en proyectos de Desarrollo de Software, cuando dentro de sus preocupaciones la seguridad queda como un trabajo extra y no como parte del desarrollo completo. Bien es sabido que un proyecto en producción que presente problemas que involucren a varias entidades, el costo asociado puede llegar a ser altísimo [5], sin olvidar que podría llegar a afectar la Confidencialidad, Integridad y Disponibilidad de los datos de los involucrados con el sistema [6]. Por esto mismo, es imperante que sean entendidos, desde el comienzo, los *concerns* de los *Stakeholders* y los Requerimientos de Seguridad asociados, y que además todos los involucrados los entiendan perfectamente. Según [4] la falta de preocupación en temas de seguridad en desarrollos de software no tiene una raíz principal, diversos factores como: la falta de estudios de seguridad en las mallas curriculares de las Universidades, pocos fondos para la investigación, la falta de iniciativa en la industria, el exceso de confianza de los desarrolladores, etc., son los causantes de las futuras vulneraciones de códigos en sistemas críticos.

La literatura que habla de la construcción de *Secure Software* o Software Seguro, indica que los practicantes de Desarrollo de Software deben entender, en gran medida, los problemas de seguridad que podrían llegar a ocurrir en sus sistemas. No basta con saber como unir las piezas, no basta con que cada pieza de por si sea segura, si los componentes del sistema no actúan de forma coordinada probablemente éste no será seguro [7], dado que la seguridad es una Propiedad Sistémica que necesita ser vista de manera holística y al inicio del proceso.

¹Application Programming Interface

1.3. Motivación: ¿Por qué estudiar el Browser?

Con la aparición de la *Web 2.0 y 3.0*, con el uso de *AJAX*, inteligencia artificial y sistemas de recomendación, permitieron nuevas formas de interacción entre usuarios y sistemas, lo que causó que el browser fuera usado extensivamente en los nuevos Desarrollos de Software dado que:

- Permite disminuir los costos de construir un programa Cliente (desde cero) para el usuario del sistema.
- Actualmente la Seguridad implementada en los *Web Browser* es bastante buena, dado que existen grandes compañías que se aseguran de ello (Google, Microsoft, Mozilla entre las más conocidas).
- El *browser* es una herramienta indispensable. La mayoría de los sistemas que lo usan en la vida cotidiana son de tipo: *online banking*, declaración de impuestos, promoción de empresas o tiendas, compras, y mucho más.

Sin embargo los sistemas que dependen del uso del *Browser*, deben de tener en cuenta las posibles amenazas de seguridad a las que se enfrentarán por el solo hecho de usarlo. Para un proyecto de gran envergadura, sería un error no tener en consideración los posibles peligros que trae el uso del *Browser*, y es el deber de todo integrante del equipo de Desarrollo tener el conocimiento de la seguridad del Cliente Web. El entendimiento de la estructura subyacente del Web Browser podría asegurar que las personas que trabajen en el desarrollo, comprendan los *trade-off* al momento de diseñar un Software que necesite la colaboración del Navegador Web [8, 9, 10].

En [1] menciona que en cursos de Ingeniería de Software los estudiantes no aprenden mucho sobre Principios de Diseño en Seguridad, ni técnicas que permitan una segura implementación de código, a menos que lo necesiten en algún momento. Más aún, la falta de este tipo de conocimiento puede hacer creer que la seguridad es un requerimiento que puede o no ser tomado en cuenta al comienzo del Desarrollo. En este trabajo el enfoque es otro, la seguridad es una propiedad sistémica que debe ser tomada en cuenta desde el inicio del sistema [3, 11, 12, 7].

Este trabajo tiene una motivación principal. Ésta es ayudar a quién lo necesite con el conocimiento necesario para entender el funcionamiento y construcción del Cliente, el Web Browser, los beneficios detrás de la Seguridad implementada en el Browser y de los peligros existentes de los que nos protegen. De esta manera se espera que alguien que lea este trabajo, tanto Estudiantes como Desarrolladores de Softwares, obtengan el conocimiento necesario al momento de trabajar junto con el Navegador Web al realizar un Desarrollo de Software que dependa de éste.

1.4. Contribuciones

El Objetivo General de esta Memoria es generar un cuerpo organizado de información sobre el Web Browser y su Seguridad, de tal manera que se pueda sistematizar, organizar y clasificar el conocimiento adquirido en un documento, con formato semi-formal, tanto para Profesionales como Estudiantes del área Informática que estén insertos en el área de Desarrollo de Software.

Este trabajo busca cumplir con los siguientes Objetivos Específicos:

- Comprender los conceptos relacionados al navegador web, sus componentes, interacciones o formas de comunicación, amenazas y ataques a los que puede estar sometido, como los también los mecanismos de defensa. Esto se realizará a través de un Estado del Arte sobre el Browser.
- Construir una Arquitectura de Referencia del Web Browser e iniciar un pequeño catálogo de Patrones de Mal Uso o de Uso Indebido. Esto permitirá condensar el conocimiento obtenido en el punto anterior a través de documentos semi-formales, lo que permitirá generar una guía para comunicar los conceptos relevantes que pudieran afectar la relación existente entre un desarrollo de software y el navegador.
- Profundizar el conocimiento en ataques relacionados con métodos de Ingeniería Social.

Particularmente se ha escogido como metodología base la dada por el autor del libro [7]. Una Arquitectura de Referencia (AR) tiene como objetivo el mismo descrito en [9, 10], éste es el ayudar a los *implementors* o desarrolladores del software, a entender los *trade-off* cuando se diseñan nuevos sistemas, y puede ayudar a los mantenedores de estos sistemas a entender el código *legacy* detrás los navegadores que trabajan a mano a mano. Además una Arquitectura de Referencia permite comparar las diferencias en decisiones de diseño del Navegador y así poder entender los cambios realizados a lo largo del Desarrollo de un sistema. Junto con lo anterior, la AR permitirá tener una visión holística del sistema y mostrará las decisiones de alto nivel para asegurar la Seguridad del sistema. Por otra parte, los Patrones de Mal Uso o Uso Indebido, permitirán enseñar y comunicar las posibles formas en que tal sistema puede ser usado inapropiadamente.

En este trabajo se presentará nuestra Arquitectura de Referencia y 2 Patrones de Uso Indebido, que usarán la AR contruida para mostrar los componentes y mensajes que una amenaza puede realizar, con tal de lograr un ataque en el Browser. Estos patrones serán presentados usando el template POSA [13] y UML, para así modelar las interacciones entre los diversos componentes de la arquitectura.

1.5. Metodología

Este trabajo se realizará de la siguiente forma:

1. Introducción de un *Framework conceptual* para entender los conceptos relacionados.
2. Contruir un Estado del Arte sobre el Browser, especialmente sobre la seguridad de éste.
3. Identificar los conceptos, actores, componentes, interacciones y funciones, en relación al tema principal.
4. Construir patrones de arquitectura que definan los componentes y responsabilidades, con el objetivo final de ser unidos en una Arquitectura de Referencia.
5. Contruir patrones de Mal Uso/Uso Indebido por medio del punto anterior.

1.6. Estructura del Documento

El presente documento trata del trabajo de Memoria que se divide en las siguientes partes:

- En el capítulo ??...
- Luego de tener un extenso conocimiento de lo que actualmente es conocido como **Web Browser**, el capítulo ??

Bibliografía

- [1] Karen M Goertzel, Theodore Winograd, Holly L McKinley, Lyndon J Oh, Michael Colon, Thomas McGibbon, Elaine Fedchak, and Robert Vienneau. Software security assurance: A state-of-art report (sar). Technical report, DTIC Document, 2007.
- [2] J. Yoder, J. Yoder, J. Barcalow, and J. Barcalow. Architectural patterns for enabling application security. *Proceedings of PLoP 1997*, 51:31, 1998.
- [3] Eduardo B Fernandez. A methodology for secure software design. In *Software Engineering Research and Practice*, pages 130–136, 2004.
- [4] Bill Whyte and John Harrison. State of Practice in Secure Software: Experts’ Views on Best Ways Ahead. IGI Global.
- [5] Carnegie Mellon University Computer Emergency Response Team. Early identification reduces total cost (segment from cert’s podcasts for bussiness leaders).
- [6] Mike Hicks. Interview to **Kevin Haley** (from **Symantec**), 2014. Mike Hicks (Profesor of Software Security course in Coursera.org).
- [7] Eduardo Fernandez-Buglioni. *Security patterns in practice: designing secure architectures using software patterns*. John Wiley & Sons, 2013.
- [8] M.M. Larrondo-Petrie, K.R. Nair, and G.K. Raghavan. A domain analysis of web browser architectures, languages and features. In *Southcon/96. Conference Record*, pages 168–174, Jun 1996.
- [9] Alan Grosskurth and Michael W. Godfrey. A reference architecture for web browsers. pages 661–664, 2005. URL: <http://grosskurth.ca/papers.html#browser-refarch>.
- [10] Alan Grosskurth and Michael W. Godfrey. Architecture and evolution of the modern web browser. URL: <http://grosskurth.ca/papers.html#browser-archevol>. Note: submitted for publication.

- [11] Eduardo B Fernandez, Michael VanHilst, Maria M Larrondo Petrie, and Shihong Huang. Defining security requirements through misuse actions. In *Advanced Software Engineering: Expanding the Frontiers of Software Technology*, pages 123–137. Springer US, 2006.
- [12] Fabricio A Braz, Eduardo B Fernandez, and Michael VanHilst. Eliciting security requirements through misuse activities. In *Database and Expert Systems Application, 2008. DEXA '08. 19th International Workshop on*, pages 328–333. IEEE, 2008.
- [13] Frank Buschman, Regine Meunier, Hans Rohnert, Peter Sommerlad, and Michael Stal. A system of patterns: pattern-oriented software architecture, 1996.