Modelado y validación de la seguridad de la información en el aspecto de integridad de los datos desde el punto de vista de las Arquitecturas de Software por medio del uso del lenguaje Secure xADL

Jaime Raymundo Martínez Solís

Dra. Perla Velasco Elizondo

Modelado y validación de la seguridad de la información en el aspecto de integridad de los datos desde el punto de vista de las Arquitecturas de Software por medio del uso del lenguaje Secure xADL

Prepropuesta de Investigación Doctoral

Elaborada por:

Jaime Raymundo Martínez Solís

Centro de Investigación en Matemáticas

Miembros del comité evaluador

Dr. Cuauthémoc Lemus Olalde

Dr. Carlos Montes de Oca Vázquez

Dra. Perla Velasco Elizondo

Índice de contenido

Introducción

Hoy en día los sistemas de información forman parte de nuestra vida cotidiana, estamos rodeados por ellos en la casa, en el trabajo e incluso los llevamos con nosotros en los teléfonos celulares y en los automóviles. En gran medida dependemos de ellos, dependemos de que la información que nos proporcionan sea correcta y que la podamos obtener en el momento en que la necesitamos.

Con la llegada del Internet los sistemas de información se conectan unos con otros proporcionándonos formas de procesar y utilizar la información de maneras que no se podrían haber imaginado con antelación.

Uno de los aspectos negativos, resultado de la misma conectividad al internet y de la dependencia que tenemos de los sistemas de información, es que también se han abierto puertas para ser atacados con las consecuencias de la pérdida de información o la pérdida de confidencialidad de datos sensibles. Esto ha sido en gran parte debido a que los sistemas no han sido diseñados adecuadamente para soportar o repeler los ataques.

En el presente documento se hace la propuesta de investigación de cómo incorporar el aspecto de Integridad de la información dentro de las arquitecturas de software y cómo evaluar en la misma arquitectura de software si se cumple o no con ese requisito.

Este documento está organizado de la siguiente manera:

En la primer parte se describe el marco teórico donde se explica de manera breve qué es y de qué trata la seguridad de la información. Posteriormente se describe el problema que se desea resolver con la investigación y los trabajos previos en el área de la seguridad en las arquitecturas de software.

En la segunda parte se describe la propuesta de investigación, el objetivo que se pretende alcanzar, las metas que se deben cumplir para alcanzar el objetivo y la justificación de porqué es importante realizar esta investigación. Después se da a conocer la hipótesis, la metodología que se llevará a cabo, el alcance y las limitaciones, la forma en que se validará el resultado de la investigación y los productos que se habrán de obtener mientras se lleva a cabo el estudio doctoral.

Definición del problema

Marco Teórico

Hoy en día todos nos encontramos rodeados por sistemas informáticos, en la casa, en el trabajo, en los centros comerciales e incluso los llevamos con nosotros en los teléfonos celulares o en las agendas electrónicas. Al hacer uso de esos sistemas informáticos confiamos que la información que maneja se encuentra segura y sin embargo continuamente se escuchan noticias de que la información no está tan segura como debiera ser. Por ejemplo, en [3] se describen los problemas referentes a la seguridad de información que han ocurrido día con día en el lapso de 10 años, de 1998 a 2008. Es por ello que podemos afirmar que la seguridad de la información es un tema importante dentro del desarrollo de software que no ha sido resuelto.

Adicionalmente, los sistemas informáticos actualmente se interconectan unos con otros dando pie a que la seguridad se vea comprometida. Un ejemplo claro de lo anterior es el caso de Internet, el cual en sus inicios fue concebido como una red confiable que permitía intercambiar información a las instituciones educativas conectadas a ella. Actualmente a la red Internet se le considera un medio no confiable. La Internet es de ámbito global, pero esta red es un medio inseguro[11].

La seguridad de la información se basa en tres pilares: Integridad, Confidencialidad y Disponibilidad.

Integridad: La información tiene el atributo de integridad cuando es oportuna, precisa, completa. La integridad de los datos es el requerimiento de que la información y los programas sean cambiados únicamente en la manera especificada y autorizada. La integridad del sistema es el requerimiento de que un sistema “ejecute sus funciones libre de manipulaciones no autorizadas, ya sea de manera deliberada o inadvertidas por parte del sistema”.

Disponibilidad: Es el requerimiento de que se pueda garantizar que el sistema trabaje de manera inmediata y no sea denegadas sus funciones a los usuarios autorizados.

Confidencialidad: Es el requerimiento de que la información confidencial no sea expuesta a individuos no autorizados[20].

En la ingeniería de software los tres atributos antes descritos, integridad, disponibilidad y confidencialidad, son listados también como atributos no funcionales de los sistemas[10][15].

Los atributos no funcionales deben ser tomados en cuenta en etapas tempranas del ciclo de vida del software pues cuando se intentan aplicar cuando el desarrollo del software está muy avanzado puede llegar a ser muy difícil poderlos obtener. Dentro del ciclo de vida del desarrollo del software la fase en la cual se deben considerar los atributos no funcionales es la de la definición de la Arquitectura del Software.

Entre las diferentes definiciones de Arquitectura de Software que podemos encontrar en la literatura tenemos:

“La arquitectura del software es un cuerpo de métodos y técnicas que nos ayudan a manejar la complejidad del desarrollo del software”[17].

“(la arquitectura)... es un conjunto de componentes y las conexiones entre ellos, la arquitectura de software de un programa o un sistema de cómputo es la estructura de estructuras del sistema, los cuales incluyen a los elementos de software, las propiedades visibles externamente de esos elementos y las relaciones entre ellos.”[10].

Las arquitecturas de software no son diagramas, aunque pueden ser representados con diagramas. Un ejemplo de lo anterior es el estilo de descripción de arquitecturas conocido como “4+1 views”[13], en el cual se hace uso de distintos diagramas para mostrar diferentes puntos de vista de la arquitectura de un sistema. Otra manera de poder describir las arquitecturas de software es por medio de los lenguajes de descripción de arquitecturas (ADLs), varios de los cuales son comparados y descritos en [2] y [12].

En [17] se establece que los ADLs describen a los sistemas a un nivel de abstracción en el cual se puede razonar acerca de la manera en que los atributos de calidad (requerimientos no funcionales incluyendo la seguridad de información) son afectados dependiendo de la manera en que son dispuestos físicamente los elementos del sistema del software. Dado que en los lenguajes de descripción de arquitecturas de software se capturan los elementos y las características de la arquitectura de un software con un nivel más alto de formalismo que con los diagramas, se puede hacer uso de ese formalismo para verificar su correctitud. Se pueden usar diferentes métodos formales para validar una arquitectura, por ejemplo la teoría de conjuntos [5], CSP, CSS [1], mu-calculus [19].

La mayor parte de las investigaciones en torno a la seguridad se han enfocado en como proveer a los sistemas de software de confidencialidad, integridad y disponibilidad, pero lo han hecho desde un punto de vista de muy bajo nivel, por ejemplo algoritmos de cifrado, uso de cortafuegos y protocolos de comunicación. Poco se ha investigado en cómo llevar el resultado de esas investigaciones al nivel de la arquitectura de software.

Definición del problema

El problema de la falta de seguridad en los sistemas de información no es un problema que sea nuevo, ha sido estudiado desde hace muchos años. Los problemas de seguridad no solo se han mantenido con los años, puede decirse que el problema se ha agudizado. Ahora no solo tenemos que cuidarnos de los virus computacionales sino que las amenazas han evolucionado pues estamos siendo constantemente atacados por amenazas que se engloban en el término de malware, los cuales incluyen keyloggers, hoaxes, worms y, phishers entre otros. Todos estos tipos de malware comprometen la seguridad de la información [21].

Los sistemas que fueron creados para trabajar originalmente en un ambiente confiable ahora son trasladados a ambientes no confiables, incluso se comunican con otros sistemas de los que no siempre se puede asegurar que no sean hostiles.

Las fallas provocadas de manera intencional son más difíciles de prever pues detrás de su aparición está la mente de una persona que hace que sucedan [14].

Una de las razones por las cuales no se incluyen de manera efectiva los aspectos de seguridad de la información en las arquitecturas de software es porque normalmente, las personas encargadas de la seguridad de la información no diseñan a los sistemas, y quienes diseñan a los sistemas no son encargados de la seguridad de la información.

Por ejemplo, los desarrolladores de software están acostumbrados a trabajar con los casos de uso, es decir con el flujo “correcto” en la que debe trabajar el software, mientras que los encargados de la seguridad del software pueden conocer o estar más conscientes de la manera “incorrecta” en la que puede ser utilizado el software (misuse cases)[9]. Si los diseñadores no toman en cuenta los misuse cases pueden realizar un diseño débil y propenso a ser vulnerado por usuarios maliciosos.

Es quizás debido a este enfoque de los arquitectos de software de pensar principalmente en el uso correcto del software por el cual no se han realizado estudios suficientes para añadir los aspectos de seguridad dentro de las arquitecturas de software.

Trabajos previos

Algunos de los trabajos referentes al tratamiento de la seguridad desde las arquitecturas de software son listados a continuación:

UMLsec: Extensión a UML en el cual se encapsulan conocimientos de ingeniería en seguridad con lo cual se hacen disponibles a los desarrolladores que no están familiarizados con la seguridad. La validación de los modelos realizados por medio de UMLsec se efectúan por medio de cálculo-lambda. Se han creado herramientas para apoyar a la verificación de modelos UMLsec [6][7].

secureUML: Diseñado originalmente para integrar las especificaciones de control de acceso en los modelos de las aplicaciones. Utiliza el enfoque RBAC (Role Based Access Control)[18].

Aspect Oriented Modeling: En [4] se usó el modelado orientado a aspectos para añadir los aspectos de seguridad de la misma manera en que se tejen otros aspectos.

Secure xADL: Es un lenguaje ADL que se basa en XML. Soporta el modelado de arquitecturas en tiempo de diseño y en tiempo de ejecución. Secure xADL es una extensión a xADL. Fue diseñado para modelar aspectos de la seguridad del software, específicamente el control de acceso[8].

Propuesta de investigación

Investigación propuesta

En este documento propongo hacer la investigación de cómo incluir y validar en las arquitecturas de software el aspecto de la integridad de la información desde el punto de vista de la seguridad del software. La validación deberá ser realizada a través de un método formal.

La metodología propuesta consta de los siguientes pasos:

1. Obtención de los requerimientos de integridad de los datos
2. Modelado inicial de la arquitectura
3. Inclusión del aspecto de la integridad de los datos en la arquitectura
4. Evaluación de la arquitectura modificada

Objetivos de la investigación

El objetivo principal de la investigación es el de crear una extensión al lenguaje de descripción de arquitecturas Secure xADL para añadirle el aspecto de integridad de la información junto con un método de validación de la arquitectura.

Para lograr el objetivo principal se identificaron las siguientes metas.

Meta 1: Diseñar la extensión al lenguaje xADL

Meta 2: Diseñar una metodología para la identificación de los puntos vulnerables en donde se deban aplicar de manera explícita las extensiones propuestas al lenguaje xADL con la finalidad de establecer el cumplimiento de los requerimientos de integridad

Meta 3: Diseñar una metodología para la evaluación del cumplimiento de las restricciones de integridad de la información en la arquitectura modelada

Meta 4: Aplicar y validar la metodología propuesta.

Las metas anteriores podrán ser cumplidas por medio de las siguientes actividades específicas:

Meta 1.

1. Identificar los conceptos principales referentes a la integridad de los datos, por ejemplo: Permisos, roles, sujetos, objetos y acceso.
2. Integrar los conceptos anteriores al lenguaje Secure xADL

Meta 2.

1. Estudio de los casos de uso, los casos de uso incorrectos (misuse cases) y la arquitectura propuesta original para localizar los puntos vulnerables en los que se puede afectar la integridad de los datos.
2. Diseño de la metodología para incluir en la arquitectura del software el aspecto de integridad de los datos

Meta 3.

1. Estudio de los métodos formales para localizar cuál de ellos es el que es más conveniente para la evaluación de la arquitectura modificada.
2. Creación de la metodología para la evaluación de la arquitectura modificada.

Meta 4.

1. Definir una muestra de arquitecturas de software de prueba en la cual se puedan aplicar las metodologías de modelado y evaluación de arquitecturas de software propuestas.
2. Aplicar a las arquitecturas de software de prueba las metodologías propuestas
3. Evaluar el resultado de la aplicación de las metodologías propuestas.

Justificación

Siendo la seguridad de la información es un aspecto importante en todo sistema de software es conveniente incorporarla desde la arquitectura. Normalmente las consideraciones referentes a la seguridad son integradas de manera tardía en el desarrollo de software. La incorporación de los aspectos de seguridad toman una actitud reactiva, una vez que ocurre un problema de seguridad se agrega un parche al sistema [16].

Por lo general no se incorporan de manera explícita los elementos de seguridad dentro de las arquitecturas de software, y por lo tanto no es fácil razonar acerca de la seguridad en las arquitecturas de software.

Hasta donde sé, no se ha estudiado cómo incluir en las arquitecturas de software el aspecto de integridad de los datos ante ataques premeditados.

Marco conceptual

Hipótesis de la investigación

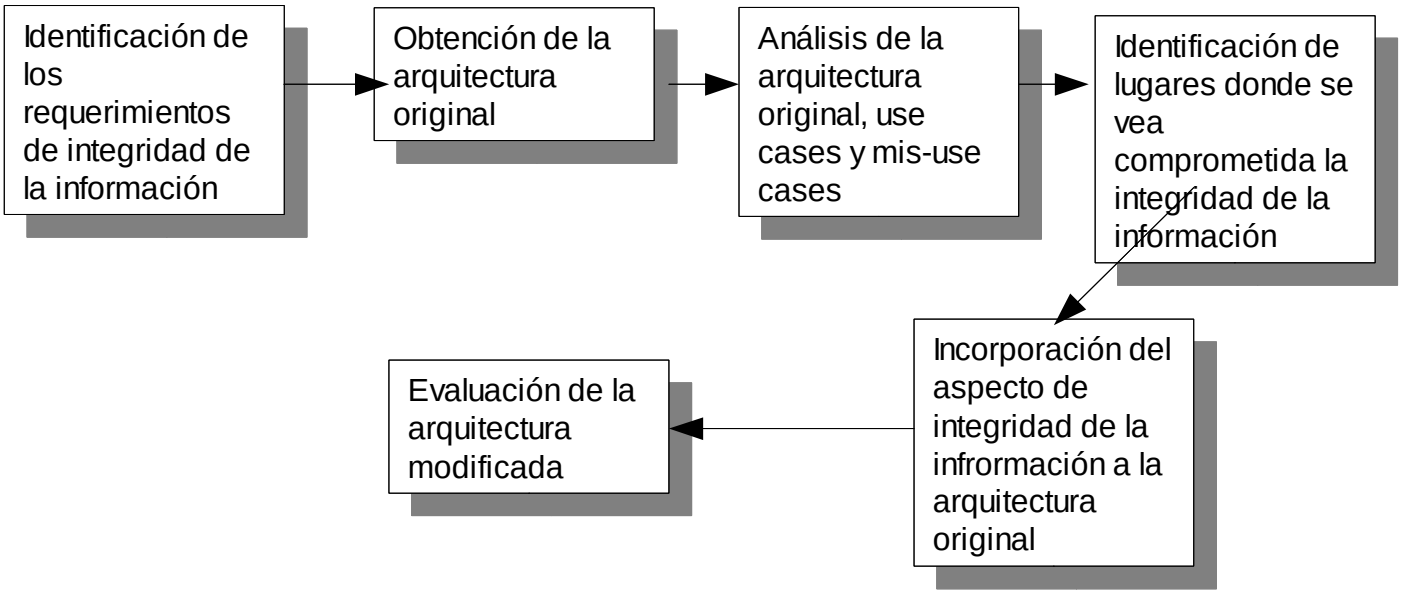
Es posible evaluar desde la arquitectura del software el aspecto de integridad de la información.

Las suposiciones para la hipótesis son:

1. En la arquitectura del software se pueden identificar los puntos débiles en los cuales se puede ver comprometida la integridad de la información.
2. Se pueden extender los lenguajes de descripción de arquitecturas para incluir los aspectos de seguridad del software, específicamente el aspecto de integridad de los datos.
3. Se puede evaluar a través de métodos formales y con ello validar que una arquitectura de software asegura la integridad de los datos.

Metodología

La metodología propuesta consta de las siguientes fases



Alcance y limitaciones

Dado que el propósito de la investigación es el de evaluar en la arquitectura del software el aspecto de integridad de los datos, los aspectos de confidencialidad y disponibilidad quedan fuera del ámbito de este estudio.

El ámbito sobre el cual se desarrolla la investigación será exclusivamente el de la arquitectura del software, por lo cual no se toma en consideración aspectos externos al software, por ejemplo aspectos culturales y organizacionales.

Validación de la investigación

Para poder validar la investigación se propone aplicar la metodología en dos casos de prueba de ejemplo y en por lo menos dos casos reales de la industria.

Los casos de prueba desarrollados para este fin tendrán las siguientes características:

Caso de prueba 1.

Arquitectura de un sistema no distribuido. Es decir, que se ejecutará en su totalidad en un solo equipo de cómputo

Caso de prueba 2.

Arquitectura de un sistema distribuido en tres capas (three – tiers)

Los casos de prueba reales de la industria serán acordados con la(s) empresa(s) que desee(n) apoyar con la investigación.

Productos de la investigación

Esta propuesta de investigación proveerá de los siguientes productos durante su desarrollo. Los productos son mostrados con cada meta indicada en el punto 2 del documento.

Meta	**Tareas**	**Producto**
1: Diseñar la extensión al lenguaje Secure xADL	1. Identificar los conceptos principales referentes a la integridad de los datos, por ejemplo: Permisos, roles, sujetos, objetos y acceso. 2. Integrar los conceptos anteriores al lenguaje Secure xADL	Ampliación al lenguaje Secure xADL (reporte)
2: Diseñar una metodología para la identificación de los puntos vulnerables en donde se deban aplicar de manera explícita las extensiones propuestas al lenguaje xADL con la finalidad de establecer el cumplimiento de los requerimientos de integridad	3. Estudio de los casos de uso, los casos de uso incorrectos (mis-use cases) y la arquitectura propuesta original para localizar los puntos vulnerables en los que se puede afectar la integridad de los datos. 4. Diseño de la metodología para incluir en la arquitectura del software el aspecto de integridad de los datos	Metodología para la obtención de vulnerabilidades de seguridad identificadas desde la arquitectura de software (reporte)
3: Diseñar una metodología para la evaluación del cumplimiento de las restricciones de integridad de la información en la arquitectura modelada	5. Estudio de los métodos formales para localizar cuál de ellos es el que es más conveniente para la evaluación de la arquitectura modificada. 6. Creación de la metodología para la evaluación de la arquitectura modificada.	Metodología para la evaluación formal de la integridad de los datos en una arquitectura de software. (reporte)
4: Aplicar y validar la metodología propuesta.	7. Definir una muestra de arquitecturas de software de prueba en la cual se puedan aplicar las metodologías de modelado y evaluación de arquitecturas de software propuestas. 8. Aplicar a las arquitecturas de software de prueba las metodologías propuestas 9. Evaluar el resultado de la aplicación de las metodologías propuestas.	Selección de casos de prueba, aplicación de las metodologías propuestas, obtención, interpretación y validación de los resultados. (reporte)

Importancia de la investigación

La presente propuesta de investigación ofrecerá los siguientes beneficios:

1. Mostrará una manera de poder incorporar aspectos de seguridad de la información desde la perspectiva de las arquitecturas de software
2. Aportará de manera específica cómo incorporar el aspecto de integridad de la información en una arquitectura de software
3. Como resultado colateral de la investigación se tendrá una metodología para poder identificar desde la arquitectura de software las vulnerabilidades que pueden ser explotados para comprometer la integridad de los datos.
4. Aportará una metodología para evaluar puntualmente el aspecto de la integridad de la información

Conclusiones

Este documento explica la pre-propuesta de investigación doctoral que tiene como objetivo el desarrollar una ampliación al lenguaje de descripción de arquitectura Secure xADL con la finalidad de incluirle el aspecto de integridad de la información junto con una metodología de aplicación y evaluación de las arquitecturas de software que la apliquen.

Dado que actualmente el aspecto de seguridad de la información no ha sido tratado con profundidad dentro del área de las arquitecturas de software, la investigación propuesta producirá contribuciones importantes al área de las arquitecturas de software.

Referencias

[1] Apostolos Zarras, Christos Kloukinas, Valerie Issarny Quality analisis of dependable systems: A developer Oriented approach
[2] Ariel D. Fuxman A Survey of Architecture Description Languages
[3] Bruce Schneider, Eugene Kaspersky, Johannes Ullrich, Juan Carlos G. Cuartango, Mikel Urizarbarrena. Hispasec una al día 1998-2008
[4] Huiqun Yu, Dongmei Liu, Xudong He, Li Yang, Shu Gao Secure Software Architectures Design by Aspect Orientation
[5] Huiqun Yu, Dongmei Liu,Xudong He, Li Yang, Shu Gao Secure Software Architectures Design by Aspect Orientation
[6] Jan Jurjens UMLsec: Extending UML for secure systems development
[7] jan jurjens, P. Shabalin Towards Tool Support for UMLsec
[8] Jie Ren, Richard N. Taylor A secure software Architecture Description language
[9] Joshua Pauli, Dianxiang Xu Misuse Case-Based Design and Analysis of Secure Software Architecture

[10] Len Bass, Paul Clements, Rick Kazman Software Architecture in practice
[11] Man Young Rhee Internet Security Cryptographic Principles, Algorithms and Protocols
[12] Nenad Medvidovic and Richard N. Taylor A Framework for Classifying and Comparing Architecture Description Languages
[13] P Kruchten The 4+1 View Model of Architecture
[14] Paulo Esteves Verissimo, Nuno Ferreira Neves, and Miguel P u p o Correia Intrusion-Tolerant Architectures:Concepts and Design
[15] Raphael Malveau, Thomas J. Mowbray Software Architect Bootcamp
[16] Siv Hilde Houmb, Jan Jurjens developing secure networked web-based systems
[17] Stephen T. Albin The art of software architecture
[18] Torsten Lodderstedt, David Basin, and Jürgen Doser SecureUML: A UML-Based Modeling Language for Model-Driven Security
[19] Zheng Qin, Jiankuan Xing, Xiang Zheng Softwar Architecture
[20] varios An Introduction to Computer Security: The NIST Handbook,Special Publication 800-12
[21] varios Information.Security.Management.Handbook.Vol 4