Comparativa de Métricas de Seguridad de Diseño Software

Daniel Mellado[1] y Eduardo Fernández-Medina[2]

[1] Agencia Estatal de Administración Tributaria,
Paseo de la Castellana, 108, 28046 Madrid, España
damefe@esdebian.org
[2] Universidad de Castilla – La Mancha, Departamento de Tecnologías y Sistemas de Información, Grupo de Investigación GSyA
Paseo de la Universidad 4, 13071 Ciudad Real, España.
Eduardo.FdezMedina@uclm.es

Abstract. Sin métricas de seguridad no podríamos medir el éxito de las políticas, mecanismos e implementaciones de seguridad, ni tampoco se podría mejorar la seguridad si no se pudiera medir. Por lo tanto, es ampliamente admitida la importancia que tiene la utilización de métricas para la calidad de la seguridad. Sin embargo, la definición de métricas de seguridad se trata de una disciplina que está aún dando los primeros pasos, y de la que hasta ahora no había muchos recursos documentados o trabajos centrados en ella. Es por ello que en este artículo se estudian los últimos modelos existentes que definan métricas de seguridad y sus componentes como aspectos que inciden en la calidad de los productos software. A fin de que sirva como base para seguir avanzando en la investigación en esta área de conocimiento.

Keywords: Seguridad, métricas, medidas, métricas de seguridad, diseño.

1. Introducción

La tendencia actual de sistemas de información cada vez más grandes y distribuidos por todo Internet ha hecho que surjan muchas amenazas nuevas de seguridad [21], haciendo que los actuales sistemas sean vulnerables a multitud de amenazas y ciber-ataques de ciber-terroristas, hackers, etc. como por ejemplo los virus que se propagan a través de Internet, ataques de ingeniería social (*phising*, etc.) o el uso inadecuado de los recursos de esta red por empleados de las empresas [4].

De hecho, la seguridad informática ha sido un campo que ha crecido enormemente desde los años 70, dando lugar a una gran cantidad de técnicas, modelos, protocolos, etc., que han venido acompañados también de una actividad muy pronunciada por parte de las organizaciones internacionales de normalización y certificación. Tanto es así, que como se indica en [13], se pueden encontrar numerosas organizaciones internacionales de estandarización que han producido una compleja estructura de estándares relativos a temáticas relacionadas con la seguridad informática, que cambian y se actualizan con mucha frecuencia.

Es ampliamente aceptado que las métricas son importantes para la seguridad de la información, ya que sin métricas de seguridad no podríamos medir el éxito de las políticas, mecanismos e implementaciones de seguridad, no se puede mejorar la seguridad si no se puede medir. Es por ello que las métricas pueden ser una herramienta efectiva para los profesionales de seguridad de las TIC a la hora de medir la fortaleza de la seguridad y los niveles de sus sistemas, productos, procesos y preparación para gestionar los aspectos de seguridad en los que están imbuidos. Las métricas también pueden ayudar en la identificación de vulnerabilidades del sistema proporcionando una guía para la priorización de acciones correctivas y en elevar el nivel de concienciación de la seguridad en la organización [26]. Este hecho ha dado lugar a que incluso en diversas leyes, normas y regulaciones se citen a las métricas de seguridad como requisito, como por ejemplo en Estados Unidos el "Federal Information Security Management Act" o el "Clinger-Cohen Act" entre otros.

Con lo que, dada la importancia que tiene la utilización de métricas para la calidad de la seguridad, la mayoría de los requisitos de calidad o no funcionales han sido estudiados y medidos extensamente. Incluso, en lo relativo al atributo de seguridad se han definido métricas para poder valorar la seguridad a nivel de sistema y a nivel de implementación de código. Se han publicado diversas normas y estándares relativos a métricas de seguridad, como los Criterios Comunes [11], ISO/IEC 27004 [12], NIST 800-55 [25] o FIPS 140-1/2 [6]. Aunque estas normas y estándares son amplios y con definiciones imprecisas de métricas de seguridad o demasiado limitadas como para cubrir una variedad amplia de situaciones de seguridad [26]. Pero la seguridad es difícil de medir, lo que provoca que haya una alta inestabilidad de métricas de seguridad, ya que la medición de la seguridad, es decir, definición de métricas de seguridad, se trata de una disciplina que está aún dando los primeros pasos, y de la que hasta ahora no había muchos recursos documentados o trabajos centrados en ella [27].

De esta forma, el objetivo de este artículo es analizar los modelos existentes que definan métricas de seguridad y sus componentes como aspectos que inciden en la calidad de los productos software.

Para ello, se ha organizado el resto del documento del siguiente modo: En la sección dos se presentan algunas de las propuestas de métricas de seguridad más relevantes. Posteriormente, en la tercera sección se analizan desde un punto de vista comparativo todas las métricas de seguridad que proponen los trabajos estudiados en la sección anterior. Por último, en la sección cuatro se presentan las principales conclusiones.

2. Métricas de Seguridad Relevantes

En esta sección se describirán los aspectos más importantes de las propuestas de métricas de seguridad del diseño definidos en los principales estándares más aceptados sobre seguridad y calidad del software y propuestas o marcos de trabajo más destacados.

2.1. Métricas de seguridad para diagramas de clases orientados a objetos [1]

En esta propuesta se centran en la seguridad del diseño de aplicaciones orientadas a objetos, de tal forma que define una serie de métricas para este tipo de aplicaciones. Estas métricas permiten a los diseñadores descubrir y solucionar vulnerabilidades de seguridad en fases tempranas del desarrollo software, y ayuda a comparar la seguridad de las distintas alternativas de diseño. En particular, los autores proponen siete métricas de seguridad para medir la encapsulación de los datos (accesibilidad) y cohesión (interacción) para una determinada clase desde la perspectiva de pérdidas potenciales de información. Estas métricas que proponen son las siguientes: Classified Instance Data Accessibility (CIDA); Classified Class Data Accessibility (CCDA); Classified Operation Accessibility (COA); Classified Mutator Attribute Interactions (CMAI); Classified Accessor Attribute Interactions (CAAI); Classified Attributes Interaction Weight (CAIW); Classified Methods Weight (CMW). El detalle de estas métricas se explica en la Sección 3 , ya que en nuestra propuesta de métricas de seguridad de diseño se incluyen éstas.

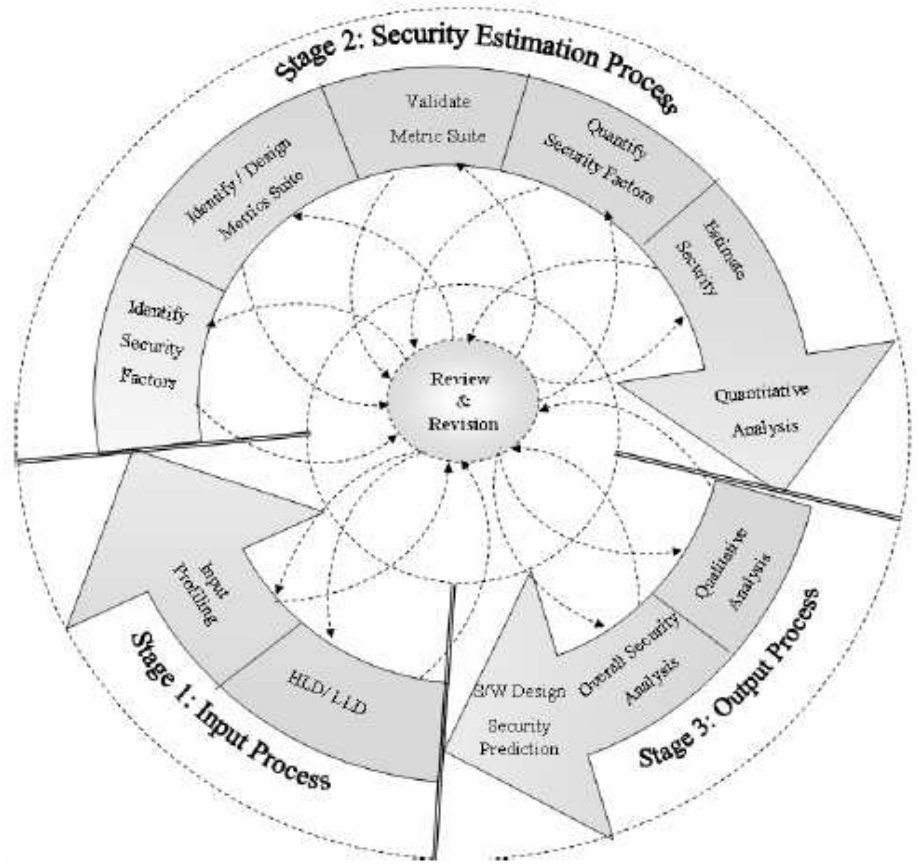


Fig. 1 - Ciclo de vida de la estimación de seguridad

2.2. Marco de trabajo para la estimación de la seguridad: perspectiva de la fase de diseño [3]

En este trabajo los autores proponen un marco de trabajo para estimar la seguridad del software desde las primeras fases del ciclo de vida de desarrollo del software, de tal manera que dicho marco de trabajo permita a los profesionales de seguridad estimar la seguridad del software y mitigar las vulnerabilidades en la fase de diseño. El marco de trabajo propuesto plantea un proceso de estimación de la seguridad del software que consta de las siguientes fases (en la Fig. 1 se muestra un diagrama del proceso propuesto):

1. Identificar los factores de seguridad
2. Identificar / Diseñar el juego de métricas
3. Validar el juego de métricas
4. Cuantificar los factores de seguridad
5. Estimar la seguridad

2.3. Criterios Comunes o ISO/IEC 15408 [11]

Los Criterios Comunes (CC) que en la actualidad se encuentran estandarizados bajo la serie de normas ISO/IEC 15408, tienen su origen en 1990 y surgen como resultado de la armonización de los criterios sobre seguridad de productos software ya utilizados por diferentes países con el fin de que el resultado del proceso de evaluación pudiese ser aceptado en múltiples países. Los CC permiten comparar los resultados entre evaluaciones de productos independientes. Para ello, se proporcionan un conjunto común de requisitos funcionales para los productos de TI (Tecnologías de la Información). Estos productos pueden ser hardware, software o firmware. El proceso de evaluación establece un nivel de confianza en el grado en el que el producto TI satisface la funcionalidad de seguridad de estos productos y ha superado las medidas de evaluación aplicadas. Los CC son útiles como guía para el desarrollo, evaluación o adquisición de productos TI que incluyan alguna función de seguridad.

Con el fin de poder certificar un producto según los Criterios Comunes se deben comprobar, por parte de uno de los laboratorios independientes aprobados, numerosos parámetros de seguridad que han sido consensuados y aceptados por 22 países de todo el mundo. El proceso de evaluación incluye la certificación de que un producto software específico verifica los siguientes aspectos:

- Los requisitos del producto están definidos correctamente.
- Los requisitos están implementados correctamente.
- El proceso de desarrollo y documentación del producto cumple con ciertos requisitos previamente establecidos.

Los Criterios Comunes establecen entonces un conjunto de requisitos para definir las funciones de seguridad de los productos y sistemas de Tecnologías de la Información y de los criterios para evaluar su seguridad. El proceso de evaluación, realizado según lo prescrito en los Criterios Comunes, garantiza que las funciones de

seguridad de tales productos y sistemas reúnen los requisitos declarados. Así, los clientes pueden especificar la funcionalidad de seguridad de un producto en términos de perfiles de protección estándares y de forma independiente seleccionar el nivel de confianza en la evaluación de un conjunto definido desde el EAL1 al EAL7

Los niveles de confianza en la evaluación definidos en el ISO/IEC 15408 van desde EAL1 (el menor) a EAL 7 (el mayor) y se definen de forma acumulativa (verificaciones de nivel n+1 implican realizar las de nivel n, $1 \leq n \geq 7$).

Los niveles EAL 5 al 7 incluyen modelos y demostraciones semi-formales y formales por tanto, se aplican a productos con objetivos de seguridad muy específicos (entorno militar, por ejemplo). Por otra parte, estos niveles requieren de la generación de una gran cantidad de documentación durante el proceso de desarrollo que debe entregarse al evaluador para que éste pueda confirmar la información. Finalmente, para la aplicación de los Criterios Comunes, existe una metodología con los criterios a evaluar para cada uno de los niveles de confianza estandarizada por la Norma ISO/IEC 18045 (ISO 18045, 2008) y denominada CEM (Common Methodology for IT Security Evaluation).

2.4. ISO/IEC 27004 [12]

La familia de normas ISO/IEC 27000 está compuesta de un conjunto de documentos, todos ellos relacionados con la gestión de la seguridad. En concreto, la 27000 incluye la definición de un vocabulario común sobre gestión de seguridad, la 27001 proporciona un modelo para establecer, implementar, operar, controlar, revisar, mantener y revisar los sistemas de gestión de seguridad de la información, la 27002 ofrece un código de buenas prácticas, la 27003 unas guías de implantación, la 27004 es relativa a métricas para la gestión de la seguridad, la 27005 es sobre gestión de riesgos, la 27006 muestra un cuerpo para la certificación de la seguridad, y la 27007 ofrece guías de auditoría. Esta familia de normas (todavía incompleta) representa un esfuerzo por la agrupación y unificación de estándares relativos a la gestión de la seguridad, y que se pretende que sea modelo de referencia en el futuro.

El empleo de este estándar ISO/IEC 27004 permitirá a las organizaciones dar respuesta a los interrogantes de cuán efectivo y eficiente es el SGSI (Sistema de Gestión de la Seguridad de la Información) y qué niveles de implementación y madurez han sido alcanzados. Estas mediciones permitirán comparar los logros obtenidos en seguridad de la información sobre períodos de tiempo en áreas de negocio similares de la organización y como parte de continuas mejoras.

Este estándar define el ámbito, como una guía sobre la especificación y uso de técnicas de medición, para proveer precisión en la observación del SGSI en cualquier tipo de organizaciones y con el propósito de crear una base para recolectar, analizar y comunicar datos relacionados a este SGSI, los cuales serán empleados para tomar decisiones que permitan mejorar el mismo.

Se basa sobre el modelo PDCA (Plan – Do – Check – Act) que es un ciclo continuo. Se podría resumir esto en la idea que, las mediciones están orientadas principalmente al “Do” (Implementación y operación de SGSI), como una entrada para el “Check” (Monitorizar y revisar), y de esta forma poder adoptar decisiones de mejora del SGSI a través del “Act”. El estándar establece que una organización debe

describir como se interrelacionan e interactúan el SGSI y las mediciones, desarrollando guías que aseguren, aclaren y documenten esta relación, con todo el detalle posible. Además, se debe desarrollar un programa de cómo ejecutar la medición de la seguridad de la información. El éxito de este programa, se basará en la asistencia o ayuda que estas mediciones aporten para adoptar decisiones, o determinar la eficiencia de los controles de seguridad. Por lo tanto este programa de mediciones debe estar basado en un "Modelo" de mediciones de seguridad de la información. La norma especifica también, como desarrollar las mediciones para poder cuantificar la eficiencia de un SGSI, sus procesos y controles. Las mediciones deben encontrarse totalmente integradas al SGSI

2.5. Una propuesta de medición de la superficie de ataque de un sistema [16]

En esta propuesta los autores proponen una métrica para determinar si un sistema software es más seguro que otro similar respecto a sus superficies de ataque. Utilizan una medida de la superficie de ataque del sistema como un indicador de la seguridad del mismo; de tal manera que cuanto mayor sea la superficie de ataque, más inseguro será el sistema. La superficie de ataque del sistema la miden en términos de tres tipos de recursos usados en ataques al sistema: métodos, canales y datos. Además, demuestran el uso de su métrica de superficie de ataque midiendo las superficies de ataque de dos servidores IMAP y demonios FTP de fuente abierta.

Para medir la superficie de ataque se realizan los siguientes tres pasos:

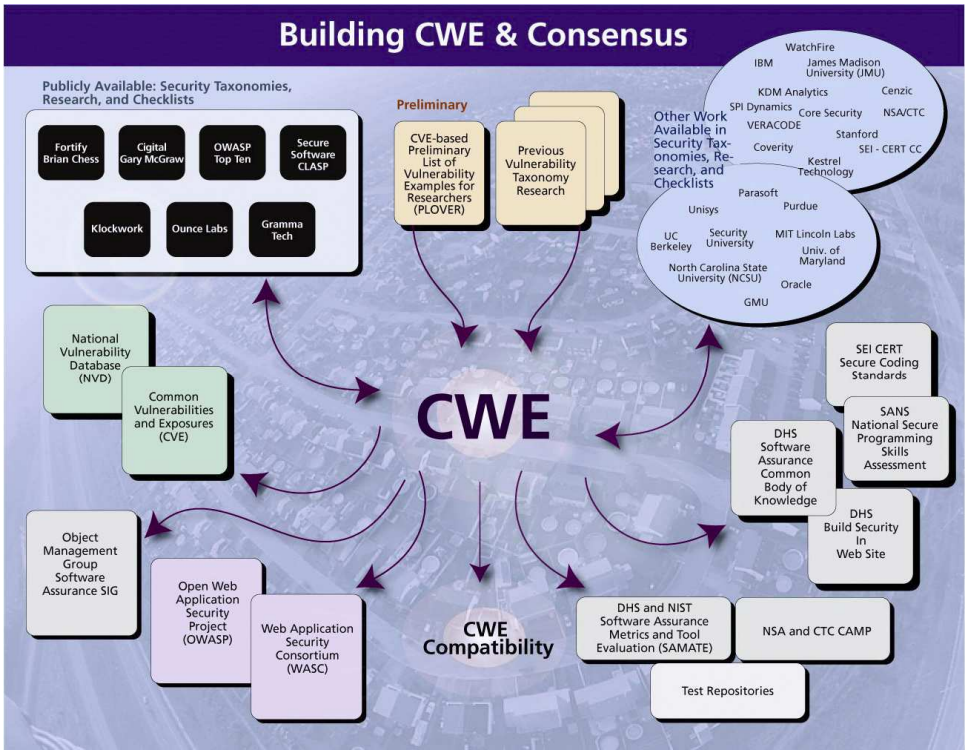
1. Dado un sistema *s*, y su entorno, *Es*, se identifica un conjunto, *M*, de puntos de entrada y salida, un conjunto, *C*, de canales, y un conjunto, *I*, de ítems de datos no confiables de *s*.
2. Se estima el daño del ratio esfuerzo-potencial , derm(m), para cada método $m \in M$, se estima el daño del ratio esfuerzo-potencial, derc(c), de cada canal $c \in C$, y el daño del ratio esfuerzo-potencial, derd(d), de cada ítem de dato $d \in I$.
3. La métrica de la superficie de ataque de s's es el triple $\langle \sum_{m \in M} der_m(m), \sum_{c \in C} der_c(c), \sum_{d \in I} der_d(d) \rangle$

Esta métrica puede ser usada como herramienta para los desarrolladores de software en el proceso de desarrollo software y por los consumidores de software en su proceso de toma de decisiones.

2.6. CWE (Common Weakness Enumeration) [18]

CWE proporciona un conjunto de debilidades software unificadas y medibles que facilitan una efectiva discusión, descripción, selección y uso de herramientas y servicios de seguridad software que permitan encontrar estas debilidades en el código fuente o en sistemas operacionales así como faciliten una mejor comprensión y gestión de las debilidades del software relacionadas con la arquitectura y diseño. En la Fig. 2 se muestra un esquema de la construcción de CWE y de cómo se establecen los consensos.

Fig. 2 CWE construcción y consenso



2.7. CVSS (Common Vulnerability Scoring System) [19]

CVSS es una iniciativa pública concebida por la National Infraestructure Assurance Council (NIAC) de EE.UU, un grupo que provee de recomendaciones al Department of Homeland Security de los EE.UU. Entre las organizaciones que lo adoptaron tempranamente destacan Cisco, US National Institute of Standards and Technology (NIST), Qualys y Oracle. En la actualidad, el CVSS está bajo la custodia del Forum for International Response Teams (FIRST).

Entre los beneficios que ofrece el CVSS están:

- **Puntuación estándar de vulnerabilidades:** El CVSS es neutro desde el punto de vista de las aplicaciones, permitiendo que distintas organizaciones asignen una puntuación a sus vulnerabilidades de TI a través de un único esquema.
- **Puntuación contextualizada:** La puntuación asignada por una organización corresponde al riesgo que la vulnerabilidad representa para dicha organización.
- **Sistema abierto:** El CVSS provee todos los detalles sobre los parámetros usados en la generación de cada puntuación permitiendo comprender tanto el

razonamiento que sustenta una puntuación como el significado de diferencias entre puntuaciones.

Las puntuaciones asignadas por el CVSS se derivan de los tres grupos de métricas siguientes:

- **Base:** Este grupo representa las propiedades de una vulnerabilidad que son inmutables en el tiempo, específicamente: complejidad de acceso, vector de acceso, y grado en que compromete la confidencialidad, integridad y disponibilidad del sistema.
- **Temporal:** Este grupo mide las propiedades de una vulnerabilidad que sí cambian en el tiempo, como por ejemplo la existencia de parches o código para su explotación.
- **Medio ambientales:** Este grupo mide las propiedades de una vulnerabilidad que son representativas de los ambientes de uso de las TI como por ejemplo la prevalencia de sistemas afectados y pérdidas potenciales.

El CVSS usa fórmulas sencillas y a partir de los grupos de métrica arriba enumerados arroja la puntuación final asociada a la vulnerabilidad.

De las **métricas de base** se deriva una puntación de 0.0 a 10.0 basado en las respuestas a las siguientes preguntas:

- Métricas de Explotabilidad (Exploitability Metrics)
  - Vector de acceso (Access vector): Local; Remoto.
  - Complejidad de ataque (Attack complexity): Baja; Alta.
  - Nivel de autenticación requerida (Level of authentication needed): No requerida; Requerida.
- Métricas de Impacto (Impact Metrics)
  - Impacto en la confidencialidad (Confidentiality impact): Ninguna; Parcial; Completa
  - Impacto en la integridad (Integrity impact): Ninguna; Parcial; Completa
  - Impacto en la disponibilidad (Availability impact): Ninguna; Parcial; Completa
  - Mayor impacto en (Impact Bias): Confidencialidad; Integridad; Disponibilidad

Las **métricas temporales** modifican la puntuación base, reduciéndola hasta en un tercio dependiendo de las respuestas a las siguientes preguntas:

- Disponibilidad del exploit (Exploitability):
  - No probada; Prototipo de ataque; Existencia de exploit; Alta
- Tipo de solución disponible (Remediation level):

    - Solución oficial; Solución temporal; Solución de contingencia; No disponible

- Nivel de verificación de la vulnerabilidad (Report Confidence)

Finalmente, las **métricas medioambientales** modifican la puntuación obtenida y generan un valor final dependiendo de las respuestas a las siguientes preguntas:

- Daño colateral potencial (Collateral damage potential):
    - Ninguno; Bajo; Medio; Alto
- Porcentaje de sistemas vulnerables (Target distribution):
    - Ninguno; Bajo; Medio; Alto

El CVSS fue diseñado de forma que fuera fácil de entender para un público general y para permitirle a cualquier organización priorizar el orden en que debe abordar las vulnerabilidades informáticas que la afecten. Lo anterior, independiente de cuál sea la tecnología utilizada por la organización en sus sistemas informáticos. La principal ventaja del CVSS es que resuelve el problema de múltiples sistemas de evaluación de vulnerabilidades, usualmente propietarios e incompatibles entre sí. Destacan entre sus fortalezas la elegancia, precisión, flexibilidad y relativa simpleza.

Como todo sistema de evaluación de vulnerabilidades, el CVSS tiene sus propias limitantes. Por ejemplo, no provee mecanismos para agregar puntuaciones individuales a través de varios sistemas informáticos o unidades organizacionales. Por sí sólo, no es adecuado para la gestión del riesgo de las TI puesto que no considera las estrategias de mitigación, como la instalación de cortafuegos o procedimientos de control de acceso. Tampoco es un repositorio de puntuaciones, como Bugtraq, o una base de datos de vulnerabilidades, como el Open Source Vulnerability Database, ni es un sistema de clasificación de vulnerabilidades, como el Common Vulnerabilities and Exposures. Sin embargo, el CVSS es relevante porque elimina duplicidad de esfuerzos en la evaluación de vulnerabilidades de TI y le permite a las organizaciones tomar decisiones con más y mejor información

2.8. CMSS (Common Misuse Scoring System) [23]

CMSS es un esquema de puntuación abierto y estandarizado para medir la severidad de las vulnerabilidades de mal uso de características de software. Una vulnerabilidad de mal uso de una característica de software es una vulnerabilidad en la que la característica proporciona una vía para comprometer la seguridad del sistema. Estas vulnerabilidades permiten de esta forma a los atacantes usar con fines malintencionados la funcionalidad supuestamente beneficiosa para la cual se idearon las características.

CMSS está relacionado con CVSS [19] y CCSS (Common Configuration Scoring System), que son métodos para puntuar temas de seguridad de brechas de software y configuración, respectivamente. Los tres sistemas estandarizados de puntuación permiten la comparación de análisis realizados por personas y empresas distintas realizados a lo largo del tiempo. CMSS se deriva de CVSS.

Las puntuaciones asignadas por el CMSS se derivan de tres grupos de métricas: base, temporales y medioambientales. Las métricas de base permiten valorar la explotabilidad intrínsica de la vulnerabilidad y el impacto en la confidencialidad, integridad y disponibilidad. Las métricas temporales miden los aspectos de variación de tiempo de la severidad de las vulnerabilidades, como la preponderancia de exploits. Las métricas medioambientales miden los aspectos de la vulnerabilidad relativos a la severidad específica para el entorno de la organización, como la implementación local de contramedidas. CMSS también incluye una fórmula que combina estas medidas para proporcionar una puntuación de la severidad de cada vulnerabilidad.

CMSS facilita a las organizaciones la toma de decisiones basándose en una valoración estandarizada cuantitativa de las vulnerabilidades de mal uso de las características del software

2.9. NIST 800-55 Security Metrics Guide for Information Technology Systems [25]

La norma NIST 800-55 es una guía de métricas de seguridad publicada por el NIST (National Institute of Standards and Technology de EEUU). La guía ayuda en el desarrollo, selección e implementación de medidas para ser usadas a nivel de sistemas de información y de programas. Estas medidas indican la efectividad de los controles de seguridad aplicados a sistemas de información y que soportan programas de seguridad. Con lo que mediante estas medidas se facilita la toma de decisiones, se mejora el rendimiento y se aumenta la trazabilidad a través de colecciones, análisis e informes de datos de rendimiento relacionados, proporcionando una manera de cuadrar la implementación, eficiencia y efectividad de los controles de seguridad de sistemas de información y programas con el éxito de la organización en la consecución de su misión. El proceso de desarrollo de medidas de rendimiento descrito en esta guía ayuda a los profesionales de la seguridad de la información de la organización en el establecimiento de relaciones entre las actividades de seguridad del sistema de información y los programas bajo su ámbito y la misión de la organización, ayudando a demostrar el valor de la seguridad de la información en sus organizaciones.

Este documento se centra en el desarrollo y colección de tres tipos de métricas:

- Métricas de implementación para medir la ejecución de la política de seguridad.
- Métricas de efectividad / eficiencia para medir la ejecución de la política de seguridad
- Métricas de impacto para medir las consecuencias de los eventos de seguridad para el negocio o su misión

Los tipos de medidas que pueden ser realmente obtenidos y que pueden ser también de utilidad para la mejora del rendimiento, dependen de la madurez del programa de seguridad de información de la organización y de la implementación de los controles de seguridad en sus sistemas de información. Aunque puede usarse diferentes tipos de métricas simultáneamente, el foco principal de las métricas de

seguridad de información cambia según va madurando la implementación de los controles de seguridad.

Además, en el anexo del documento se ponen ejemplos de métricas de seguridad, adaptación de parte de las cuales se incluyen en nuestra propuesta descrita en la Sección 3.

2.10. Métricas de seguridad para sistemas software [27]

Los autores describen una propuesta para definir métricas de seguridad software basadas en vulnerabilidades incluidas en sistemas software y en sus impactos en la calidad del software. Para ello, se basan en CVSS y en CVE (Common Vulnerabilities and Exposures), un estándar de la industria para vulnerabilidades y nombres de revelación.

Las métricas planteadas por los autores son las siguientes:

$$SM(s) = \sum_{n=1}^{m} (P_n \times W_n), \quad (1) \qquad W_n = \frac{\sum_{i=1}^{K} V_i}{K} \quad (2)$$

$$P_n = \frac{R_n}{\sum_{i=1}^{m} R_i} \quad (3) \qquad R_n = \frac{K}{M} \quad (4)$$

$$\sum_{n=1}^{m} P_n = 1 \quad (5)$$

Donde *SM(s)* representa la métrica de seguridad para el software *s*, y *Wi* (i = 1, 2, …, m) son la severidad de aquellas debilidades representativas del software *s*. Suponiendo que la debilidad correspondiente a *Wn* tiene *k* vulnerabilidades y sus correspondientes puntuaciones de base según CVSS son *V1, V2*, …, *Vk*. La severidad de esta debilidad, *Wn,* está definida como la puntuación media de ellas, como se muestra en la fórmula (2). En la fórmula (1), cada *Pi* (i = 1, 2, …, m) representa el riesgo de cada debilidad correspondiente. Se usa el porcentaje de ocurrencia de cada debilidad representativa en el total de ocurrencias de debilidades para calcular *Pi* como se muestra en la fórmula (3). Donde *Rn* es la frecuencia de ocurrencia de cada debilidad representativa sobre un lapso de tiempo en meses, tal y como se muestra en la fórmula (4), donde *K* es el número de debilidades, y *M* es el número de meses. Para tener el valor de la métrica *SM(s)* en un rango de 0 a 10, se requiere la fórmula (5) para normalizar *Pn*.

3 Estudio de las Métricas

En primer lugar, y como paso previo a la construcción de un modelo de métricas de seguridad de diseño de software, hemos realizado un análisis de las distintas

características, sub-características y sub-sub-características relacionadas con la seguridad, presente en los diversos modelos de métricas que hemos considerado en la sección anterior. Para ello, hemos construido la Tabla 2, donde se indican las distintas propiedades de seguridad de las propuestas analizadas (para mayor legibilidad, dichas propuestas han sido numeradas, y la correspondencia se ofrecen en la Tabla 1), y se especifica en el cruce, la relación de cada una de esas propiedades con cada propuesta (en blanco cuando la propiedad no la contempla la propuesta, "P" cuando la contempla parcialmente, "X" cuando aparece claramente contemplado como parte de la propuesta).

Tabla 1 Tabla de correspondencia de propuestas

Núm. Propuesta	Nombre Propuesta
1	Métricas de seguridad para diagramas de clases orientados a objetos [1]
2	Marco de trabajo para la estimación de la seguridad: perspectiva de la fase de diseño [3]
3	Criterios Comunes
4	ISO/IEC 27004
5	Una propuesta de medición de la superficie de ataque de un sistema [16]
6	CWE
7	CVSS
8	CMSS
9	NIST 800-55
10	Métricas de seguridad para sistemas software [27]

Tabla 2 Tabla comparativa resumen de propuestas

Característica \ Propuesta	1	2	3	4	5	6	7	8	9	10
Autenticidad	X		X	X	P	X	X	X	X	P
Confidencialidad	X		X	X	P	X	X	X	X	P
Conformidad		X	X	X	P	X	X	X	X	P
Detección de Ataques	P		P	P	X	X	X	X	P	P
Disponibilidad				P		X	X	X	P	P
Integridad	X		X	X	P	X	X	X	X	P
No Repudio			X	P		P	P	P	P	P
Trazabilidad	P	X	X	X		P	P	P	X	P
Conformidad (*safety*)		X	X	X		X	X	X	X	P
Seguridad y salud del operador				P		P	P	P	P	P
Seguridad y salud pública				P		P	P	P		
Daño comercial						P	P	P	P	P
Daño medioambiental						P	X	P		

Las diferencias en la consideración de unas propiedades y otras como características, sub-características, o incluso como parte de sub-sub-características, se justifica básicamente por dos hechos: el primero tiene que ver con el cambio en la

consideración de la seguridad que ha habido en los últimos años (eso justifica la variación entre la ISO/IEC 9126 y la ISO/IEC 25010), mientras que el segundo es la orientación de las propuestas, ya que aquellas que consideran la seguridad como sub-característica, son propuestas que consideran la calidad de manera general, mientras que las que consideran estas propiedades como características, es porque se trata de propuestas claramente orientadas a la seguridad.

Por ello, se ha construido un grupo canónico de características de seguridad, basado en el modelo de calidad de seguridad propuesto por [7], que sirve como base para comparar dichas características de seguridad desde la perspectiva de la medición de la seguridad en el diseño software. Es decir, que en la Tabla 2 se comparan cómo las propuestas descritas anteriormente miden este grupo de características de seguridad desde la perspectiva del diseño software.

Un análisis de la Tabla 2 indica que es difícil cubrir todas las propiedades de la seguridad desde la perspectiva del diseño, ya que se puede observar que las distintas propuestas de métricas de seguridad de diseño tratan de cubrir unas u otras características de la seguridad a nivel de diseño, pero no cubren todas las características de seguridad del modelo de seguridad utilizado como base de la comparativa. Asimismo, la mayoría de estas propuestas se centran en métricas de seguridad generales y en su mayor parte son realmente aplicables en fases posteriores a la etapa de diseño software. De tal forma que muchas de ellas hay que hacer un esfuerzo de adaptación para que sean aplicables desde la perspectiva del diseño software, sabiendo que cuanto antes y mejor se pueda medir la seguridad más económicamente rentable a la vez que robusto será el sistema de información

Por lo tanto, consideramos que el desarrollo de un modelo de métricas de seguridad de diseño software es un área de conocimiento aún por desarrollarse y que debe ser objeto de investigación para seguir avanzando en dicha materia. Además, dicho modelo de métricas de seguridad debería estar alineado con algún modelo de calidad de seguridad que le sirva de referencia así como con estándares de seguridad y le permita medir así la seguridad del diseño del software completamente, es decir, que considere todas las características del modelo de calidad de seguridad y de los estándares de seguridad relacionados que correspondan. Como por ejemplo las métricas deberán estar alineadas con el estándar ISO/IEC 27004 fundamentalmente en lo relativo a la medición de la seguridad de SGSI (Sistemas de Gestión de la Seguridad de Sistemas de Información).

4 Conclusiones

Aunque ya han aparecido algunas propuestas que se preocupan de abordar la seguridad de manera sistemática junto al desarrollo de los productos software [2, 5, 8, 9, 14, 15, 20, 22], así como normas y estándares de métricas de seguridad como los anteriormente expuestos, además de trabajos sobre métricas de seguridad basados en la puntuación de vulnerabilidades o debilidades (CVSS [19], CMSS [23], CWE [18]), basados en el análisis de código fuente [10], basadas en la medición de la seguridad de la arquitectura del sistema [16], basadas en la medición de la seguridad de los diagramas de clases orientados a objetos [1], o basadas en el riesgo ([24], o

MAGERIT [17]). Siendo muchas de estas propuestas muy interesantes, habitualmente tratan la seguridad de un modo parcial, y sin ofrecer un claro seguimiento de esos aspectos de seguridad a lo largo del proceso de desarrollo, y haciendo que la definición de métricas de seguridad a nivel de diseño haya recibido poca atención en estos últimos años [1].

Por lo tanto, se hace necesaria definir un conjunto de métricas tanto a nivel de diseño como posteriormente y de forma relacionada a nivel de implementación que nos permitan evaluar el nivel de cumplimiento deseado de los requisitos de seguridad que se hayan especificado en las etapas de análisis del software. Y que además, dichas métricas se integren con un modelo de seguridad (como componente de calidad) que claramente haya identificado una taxonomía de requisitos de seguridad para que éstos puedan ser identificados, modelados e implementados, junto al resto de requisitos, tanto funcionales como no funcionales.

En futuros trabajos y teniendo como base las propuestas analizadas, se propondrá tanto un modelo de seguridad como un modelo de métricas de seguridad de diseño. Dicho modelo será una propuesta integradora de conceptos con el fin de ofrecer una visión común en el área, tanto en lo que refiere a características y sub-características como a su definición formal.

Agradecimientos. Esta investigación es parte de los proyectos: MEDUSAS (IDI-20090557), financiado por el Centro para el Desarrollo Tecnológico Industrial, BUSINESS (PET2008-0136) concedido por el Ministerio de Ciencia e Innovación de España y SEGMENT (HITO-09-138) y SISTEMAS (PII2I09-0150-3135) financiados por la Consejería de Educación y Ciencia de la Junta de Comunidades de Castilla-La Mancha.

References

1. B. Alshammari, C. Fidge, and D. Corney, *Security Metrics for Object-Oriented Class Designs.* 2009 Ninth International Conference on Quality Software, 2009: p. 11-20.
2. D. Basin, J. Doser, and T. Lodderstedt, *Model Driven Security: from UML Models to Access Control Infrastructures.* ACM Transactions on Software Engineering and Methodology, 2006. **15**(1): p. 39-91.
3. S. Chandra, R.A. Khan, and A. Agrawal, *Security Estimation Framework: Design Phase Perspective.* 2009 Sixth International Conference on Information Technology - New Generations, 2009: p. 254-259.
4. K.-K.R. Choo, R.G. Smith, and R. McCusker, *Future directions in technology-enabled crime: 2007–09*, in *Research and Public Policy Series*, Australian_Government, Editor. 2007, Australian Institute of Criminology.
5. E. Fernandez-Medina and M. Piattini, *Designing Secure Databases.* Information and Software Technology, 2005. **47**(7): p. 463-477.
6. FIPS, *FIPS 140-2*, in *Security Requirements for Cryptographic Modules*. 2001, Federal Information Processing Standardization - National Institute of Standards and Technology
7. A.E. Fornaris, L.E. Sánchez, and E. Fernández-Medina, *Modelo de calidad para la seguridad.* RECSI (X Reunión Española sobre Criptología y Seguridad de la Información), 2010: p. (submitted).

8. C. Gutiérrez, E. Fernandez-Medina, and M. Piattini, *Towards a Process for Web Services Security.* Journal of Research and Practice in Information Technology, 2006. **38**(1): p. 57-67.
9. M. Hafner, R. Breu, B. Agreiter, and A. Nowak, *SECTET: An Extensible Framework for the realization of Secure inter-organizational Workflows.* Internet Research, 2006. **16**(5): p. 491-506.
10. I. Chowdhury, B. Chan, and M. Zulkernine, *Security metrics for source code structures*, in *Proceedings of the Fourth International Workshop on Software Engineering for Secure Systems*. 2008, ACM: Leipzig, Germany.
11. ISO/IEC, *ISO/IEC 15408:2005 Information technology - Security techniques - Evaluation criteria for IT security, (Common Criteria v3.0)*. 2005.
12. ISO/IEC, *ISO/IEC 27004:2009 - Information technology -- Security techniques -- Information security management -- Measurement.* 2009.
13. ITU, *ICT Security Standards Roadmap* 2009, International Telecommunication Union.
14. J. Jürjens, *UMLsec: Extending UML for secure systems development*, in *UML 2002 - The Unified Modeling Language, Model engineering, concepts and tools*, J. Jézéquel, H. Hussmann, and S. Cook, Editors. 2002, Springer. LNCS 2460.: Dresden, Germany. p. 412-425.
15. J. Jürjens, *Secure Systems Development with UML*. 2004: Springer-Verlag.
16. P.K. Manadhata, K.M.C. Tan, R.A. Maxion, and J.M. Wing, *An Approach to Measuring A System's Attack Surface*. 2007, Carnegie Mellon University: Pittsburgh,.
17. MAP, *Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información (MAGERIT - v 2)*. 2005, (Ministry for Public Administration of Spain).
18. R.A. Martin, *Common Weakness Enumeration (CWE v1.8)*. 2010, National Cyber Security Division of the U.S. Department of Homeland Security.
19. P. Mell, K. Scarfone, and S. Romanosky, *A Complete Guide to the Common Vulnerability Scoring System (CVSS 2.0)*. 2007, NIST and Carnegie Mellon University.
20. D. Mellado, E. Fernandez-Medina, and M. Piattini, *A Common Criteria Based Security Requirements Engineering Process for Development of Securie Information Systems.* Computer Standards & Interfaces, 2006. **29**(2): p. 244-253.
21. A.L. Opdahl and G. Sindre, *Experimental comparison of attack trees and misuse cases for security threat identification.* Information and Software Technology. In Press, Corrected Proof, 2008.
22. A. Rodríguez, E. Fernández-Medina, and M. Piattini. *M-BPSec: A Method for Security Requirement Elicitation from a UML 2.0 Business Process Specification*. in *3rd International Workshop on Foundations and Practices of UML*. 2007. Auckland, New Zealand.
23. E.V. Ruitenbeek and K. Scarfone, *The Common Misuse Scoring System (CMSS): Metrics for Software Feature Misuse Vulnerabilities*, in *NIST Interagency Report 7517*. 2009, National Institute of Standards and Technology.
24. O.S. Saydjari, *Is Risk a good security metric?* Quality of Protection Workshop – Security Measurements and Metrics (QoP'06), 2006: p. 59-60.
25. M. Swanson, N. Bartol, J. Sabato, J. Hash, and L. Graffo, *Security Metrics Guide for Information Technology Systems*, in *NIST Special Publication 800-55 Revision 1*. 2008, National Institute of Standards and Technologies.
26. A.J.A. Wang, *Information Security Models and Metrics.* 43nd ACM Southeast Conference, 2005: p. 2-178 to 2-184.
27. J.A. Wang, H. Wang, M. Guo, and M. Xia, *Security Metrics for Software Systems*, in *Proceedings of the 47th Annual Southeast Regional Conference (ACMSE '09)*. 2009.