Análisis y Modelado de Amenazas

Una revisión detallada de las metodologías y herramientas emergentes	
Versión:	**Fecha de creación:**
1.0	18 / 12 / 2006

Autor: Daniel P.F
metal AT/DOT hacktimes.com

Tabla de contenido

¿Qué es el modelado de amenazas?

El modelado de amenazas es una técnica de ingeniería cuyo objetivo es ayudar a identificar y planificar de forma correcta la mejor manera de mitigar las amenazas de una aplicación o sistema informático.

Para aprovechar mejor los beneficios que proporciona su uso, en un escenario ideal debería iniciarse desde las primeras etapas del desarrollo y planificación de cualquier aplicación o sistema.

La verdadera ventaja de esta técnica reside en el hecho de que al aplicarse como un proceso durante todo el ciclo de vida de desarrollo del software, supone un enfoque diferente al tradicional análisis de riesgos y la posterior aplicación de medidas que contribuyan a mejorar la seguridad. Es por esto que el análisis y modelado de amenazas, aunque es un campo relativamente nuevo, está despertando un gran interés.

En este sentido, y de forma reciente, Microsoft ha sido uno de los grandes impulsores de esta técnica, llegando a resultar su particular visión sobre el tema (por el momento) bastante bien encaminada. Al menos eso es lo que opinan distintas entidades como la OWASP Foundation y expertos de reconocidas empresas de seguridad a nivel internacional. Aunque, para alivio de algunos posibles detractores acérrimos de soluciones procedentes de entidades con ánimo de lucro, existen otras alternativas en desarrollo. Tanto en lo referente a metodologías como a herramientas.

El análisis y modelado de amenazas, trata por lo tanto, de un proceso que nos va a facilitar la comprensión de las diferentes amenazas de seguridad a las que va a estar expuesto un sistema o una aplicación, y cuya finalidad es preparar las defensas adecuadas durante las fases de diseño, implementación y también durante su posterior revisión y testeo. Se trata de identificar las amenazas y definir una política adecuada que nos permita mitigar el riesgo a unos niveles aceptables, de una forma efectiva y en base a unos costes razonables.

Básicamente, consiste en revisar el diseño y/o la arquitectura siguiendo una metodología que nos facilite encontrar y corregir los problemas de seguridad actuales o futuros. La idea que persigue esta técnica es que los diferentes actores involucrados (desarrolladores, testers, gerencia, administradores de sistemas, pentesters, consultores …etc.) participen en el proceso de identificación de las posibles amenazas. Tomando una actitud defensiva y también poniéndose en ocasiones, en el papel de un posible atacante. Siguiendo este enfoque, se fuerza a todos ellos a explorar las debilidades que puedan surgir o que ya estén presentes, y se determina de este modo si existen las oportunas contramedidas, o en caso contrario, se definen.

Al mismo tiempo, la realización de un modelado de amenazas contribuye a identificar y cumplir con los objetivos de seguridad específicos de cada entorno y facilita la priorización de tareas en base al nivel de riesgo resultante.

Consideraciones previas al proceso de modelado

No está de más tener presente que la explotación con éxito de un sólo fallo de seguridad podría llegar a comprometer todo el sistema. Por ello, y para conseguir que nuestro modelado de amenazas resulte efectivo, es importante tener en cuenta que se debe realizar como un proceso sistemático, en continua actualización. De este modo conseguiremos identificar y mitigar el mayor número posible de amenazas y vulnerabilidades.

Nuestro modelado de amenazas debería ser capaz de:

- Identificar amenazas potenciales así como las condiciones necesarias para que un ataque se logre llevar a cabo con éxito.
- Facilitar la identificación de las condiciones o aquellas vulnerabilidades que, una vez eliminadas o contrarrestadas, afectan a la existencia de múltiples amenazas.
- Proporcionar información relevante sobre cuales serían las contramedidas más eficaces para contrarrestar un posible ataque y/o mitigar los efectos de la presencia de una vulnerabilidad en nuestro sistema/aplicación.
- Proveer información sobre cómo las medidas actuales previenen la consecución de ataques.
- Transmitir a la gerencia la importancia de los riesgos tecnológicos en términos de impacto de negocio.
- Proporcionar una estrategia sólida para evitar posibles brechas de seguridad.
- Facilitar la comunicación y promover una mejor concienciación sobre la importancia de la seguridad.
- Simplificar la posterior actualización mediante el uso de componentes reutilizables.

Teniendo esto presente, podemos proceder a la puesta en marcha de un grupo de trabajo que facilitará la creación del modelado.

Se presupone que entre los participantes del grupo, existe algún integrante del "Tiger Team" y/o uno o varios consultores externos especializados en seguridad. También, se debería asignar un responsable o jefe de proyecto que coordine el proceso y se ocupe además de la recopilación de la información. Es aconsejable la participación no sólo de los desarrolladores y administradores de sistemas, sino también del equipo de testing, calidad (si lo hubiera), así como de algún miembro del equipo directivo/gerencia que pueda aportar información relevante sobre los procesos de negocio afectados por la aplicación.

Para llevar a cabo una identificación correcta de las posibles vulnerabilidades y amenazas que puedan afectar a nuestra aplicación y/o sistema, conviene tener presente ciertos factores como pueden ser: la validación de los datos de entrada, los procesos de autenticación y autorización, la configuración de las aplicaciones y sistemas, la administración de las excepciones que puedan generarse, la posibilidad de manipulación de parámetros, cuestiones referentes a la solidez de las medidas de criptografía y protección de datos confidenciales (almacenados y/o en tránsito), las técnicas empleadas para el registro y auditoria de las actividades, o el tratamiento de las sesiones.

A modo orientativo, la estrategia a seguir, podría ser similar a la que se propone a continuación:

1. **Recopilación de información**

   - Localizar la documentación escrita.
   - Realizar entrevistas a las partes implicadas.
   - Realizar una inspección general del sistema.

2. **Análisis y Brainstorming**

   - Identificación de los activos críticos: Asegurar una inversión correcta en tiempo y recursos es el objetivo último de nuestro modelado. Por lo que resulta vital determinar cuales son los activos que no pueden ser comprometidos sin acarrear consecuencias negativas para el negocio. Utilizando anotaciones, dibujos y un listado de componentes, el grupo de trabajo describe el sistema y se crea una lista de los activos, identificando los más críticos. Conviene prestar especial cuidado en no perder demasiado tiempo tratando de identificar activos intangibles, ya que suelen ser difíciles de describir.

3. **Creación de un borrador del modelado**

   - Descomponer el sistema.
   - Identificación de interdependencias con otros sistemas.
   - Identificación de posibles puntos de ataque y amenazas.
   - Clasificación y priorización de las amenazas.
   - Definición de las contramedidas.

4. **Revisión y actualización**

   - Corrección y mejora el modelado a medida que el sistema evoluciona y se toman decisiones sobre la tecnología utilizada.
   - Revisión de las amenazas, los riesgos y las contramedidas antes de su implementación.
   - Ajuste del modelado en función de los resultados obtenidos tras una revisión inicial de la seguridad, previa a la puesta en explotación.

5. **Implementación de las medidas correctivas**

   - En función de la magnitud del riesgo y el coste asociado a su mitigación, se implementan las diferentes contramedidas.

Conseguiremos de este modo, no sólo obtener un marco de referencia que nos permitirá focalizar mejor la inversión en seguridad, sino que al mismo tiempo lograremos mitigar de manera más efectiva las posibles amenazas. No nos olvidemos tampoco que el hecho de implicar a los diferentes actores (desarrolladores, gerentes, administradores de sistemas, testers …. etc) en cualquier proceso que ataña al término seguridad ya es de por sí todo un logro, muy positivo, y aún será mayor si esta planificación se realiza desde las primeras etapas del desarrollo.

Integración de la seguridad en el ciclo de vida del desarrollo

¿Qué ocurre cuando se integra la seguridad en el ciclo de vida del desarrollo del software?, ¿Realmente merece la pena?, ¿Cómo se ve afectado un producto cuando la seguridad se deja sistemáticamente a un lado?.

El modelo adoptado por Microsoft mediante la integración del análisis y modelado de amenazas en el ciclo de vida de desarrollo, se presenta a priori como un modelo válido, y parece que está dando sus frutos. Una prueba de ello es la información que se desprende de un reciente artículo de *David Litchfield (NGSSoftware)* , reconocido consultor y descubridor de un elevado número de vulnerabilidades. En este documento, el autor efectúa una comparativa entre la seguridad de dos conocidos gestores de bases de datos: **Oracle** y **SQL Server**. En él, se puede apreciar tanto la nefasta respuesta por parte de Oracle respecto a la política de corrección de vulnerabilidades, como el elevadísimo número de estas que han sido descubiertas en los últimos 5 años en comparación con las que se han detectado en las distintas versiones de SQL Server.

Sin duda, fruto de una política de seguridad más responsable por parte de Microsoft (o en vías de serlo) y de la adopción del **SDLC** (Secure Development Life Cicle).

Dejando a un lado las polémicas sobre si es justa o no esta comparativa, no parece una postura muy sensata la adoptada por Oracle. Recordemos que han batido el record de mayor tiempo transcurrido desde la notificación de un bug en su software y la salida del parche oficial: más de 2 años !!!
(http://seclists.org/bugtraq/2005/Oct/0085.html)

Impresionante. Como lo es también el elevadísimo número de bugs detectados, las críticas desafortunadas al trabajo de quienes colaboran por su cuenta en su descubrimiento, y el hecho de haber liberado parches cuyo objetivo era más bien dificultar el trabajo de estos últimos que arreglar los fallos de manera efectiva. (http://seclists.org/bugtraq/2005/Oct/0056.html)

Por no mencionar los bugs que no son subsanados en un tiempo razonable aduciendo excusas que, sinceramente, no cuelan.
(http://archives.neohapsis.com/archives/fulldisclosure/2005-11/0449.html)

- Which database is more secure? Oracle vs Microsoft (http://www.ngssoftware.com/research/papers/comparison.pdf)

Con los ejemplos anteriores se pone de manifiesto la importancia del análisis y modelado de amenazas y cómo afecta a corto-medio plazo su utilización.

Metodologías (State of the art)

Existen diversas metodologías que intentan ayudar en la medición y mitigación del riesgo inherente al desarrollo de software, las más conocidas son:

- Ace Threat Analysis and modeling.
- CORAS.
- Trike.
- PTA (Practical Threat Analysis).

▪ Ace Threat Analysis and Modeling

Microsoft ha desarrollado una metodología de análisis y modelado de amenazas, basada en la combinación de ideas propias con las de parte del equipo de @stake y que recientemente ha incorporado a sus filas. Esta metodología ha ido evolucionando y recogiendo ideas de diversos enfoques.

La versión inicial, se basa en el uso de árboles de ataques para luego extrapolar las amenazas y realizar una clasificación y un ranking de estas con el fin de priorizar las actuaciones necesarias para mitigar el riesgo. Mientras que en la segunda versión de esta metodología, han intentado aclarar los conceptos de amenaza, ataque y vulnerabilidad, actualizando su herramienta de modelado y cambiando sustancialmente el punto de vista original.

Desde un punto de vista de un posible atacante se trata de identificar:

- Los puntos de entrada de la aplicación.
- Los activos a proteger.
- Los diferentes niveles de confianza.

Se establece cual es el nivel de seguridad del sistema:

- Mediante casos de uso.
- Conociendo las distintas dependencias.
- Utilizando modelos del sistema.

Se determina cuales son las amenazas:

- Se identifican las amenazas.
- Se analizan y clasifican.
- Se identifican las vulnerabilidades a las que se ve expuesto el sistema.

Pasos del modelado de amenazas según Microsoft

Según la metodología propuesta por Microsoft, los cinco pasos del proceso de modelado de amenazas son:

1. **Identificar los objetivos de seguridad:** Determinar cuales son los objetivos ayudará a cuantificar el esfuerzo se debe dedicar a los siguientes pasos.

2. **Crear una descripción general de la aplicación:** Identificar los actores involucrados y las características más importantes de la aplicación facilitará la identificación de las amenazas más importantes.

3. **Descomponer la aplicación:** Una vez que se conoce la arquitectura, es preciso identificar las funcionalidades y los módulos susceptibles de provocar un mayor impacto en la seguridad.

4. **Identificar amenazas:** Con la información recopilada, y en función del contexto y el escenario de la aplicación, se procede a la identificación de las amenazas más importantes.

5. **Identificar vulnerabilidades:** Revisar las diferentes capas de la aplicación para identificar los puntos débiles.

Durante el desarrollo del modelado, se utilizan diferentes métodos, como los árboles de ataques y los diagramas de flujo de datos.

Para saber cuales son las amenazas es preciso conocer cuales son los puntos de entrada, los niveles de confianza y los activos de mayor interés. Para ello se utilizan los diagramas de flujo de datos, para comprender la lógica de la aplicación y saber cómo puede afectar el tratamiento de los datos a la integridad de los activos.

Los árboles de ataques, ayudan a identificar las amenazas y analizar cuales serían las formas de mitigarlas. En el nodo principal del árbol situamos el objetivo del atacante. Los nodos hijo representan las diferentes formas que tiene el atacante de conseguir sus objetivos. Estos nodos (subobjetivos) pueden representar métodos mutuamente exclusivos (OR) de conseguir el objetivo padre, o bien nodos que representen todas las acciones a efectuar necesarias para conseguir un objetivo (AND).

Una vez efectuados los pasos iniciales del proceso, se procede a realizar una **clasificación y puntuación de las amenazas** , de modo que el equipo de trabajo pueda determinar las distintas prioridades.

Si seguimos el modelo propuesto por Microsoft, hay dos esquemas básicos de clasificación y puntuación: STRIDE y DREAD.

STRIDE es más bien un sistema de clasificación, mientras que **DREAD** es un modelo que nos facilitará la puntuación (Damage potential, Reproducibililty, Exploitability, Affected users, Discoverability). Una forma efectiva de alcanzar unos mejores resultados podría ser variar y adaptar alguno de estos dos métodos a la situación específica de cada negocio y sus particularidades.

Método de clasificación STRIDE

El término STRIDE es el acrónimo de "Spoofing identity, Tampering with data, Repudiation, Information disclosure, Denial of service, Elevation of privilege". Es decir, suplantación de identidad, manipulación de datos, repudio, revelación de información, denegación de servicio y elevación de privilegios.

Para seguir el método STRIDE, se descompone el sistema en componentes significativos, tras analizar cada componente para comprobar si es susceptible de sufrir amenazas, se proponen acciones que traten de mitigarlas. A continuación, se repite el proceso hasta llegar a una situación cómoda con las amenazas restantes.

Como se utiliza un modelo de información basado en patrones, se podrán identificar los patrones de soluciones y problemas repetibles y organizarlos en categorías. Utilizando estas categorías para descomponer la aplicación para un mayor análisis, e identificando las vulnerabilidades de la aplicación relacionadas con cada categoría. De esta manera, se promueve la reutilización de la información y una comunicación más eficaz.

- **Spoofing Identity (Suplantación de identidad):**

Los usuarios no deberían ser capaces de hacerse pasar por otros usuarios. Es necesario disponer de una gestión apropiada de los procesos de autenticación. En

particular, es aconsejable prestar atención a aquellas situaciones en las que pueda ser posible realizar un robo de sesiones o en aquellas en las que se deba valorar una inapropiada implementación de los sistemas de autenticación que pueda suponer un riesgo excesivo para la seguridad global del sistema, como puede ocurrir en determinados entornos con la implantación de soluciones SSO (single sign-on).

- **Tampering with Data (Manipulación de datos):**

Durante el desarrollo de software, por desgracia es habitual sacrificar la implementación de medidas de seguridad en beneficio de unos menores tiempos de desarrollo.

Una de las primeras medidas preventivas que se suelen sacrificar es el filtrado y la validación de los datos enviados a y recibidos de los usuarios de la aplicación y/o del sistema.

En el caso específico de las aplicaciones desarrolladas para un entorno web, confiar en exceso en la buena fe del usuario es un error que nos puede costar muy caro. Como ejemplos claramente ilustrativos de este hecho, bastaría con recordar al lector vulnerabilidades de sobra conocidas, como la inyección de sentencias SQL o el XSS, y en general aquellos casos en los que las aplicaciones proporcionan al usuario, datos que no son obtenidos (y por tanto considerados en principio como fiables) directamente de la propia aplicación, se realiza algún tipo de cálculo con ellos y posteriormente se almacenan. Ya que estos datos pueden ser susceptibles de una manipulación efectuada por un usuario malintencionado que disponga de las herramientas adecuadas.

Por otra parte, situaciones como la presencia en un sistema de un binario comprometido pueden llevar a una grave amenaza para la seguridad del mismo. Tanto la validación como la integridad de los datos son cuestiones de vital importancia.

- **Repudiation (Repudio):**

Establecer un nivel adecuado del seguimiento de las acciones realizadas por los usuarios de la aplicación puede evitar la aparición de situaciones no deseadas. Se debe intentar garantizar el no repudio de los usuarios.

Cada aplicación y/o sistema en base a su funcionalidad, características y entorno presenta diferentes riesgos, y por lo tanto las medidas de registro de las actividades que efectúan los usuarios serán también diferentes. De modo que, por ejemplo las medidas de seguimiento y registro a implantar en una aplicación que efectúa operaciones bursátiles deberán ser más rígidas que las necesarias para un simple gestor de contenidos.

Una aplicación que permite a sus usuarios efectuar operaciones de compra-venta de acciones en bolsa, no se puede permitir el lujo de perder el rastro de ninguna operación realizada (con éxito, o no). De no ser así, y de producirse un error durante el transcurso de las operaciones, cualquiera de las partes involucradas podría salir perjudicada.

Supongamos que se produce un error en una transacción de tal magnitud que la aplicación efectúa la venta de todas las acciones que posee el usuario x. O quizás el propio usuario, por error haya efectuado una venta no deseada y trata de imputar su momento de ineptitud al ya clásico recurso "un fallo informático". Si la entidad responsable de la aplicación no es capaz de verificar de cual de las dos situaciones se trata, es poco probable, pero si posible que deba terminar por asumir esa pérdida económica, o cuanto menos arriesgarse a perder un cliente.

Obviamente, de producirse este caso extremo, es bastante probable que se den además otras circunstancias que deberían ser subsanadas de inmediato y que nunca deberían existir en aplicaciones de esta envergadura (ej: la no atomicidad de las operaciones).

- **Information Disclosure (Revelación de información):**

Desde un punto de vista tanto técnico como a nivel de negocio, la existencia en una aplicación de vulnerabilidades que permitan extraer información sensible es un factor claro de riesgo que en ocasiones puede derivar en una pérdida económica.

Algunas de estas situaciones podrían ser: la utilización insegura de memoria compartida que facilite la obtención de credenciales (ej: claves WEP en dispositivos inalámbricos) , la obtención de un listado de usuarios o e-mails válidos (para su posterior uso en el envío de spam o en el crackeo de credenciales por fuerza bruta), la obtención de información que facilite la identificación del entorno (sistemas, aplicaciones, soluciones comerciales empleadas …etc), el acceso a rutas o archivos en disco con información sensible, el uso de campos ocultos en formularios web para el intercambio de información confidencial, el uso inadecuado de las directivas de no cacheado en las cabeceras http …etc.

En general, cualquier información que pueda facilitar la tarea a un atacante proporcionando detalles del funcionamiento de la propia aplicación, sistema o del usuario que favorezcan una posterior explotación, o información cuya obtención y/o divulgación suponga una pérdida de confianza y reputación de cara a posibles o ya existentes usuarios, clientes, partners o inversores.

- **Denial of Service (Denegación de servicio):**

A la hora de diseñar una aplicación, o en el momento de añadir una nueva funcionalidad, es conveniente evitar aquellas situaciones que puedan devengar en la consecución de un ataque de denegación de servicio. Especialmente si está orientada hacia un uso masivo.

En ocasiones, la propia funcionalidad y entorno de la aplicación / sistema dificultará la optimización de los recursos. En todos aquellos casos en los que esto no es así, es conveniente tratar de proporcionar un uso racional de los mismos, ya sea evitando cálculos complejos, búsquedas intensivas o el acceso a archivos de gran tamaño a usuarios no autenticados y autorizados para ello.

- **Elevation of Privilege (Elevación de privilegios):**

En el caso de que la aplicación o el sistema proporcionen diferentes niveles de privilegio en función de los distintos tipos de usuarios, todas las acciones que conlleven el uso de privilegios deben ser filtradas a través de un mecanismo adecuado de autorización. Este método de validación de los privilegios deberá ser lo suficientemente robusto para impedir una posible escalada de privilegios.

En ningún caso se debe confiar en acciones tales como la ocultación de interfaces de acceso restringido, o el simple uso de estándares criptográficos, como ejemplos de medidas que mitiguen una deficiente autorización.

Para almacenar los datos que vamos recogiendo, podemos utilizar una plantilla con información similar a la contenida en la siguiente tabla:

Descripción	Inyección de comandos SQL
Objetivo de la amenaza	Componente de acceso a base de datos
Nivel de riesgo	¿?
Técnicas de ataque	El atacante introduce comandos SQL en el campo usuario utilizado para formar una sentencia SQL.
Contramedidas	Filtrar el contenido del campo usuario, utilizar un procedimiento almacenado que utilice parámetros para acceder a la base de datos.

Método de puntuación DREAD

Una vez que tenemos identificada la lista de amenazas, el siguiente paso consiste en puntuarlas de acuerdo al riesgo que suponen. Esto nos permitirá priorizar las actuaciones a efectuar para mitigar el riesgo. Recordemos que, el riesgo se puede cuantificar como el resultado de multiplicar la probabilidad de que la amenaza se produzca, por el daño potencial de esta.

Riesgo = Probabilidad * Daño potencial.

Podemos emplear una escala del 1 al 10 para valorar la probabilidad de ocurrencia, donde el 1 representaría una amenaza que es poco probable que ocurra, y 10 sería una amenaza muy probable. De igual forma, con el daño potencial, 1 indicaría un daño mínimo y 10 un daño máximo. Este enfoque, tan simplista, nos permite en un primer momento clasificar las amenazas en una escala entre 1-100 que podemos dividir en tres partes según su riesgo: Alto, Medio, Bajo.

Ej: Probabilidad 10 , Daño potencial 7, el riesgo será: Riesgo = 10 * 7= 70

Una amenaza con un riesgo alto, debería ser paliada de forma rápida. Una amenaza con un riesgo medio, aunque importante, es de menor urgencia, y por último las de riesgo bajo, se podrían llegar a ignorar dependiendo del coste y del esfuerzo necesarios.

El problema de aplicar este sencillo método, radica en la dificultad de valorar de igual forma el riesgo cuando esta valoración se efectúa entre varias personas.

El método **DREAD**, trata de facilitar el uso de un criterio común respondiendo a las siguientes cuestiones:

- **Damage potential** (Daño potencial): ¿Cual es el daño que puede originar la vulnerabilidad si llega a ser explotada?
- **Reproducibility** (Reproducibilidad): ¿Es fácil reproducir las condiciones que propicien el ataque?
- **Exploitability** (Explotabilidad): ¿Es sencillo llevar a cabo el ataque?
- **Affected users** (Usuarios afectados): ¿Cuantos usuarios se verían afectados?
- **Discoverability** (Descubrimiento): ¿Es fácil encontrar la vulnerabilidad?

De modo que, por ejemplo, se podría establecer un sistema de puntuación en el cual, una media en el rango 12-15 seria considerado un riesgo alto, entre 8-11 medio, y entre 5-7 bajo.

A continuación, se muestra lo que podría ser una típica tabla de puntuación para priorizar las amenazas:

	Puntuación	Alto (3)	Medio (2)	Bajo (1)
D	**Damage potential (Daño potencial)**	El atacante podría ejecutar aplicaciones con permiso de administrador; subir contenido.	Divulgación de información sensible	Divulgación de información trivial
R	**Reproducibility (Reproducibilidad)**	El ataque es fácilmente reproducible.	El ataque se podría reproducir, pero sólo en condiciones muy concretas, ejemplo: condición de carrera.	Ataque difícil de reproducir, incluso conociendo la naturaleza del fallo.
E	**Exploitability (Explotabilidad)**	Un programador novel podría implementar el ataque en poco tiempo.	Un programador experimentado podría implementar el ataque.	Se requieren ciertas habilidades y conocimientos para explotar la vulnerabilidad.
A	**Affected users (Usuarios afectados)**	Todos los usuarios, configuración por defecto …	Algunos usuarios, no es la configuración por defecto.	Pocos usuarios afectados.
D	**Discoverability (Descubrimiento)**	Existe información pública que explica el ataque. Vulnerabilidad presente en una parte de la aplicación muy utilizada.	La vulnerabilidad afecta a una parte de la aplicación que casi no se utiliza. No es muy probable que sea descubierta.	El fallo no es trivial, no es muy probable que los usuarios puedan utilizarlo para causar un daño potencial.

Si tomamos como ejemplo un caso donde exista la posibilidad de inyectar comandos SQL en la aplicación, de acuerdo a la clasificación DREAD obtendríamos:

Amenaza	D	R	E	A	D	Total	Puntuación
Inyección de comandos SQL	3	3	3	2	3	14	Alto

Ahora que ya sabemos cual es el riesgo, estamos en disposición de actualizar la información referente a esta amenaza obtenida durante la clasificación con el método STRIDE en el ejemplo anterior, de manera que quedaría así:

Descripción	Inyección de comandos SQL
Objetivo de la amenaza	Componente de acceso a base de datos
Nivel de riesgo	Alto
Técnicas de ataque	El atacante introduce comandos SQL en el campo usuario utilizado para formar una sentencia SQL.
Contramedidas	Filtrar el contenido del campo usuario, utilizar un procedimiento almacenado que utilice parámetros para acceder a la base de datos.

Evolución de la metodología

A mediados de 2006, Microsoft ha anunciado lo que denomina **ACE Threat Analysis and Modeling v2**, que no es ni más ni menos que la revisión de la metodología anterior.

Esta nueva versión se basa en la idea principal de **cambiar la perspectiva del análisis y reorientarlo desde el punto de vista de un defensor.** Según el ACE Team, el defensor está en una posición mejor que el atacante para comprender cuales son las amenazas que afectan al sistema. Se presupone que las distintas personas implicadas en su desarrollo y puesta en funcionamiento conocen de manera directa que es lo que hace el software, cómo lo hace y cuales serian las partes más vulnerables. Mientras que un atacante debe orientarse únicamente haciendo uso de especulaciones en base a su observación y las pruebas que efectúa sobre las partes de la aplicación expuestas públicamente o que han sido descubiertas durante la etapa de reconocimiento en un ataque.

Además de este cambio de enfoque, se ha tratado de simplificar el trabajo de aquellos que no tienen necesariamente por que ser "expertos" en temas de seguridad. Se abandona el uso de los métodos STRIDE y DREAD y **se fundamenta todo el proceso en una idea clave:**

UN EVENTO NO ES UNA AMENAZA SI NO SE TRADUCE EN UN IMPACTO NEGATIVO PARA EL NEGOCIO.

Eventos que anteriormente eran considerados una posible amenaza, como por ejemplo cualquier situación que provoque una excepción y facilite información para aprovechar un posible vector de ataque, ahora no lo son, ya que no se puede derivar de ellos un impacto negativo para el negocio de forma directa. En todo caso se podrían considerar como el punto de partida para un potencial ataque, pero no como una amenaza.

De igual forma, el compromiso de una base de datos que contenga números de tarjetas de crédito de nuestros clientes SI es una amenaza, ya que hay un perjuicio claro para el negocio.

Para comprender mejor cual es el verdadero impacto de una amenaza, se identifica en primer término cual es el contexto en el que esta se sitúa. Esto es lo que se

conoce como el Contexto de la aplicación, y se determina descomponiendo la aplicación en datos, roles, componentes y si las hubiera, dependencias externas.

Sin embargo, a pesar de las modificaciones, se mantiene la idea original de realizar el análisis y modelado como un proceso iterativo desde las etapas iniciales del desarrollo del software, añadiendo más detalles al modelado a medida que se avanza y va evolucionando la aplicación / sistema.

Se comienza definiendo las amenazas. Posteriormente se identifica cómo estas amenazas se pueden llegar a materializar mediante vulnerabilidades potenciales y ataques asociados. Esto nos va a permitir implementar las medidas correctivas que mitiguen las amenazas.

Todo esto se consigue, haciendo uso de unas librerías predefinidas (incorporadas en la herramienta de modelado), donde se describen los diferentes ataques asociados con las distintas amenazas, así como las formas más efectivas de mitigarlos.

La actualización de estas librerías de ataques, es tarea del personal con conocimientos de seguridad. De este modo, se libera de trabajo al resto y se asegura que el mantenimiento de esta información es el más idóneo.

El cambio de metodología también viene acompañado de una nueva herramienta, que veremos más adelante. Esta herramienta, permite generar de forma automática modelos de amenazas basándose en un determinado contexto de la aplicación, enlazando posteriormente esas amenazas con las correspondientes contramedidas identificadas en la librería de ataques.

Los pasos a seguir en la aplicación de la nueva metodología ACE Threat Analysis & Modeling v2 son los siguientes:

Definir	Identificar los distintos roles
	Identificar los datos
	Definir la matriz de control de acceso sobre los datos
	Usar la matriz de control para definir cuales son los casos de uso
	Identificar los componentes
	Definir la secuencia de llamadas para cada caso de uso
Modelar	Generar el listado de amenazas
	Identificar las contramedidas para cada amenaza con la información presente en la librería de ataques
	Identificar cómo se trata el riesgo de cada amenaza
Cuantificar	Determinar el impacto del riesgo asociado con cada amenaza
	Determinar la probabilidad de riesgo asociado con cada amenaza
Validar	Validar el modelado. Realizar las optimizaciones y mejoras oportunas

Una vez que se define el contexto de la aplicación en el que está presente la amenaza, el siguiente paso es la creación de casos de uso (Use Cases), donde se listarán las posibles llamadas (Calls), es decir, las formas en las que un sujeto interactúa con un objeto.

Hay que tener en cuenta que las amenazas se basan en un modelo de inclusiones / exclusiones para efectuar su análisis de riesgos y su clasificación de seguridad. Se listan las inclusiones, y todo lo demás deben ser exclusiones. Por lo tanto se centra en las llamadas incluidas en el modelo de amenazas que se derivan del contexto de la aplicación.

De manera que se seguiría este esquema:

Sujeto ejecuta Acción mediante Componente

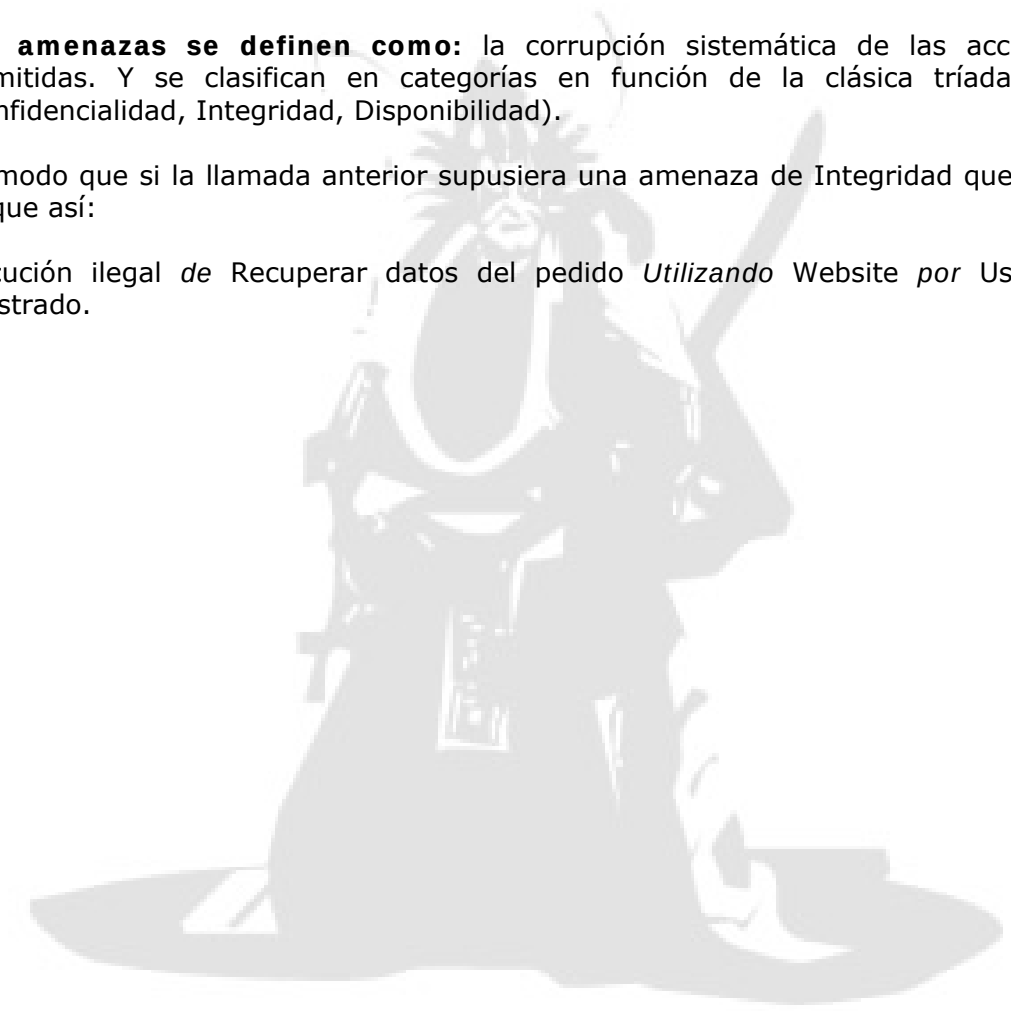
Por ejemplo:

Usuario registrado Ejecuta **Recuperar datos del pedido** mediante **Website**

Las amenazas se definen como: la corrupción sistemática de las acciones permitidas. Y se clasifican en categorías en función de la clásica tríada CIA (Confidencialidad, Integridad, Disponibilidad).

De modo que si la llamada anterior supusiera una amenaza de Integridad quedaría tal que así:

Ejecución ilegal *de* Recuperar datos del pedido *Utilizando* Website *por* Usuario registrado.



▪ CORAS

http://coras.sourceforge.net

CORAS (Consultative Objective Risk Analysis System), es un proyecto creado por la unión europea con el objetivo de proporcionar un framework orientado a sistemas donde la seguridad es crítica, facilitando el descubrimiento de vulnerabilidades de seguridad, inconsistencias, y redundancias.

CORAS proporciona un método basado en modelos, para realizar **análisis de riesgos**, y se basa en el uso de tres componentes:

- Un lenguaje de modelado de riesgos basado en el UML.
- La metodología CORAS, una descripción paso a paso del proceso de análisis con una directriz para construir los diagramas CORAS.
- Una herramienta para documentar, mantener y crear los informes del análisis.

Aunque **no es** exactamente **un framework para el modelado de amenazas**, su uso orientado a tal fin, puede contribuir a la reducción de riesgos y la adopción de unas correctas contramedidas, por lo que me ha parecido interesante mencionarlo. Tan sólo trataré de dar una visión general sobre el mismo, sin extenderme demasiado. Al final de este artículo se incluyen referencias a otra documentación que puede ser de interés para quien desee profundizar en el uso de CORAS.

La metodología CORAS, hace un uso intensivo de los diagramas.

Existen 5 tipos diferentes:

Diagrama superficial de activos: Muestra una visión general de los activos y cómo el daño sobre un activo puede afectar al resto.

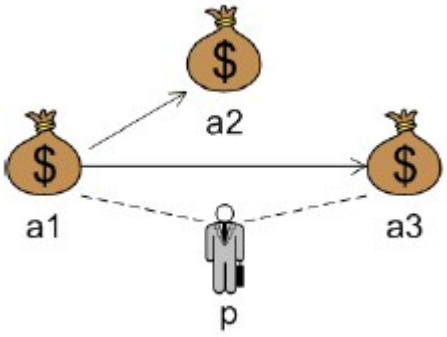
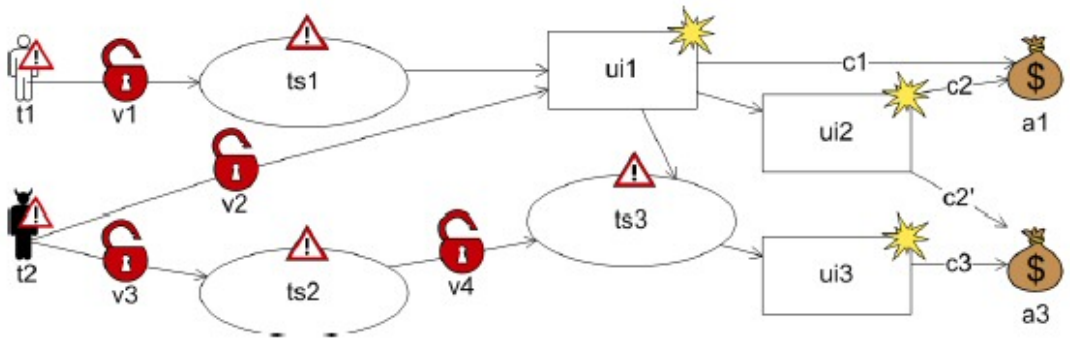


Diagrama de amenazas: Muestra una visión completa de la secuencia de eventos iniciados por las amenazas y las consecuencias que tienen éstas sobre los activos. Sus componentes básicos se muestran en el ejemplo inferior, y son: amenazas deliberadas, amenazas accidentales, amenazas no-humanas, vulnerabilidades, escenarios de amenazas, incidentes no deseados y activos.



Ejemplo de diagrama de amenazas

Diagrama superficial de riesgo: Es un resumen del diagrama de amenazas, mostrando los riesgos. Tiene 5 componentes básicos: amenazas deliberadas, accidentales y no-humanas, riesgos y activos. A cada riesgo se le asigna un valor.

Diagrama de tratamiento: ofrece una visión completa de las contramedidas propuestas. Se basa en el diagrama de amenazas, sustituyendo las consecuencias del impacto sobre los activos con los riesgos procedentes del diagrama superficial de riesgo, y añadiendo los escenarios de contramedidas propuestos.

Diagrama superficial de tratamiento: es un resumen de las contramedidas, añadiendo los distintos escenarios posibles y mostrando las relaciones entre los distintos elementos propuestos para tratar el riesgo.

Los siete pasos necesarios para realizar un análisis de riesgos utilizando CORAS podrían resumirse de la siguiente forma:

- **Paso 1:** Se realiza una entrevista inicial entre los representantes del cliente y los analistas para conocer cuales son los objetivos principales del análisis. Se recopila la información necesaria en función de los requisitos del cliente.

Tareas:	**Participantes:**
▪ Se presenta el método que se va a utilizar en el análisis. ▪ El cliente establece las metas y presenta el sistema objeto del análisis. ▪ Se fija el enfoque y el ámbito del análisis. ▪ Se planifican reuniones y talleres de trabajo.	▪ Analistas ▪ Representantes del cliente: - Personal con capacidad de decisión - Expertos técnicos (opcional) - Usuarios (opcional)
1. Durante esta primera etapa del análisis puede ser conveniente utilizar dibujos o anotaciones en una pizarra para describir el sistema objetivo. **2.** La presentación se puede mejorar posteriormente con métodos más formales como el uso de UML o diagramas de flujo de datos.	

Paso 1: Reunión inicial

- **Paso 2:** Una segunda reunión con el cliente para comprobar que la información suministrada al analista ha sido suficiente. Se realiza un análisis superficial, procediendo a identificar las primeras amenazas, vulnerabilidades, los diferentes escenarios, así como los posibles incidentes no deseados.

Tareas:	**Participantes:**
▪ Los analistas presentan su visión y comprensión del sistema. ▪ Se identifican los activos. ▪ Se realiza un análisis superficial.	▪ Analistas ▪ Representantes del cliente: - Personal con capacidad de decisión - Expertos técnicos (opcional) - Usuarios (opcional)

Diagramas de activos:

- Dibujar un área que represente de forma lógica o física el objetivo del análisis.
- Colocar en ella los activos directos.
- Colocar los activos indirectos fuera del área.
- Indicar mediante flechas cómo los activos pueden afectar al resto.
- Los activos se deben clasificar según su importancia.
- Si existen varios clientes involucrados en el mismo proyecto, cada activo debe asociarse a su correspondiente cliente.

Descripciones del objetivo:

- Utilizar una notación formal o estandarizada como puede ser UML. Asegurarse de que está debidamente documentada para que todos los participantes puedan comprenderla.
- Crear modelos de las funcionalidades y características del objetivo, tanto estáticas (configuraciones de hardware, diseño de la red … etc) , como dinámicas (procesos de trabajo, flujo de información …etc).
- Es recomendable el uso de diagramas UML de clase y colaboración para las partes estáticas de la descripción.
- Para las partes dinámicas, son recomendables los diagramas UML de actividad y de secuencia.

Paso 2: Análisis superficial

- **Paso 3:** Se realiza una descripción más detallada del sistema a analizar, facilitando al cliente la documentación necesaria para su aprobación.

Tareas:	**Participantes:**
▪ El cliente aprueba las descripciones del objetivo y de los activos asociados. ▪ Se puntúan y clasifican los activos según su importancia.	▪ Los mismos que en el paso anterior, pero esta vez es preciso que estén presentes aquellas personas con

▪ Se establecen escalas de consecuencias para cada activo según el ámbito del análisis. ▪ Se define una escala de ocurrencias. ▪ El cliente debe decidir el criterio de evaluación del riesgo para cada activo según el ámbito del análisis.	capacidad de decisión.

Paso 3: Aprobación

- **Paso 4:** El analista, junto con las personas que mejor conocen el sistema, identifican todos los posibles incidentes no deseados, amenazas, vulnerabilidades y escenarios.

Tareas:	**Participantes:**
• Se completan los diagramas iniciales de amenazas, se identifican las vulnerabilidades, amenazas, escenarios y los posibles incidentes no deseados.	▪ Analistas ▪ Representantes del cliente: - Personal con capacidad de decisión (opcional) - Expertos técnicos - Usuarios
Diagramas de amenazas: **1.** Utilizar el área del diagrama de activos y añadir más áreas si es necesario. **2.** Modelar diferentes tipos de amenazas en diagramas separados. **3.** Los activos se listan fuera del área, a su derecha. **4.** A la izquierda, se sitúan las amenazas. Fuera del área se sitúan las amenazas externas (intrusos …etc). **5.** Los incidentes no deseados se sitúan dentro del área señalando las relaciones con los activos afectados. **6.** Los activos que no son dañados por ningún incidente se eliminan del diagrama. **7.** Se añaden escenarios de amenazas entre las amenazas y los incidentes no deseados, según una secuencia lógica de ocurrencia. **8.** Se añaden las vulnerabilidades antes que el escenario de amenazas o los incidentes no deseados. Ej: Se incluye antes una vulnerabilidad llamada "solución de backup inapropiada" que el escenario "la solución de backup falla".	

Paso 4: Identificación de riesgos

- **Paso 5:** Se estiman las consecuencias y los valores de ocurrencia para cada uno de los posibles incidentes no deseados que se han identificado en los pasos anteriores.

Tareas:	**Participantes:**
▪ Se estima la probabilidad de ocurrencia de cada escenario de amenazas. En ella se basa la	▪ Analistas ▪ Representantes del cliente: - Personal con capacidad de

probabilidad de ocurrencia de los incidentes no deseados. ▪ Para cada relación entre un incidente no deseado y los activos que se ven afectados se estima una consecuencia.	decisión - Expertos técnicos - Usuarios
Estimación de riesgos en los diagramas de amenazas: **1.** Se incluye la estimación de ocurrencia para cada escenario de amenazas. **2.** Se añade la estimación de ocurrencia de los incidentes no deseados basándonos en los escenarios de amenazas. **3.** Se anota cada relación entre los incidentes no deseados y los activos, así como sus consecuencias en función de una escala de consecuencias para cada activo afectado.	

Paso 5: Estimación de riesgos

- **Paso 6:** Se proporciona al cliente un borrador del análisis para una primera revisión y corrección.

Tareas:	**Participantes:**
▪ Se ajustan o se confirman los valores previamente estimados para las ocurrencias de los incidentes no deseados y las posibles consecuencias. ▪ Si es preciso, se realizan los ajustes necesarios para las zonas de riesgo aceptable en las matrices de riesgo. ▪ Se crea un diagrama de riesgo para mostrar una visión general del mismo.	▪ Analistas ▪ Representantes del cliente: - Personal con capacidad de decisión - Expertos técnicos (opcional) - Usuarios (opcional)
Diagramas de riesgos: **1.** Utilizando el diagrama de amenazas, se substituyen todos los incidentes no deseados con símbolos de riesgo, mostrando una descripción breve del riesgo y si el riesgo es aceptable o no. **2.** Se eliminan los escenarios de amenazas y las vulnerabilidades, pero manteniendo las relaciones entre las amenazas y los riesgos. **3.** Si resulta útil, se parten los diagramas de riesgo en varios de acuerdo con el tipo de amenaza, parte del objetivo o la importancia del activo.	

Paso 6: Evaluación de riesgos

- **Paso 7:** Se establece el tratamiento del riesgo, es decir, las contramedidas en función del coste/beneficio.

Tareas:	**Participantes:**
▪ Añadir tratamientos en los diagramas de amenazas.	▪ Analistas ▪ Representantes del cliente:

▪ Estimar el coste/beneficio de cada tratamiento y decidir cuales se van a usar. ▪ Mostrar los tratamientos en los diagramas de riesgo.	- Personal con capacidad de decisión - Expertos técnicos - Usuarios

Diagramas de tratamiento:

1. Utilizar los diagramas de amenazas como base, incluyendo todas las flechas para mostrar las relaciones entre los incidentes no deseados y los activos así como los iconos para representar el riesgo. Mostrar sólo los riesgos inaceptables.
2. Incluir en el diagrama los tratamientos, apuntando hacia dónde serán aplicados.

Diagramas de tratamiento superficial:

1. Utilizando los diagramas de riesgo como base, eliminar los riesgos aceptables.
2. Añadir tratamientos según el diagrama de tratamientos.

Paso 7: Tratamiento del riesgo

▪ Trike

http://www.octotrike.org

Los creadores de Trike, aportan a la comunidad open source, un framework y una metodología conceptual, acompañada por una herramienta que intenta facilitar el proceso de modelado. La metodología, está diseñada con el propósito de permitir al analista describir de forma completa y precisa las características de seguridad de un sistema, desde los detalles de alto nivel de la arquitectura, hasta la implementación. O al menos en teoría. ;)

Según sus autores, Trike está aún en desarrollo, y aunque supuestamente debería proporcionar un nivel suficiente de detalle para permitir su uso práctico en un entorno real, yo la verdad, tengo serias dudas.

A día de hoy, la última versión disponible de la herramienta, data del 7 de Febrero de este año 2006, y el borrador de la metodología es de 2005. Por lo que no parece que estén muy pendientes de su actualización. Una lástima.

Sinceramente, se me hace difícil extraer algo útil del documento oficial. Espeso y demasiado conceptual, a mi parecer. Como ellos mismos dicen, gran parte de la metodología está aún en fase experimental. Creo que no mienten ;)

Para quien esté interesado en echarle un vistazo, la metodología que propone Trike, está disponible aquí:
http://www.octotrike.org/Trike_v1_Methodology_Document-draft.pdf

▪ PTA (Practical Threat Analisys)

La empresa PTA Technologies ha desarrollado su propia metodología que trata de solventar las limitaciones que según ellos tiene la versión 1 de la metodología propuesta por Microsoft. PTA CTMM (Calculative Threat Modeling Methodology) es el nombre que recibe.

Antes de comenzar el proceso de modelado, el analista debe familiarizarse con la aplicación / sistema. Resultando particularmente útil recopilar la siguiente documentación con el fin de que nos sirva de ayuda en el momento de decidir si se aplican o no los distintos escenarios del sistema que vamos a modelar:

- Descripción funcional del sistema donde se incluyan casos de uso típicos.
- Diagrama de la arquitectura del sistema y documentación de los distintos módulos.
- Un diccionario de términos, donde se expliquen los distintos vocablos utilizados en los restantes documentos.

Etapas de la metodología

1. **Identificación de activos.** Se determina cuales son los activos de mayor valor que deben ser protegidos ante posibles daños, con el fin de determinar las prioridades.
2. **Identificación de las vulnerabilidades.** Dependiendo de la arquitectura, funcionalidad y la lógica del negocio se determina de forma iterativa cuales son las vulnerabilidades.
3. **Definición de contramedidas.** Se establecen las contramedidas a adoptar en función de las vulnerabilidades y del coste que supondrá su implementación.
4. **Creación de escenarios de amenazas y planes de mitigación.**
   Se identifican los distintos elementos de las amenazas y los parámetros de la forma siguiente:

   - Mediante una breve descripción del escenario.
   - Identificando los activos que se ven amenazados y su nivel de daño potencial.
   - Se calcula el nivel de riesgo de la amenaza de forma automática, basándose en el daño total que podría ser ocasionado y en la probabilidad de ocurrencia de la amenaza.
   - Identificando cuales son las vulnerabilidades del sistema que pueden ser explotadas para que la amenaza exista. Automáticamente se genera una lista de contramedidas.
   - Se selecciona la combinación de contramedidas más efectiva de acuerdo al plan de mitigación más conveniente.

Para comenzar el proceso de análisis de las amenazas, se puede utilizar una serie predefinida de activos, vulnerabilidades y contramedidas típicas. Como resultado del proceso de análisis obtendremos:

- Un listado de las amenazas, su riesgo y daño potencial sobre los activos si éstas se materializan.
- Una lista de activos y su riesgo financiero.

- Las contramedidas, junto con el efecto general de mitigación obtenido, así como el coste-efectividad relativo a la contribución de cada contramedida a la reducción del riesgo del sistema.
- El máximo riesgo financiero del sistema, el riesgo final del sistema una vez que se hayan implementado todos los planes de mitigación. El nivel actual de riesgo presente en el sistema en base al nivel de implementación del plan de contramedidas.

La herramienta que acompaña a la metodología PTA permite el uso de etiquetas para describir las diferentes áreas de la arquitectura del sistema. Un listado de etiquetas con información relevante nos facilitará la clasificación de los diferentes elementos que componen el modelado.

Es recomendable también, examinar como se comporta el modelo en respuesta a los cambios que hagamos en los distintos parámetros y realizar distintas pruebas de escenarios que puedan contribuir a ajustar el modelado a un escenario lo más realista posible.

1. Identificación de activos

La identificación de los activos, así como la asignación de su valor financiero y la estimación de las pérdidas financieras ocasionadas por un posible daño en los activos, es un proceso clave en el análisis de amenazas. Recordemos que la correcta priorización de las medidas correctivas de las amenazas se basa en gran medida en una valoración correcta de los diferentes activos.

Determinar un valor adecuado para cada activo puede ser una tarea complicada, especialmente cuando se trata de activos intangibles, y lo será más aún cuando la persona al cargo del análisis tiene un perfil más bien técnico. Por esto es importante contar con el apoyo de personal que revise de forma periódica el valor de estos activos, como puede ser personal del departamento de marketing, legal, administración …etc. De este modo, el analista puede ir realizando diferentes pruebas y ajustando el modelado con una mayor precisión y detalle.

2. Identificación de las vulnerabilidades

Para una identificación correcta de las vulnerabilidades, es preciso que el analista conozca en detalle la funcionalidad, arquitectura y los procesos de implementación y puesta en funcionamiento de la aplicación / sistema. Para ello, es recomendable la colaboración con el resto del personal. Una o varias personas especializadas en seguridad internas o externas a la organización resultarán claves en el proceso de identificación, comprensión y una apropiada valoración de las vulnerabilidades.

3. Definición de contramedidas

En la aplicación de la metodología PTA, la definición de las contramedidas a adoptar para mitigar las amenazas que afectan a nuestra aplicación dará como resultado:

Por una parte, un listado en el que se incluye el coste de implementación de cada contramedida, junto con sus etiquetas de información asociadas. Si la contramedida ya ha sido aplicada, se deberá indicar tal suceso al efecto de poder llevar un seguimiento también del nivel actual de riesgo al que está sometido el sistema.

Por otro lado, se asocian las distintas contramedidas a sus correspondientes vulnerabilidades. Pudiendo resultar útil en ocasiones, adoptar contramedidas que mitiguen al mismo tiempo varias vulnerabilidades.

4. Creación de escenarios de amenazas y planes de mitigación

A la hora de preparar un escenario, resulta interesante realizar una clasificación de los potenciales tipos de atacantes, así como el nivel de conocimientos que éstos pueden necesitar para realizar un ataque en concreto, y cuales pueden ser sus objetivos prioritarios. Esto nos ayudará a focalizar mejor la inversión a la hora de priorizar nuestras actuaciones.

Recordemos también, que en no pocas ocasiones los ataques proceden desde el interior del propio entorno, por lo que a menudo deberemos prestar una especial atención a este tipo de atacantes y los activos más importantes a proteger.

Una buena manera de proceder será definir un tipo de atacante para cada tipo de usuario que aparezca en los diferentes casos de uso, y añadir otros tipos de atacantes a un nivel más general. Se deben documentar también las distintas formas de acceso al sistema así como los puntos de entrada de la aplicación.

Para comenzar a construir un escenario y un plan de mitigación de las amenazas, deberemos en primer lugar, introducir en la herramienta PTA, la información relativa a una amenaza, siguiendo un proceso de descomposición e iterativo. En la descripción de la amenaza, se incluirán las acciones de las que se puede servir el atacante para intentar materializar la amenaza, así como el impacto estimado que tendría la materialización de esta amenaza en nuestro sistema. La descripción se utilizará como referencia en los siguientes pasos y se irá actualizando si es preciso.

Identificamos la lista de activos que pueden verse afectados por la amenaza, y se introduce el valor máximo de daños estimado que podría causar la amenaza en cada uno de los activos. Este valor se toma como referencia para calcular automáticamente el daño total, lo que serían las pérdidas financieras.

Se fija en el escenario la probabilidad de ocurrencia anual de una amenaza. Entendiéndose ésta como un valor comprendido en el rango 0-1. Siendo 0 el valor correspondiente a una amenaza cuya materialización durante un año consideramos improbable, y 1 para aquellas que estimamos se materializarán al menos una vez a lo largo de un año. El riesgo asociado a la amenaza se calcula automáticamente en función del daño total de la amenaza y su probabilidad de ocurrencia.

A la hora de construir un plan efectivo de medidas que mitiguen las distintas amenazas, el analista deberá seleccionar aquellas que le parezcan más apropiadas en función de diversos factores como su experiencia o el coste de implementación de éstas.

Para facilitar la elección de las contramedidas podemos plantearnos también preguntas como:

- ¿Cual será el nivel de mitigación que proporcionará cada contramedida frente a su amenaza si fuera ésta la única contramedida implementada en el plan?

- ¿Cómo afectan el resto de contramedidas del plan al riesgo concreto que supone esta amenaza?

Herramientas

Microsoft TAM v2.1

http://www.microsoft.com/downloads/details.aspx?FamilyID=59888078-9daf-4e96-b7d1-944703479451&displaylang=en

Desde mediados de este año (2006), Microsoft ha puesto a disposición del público de manera gratuita la versión 2.0 de una herramienta que facilita el Análisis y Modelado de Amenazas. Aunque inspirada en la versión original de la herramienta creada por Frank Swiderski (ex @stake) y que sirve de acompañamiento perfecto al libro del mismo autor titulado "Threat Modeling" [1], esta nueva versión, ya difiere sustancialmente de la original, favoreciendo el uso de la nueva metodología. El día 1 de Diciembre se ha actualizado a la versión 2.1, incorporando algunas mejoras en el asistente y en el sistema de plugins.

La impresión que uno tiene cuando prueba por primera vez esta herramienta es que se encuentra ante una aplicación madura, con un interfaz elegante, práctico y al mismo tiempo sencillo. Prueba de ello es la facilidad con la que se puede hacer un esqueleto básico de nuestro primer modelado siguiendo el asistente, para completarlo con posterioridad.

Entre las características destacables de esta última versión merece la pena citar las siguientes:

- Creación de un modelo básico mediante un asistente.
- Biblioteca por defecto de ataques con una guía descriptiva de contramedidas.
- Generación automática de amenazas y casos de uso.
- Navegación usando el componente treeview pudiendo visualizar todos los nodos expandidos de forma simultánea.
- Flujo de llamadas, ámbito de ataque, árbol de amenazas (exportables a Visio).
- Generación de informes y estadísticas exportables a html.
- Posibilidad de exportar las contramedidas y los casos de pruebas de ataque a un servidor Visual Studio Team Foundation Server (TFS).
- Tutoriales en video (muy básicos).

Otra funcionalidad digna de mención es la posibilidad que nos proporciona la herramienta para hacer uso de diferentes técnicas de medición del riesgo que se pueden incorporar al programa mediante plug-ins.

El Análisis y Modelado de Amenazas propuesto por Microsoft se utiliza para identificar las amenazas a las que están expuestas las aplicaciones en el momento mismo de su diseño. Para lo cual se siguen una serie de pasos:

1. Identificar los activos.
2. Crear una descripción de la arquitectura.
3. Descomponer la aplicación.
4. Identificar las amenazas.
5. Documentar las amenazas.
6. Asignar prioridades a las amenazas.

A partir de los requerimientos y la descripción de la arquitectura, la herramienta trata de identificar de manera automática las amenazas, al tiempo que produce una serie de elementos como son:

- Matrices de acceso a datos.
- Casos de uso.
- Diagramas de flujos de datos, de llamada, y de confianza.
- Superficie de ataque.
- Informes.

Desde la misma herramienta también se pueden asignar prioridades a cada una de las amenazas y especificar cada una de las respuestas, para así conformar la base de datos de amenazas.

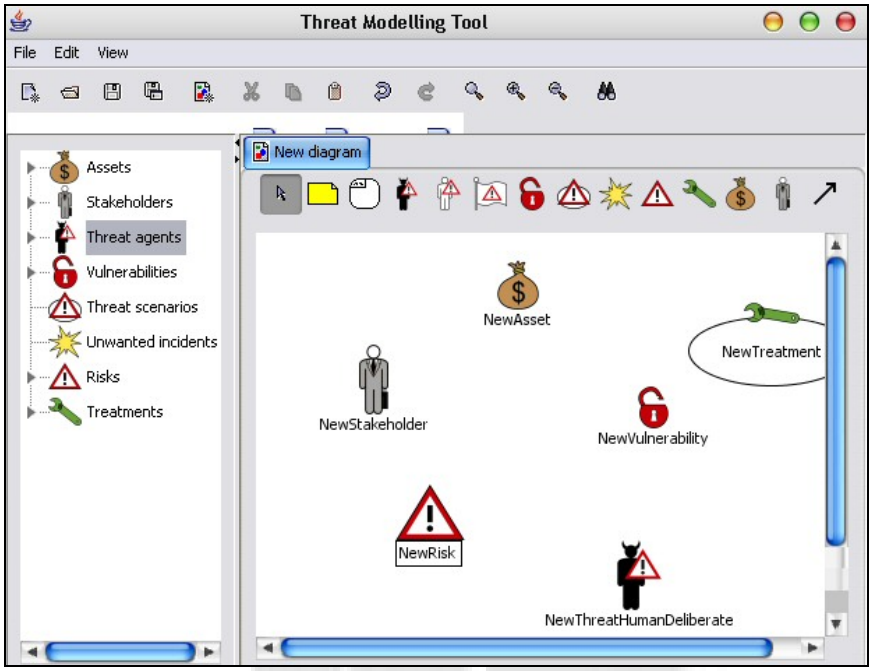
La herramienta ofrecida por Microsoft cumple su función con creces, de forma efectiva, simple y a mi parecer bastante intuitiva. Me gustaría aclarar que nunca he profesado excesivo fervor ni a favor ni en contra de aplicaciones, soluciones o sistemas operativos ni open source ni propietarias tampoco. Cada una tiene sus más y sus menos, pero en mi opinión y como se mostrará más adelante en este documento, esta es a fecha de hoy , posiblemente la mejor herramienta disponible (y gratuita). Por lo que me atrevo a decir, aunque sea por una vez, que con esto si se han ganado los "kudos". ;)

En lugar de acompañar este texto con algunas capturas de la herramienta, me ha parecido más interesante, preparar un video ilustrativo mostrando sus posibilidades a través del ejemplo que la acompaña sobre la aplicación ibuyspy.

El video está disponible en http://metal.hacktimes.com

CORAS

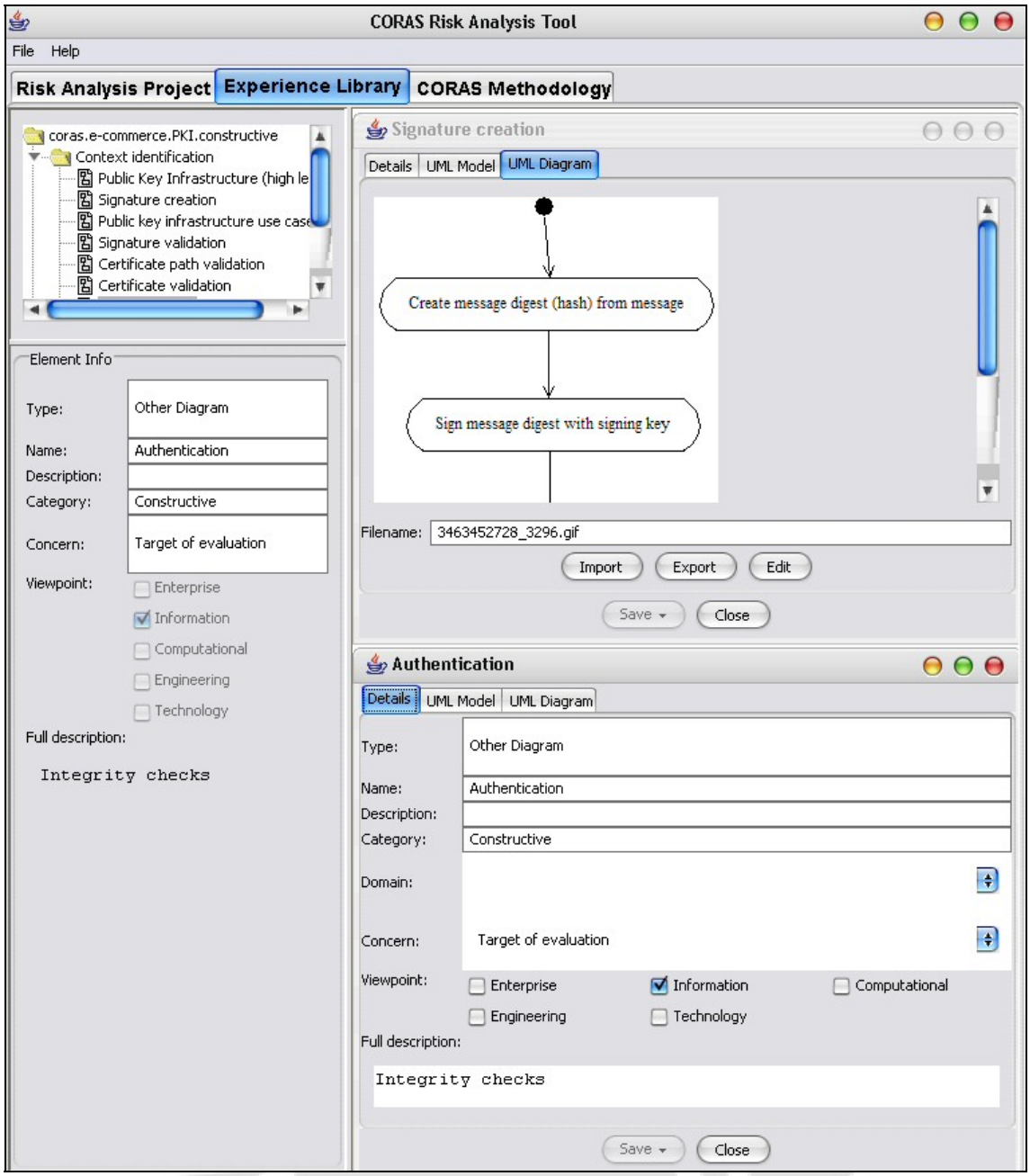
Son dos las herramientas que acompañan a la metodología CORAS. Un editor de diagramas, y la aplicación cliente-servidor. Ambas están basadas en Java. La aplicación principal permite crear nuevos proyectos de análisis, documentación, editar resultados de los análisis, generar informes así como reutilizar información procedente de otros análisis.



Editor de diagramas CORAS

El servidor está basado en tecnología EJB (Enterprise Java Beans) sobre un JBoss application server. El cliente proporciona una interfaz gráfica más o menos eficaz. Aunque es una lástima que la ejecución de ambas aplicaciones juntas afecten tanto al rendimiento de un equipo medio.

La herramienta, utiliza dos bases de datos a modo de repositorios. El repositorio de evaluación almacena todos los resultados procedentes del análisis, mientras que el repositorio de experiencia, contiene resultados reutilizables procedentes de anteriores análisis como modelos UML, procedimientos, o listas de comprobación.



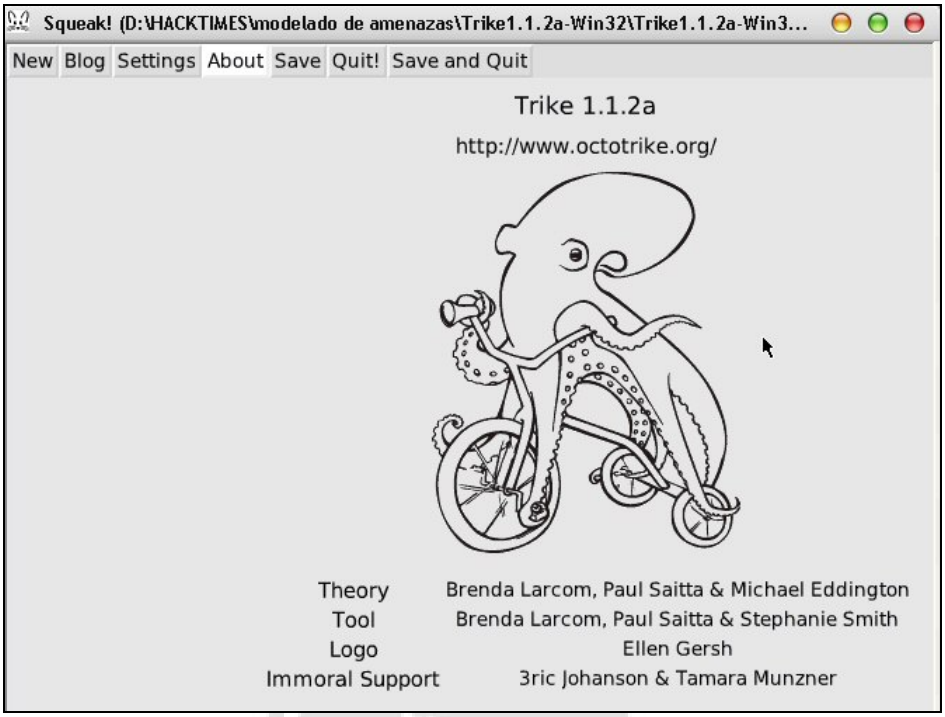
Herramienta de modelado

Además de los modelos UML, la herramienta también permite incluir tablas, diagramas de árboles de eventos, registros de detección de intrusiones y por supuesto descripciones en texto para acompañar el modelado. Se incluye además una API para su integración con otras herramientas.

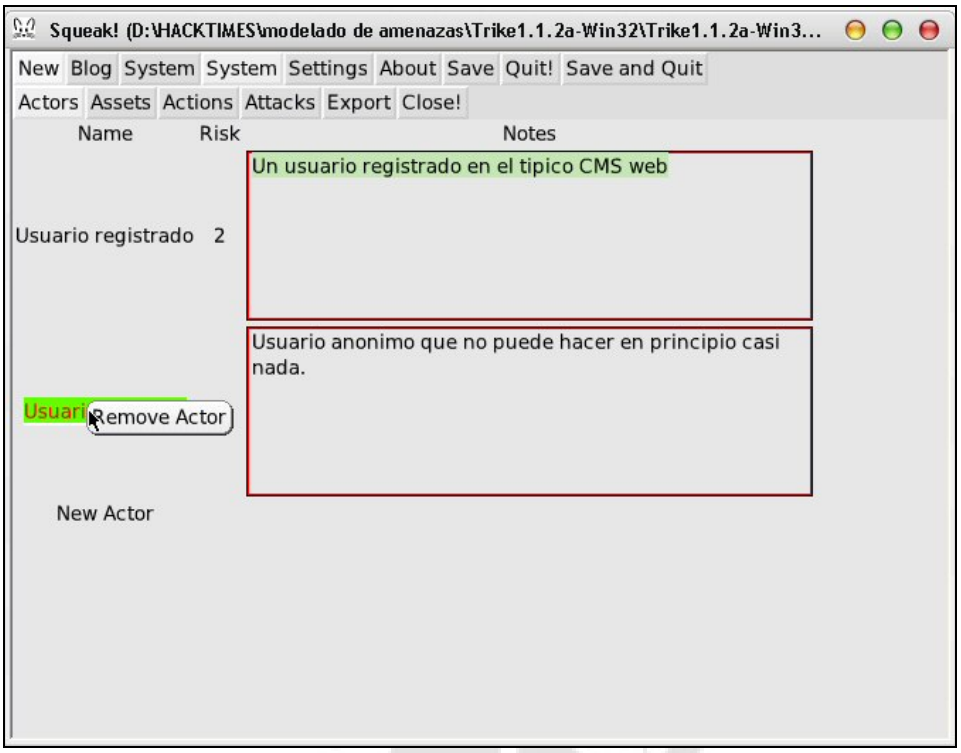
Trike

(Un pulpo en un triciclo)
http://www.octotrike.org

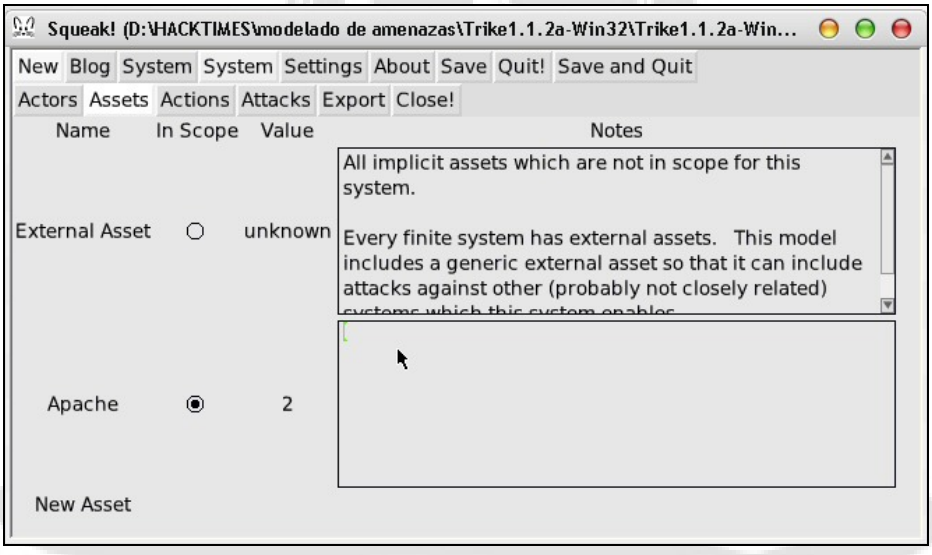
Este curioso "engendro" hecho en squeak (http://www.squeak.org), cuenta con "binarios" disponibles para win y mac osx. Como se puede apreciar en la siguiente captura de pantalla, ya sólo el interfaz le quita a uno las ganas.



Bien, una vez superado el susto inicial, la esperanza de encontrar algo realmente práctico se diluye tras las siguientes pantallas.

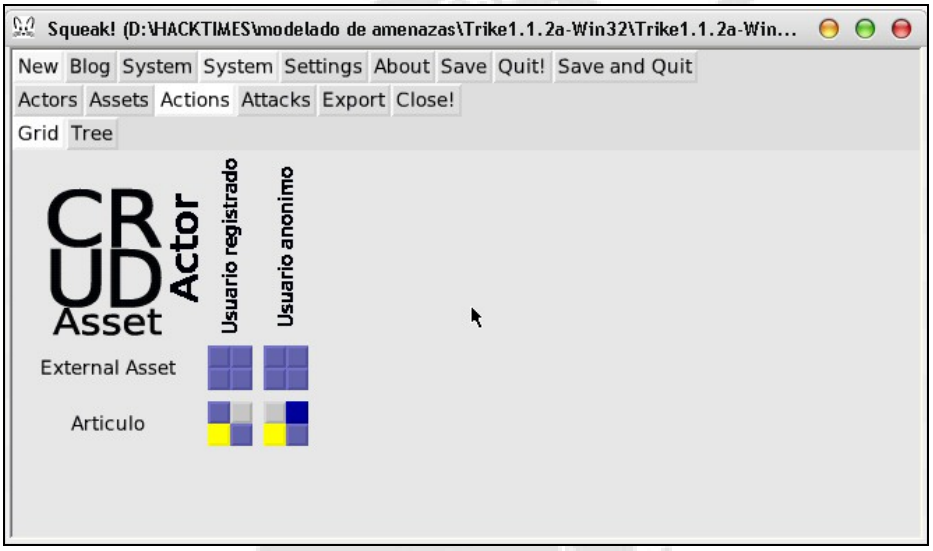


Remove Actor, quizá sea una buena idea. Sigamos.



Identificando activos.
Parece dificil rellenar toda esta información a dos manos!

Vista de las posibles acciones en forma de árbol, con sus nodos y demás.



Desconozco el significado de los colores que se le pueden dar a los diferentes permisos: create, read, update, delete. Supongo que se refieren a las distintos casos de uso (permitido, permitido según la condición, denegado).

Curiosa herramienta, como prueba de concepto supongo que da el pego, pero a la hora de la verdad me cuesta creer que alguien pueda hacer un uso real de ella.

Parafraseando al mítico Bruce Lee me atrevería a decir no sin por ello dejar de hacer un poquito de guasa … "If you put water into a bottle and it becomes the bottle, you put squeak into a teapot and it becomes …. an octopus in a bicycle." ;) Mejor doy por finalizado el testeo. Sólo puntualizar que la otra maravilla que no he mostrado es el menú que permite exportar a un archivo xml. Ah, y la información no se actualiza si se modifican los datos introducidos anteriormente.

En definitiva, una aplicación, a día de hoy, muy verde, y carente a mi parecer de toda posibilidad práctica de uso. Para haber sido presentada en ToorCon ha resultado toda una decepción. Por suerte, hemos visto que hay alternativas.

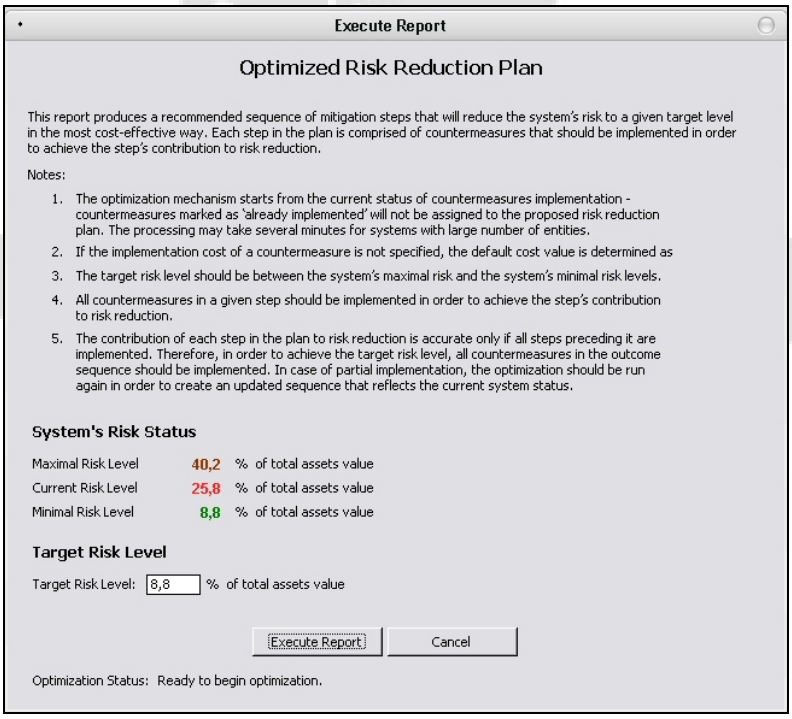
PTA

La herramienta de modelado que PTA pone a disposición del público es gratuita para estudiantes, investigadores, desarrolladores de software y consultores de seguridad independientes.

Esta aplicación esta basada en el motor de access, y durante las pruebas que he realizado me he encontrado en más de una ocasión con errores de la aplicación, llegando incluso a colgarse esta al entrar en un bucle infinito. Resulta curioso también el hecho de que entre los requerimientos se advierta la necesidad de contar con permisos de administrador del sistema para ejecutarla. Aún así, parece una herramienta interesante.

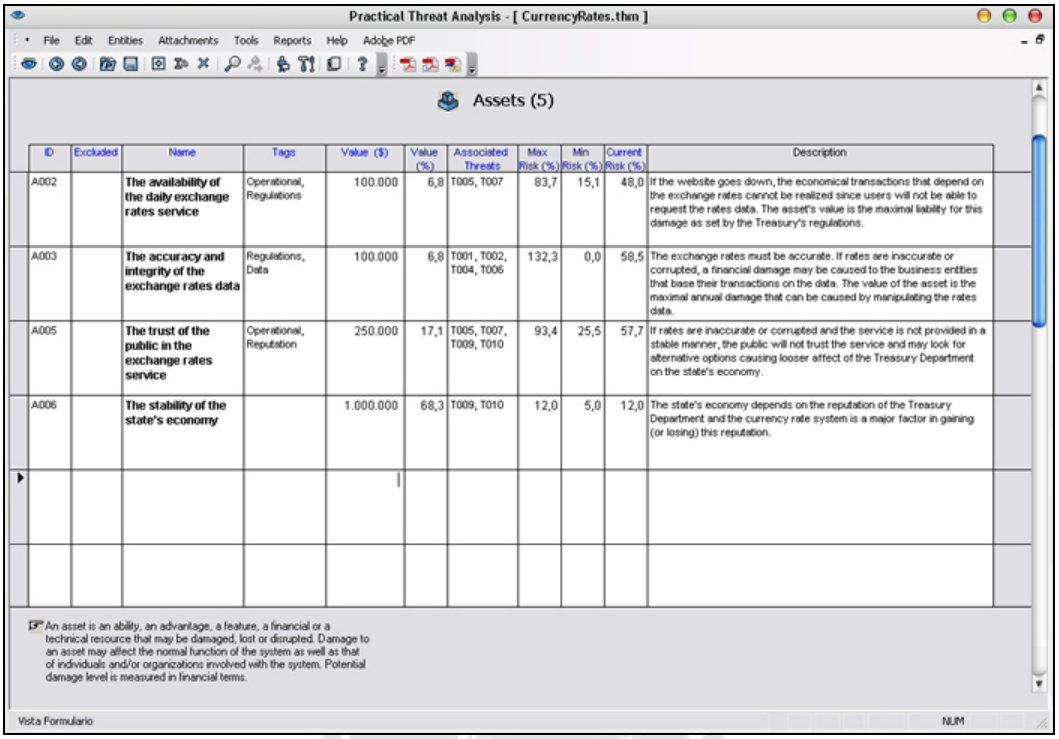
Como puntos más destacables se pueden citar los siguientes:

- El esquema de la base de datos se puede personalizar para incluir más tipos de entidades como pasos de auditoría, topología …etc.
- Permite la importación automática de datos de entidades y sus parámetros desde fuentes externas como escáneres de vulnerabilidades (Nessus) o appliances de red.
- Los distintos métodos de cálculo se pueden adaptar a diferentes situaciones. Por ejemplo los valores financieros que se asignan a los distintos activos se pueden calcular utilizando diferentes fórmulas.
- Utiliza librerías de entidades. Permite elaborar check lists en conformancia con diferentes estándares de seguridad como ISO 17799 y BS7799.
- Permite la elaboración de un variado número de informes. En el siguiente ejemplo se muestra la creación de un plan de medidas para mitigar las amenazas en función de un determinado nivel de riesgo.

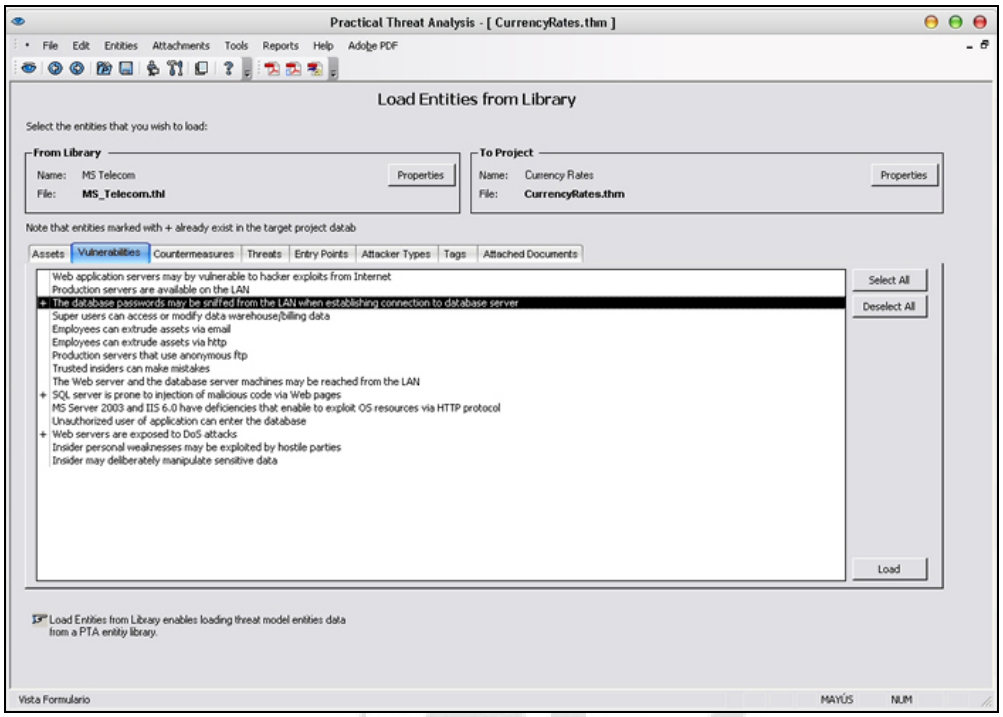


PTA es una herramienta que esta pensada para facilitar el trabajo al analista y permitirle utilizar hasta cierto punto su propia metodología.

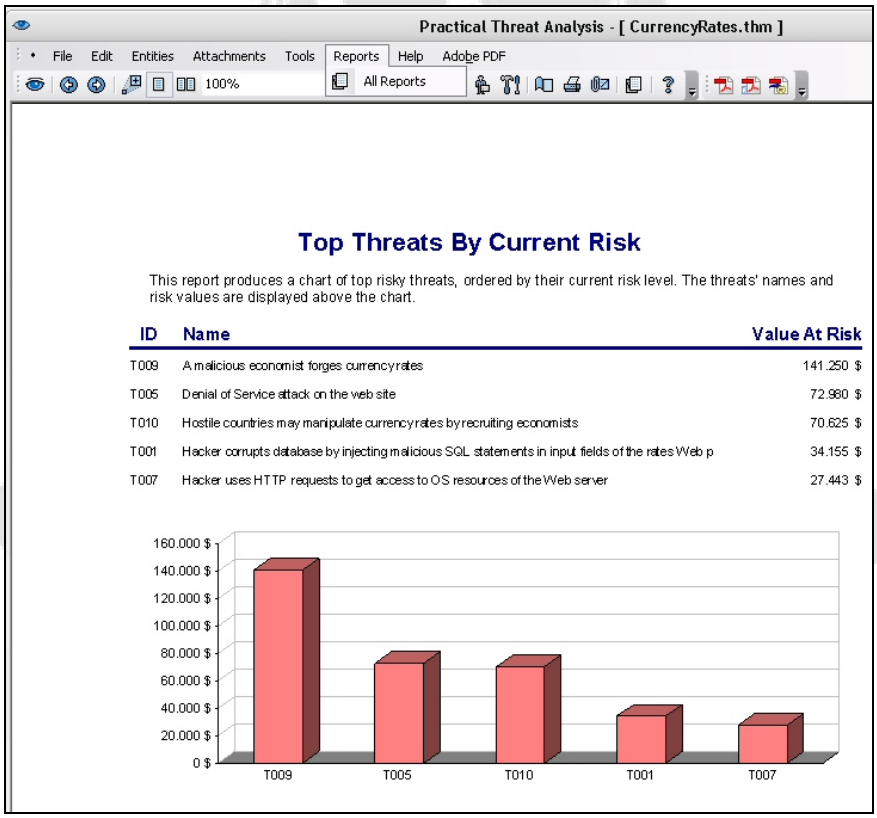
El hecho de que se puedan cargar librerías externas de los distintos elementos que componen nuestro modelado, permite un cierto grado de flexibilidad, lo que facilita la creación de checklists y documentación que se ajuste a los procedimientos de evaluación que requieren los diferentes estándares como: (ISO/IEC) 17799, British Standard (BS) 7799, System Security Engineering Capability Maturity Model (SSE-CMM), (OCTAVE), (COBIT), (ITIL), (SARA), Business Impact Analysis (BIA).



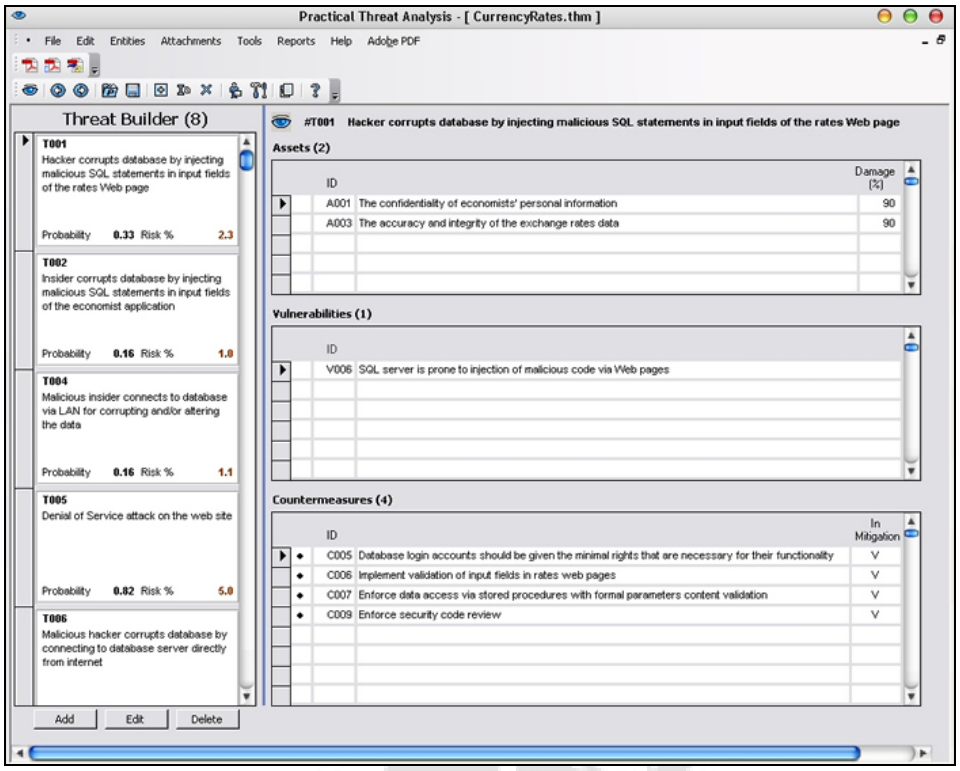
Pantalla de definición de activos, dónde podemos introducir datos como su descripción, asignarles valores o ver las amenazas que les afectan.



Selección de las distintas librerías que nos permitirán añadir entidades a nuestro proyecto.



Ejemplo de un informe generado con PTA mostrando el ranking de amenazas generado por nuestro modelado de acuerdo a su nivel de riesgo.



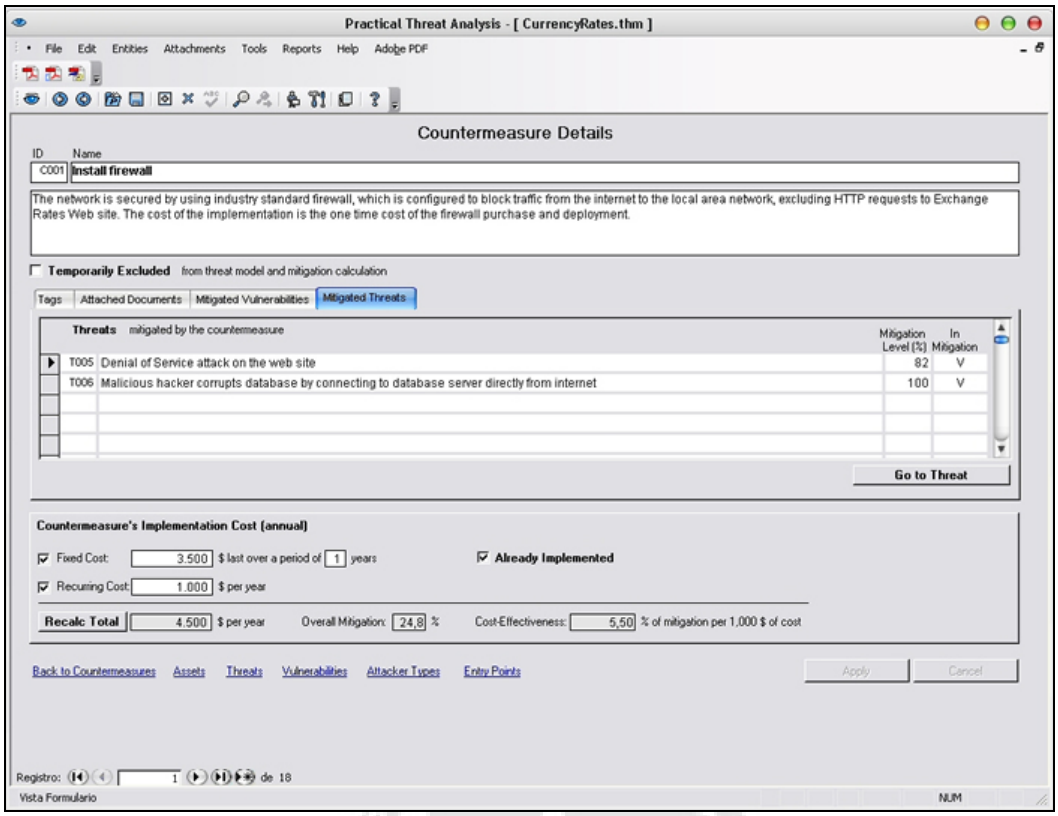
Creación de las amenazas. Podemos añadir nuevos activos, vulnerabilidades, contramedidas y amenazas que afectan a nuestros activos, incluso especificar el nivel de daño estimado de estos.

Informes en función del valor ROSI (Return On Security Investment):

Resulta particularmente interesante la posibilidad de generar informes sobre los distintos pasos para mitigar las amenazas en función del retorno de la inversión en seguridad. Al generar este tipo de informe, PTA realiza los cálculos de acuerdo a la siguiente fórmula:

$$ROSI = \frac{(\sum \text{Valor en riesgo} * (\text{Nivel de mitigación}/100)) - \text{Coste de mitigación}}{\text{Mitigation Cost}} * 100$$

- ∑ Sumatorio de todas las amenazas mitigadas por el plan
- El valor en riesgo (Risk Exposure – Annual Loss Expectancy) es el daño que causa la amenaza multiplicado por la probabilidad de ocurrencia de la amenaza , es decir, el número de veces que se espera se materialice la amenaza por año.
- El nivel de mitigación (porcentaje) es el estimado que proporciona el plan.
- El coste de mitigación se refiere al coste anual de implementar todas las contramedidas presentes en el plan.



PTA nos permite profundizar también en el detalle de los distintos elementos que conformarán el modelado. Incluso fijar un coste estimado para las contramedidas, y llevar un seguimiento de su grado de implementación en la fase actual.

Existe también una versión Enterprise, de pago que incluye librerías para ISO 17799 y BS 7799 2002. Se puede encontrar información más detallada sobre sus características en la web de PTA Technologies.

http://www.ptatechnologies.com/PTAEnterprise.htm

Conclusiones

Conviene no olvidar que el éxito de implementar un modelado de amenazas pasa por realizar sucesivas iteraciones durante la evolución y desarrollo del proyecto en cuestión, al tiempo que se debe intentar favorecer la reutilización de componentes que puedan servirnos para posteriores mejoras o para su uso en otros proyectos que tengan cierta similitud. Esta reutilización de componentes se hace particularmente efectiva en el caso de las aplicaciones web, ya que en numerosas ocasiones su propia naturaleza hace que sea frecuente la presencia de idénticas funcionalidades en diferentes aplicaciones.

Las revisiones y mejoras de nuestro modelado se deben ir incorporando a medida que se produzcan cambios en el entorno de la aplicación ó sistema, su diseño, implementación, configuración y por supuesto en el momento en el cual se modifiquen o aparezcan nuevos casos de uso.

De todos es sabido que la incorporación de medidas de seguridad en las técnicas de programación e implementación de cualquier sistema informático puede no parecer rentable a corto plazo. Pero las malas prácticas nos terminan demostrando que a medio-largo plazo esta percepción es totalmente errónea. Es por esto que el uso de este enfoque viene a desempeñar un papel fundamental en todo ciclo de vida de desarrollo que aspire a considerarse seguro.

Como se ha mostrado a lo largo de este documento, el modelado de amenazas se puede enfocar desde diferentes perspectivas, y en última instancia la propia subjetividad de conceptos como "amenaza" y "riesgo" es un proceso abierto a diferentes interpretaciones. Aún así, el seguimiento de una metodología y un enfoque de mejora continua siempre resultará un factor clave de cara a mejorar la seguridad de una aplicación o un sistema.

Sin duda, la divulgación del análisis y modelado de amenazas, se nos muestra a priori, como una oportunidad interesante de utilizarla como catalizador para realizar una concienciación efectiva de la seguridad, y también como un ejemplo de los beneficios que nos pueden aportar unas buenas prácticas iniciadas desde las etapas más tempranas del desarrollo. El tiempo nos dirá cuales son las metodologías y las herramientas más efectivas y también si hemos sabido sacarle provecho realmente. Veremos ;)

Agradecimientos

No me gustaría dejar pasar la oportunidad de mostrar mi agradecimiento a tres personas que me han servido de apoyo en más de una ocasión, también durante la redacción de este artículo con sus comentarios y sugerencias:

Iñaki López (a.k.a –ilo www.reversing.org) , Juan de la Fuente (a.k.a Freed0m www.hacktimes.com) y Yago (www.security-projects.com). Gracias a los tres por estar siempre ahí soportando mis locuras ;)

También expresar mi agradecimiento a S.R.F (Microsoft Ace Team) por sus matizaciones respecto a la herramienta y metodología de Microsoft.

Cualquier feedback, ampliación o discusión sobre el tema tratado en el artículo será bien recibido, ya sea vía e-mail o a través de los foros de hacktimes.

Espero que la lectura haya sido tan grata para usted como lo ha sido para mí la redacción de estas líneas, y al mismo tiempo, lo suficientemente interesante para despertar un poco su curiosidad por el tema expuesto en esta ocasión.

Un saludo.

Daniel P.F
a.k.a metal AT/DOT hacktimes.com

Referencias de interés

Si se desea profundizar en el análisis y modelado de amenazas, estas son algunas referencias de posible interés para el lector:

- Guerrilla Threat Modeling
  http://blogs.msdn.com/ptorr/archive/2005/02/22/GuerillaThreatModelling.aspx

  Un artículo bastante interesante que pretende simplificar el proceso de modelado de amenazas para aquellos casos en los que no se desea o no se dispone del tiempo suficiente para realizar un modelado de amenazas de forma exhaustiva. Enfocado como una guía de revisión de los puntos más importantes de un modelado, para saber si es conveniente o no profundizar más en el análisis de cada componente individual.

- Rapid threat modeling
  http://www.blackhat.com/presentations/bh-usa-05/bh-us-05-aggarwal-update.pdf

- Threat Modeling (Microsoft Press)
  http://www.microsoft.com/MSPress/books/6892.asp

  El libro oficial de Microsoft que trata en detalle todo el proceso de modelado de amenazas, con ejemplos ilustrativos de diferentes escenarios.

- Writing Secure Code 2nd Edition (capítulo 4)
  http://www.microsoft.com/mspress/books/5957.asp

- Value Driven Security Threat Modeling Based on Attack Path Analysis
  http://sunset.usc.edu/events/2006/CSSE_Convocation/publications/ChenValueBasedSecurityThreatModel.pdf

  Este documento propone un método cuantitativo, basado en un análisis de los árboles de ataques sobre sistemas que utilizan soluciones comerciales estándar.

- Attack Trees (Bruce Schneier)
  http://www.schneier.com/paper-attacktrees-ddj-ft.html

  Un documento supuestamente interesante del mítico Bruce Schneier tratando los árboles de ataques. Personalmente casi me quedaría con unas risas entre lectura y lectura (http://geekz.co.uk/schneierfacts/).

- Application threat modeling resources (Microsoft)
  http://msdn2.microsoft.com/en-us/security/aa570413.aspx

- Using the CORAS Threat Modelling Language to Document Threat Scenarios
  http://folk.uio.no/massl/uml-sa/SINTEF-deseca-report.pdf

- Model-based risk assessment – the CORAS approach
  http://folk.uio.no/massl/publications/nik02-coras.pdf

- Building Security Into The Software Life Cycle
  http://www.blackhat.com/presentations/bh-usa-06/bh-us-06-Morana-R3.0.pdf

Glosario de términos

A continuación se ha intentado aclarar algunos de los conceptos empleados a lo largo de este documento. Se advierte al lector que las distintas metodologías pueden hacer diferentes matizaciones sobre estas definiciones, por lo que conviene revisar la documentación oficial en el caso de decantarse por una de ellas.

Activo: Se trata de un recurso de valor que se debe proteger. Pueden ser activos intangibles como la reputación de su empresa, o tangibles como la información que se procesa o los datos de un cliente. También se podría considerar un activo un recurso que utilizado de forma indebida facilite el acceso no autorizado o provoque la no disponibilidad de otro activo.

Amenaza: Un evento de posible ocurrencia que podría dañar o comprometer un activo o un objetivo estratégico.

Vulnerabilidad: Una vulnerabilidad es un punto débil que de explotarse con éxito puede llegar a originar la consecución de una amenaza.

Ataque: Una acción que se sirve de una o varias vulnerabilidades para materializar una amenaza.

Riesgo: la probabilidad (cuantificable) de sufrir una pérdida debido a una amenaza materializada. Su valor depende de dos factores: la frecuencia con la que ocurre la amenaza; el impacto del daño que puede causar.

Incidente no deseado: Evento que podría dañar o reducir el valor de los activos.

Contramedida: Se establece para hacer frente a las vulnerabilidades y reducir la probabilidad de sufrir un ataque o el impacto que pueda originarse de la consecución de una amenaza.

Principio de menor privilegio: Se basa en no conceder más privilegios de los absolutamente necesarios.

Separación de privilegios: Evitar que las operaciones se basen en una única condición.

Confidencialidad: Sólo los usuarios debidamente autorizados deben tener acceso a los datos y/o los recursos de un modo apropiado.

Integridad: Sólo los usuarios debidamente autorizados pueden modificar los datos y/o los recursos de modo apropiado y controlado.

Disponibilidad: Los recursos deben estar disponibles cuando son necesarios y deben funcionar a un nivel aceptable.

No repudio: Se basa en el hecho de que las acciones realizadas por los usuarios deben quedar registradas de un modo tal que estos no puedan negar posteriormente el haberlas realizado.

Oviedo , 18 de Diciembre de 2006