Modelamiento de Amenazas en el Desarrollo de Software

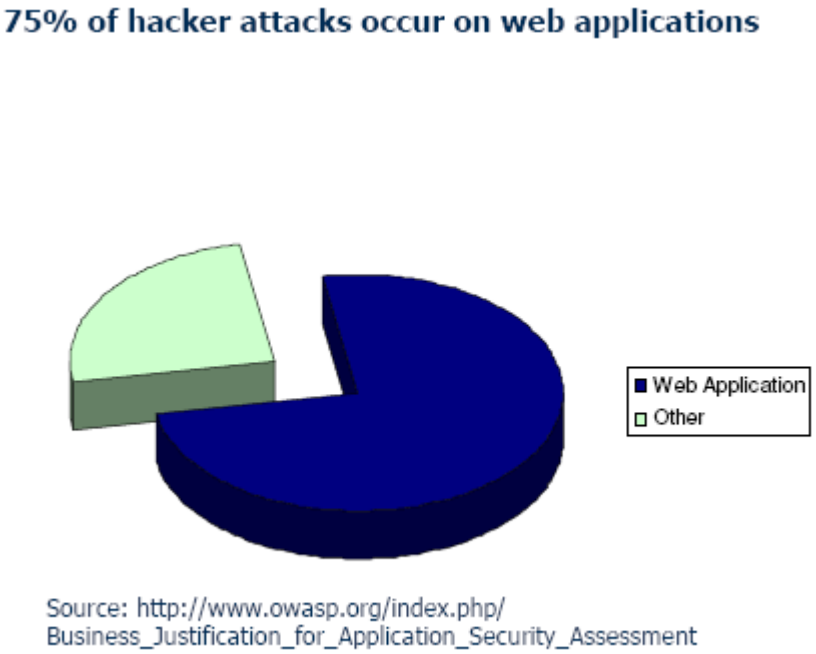
Davor A. Pavisic
Jalasoft CTO

Agenda

- **La importancia**
- **Definición**
- **Beneficios**
- **Metodología**
- **Conclusiones**

Porque Modelar Amenazas?

- **La seguridad de aplicaciones es un problema importante.**



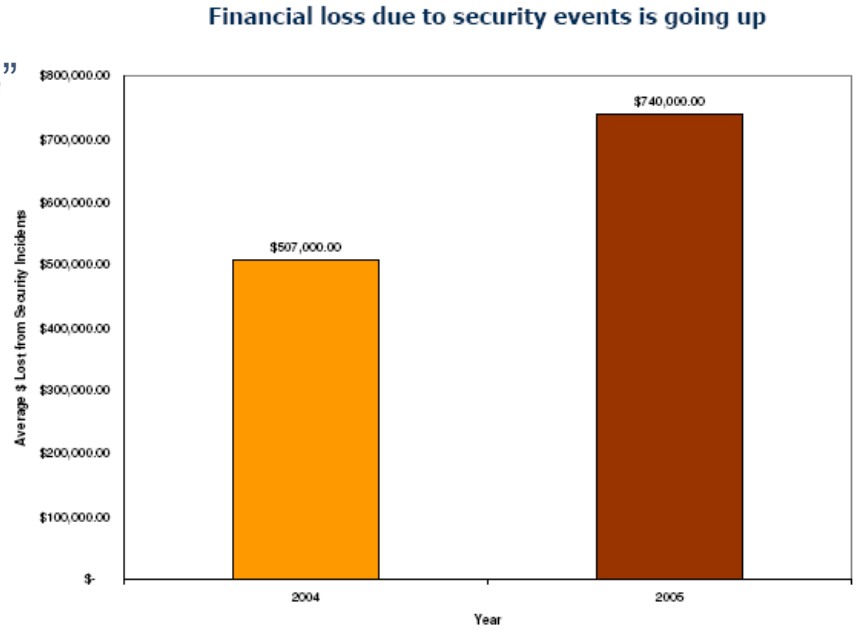
Source: http://www.owasp.org/index.php/
Business_Justification_for_Application_Security_Assessment

Porque Modelar Amenazas?

- **Las perdidas financieras a causa de los problemas de seguridad están en aumento:**

Costos de una vulnerabilidad explotada:

- Ventas/otros por Aplicación “off line”
- “Incident Response Team”
- Costo de desarrollar un “patch”
- Costo de testear el “patch”
- posibles juicios
- Reputación de la compañía



Source: http://www2.csoonline.com/info/release.html?CID=24531

Porque Modelar Amenazas?

- **Errores de Seguridad en el Diseño son prevalentes.** 47% de los defectos de software tienen que ver con seguridad y pudieron ser evitados durante el diseño. La mitad de ellos son de mediano a alto impacto y fácilmente explotables.
- **Corregir Errores de Diseño es costoso.** El costo de corregir un error durante el diseño es 7 veces mas barato que durante la implementación y 100 veces mas barato que corregirlo cuando esta en producción.
- **El modelado de amenazas es uno de las mejores formas de resolver este problema:** 70% de los problemas encontrados durante las pruebas de seguridad fueron descubiertos con el modelado de amenazas.

Evaluación de la Seguridad en Aplicaciones

- **Errores de Diseño relacionados a seguridad**
  - Causados por la ausencia de requerimientos de seguridad, conocimiento de diseño orientado a la seguridad, etc.
  - No pueden ser identificados vía herramientas de seguridad ya que estas no tienen el conocimiento contextual de la aplicación
  - Pueden ser identificados usando el modelado de amenazas.
- **Bugs relacionados a seguridad**
  - Bugs en el código pueden fácilmente resultar en vulnerabilidades
  - Pueden ser identificados con herramientas de análisis de código.
  - Requieren que los desarrolladores entiendan los problemas de seguridad y sigan estándares seguros de codificación.

Definición del modelado de amenazas.

- **Un enfoque sistemático y estratégico para enumerar y cuantificar amenazas dentro del entorno de una aplicación con el objetivo de minimizar el riesgo y sus niveles asociados de impacto.**
  - Permite al personal de seguridad comunicar el daño posible y priorizar los esfuerzos para mitigar los errores de seguridad.
  - Es simplemente tomar el rol de un atacante y pensar en esos términos.

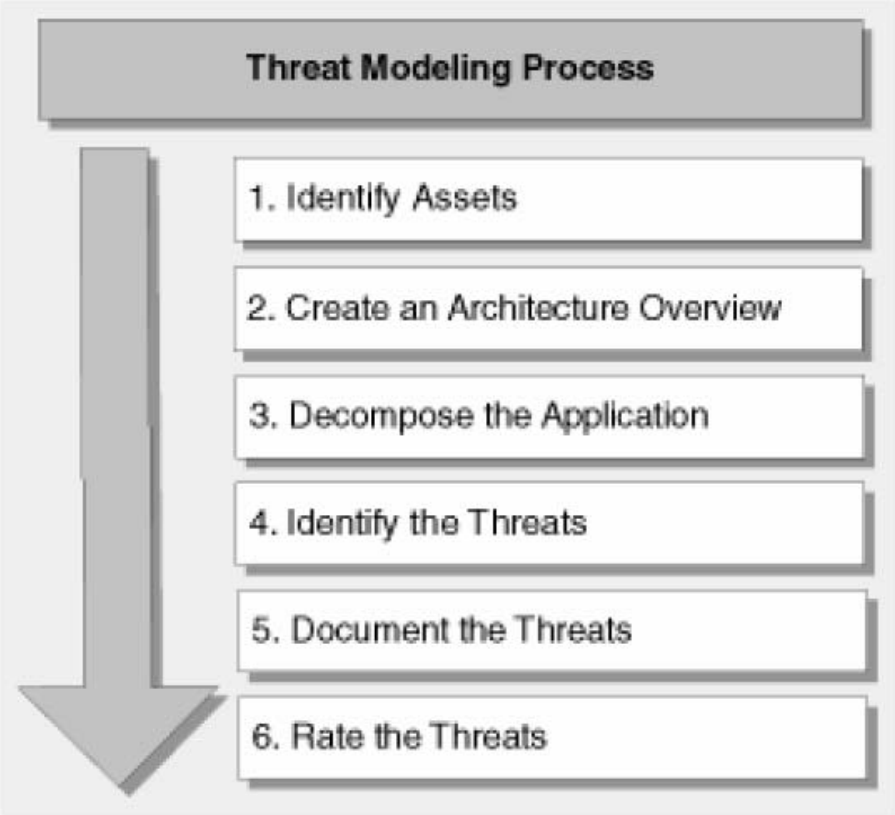
Definición del modelado de amenazas

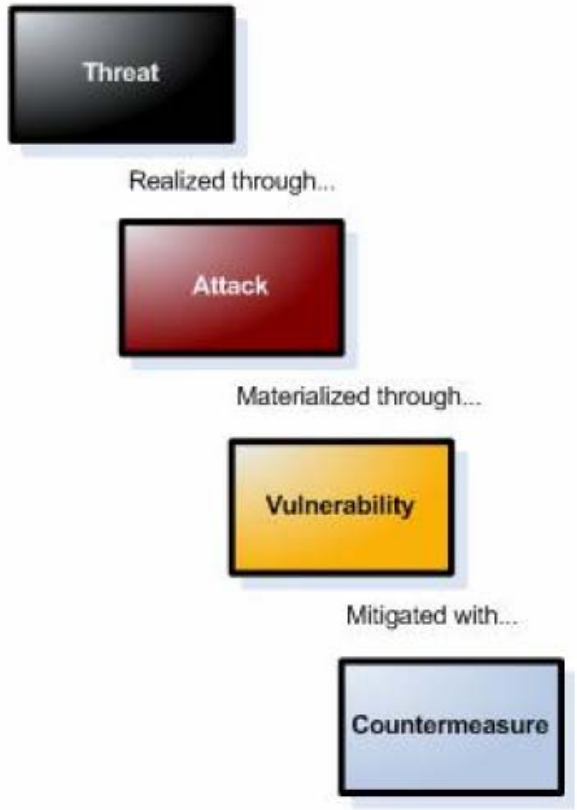
- **Activos (recursos/assets):**
  - Que datos valiosos y equipos deben ser protegidos?
- **Determinar posibles amenazas**
  - Que puede hacer un atacante al sistema?
- **Vulnerabilidades.**
  - Que fallas en el sistema pueden permitir a un atacante realizar su amenaza?

Quienes se benefician?

- **El modelado de amenazas provee diferentes beneficios a las partes interesadas dependiendo de su rol y responsabilidad:**
  - Arquitectos
  - Desarrolladores
  - Ingenieros de control de calidad
  - Gerentes de proyecto
  - Consultores
  - Etc.
- **El modelado puede ser realizado por cualquiera de los roles arriba mencionados.**

Metodologia de modelado de amenazas (version Microsoft)





El proceso de modelado de amenazas

- **El punto de vista del adversario.** Es imprescindible ver el sistema desde este punto de vista. Threat Modeling toma el enfoque de “afuera hacia adentro” que es lo que normalmente haría un adversario.
- **Puntos de Entrada, Activos, y niveles de acceso.** Entre los datos que un adversario desearía obtener para ver la factibilidad de atacar un sistema con cierto beneficio están estos.

Puntos de entrada.

- **Cualquier lugar donde donde se transfieren datos/control entre el sistema que se modela y otro sistema….**
  - Web services
  - El API de una aplicación
  - Sockets abiertos
  - Llamadas a procedimientos remotos (RPCs)
  - Incluso una consola con línea de comandos
- **Para identificar los puntos de entrada….** Uno se debe preguntar como un adversario puede interactuar con el sistema.
  - Los puntos de entrada muestran todos los lugares por donde un adversario podría atacar el sistema

Recursos (Assets)

- **Son los recursos que el adversario desea:**
  - Obtener
  - Cambiar
  - Destruir
- **Pueden ser tangibles o abstractos**
  - Numero de tarjeta de crédito en una DB
  - O por ejemplo la consistencia de datos.
- **Es imposible tener una amenaza si no hay un activo**

Niveles de confianza (trust levels)

- **Estos definen como entidades externas son caracterizadas por el sistema.**
  - Generalmente están relacionados con privilegios o credenciales
  - Estos deben ser relacionados a los puntos de entrada y a los assets.
  - Estos niveles de confianza definen el privilegio que una entidad externa debería tener para utilizar de forma legitima un punto de entrada o funcionalidad en el punto de entrada…. Esto implica directamente a que activos esta entidad externa puede tener.

Caracterizando la seguridad del sistema

- **Definir escenarios de uso (use scenarios).**
  - Como los componentes serán usados (o, como no deberían ser utilizados).
- **Definir dependencias y listar supuestos**
  - Librerías externas y supuestos relacionados a seguridad que pueden afectar a la seguridad.
- **Modelar el sistema**
  - Diagramas de flujos de datos o similares son importantes
  - Estos responden a preguntas como como son afectados los recursos protegidos o en que ocasión una entidad externa puede manipular un recurso.

Determinar las amenazas

- enumerar las amenazas crea el perfil de amenaza para un sistema que los arquitectos y desarrolladores deben tratar de mitigar.
  - **Identificar**: para cada punto de entrada, tratar de determinar como un adversario podría tratar de afectar un recurso.
  - **Analizar**: el equipo de desarrollo modela las amenazas para determinar si estas están mitigadas. Usando Arboles de Amenazas, es posible descomponer una amenaza a varias condiciones individuales que pueden ser verificables.
  - Las amenazas que no son mitigadas se tornan en vulnerabilidades.

Clasificacion de Amenazas “STRIDE”

- **Una clasificacion de acuerdo a los efectos:**
  - **Spoofing**. Permite al adversario pasar como un usuario, componente u otro sistema que tiene una identidad en el sistema que esta siendo modelado
  - **Tampering**. Modificación de datos dentro del sistema para lograr un objetivo malicioso.
  - **Repudiation**. La habilidad de un adversario para denegar la ejecución de una actividad porque el sistema no tiene suficiente evidencia para verificar la no autenticidad del adversario.
  - **Information Disclosure**. Exposición de información confidencial/protegida.
  - **Denial of Service**. Cuando se puede evitar que usuarios legítimos puedan acceder a la funcionalidad normal del sistema
  - **Elevation of Privilege**. Cuando un adversario usa medios ilegítimos para asumir un trust level que tiene diferentes privilegios que los que tiene normalmente.

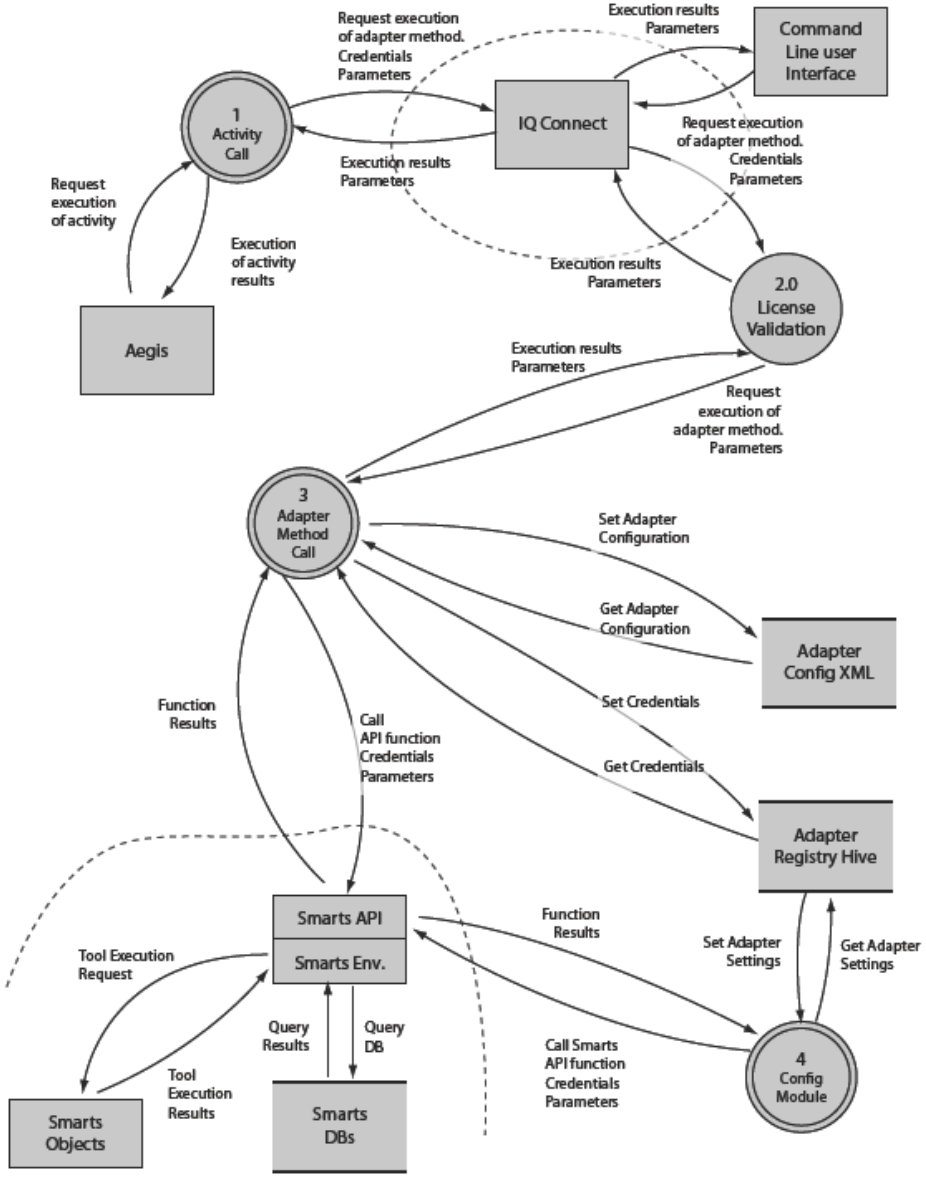
Recursos (assets)

Protected Resources

The Protected Resources table describes the data or functionality that the component needs to protect.Â It lists the minimum Access Category that should be allowed to access the resource.

Protected Resources			
ID	Name	Description	Trust Level
1	Smarts DB	This is the Smarts Data Base. This DB is distributed among all the available Domain Servers and has its own internal format.	(4) Smarts User / Password.
2	Smarts Objects	Objects registered in the Smarts environment can range from Network Devices (routers, switches, etc.) to applications. These objects can be manipulated through a special set of â€œSmarts Toolsâ€. These tools can be scripts or even executables configuredd by the Smarts user. Using these tools, a Smarts User could do almost anything in the environment. Having special security controls on the execution of these tools is critical to the well beeing of the whole Smarts Environment.	

Diagramas de Flujo de Datos – Nivel 1



Escenarios de Uso (ejemplo)

Use Scenarios

The Use Scenario table provides information about the expected use of the component. Using or deploying the component in a way that violates a Use Scenario may impact the security of the component.

ID	Description
1	The adapter will be installed in an environment where Aegis has been previously installed and will comply with all requirements. Aegis users and corresponding trust levels are assumed to be correct. Credential validations for accessing IQConnect are assumed to be correct. The adapter must have access to a OpsMgr API and an OpsMgr environment. The adapter is not liable for verifying the authenticity of such API nor preventing such API from being tampered. The OpsMgr user that will be used to access the OpsMgr environment through the OpsMgr API should have sufficient privileges to accomplish the required actions in OpsMgr through the provided adapter activities.
2	The adapter will be installed in an environment where Aegis has been previously installed and will comply with all requirements. Aegis users and corresponding trust levels are assumed to be correct. Credential validations for accessing IQConnect are assumed to be correct. The adapter must have access to a OpsMgr API and an OpsMgr environment. The adapter is not liable for verifying the authenticity of such API nor preventing such API from being tampered. The OpsMgr user that will be used to access the OpsMgr environment through the Smarts API should have sufficient privileges to accomplish the required actions in OpsMgr through the provided adapter activities. The Aegis web site will be installed in a different IIS server than where the Aegis server is installed.

Niveles de confianza (Trust Levels)

Trust Levels

The Trust Levels table describes privilege levels that are associated with Entry Points and Protected Resources.

ID	Name	Description
1	Adapter	If the Client Context = null then the caller is another provider and full access is granted.
2	Super-User	If the Client Context 6= null then a method is called to verify if the client is a super user. If (isSuperUser = true) then full access is granted
3	other client	If the Client Context 6= null then a method is called to verify if the client is a super user. If (isSuperUser = false) then a static PSC_GUID dataAccessPermission = PSC_GUID::fromString("A1A2A3A4-0000-0031-0001-A5A6A7A8A9A0") is created and then a method is called bool allowed = false; getProviderManager()->secACLIsUserAllowedTo(clientContext,PSC_String(), dataAccessPermission, allowed). This method will return if access is granted or not. If granted, full access is provided
4	Smarts User / Password.	A valid Smarts Server user and corresponding password to access the SMARTS API.

Dependencias Externas (external dependencies)

External Dependencies

The External Dependency lists dependencies on other components or products that can impact security.Â These are assumptions that are made about their usage or behavior.Â Inconsistencies can lead to security weaknesses.

ID	Description
1	The OpsMgr API. The adapter uses this API to access OpsMgr servers.
2	Apache Xerces. The adapter uses this library for xml manipulation.
3	Win 32 dpapi. The adapter usses this library for local encryption.
4	Microsoft runtime libraries.

Puntos de entrada/salidada (Entry Points)

Entry Points

The Entry Points table describes the interfaces through which external entities can interact with the component, either through direct interaction or indirectly supplying it with data.

Entry Point			
ID	Name	Description	Trust Level
1	GetAuditTrail	Method of the JSIQSMARTS_Notification class. Returns the audit trail for a specified Notification.	(1) Adapter (2) Super-User (3) other client
2	GetImpactedNotificationNames	Method of the JSIQSMARTS_Notification class. Gets notification names that are explained by the selected notification.	(1) Adapter (2) Super-User (3) other client

Amenazas (threats)

Threats

The Threats to the component are listed in the Threats tables.Â These do not imply vulnerabilities.Â Rather, they are actions that a malicious external entity might try to do to exploit the system.

Threats	
Threat	
ID	1
Name	Adversary changes status/ownership of a notification
Description	Adversary is able to Acknowledge / Unacknowledge and take / release ownership of a Smarts Notification. This would allow the adversary to reduce the importance of a network problem or vice versa.
STRIDE Classification	Tampering
Mitigated?	Yes
Known Mitigation	The IQ Connect access password should not be disclosed by end users Related Use Scenarios: (1) (2) Related External Security Notes: (1) Several threats have been identified related to the disclosure o...
Investigation Notes	
Entry Points	(1) GetAuditTrail
Protected Resources	(1) Smarts DB (2) Smarts Objects
Threat Tree	

Vulnerabilidades

- **Una amenaza no mitigada se convierte en una vulnerabilidad que puede ser explotada.**
- **Es importante conocer las vulnerabilidades existentes para poder tomar una decisión: arreglar o correr el riesgo?**

Conclusiones

- **El modelado de amenazas es simplemente un sistema para determinar y cuantificar amenazas y vulnerabilidades.**
- **Es posible que implique un poco de trabajo adicional en el proceso de desarrollo de software pero ampliamente justificable si se considera el "ROI".**