Diseñando y Documentando la Arquitectura de Seguridad

Por Luis Daniel Benavides Navarro

La presentación

- Basada en el libro: Rozanski N, Woods E. "Software Systems Architecture" Addison-Wesley. 2005

Antes de empezar

- **La arquitectura** de un sistema es la estructura o estructuras del sistema, que comprenden elementos de software, las propiedades externas de esos componentes, y las relaciones entre ellos.
- **Descripción arquitectural** es un conjunto de productos que documentan la arquitectura de una forma que los stakeholders puedan comprender, y de forma que demuestre que cumple con las preocupaciones

Perspectiva de seguridad

Atributo de Calidad

- Sobre quien hace acciones sobre recursos
  - Controlar
  - Monitorear
  - Auditar
- En caso de fallas en los mecanismos de seguridad
  - Detectar
  - Recuperarse

Aplicable sobre

- Cualquier sistema
  - Con Interfaces publicas
    - Con múltiples usuarios cuya identidad es relevante
    - Donde el acceso a operaciones o a la información tiene que ser controlado

Preocupaciones

- Políticas
- Amenazas
- Mecanismos
- Rendición de cuentas (Accountability)
- Disponibilidad
- Detección y recuperación

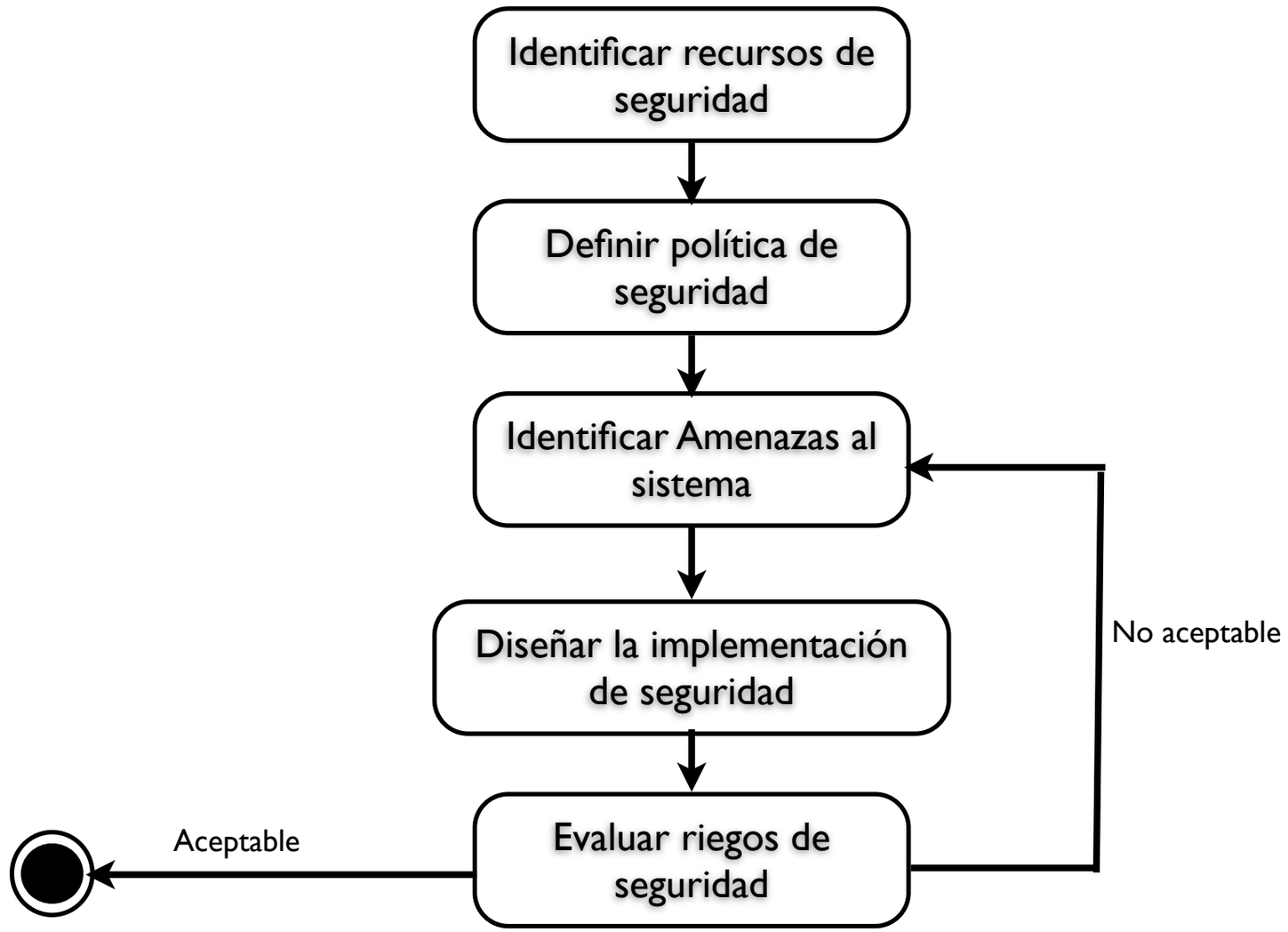
Problemas comunes y errores al crear la arquitectura

- Política de seguridad complejas
- Tecnologías de seguridad no probadas
- Sistema no diseñado para fallar
- Aproximación apegada a una tecnología
- Dependencia exagerada en tecnología
- Seguridad un ciudadano de segundo nivel
- Seguridad ad-hoc

Impacto sobre las vistas

Vista	Impacto
Funcional	Estructura funcional modificada por seguridad
Información	Modificada por seguridad, e.g., partición datos por sensibilidad
Concurrencia	Modificada por seguridad, e.g., confinamiento y separación
Desarrollo	Definir políticas de código, mejores prácticas, revisión de código
Despliegue	Impacto importante, Hardware especializado, Software especializado
Operacional	Alto impacto, vital importancia, como es operado el software, que prácticas se siguen, segregación de funciones, etc.

Actividades recomendadas



Identificar recursos de seguridad

Resource	Sensitivity	Owner	Access Control
Customer account records	Personal information of value for identity theft or invasion of privacy	Customer Care Group	No direct data access
Descriptive product catalog entries	Defines what is for sale and its description; if changed maliciously, could harm the business	Stock Management Group	No direct data access
Pricing product catalog entries	Defines pricing for catalog items; if maliciously or accidentally modified, could harm business or allow fraud	Pricing Team in Stock Management Group	No direct data access
Business operations on customer account records	Needs to be controlled to protect data access and integrity	Customer Care Group	Access to individual record or all records by authenticated principal
Descriptive catalog operations	Needs to be controlled to protect data access and integrity	Stock Management Group	Access to catalog modification operations by authenticated principal
Pricing catalog modification operations	Needs to be controlled to protect data access and integrity	Pricing Team	Access to price modification operations by authenticated principal, with accountability of changes
. . .	. . .	. . .	. . .

Ejemplo: extraído de "Software System Architecture" por Rozanski y Woods.

Definir política de seguridad

- Agrupar principals
- Agrupar Objetos
- Asignar acciones a grupos
- Identificar acciones sensibles
- Req. de integridad
- Políticas de claves
- Notación: texto y tablas

	User Account Records	Desc. Catalog Records	Pricing Records	User Account Operations	Desc. Catalog Operations	Price Change Operations
Data administrator	Full with audit	Full with audit	Full with audit	All with audit	All with audit	All with audit
Catalog clerk	None	None	None	All	Read-only operations	None
Catalog manager	None	None	None	Read-only operations with audit	All	All with audit
Product price administrator	None	None	None	None	Read-only operations	All
Customer Care clerk	None	None	None	All	Read-only operations	None
Registered customer	None	None	None	All on own record	Read-only operations	None
Unknown Web-site user	None	None	None	None	Read-only operations	None

Ejemplo: extraído de "Software System Architecture" por Rozanski y Woods.

Identificar amenazas

- Responde a las preguntas
  - Quien puede atacar?
  - Como puede atacar?
    - Cuales son las características del atacante?
    - Que consecuencias tiene el ataque?
- La salida se llama el modelo de amenazas
- Notación: Texto, gráficos y tablas

Diseñar la implementación de seguridad

- Diseñar mecanismos para mitigar amenazas
- Diseñar estrategia de detección y recuperación
- Evaluar la tecnología
- Integrar la tecnología a la estructura del sistema
- Notación: La correspondiente a cada vista, y UML en un documento dedicado a la arq. de seguridad (Opcional)
- **Aplicar principios de diseño de sistemas de seguridad**
- **Seleccionar la mejor herramienta para el trabajo**

Evaluar riesgos de seguridad

- Evaluar riesgos probabilidad de ocurrencia y costos.
- Decidir si el riesgo es aceptable

Risk	Estimated Cost	Estimated Likelihood	Notional Cost
Attacker gains direct database access.	$8,000,000	0.2%	$16,000
Web-site flaw allows free orders to be placed and fulfilled.	$800,000	4%	$32,000
Social-engineering attack on a customer service representative results in hijacking of customer accounts.	$4,000,000	1.5%	$60,000
. . .	. . .	. . .	. . .

Ejemplo: extraído de “Software System Architecture” por Rozanski y Woods.

Tácticas de arquitectura

Aplique principios reconocidos de seguridad

- Principio del menor privilegio
- Asegure el enlace más débil
- Defienda en profundidad
- Separe y confine
- Mantengalo simple
- Minimice los secretos
- Opciones por defecto seguras
- Falle de manera segura
- Asuma que los externos no son confiables
- Auditoría sobre eventos sensibles

Autentique principals

- Usuario y password
- Certificados digitales
- Tecnologías de hardware basada en token
- Sistemas Single-Sign-On (e.g., siteminder)

Autorice Acceso

- IBM Policy Director
- Entrust GetAccess
- JEE seguridad declarativa

Asegure secreto de la información

- Use cifrado
- Con recursos limitados use cifrado solo cuando sea necesario
- SSL / TLS
- Certificados digitales

Asegure Integridad de la información

- Recuerde cifrado + firma
- Garantice origen de mensajes
- Autentique con otro mecanismo

Garantice la rendición de cuentas

- Auditoría de mensajes
- No repudiación de mensajes
- Implemente procesos de revisión independiente

Proteja disponibilidad

- Considere ataques de negación del servicio (DoS)
- Reiniciar no es suficiente
- Revise mensajes desde un mismo origen

Otras tácticas

- Integre tecnologías de seguridad
- Provea Administración de seguridad
- Utilice infraestructura de seguridad de terceros (e.g., Application Servers)

CASO DE ESTUDIO: Estilo de Arquitectura PACE

- Basado en: 2004. PACE: An Architectural Style for Trust Management in Decentralized Applications. In Proceedings of the Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA '04). IEEE Computer Society, Washington, DC, USA, 221-.

Objetivo

- Definir un estilo arquitectural para la administración de confianza, en aplicaciones descentralizadas

Contexto

- **Una arquitectura descentralizada** es una colección de entidades, llamadas pares, que interactuan sin la presencia de una entidad central de confianza.
- **Relación de confianza**
  - Confianza basada en verificación de credenciales
  - Confianza basada en reputación

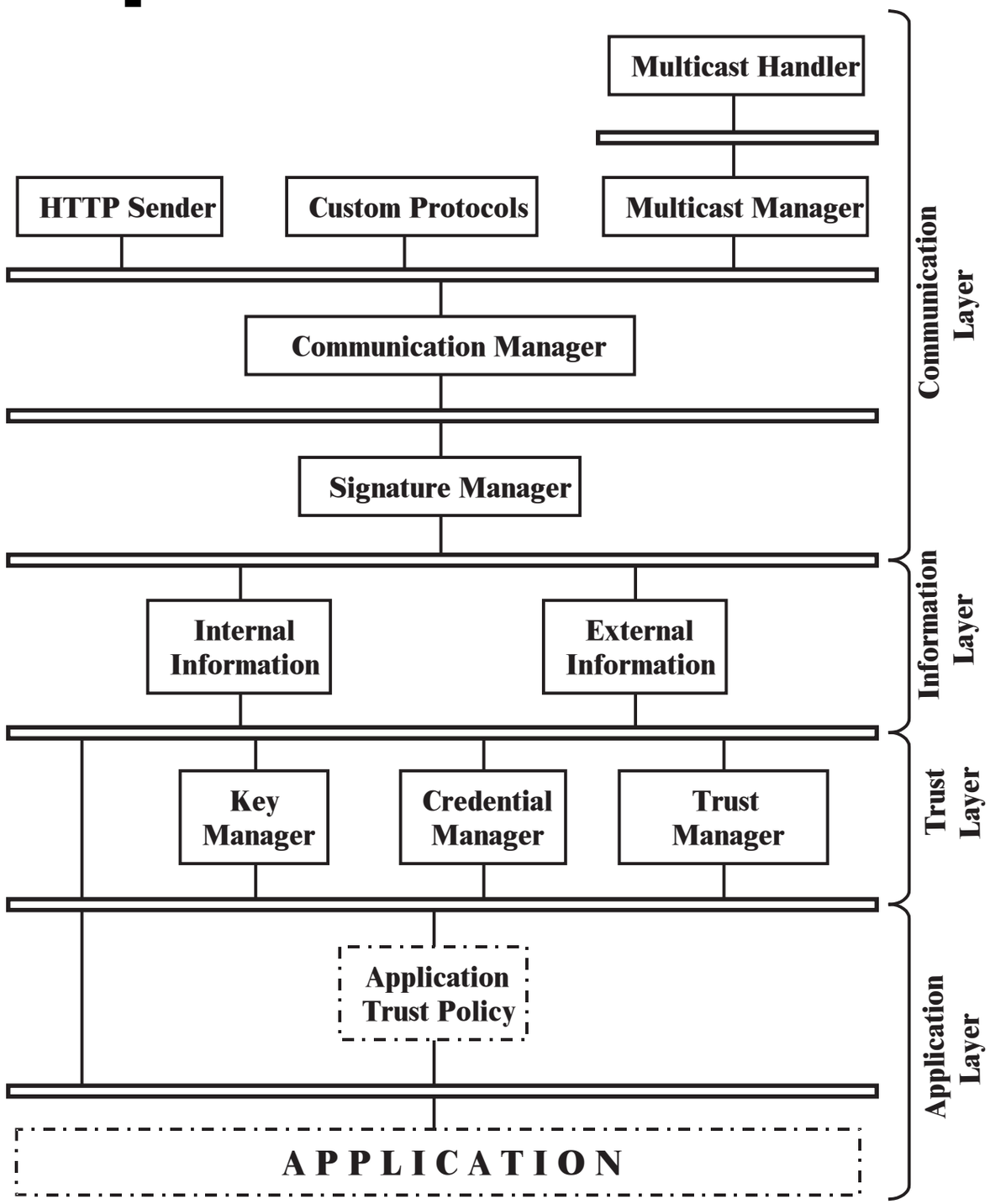
Identificación de Amenazas

- Personificación
- Acciones fraudulentas
- Representaciones falsas
- Coalición del mal
- Negación del servicio
- Presencia de desconocidos
- Decidir en quien confiar
- Conociemiento por fuera del sistema

Modelo de solución

- Estilo arquitectural basado en eventos
- Identidades digitales vs. Identidades físicas
- Separación de datos externos e internos
- Nivel de confianza hecho explícito
- Nivel de confianza comparable (e.g. representado por un número)
- Utilización de capas
- Confianza implicita de los componentes internos

Implementación



Críticas a la arquitectura

- Basada en la información del paper
  - Para verificar identidades usa public-key para firmar los mensajes, lo cual implica la presencia de certificados
  - Sin embargo, el manejador de llaves es el que genera las llaves dinámicamente lo cual impide que la verificación de terceros en confianza
  - Esto claramente es un error de seguridad pues yo puedo firmar los mensajes con mis propios certificados

Críticas a la arquitectura I I

- Falta información concreta de componentes
- No especifica las tácticas usadas
- Es ambigua en la tecnologías sugeridas
- Para los stakeholders es difícil entender la arquitectura
- Parece tener errores de manejo de seguridad en los esquemas de firma de mensajes