Conceptos de seguridad

Docente : Héctor Xavier Limón Riaño
email: xavier120@hotmail.com

Antes de empezar...

- En términos generales, ¿Qué es la seguridad?
- ¿Qué es la ciberseguridad?
- ¿De qué lado estás, equipo rojo o azul?
- ¿Desde qué punto de vista has visto a la seguridad hasta ahora?
- ¿Qué relación crees que tenga la programación y la seguridad?

Actividad

- Lee el siguiente post: `https://blog.sucuri.net/2018/10/owasp-top-10-security-risks-part-i.html`
- ¿Qué son los ataques de inyección?
- ¿Qué son los problemas de autenticación rota?
- ¿Qué tienen en común estos dos tipos de ataque y muchos que se encuentran en la lista?
- ¿Cómo se pueden prevenir?

Introducción

Tendencias de seguridad

- Una preocupación latente de las organizaciones y empresas modernas es cómo resistir a ataques y continuar operando, así como salvaguardar sus datos privados
- Los ataques modernos pueden ocurrir por diversas motivaciones y ser perpetrados por personas desde aquellas pertenecientes grupos de crimen organizado que buscan monetizar brechas de seguridad, hasta hacktivistas que buscan castigar a las organizaciones que ellos tachan de inmorales o contrarias a la sociedad pública

Tendencias de seguridad

- Aunque parezca sacado de una novela cyberpunk, con la tendencia creciente hacía el Internet de las cosas, es cuestión de tiempo para vulnerabilidades de seguridad representen una amenaza verdadera y directa para la vida e integridad de seres humanos
- Se abre un escenario para nuevas formas de ciberterrorismo
- Se tiene necesidad de personas preparadas que jueguen del lado defensivo, del equipo azul, pero...

Tendencias de seguridad

- El lado azul es el menos popular y los medios lo suelen poner como los perdedores siempre
- La industria de la seguridad en la información se encuentra actualmente pasando por un periodo de desempleo negativo; esto es, hay más puestos abiertos para cubrir que candidatos que puedan cubrirlo
- Contratar personas es de por si difícil y contratar buenas personas es todavía más difícil

Seguridad y sus elementos

- La seguridad involucra una gran cantidad de elementos y cosas a considerar
- Desde la infraestructura física de red hasta la programación misma de los servicios
- Un plan de seguridad completo involucra los siguientes aspectos generales:
  - Sentar bases
  - Establecer equipos
  - Establecer una postura base de seguridad
  - Evaluación de amenazas y riesgos
  - Priorización
  - Creación de hitos
  - Realizar simulacros y ejercicios basados en casos
  - Expansión y mejora continua de los equipos de seguridad

¿Dónde me ubico yo?

- Un licenciado en Redes y Servicios de Cómputo debe conocer todos los aspectos de seguridad
- Aunque el enfoque no sea desarrollo de software, debe ser capaz de identificar problemas de seguridad en el código y estar involucrado en los procesos de mejora de seguridad, incluyendo aquellos referentes a auditoría de código
- Los problemas de seguridad pueden afectar a toda la operación de la organización, la labor de alguien de su perfil es mantener la disponibilidad de los servicios y proteger la información crítica

El problema de la seguridad de software

- Se tiene la creencia de que la forma más efectiva de mejorar la seguridad del software es estudiando errores pasados de seguridad y prevenir que vuelvan a pasar en el futuro
- ¿Viendo la lista más actual de vulnerabilidades de OWASP crees que se esté aprendiendo de los errores?
- ¿Dónde está el problema entonces?

El problema de la seguridad de software

- Se tiene la falsa creencia de que la seguridad es un aspecto que únicamente le debería preocupar a los operativos
- La realidad es que debería preocuparle a todos

El problema de la seguridad de software

- Un punto central del problema es la forma en que se desarrolla software
- Tradicionalmente el enfoque se ha centrado en la calidad desde el punto de vista de defectos en la funcionalidad
- No se le suele dar mayor peso a la seguridad
- Pero un sistema que funciona bien no es necesariamente seguro

El problema de la seguridad de software

- Los usos inesperados de un programa vulnerable pueden verse como características extra no intencionadas, formas de uso para los cuales el programa no fue diseñado, pero que permite
- **Un programa seguro es aquel que cumple con la funcionalidad esperada (requerimientos) y nada más**

El problema de la seguridad de software

- El desarrollador muchas veces no se preocupa de los usuarios que utilizarán el sistema, prevé que éstos puedan cometer errores e ingenuamente no considera usuarios directamente malintencionados
- En general, cuando se habla de seguridad, siempre se debe considerar la existencia de un rival, alguien que intencionadamente y con posibles habilidades educadas intenta vulnerar el sistema
- Una mentalidad a erradicar es: "he hecho varios sistemas antes sin considerar seguridad y nunca me han atacado, ¿Porqué debería preocuparme?"

Características de seguridad ! = Características seguras

- A veces los programadores piensan en la seguridad, pero muy a menudo, lo piensan únicamente en el sentido de características de seguridad
- Tales como: cifrado de información, políticas de passwords, mecanismos de control de acceso, etc.
- Para que un programa sea seguro, todas sus porciones deben ser seguras, no sólo los plácitos que explícitamente hacen referencia a seguridad

Características de seguridad ! = Características seguras

- Una característica de seguridad puede fallar y poner en entre dicho toda la seguridad del sistema en muchas formas, pero...
- Usualmente hay muchas más formas de comprometer la seguridad a través de características defectuosas que no tienen nada que ver con seguridad, por ejemplo mal manejo de memoria
- Un mal manejo de memoria puede abrir puertas para ejecutar código malicioso

Características de seguridad ! = Características seguras

- Las características de seguridad se implementan con la idea de que deben funcionar correctamente para mantener la seguridad del sistema
- Las características que directamente no tienen que ver con seguridad directamente usualemnte no reciben atención, aunque sean igual de críticas
- Los programadores a menudo comenten este error y dejan de pensar en la seguridad cuando en realidad deberían enfocarse en ella

Tarea

- Realiza una investigación sobre DevSecOps de una cuartilla, en dicha investigación debes cubrir los siguiente puntos:
  - ¿Qué es DevSecOps?
  - ¿Cuál es su importancia?
  - ¿Qué prácticas incluye?
  - ¿Cómo se relaciona con la experiencia educativa?

Vulnerabilidades

Introducción

- Defectos de seguridad que pueden ser explotados por un atacante
- Se pueden detectar mediante ejercicios como la auditoría de código, pentesting y herramientas automatizadas
- Se pueden prevenir siguiendo buenas prácticas y conciencia de seguridad integral

Clasificación básica

Posible clasificación de alto nivel

- Defectos genéricos
- Defectos de contexto

Defectos genéricos

- Problema que puede ocurrir en prácticamente cualquier programa escrito en un lenguaje
- Pueden ser problemas específicos del lenguaje, por ejemplo, C es más propenso a vulnerabilidades de memoria que lleve a un ataque de buffer overflow
- Las herramientas de análisis estático de código son buenas para detectar este tipo de problemas

Defectos de contexto

- Requiere conocimiento semántico del problema que intenta resolver el programa
- Por ejemplo, en un programa que maneja tarjetas de crédito debe cuidarse que nunca sea visible el número completo de tarjeta
- Cabe decir que defectos a este nivel pueden no ser sólo de codificación, sino también de diseño, por ejemplo tener un módulo que descarga archivos adjuntos de correo y ejecuta el código que contienen
- Este tipo de defectos requieren de configuración en una herramienta de análisis estático de código

Clasificación extendida

- Validación de entrada y representación: inyecciones
- Abuso de API: no seguir contrato
- Características de seguridad: información sensible visible
- Manejo de errores: dejar errores visibles
- Calidad de código: código fácilmente estrésable
- Encapsulación: mal manejo de privacidad

Ejercicio

- Revisa el sistema en la siguiente sitio: `http://quieroaprender.mx:8000/`
- Determina 3 posibles vulnerabilidades genéricas y su tipo (de la clasificación extendida)
- Determina una posible vulnerabilidad de contexto y su tipo
- No tienen que ser vulnerabilidades que realmente tenga el sistema, sólo identifica posibilidades
- Propón una forma de mitigar las vulnerabilidades

Modelado de amenazas

Introducción

- Una amenaza es el objetivo de un atacante, para otros autores una amenaza es el atacante mismo
- La idea del modelado es entender posibles amenazas de seguridad del sistema, determinar riesgos y establecer medidas de mitigación apropiadas
- Permite elevar la conciencia de la organización hacia la seguridad

Introducción

- El modelado de amenazas se encuentra en la etapa de diseño de software
- Esto quiere decir que debe realizarse antes que revisiones de código o pruebas de seguridad
- Es probablemente la actividad de desarrollo más importante de seguridad ya que encamina todo el desarrollo
- Los defectos potenciales se localizan temprano y pueden resolverse rápido de forma barata

Introducción

- En este curso se verá el método propuesto por el ciclo de vida de desarrollo seguro (SDL) de Microsoft
- Se utiliza como base la clasificación de riesgos STRIDE explicada más adelante
- También se utiliza como apoyo diagramas de diseño como diagramas de flujo de datos (DFD) y diagramas de contexto

Beneficios

- Contribuye al proceso de manejo de riesgos porque las amenazas al software e infraestructura son riesgos para el usuario y el ambiente donde se despliega el software
- Permite descubrir amenazas antes de que el sistema empiece a codificarse
- Revalida el diseño y arquitectura al hacer que el equipo de desarrollo revise de nuevo el diseño
- Contribuye a reducir la superficie de ataque
- Ayuda a guiar el proceso de revisión de código
- Guía el proceso de pentesting
- Clarifica la selección de medidas de mitigación

Artefactos de modelado de amenazas

- La salida principal de un proceso de modelado de amenazas es un documento que describe la información de trasfondo de la aplicación
- El documento define el modelo de alto nivel de la aplicación, usando diagramas de flujo de datos

Artefactos de modelado de amenazas

- Lo más importante a identificar en dichos diagramas es:
  - Escenarios de uso: usos generales del usuario, configuración de despliegue
  - Dependencias externas: productos, componentes o servicios de los que depende el sistema
  - Datos intercambiados entre elementos
  - Límites de privilegios: establecen cambios en los privilegios ya sea hacía arriba o hacía abajo

Etapas generales del modelado de amenazas

- Preparar: los diseñadores de sistema modelan mediante DFDs el flujo de datos
- Analizar: se analizan los modelos para localizar posibles amenazas
- Determinar medidas de mitigación: se determina para cada amenaza una forma de mitigarla con sus posibles técnicas y tecnologías asociadas
- Implementar medidas de mitigación

Etapas generales del modelado de amenazas

- Al seguir un procedimiento de modelado de amenazas desde la etapa de diseño, se está favoreciendo una aproximación pro-activa de seguridad
- Esto en contraposición a una aproximación reactiva, donde se reacciona hasta que hay un problema
- Como dice el dicho: prevenir es mejor que lamentar

Preparar

- Es la etapa de mayor importancia, si se realiza mal el resto del modelado estará mal
- En este curso se utilizaran diagramas de flujos de datos (DFDs) que muestran la comunicación entre procesos del sistema y entidades externas como usuarios u otros sistemas
- Los DFDs pueden realizarse por niveles, al nivel más general (nivel 0) se le conoce como diagrama de contexto
- Se pueden realizar tantos niveles como haga falta dependiendo de la complejidad del sistema
- En este curso sólo se trataran diagramas de nivel 0 y nivel 1
- Ver material: What_are_Data_Flow_Diagrams.doc

Diagrama de contexto

- El sistema se identifica como un único proceso
- Se identifican las entidades externas y su interacción general con el sistema a través de flujos de datos
- Se identifican límites de privilegios
- Es importante numerar cada componente
- Cada flujo se etiqueta con un verbo o bien un sustantivo, es usual encontrar verbos CRUD (create, read, update, delete) pero se debe preferir aquella etiqueta que describa mejor los datos que se transfieren

Diagrama de contexto

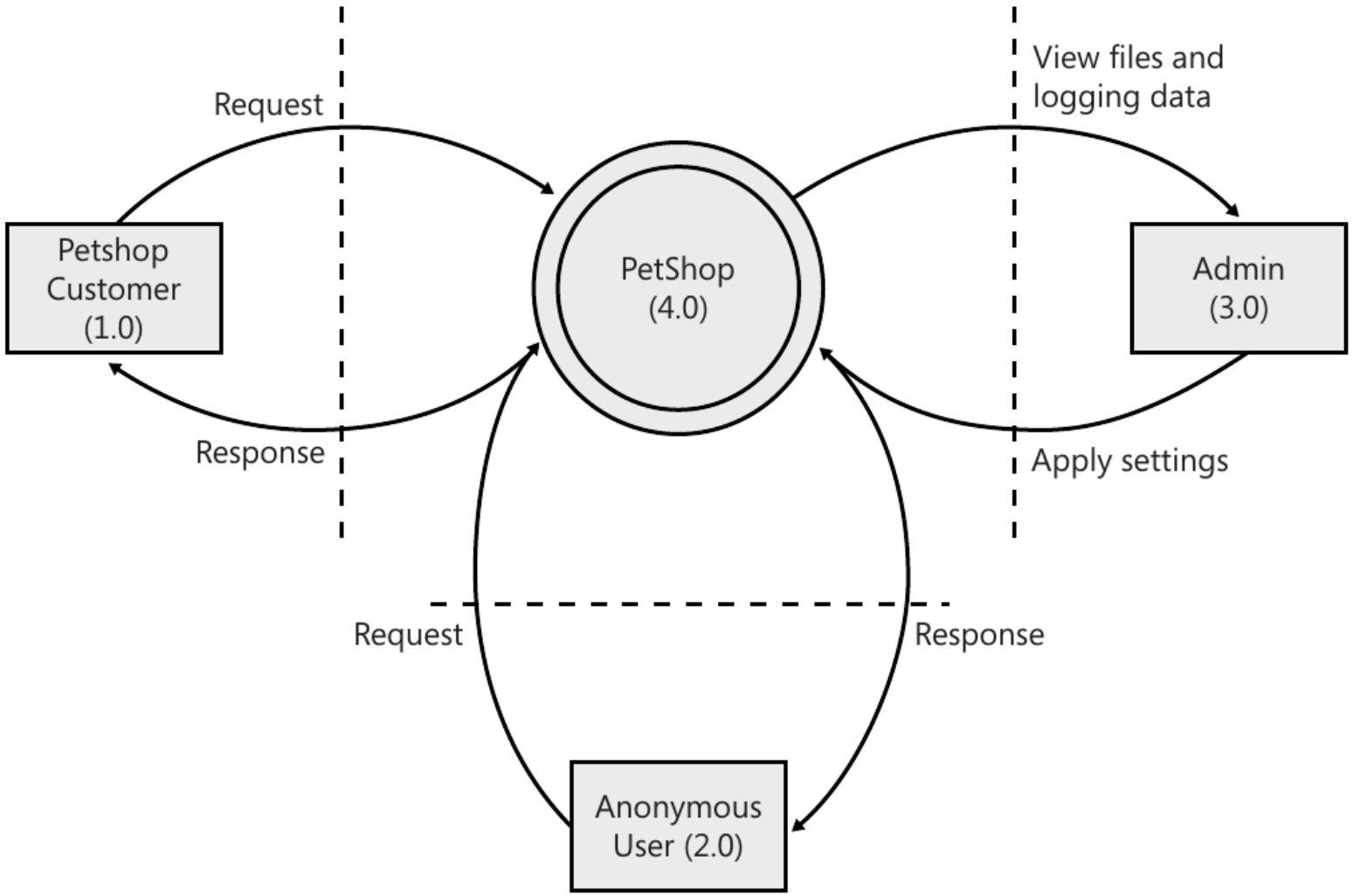


Diagrama de nivel 1

- El sistema general se descompone en subsistemas
- A este nivel se introducen almacenes con los que los procesos interactúan
- Es posible que sigan apareciendo procesos complejos, para cada proceso complejo se debe bajar de nuevo de nivel y hacer un DFD correspondiente
- En el curso nos quedaremos hasta el nivel 1

Diagrama de nivel 1

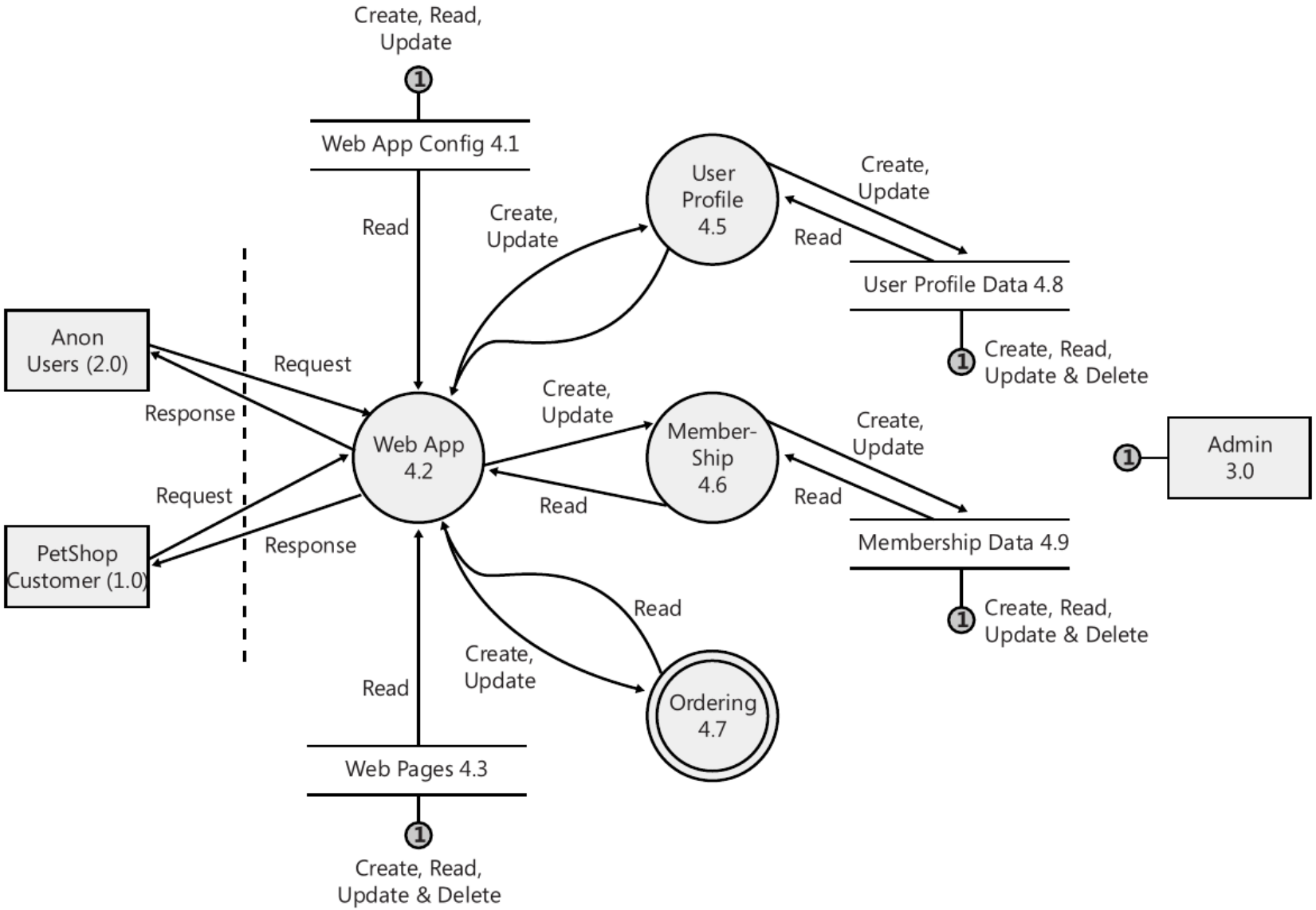
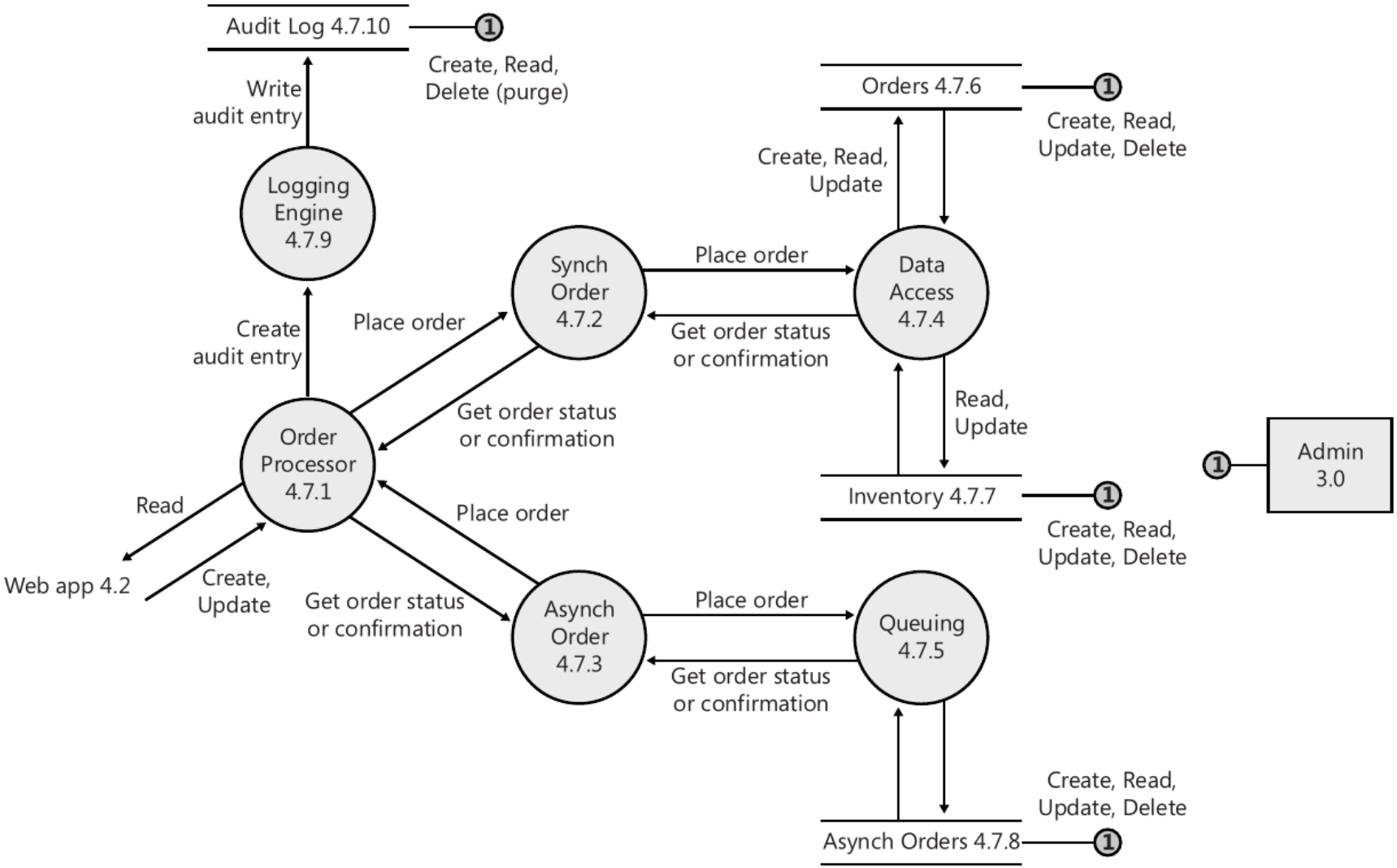


Diagrama de nivel 2



Actividad

- Para el sistema de evaluación de código determina el DFD de nivel 0 y de nivel 1

Determinar tipos de amenazas

Tradicional CIA:

- Confidencialidad
- Integridad
- Disponibilidad

Confidencialidad

- Las organizaciones trabajan a partir de un conjunto de secretos: datos de nómina, información médica, contraseñas, secretos industriales, etc.
- El accesos a esos secretos debe estar estrictamente restringido, de lo contrario la información filtrada puede causar graves repercusiones
- La confidencialidad es mantener esos secretos a salvo, sólo accesibles por las personas o sistemas autorizados
- Una forma de mantener la confidencialidad es por ejemplo el cifrado

Integridad

- Tiene que ver con mantener un estado esperado de la información, esto es, que la información no sea alterada o destruida
- Las alteraciones pueden suceder a través de transmisiones o bien directamente en disco
- La integridad puede ser comprometida no sólo por atacantes, sino también por fallos de hardware o software
- Una forma de asegurarse de que la información se mantiene integra es mediante algoritmos de hash

Disponibilidad

- Se refiere a la capacidad que tiene el sistemas y sus servicios para mantener un estado de operación y así brindar servicio a clientes
- Las alteraciones en la disponibilidad pueden dar de baja el servicio o bien mermar su tiempo de respuesta
- Es de suma importancia ya que muchas organizaciones dependen del mantenimiento continuo en la operación
- La forma más común de ataque a la disponibilidad es un ataque DOS o DDOS
- Este es del tipo de ataques que es más difícil defenderse

STRIDE

- Otra clasificación más completa que CIA y que considera problemas actuales de seguridad
- Tiene los siguientes tipos:
  - Spoofing Identity
  - Tampering
  - Repudiation
  - Information Disclosure
  - Denial of Service
  - Elevation of Privilege

Spoofing Identity

- Las amenazas de este tipo le permiten al atacante hacerse pasar por algo o alguien más
- Como un servidor haciéndose pasar por Microsoft.com o código que se hace pasar por una biblioteca de sistema

Tampering

- Involucran modificaciones maliciosas de datos o código
- Los datos o código manipulado puede estar en un medio volátil o no volátil o incluso en la red

Repudiation

- Un atacante realiza una amenaza de repudio al negar haber efectuado una acción que otras personas no pueden confirmar o contradecir
- Por ejemplo, un usuario realiza repudio cuando efectúa una operación ilegal en un sistema que no puede rastrear dicha operación prohibida
- El no repudio es la habilidad del sistema para tratar amenazas de repudio
- Para lograr no repudio se debe tener evidencia mediante logging y registros de eventos

Information Disclosure

- Involucra la exposición de información a individuos que se supone no deberían tener acceso a ella
- Incluye tanto el acceso a archivos como a datos que viajan a través de la red
- Una amenaza de este tipo si no se trata puede convertirse en una violación de privacidad si los datos en cuestión son confidenciales o personales

Denial of Service

- Son ataques que deniegan o degradan el acceso a un servicio por parte de sus usuarios
- Es necesario protegerse de esta amenaza para mejorar la disponibilidad y confiabilidad del sistema

Elevation of Privilege

- Esta amenaza ocurre cuando un usuario o proceso gana un incremento en sus capacidades de permisos
- A menudo se da a través de un usuario anónimo que explota un bug de programación para tener acceso a privilegios de administrador

Identificar amenazas al sistema

- Una vez tienes tus DFDs se necesitan listar sus elementos ya que necesitan ser protegidos contra ataques
- En la lista no deben incluirse procesos complejos sino los elementos internos que lo conforman

Identificar amenazas al sistema

Table 9-3 **DFD Elements Within the Pet Shop 4.0 Application**

DFD Element Type	DFD Item Numbers
External Entities	Pet Shop customer (1.0)
	Anonymous user (2.0)
	Administrator (3.0)
Processes	Web application (4.2)
	User profile (4.5)
	Membership (4.6)
	Order processor (4.7.1)
	Synchronous order processor (4.7.2)
	Asynchronous order processor (4.7.3)
	Data access component (4.7.4)
	Queuing component (4.7.5)
	Auditing engine (4.7.9)
Data Stores	Web application configuration data (4.1)
	Web pages (4.3)
	User profile data (4.8)
	Membership data (4.9)
	Orders data (4.7.6)
	Inventory data (4.7.7)
	Asynch orders data (4.7.8)
	Audit-log data (4.7.10)

Identificar amenazas al sistema

Table 9-3 DFD Elements Within the Pet Shop 4.0 Application

DFD Element Type	DFD Item Numbers
Data Flows (partial list for brevity)	Anonymous user request (2.0→4.2)
	Anonymous user response (4.2→2.0)
	Pet Shop customer request (1.0→4.2)
	Pet Shop customer response (4.2→1.0)
	Web application reading configuration data (4.1→4.2)
	Web pages read by Web application (4.3→4.2)
	Admin creating or updating Web application configuration data (3.0→4.1)
	Admin reading Web application configuration data (4.1→3.0)
	Admin creating, updating, or deleting Web pages (3.0→4.3)
	Admin reading Web pages (4.3→3.0)
	Web application creating or updating an order (4.2→4.7.1)
	Web application reading an order (4.7.1→4.2)

Identificar amenazas al sistema

- De la lista anterior se puede aplicar un proceso de reducción
- Si tienes dos o más elementos del mismo tipo (por ejemplo dos o más procesos) que se encuentran detrás del mismo límite de confianza se puede modelar el elemento como una misma entidad, siempre y cuando los elementos usan la misma tecnología y manejan datos similares
- Esta reducción puede quitar muchos flujos, en la mayoría de casos es posible reducir todos los flujos bidireccionales

Identificar amenazas al sistema

Table 9-4 **Reduced DFD Elements Within Pet Shop 4.0**

DFD Element Type	DFD Item Number
External Entities	Pet Shop customer (1.0)
	Anonymous user (2.0)
	Administrator (3.0)
Processes	Web application (4.2)
	User profile (4.5)
	Membership (4.6)
	Order processor (4.7.1)
	Sync/Async order processors (4.7.2 and 4.7.3)
	Data-access or queuing components (4.7.4 and 4.7.5)
	Auditing engine (4.7.9)
Data Stores	Web application configuration data (4.1)
	Web pages (4.3)
	Use profile data (4.8)
	Membership data (4.9)
	Order and async orders data (4.7.6 and 4.7.8)
	Inventory data (4.7.7)
	Audit-log data (4.7.10)
Data Flows (partial list)	Web application reading configuration data (4.1→4.2)
	Web pages read by Web application (4.3→4.2)
	Anonymous user request/response (2.0→4.2→2.0)
	Pet Shop customer request/response (1.0→4.2→1.0)
	Admin reading, creating, updating Web application configuration data (3.0→4.1→3.0)
	Admin reading, creating, updating, deleting Web pages (3.0→4.3→3.0)
	Web application reading, creating, updating an order (4.2→4.7.1→4.2)

Identificar amenazas al sistema

- Una vez se tiene la lista anterior se puede aplicar STRIDE a cada elemento, a partir del siguiente mapeo
- Esencialmente todo lo que está en el DFD es propenso a ser atacado y la naturaleza del ataque potencial es determinado por el tipo de elemento

Table 9-5 Mapping STRIDE to DFD Element Types

DFD Element Type	S	T	R	I	D	E
External Entity	X		X			
Data Flow		X		X	X	
Data Store		X	†	X	X	
Process	X	X	X	X	X	X

Identificar amenazas al sistema

- El siguiente paso es asociar las amenazas con los elementos identificados

Table 9-7 Threats to the System

Threat Type (STRIDE)	DFD Item Numbers
Spoofing	External entities: (1.0), (2.0), (3.0)
	Processes: (4.2), (4.5), (4.6), (4.7.1), (4.7.2 and 4.7.3), (4.7.4 and 4.7.5), (4.7.9)
Tampering	Processes: (4.2), (4.5), (4.6), (4.7.1), (4.7.2 and 4.7.3), (4.7.4 and 4.7.5), (4.7.9)
	Data stores: (4.1), (4.3), (4.8), (4.9), (4.7.6 and 4.7.8), (4.7.7), (4.7.10)
	Data flows: (4.1→4.2), (4.3→4.2), (2.0→4.2→2.0), (1.0→4.2→1.0), (3.0→4.1→3.0), (3.0→4.3→3.0), (4.2→4.7.1→4.2)
Repudiation	External entities: (1.0), (2.0), (3.0)
	Data flow: (4.7.10)
Information disclosure	Processes: (4.2), (4.5), (4.6), (4.7.1), (4.7.2 and 4.7.3), (4.7.4 and 4.7.5), (4.7.9)
	Data stores: (4.1), (4.3), (4.8), (4.9), (4.7.6 and 4.7.8), (4.7.7), (4.7.10)
	Data flows: (4.1→4.2), (4.3→4.2), (2.0→4.2→2.0), (1.0→4.2→1.0), (3.0→4.1→3.0), (3.0→4.3→3.0), (4.2→4.7.1→4.2)

Identificar amenazas al sistema

Table 9-7 Threats to the System

Threat Type (STRIDE)	DFD Item Numbers
DoS	Processes: (4.2), (4.5), (4.6), (4.7.1), (4.7.2 and 4.7.3), (4.7.4 and 4.7.5), (4.7.9)
	Data stores: (4.1), (4.3), (4.8), (4.9), (4.7.6 and 4.7.8), (4.7.7), (4.7.10)
	Data flows: (4.1→4.2), (4.3→4.2), (2.0→4.2→2.0), (1.0→4.2→1.0), (3.0→4.1→3.0), (3.0→4.3→3.0), (4.2→4.7.1→4.2)
EoP	Processes: (4.2), (4.5), (4.6), (4.7.1), (4.7.2 and 4.7.3), (4.7.4 and 4.7.5), (4.7.9)

Plan de mitigación

Estrategias de mitigación:

- No hagas nada: OK para amenazas de bajo riesgo
- Quita la característica: reduce riesgo a 0, balance entre funcionalidad y seguridad
- Deshabilita la característica: menos drástico que remover, puede aplicarse por rol de usuario
- Advierte al usuario: el usuario puede no entender los riesgos
- Mitiga la amenaza mediante tecnología: medida más común, por ejemplo usar autenticación o cifrado

Técnicas de mitigación

Table 9-8 Mitigation Techniques Based on STRIDE Threat Type

Threat Type	Mitigation Technique
Spoofing	Authentication
Tampering	Integrity
Repudiation	Non-repudiation services
Information disclosure	Confidentiality
DoS	Availability
EoP	Authorization

Ejemplos de tecnologías por técnica

Table 9-9 Mitigation Technologies

Mitigation Technique	Mitigation Technology
Authentication	Authenticate principals: ■ Basic authentication ■ Digest authentication ■ Cookie authentication ■ Windows authentication (NTLM) ■ Kerberos authentication ■ PKI systems such as SSL/TLS and certificates ■ IPSec ■ Digitally signed packets Authenticate code or data: ■ Digital signatures ■ Message authentication codes ■ Hashes

Ejemplos de tecnologías por técnica

Integrity	■ Windows Vista Mandatory Integrity Controls ■ ACLs ■ Digital signatures ■ Message authentication codes
Non-repudiation services	■ Strong authentication ■ Secure auditing and logging ■ Digital signatures ■ Secure time-stamps ■ Trusted third parties
Confidentiality	■ ACLs ■ Encryption
Availability	■ ACLs ■ Filtering ■ Quota ■ Authorization
Authorization	■ ACLs ■ Group or role membership ■ Privilege ownership ■ Permissions

Ejemplos de tecnologías por técnica

- El siguiente paso es tomar todas las amenazas del modelado, observar las técnicas de mitigación asociadas y determinar una tecnología de mitigación apropiada

Table 9-11 Defenses Used in Portions of Pet Shop 4.0

Asset	Asset Type	Example Threat	Example Mitigation
(1.0→4.2→1.0)	Data flow from Pet Shop user to Web application and back	I	SSL/TLS (also mitigates the tampering threat)
4.7.10	Audit log data store	T	ACL and MAC
4.7.1	Order processor process	T and E	ACL, MAC, and reduced process privilege

Ejercicio

- Toma como punto de partida el DFD de nivel 1 realizado por el profesor para el ejemplo del sistema de evaluación de código
- Identifica todos los elementos del DFD en una tabla, realiza las reducciones convenientes
- Asocia los elementos con los tipos de amenazas STRIDE
- Para cada elemento toma como base una posible amenaza STRIDE que aplique y propón una tecnología de mitigación apropiada