UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Departamento Ingeniería en Sistemas de Información

ASIGNATURA:	SEGURIDAD EN APLICACIONES WEB
DEPARTAMENTO:	ING. EN SIST. DE INFORMACION
AREA:	ELECTIVA
BLOQUE	TECNOLOGÍAS APLICADAS

MODALIDAD:	Cuatrimestral
HORAS SEM.:	6 horas
HORAS/AÑO:	96 horas
HORAS RELOJ	72
NIVEL:	5º
AÑO DE DICTADO:	2015

Objetivos

- Que el futuro profesional comprenda conceptualmente qué son las vulnerabilidades que afectan a las aplicaciones WEB, cuáles son las causas que las generan y cómo es posible explotarlas.
- Que el alumno entienda la evolución que ha tenido la tecnología en los últimos años y el contexto en el cual surgen estas vulnerabilidades.
- Que el estudiante comprenda los riesgos que traen aparejadas dichas vulnerabilidades en cada caso y sus alcances.
- Que el estudiante se concientice sobre el concepto de seguridad y que lo incorpore a sus tareas diarias.
- Que el futuro profesional desarrolle un criterio crítico sobre el desarrollo de aplicaciones WEB y su seguridad.
- Concientizar al alumno en las buenas prácticas de programación y brindarle herramientas que no estén basadas solamente en la correcta programación e implementación.
- Que el futuro profesional comprenda la necesidad de la investigación y actualización del actual estado del arte.
- Que el alumno conozca y maneje herramientas que le permitan mantener plataformas WEB con adecuados niveles de seguridad.
- Que el estudiante comprenda conceptualmente los distintos enfoques que tienen las soluciones existentes.
- Coordinar charlas de proveedores de soluciones.

Contenidos Mínimos (Programa Sintético).

- Desarrollo conceptual de que es una Vulnerabilidad en una Aplicación WEB

- Que elementos propios de una codificación de una solución WEB producen una Vulnerabilidad.
- Que elementos ajenos a la propia codificación de un desarrollo WEB producen una Vulnerabilidad
- Enumeración e introducción a las vulnerabilidades más reconocidas y el porqué de este reconocimiento.
- Explotación de las vulnerabilidades. Que se puede hacer u obtener en cada explotación. Límites y alcances.
- Que es un WAF (Firewall de Aplicaciones WEB).
- Como funciona y como se implementa.
- Que puede hacer para mitigar las Vulnerabilidades.
- Como se generan las reglas del WAF.
- Estado del arte y amenazas complejas

Contenido Analítico:

UNIDAD 1: Contexto
Estado del arte en la construcción de soluciones WEB. Avance y cambios en la tecnología en la última década. Tendencias. Desafíos presentes y futuros del desarrollador y su responsabilidad en la seguridad

UNIDAD 2: Nociones fundamentales
Conocimientos sobre vulnerabilidades y cómo explotarlas. Conocimientos sobre exploits. Riesgos de las vulnerabilidades, su impacto. Por qué una aplicación es vulnerable. Que produce una vulnerabilidad. Ejemplos de codificaciones que hacen a las aplicaciones vulnerables.

UNIDAD 3: Implementación e infraestructura.
Por qué una página WEB es estática. Página WEB dinámica, basada en CGI-BIN. Página WEB dinámica, basada en código interpretado (PHP, ASP). Página WEB dinámica con consulta a BD. Página WEB dinámica con un servidor de aplicaciones. Web server - tipos y cuál es su función. Servidor de Aplicaciones - tipos y cuál es su función. Base de datos - tipos y cuál es su función. Proxys reversos – tipos y cuál es su función. Terminador de túneles SSL – tipos y cuál es su función. Dónde y por qué. Infraestructura. Dónde y cómo se vinculan. DMZ. Servidores WEB. Servidores de Aplicaciones. Servidores de Base de Datos. Proxys. SSL.

UNIDAD 4: Firewall de Aplicaciones
Presentación de soluciones como Firewall de Aplicaciones:

- ¿Qué es?¿Cómo funciona?

- ¿Qué alcance tiene?
- ¿Qué limitantes tiene?¿Por qué?
- ¿Cómo se implementa?
- ¿Cómo se mantiene?

Firewall de aplicaciones (web y de base de datos).
Armado de los Laboratorios. WAF. Aplicaciones Web Vulnerables. Herramientas de investigación y explotación de vulnerabilidades.

UNIDAD 5: Vulnerabilidad de Inyección.
Las fallas de inyección, tales como SQL, OS, y LDAP, ocurren cuando datos no confiables son enviados a un intérprete como parte de un comando o consulta. Los datos hostiles del atacante pueden engañar al intérprete en ejecutar comandos no intencionados o acceder datos no autorizados.

UNIDAD 6: Vulnerabilidad de Pérdida de Autenticación y Gestión de Sesiones.
Las funciones de la aplicación relacionadas a autenticación y gestión de sesiones son frecuentemente implementadas incorrectamente, permitiendo a los atacantes comprometer contraseñas, claves, token de sesiones, o explotar otras fallas de implementación para asumir la identidad de otros usuarios.

UNIDAD 7: Vulnerabilidad de Secuencia de Comandos en Sitios Cruzados (XSS)
Las fallas XSS ocurren cada vez que una aplicación toma datos no confiables y los envía al navegador web sin una validación y codificación apropiada. XSS permite a los atacantes ejecutar secuencia de comandos en el navegador de la víctima, los cuales pueden secuestrar las sesiones de usuario, destruir sitios web, o dirigir al usuario hacia un sitio malicioso.

UNIDAD 8: Vulnerabilidad de Referencia Directa Insegura a Objetos
Una referencia directa a objetos ocurre cuando un desarrollador expone una referencia a un objeto de implementación interno, tal como un fichero, directorio, o base de datos. Sin un chequeo de control de acceso u otra protección, los atacantes pueden manipular estas referencias para acceder a datos no autorizados.

UNIDAD 9: Vulnerabilidad de Configuración de Seguridad Incorrecta
Una buena seguridad requiere tener definida e implementada una configuración segura para la aplicación, marcos de trabajo, servidor de aplicación, servidor web, base de datos, y plataforma. Todas estas configuraciones deben ser definidas, implementadas, y mantenidas ya que por lo general no son seguras por defecto. Esto incluye mantener todo el software actualizado, incluidas las librerías de código utilizadas por la aplicación.

UNIDAD 10: Vulnerabilidad de Exposición de datos sensibles
Muchas aplicaciones web no protegen adecuadamente datos sensibles tales como números de tarjetas de crédito o credenciales de autenticación. Los atacantes

pueden robar o modificar tales datos para llevar a cabo fraudes, robos de identidad u otros delitos. Los datos sensibles requieren de métodos de protección adicionales tales como el cifrado de datos, así como también de precauciones especiales en un intercambio de datos con el navegador.

UNIDAD 11: Vulnerabilidad de Ausencia de Control de Acceso a Funciones
La mayoría de aplicaciones web verifican los derechos de acceso a nivel de función antes de hacer visible la misma interfaz de usuario. A pesar de esto, las aplicaciones necesitan verificar el control de acceso en el servidor cuando se accede a cada función. Si las solicitudes de acceso no se verifican, los atacantes podrán realizar peticiones sin la autorización apropiada.

UNIDAD 12: Falsificación de Peticiones en Sitios Cruzados (CSRF)
Un ataque CSRF obliga al navegador de una víctima autenticada a enviar una petición HTTP falsificado, incluyendo la sesión del usuario y cualquier otra información de autenticación incluida automáticamente, a una aplicación web vulnerable. Esto permite al atacante forzar al navegador de la víctima para generar pedidos que la aplicación vulnerable piensa que son peticiones legítimas provenientes de la víctima.

UNIDAD 13: Vulnerabilidad de Utilización de componentes con vulnerabilidades conocidas
Algunos componentes tales como las librerías, los frameworks y otros módulos de software casi siempre funcionan con todos los privilegios. Si se ataca un componente vulnerable esto podría facilitar la intrusión en el servidor o una pérdida seria de datos. Las aplicaciones que utilicen componentes con vulnerabilidades conocidas debilitan las defensas de la aplicación y permiten ampliar el rango de posibles ataques e impactos.

UNIDAD 14: Vulnerabilidad de Redirecciones y reenvíos no validados
Las aplicaciones web frecuentemente redirigen y reenvían a los usuarios hacia otras páginas o sitios web, y utilizan datos no confiables para determinar la página de destino. Sin una validación apropiada, los atacantes pueden redirigir a las víctimas hacia sitios de phishing o malware, o utilizar reenvíos para acceder páginas no autorizadas.

Bibliografía.

Bibliografía Obligatoria:

- Modsecurity Handbook (2010) Escrito por Ivan Ristic
- Web Application Security, A Beginner's Guide (2011) Escrito por Bryan Sullivan,Vincent Liu
- HACKING EXPOSED WEB APPLICATIONS, 3rd (2010) Edition Escrito por Joel Scambray,Vincent Liu,Caleb Sima

Bibliografía sugerida:

- Java Coding Guidelines: 75 Recommendations for Reliable and Secure 2013) Programs Escrito por Fred Long, Dhruv Mohindra, Robert C. Seacord
- Pro ASP.NET Web API Security: Securing ASP.NET Web API (2013) Escrito por Badrinarayanan Lakshmiraghavan
- Securing Ajax Applications: Ensuring the Safety of the Dynamic Web (2007) Escrito por Christopher Wells
- Web Security Testing Cookbook (2008) Escrito por Paco Hope,Ben Walther

Otros Recursos:

- Open Web Application Security Project. https://www.owasp.org/
- Backtrack Academy. http://www.backtrackacademy.com
- SpiderLabs® is Trustwave's elite team of ethical hackers, forensic investigators and researchers helping organizations fight cybercrime, protect data and reduce risk. https://www.trustwave.com/Resources/SpiderLabs-Blog
- Stack Overflow is a question and answer site for professional and enthusiast programmers. http://stackoverflow.com/

Correlativas

Para Cursar:

Cursadas:

- Administración de Recursos
- Redes de Información

Aprobadas:

- Todas las asignaturas del 3º Nivel

Para rendir:

Aprobadas:

- Administración de Recursos
- Redes de Información