UNIVERSIDAD DE LA CUENCA DEL PLATA

Ingeniería en Sistemas de Información

Actividad Áulica 27

Paradigmas de Programación III

Profesor: Mg. Agustin Encina

Carrera: Ingeniería en Sistemas de Información.

Cátedra: Paradigmas de Programación III

Año: Tercero.

Comisión: “A” (única).

Sede: Posadas.

Alumnos:

- Becerra, Tobias DNI: 46.084.038

Tipo de Amenaza	Causa Principal	Consecuencias	Controles Preventivos
Ataques Cibernéticos (Ej: Inyección SQL, XSS, DDoS)	Intención maliciosa de terceros, búsqueda de vulnerabilidades conocidas (OWASP Top 10), falta de sanitización de datos.	Robo de información sensible, pérdida de integridad de datos, caída del servicio, daño reputacional.	Implementación de firewalls (WAF), validación de entradas (sanitización), uso de sentencias preparadas (PDO), autenticación robusta.
Desastres Naturales (Ej: Inundaciones, incendios, terremotos)	Fenómenos incontrolables de la naturaleza que afectan la infraestructura física (servidores, datacenters).	Destrucción de hardware, interrupción prolongada del servicio, pérdida irrecuperable de datos locales.	Plan de recuperación de desastres (DRP), copias de seguridad (backups) geográficamente distribuidas (nube), infraestructura redundante.
Errores Humanos (Ej: Borrado accidental, configuración errónea)	Falta de capacitación, cansancio, negligencia o interfaces de administración confusas (mala usabilidad).	Eliminación de bases de datos, exposición de credenciales, brechas de seguridad no intencionales.	Capacitación constante, principios de privilegios mínimos, confirmación de acciones críticas (UX), auditoría de logs.
Vulnerabilidades Técnicas (Ej: Bugs de software, parches faltantes)	Errores en la lógica de programación, uso de librerías obsoletas o sistemas operativos sin actualizar.	Puertas traseras para atacantes, inestabilidad del sistema, comportamiento inesperado de la aplicación.	Actualización constante de dependencias, pruebas de seguridad (pentesting), revisión de código, gestión de parches.

Conclusión Individual

La relación entre accesibilidad, usabilidad y seguridad es fundamental para el éxito de cualquier aplicación web moderna. No pueden tratarse como silos independientes; una aplicación segura pero imposible de usar (por ejemplo, con requisitos de contraseña excesivamente complejos sin guía) fallará porque los usuarios buscarán atajos inseguros (como anotar claves en papel), comprometiendo la seguridad por falta de usabilidad. Del mismo modo, una aplicación muy usable pero que no es accesible discrimina a un sector de la población, limitando su alcance y cumplimiento legal.

Respecto a las normativas, considero que la norma más relevante actualmente es la ISO/IEC 25010. Es importante destacar que la norma mencionada en la bibliografía antigua, la ISO

9126, ha sido reemplazada y evolucionada por la familia ISO 25000 (SQuaRE). La ISO 25010 es crucial porque redefine el modelo de calidad de software, integrando la "Usabilidad" no solo como facilidad de aprendizaje, sino incluyendo la protección contra errores de usuario y la accesibilidad como sub-características vitales. Esta norma nos brinda un marco de trabajo holístico donde la calidad no es solo que el código "funcione", sino que aporte valor, sea seguro y útil para todos los usuarios