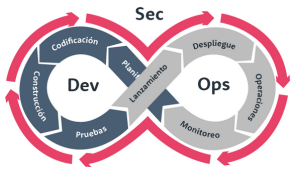
IS727272 - Cordero Hernández, Marco Ricardo



DESARROLLO DE SOFTWARE SEGURO

• CONCEPTO

El desarrollo de software seguro es una metodología (usualmente asociada con DevSecOps) que tiene el objetivo de crear software que incorpore la seguridad en cada fase del ciclo de vida del desarrollo.

• CONSIDERACIONES

Los desarrolladores y encargados de la seguridad en algún producto tienen a su disposición herramientas como frameworks, algoritmos para encriptar datos y otras técnicas útiles para prevenir fallos y mitigar afectaciones al segundo que suceden.

• POSIBLES RIESGOS (PARA LAS EMPRESAS)

Los posibles riesgos de seguridad en compañías multimillonarias como las que se encuentran en Silicon Valley son los mismos que pueden llegarse a dar en pequeños y medianas empresas, tales como:

- Filtración de datos
- Inyección de comandos SQL
- Accesos indebidos

• MÁS ALLÁ DEL FALLO

El consciente colectivo puede recordar un incidente relacionado a fallos de seguridad de software cuando se trata de una afectación que trasciende lo generacional, como lo puede ser ataques en guerras psicológicas, por ejemplo, enviar alertas falsas de impacto de misiles a usuarios desprevenidos.

Otro ejemplo es el de manipular las elecciones de algún país mediante la modificación de datos, lo cual se estaría permitiendo por brechas de seguridad inadvertidas en software gubernamental.

• NECESIDAD

El impacto que una brecha de seguridad encontrada en algún sistema puede llegar a generar va desde la afectación de la integridad y reputación de la empresa, hasta el quiebre de la misma, ya sea por una baja de consumidores o por litigaciones multimillonarias que finalmente dejarían sin recursos monetarios a las compañías.

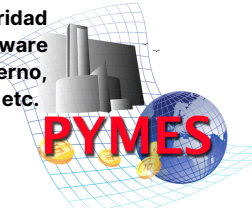
Con la certidumbre y aplicación del desarrollo seguro se buscaría mitigar estas posibles fallas.

• ¿POR QUÉ SE IGNORA?

En desarrollos comunes usualmente se pasa por alto la seguridad debido a una combinación de factores, tales como: falta de tiempo (deadlines), falta de conocimiento, deuda técnica, dirty fixes, etc.

• ALCANCE COMÚN

En contextos de desarrollo no especializado, como lo puede ser en las PyMes, es vital contar con herramientas auxiliares para prevenir brechas catastróficas en la medida de lo posible. Lograr esto es posible a través de herramientas de seguridad independientes, tales como antivirus, software oficial, servicios de alojamiento externo, encriptación, capactiación para evitar *phishing*, etc.



MATERIAL CONSULTADO

Hyperproof Team. (2022). *Secure Software Development: Best Practices, Frameworks, and Resources*. Recuperado de https://hyperproof.io/resource/secure-software-development-best-practices/.

Trąd, O. (2023). *10 cyber security risks in software development and how to mitigate them*. Recuperado de https://devtalents.com/cyber-security-during-software-development/.

Rutledge, K., McDaniel, M., Teng, S. et al. (2022). *Y2K bug*. Recuperado de https://education.nationalgeographic.org/resource/Y2K-bug/

Babur, K. (2020). *Why are Government Agencies So Vulnerable to Hacking?*. Recuperado de https://www.a10networks.com/blog/why-are-government-agencies-so-vulnerable-to-hacking/.

Lerner, M. (2020). *Government tech projects fail by default. It doesn't have to be this way.*. Recuperado de https://www.belfercenter.org/publication/government-tech-projects-fail-default-it-doesnt-have-be-way.

Segura, T. (2023). *Top 10 Practices for Secure Software Development*. https://blog.gitguardian.com/top-10-practices-for-secure-software-development/.

Ortega, C. (s.f.). *Seguridad informática para pymes*. Recuperado de https://www.questionpro.com/blog/es/seguridad-informatica-para-pymes/.