MIRA BIEN LA URL ANTES DE CLICKEAR LE DECIMOS AL USUARIO

- Nuevos TLDs / CountryCodes
  - dominios .zip - ¿Cuánto cuestan?
    - https://mrd0x.com/file-archiver-in-the-browser/
    - https://github.com/certunlp/example/archivo/refs/tags/@mrd0x.zip
- Encoding, encoding, encoding
  - https://mercadolibre.com/apple-iphone-15-pro-128-gb-titanio-negro/p/MLA27172704/@t.ly/vzyFy

A04:2021 – DISEÑO INSEGURO

- Una nueva categoría en la versión 2021.
- Se centra en los riesgos relacionados con el diseño y las fallas arquitectónicas, exhortando a un mayor uso de: modelado de amenazas, patrones de diseño seguros y arquitecturas de referencia.
- Es necesario ir más allá de la codificación y adoptar actividades cruciales para obtener Seguridad por Diseño.
- Debemos "mover a la izquierda" del proceso de desarrollo las actividades de seguridad.

A04:2021 – DISEÑO INSEGURO

- Las CWE notables incluidas son:
  - CWE-209: Generación de mensaje de error que contiene información confidencial
  - CWE-256: Almacenamiento desprotegido de credenciales
  - CWE-501: Violación de las fronteras de confianza
  - CWE-522: Credenciales protegidas insuficientemente.
  - CWE-640: Weak Password Recovery Mechanism for Forgotten Password

DISEÑO INSEGURO VS IMPLEMENTACIÓN INSEGURA

- Existe una diferencia entre un diseño inseguro y una implementación insegura.
- Incluso un diseño seguro puede tener defectos de implementación que conduzcan a vulnerabilidades que pueden explotarse.
- Un diseño inseguro no se puede arreglar con una implementación perfecta -> los controles de seguridad necesarios nunca se crearon para defenderse de ataques específicos.

DISEÑO SEGURO

- El diseño seguro del software se refiere al proceso de planificar y crear software incorporando principios y prácticas de seguridad desde las etapas iniciales del desarrollo.
- El objetivo es prevenir vulnerabilidades y ataques, garantizando que el software sea resistente frente a amenazas y riesgos de seguridad.
- Este enfoque proactivo implica considerar la seguridad en cada una de las fases del Ciclo de Vida del Desarrollo del Software
  - Pasar de SDLC a S-SDLC

CICLO DE DESARROLLO SEGURO (S-SDLC)

- Las fases típicas del S-SDLC incluyen pensar en seguridad en todas las etapas del SDLC:
  - Planificación: evaluar riesgos y estandares, certificaciones, ...
  - Análisis de Requisitos: incorporar requisitos de seguridad, modelado de amenazas, ...
  - Diseño: Diseñar pensando en minimizar problemas de seguridad. Incluir revisiones de diseño
  - Desarrollo / Implementación: Usar metodologias de desarrollo seguro, SAST, ...
  - Pruebas: incorporar DAST, ...
  - Despliegue: revision de seguridad (pentest), vuln scans, ...
  - Mantenimiento: vuln scans, monitoreo de seguridad y gestion de parches, ...
  - Retiro: tener un plan de desactivacion seguro

A04:2021 – CÓMO SE PREVIENE

- Establezca y use un ciclo de desarrollo seguro apoyado en Profesionales
- Establezca y utilice principios de diseño seguros:
  - Defensa en profundidad
  - Principio de menor privilegio / Zero Trust
  - Modelado de amenazas
- Escriba pruebas unitarias y de integración para validar que todos los flujos críticos son resistentes al modelo de amenazas.
- Separe las capas del sistema y las capas de red según las necesidades de exposición y protección.
- Limitar el consumo de recursos por usuario o servicio.