USM

UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

Seguridad de Sistemas

Clase 6: Análisis de vulnerabilidades web

Contenidos

USM

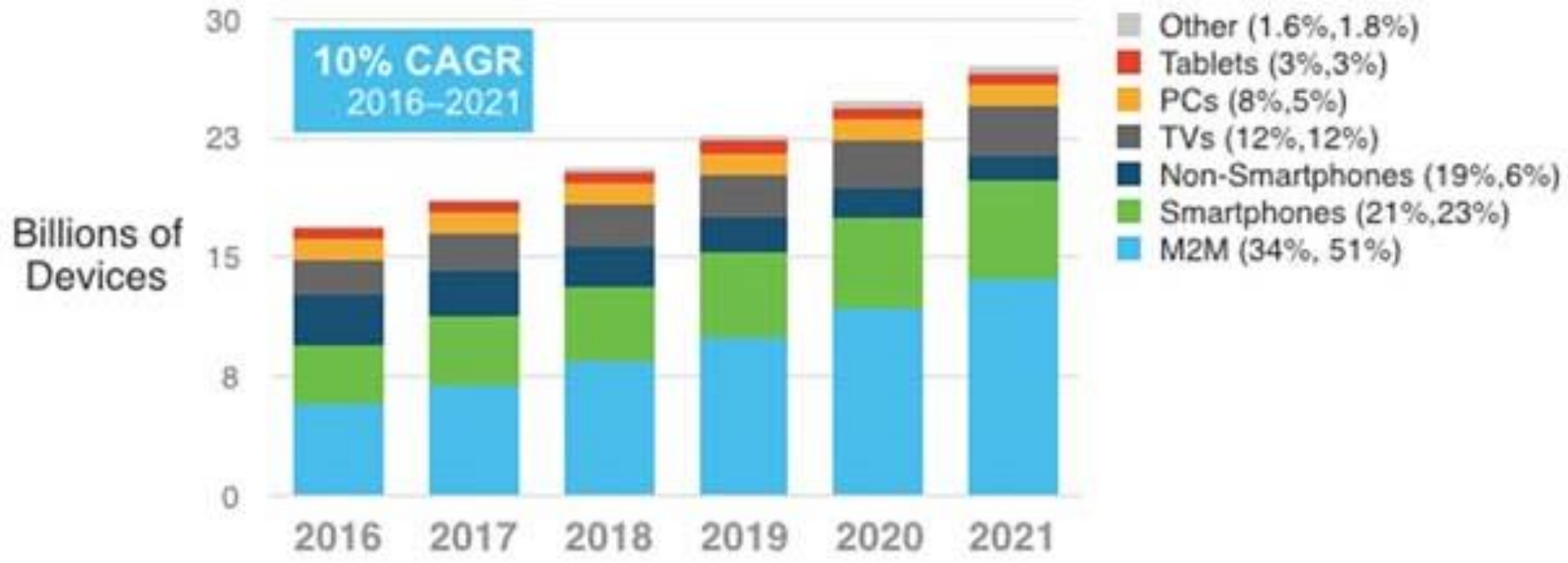
UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

- Conocer los principales riesgos en las aplicaciones web
- Conocer las principales técnicas de seguridad para aplicaciones web
- Conocer las técnicas de análisis de aplicaciones web, código fuente y SSL

Introducción

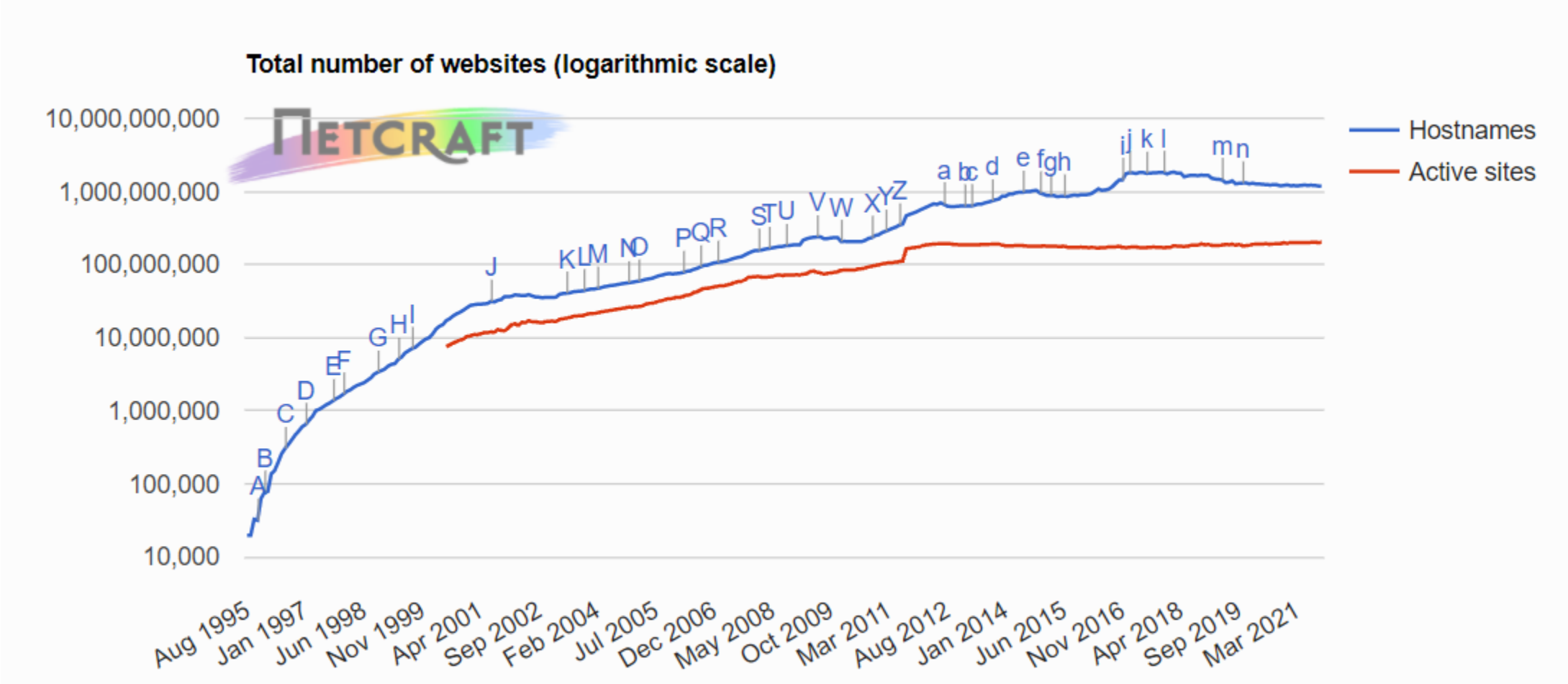
UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

- Crecimiento del tráfico web en Internet



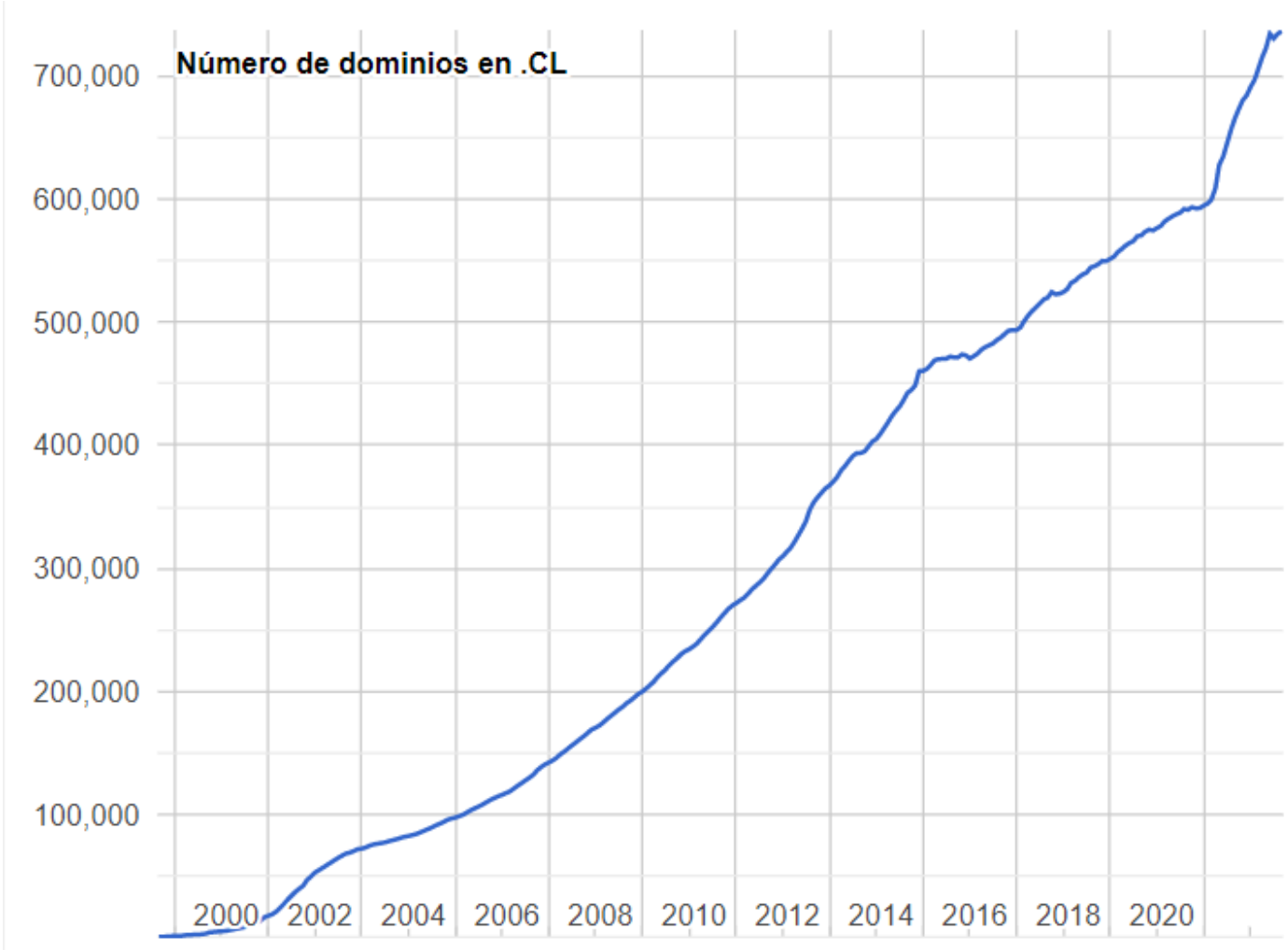
Crecimiento en Internet

UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA



Fuente: Netcraft 2021 (escala logarítmica)

Crecimiento de Internet en Chile



Fuente: NIC Chile 2021

Tipos de aplicaciones web

UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

- La tendencia del uso de las aplicaciones web ha cambiado profunda y rápidamente, migrando hacia aplicaciones de mayor interacción de usuarios, entre las principales se cuentan:

- Aplicaciones financieras (bancos)

- Compras por Internet

- Redes Sociales

- Marketing

Amenazas a las aplicaciones web

UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

- Robo de información

- Pérdida de información

- Denegación de servicio

- Defacement

- Suplantación de usuario

- Fraude

- Secuestro de sesión

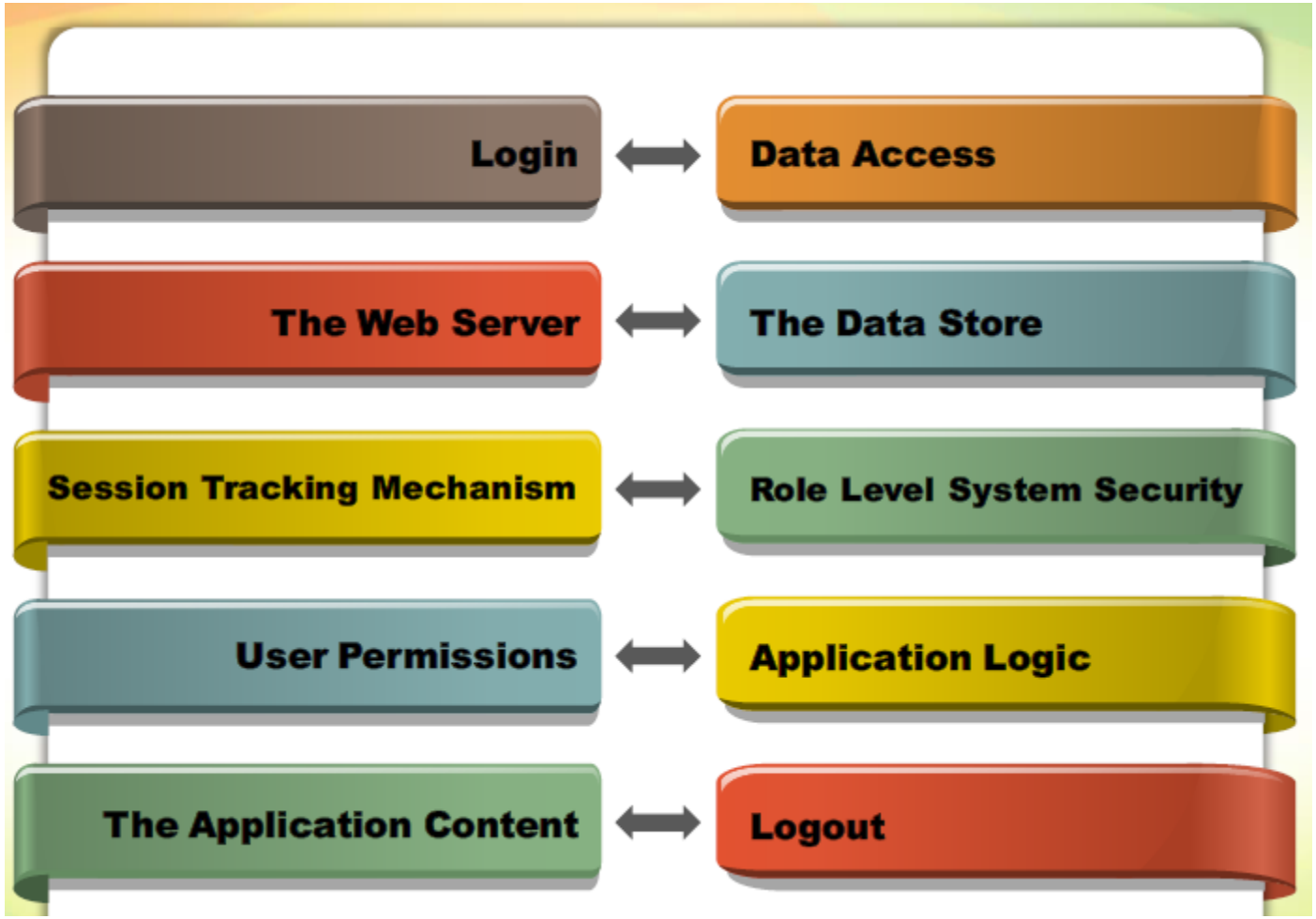
- Manipulación de parámetros

Seguridad en aplicaciones web

- **<u>Time to market:</u>**
- Dado que hoy en día, la principal preocupación de las empresas en obtener un rédito comercial de las aplicaciones web, la premura con la cual se publican produce una merma importante en la seguridad.
- Generalmente los procesos de revisión de seguridad se realizan al final del ciclo de desarrollo de las aplicaciones o bien cuando estas ya están en producción.
- Muchas veces es tarde o demasiado costoso corregir fallas de seguridad en dichas instancias.

Componentes de aplicaciones web

USM
UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA



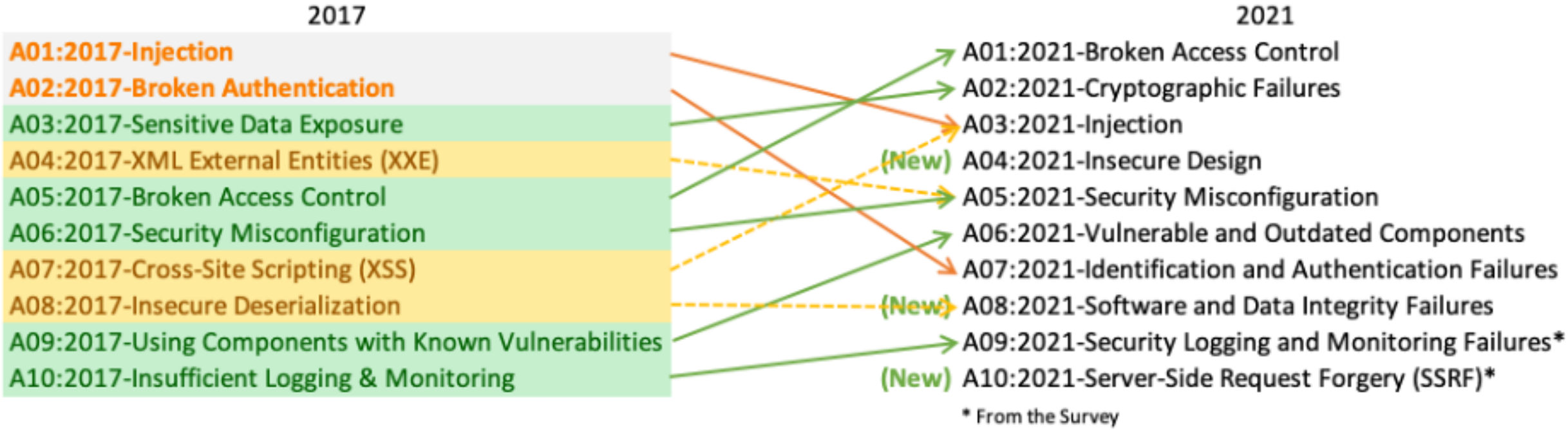
OWASP

UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

- Open Web Application Security Project (OWASP) es una comunidad en línea que produce artículos, metodologías, documentación, herramientas y tecnologías disponibles gratuitamente en el campo de la seguridad de aplicaciones web.
- Este proyecto proporciona recursos abiertos y gratuitos. Está dirigido por una organización sin fines de lucro llamada The OWASP Foundation.
- El OWASP Top 10 - 2021 es el resultado publicado de una investigación reciente basada en datos completos recopilados de más de 40 organizaciones asociadas.

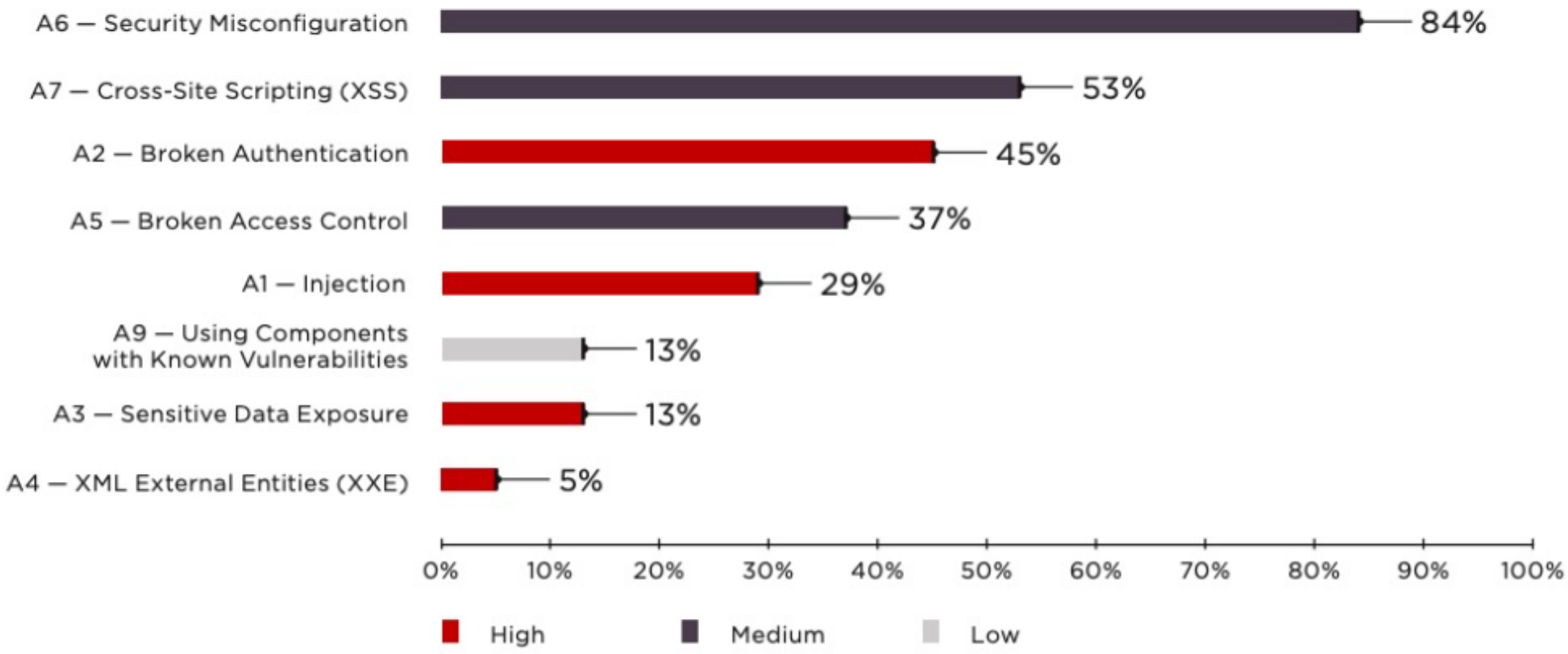
OWASP Top 10

UNIVERSIDAD TECNICA FEDERICO SANTA MARIA



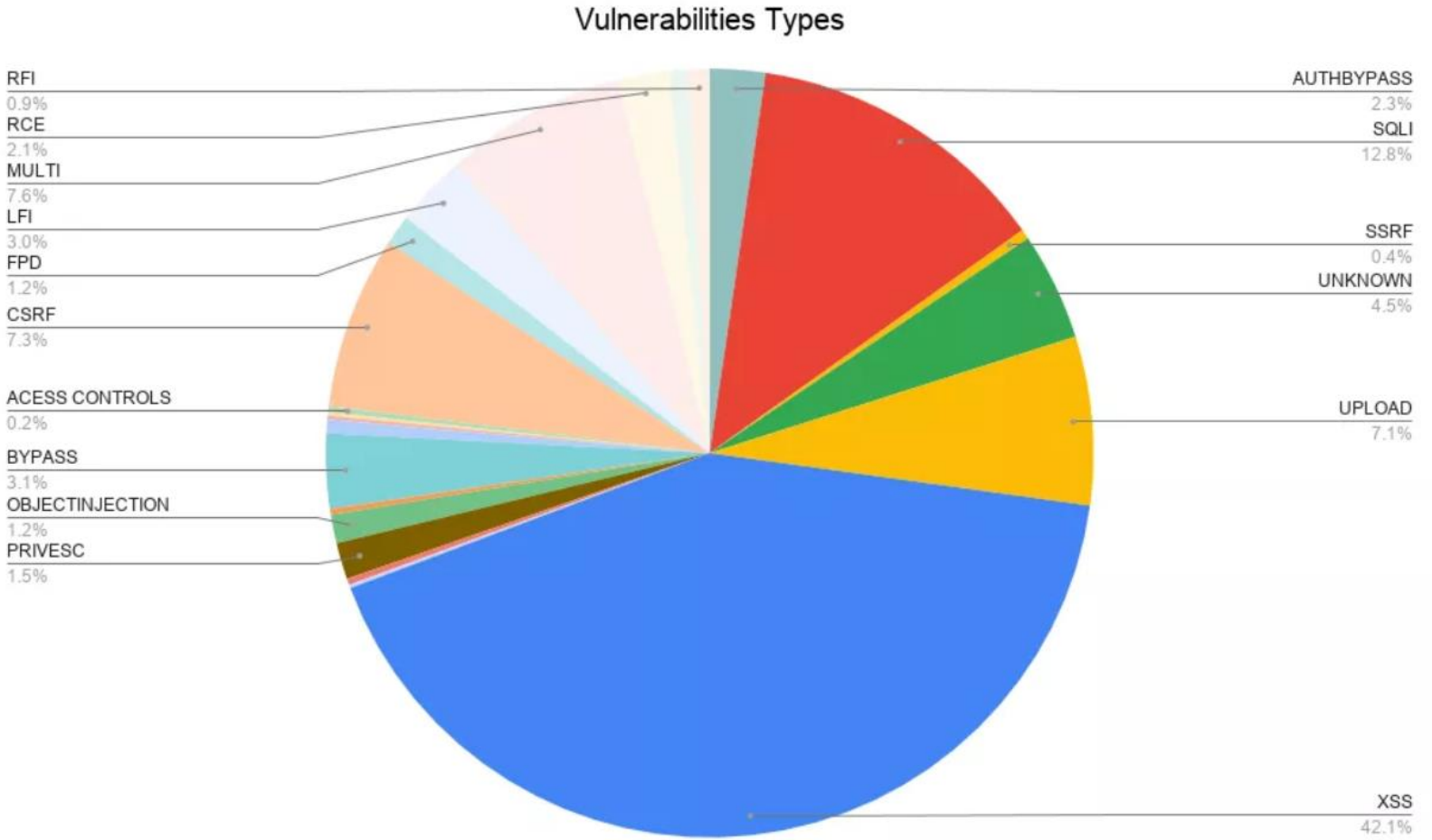
https://owasp.org/Top10/

Vulnerabilidades web más comunes



Fuente: PTSecurity 2021

Vulnerabilidades web más comunes

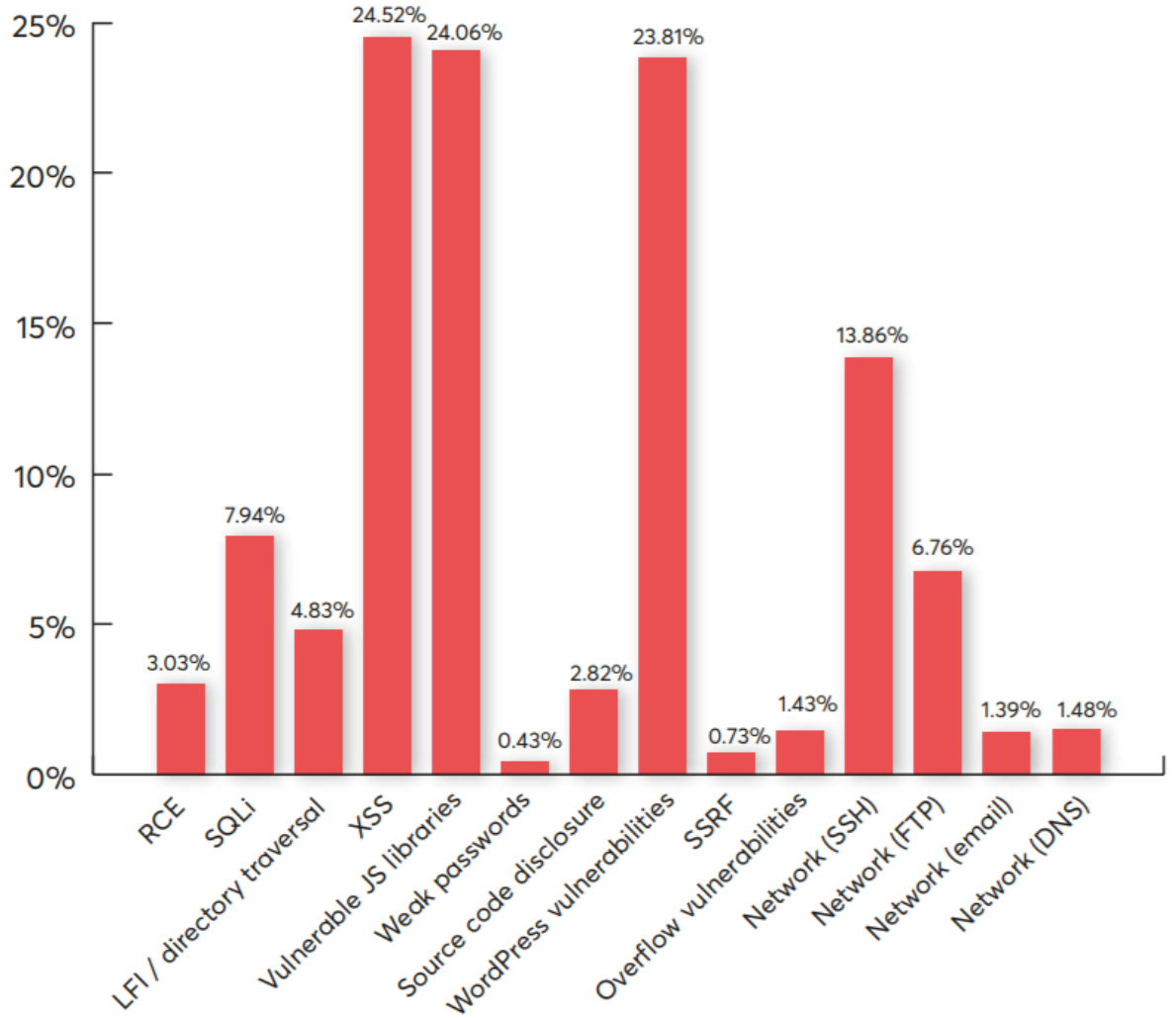


Fuente: WhiteSecurity 2021

Vulnerabilidades web más comunes

UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

Fuente: Acunetix 2021

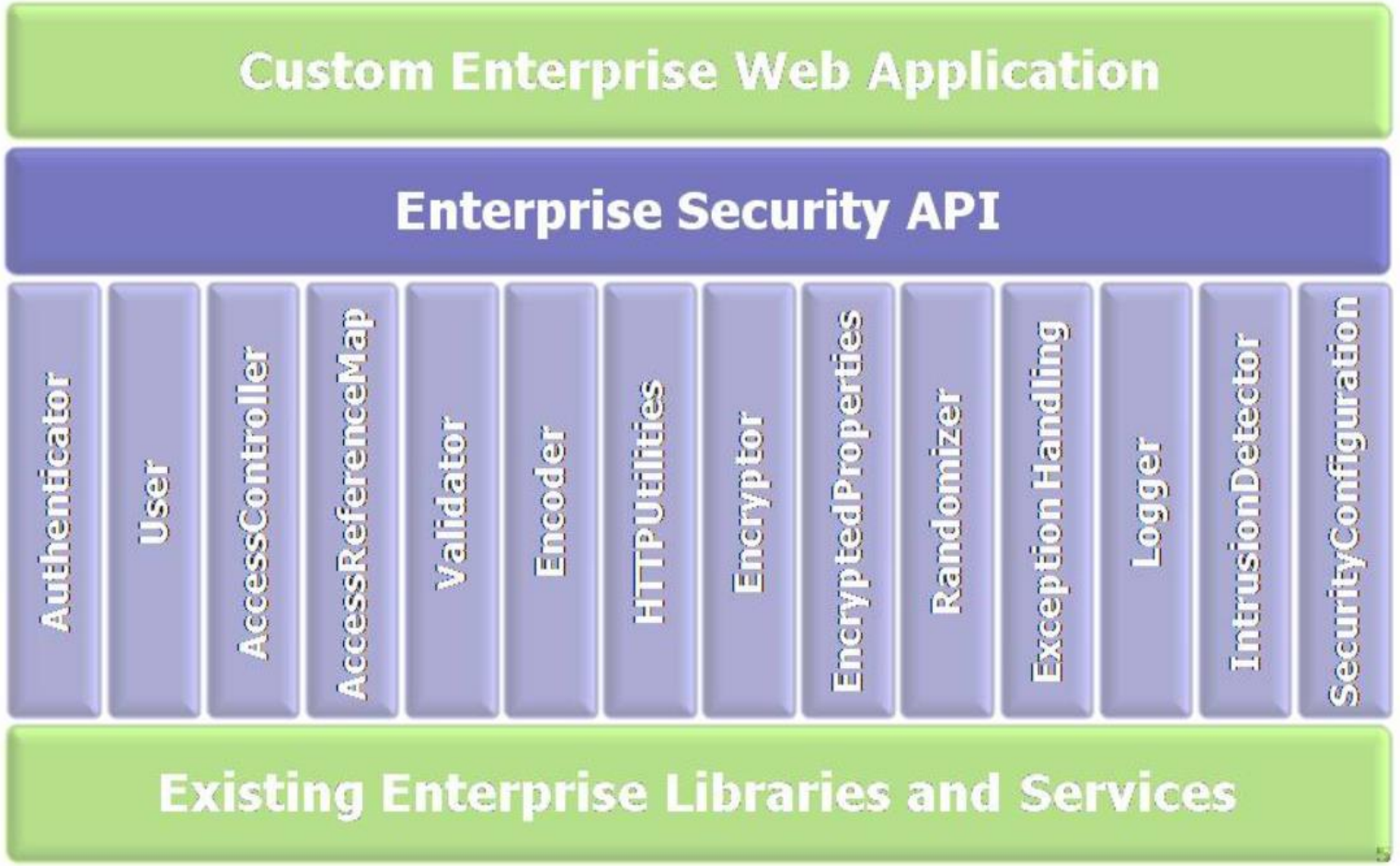


Recomendaciones

UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

- **<u>Algunas recomendaciones son:</u>**
- cifrar el tráfico cliente servidor con SSL o TLS
- validar los datos de entrada
- Utilizar listas blancas
- Utilizar rutinas de seguridad (ESAPI)
  - https://owasp.org/www-project-enterprise-security-api/
- Utilizar protección WAF

OWASP ESAPI


Custom Enterprise Web Application
Enterprise Security API
Authenticator
User
AccessController
AccessReferenceMap
Validator
Encoder
HTTPUtilities
Encryptor
EncryptedProperties
Randomizer
Exception Handling
Logger
IntrusionDetector
SecurityConfiguration
Existing Enterprise Libraries and Services

Análisis de vulnerabilidades web

USM

UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

- **<u>Tipos de evaluaciones:</u>**
- Para realizar un análisis de seguridad de aplicaciones web, existen tres metodologías utilizadas:
  - Evaluación de vulnerabilidades (VA): este proceso tiene por objetivo identificar y clasificar las vulnerabilidades de una aplicación web, es un proceso que se realiza, fundamentalmente, con herramientas.
  - Test de penetración o Ethical Hacking: tiene por objetivo lograr la explotación de las vulnerabilidades más criticas, para de esta forma generar la evidencia de la existencia de dicha vulnerabilidad y el nivel de daño que puede causar. Además se obliga al reporte de las vulnerabilidades explotadas a algún organismo de seguridad o bien al dueño de la aplicación.

Metodología

Fases de un proceso de análisis de Seguridad en aplicaciones web

Metodología

UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

- **<u>Las fases de un análisis de vulnerabilidad son las siguientes:</u>**
  - Inventariar los activos que serán parte del análisis
  - Asignar un valor de importancia a cada uno de los activos en función de rol en el negocio de la compañía
  - Identificar las vulnerabilidades con una o mas herramientas que además las cataloguen en función de su nivel de riesgo
  - Eliminar los falsos positivos que se produzcan con la información adicional de los activos que están bajo análisis
  - Entregar un reporte con el detalle y la clasificación de todas las vulnerabilidades encontradas y el procedimiento de mitigación en cada caso.
  - Opcional: la ejecución de la mitigación, generalmente no forma parte de este proceso.

Metodología OWASP

- El WSTG (Web Security Testing Guide) es una guía completa para probar la seguridad de aplicaciones web y servicios web.
- Creado por los esfuerzos de colaboración de los profesionales de la ciberseguridad y los voluntarios dedicados, el WSTG proporciona un marco de mejores prácticas utilizadas por los evaluadores de seguridad y pentester de las organizaciones de todo el mundo.
- https://owasp.org/www-project-web-security-testing-guide/v42/
  - Métodos de revisión de aplicaciones
  - Métodos de revisión de control de acceso
  - Testing de vulnerabilidades
  - Revisión de criptografía

Revisión de Seguridad SSL/TLS

UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

- **<u>Los principales aspectos a revisar son:</u>**
- Versión del protocolo
- Algoritmos de HASH
- Algoritmos de cifrado
- Negociación con diferentes tipos de clientes
- Validez del certificado digital
- Certificado digital auto-firmado

Revisión de Seguridad SSL/TLS

UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

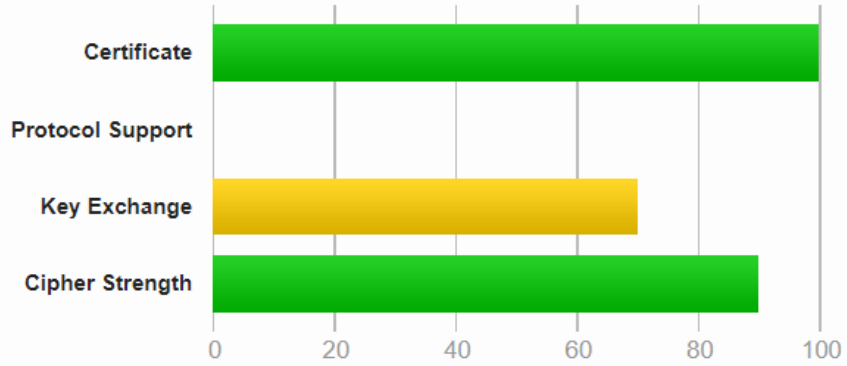
- **Breve historia de SSL/TLS:**
  - SSL 1.0, 2.0 y 3.0: Desarrolladas por la empresa Netscape, la versión 1.0 fue experimental y las versiones 2.0 y 3.0 fueron productivas lanzadas en 1995 y 1996 respectivamente.
  - En 2014 se publicó la vulnerabilidad POODLE, por el equipo de investigación de Google, lo que significó el fin de SSL 3.0
  - En 1999 ya se había estandarizado TLS 1.0 que fue el reemplazo de SSL 3.0.
  - TLS 1.1 fue estandarizada en 2006 solucionando algunas fallas que tenía la versión anterior.
  - Ambas fueron declaradas obsoletas en 2018.
  - Ya se habían publicado sus sucesoras TLS 1.2 y TLS 1.3 en 2008 y 2018 respectivamente.

Revisión de Seguridad SSL/TLS

UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

Overall Rating

F



Visit our documentation page for more information, configuration guides, and books. Known issues are documented here.

This server supports SSL 2, which is obsolete and insecure, and can be used against TLS (DROWN attack). Grade set to F. MORE INFO »

This server is vulnerable to the POODLE attack. If possible, disable SSL 3 to mitigate. Grade capped to C. MORE INFO »

This server supports weak Diffie-Hellman (DH) key exchange parameters. Grade capped to B. MORE INFO »

This server accepts RC4 cipher, but only with older protocols. Grade capped to B. MORE INFO »

This server supports TLS 1.0 and TLS 1.1. Grade capped to B. MORE INFO »

Ranking

UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

Web Site Security Testing Tools

#1) Acunetix

#2) NTOSpider

#3) Netsparker

#4) Brakeman

#5) SiteDigger

#6) NMap (Network Mapper)

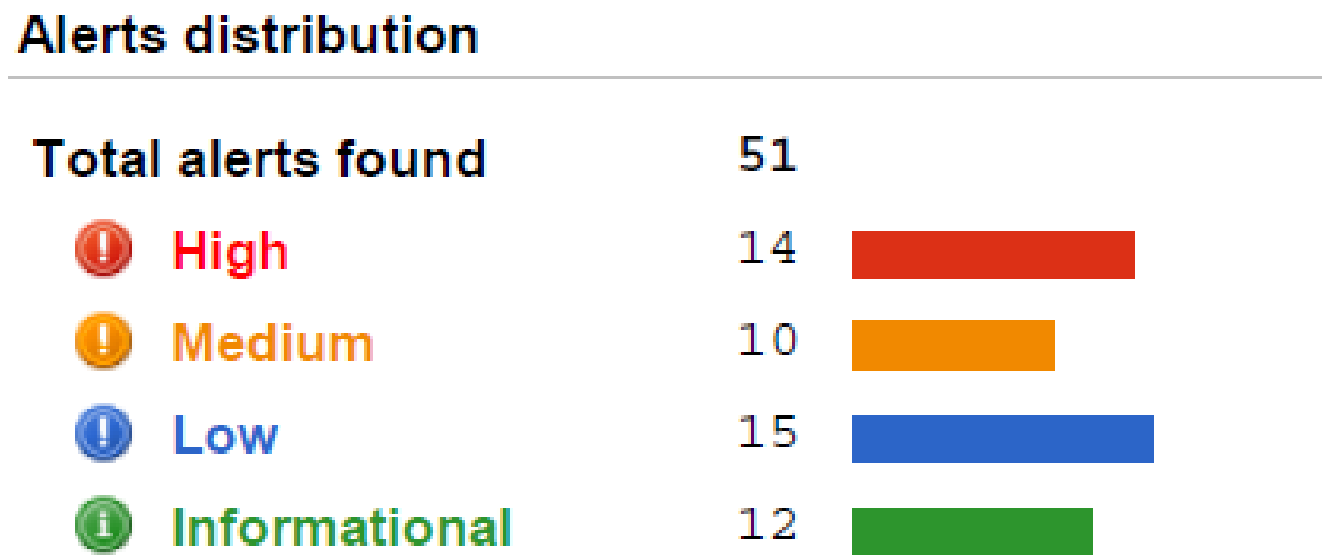
#7) OWASP

Las mejores herramientas de Análisis de Seguridad web según el ranking 2021 de Software Testing Help 2021

Acunetix

UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

- Ejemplo de clasificación de vulnerabilidades según Acunetix: herramienta especializada en vulnerabilidades de aplicaciones web
- Utiliza cuatro niveles:



Acunetix

- Ejemplo de reporte de una vulnerabilidad en una aplicación web

Parameter	passw

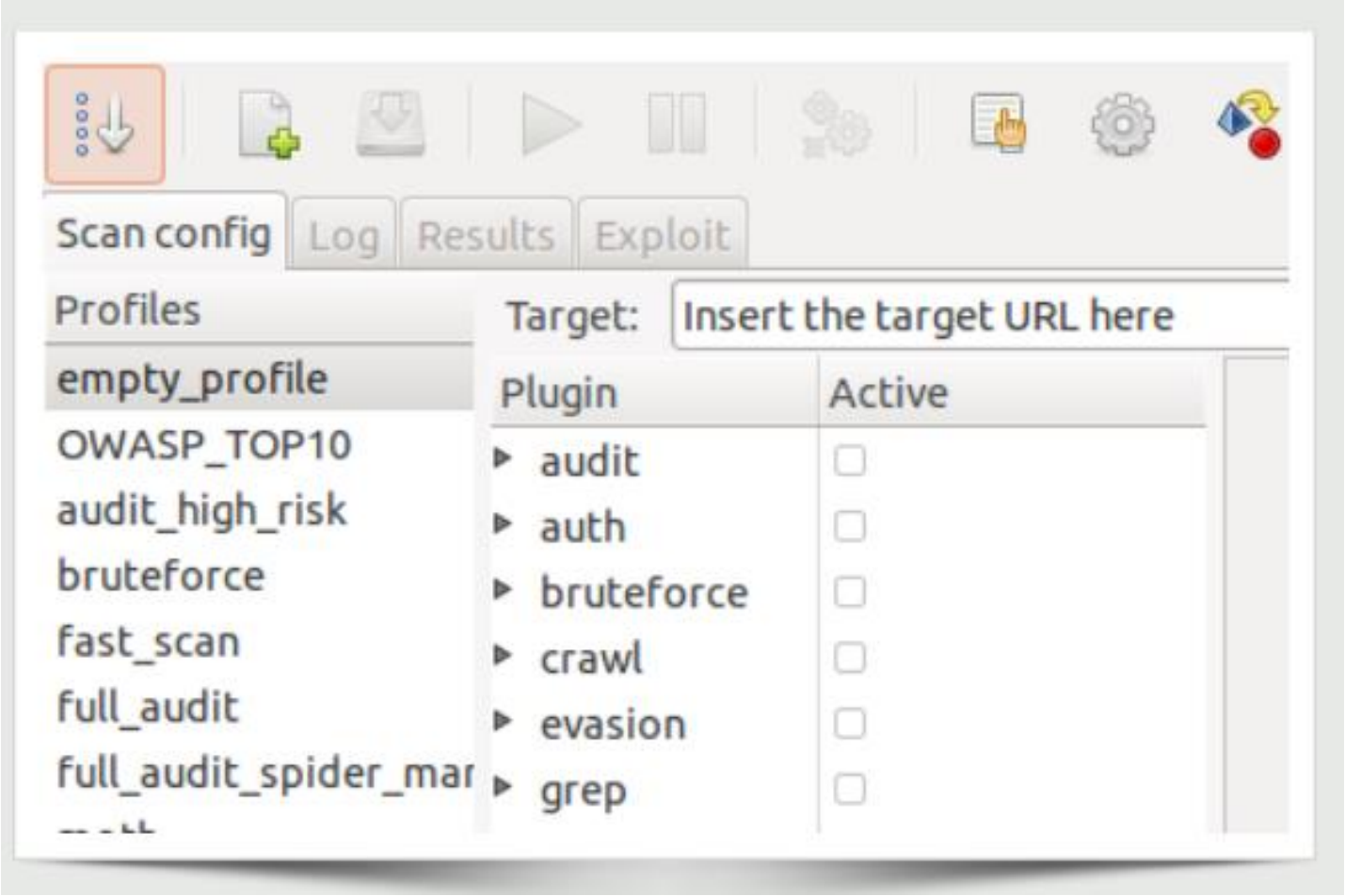
Alert group	**SQL injection**
Severity	**High**
Description	This script is possibly vulnerable to SQL Injection attacks. SQL injection is a vulnerability that allows an attacker to alter backend SQL statements by manipulating the user input. An SQL injection occurs when web applications accept user input that is directly placed into a SQL statement and doesn't properly filter out dangerous characters. This is one of the most common application layer attacks currently being used on the Internet. Despite the fact that it is relatively easy to protect against, there is a large number of web applications vulnerable.
Recommendations	Your script should filter metacharacters from user input. Check detailed information for more information about fixing this vulnerability.
Detailed	Quote from SQL Injection Attacks by Example - http://www.unixwiz.net/techtips/sql-injection.html SQL injection mitigations

W3AF

UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

- w3af es un marco de auditoría y ataque de aplicaciones web. El objetivo del proyecto es crear un marco que le ayude a proteger sus aplicaciones web mediante la búsqueda y explotación de todas las vulnerabilidades de las aplicaciones web.
- Enlaces:
- https://github.com/andresriancho/w3af
- http://w3af.org/

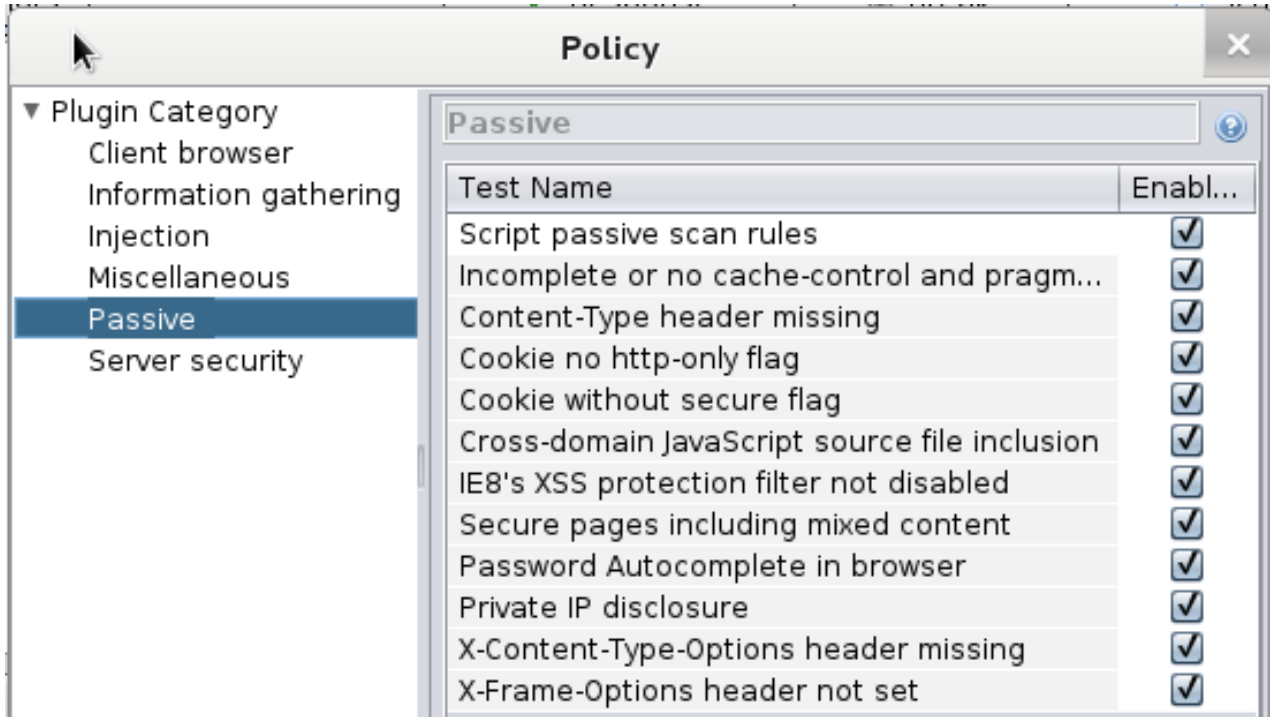
W3AF


Scan config
Log
Results
Exploit
Profiles
empty_profile
OWASP_TOP10
audit_high_risk
bruteforce
fast_scan
full_audit
full_audit_spider_mar
Target:
Insert the target URL here
Plugin
Active
audit
auth
bruteforce
crawl
evasion
grep

Zaproxy de OWASP

UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

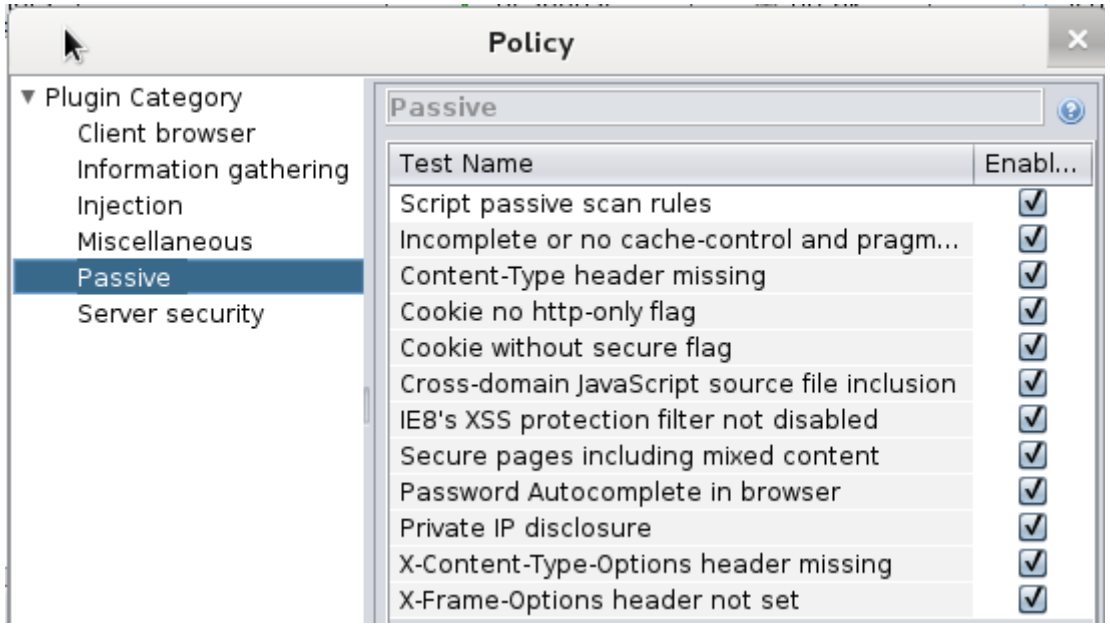
- Esta herramienta permite inyectar comandos para validar la existencia de vulnerabilidades en la aplicación web, tales como SQL Injection, CRLF, etc.



Zaproxy de OWASP

UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

- También permite realizar revisiones de seguridad pasivas en las aplicaciones web, con el objeto de determinar si esta es vulnerable, utilizando la técnica del “parsing” que consiste en ejecutar un comando y medir la respuesta.

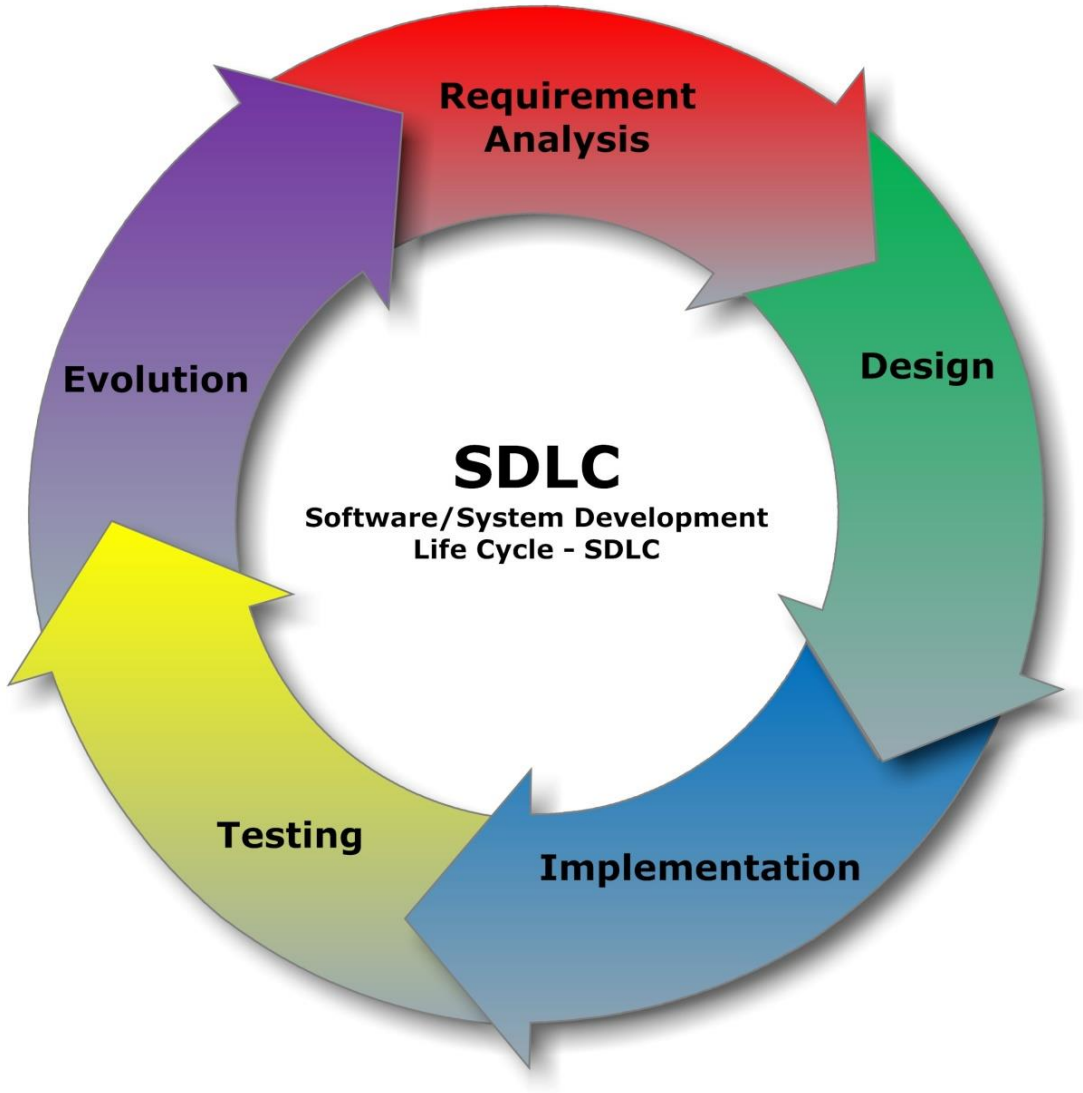


SDLC

UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

- Software Development Life Cycle: Corresponde a la metodología de desarrollo que permite crear o modificar aplicaciones en fases.
- Hoy en día, se aplica a todo tipo de desarrollo de aplicaciones, en particular a aplicaciones web.
- Las fases de las que consta esta metodología son:
  - Análisis
  - Diseño
  - Desarrollo
  - Pruebas
  - Finalización

SDLC



SDLC

UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

- **<u>Fases de un SDLC:</u>**
- Análisis: en esta etapa se definen los objetivos de la aplicación, su viabilidad y análisis de requerimientos en forma detallada y la elección de la metodología y lenguaje de programación
- Diseño: acá se describen en detalle los diagramas y procesos de negocio que formarán parte de la aplicación, los diagramas de jerarquía y reglas de negocio, en general todo el detalle para que el programador pueda empezar a trabajar

SDLC

- Desarrollo: realización del código de programación, unidades de prueba, todos los módulos que forman parte de la aplicación previo a su integración al modulo principal
- Pruebas: todas las pruebas a las que pueda ser sometida la aplicación, funcionales, de estrés, pruebas de usuario, pruebas de seguridad. En caso de alguna falla, el módulo afectado debe volver a la fase correspondiente.
- Finalización: corresponde a la entrega del proyecto, su implementación y validación de parte del área usuaria

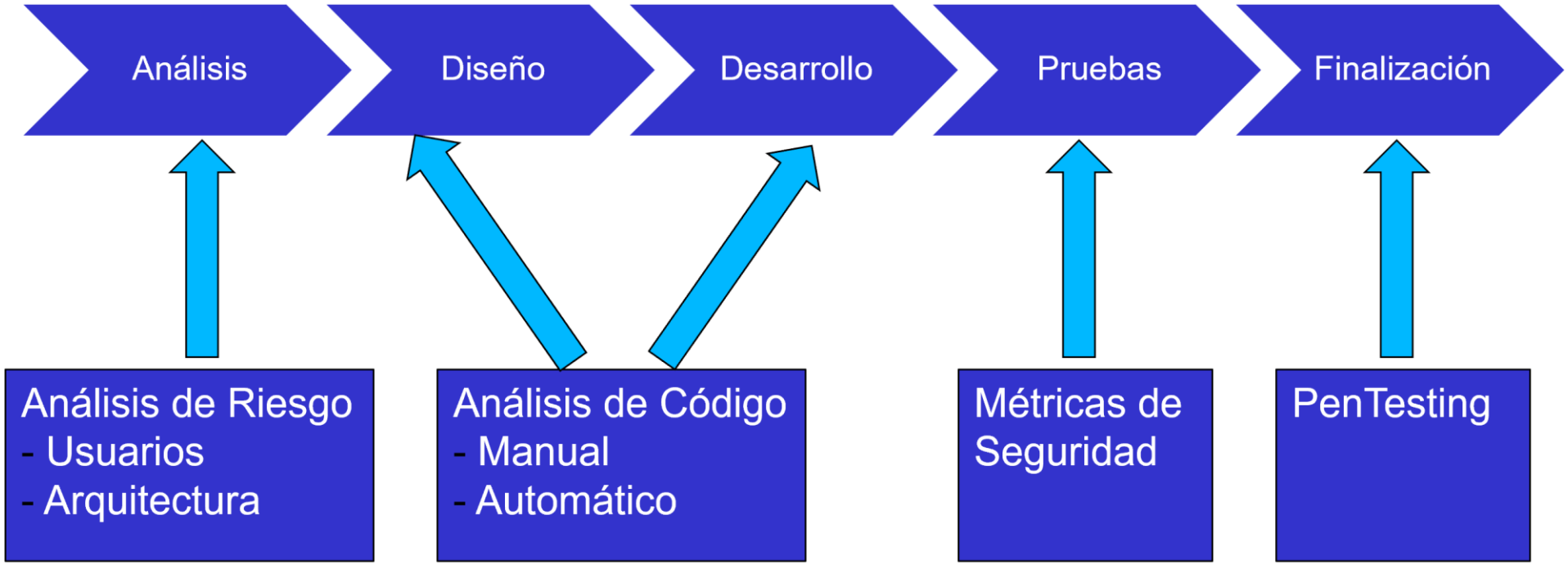
SDLC

UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

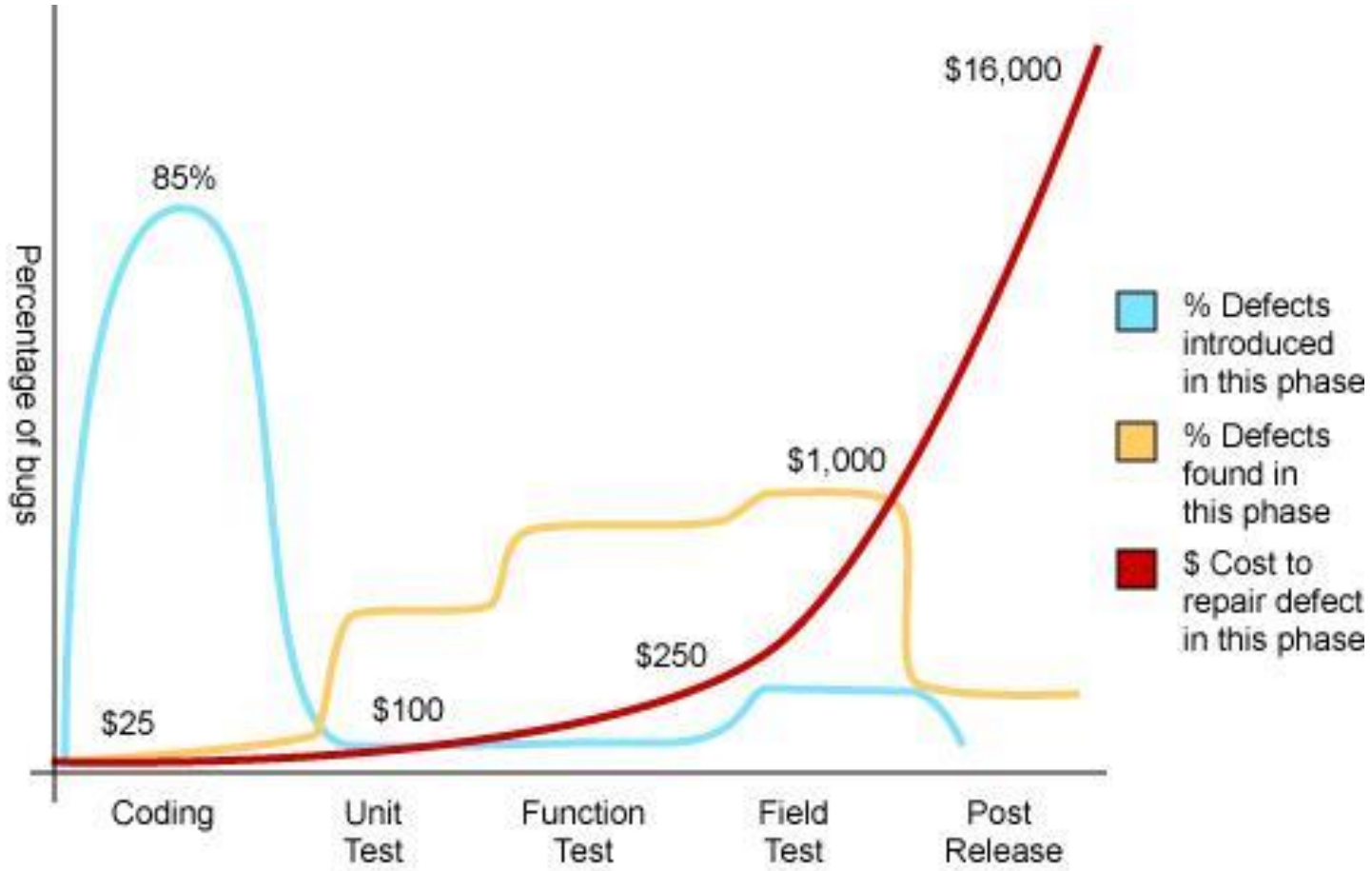
- **<u>Seguridad en SDLC:</u>**
- Consiste en aplicar revisiones de seguridad en cada una de las fases del proceso, con el objetivo de minimizar el impacto de una posible vulnerabilidad o error de programación.
- Según los especialistas, las etapas de revisión prematuras reducen los costos de mitigación posteriores
- Resolviendo el 50% de las vulnerabilidades en la etapa de desarrollo, se ahorra un 75% en mitigación
  - Fuente: Gartner

SDLC

- **OWASP en SDLC**
- OWASP recomienda la siguiente metodología para revisión de código dentro del modelo SDLC.



SDLC


Percentage of bugs
85%
$16,000
$1,000
$250
$100
$25
Coding
Unit Test
Function Test
Field Test
Post Release
% Defects introduced in this phase
% Defects found in this phase
$ Cost to repair defect in this phase

UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

Resumen

- Estadísticas de aplicaciones web
- Amenazas a las aplicaciones web
- OWASP
- Vulnerabilidades web
- Análisis de vulnerabilidades web
  - Metodología
- Seguridad SSL/TLS
- Herramientas
- SDLC

EX UMBRA IN SOLEM
USM
UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA