T2.2.1 Seguridad del Software

Garantía y Seguridad de la Información.

El problema …

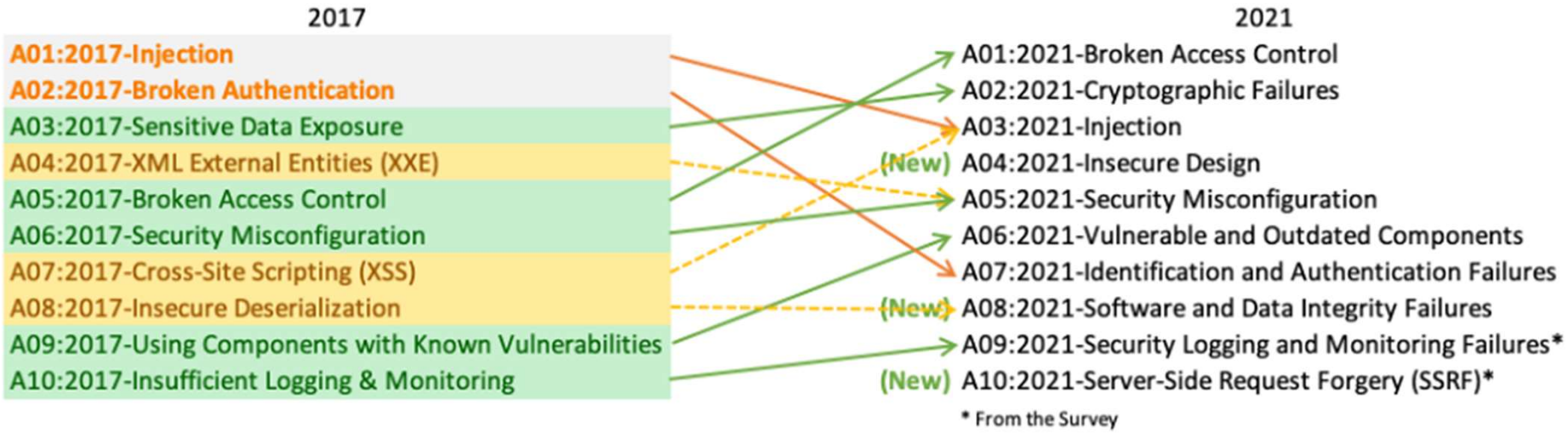
- Muchas vulnerabilidades de seguridad surgen de prácticas de programación incorrectas o defectuosas
  - Veracode State of Software Security Report
  - Lista de Errores SW más peligrosos CWE/SANS Top-25
  - Open Web Application Security Project (OWASP)
    - La lista de los Top 10

Top25 CWE/SANS List 2022

Below is a list of the weaknesses in the 2022 CWE Top 25, including the overall score of each. The KEV Count (CVEs) shows the number of CVE-2020/CVE-2021 Records from the CISA KEV list that were mapped to the given weakness.

Rank	ID	Name	Score	KEV Count (CVEs)	Rank Change vs. 2021
1	CWE-787	Out-of-bounds Write	64.20	62	0
2	CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	45.97	2	0
3	CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	22.11	7	+3 ▲
4	CWE-20	Improper Input Validation	20.63	20	0
5	CWE-125	Out-of-bounds Read	17.67	1	-2 ▼
6	CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')	17.53	32	-1 ▼
7	CWE-416	Use After Free	15.50	28	0
8	CWE-22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')	14.08	19	0
9	CWE-352	Cross-Site Request Forgery (CSRF)	11.53	1	0
10	CWE-434	Unrestricted Upload of File with Dangerous Type	9.56	6	0
11	CWE-476	NULL Pointer Dereference	7.15	0	+4 ▲
12	CWE-502	Deserialization of Untrusted Data	6.68	7	+1 ▲
13	CWE-190	Integer Overflow or Wraparound	6.53	2	-1 ▼
14	CWE-287	Improper Authentication	6.35	4	0
15	CWE-798	Use of Hard-coded Credentials	5.66	0	+1 ▲
16	CWE-862	Missing Authorization	5.53	1	+2 ▲
17	CWE-77	Improper Neutralization of Special Elements used in a Command ('Command Injection')	5.42	5	+8 ▲
18	CWE-306	Missing Authentication for Critical Function	5.15	6	-7 ▼
19	CWE-119	Improper Restriction of Operations within the Bounds of a Memory Buffer	4.85	6	-2 ▼
20	CWE-276	Incorrect Default Permissions	4.84	0	-1 ▼
21	CWE-918	Server-Side Request Forgery (SSRF)	4.27	8	+3 ▲

Top10 OWASP 2021



La Solución …

- NIST report NISTIR 8151 (2016) recomienda:
  - Atajar las vulnerabilidades antes de que aparezcan, usando métodos de especificación y desarrollo de software mejorados.
  - Localizar las vulnerabilidades antes de que puedan ser explotadas, usando técnicas de prueba mejores y más eficientes.
  - Reducir el impacto de las vulnerabilidades construyendo arquitecturas más resilientes
- ¿Seguridad de SW = Calidad y Fiabilidad del SW ?
  - Calidad y Fiabilidad: asociada a fallos de entrada, interacción o procesamiento de tipo aleatorio (errores, imprevistos, …)
    - Medida: Desarrollo estructurado, testing, testing, testing, eliminar bugs.
  - Seguridad: el atancante es quien elige la distribución de probabilidad.
    - Medida: Pon a prueba el SW (DevSecOps)

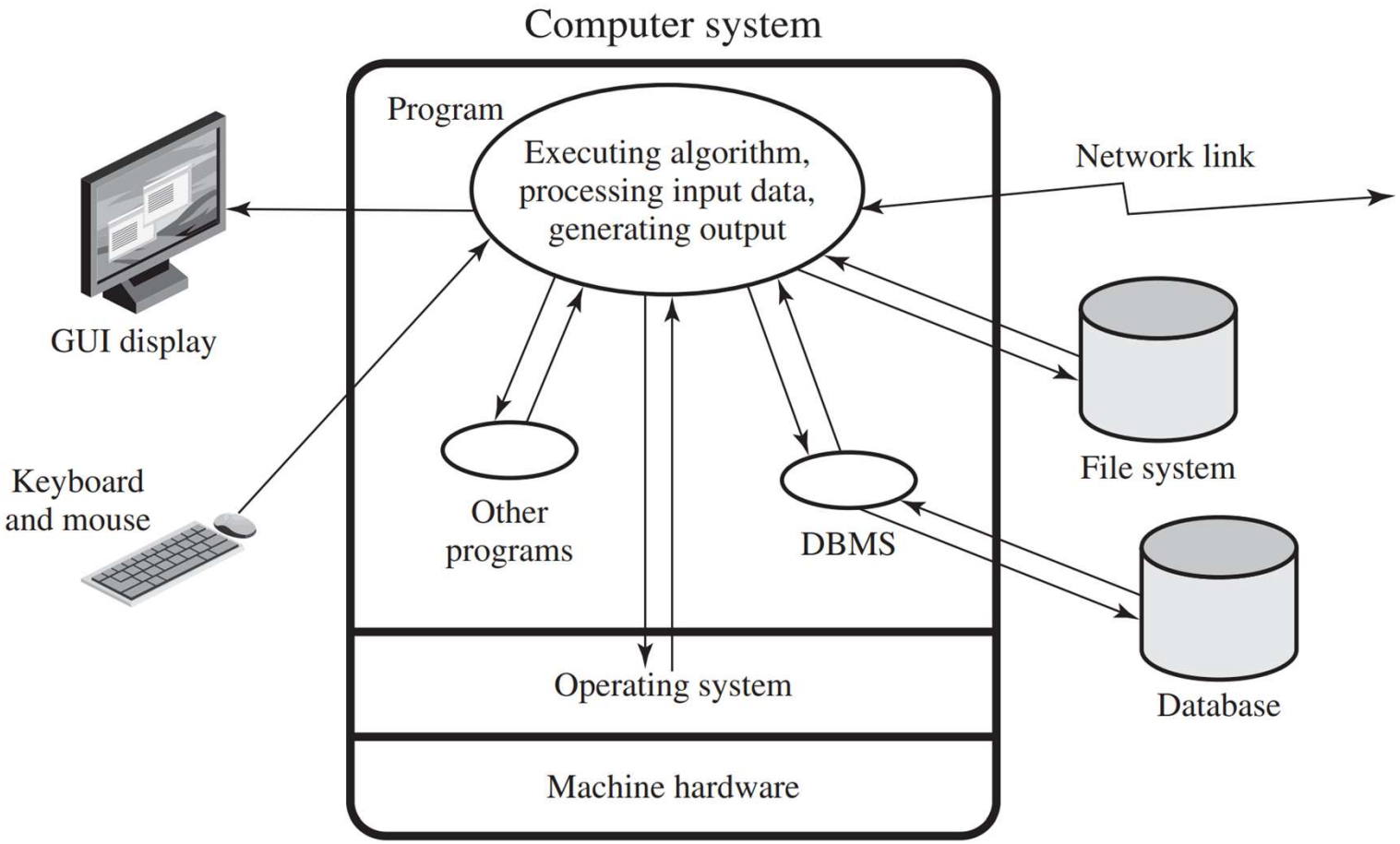
Programación Segura o Defensiva

- **Proceso de diseño e implementación de software para que siga funcionando incluso bajo condiciones de ataque.**
  - El SW construido de esta forma detectará las condiciones erróneas o imprevistas asociadas a un ataque y:
    - Seguirá ejecutando de forma segura.
    - Se detendrá la ejecución de forma ordenada y segura.
  - Regla de oro:
    - **NO DES NADA POR SUPUESTO**
    - **COMPRUEBA TODAS LAS SUPOSICIONES**
    - **MANEJA TODOS LOS POSIBLES ESTADOS DE ERROR**
  - **Cambio de Mentalidad**:
    - **De:** El programa debe resolver el problema para la mayor parte de los usuarios la mayor parte del tiempo.
    - **A**: Cuáles son las consecuencias de un fallo y las técnicas que pueden haber empleado los atacantes para sacar partido del mismo.
    - ¿Equipos mixtos?

Algunos errores clave

Derivados de la experiencia …

Vista abstracta de un programa



Tomado de: *Computer Security Principles and Practice*, William Stallings y Lawrie Brown

Manejo de la ENTRADA del programa

- Uno de los más comunes. Asociado a los datos externos al programador que se podrán aportar como entrada al programa y que no son conocidos en tiempo de programación
- PROBLEMAS
  - Tamaño de Entrada y Buffer Overflow
  - Interpretación de la Entrada
    - Fallos tratamiento información binarya
    - **Ataques de Inyección**
      - **Command inyection**
      - **SQL inyection**
      - **Code inyection**
      - **Cross-site scripting (XSS) / reflection**
- SOLUCIONES
  - Validación de la Sintaxis de la Entrada
    - Expresiones regulares
    - Caracteres de escape
  - Emborronamiento de la entrada (Input Fuzzing)
    - Prueba de entradas aleatorias

Escritura de código seguro

- PREGUNTAS CLAVE
  - Resuelve el problema el algoritmo implementado.
    - Requiere revisar el propio algoritmo y sus debilidades
  - Representan las instrucciones máquina correctamente el algoritmo
    - Asegurarse de que intérprete y/o compilador son seguros
      - No incluyen código extra
  - El almacenamiento y manejo de datos en memoria y registros es válido y con sentido.
    - Interpretación adecuada de los datos (Type-safe (C) vs. Type-free languages (Python))
    - Uso correcto de la memoria: reservar cuando se necesita (en la cantidad adecuada) y liberar cuando no se necesita (recolección de basura)
    - Prevenir condiciones de carrera en acceso a memoria compartida (barreras, semáforos, …)

Interacción con SO y otros programas

- Aspecto **crítico** desde el punto de vista de la seguridad.
  - Los programas no se ejecutan aisladamente.
  - El SO media el acceso a los recursos y los reparte entre todos los procesos.
  - El SO construye un **entorno de ejecución** para el programa
    - Son como las entradas externas al programa, desde el punto de vista de la seguridad …

- ELEMENTOS CLAVE:
  - Revisa y usa adecuadamente las **variables de entorno**
    - LD_LIBRARY_PATH, PATH, IFS, típicos objetivos de ataques.
  - Uso apropiado del **menor privilegio posible**.
  - Revisa las **llamadas al sist**ema y funciones de **Bibliotecas Estándar**.
    - Podrían no funcionar como se espera (e.g. 'borrar' ficheros)
  - Usos adecuados para **prevenir race conditions y deadlocks**.
  - Uso seguro de **ficheros temporales**.
  - Revisar y controlar comunicaciones con otros programas.

Manejo de la SALIDA DEL PROGRAMA

- Resultados de ejecución del programa
  - Se almacenan para uso o revisión posterior
  - Pueden ser binarios o textuales

- CLAVES:
  - La salida debe **ajustarse realmente a la forma e interpretación esperadas**.
  - Ojo con la **suposición de origen común**.
    - Uso de meta-secuencias de caracteres (e.g. presentación en pantalla)
    - Generación de salida que contiene código malicioso para el siguiente
  - Gestión adecuada de los cambios de juegos de caracteres y/o de codificación.

Referencias

- Base:
  - Capítulo 11 del Stalling & Brown "Computer Security"
  - X.800 y NIST 800