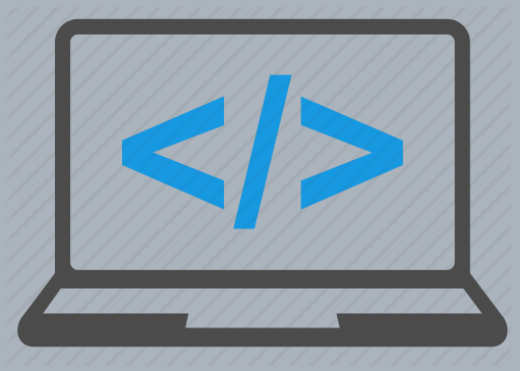
Seguridad Básica de Aplicaciones

Fundamentos y Herramientas

William Marchand Niño



Conceptos

Servicios y conceptos asociados

- Servicio WEB
- Servicio DNS
- Aplicación web
- Página web (website)
- Framework para desarrollo
- HTTP y Métodos HTTP

Mensajes HTTP

- 400: Bad Request
- 403: Forbidden
- 404: Not Found
- 200: OK
- 300: Moved Permanently
- 302: Found
- 500: Internal Server Error
- 502: Bad Gateway
- 505: HTTP Version Not Supported

Metodología

OSTTMM

OSSTMM 3 – The Open Source Security Testing Methodology Manual

- Fase de inducción
- Fase de interacción
- Fase de indagación
- Fase de intervención

OSTTMM Fase de Inducción

En la fase de inducción, el analista comienza la auditoría con una **comprensión de los requisitos de auditoría**, el alcance y las restricciones para la auditoría de este alcance. A menudo, el tipo de prueba se determina mejor después de esta fase.

MODULOS

- **Revisión**

  La revisión de la cultura, normas, normas, regulaciones, legislación y políticas aplicables al objetivo.

- **Logística**

  La medición de las restricciones de interacción, como la distancia, la velocidad y la falibilidad para determinar los márgenes de precisión dentro de los resultados.

- **Verificación de detección activa**

  La verificación de la práctica y la amplitud de la detección de la interacción, la respuesta y la previsibilidad de la respuesta.

OSTTMM Fase de Interacción

El núcleo de la prueba de seguridad básica requiere conocer el alcance en relación con las interacciones con los objetivos y los activos. Esta fase definirá el alcance.

MODULOS

- **Auditoria de visibilidad**

  La determinación de los objetivos dentro del ámbito. La visibilidad se considera como "presencia" y no se limita a la vista humana.

- **Verificación de acceso**

  La medición de la amplitud y profundidad de los puntos de acceso interactivos dentro del objetivo y la autenticación requerida.

- **Verificación de confianza**

  La determinación de las relaciones de confianza desde y entre los objetivos. Existe una relación de confianza donde el objetivo acepta la interacción entre objetivos del alcance.

- **Verificación de controles**

  La medición del uso y la efectividad de los controles de pérdida basados en procesos: no repudio, confidencialidad, privacidad e integridad.

OSTTMM Fase de Indagación (1/2)

Gran parte de la auditoría de seguridad se trata de la información que el Analista descubre. En esta fase, los diversos tipos de valor o el perjuicio de la información mal colocada y mal administrada como un activo se ponen de relieve

MODULOS

- Proceso de verificación

  La determinación de la existencia y la eficacia del registro y el mantenimiento de los niveles de seguridad o diligencia reales existentes definidos por los controles de revisión y de indemnización.

- Verificación de configuración / Verificación de entrenamiento

  La investigación del estado estable (operación normal) de los objetivos, ya que han sido diseñados para operar en condiciones normales.

- Validación de propiedad

  La medida de la amplitud y profundidad en el uso de propiedad intelectual o aplicaciones ilegales o sin licencia dentro del objetivo.

OSTTMM Fase de Indagación (2/2)

MODULOS

- **Revisión de segregación**

  Una determinación de los niveles de información de identificación personal definidos por la revisión.

- **Verificación de la exposición**

  La búsqueda de información de libre acceso que describa la visibilidad indirecta de objetivos o activos dentro del canal elegido del alcance.

- **Inteligencia competitiva**

  La búsqueda de información de libre acceso, directa o indirectamente, que podría dañar o afectar negativamente al propietario objetivo a través de medios externos y competitivos.

OSTTMM Fase de Intervención (1/2)

Estas pruebas se centran en los recursos que los objetivos requieren en el alcance.

Esos recursos pueden ser influenciados, modificados, sobrecargados o "*starved*" para causar penetración o interrupción.

Esta es a menudo la fase final de una prueba de seguridad para asegurar que las interrupciones no afecten las respuestas de las pruebas menos invasivas y porque la información para realizar estas pruebas puede no conocerse hasta que se hayan llevado a cabo otras fases.

OSTTMM Fase de Intervención (2/2)

MODULOS

- **Verificación de cuarentena**

  La determinación y medición del uso efectivo de la cuarentena para todos los accesos y dentro del objetivo

- **Auditoria de privilegios**

  El mapeo y la medición del impacto del uso incorrecto de los controles, las credenciales y los privilegios, o la escalada no autorizada de privilegios

- **Validación de la supervivencia / continuidad del servicio**

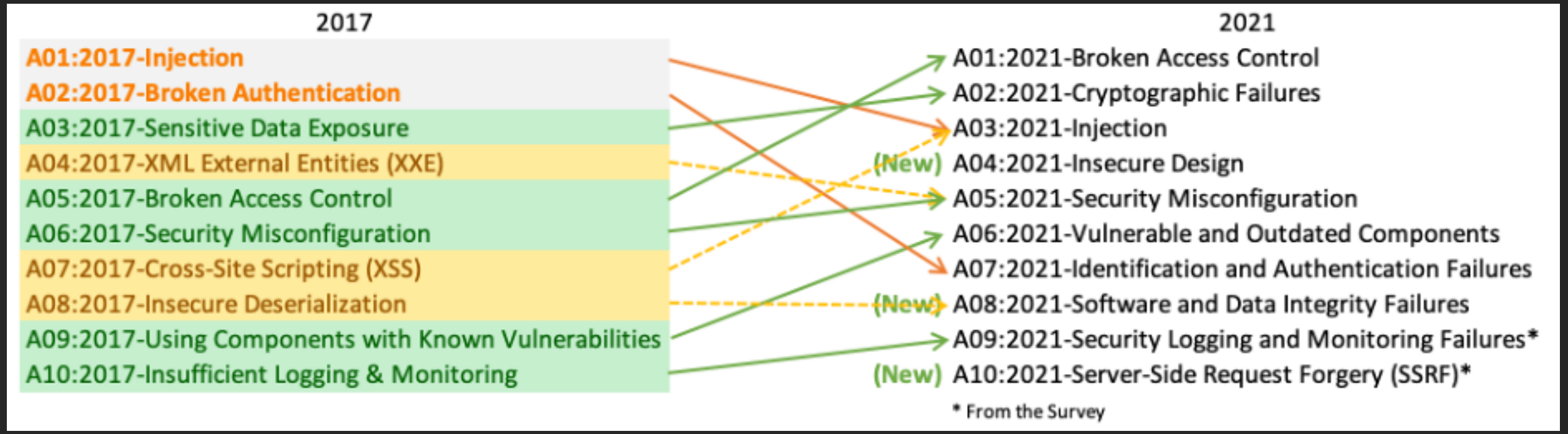
  La determinación y medición de la resiliencia del objetivo ante cambios excesivos o adversos en los que se verían afectados los controles de continuidad y resiliencia.

- **Alertas y Revisión de Logs**

  Una revisión de las actividades de auditoría realizadas con la profundidad real de esas actividades según lo registrado por el objetivo.

OWASP Web


OWASP
Open Web Application
Security Project
WEB
SECURITY
TESTING
GUIDE
VERSION 4.2



Top Ten OWASP

OWASP Mobile

Top 10 Mobile Risks - Final List 2016

- M1: Improper Platform Usage
- M2: Insecure Data Storage
- M3: Insecure Communication
- M4: Insecure Authentication
- M5: Insufficient Cryptography
- M6: Insecure Authorization
- M7: Client Code Quality
- M8: Code Tampering
- M9: Reverse Engineering
- M10: Extraneous Functionality

OWASP Mobile

OWASP API Security Top 10 2019

The Ten Most Critical API Security Risks

API1:2019 - Broken Object Level Authorization
API2:2019 - Broken User Authentication
API3:2019 - Excessive Data Exposure
API4:2019 - Lack of Resources & Rate Limiting
API5:2019 - Broken Function Level Authorization
API6:2019 - Mass Assignment
API7:2019 - Security Misconfiguration
API8:2019 - Injection
API9:2019 - Improper Assets Management
API10:2019 - Insufficient Logging & Monitoring

Ataques a Aplicaciones Web

SQL Injection

- SELECT * FROM tabla

  Extrae todos los registros de la tabla.

- UPDATE tabla SET password = ‘hackeame’ WHERE user = 'admin‘

  Cambia de valor el campo password de la tabla para el usuario admin.

- SELECT * FROM tabla WHERE user=‘Admin’ AND password=‘mypassword’

  Devuelve el registro del usuario Admin cuyo password sea mypassword

SQL Injection

- Es la posibilidad de insertar sentencias de SQL en formularios o lugares de una aplicación que utiliza consultas SQL válidas que fueron configuradas y programadas por los desarrolladores, de tal forma que se pueda manipular o extraer información de una Base de Datos

SQL Injection

- Se tiene el siguiente código referencial:

```
$user = $_POST[ 'username' ];
$pass = $_POST[ 'password' ];
$query = "SELECT * FROM usuarios WHERE
username='$user' AND password='$pass';";
```

- Inyectando código SQL

```
$user =admin’ OR ‘1’=‘1 ;
$pass =cualquiera’ OR ‘1’=‘1 ;
$query = "SELECT * FROM usuarios WHERE username='
admin’ OR ‘1’=‘1' AND password='cualquiera’ OR
‘1’=‘1';";
```

SQL Injection. Ejemplo

```
SELECT * from users where login='malo' or '1'='1' and firstname='muymalo' or '1'='1';
```

```
mysql> SELECT * from users where login='malo' or '1'='1' and firstname='muymalo' or '1'='1';
+----+--------------+-----------+----------+----------------------------------+------+----------+---------------------+---------------------+
| id | login        | firstname | lastname | password                         | salt | tradebux | created_on          | last_login_on       |
+----+--------------+-----------+----------+----------------------------------+------+----------+---------------------+---------------------+
|  1 | Sample User  | Sample    | User     | 3e912f8fc814831804d735dc2fcbc3cfa75c28e3 | NjM2 |      130 | 2009-01-05 14:29:00 | 2017-01-19 22:56:14 |
|  2 | bob          | I Am Bob  | Gilbert  | abd09072e674720d87ddd27122f67eedbc4b0d08 | Mjkx |       96 | 2009-01-05 14:51:05 | 2009-02-18 14:54:26 |
|  4 | scanner1     | Scanner   | 1        | af256af3d4fda990dbe546daa04e5c75eae356ea | ODUy |      100 | 2009-02-18 14:46:21 | 2009-02-18 14:46:21 |
|  5 | scanner2     | Scanner   | 2        | f9335d39b2b78018c2b8affa7fc7b0917a3300a7 | MzI5 |      100 | 2009-02-18 14:46:34 | 2009-02-18 14:46:34 |
|  6 | scanner3     | Scanner   | 3        | 43754746b4043c852864bb321e4f2648d1421c18 | Nzk3 |      100 | 2009-02-18 14:46:51 | 2009-02-18 14:46:51 |
|  7 | scanner4     | Number    | 4        | e514a672396679528c766a92a857eac4b22bc667 | NjEx |      100 | 2009-02-18 14:47:04 | 2009-02-18 14:47:04 |
|  8 | scanner5     | Number    | 5        | f38ae9b0b6b1ad2a2a2721841c0cc89b31e044cb | NTQw |      100 | 2009-02-18 14:47:18 | 2009-02-18 14:47:18 |
|  9 | wanda        | Wanda     | Granat   | 4e4465300b14b314384a6375a837f0532822d3c8 | Nzcz |      100 | 2009-02-18 14:53:23 | 2009-02-18 14:53:23 |
| 10 | calvinwatters | Calvin    | Watters  | 81418ed6e9bd15076d2f43e17b9f5a27c7e55ef7 | Nzc5 |      100 | 2009-02-18 14:56:11 | 2009-02-18 14:56:11 |
| 11 | bryce        | Bryce     | Boe      | 478fb0b83851b3d16ffc5a2554a4d616f1235156 | NjY3 |       74 | 2009-02-18 14:57:36 | 2017-01-16 00:31:15 |
+----+--------------+-----------+----------+----------------------------------+------+----------+---------------------+---------------------+
10 rows in set (0.00 sec)
```

SQL Injection. Ejemplo

Utilizando UNION

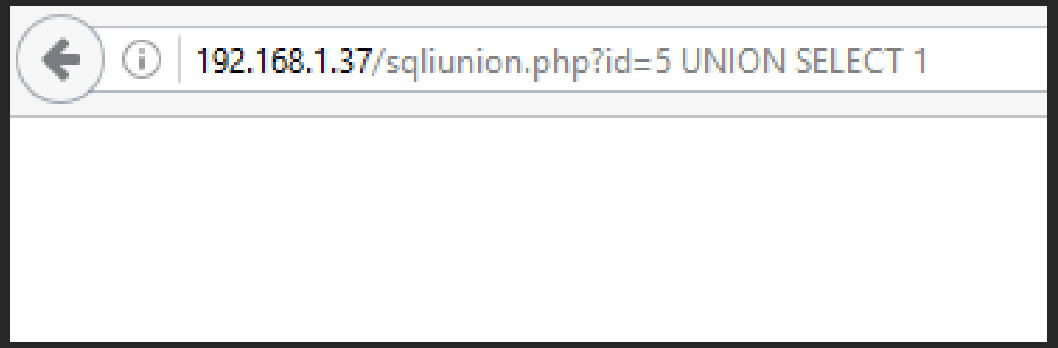
Buscar la cantidad de columnas.

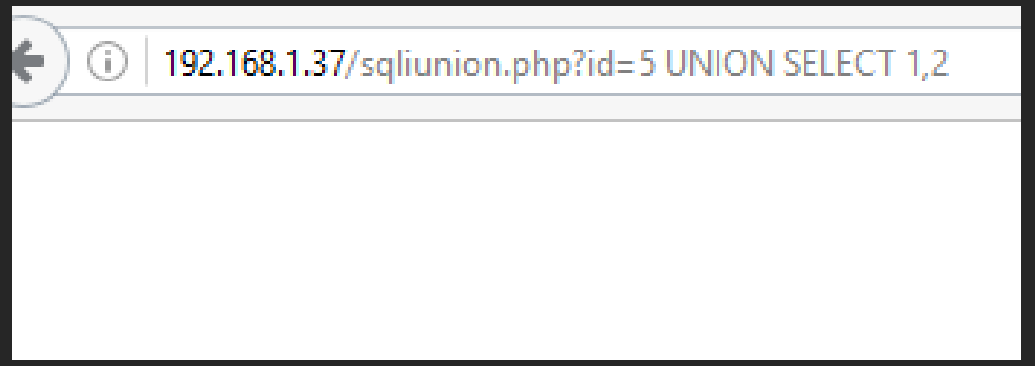
Sqliunion.php?=1 UNION SELECT 1

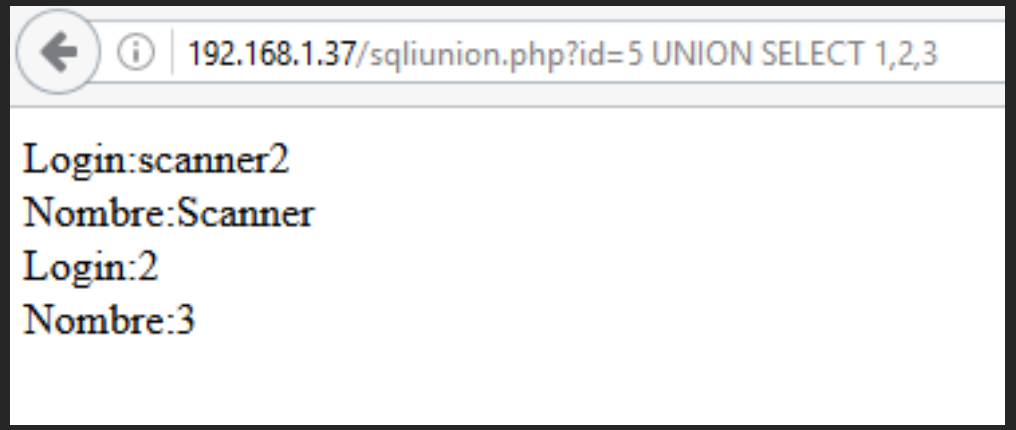
Sqliunion.php?=1 UNION SELECT 1,2

Sqliunion.php?=1 UNION SELECT 1,2,3

SQL Injection. Ejemplo







Algunas opciones de consultas en SQLi

UNION SELECT 1,2,version() → versión del motor de base de datos
UNION SELECT 1,2,database() → base de datos actual.
UNION SELECT 1,2,user() → usuario activo de la base de datos
UNION SELECT 1,2,schema_name FROM information_schema.schemata → Lista de bases de datos.
UNION SELECT 1,2,table_name FROM information_schema.tables → Lista de tablas
UNION SELECT 1,2,table_name FROM information_schema.tables WHERE table_schema='mibasedatos' → Tablas de una base de datos específica.
UNION SELECT 1,2,column_name FROM information_schema.columns WHERE table_name='users' → columnas de una table específica.
UNION SELECT 1,login,password FROM mibasedatos.users → registros de usuarios y passwords
UNION SELECT 1,2,CONCAT(login,' : ' ,password) FROM mibasedatos.users → registros de usuarios y passwords concatenados.

Mitigación de SQLi

Validación de datos

- Cada dato debe ser validado cuando se recibe para asegurarse que es del tipo correcto, y rechazado si no pasa ese proceso de validación.

Sanitización de datos

- Se centra en manipular los datos para asegurarse que son seguros, eliminando cualquier parte indeseable y normalizándolos en la forma correcta.

Escapar los caracteres especiales utilizados en las consultas SQL

- Al “escapar caracteres” estamos haciendo referencia a añadir la barra invertida “\” delante de las cadenas utilizadas en las consultas SQL para evitar que estas corrompan la consulta. Algunos de estos caracteres especiales que es aconsejable escapar son las comillas dobles (“) y las comillas simples (‘).

Asignar mínimos privilegios al usuario que conectará con la base de datos

Laboratorio

- Extraer los nombres de usuario y hashes de contraseñas de la base de datos y del sistema operativo si fuese posible. Aplicación: DVWA.
- Descubrir la mayor cantidad de contraseñas válidas.
- Para evidenciar el trabajo, deberá agregar una cuenta (usuario, password) en la base de datos de DVWA y deberá aparecer en las capturas de pantalla que realice.

XSS Cross Site Scripting

- Forma de ataque que aprovecha el mal filtrado de datos en una aplicación web. Es un ataque que se orienta al lado del cliente.
- Para cambiar el comportamiento de la aplicación se deberá inyectar código como por ejemplo JavaScript u otro tipo que se ejecute del lado del cliente (*client-side*).
- Lo que se realice con esta técnica no afecta directamente al servidor, sin embargo se pueden obtener cuentas de usuarios o redireccionar las visitas.
- Los tipos de XSS mas comunes son el Reflejado y el Almacenado.

XSS Reflejado

- La inyección de código no permanece en la página que pueda afectar a todos los visitantes de la aplicaciones, por el contrario es dirigido a usuarios específicos.
- El código mas simple utilizado para verificar la vulnerabilidad XSS es el siguiente:

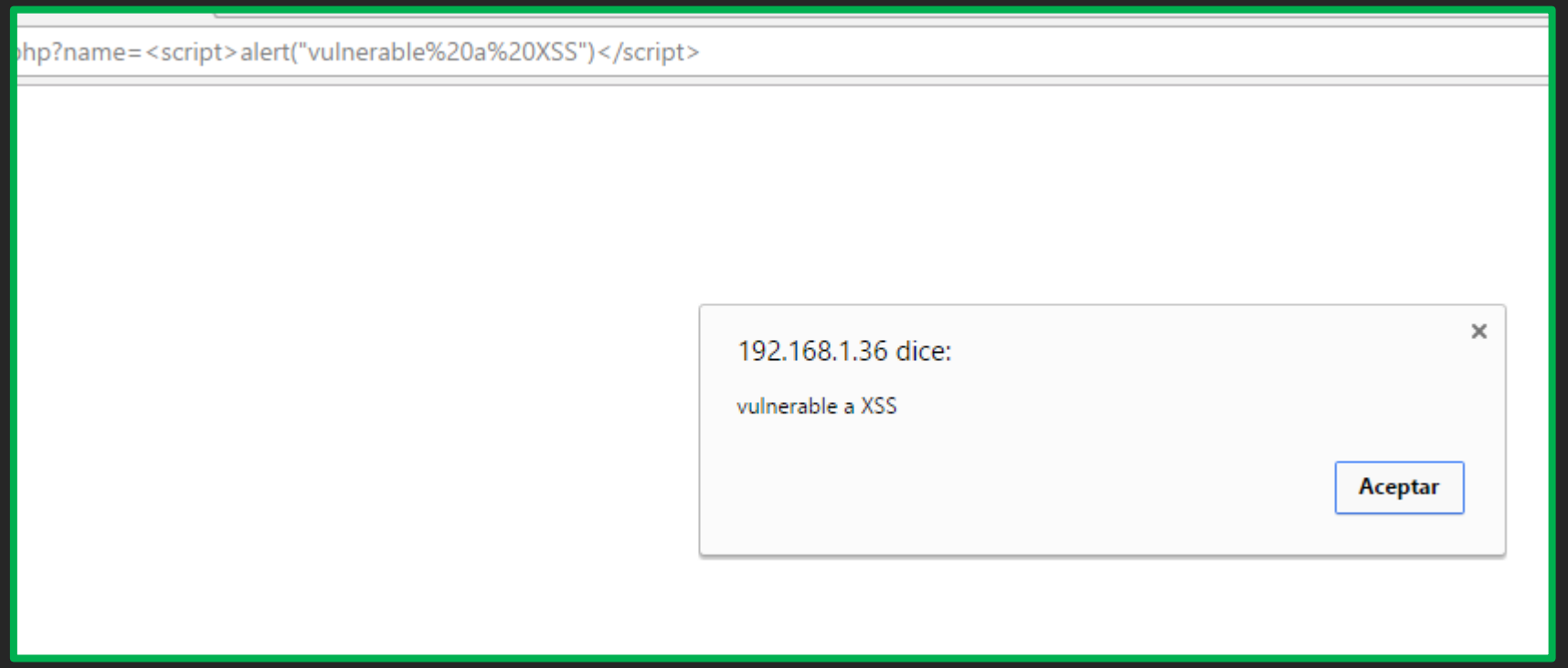
<script>alert(“vulnerable a XSS”)</script>

- Opciones:

<script>alert(1)</script>

<script>alert(“XSS”)</script>

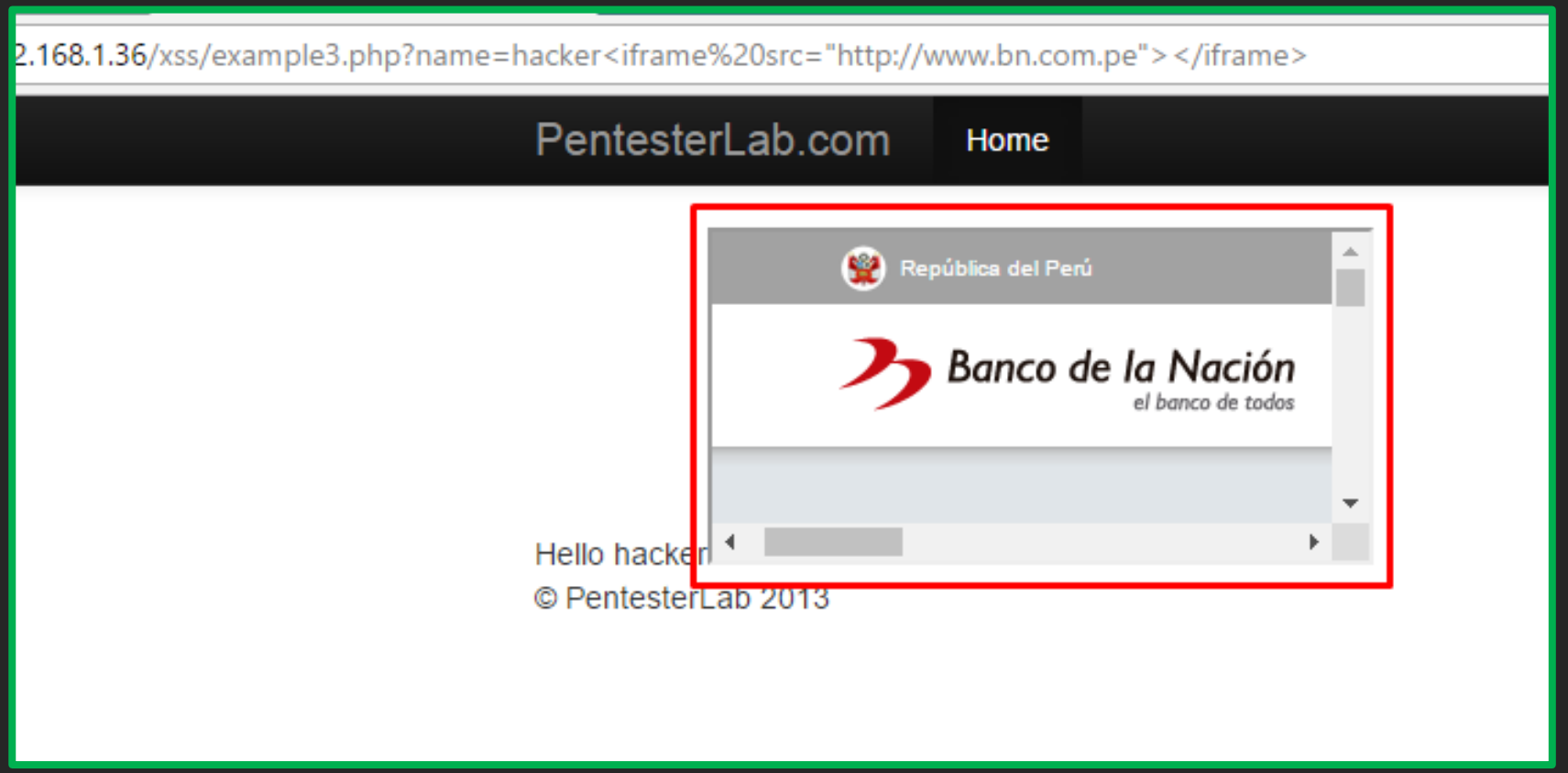
XSS Reflejado



XSS Reflejado

- Insertar paginas maliciosas dentro de la página real.

<iframe src="http://www.bn.com.pe"></iframe>



XSS Reflejado

- Obtener cookies

```
<script>alert(document.cookie)</script>
```

- Se puede generar un ataque de obtener cookies con los inicios de sesión que el usuario haya realizado, por ejemplo, bancos, compras, servidores, etc.

Mitigación de XSS

Validación de datos

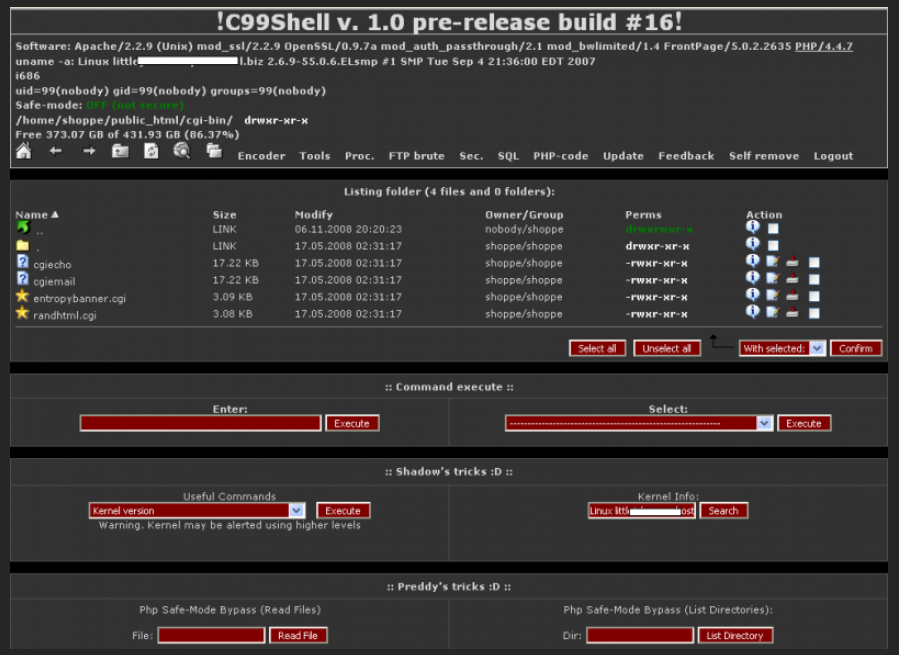
- Cada dato debe ser validado cuando se recibe para asegurarse que es del tipo correcto, y rechazado si no pasa ese proceso de validación.

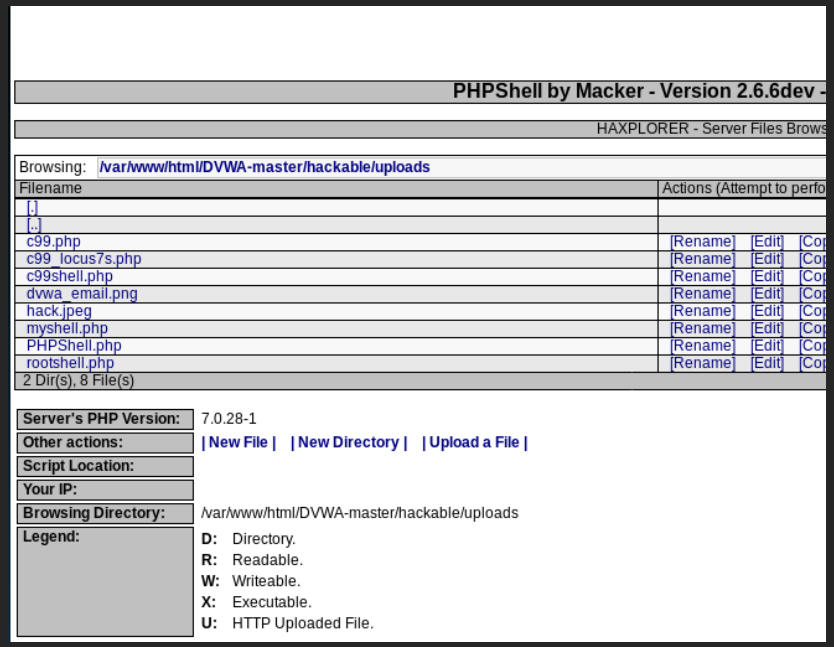
Sanitización de datos

- Se centra en manipular los datos para asegurarse que son seguros, eliminando cualquier parte indeseable y normalizándolos en la forma correcta.

File Upload

- Posibilidad de subir archivos maliciosos a servidores o maquinas objetivos para posteriormente ejecutarlos y obtener diversos resultados como, ejecución de un exploit, payload, webshell, etc.





Mitigación de File Upload

Validación de datos

- Tamaño de archivo, tipo, extensiones.

Eliminar metadatos

- Antes de almacenar los archivos se deben quitar los metadatos.

Laboratorio

- En la aplicación: DVWA, probar los ataques de XSS Reflejado y File Upload