



# Web App Vulnerability Scanning

Security Architecture & Tool Sets

# Web Application Vulnerability Scanning

- Dedicated web app vulnerability scanners do better than Nessus, Nmap, and OpenVAS
- Identify problems with applications and the underlying web servers, databases, and infrastructure
- Examples
  - Acunetix WVS
  - Archi
  - Burp Suite
  - IBM's AppScan
  - HP's WebInspect
  - Netsparker
  - QualysGuard's Web Application Scanner
  - W3AF



# Acunetix

acunetix

Administrator 4

Most Vulnerable Targets

		Vulnerability Severity	
Criticality	Target	High	Medium
NORMAL	<a href="http://testphp.vulnweb.com">http://testphp.vulnweb.com</a>	46	64
NORMAL	<a href="http://testasp.vulnweb.com">http://testasp.vulnweb.com</a>	20	36
NORMAL	<a href="http://testaspnet.vulnweb.com">http://testaspnet.vulnweb.com</a>	9	18

Top Vulnerabilities

Vulnerability	Times Identified
HTML form without CSRF protection	50
Blind SQL Injection	21
SQL Injection	20
User credentials are sent in clear text	19
Cross site scripting	19
Application error message	15
Directory listing	10
Unencrypted __VIEWSTATE parameter	10
Weak password	3
Backup files	2

Vulnerabilities by Severity

Severity	Percentage
High	33.78%
Medium	53.15%
Low	13.06%

Vulnerabilities by Severity and Criticality

Business Criticality	Severity		
	High	Medium	Low
Critical	0	0	0
High	0	0	0
Normal	112	118	29
Low	0	0	0

© 2016 Acunetix Ltd.

JD JASON DION  
TRAINING THE CYBER SECURITY WORKFORCE

JD JASON DION  
TRAINING THE CYBER SECURITY WORKFORCE

# Acunetix

**SQL injection (verified)**

**Vulnerability description**

This script is possibly vulnerable to SQL Injection attacks.

SQL injection is a vulnerability that allows an attacker to alter backend SQL statements by manipulating the user input. An SQL injection occurs when web applications accept user input that is directly placed into a SQL statement and doesn't properly filter out dangerous characters.

This is one of the most common application layer attacks currently being used on the Internet. Despite the fact that it is relatively easy to protect against, there is a large number of web applications vulnerable.

This vulnerability affects [/listproducts.php](#).

Discovered by: Scripting (Sql\_Injection.script).

**Attack details**

URL encoded GET input `artist` was set to `(select 1 and row(1,1)>(select count(),concat(concat(CHAR(52),CHAR(67),CHAR(117),CHAR(98),CHAR(52),CHAR(117),CHAR(78),CHAR(77),CHAR(72),CHAR(79),CHAR(55)),floor(rand()*2))x from (select 1 union select 2)a group by x limit 1)`

Injected pattern found:

`4Cub4uNMH07`

View HTTP headers

View HTML response

Launch the attack with HTTP Editor

Retest alert(s)

Mark this alert as a false positive

**The impact of this vulnerability**

An attacker may execute arbitrary SQL statements on the vulnerable system. This may compromise the integrity of your database and/or expose sensitive information.

Depending on the back-end database in use, SQL injection vulnerabilities lead to varying levels of data/system access for the attacker. It may be possible to not only manipulate existing queries, but to UNION in arbitrary data, use subselects, or append additional queries. In some cases, it may be possible to read in or write out to files, or to execute shell commands on the underlying operating system.

Certain SQL Servers such as Microsoft SQL Server contain stored and extended procedures (database server functions). If an attacker can obtain access to these procedures it may be possible to compromise the entire machine.

**How to fix this vulnerability**

Your script should filter metacharacters from user input.

Check detailed information for more information about fixing this vulnerability.

**Detailed information**

Click here for more detailed information about this vulnerability

**Web references**

- [Acunetix SQL Injection Attack](#)
- [Advanced SQL Injection](#)
- [Security Focus - Penetration Testing for Web Applications \(Part Two\)](#)
- [More Advanced SQL Injection](#)



# Manual Scanning

- Uses an interception proxy to capture communications between browser and server
- Testers can modify data sent and received
- Examples
  - Tamper Data for Firefox and Chrome
  - HttpFox
  - Fiddler
  - Burp Suite



# Tamper Data

Tamper Data - Ongoing requests

Start Tamper Stop Tamper Clear Options Help

Filter Show All

Time	Size	Method	Status	Content Type	URL
14:43:48.010	1150	GET	200	image/x-icon	http://www.leviaducdemillau.com/favicon.ico
14:43:52.757	10854	GET	200	application/x-shockwave-flash	http://www.leviaducdemillau.com/data/pages/en_page1_...
14:43:56.082	74463	GET	200	application/x-shockwave-flash	http://www.leviaducdemillau.com/data/modules/en_actu...
14:43:56.090	20800	GET	200	application/x-shockwave-flash	http://www.leviaducdemillau.com/data/modules/en_tele...
14:43:56.091	1049927	GET	200	application/x-shockwave-flash	http://www.leviaducdemillau.com/data/modules/slides_i...
14:44:00.657	14577	GET	200	application/xml	http://www.leviaducdemillau.com/actus.xml
14:44:13.348	unknown	GET	pending	unknown	http://www.leviaducdemillau.com/actus.xml

Request Header Name Request Header Value

Host	www.leviaducdemillau.com
User-Agent	Mozilla/5.0 (Windows NT 6.1; WOW64; rv:10.0.2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/26.0.1410.63 Safari/537.36
Accept	text/html,application/xhtml+xml,application/xml,application/javascript,application/json
Accept-Language	en-us,en;q=0.5
Accept-Encoding	gzip, deflate
DNT	1
Connection	keep-alive
Referer	http://www.leviaducdemillau.com/en/

Response Header Name Response Header Value

Status	OK - 200
Date	Sat, 24 Mar 2012 09:1
Server	Apache/1.3.41 (Unix)
Last-Modified	Wed, 11 Jan 2012 20:1
Etag	"2febd7-36f6-4f0dee-0000000000000000"
Accept-Ranges	bytes
Content-Length	14070
Keep-Alive	timeout=15, max=99
Connection	Keep-Alive
Content-Type	application/x-shockwave-flash

Tamper with request?

http://derekallard.com/about/

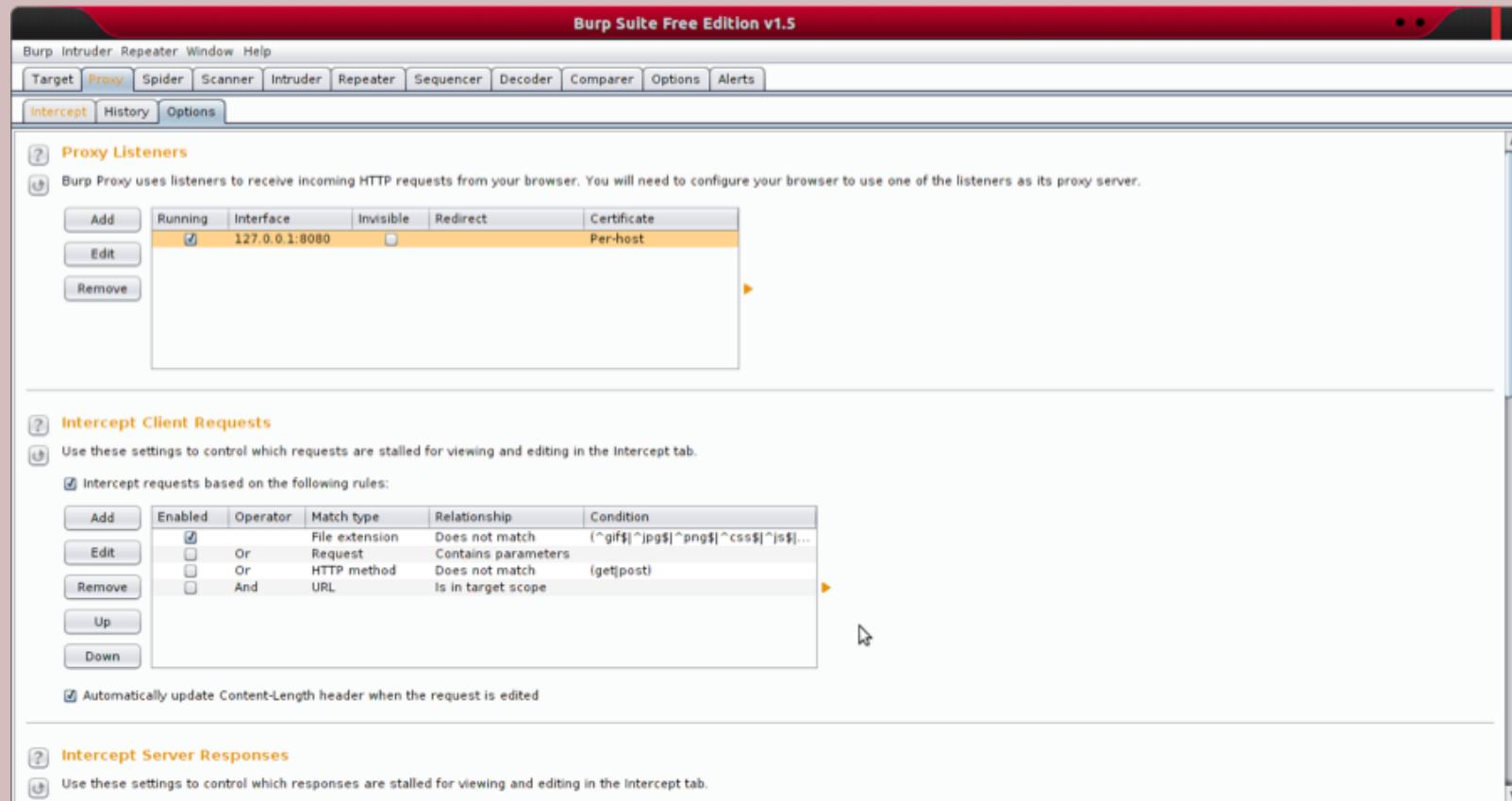
Continue Tampering?

Tamper Submit Abort Request



# Burp Suite

- Automated and Manual modes



# Outsource Your Scanning

- Even the best vulnerability scanners will miss business logic issues and other flaws
- Outsourcing to a security firm can identify issues that a web application scanner can't
- These firms can provide both static and dynamic analysis of your applications

