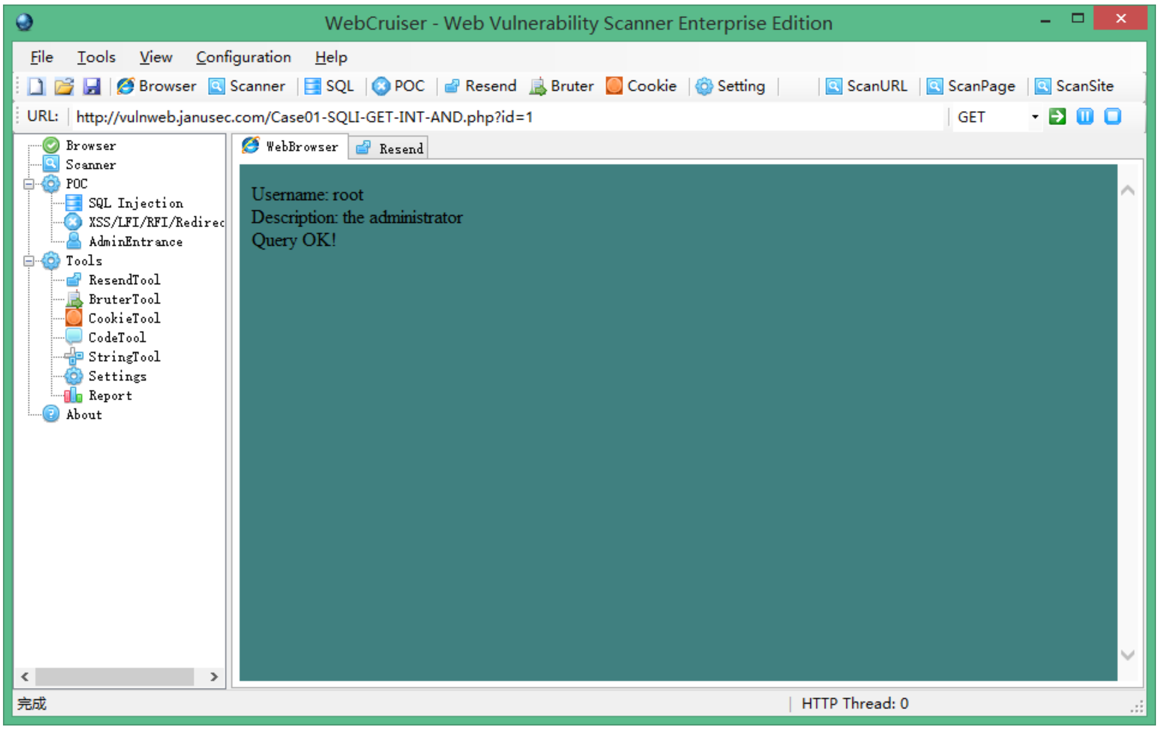
WebCruiser Web Vulnerability Scanner User Guide

Content

V3.1.0 by Janusec

http://www.janusec.com

1. Software Introduction



WebCruiser - Web Vulnerability Scanner Enterprise Edition
File Tools View Configuration Help
Browser Scanner SQL POC Resend Bruter Cookie Setting ScanURL ScanPage ScanSite
URL: http://vulnweb.janusec.com/Case01-SQLI-GET-INT-AND.php?id=1
GET
Browser
Scanner
POC
SQL Injection
XSS/LFI/RFI/Redirec
AdminEntrance
Tools
ResendTool
BruterTool
CookieTool
CodeTool
StringTool
Settings
Report
About
WebBrowser
Resend
完成
HTTP Thread: 0

Username: root
Description: the administrator
Query OK!

WebCruiser - Web Vulnerability Scanner, a compact but powerful web security scanning tool! It has a Crawler and a Vulnerability Scanner (SQL Injection, Cross Site Scripting, XPath Injection etc.).

It can support scanning website as well as POC (Proof of concept) for web vulnerabilities: SQL Injection, Cross Site Scripting, Local File Inclusion, Remote File Inclusion, Redirect etc.

The most typical feature of WebCruiser comparing with other Web Vulnerability Scanners is that WebCruiser Web Vulnerability Scanner focuses on high risk vulnerabilities, and WebCruiser can scan a designated vulnerability type, or a designated URL, or a designated page separately, while the others usually will not.

Key Features:

* Crawler (Site Directories and Files).

* Vulnerability Scanner: SQL Injection, Cross Site Scripting, LFI, RFI, Redirect

etc.

* WAVSEP v1.5 SQL Injection & XSS test cases 100% covered.

* SQL Injection POC Tool: GET/Post/Cookie Injection POC (Proof of Concept).

* SQL Injection for SQL Server: PlainText/Union/Blind Injection.

* SQL Injection for MySQL: PlainText/Union/Blind Injection.

* SQL Injection for Oracle: PlainText/Union/Blind/CrossSite Injection.

* SQL Injection for DB2: Union/Blind Injection.

* SQL Injection for Access: Union/Blind Injection.

* POC Tool for XSS, LFI, RFI, Redirect etc.

* Resend Tool.

* Bruter Tool.

* Cookie Tool.

Requirement: IE8+Requirement: .NET Framework 2.0+, IE8+

Software Disclaimer:

* Authorization must be obtained from the web application owner;

* This program will try to get each link and post any data when scanning;

* Backup the database before scanning so as to avoid disaster.

* Using this software at your own risk.

2. Main Function

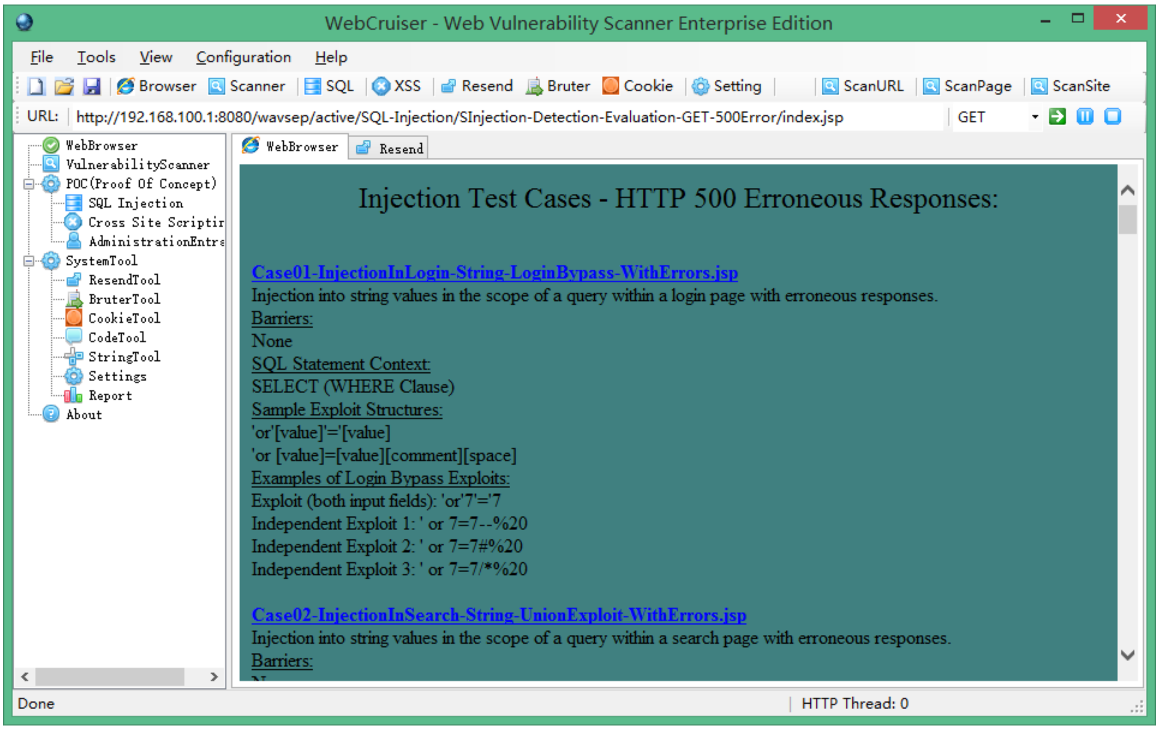
2.1. Web Vulnerability Scanner

WebCruiser - Web Vulnerability Scanner provides 3 kinds of scanning mode:

✧ ScanURL: Scan current URL only.

✧ ScanPage: Scan current page and all links within it, links under other directories will be skipped.

✧ ScanSite: Scan the whole site with the same domain.

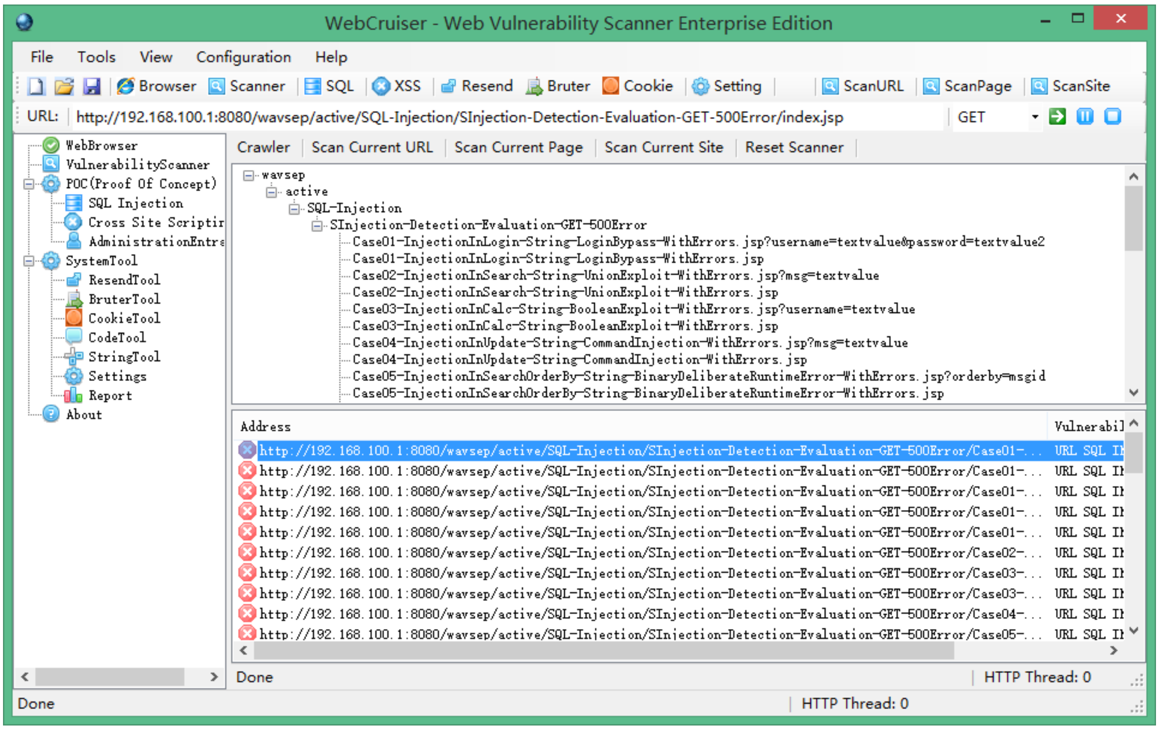

WebCruiser - Web Vulnerability Scanner Enterprise Edition
File Tools View Configuration Help
Browser Scanner SQL XSS Resend Bruter Cookie Setting ScanURL ScanPage ScanSite
URL: http://192.168.100.1:8080/wavsep/active/SQL-Injection/SInjection-Detection-Evaluation-GET-500Error/index.jsp
GET
WebBrowser
VulnerabilityScanner
POC(Proof Of Concept)
SQL Injection
Cross Site Scriptir
AdministrationEntra
SystemTool
ResendTool
BruterTool
CookieTool
CodeTool
StringTool
Settings
Report
About
WebBrowser
Resend
Done
HTTP Thread: 0

Injection Test Cases - HTTP 500 Erroneous Responses:

Case01-InjectionInLogin-String-LoginBypass-WithErrors.jsp
Injection into string values in the scope of a query within a login page with erroneous responses.
Barriers:
None
SQL Statement Context:
SELECT (WHERE Clause)
Sample Exploit Structures:
'or'[value]'='[value]
'or [value]=[value][comment][space]
Examples of Login Bypass Exploits:
Exploit (both input fields): 'or'7'='7
Independent Exploit 1: ' or 7=7--%20
Independent Exploit 2: ' or 7=7#%20
Independent Exploit 3: ' or 7=7/*%20

Case02-InjectionInSearch-String-UnionExploit-WithErrors.jsp
Injection into string values in the scope of a query within a search page with erroneous responses.
Barriers:

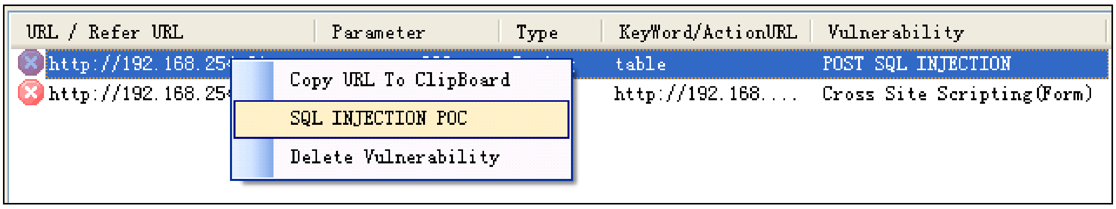
Scan Result (Above is Site Structure, and the following table is vulnerabilities):



```
wavsep
  active
    SQL-Injection
      SInjection-Detection-Evaluation-GET-500Error
        Case01-InjectionInLogin-String-LoginBypass-WithErrors.jsp?username=textvalue&password=textvalue2
        Case01-InjectionInLogin-String-LoginBypass-WithErrors.jsp
        Case02-InjectionInSearch-String-UnionExploit-WithErrors.jsp?msg=textvalue
        Case02-InjectionInSearch-String-UnionExploit-WithErrors.jsp
        Case03-InjectionInCalc-String-BooleanExploit-WithErrors.jsp?username=textvalue
        Case03-InjectionInCalc-String-BooleanExploit-WithErrors.jsp
        Case04-InjectionInUpdate-String-CommandInjection-WithErrors.jsp?msg=textvalue
        Case04-InjectionInUpdate-String-CommandInjection-WithErrors.jsp
        Case05-InjectionInSearchOrderBy-String-BinaryDeliberateRuntimeError-WithErrors.jsp?orderby=msgid
        Case05-InjectionInSearchOrderBy-String-BinaryDeliberateRuntimeError-WithErrors.jsp
```

Address	Vulnerabil
http://192.168.100.1:8080/wavsep/active/SQL-Injection/SInjection-Detection-Evaluation-GET-500Error/Case01-...	URL SQL IN
http://192.168.100.1:8080/wavsep/active/SQL-Injection/SInjection-Detection-Evaluation-GET-500Error/Case01-...	URL SQL IN
http://192.168.100.1:8080/wavsep/active/SQL-Injection/SInjection-Detection-Evaluation-GET-500Error/Case01-...	URL SQL IN
http://192.168.100.1:8080/wavsep/active/SQL-Injection/SInjection-Detection-Evaluation-GET-500Error/Case01-...	URL SQL IN
http://192.168.100.1:8080/wavsep/active/SQL-Injection/SInjection-Detection-Evaluation-GET-500Error/Case01-...	URL SQL IN
http://192.168.100.1:8080/wavsep/active/SQL-Injection/SInjection-Detection-Evaluation-GET-500Error/Case02-...	URL SQL IN
http://192.168.100.1:8080/wavsep/active/SQL-Injection/SInjection-Detection-Evaluation-GET-500Error/Case03-...	URL SQL IN
http://192.168.100.1:8080/wavsep/active/SQL-Injection/SInjection-Detection-Evaluation-GET-500Error/Case03-...	URL SQL IN
http://192.168.100.1:8080/wavsep/active/SQL-Injection/SInjection-Detection-Evaluation-GET-500Error/Case04-...	URL SQL IN
http://192.168.100.1:8080/wavsep/active/SQL-Injection/SInjection-Detection-Evaluation-GET-500Error/Case05-...	URL SQL IN

4. Right click vulnerability, and then you can launch SQL Injection or Cross Site Scripting POC (Proof of Concept):



2.2. SQL Injection Tool

Scanning is not necessary for SQL Injection POC, you can launch POC by input the URL directly, or launch from the Scanner.

WebCruiser support:

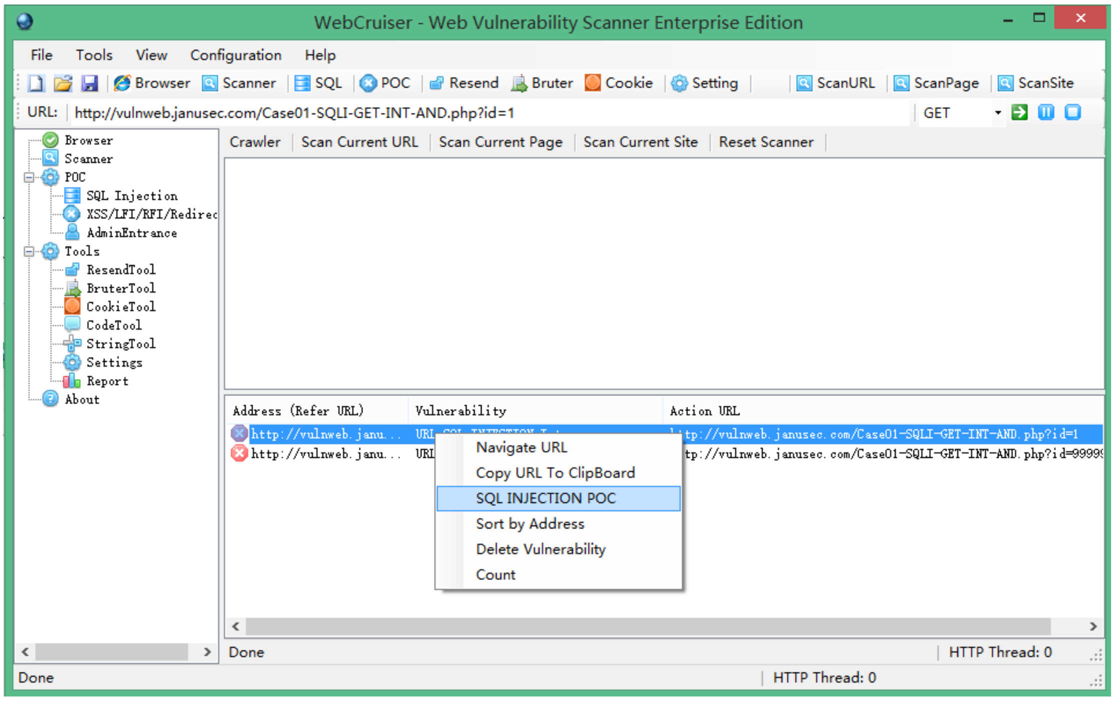
* GET/Post/Cookie Injection;

* SQL Server: Plaintext/FieldEcho(Union)/Blind Injection;

* MySQL/DB2/Access: FieldEcho(Union)/Blind Injection;

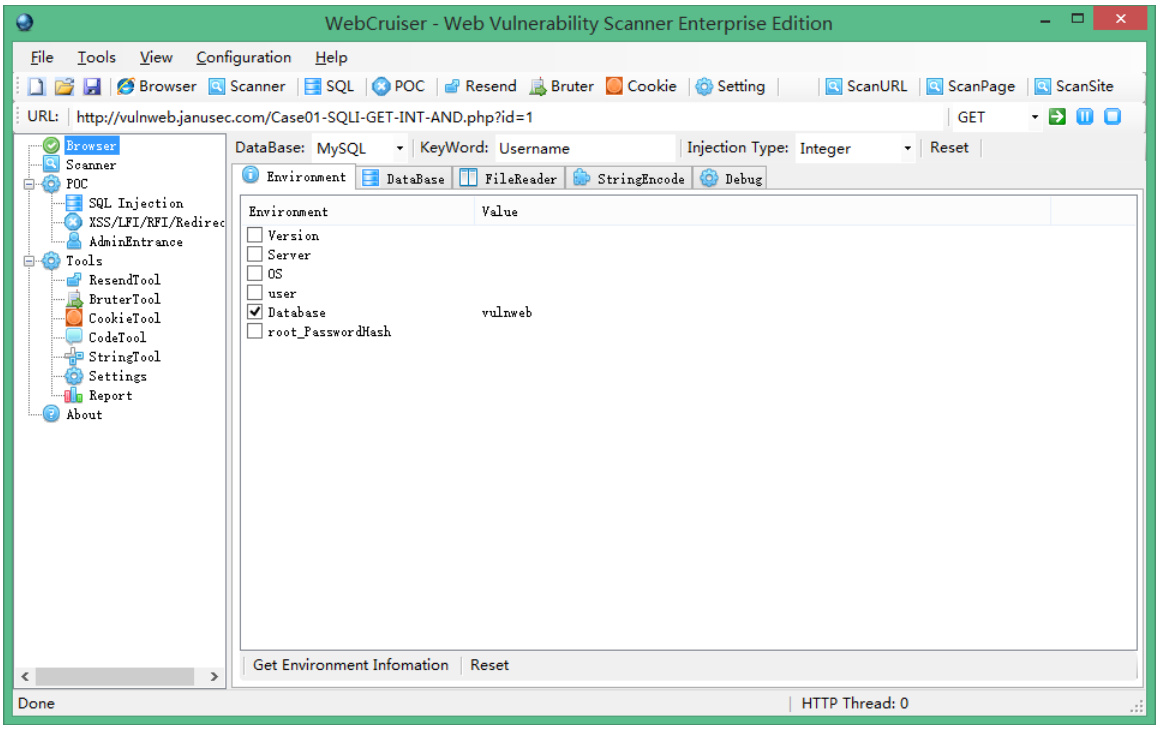
* Oracle: FieldEcho(Union)/Blind/CrossSite Injection;

Right click vulnerability, and select SQL Injection POC.



It will launch the SQL Injection POC tool and fill the relevant information.

This is a SQL Injection Demo.


WebCruiser - Web Vulnerability Scanner Enterprise Edition
File Tools View Configuration Help
Browser Scanner SQL POC Resend Bruter Cookie Setting ScanURL ScanPage ScanSite
Browser
Scanner
POC
SQL Injection
XSS/LFI/RFI/Redirec
AdminEntrance
Tools
ResendTool
BruterTool
CookieTool
CodeTool
StringTool
Settings
Report
About
Environment DataBase FileReader StringEncode Debug
Get Environment Infomation
Reset
Done
HTTP Thread: 0

URL: http://vulnweb.janusec.com/Case01-SQLI-GET-INT-AND.php?id=1 GET

DataBase: MySQL KeyWord: Username Injection Type: Integer Reset

Environment	Value
□ Version	
□ Server	
□ OS	
□ user	
☑ Database	vulnweb
□ root_PasswordHash	

Tips: Scan log is off by default. If you need the detailed log, open Registry:

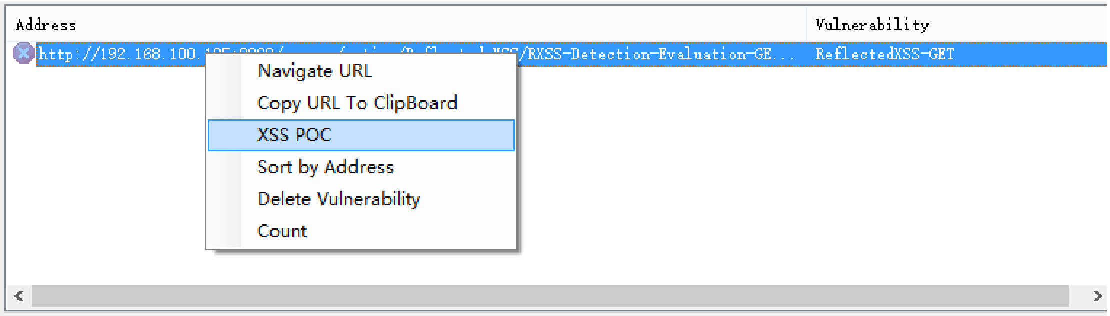
HKCU\Software\Sec4App\WebCruiser

Add a new String value: Edition, and set data to "Debug", then restarts WebCruiser. It will create log file like WebCruiseryyyymmdd.log under the same directory.

Edition	REG_SZ	Debug

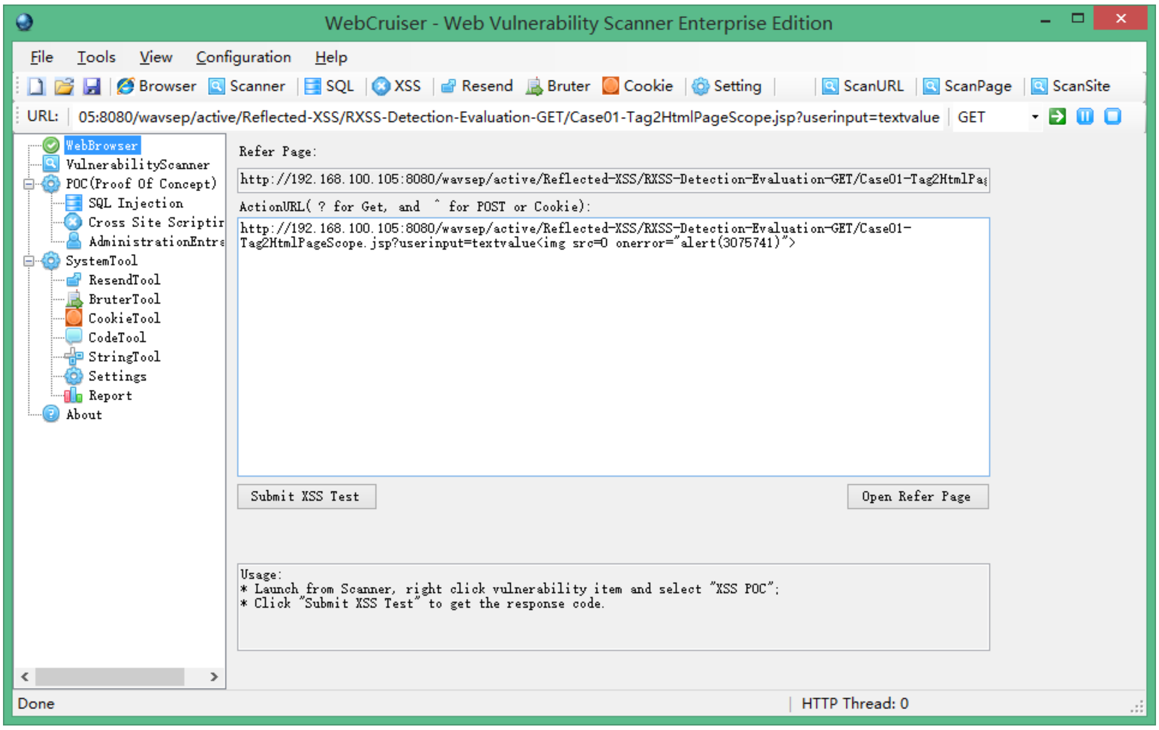
2.3. Cross Site Scripting

1. Right Click Vulnerability in Scanner and selects "XSS POC":



Address	Vulnerability
http://192.168.100. /RXSS-Detection-Evaluation-GE...	ReflectedXSS-GET

2. Click "Submit XSS Test".



URL: 05:8080/wavsep/active/Reflected-XSS/RXSS-Detection-Evaluation-GET/Case01-Tag2HtmlPageScope.jsp?userinput=textvalue GET

Refer Page:

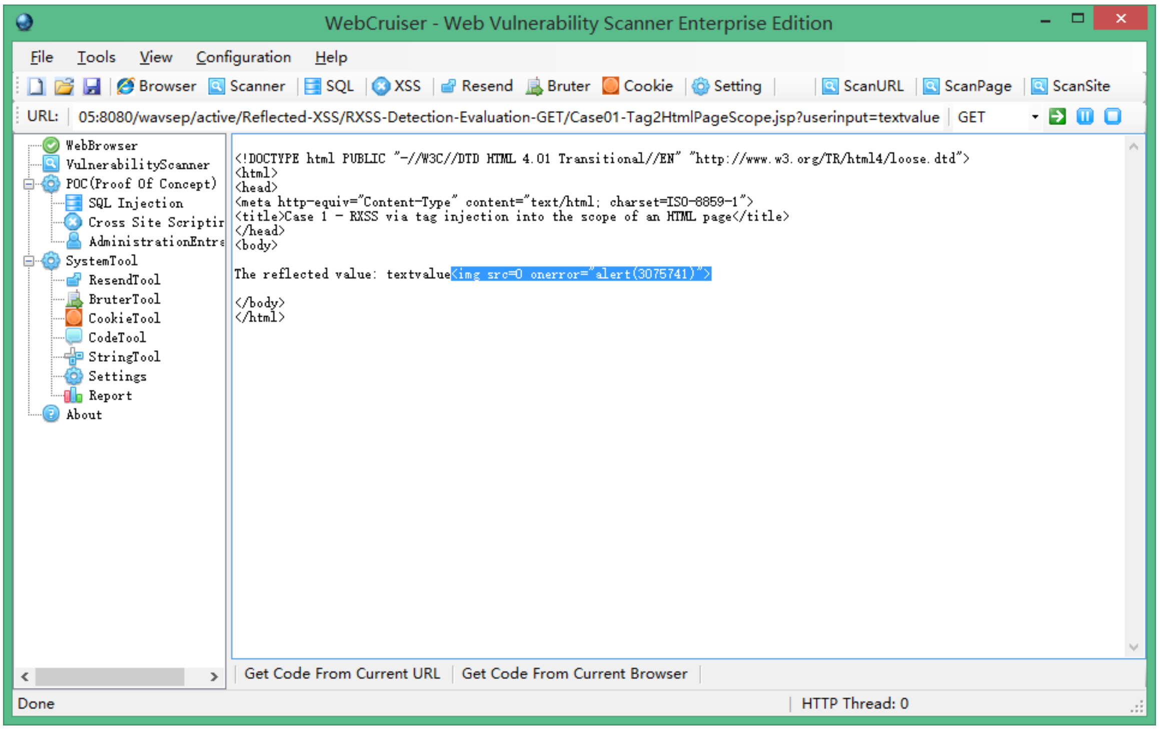
```
http://192.168.100.105:8080/wavsep/active/Reflected-XSS/RXSS-Detection-Evaluation-GET/Case01-Tag2HtmlPag
```

ActionURL(? for Get, and ^ for POST or Cookie):

```
http://192.168.100.105:8080/wavsep/active/Reflected-XSS/RXSS-Detection-Evaluation-GET/Case01-
Tag2HtmlPageScope.jsp?userinput=textvalue<img src=0 onerror="alert(3075741)">
```

```
Usage:
* Launch from Scanner, right click vulnerability item and select "XSS POC";
* Click "Submit XSS Test" to get the response code.
```

3. Usually your input will occur in the Response Code:



URL: 05:8080/wavsep/active/Reflected-XSS/RXSS-Detection-Evaluation-GET/Case01-Tag2HtmlPageScope.jsp?userinput=textvalue GET

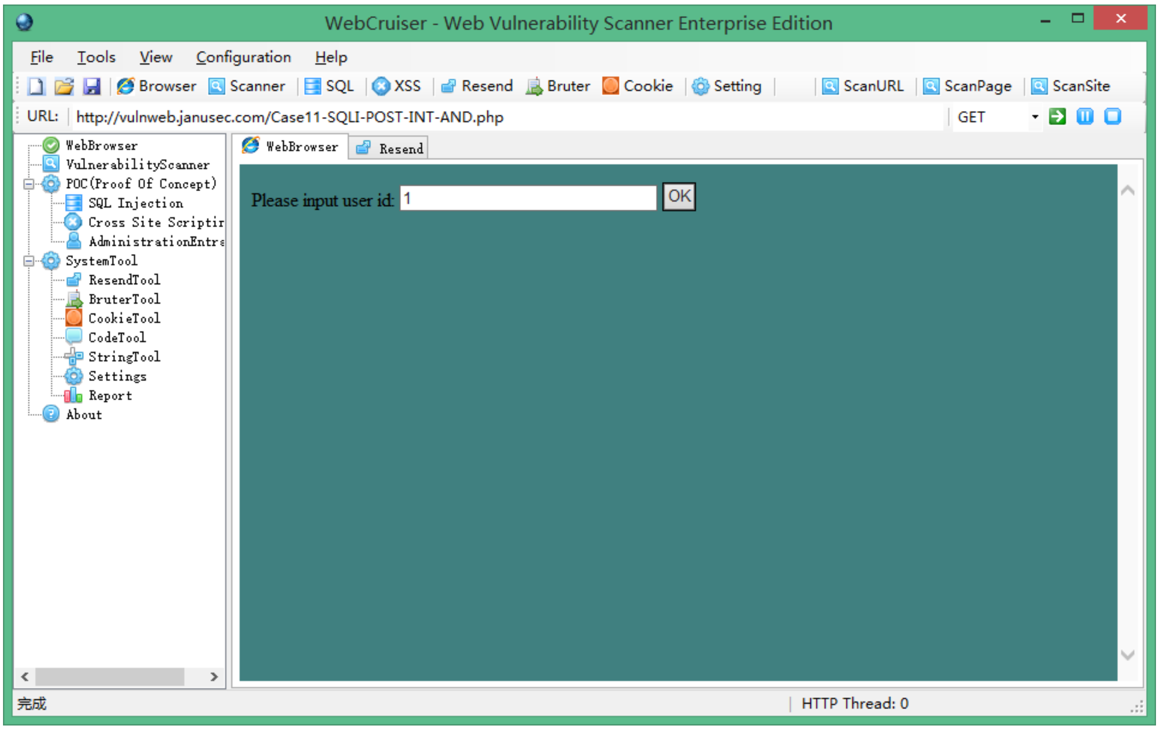
```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Case 1 - RXSS via tag injection into the scope of an HTML page</title>
</head>
<body>

The reflected value: textvalue<img src=0 onerror="alert(3075741)">

</body>
</html>
```

2.4. LFI/RFI/Redirect POC

LFI/RFI/Redirect POC is the same with XSS POC.

2.5. Resend Test Tool

When you Post any data, WebCruiser will capture the Post data automatically. First, let's login a demo application:

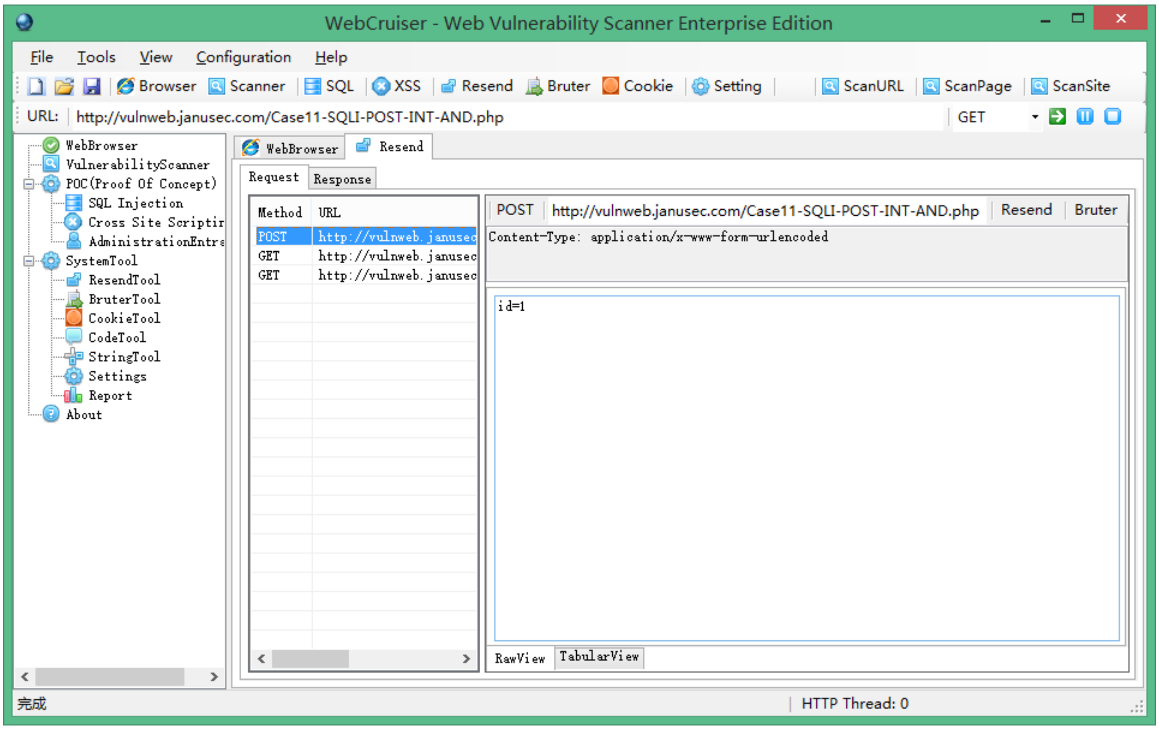


URL: http://vulnweb.janusec.com/Case11-SQLI-POST-INT-AND.php GET

Please input user id: 1 OK

完成 | HTTP Thread: 0

Switch to tab page "Resend", the Post data has been captured here:



URL: http://vulnweb.janusec.com/Case11-SQLI-POST-INT-AND.php GET

Method	URL
POST	http://vulnweb.janusec
GET	http://vulnweb.janusec
GET	http://vulnweb.janusec

POST http://vulnweb.janusec.com/Case11-SQLI-POST-INT-AND.php

```
Content-Type: application/x-www-form-urlencoded
```

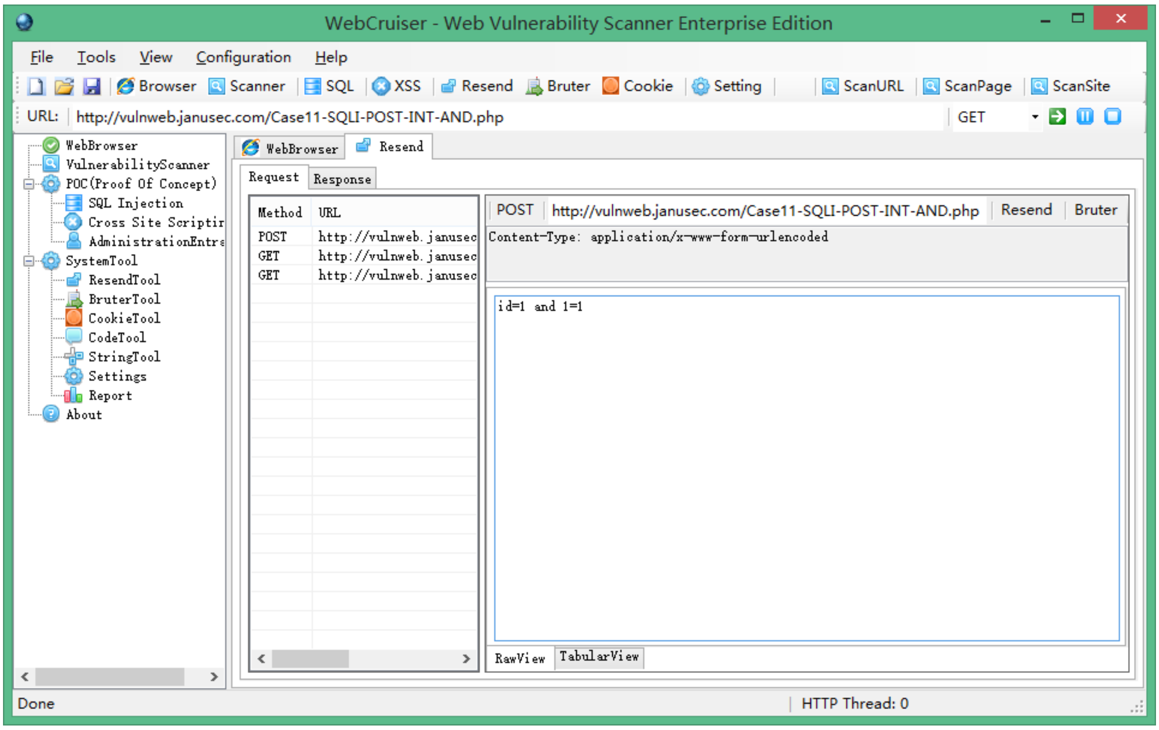
```
id=1
```

完成 | HTTP Thread: 0

Now, you can modify the post data and resend them.

Let's try to use it for SQL Injection:

First, modify the value of id to *1 and 1=1*



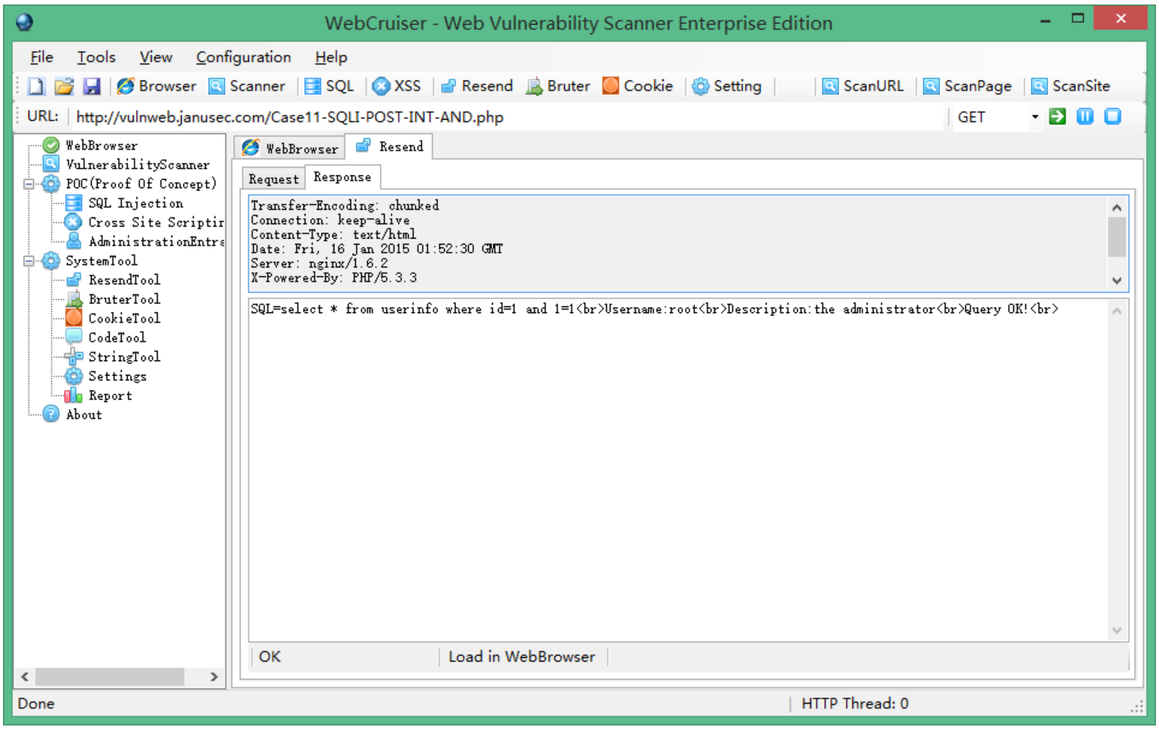
URL: http://vulnweb.janusec.com/Case11-SQLI-POST-INT-AND.php GET

```
Method  URL
POST    http://vulnweb.janusec
GET     http://vulnweb.janusec
GET     http://vulnweb.janusec
```

POST http://vulnweb.janusec.com/Case11-SQLI-POST-INT-AND.php

```
Content-Type: application/x-www-form-urlencoded
```

```
id=1 and 1=1
```



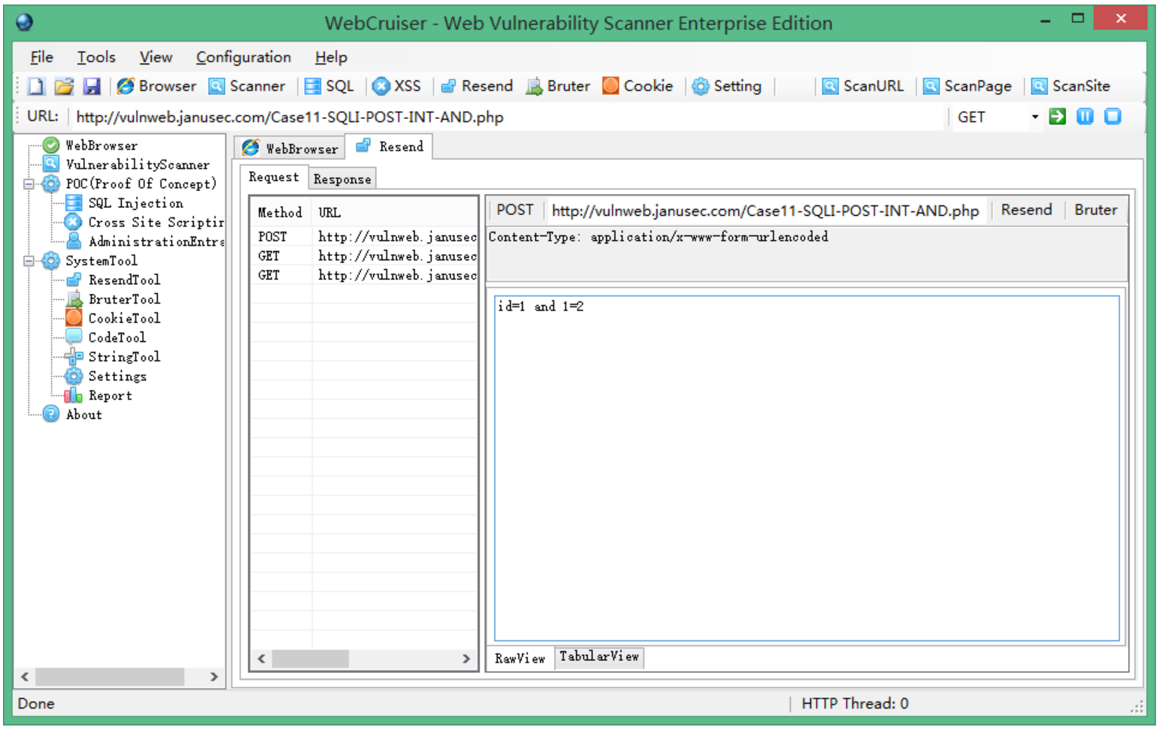
URL: http://vulnweb.janusec.com/Case11-SQLI-POST-INT-AND.php GET

```
Transfer-Encoding: chunked
Connection: keep-alive
Content-Type: text/html
Date: Fri, 16 Jan 2015 01:52:30 GMT
Server: nginx/1.6.2
X-Powered-By: PHP/5.3.3
```

```
SQL=select * from userinfo where id=1 and 1=1<br>Username:root<br>Description:the administrator<br>Query OK!<br>
```

OK

Second, modify the value of id to *1 and 1=2*



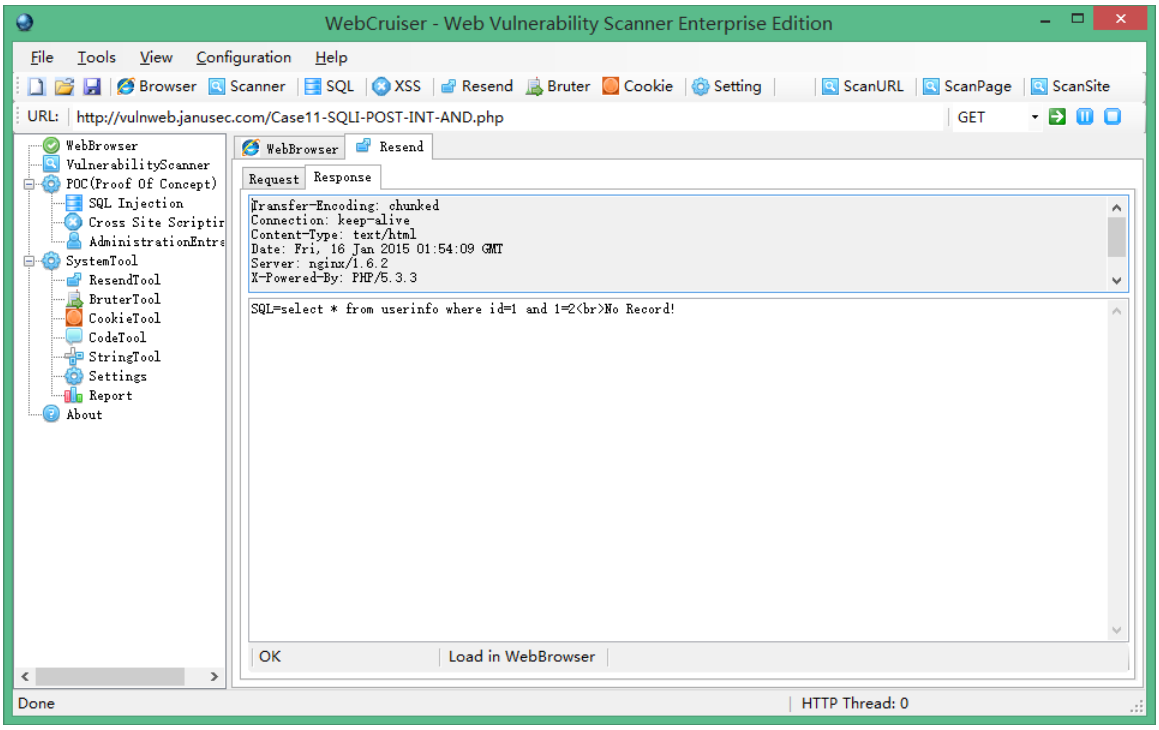
URL: http://vulnweb.janusec.com/Case11-SQLI-POST-INT-AND.php GET

```
Method  URL
POST    http://vulnweb.janusec
GET     http://vulnweb.janusec
GET     http://vulnweb.janusec
```

POST http://vulnweb.janusec.com/Case11-SQLI-POST-INT-AND.php

```
Content-Type: application/x-www-form-urlencoded
```

```
id=1 and 1=2
```



URL: http://vulnweb.janusec.com/Case11-SQLI-POST-INT-AND.php GET

```
Transfer-Encoding: chunked
Connection: keep-alive
Content-Type: text/html
Date: Fri, 16 Jan 2015 01:54:09 GMT
Server: nginx/1.6.2
X-Powered-By: PHP/5.3.3
```

```
SQL=select * from userinfo where id=1 and 1=2<br>No Record!
```

We got different response. It means that this application has a vulnerability of SQL Injection.

3. DVWA Demo

DVWA (Damn Vulnerable Web Application) V1.8 Test Demo with WebCruiser.

3.1. Environment

Environment:

OS:Windows 8.1 or Windows 7

Runtime: .Net Framework 3.5

PHP+MySQL:XAMPP V3.2.1

DVWA settings in config.inc.php:

$_DVWA['db_server'] = 'localhost';

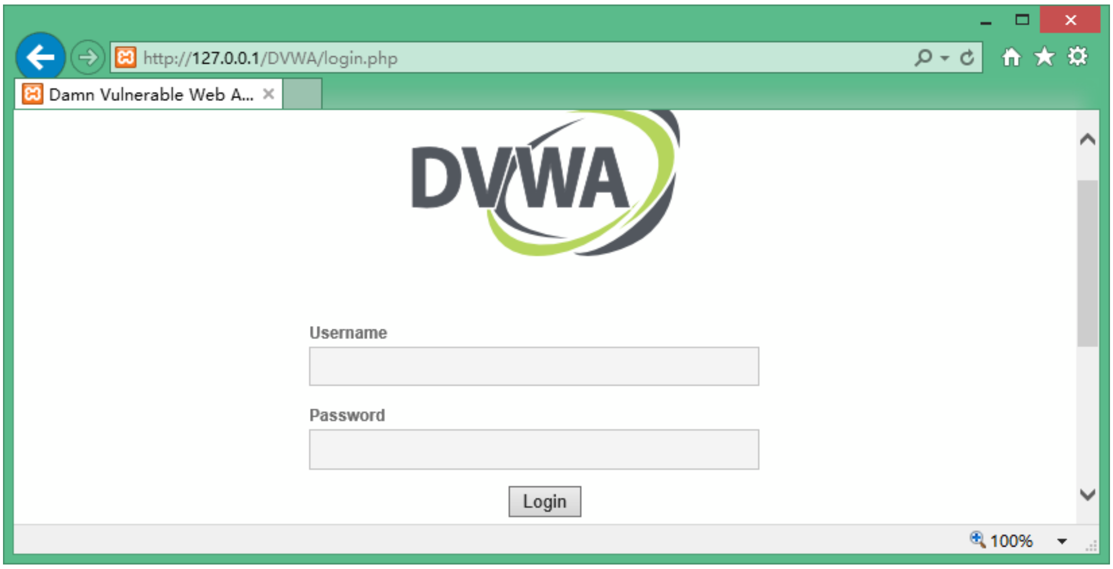
$_DVWA['db_database'] = 'dvwa';

$_DVWA['db_user'] = 'root';

$_DVWA['db_password'] = ‘123456’;

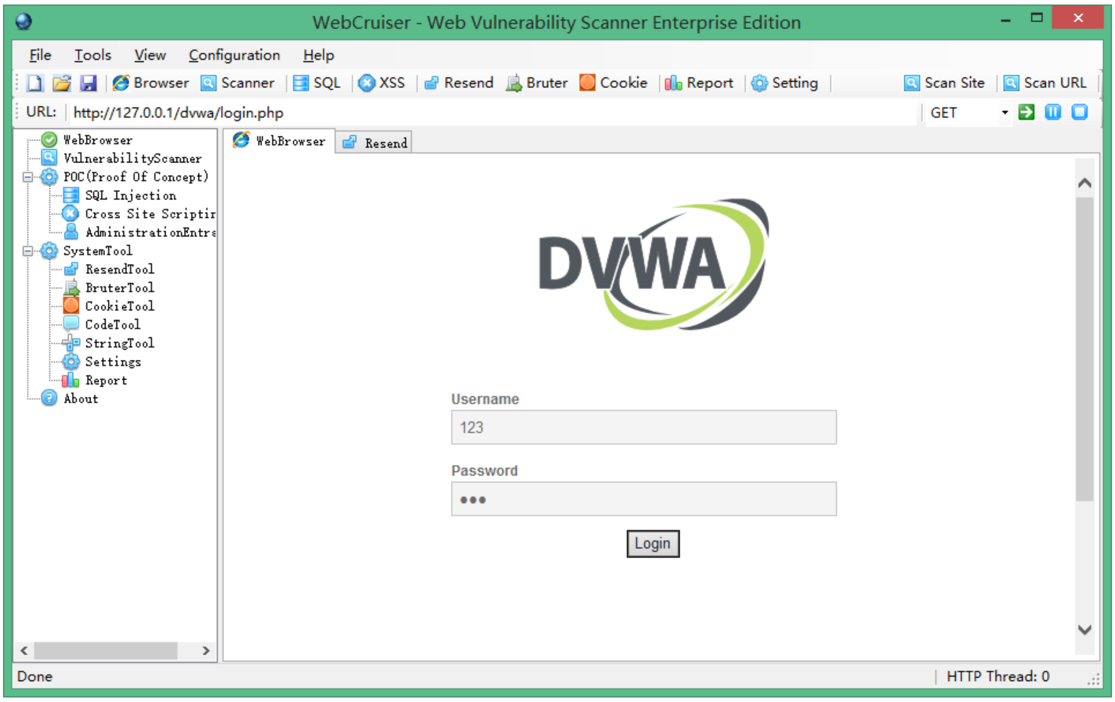
$_DVWA['default_security_level'] = "low";

Open http://127.0.0.1/DVWA/login.php



3.2. Brute Force

First, input any username and password which are wrong, here we input 123 and 456:

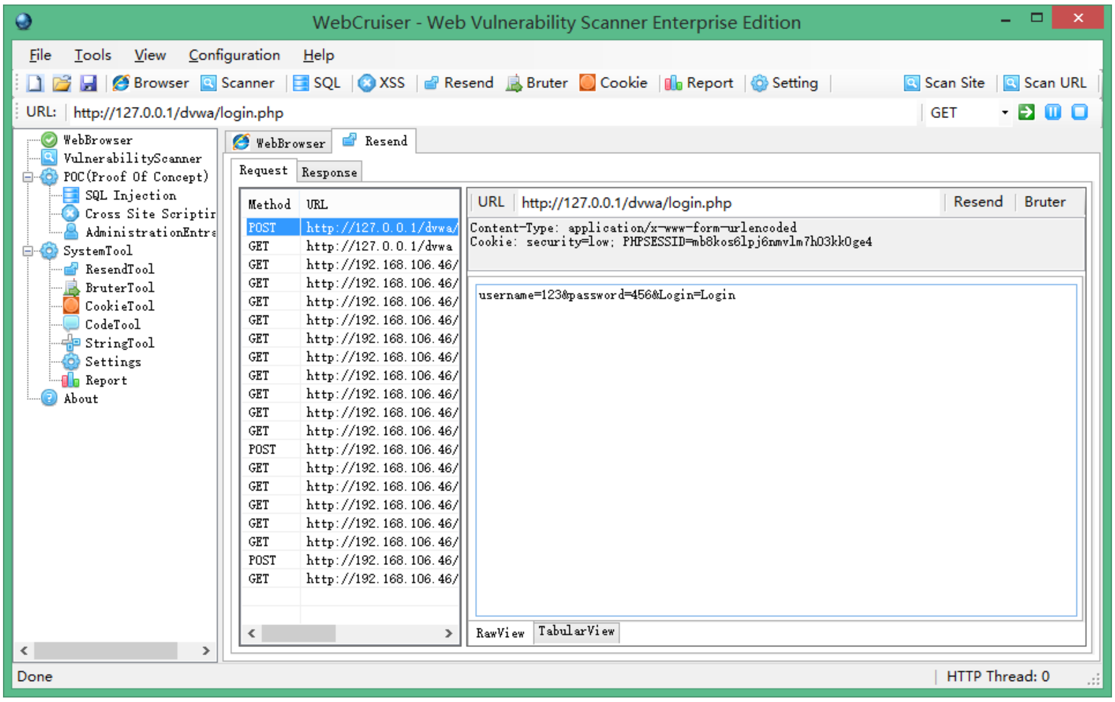


URL: http://127.0.0.1/dvwa/login.php GET

Username: 123

Password: ●●●

submit it and switch to the “Resend” tab.



URL: http://127.0.0.1/dvwa/login.php GET

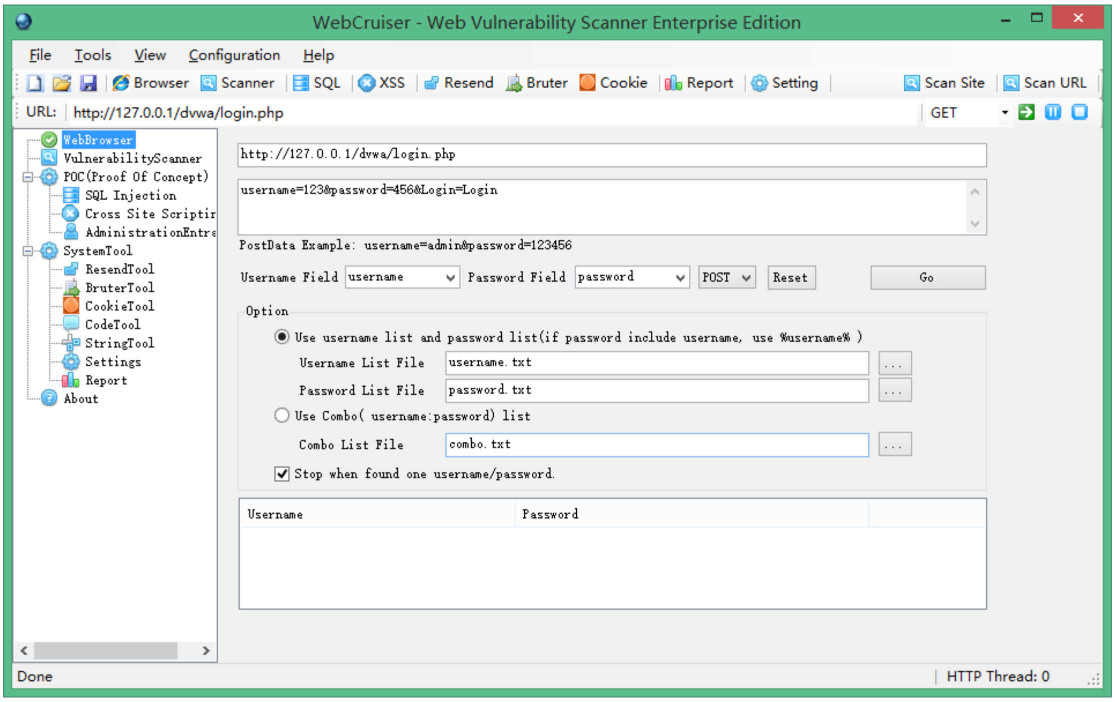
Method	URL
POST	http://127.0.0.1/dvwa/
GET	http://127.0.0.1/dvwa
GET	http://192.168.106.46/
GET	http://192.168.106.46/
GET	http://192.168.106.46/
GET	http://192.168.106.46/
GET	http://192.168.106.46/
GET	http://192.168.106.46/
GET	http://192.168.106.46/
GET	http://192.168.106.46/
GET	http://192.168.106.46/
GET	http://192.168.106.46/
POST	http://192.168.106.46/
GET	http://192.168.106.46/
GET	http://192.168.106.46/
GET	http://192.168.106.46/
GET	http://192.168.106.46/
GET	http://192.168.106.46/
POST	http://192.168.106.46/
GET	http://192.168.106.46/

URL http://127.0.0.1/dvwa/login.php

```
Content-Type: application/x-www-form-urlencoded
Cookie: security=low; PHPSESSID=mb8kos6lpj6nmvlm7h03kk0ge4
```

```
username=123&password=456&Login=Login
```

Here lists http requests, and the top one is the newest one, just click it and then the right panel will show the detailed message. Click “Bruter” to launch the bruter.



URL: http://127.0.0.1/dvwa/login.php GET

http://127.0.0.1/dvwa/login.php

username=123&password=456&Login=Login

PostData Example: username=admin&password=123456

Username Field: username Password Field: password POST

Option

● Use username list and password list(if password include username, use %username%)

Username List File: username.txt

Password List File: password.txt

○ Use Combo(username:password) list

Combo List File: combo.txt

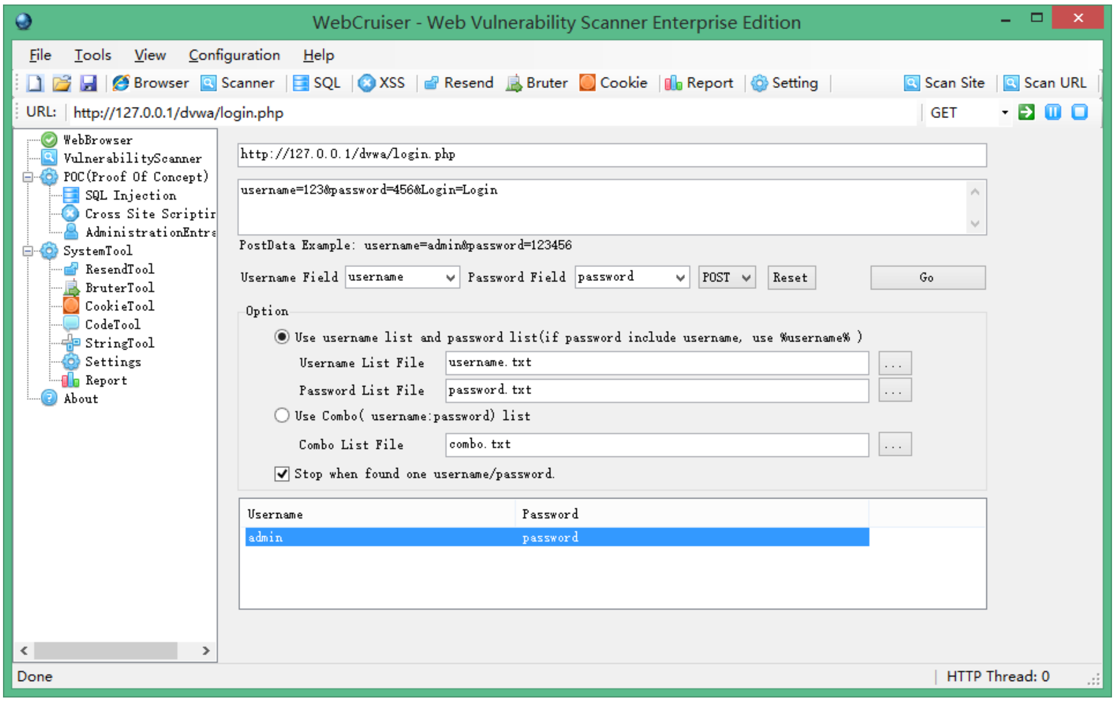
☑ Stop when found one username/password.

Username	Password

If the form uses “username” and “password” as the parameter name, they will be filled in the selectable fields automatically, if not, select them manually.

Bruter has two ways to get the username/password, one is using separate username list and password list, another is using combo list which is composed of username:password.

Click “Go” to launch it.



URL: http://127.0.0.1/dvwa/login.php GET

http://127.0.0.1/dvwa/login.php

username=123&password=456&Login=Login

PostData Example: username=admin&password=123456

Username Field username Password Field password POST

Option

● Use username list and password list(if password include username, use %username%)

Username List File username.txt

Password List File password.txt

○ Use Combo(username:password) list

Combo List File combo.txt

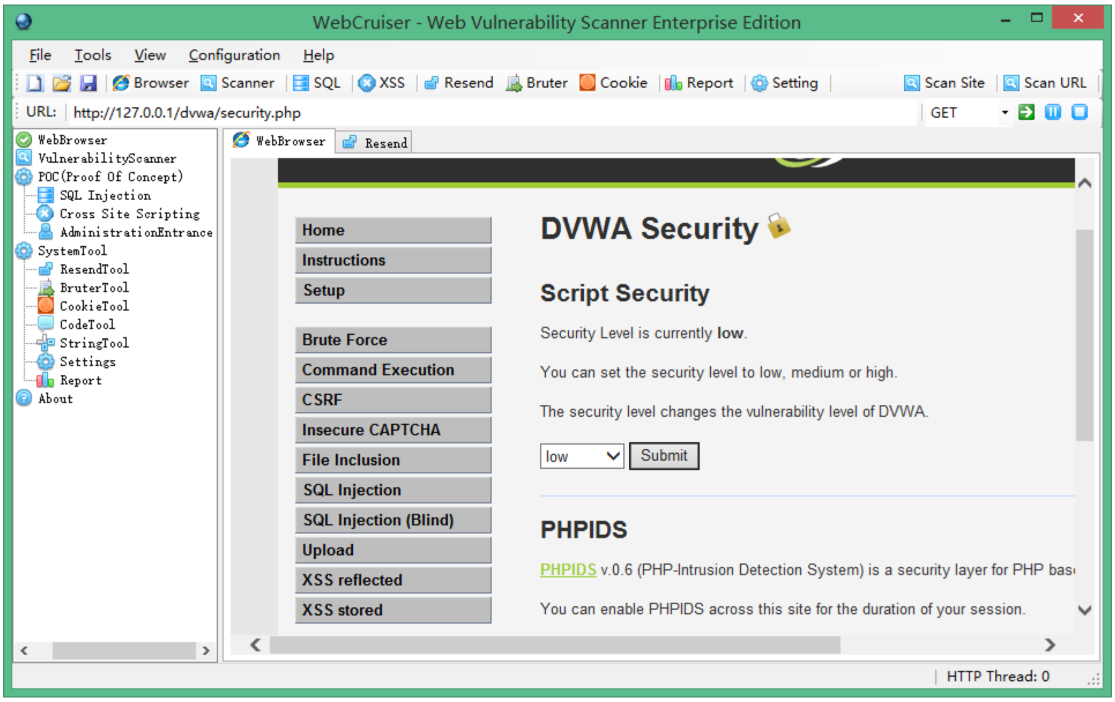
☑ Stop when found one username/password.

Username	Password
admin	password

HTTP Thread: 0

After a while, it found it: admin/password .

Switch to WebBrowser, input the username and password.



URL: http://127.0.0.1/dvwa/security.php GET

DVWA Security

Script Security

Security Level is currently **low**.

You can set the security level to low, medium or high.

The security level changes the vulnerability level of DVWA.

low

PHPIDS

PHPIDS v.0.6 (PHP-Intrusion Detection System) is a security layer for PHP basi

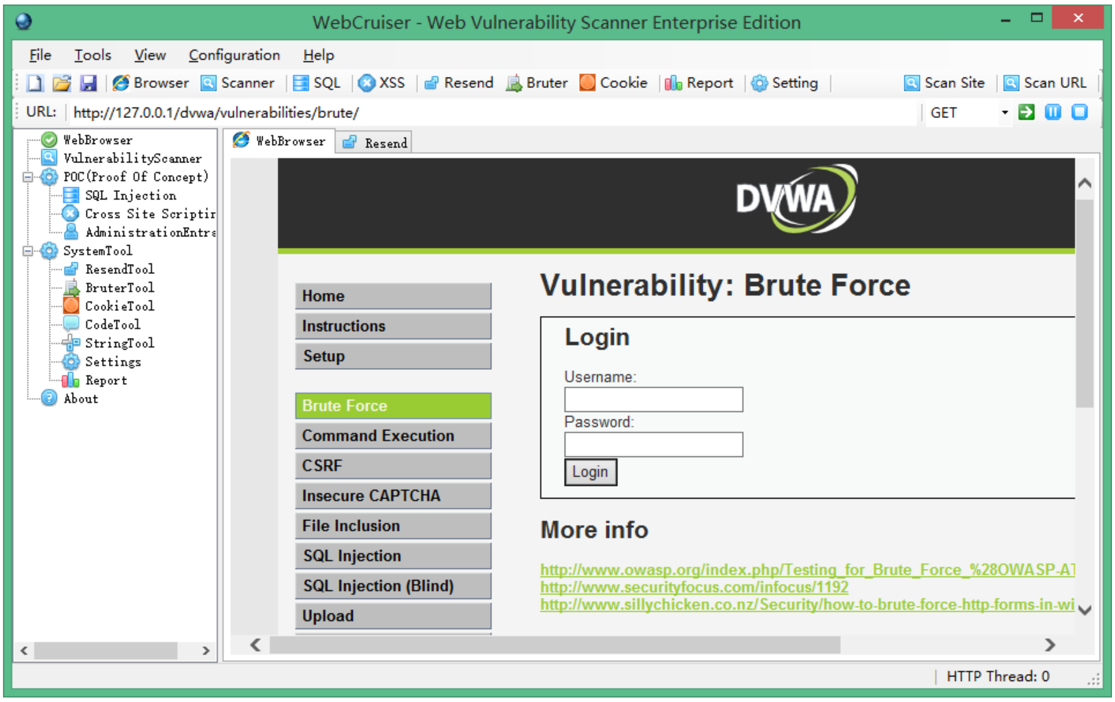
You can enable PHPIDS across this site for the duration of your session.

HTTP Thread: 0

After logged in, check the “DVWA Security” page, make sure the security level is low.

Now, begin the test.

The first test within the DVWA is another Brute Force.



URL: http://127.0.0.1/dvwa/vulnerabilities/brute/

Vulnerability: Brute Force

Login

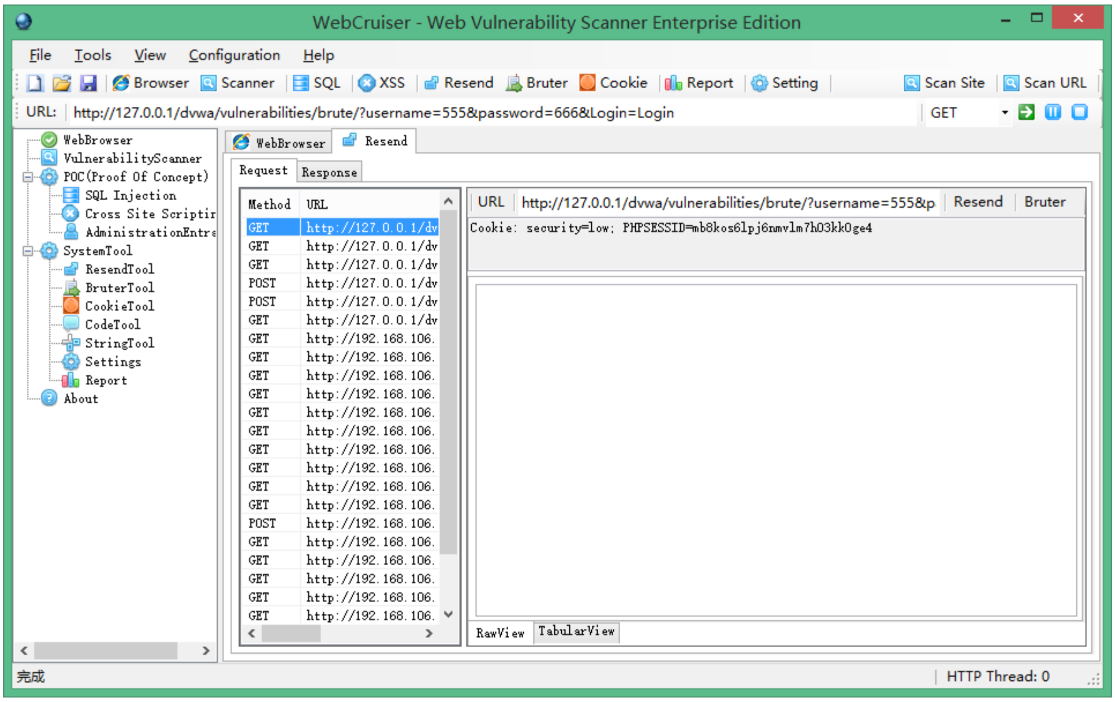
Username:

Password:

More info

http://www.owasp.org/index.php/Testing_for_Brute_Force_%28OWASP-AT
http://www.securityfocus.com/infocus/1192
http://www.sillychicken.co.nz/Security/how-to-brute-force-http-forms-in-wi

Like the login form, input any username and password, and then switch to "Resend":



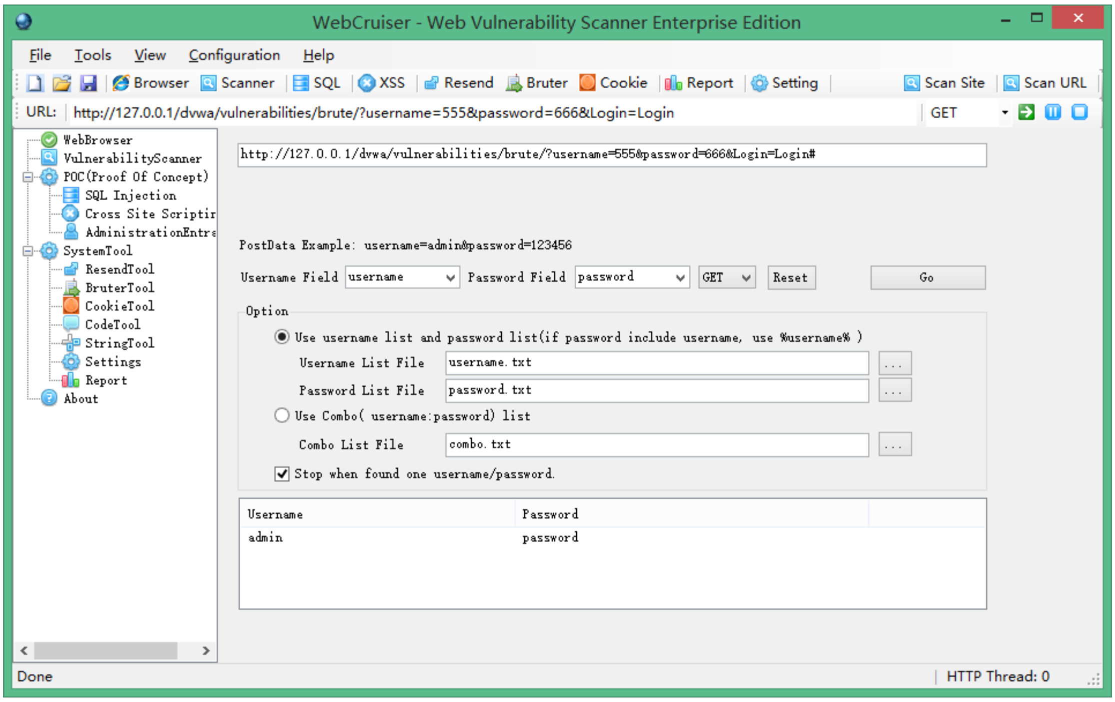
URL: http://127.0.0.1/dvwa/vulnerabilities/brute/?username=555&password=666&Login=Login

Method	URL
GET	http://127.0.0.1/dv
GET	http://127.0.0.1/dv
GET	http://127.0.0.1/dv
POST	http://127.0.0.1/dv
POST	http://127.0.0.1/dv
GET	http://127.0.0.1/dv
GET	http://192.168.106.
GET	http://192.168.106.
GET	http://192.168.106.
GET	http://192.168.106.
GET	http://192.168.106.
GET	http://192.168.106.
GET	http://192.168.106.
GET	http://192.168.106.
GET	http://192.168.106.
GET	http://192.168.106.
POST	http://192.168.106.
GET	http://192.168.106.
GET	http://192.168.106.
GET	http://192.168.106.
GET	http://192.168.106.
GET	http://192.168.106.

URL http://127.0.0.1/dvwa/vulnerabilities/brute/?username=555&p

```
Cookie: security=low; PHPSESSID=mb8kos6lpj6nmvlm7h03kk0ge4
```

Click "Bruter" to continue:



URL: http://127.0.0.1/dvwa/vulnerabilities/brute/?username=555&password=666&Login=Login GET

http://127.0.0.1/dvwa/vulnerabilities/brute/?username=555&password=666&Login=Login#

PostData Example: username=admin&password=123456

Username Field username Password Field password GET

Option

● Use username list and password list(if password include username, use %username%)

Username List File username.txt

Password List File password.txt

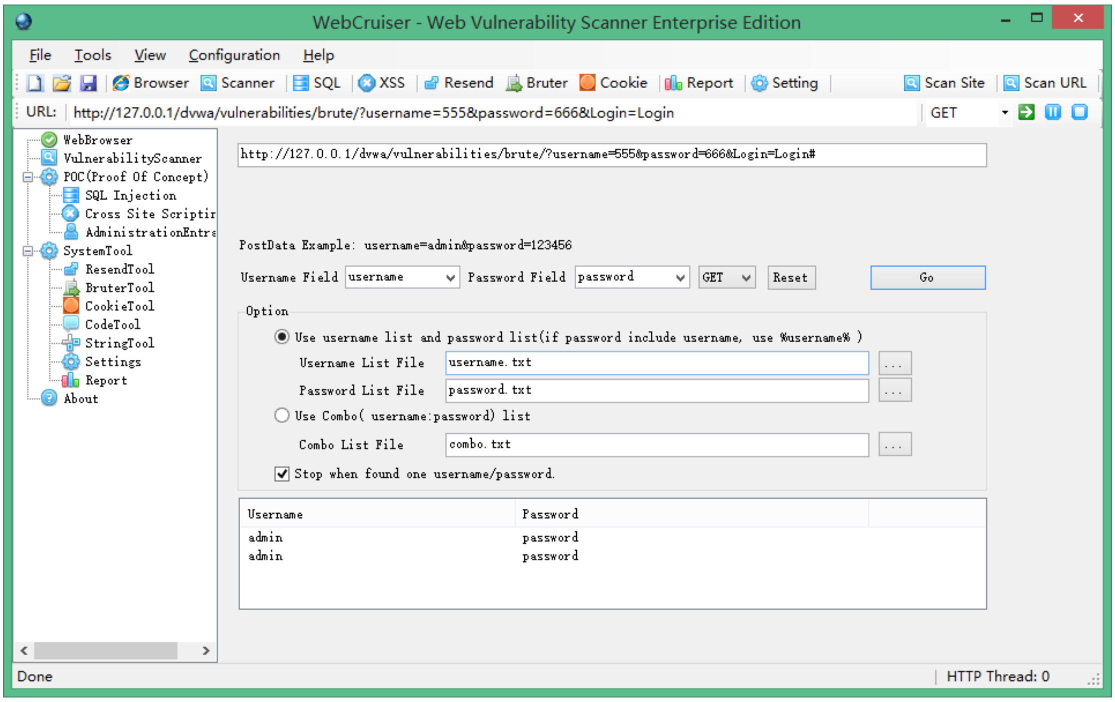
○ Use Combo(username:password) list

Combo List File combo.txt

☑ Stop when found one username/password.

Username	Password
admin	password

This is a GET request, click “Go”:



URL: http://127.0.0.1/dvwa/vulnerabilities/brute/?username=555&password=666&Login=Login GET

http://127.0.0.1/dvwa/vulnerabilities/brute/?username=555&password=666&Login=Login#

PostData Example: username=admin&password=123456

Username Field username Password Field password GET

Option

● Use username list and password list(if password include username, use %username%)

Username List File username.txt

Password List File password.txt

○ Use Combo(username:password) list

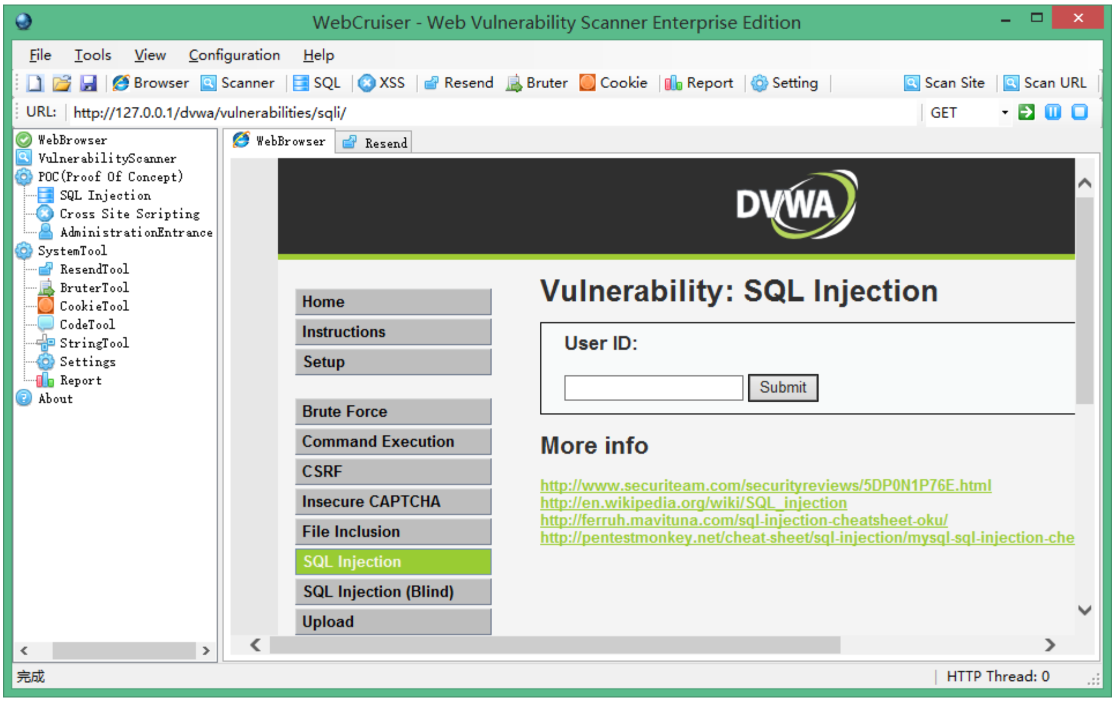
Combo List File combo.txt

☑ Stop when found one username/password.

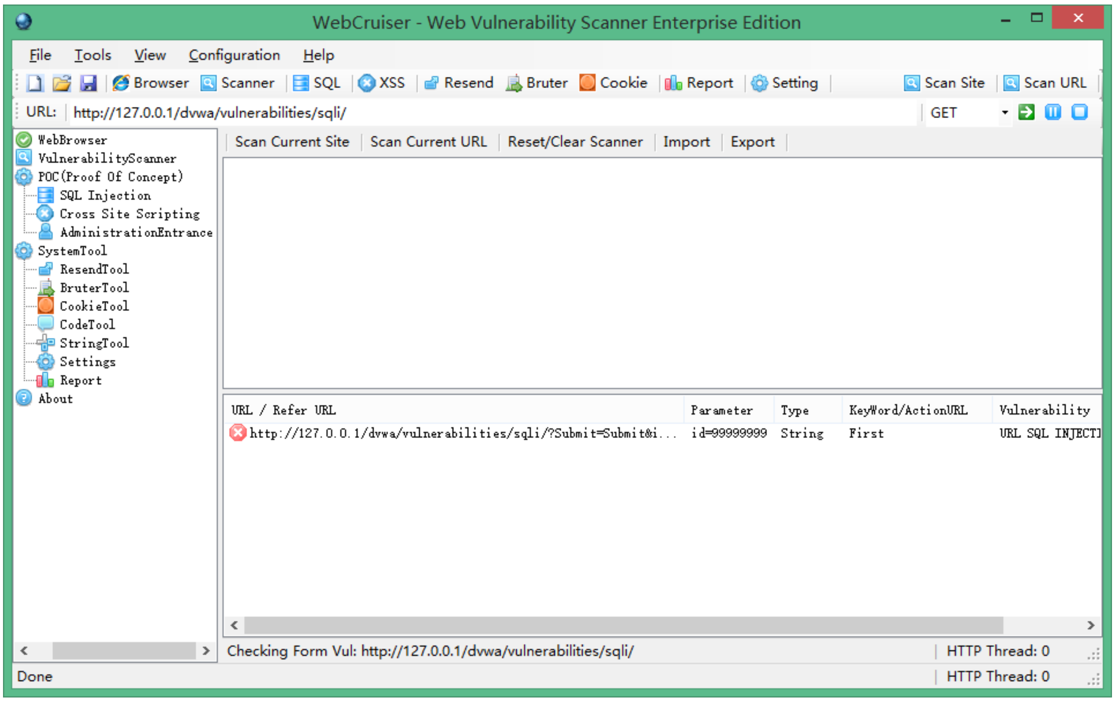
Username	Password
admin	password
admin	password

Found username/password: admin/password.

3.3. SQL Injection


WebCruiser - Web Vulnerability Scanner Enterprise Edition
File Tools View Configuration Help
Browser Scanner SQL XSS Resend Bruter Cookie Report Setting Scan Site Scan URL
URL: http://127.0.0.1/dvwa/vulnerabilities/sqli/
GET
WebBrowser
VulnerabilityScanner
POC(Proof Of Concept)
SQL Injection
Cross Site Scripting
AdministrationEntrance
SystemTool
ResendTool
BruterTool
CookieTool
CodeTool
StringTool
Settings
Report
About
WebBrowser
Resend
DVWA
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
Insecure CAPTCHA
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
Vulnerability: SQL Injection
User ID:
Submit
More info
http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/
http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-che
完成
HTTP Thread: 0

Click “Scan URL” at the top right menu:

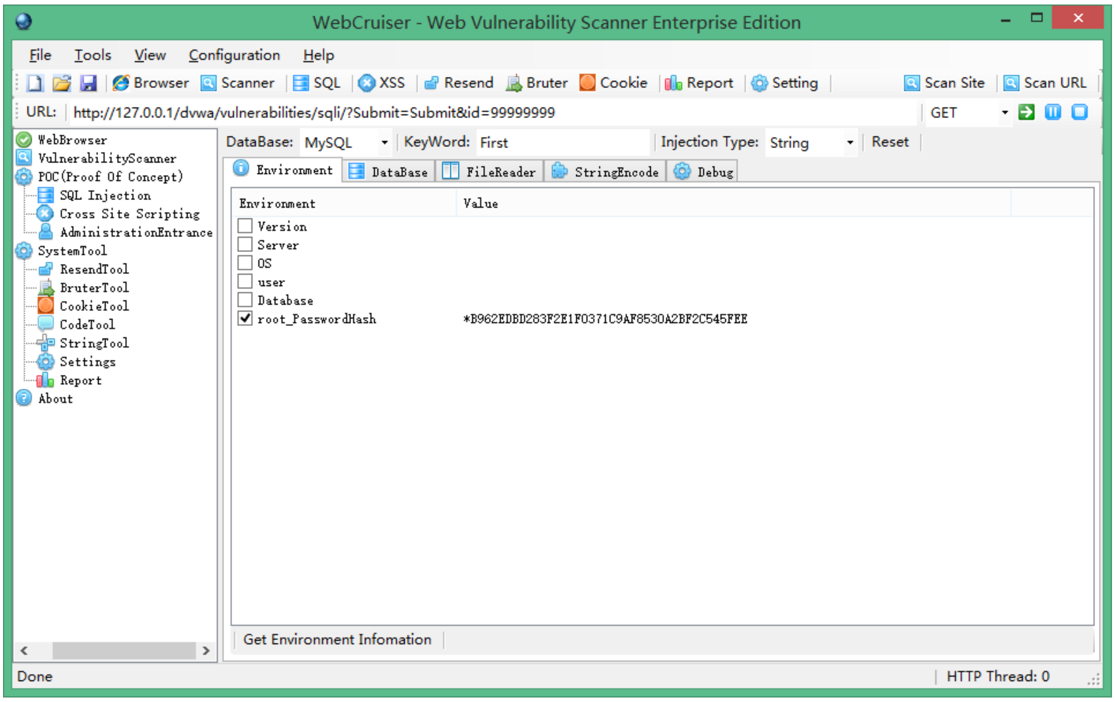


URL / Refer URL	Parameter	Type	KeyWord/ActionURL	Vulnerability
http://127.0.0.1/dvwa/vulnerabilities/sqli/?Submit=Submit&i...	id=99999999	String	First	URL SQL INJECTI

Checking Form Vul: http://127.0.0.1/dvwa/vulnerabilities/sqli/ HTTP Thread: 0

Done HTTP Thread: 0

Right click the vulnerability and select “SQL INJECTION POC”:



URL: http://127.0.0.1/dvwa/vulnerabilities/sqli/?Submit=Submit&id=99999999 GET

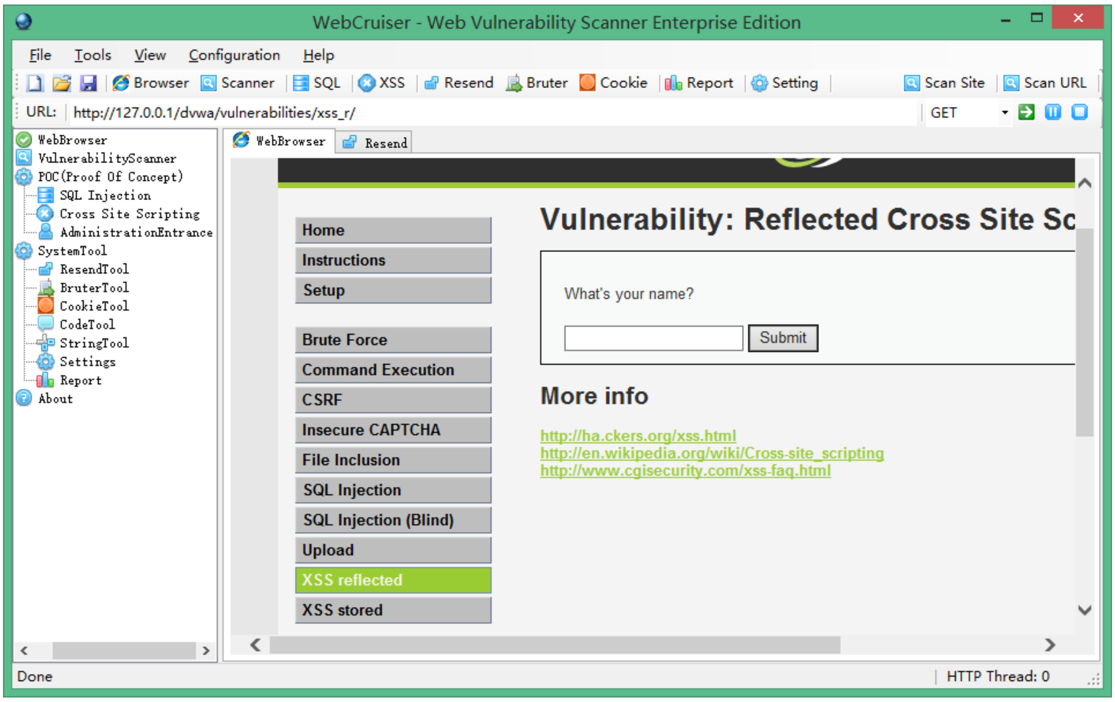
DataBase: MySQL | KeyWord: First | Injection Type: String

Environment	Value
□ Version	
□ Server	
□ OS	
□ user	
□ Database	
☑ root_PasswordHash	*B962EDBD283F2E1F0371C9AF8530A2BF2C545FEE

Oh, we got the encrypted password of root.

3.4. XSS

XSS Reflected, ”Scan URL”:



URL: http://127.0.0.1/dvwa/vulnerabilities/xss_r/ GET

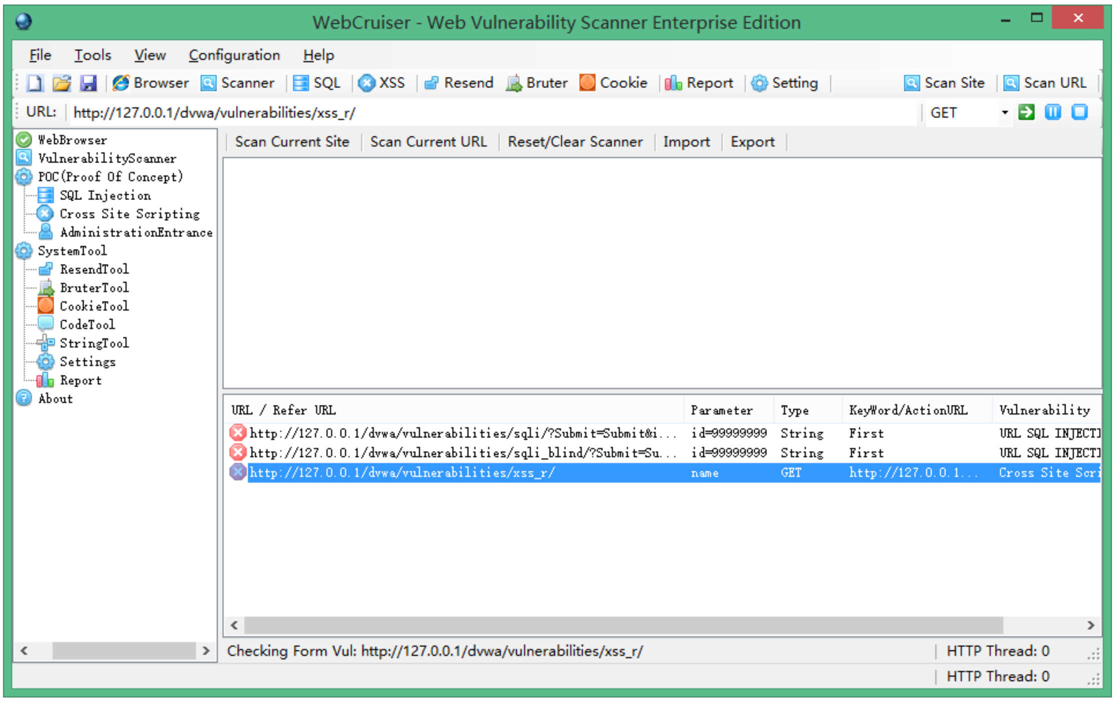
Vulnerability: Reflected Cross Site Sc

What's your name?

More info

http://ha.ckers.org/xss.html
http://en.wikipedia.org/wiki/Cross-site_scripting
http://www.cgisecurity.com/xss-faq.html

Found one XSS:



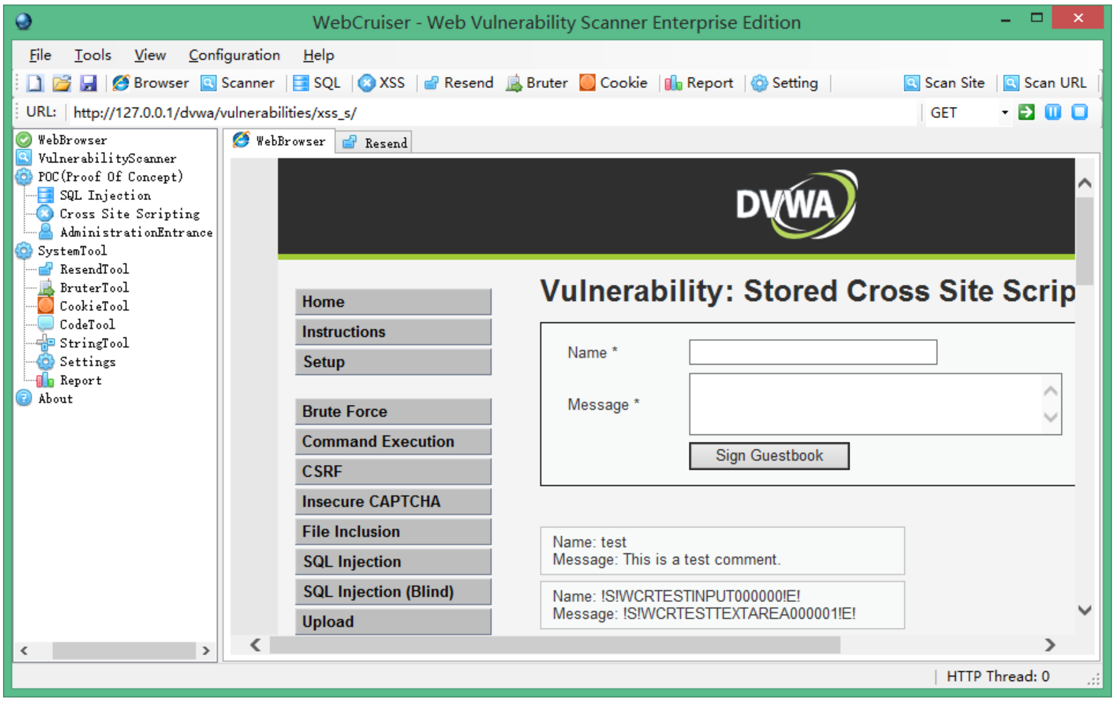
URL / Refer URL	Parameter	Type	KeyWord/ActionURL	Vulnerability
http://127.0.0.1/dvwa/vulnerabilities/sqli/?Submit=Submit&i...	id=99999999	String	First	URL SQL INJECTI
http://127.0.0.1/dvwa/vulnerabilities/sqli_blind/?Submit=Su...	id=99999999	String	First	URL SQL INJECTI
http://127.0.0.1/dvwa/vulnerabilities/xss_r/	name	GET	http://127.0.0.1...	Cross Site Scri

Checking Form Vul: http://127.0.0.1/dvwa/vulnerabilities/xss_r/

HTTP Thread: 0

HTTP Thread: 0

Continue stored XSS, "Scan URL":



Vulnerability: Stored Cross Site Scrip

Name *

Message *

Name: test
Message: This is a test comment.

Name: !S!WCRTESTINPUT000000!E!
Message: !S!WCRTESTTEXTAREA000001!E!

We got it.

4. WAVSEP Test Report

WAVSEP v1.5 all SQL Injection & XSS test cases 100% covered, test report is available here:

http://www.janusec.com/download/WebCruiser_Web_Vulnerability_Scanner_Test_Report.pdf

4.1. Product and Test Cases

WAVSEP (Web Application Vulnerability Scanner Evaluation Project) v1.5

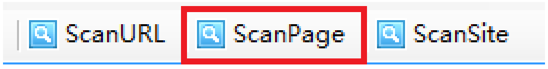
WAVSEP Environment: Windows8.1 + XAMPP (Tomcat + MySQL)

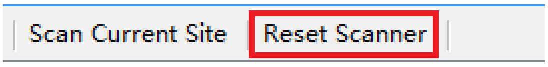
WebCruiser Web Vulnerability Scanner Enterprise Edition V3.1.0

4.2. Test Method

In order to get the test results quickly, we use a new feature of WebCruiser Web Vulnerability Scanner, which is “Scan Page”, which means it will scan all links in a page once a time. This function requires that the links locate under the same or sub directory, links under other directories will be skipped.

When start a new page scan, click “Reset Scanner” to clear previous result, and navigate to new page, and then click “ScanPage”





4.3. SQL Injection Test Report

Input Vector	Test Cases	Cases Count	Report	Pass Rate
GET Input	Erroneous 500	19	19	100%

Vector	Responses			
	Erroneous 200 Responses	19	19	100%
	200 Responses With Differentiation	19	19	100%
	Identical 200 Responses	8	8	100%
POST Input Vector	Erroneous 500 Responses	19	19	100%
	Erroneous 200 Responses	19	19	100%
	200 Responses With Differentiation	19	19	100%
	Identical 200 Responses	8	8	100%
GET Input Vector – Experimental	Insert / Delete / Other	1	1	100%
POST Input Vector - Experimental	Insert / Delete / Other	1	1	100%

4.4. XSS Test Report

Input Vector	Test Cases	Cases Count	Report	Pass Rate

GET Input Vector	ReflectedXSS	32	32	100%
POST Input Vector	ReflectedXSS	32	32	100%
Cookie Input Vector - Experimental	ReflectedXSS	1	1	100%
GET Input Vector - Experimental	ReflectedXSS	11	11	100%
POST Input Vector - Experimental	ReflectedXSS	11	11	100%
GET Input Vector - Experimental	DomXSS	4	4	100%

4.5. LFI Test Report

Input Vector	Test Cases	Cases Count	Report	Pass Rate
Get Input Vector	Erroneous HTTP 500 Responses	68	68	100%
	Erroneous HTTP 404	68	68	100%

	Responses			
	Erroneous HTTP 200 Responses	68	68	100%
	HTTP 302 Redirect Responses	68	68	100%
	HTTP 200 Responses With Differentiation	68	68	100%
	HTTP 200 Responses with Default File on Error	68	68	100%
POST Input Vector	Erroneous HTTP 500 Responses	68	68	100%
	Erroneous HTTP 404 Responses	68	68	100%
	Erroneous HTTP 200 Responses	68	68	100%
	HTTP 302 Redirect Responses	68	68	100%
	HTTP 200 Responses With Differentiation	68	68	100%
	HTTP 200 Responses with Default File on Error	68	68	100%

4.6. RFI Test Report

Input Vector	Test Cases	Cases Count	Report	Pass Rate
Get Input Vector	Erroneous HTTP 500 Responses	9	9	100%
	Erroneous HTTP 404 Responses	9	9	100%
	Erroneous HTTP 200 Responses	9	9	100%
	HTTP 302 Redirect Responses	9	9	100%
	HTTP 200 Responses With Differentiation	9	9	100%
	HTTP 200 Responses with Default File on Error	9	9	100%
POST Input Vector	Erroneous HTTP 500 Responses	9	9	100%
	Erroneous HTTP 404 Responses	9	9	100%
	Erroneous HTTP 200 Responses	9	9	100%
	HTTP 302 Redirect	9	9	100%

	Responses			
	HTTP 200 Responses With Differentiation	9	9	100%
	HTTP 200 Responses with Default File on Error	9	9	100%

4.7. Redirect Test Report

Input Vector	Test Cases	Cases Count	Report	Pass Rate
Get Input Vector	HTTP 302 Redirect Responses	15	15	100%
	HTTP 200 Responses With Javascript Redirect	15	15	100%
POST Input Vector	HTTP 302 Redirect Responses	15	15	100%
	HTTP 200 Responses With Javascript Redirect	15	15	100%

4.8. False Positive Test Report

False Vuln	Test Cases	Cases Count	Report	Pass Rate
SQL Injection	False Positive	10	0	100%
XSS	False Positive	7	0	100%

5. Order/Registration

WebCruiser - Web Vulnerability Scanner Order page:

http://www.janusec.com/downloads/

If you like it, you can order it from MyCommerce or Avangate:

Professional Edition (Non-Commercial License):

https://shopper.mycommerce.com/checkout/product/25854-1

https://secure.avangate.com/order/checkout.php?PRODS=4540814&QTY=1&CART=1

Enterprise Edition (Commercial License):

https://shopper.mycommerce.com/checkout/product/25854-2

https://secure.avangate.com/order/checkout.php?PRODS=4540841&QTY=1&CART=1

MyCommerce or Avangate will send you the Registration Code.

Thank you for choosing WebCruiser.

6. FAQ

Q: Why I can not run WebCruiser on my computer?

A: It need Windows with .Net Framework 2.0 or above, if you have not installed .Net Framework, please downloads it from Microsoft web site. Usually, Windows XP and earlier has not .Net Framework installed, but Windows Vista and Windows 7 has .Net Framework Integrated already. The URL for .Net Framework 2.0 (3.5) is:

http://www.microsoft.com/downloads/details.aspx?FamilyID=0856EACB-4362-4B0D-8EDD-AAB15C5E04F5&displaylang=en

Q: What is the difference between the Free, Professional and Enterprise Edition?

A: They are different in License type.

- ✧ Professional Edition is for security professionals, masters of individual websites etc., non-commercial purpose, 12-month update and support service;
- ✧ Enterprise Edition is for enterprises, institution, or commercial organizations, 12-month update and support service with top priority.

Q: What is the most typical feature of WebCruiser comparing with other Web Vulnerability Scanners?

A: First, WebCruiser Web Vulnerability Scanner focuses on high risk vulnerabilities. Second, WebCruiser can scan a designated vulnerability type, or a designated URL, or a designated page separately, while the others usually will not.

http://www.janusec.com/documentation/

Support Web Site:

http://www.janusec.com

Support E-mail: janusecurity#gmail.com