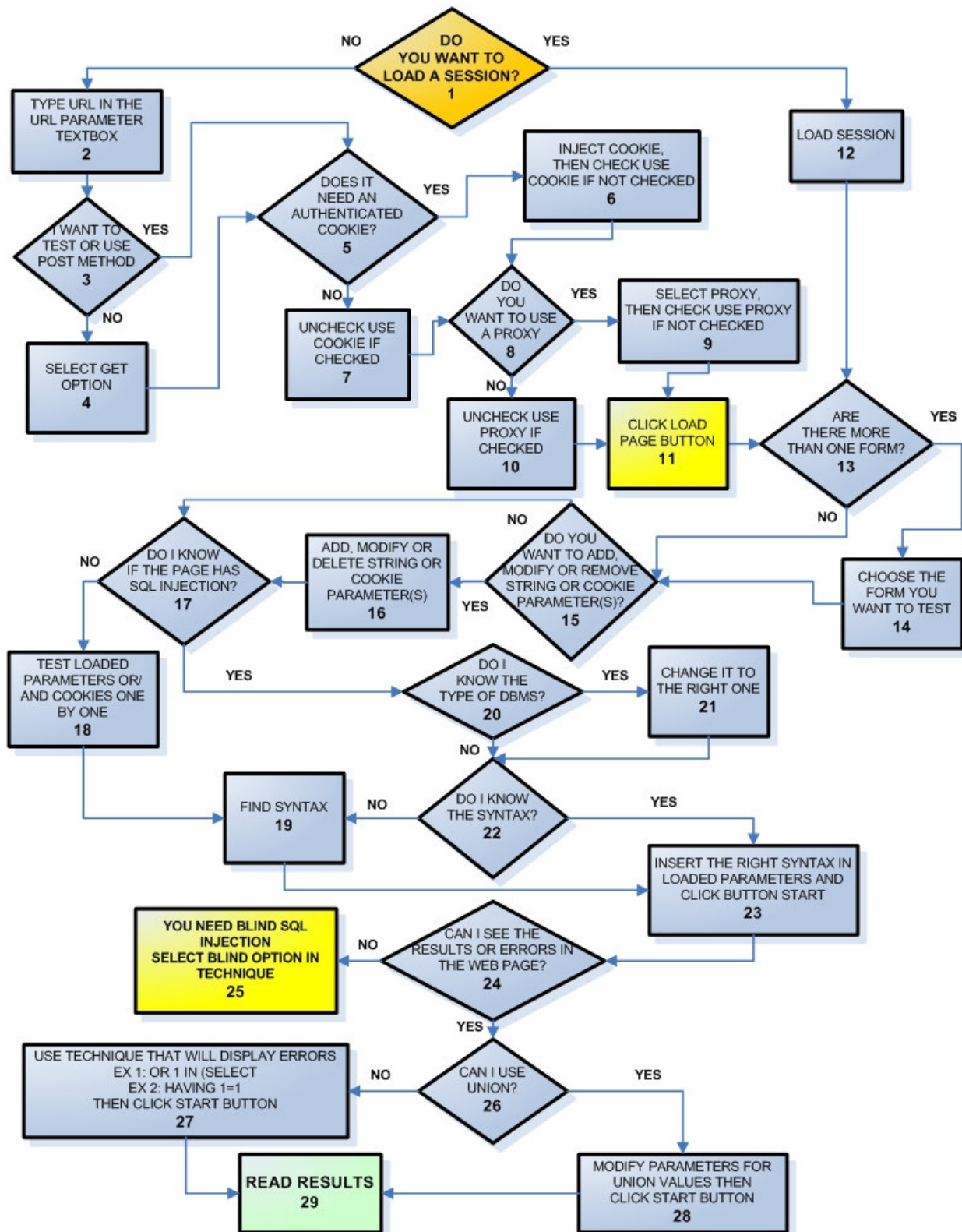
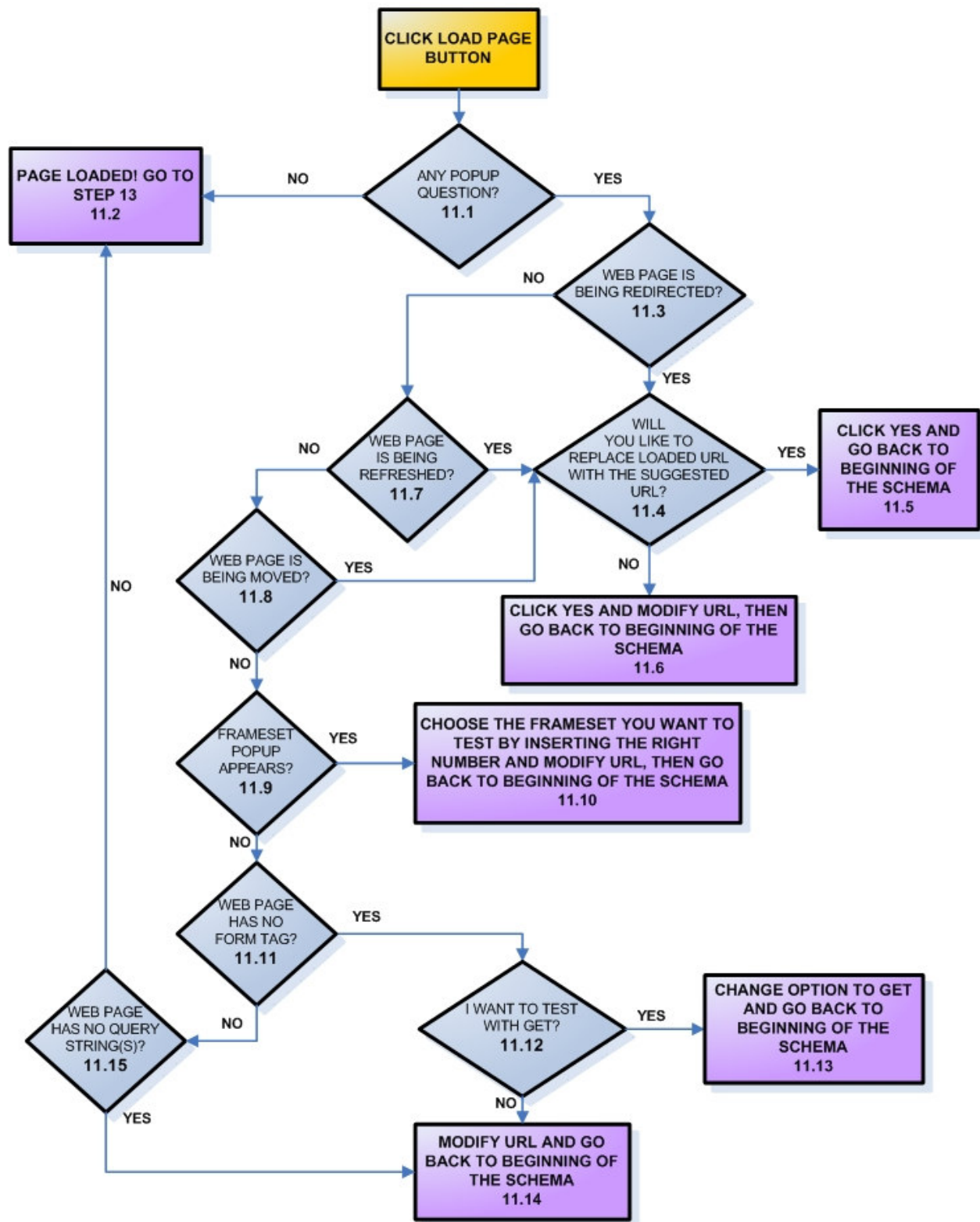


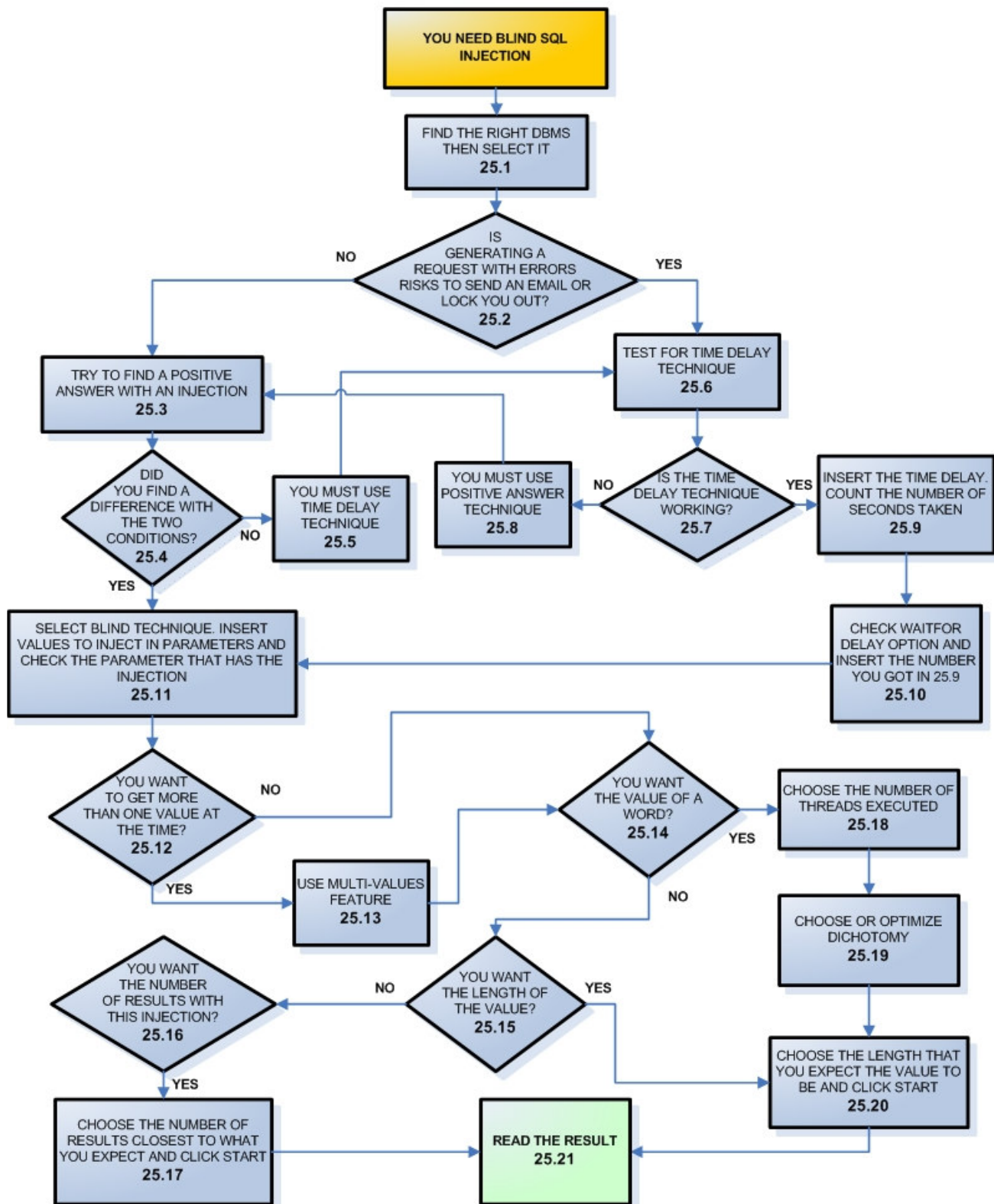
# MAIN TUTORIAL SCHEMA



# LOAD PAGE TUTORIAL SCHEMA



# BLIND SQL INJECTION TUTORIAL SCHEMA



Please follow first the main schema and if you need more details go to their respective number detailed below. Moreover, in two circumstances you will get drill down schemas (11 and 25). There you can look at the sub-schema for more details and like the main schema go in their respective number for more details.

## 1. DO YOU WANT TO LOAD A SESSION?

We start the tutorial with that question since you might have already initiated a session before and just want to reuse it. If it's the case then your answer is **YES – GO TO STEP 12.**

If you have never saved any session your answer is **NO – GO TO STEP 2.**

In other times you do not want to load any session or wish to start from scratch then your answer is **NO – GO TO STEP 2.**

### HINT 1

To save session can be a real time saver when you want to get back to your tests exactly where you were when you stopped. Or just to save the session with different values.

### HINT 2

Once the session is saved it is possible to go modify it directly in the XML file. There you can either change some values, remove some of them, update a JSESSIONID or even add a new form! Keep in mind that this file could be more useful than just a session repository. However, new in version 1.2 you can now modify all parameters and cookies directly in the application. It's more convenient and easy to save after.

### HINT 3

If you are making tests with a web site using Java which keeps the web site context with a session id it is possible to update the cookie directly in the XML file. What you need to do is to modify the value of *SubmitUri* in the *HtmlForm* tag with the current session id. To do so you need to add a semi-colon (;) with the java session id right before the web page name and right before the query string values

Ex:

***MyPage.jsp;JSESSIONID=D23TfhU3fdf7884HDSA4hfdGs?Param1=test&Param2=1***

### NOTE 1

Remember that if your session was working at the time you were using it, it might no more work the next time you load it. Two reasons are possible:

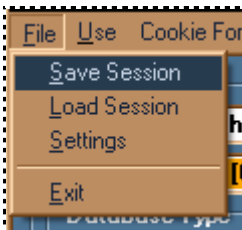
1. The cookie is no more valid. Either re-inject it (see **STEP 6** for more details) or if it's a Java web site, open the XML file and modify it accordingly to the **Hint 3**.
2. Or the web site has changed and some or all values are no working. Just try to reload the page with the Load button to see what's happening.

**NOTE 2**

Now in version 1.2 you can save all injected and loaded cookies in the saved session. Please remember as mentioned before that the session will be valid only for its expected time life.

Of course in order to be able to load a session you need to save it first. To do so, you first need to successfully load a page and from that point you can save it. It does not matter if you have or haven't tested the web site yet for SQL injection as long as you loaded it you can save it.

You will find the save session under the menu File:



From there you just need to save it like any normal file you would save under Windows.

- IF YOUR ANSWER IS **NO** GO TO **STEP 2**
- IF YOUR ANSWER IS **YES** GO TO **STEP 12**

**2. TYPE URL IN THE URL PARAMETER TEXTBOX**

- Get the URL where you want to see if there are SQL injections, or that you already know there are.
- Copy paste it in that textbox

**NOTE 1**

If you have http:// or https:// missing the application will raise an error. I realized that it's much clearer that way and that the user should have the control to use the protocol he wants.

Please note that SSL is now supported since version 1.1

**NOTE 2**

It might happen that you already know there are SQL injections in a web site because you've got a positive answer from an automated tool such as Paros. I think it could be a good technique to use an automated tool to make the rough work of analyzing every page then to come back to exploit it with SQL Power Injector, unless of course you don't want to leave too much traces that those kind of applications leave.



**GO TO STEP 3****3. I WANT TO TEST OR USE POST METHOD**

That question could be quite easy to answer if your URL doesn't contain any query string variables. If it does and it's what you want to do then your answer is **NO – GO TO STEP 4.**

However sometimes you have query string variables but you still want to test it for POST so your answer is **YES – GO TO STEP 5.**

If you want to test for POST so your answer is **YES – GO TO STEP 5.**

**NOTE**

If you choose to use the POST method and there are no form tags in the loaded web page you will get a popup text stating that there is nothing. In that case, either you've got the wrong URL or you should use GET method.

**HINT**

If you've got the message stating there are no Form tags in the requested URL, go back to the web page in the browser and right click close to the textbox you want to inject. There in the Properties copy paste the URL, it might be a different one containing your textboxes you want to test.

- IF YOUR ANSWER IS **NO** GO TO **STEP 4**
- IF YOUR ANSWER IS **YES** GO TO **STEP 5**

**4. SELECT GET OPTION**

- Select GET option

**NOTE**

Of course if you don't have any query string variables or they are malformed you will get a popup message stating that you don't have any and should select POST. In case they are malformed (by a bad copy paste for example) you just need to correct it.

**GO TO STEP 5****5. DOES IT NEED A COOKIE?**

This case can occur when your SQL injection will be in a page where you need to be previously authenticated. Without this cookie you will never be able to load the right page and will always be kicked out.

If it's your case and you need to login first then your answer is **YES – GO TO STEP 6**.

If you don't need to login in then the answer is **NO – GO TO STEP 7**.

Another scenario is that you will need one to keep the context of the page with a session cookie even though there is no authentication at the beginning. A good example would be the .Net VIEWSTATE object that is passed from one page to the other with the help of the cookie session. If it doesn't exist, then you get an error message on the next page (in POST mode)

If it's your case and you need to keep the context then your answer is **YES – GO TO STEP 6**.

If you don't need to keep the context then the answer is **NO – GO TO STEP 7**.

**NOTE 1**

By default the option is set to *Use cookie*.

**NOTE 2**

In some rare occasion you won't want to keep the context of the cookie to be passed from one injection to the other (see blind injection technique **STEP 25**) and will want to create a fresh new cookie each time.

This case can occur when you are in a login page and succeed to have found your positive answer (special technique explained in **STEP 25.3**) by bypassing the login page (with the '**or 1=1--**' technique for example) but once in the web site there're no ways to get SQL injection. At that point, you'll need to execute your SQL injection with the positive answer technique, if you choose so and each time you have a true occurrence you'll be automatically sent to the other page when you request the page again (with the session cookie set the web application assumes that you're already authenticated). So no matter if your condition is true or not, it will redirect you there. Thus, you'll end up having only true answers... You should then consider that you don't need any cookie and go to **STEP 7**.

- IF YOUR ANSWER IS **NO** GO TO **STEP 7**
- IF YOUR ANSWER IS **YES** GO TO **STEP 6**

## 6. INJECT COOKIE, THEN CHECK USE COOKIE IF NOT CHECKED

In this step it's a bit trickier, since there is no login feature that will get the resulting authenticated session cookie. In order to achieve so, you will need to use a proxy application to trap the cookie generated or use a browser that has the feature to have the cookie information in its properties such as Mozilla.

### NOTE 1

You'll get about the same information in the question "*I know I can SQL inject but only after I'm logged in the web application, what can I do?*" in the FAQ section of the website.

However, new to this version you can use the Mozilla Firefox plugin that will automatically launch SQL Power Injector with the current state (all query strings if GET is chosen or else the inputs if POST is selected and all the cookies created while browsing the web site with Mozilla Firefox). It is by far the easiest and more convenient way.

### NOTE 2

You'll need to install the plugin in Mozilla Firefox first. And after it's pretty straight forward to use, once you want to load this page from SQL Power Injector, either go to the menu Tools then click on "SQL Power Injector Plugin" or simply right click on the website and click on "SQL Power Injector Plugin".

Last detail, it will by default load the page with POST method, so if you want to change it you will need to go in Tools menu inside Mozilla Firefox, then on Add-Ons. There find SQL Power Injector plugin and click on Options. From there you can chose which method you want to load it by default. I might make this option more accessible later on.

I will give more details on how to use the Plugin inside the FAQ section of the website.

### NOTE 3

A new feature with this version is the "Cookie For Load Page" management. How it is used will be explained later. Please make sure you understand this new method if you think you might need to use it often.

There are now two ways to inject a cookie that will be used during the "Load Page" phase. The first way looks more like the old method used in the previous versions and the second one now integrates a new window with a Datagrid on which it's easy to add, modify or remove cookies.

Here's what you need to do to get the authenticated session cookie:

1. With a normal browser go on the web site and log on. Normally the web site will create a session with a session token in the cookie.



2. You need to get it either with the browser preferences (Mozilla, Netscape and the like) or with a proxy application such as Paros (<http://www.parosproxy.org/index.shtml>) or webscarab (<http://www.owasp.org/software/webscarab.html>) to name a few... I personally recommend using a proxy application because with the browsers you won't get the cookie information in the right format, that is to say:  
*MySessionID=AGDAFHAD3142324*. Once you found it, just copy it.

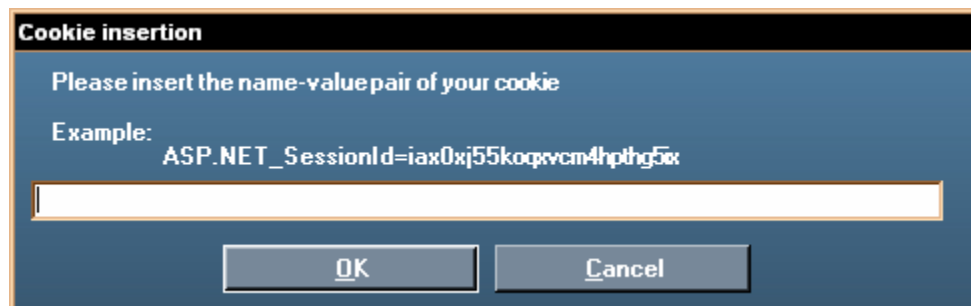
```
GET http://localhost/VulnerableSQLSite/Search.asp HTTP/1.0
Accept: */*
Accept-Language: en-us
Pragma: no-cache
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.0.3705; .NET C
Host: localhost
Proxy-Connection: Keep-Alive
Cookie: ASPSESSIONIDSSRRTBBC=EBIPJBCBEDMHMDIPGOFLLBBCN
```

### First method

3. Now, with this information go in the application SQL Power Injector and in the menu "Cookie For Load Page" click on "Insert Cookie"



4. An input box will be displayed, just paste the cookie value there and click ok. The format should look like this for a JSP session for example:  
*JSESSIONID=D23TfhU3fdf7884HDSA45hfdGs*

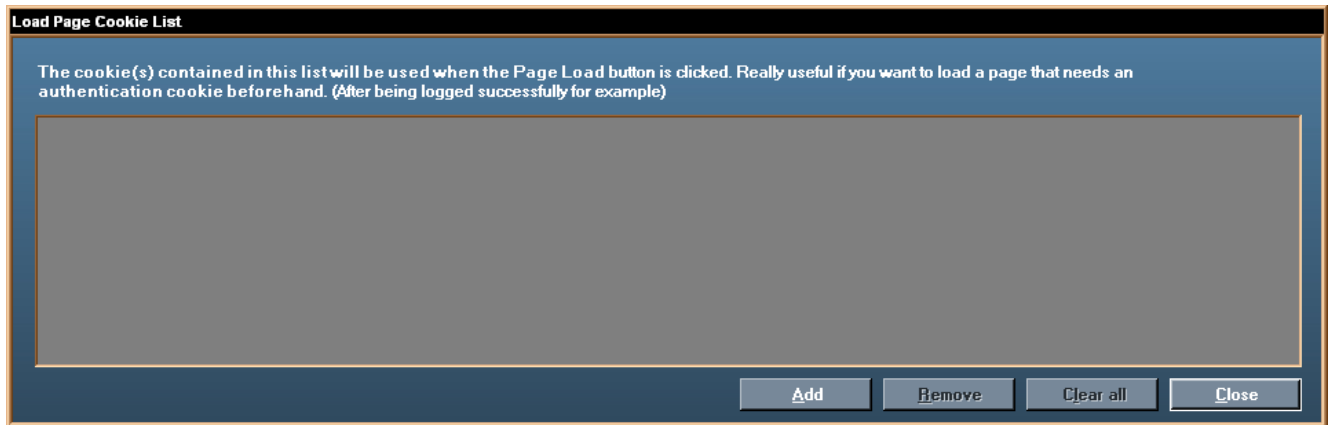



## Second method

- Now, with this information go in the application SQL Power Injector and in the menu "Cookie For Load Page" click on "Manage"



- A window will appear with an empty Datagrid which will contain eventually all the cookies you want to be used when the page is loaded

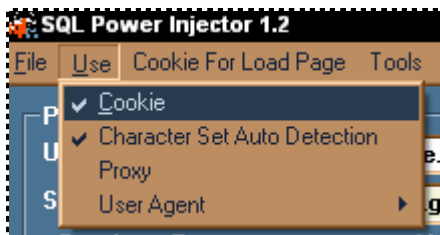


- There you just need to click on Add button  and an input box will appear. Just follow the instructions given in the first method on step 4.

### NOTE 4

As you can see you can manage and visualize the cookies for the Load Page phase easily with this Datagrid. In order to remove a cookie you need to select one or more cookie on the Datagrid.

Once you're done, you will need to make sure that the application will use the cookie in the menu Use and then that Cookie is checked. If it's not checked, do so.



**IMPORTANT NOTE**

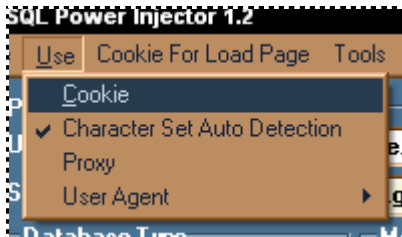
The cookies added inside the “Cookie For Load Page” will be used and added to the cookie list only after you load a page. However, if you add any cookies after you loaded the page inside the “Cookie For Load Page” they will not be added to the list.

If you want to add any cookie after you loaded the page for other tests I suggest you use the Datagrid Cookie Parameters (See **STEP 18** for more details)

**GO TO STEP 8**

**7. UNCHECK USE COOKIE IF CHECKED**

As the title implies uncheck the use cookie option if it's checked. You'll find this option in the menu under Use and Cookie.



**GO TO STEP 8**

**8. DO YOU WANT TO USE A PROXY**

You might need to use a proxy for many reasons. For example, if you need to go through a corporate proxy to go out to the Internet. Or you might not want your IP address to be recognized by the target website (still in all legality of course)

If it's your case and you need use a proxy then your answer is **YES** – **GO TO STEP 9**.

If you don't need to use a proxy then the answer is **NO** – **GO TO STEP 10**.

**NOTE 1**

By default the option is set to **not Use Proxy**.

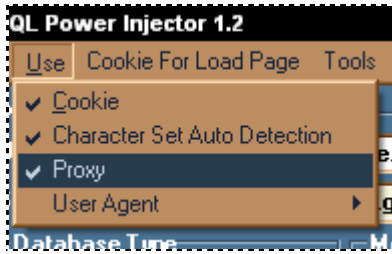
**NOTE 2**

The connection might be impacted by the proxy, so be warned.

- IF YOUR ANSWER IS **NO** GO TO **STEP 10**
- IF YOUR ANSWER IS **YES** GO TO **STEP 9**

## 9. SELECT PROXY, THEN CHECK USE PROXY IF NOT CHECKED

If you have already set a proxy in the settings, then you only need to check the option Proxy under the menu Use

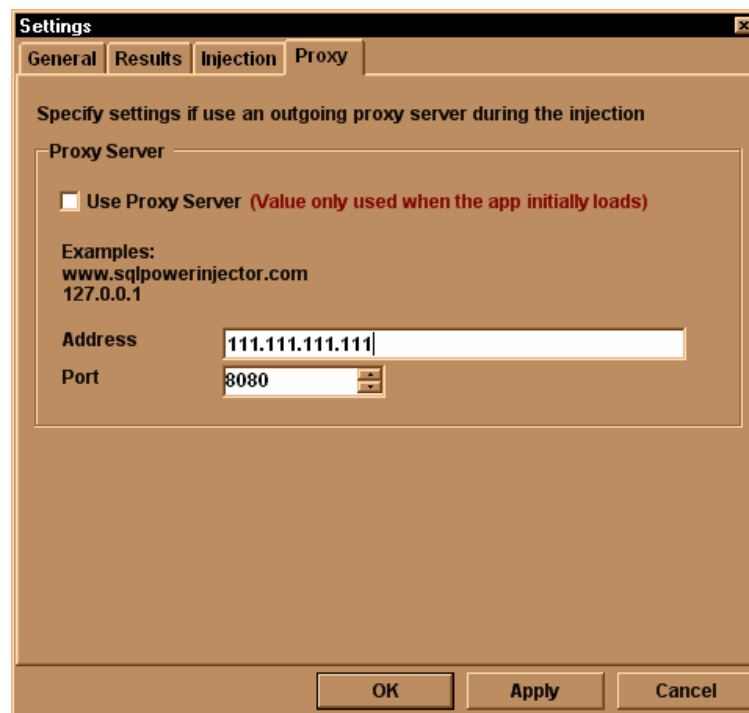


In case you didn't already set the proxy here is how:

1. Go on the menu File and then click on Settings



2. A settings dialog box will appear, from there click on the tab chose Proxy



3. At this moment you just need to add your IP address and port number. Then click on OK or Apply then OK.

**NOTE 1**

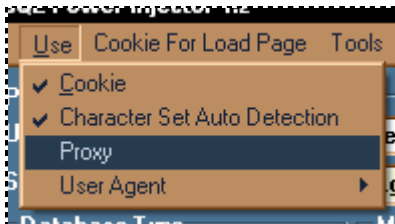
As you will notice many other settings can be found but since they are self explanatory it will not be further discussed in this tutorial.

**NOTE 2**


It is useful to know that if you check the Use Proxy Server option in the settings dialog box it will only be used when the application initially loads.

**GO TO [STEP 11](#)****10. UNCHECK USE PROXY IF CHECKED**

As the title implies uncheck the use proxy option if it's checked. You'll find this option in the menu under Use and Proxy.

**GO TO [STEP 11](#)****11. CLICK LOAD PAGE BUTTON**

This step has been drilled down to sub-steps to facilitate the understanding as you can notice as well in the schemas. It was created like this to alleviate the complexity of one step in a different schema where it can be exploded in a many more understandable sub-steps. Here we assume that the user will logically follow those steps in the reading order.

- Click on Load Page button 

**GO TO [STEP 11.1](#)****11.1 ANY POPUP QUESTION?**

If there is no popup message after you have loaded the page then everything is all right and the answer of the question is **NO** – **GO TO [STEP 11.2](#)**.



If you've got a popup message stating something went wrong then the answer is **YES** – **GO TO STEP 11.3**

**NOTE**

The next popup error messages discussed in the sub-steps include only the most common errors. There is no way to cover every possibility of error that an application could have, but at least the ones that are possible in the using context are explained.

- IF YOUR ANSWER IS **NO** GO TO **STEP 11.2**
- IF YOUR ANSWER IS **YES** GO TO **STEP 11.3**

**11.2 PAGE LOADED! GO TO STEP 13**

The page has been loaded successfully and you can get back to the main tutorial schema to continue the process.

You should get values in the String and/or Cookie Parameters Datagrids. These values in case of the String Parameters Datagrid will be from the form (INPUT tag of text, hidden, password, select, radio button and textarea type) if you have selected POST method or the values of the query strings in the URL if you have chosen GET method.

And if any cookie has been set during the loading phase then you will see them displayed in the Cookie Parameter Datagrid. Or if you already added one or more cookies inside the “Cookie For Load Page” menu.

It should look something like that:

String Parameters		Cookie Parameters				Add	Remove		
	<input type="checkbox"/>	Name	Starting string	Varying string		Ending string			
▶	<input type="checkbox"/>	client	firefox-a						
	<input type="checkbox"/>	rls	org.mozilla:en-US:official						

**NOTE 1**

Of course the variables could be possibly different than this example since it depends entirely of each web page loaded.

**NOTE 2**

If there is more than one form then you will see the detected parameters in the Datagrid only one form at the time. To see the other(s) you will need to go in the Submit URL Combobox and choose another one. At that moment, the values of that form will be displayed in the Datagrid.

It does not apply to the Cookie Parameters list however since the cookies exist for that page.

**GO TO STEP 13 ON MAIN TUTORIAL SCHEMA**

### 11.3 WEB PAGE IS BEING REDIRECTED?

The URL you have loaded has to be redirected, which is why you might want to follow the redirection to get to the real web page.

It can happen by a redirection 3XX HTTP code or by client-side code that will redirect the user as soon as the page is being displayed.

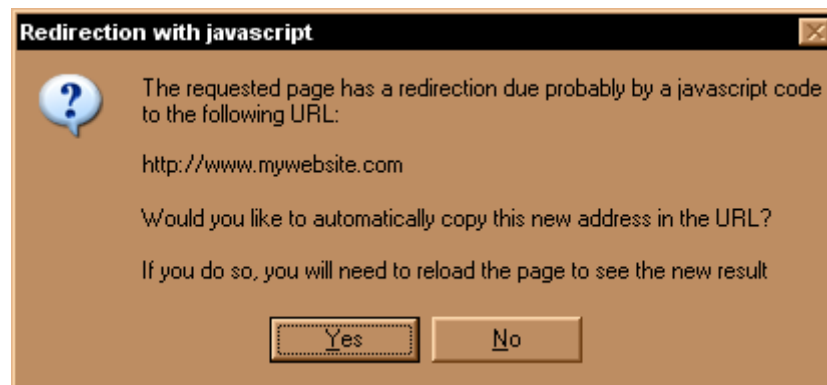
If you get the redirection popup message then your answer is **YES – GO TO STEP 11.4.**

If you don't get any popup message stating that there is a redirection then your answer is **NO – GO TO STEP 11.7.**

The question will look something like that:



Or if it is being redirected by client-side code:



- IF YOUR ANSWER IS **NO** GO TO **STEP 11.7**
- IF YOUR ANSWER IS **YES** GO TO **STEP 11.4**

#### **11.4 WILL YOU LIKE TO REPLACE LOADED URL WITH THE SUGGESTED URL?**

At that point the suggested URL might point to somewhere you don't want to go. It gives you the choice to replace it automatically in the URL textbox parameter so the answer is **YES – GO TO STEP 11.5** or to keep the old one so the answer is **NO – GO TO STEP 11.6**.

- IF YOUR ANSWER IS **NO** GO TO **STEP 11.6**
- IF YOUR ANSWER IS **YES** GO TO **STEP 11.5**

#### **11.5 CLICK YES AND GO BACK TO BEGINNING OF THE SCHEMA**

Answer yes and automatically your URL will be the one suggested in the URL parameter textbox. You'll have to click again Click Load Page button.

After, you will need to go back to beginning of the “Load page” schema to continue the process (**STEP 11**).

**GO TO STEP 11 AT THE BEGINNING OF THE LOAD PAGE SCHEMA**

#### **11.6 CLICK YES AND MODIFY URL, THEN GO BACK TO BEGINNING OF THE SCHEMA**

- Click on yes
- If you need to modify or change the URL that has been loaded, do so
- Click Load Page button again

Once done, you will need to go back to beginning of the “Load page” schema to continue the process (**STEP 11**).

**GO TO STEP 11 AT THE BEGINNING OF THE LOAD PAGE SCHEMA**

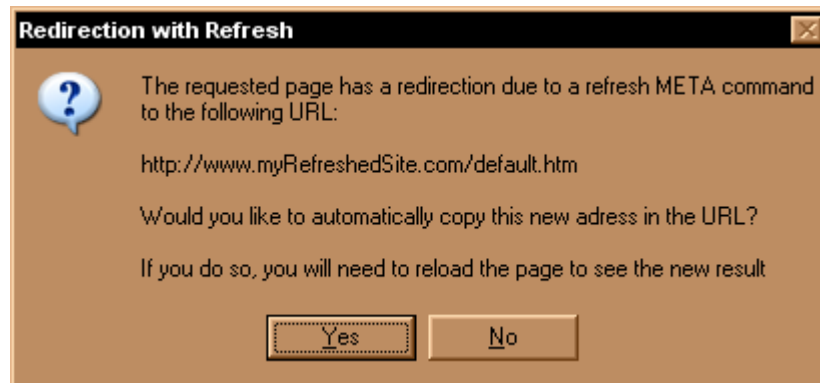
#### **11.7 WEB PAGE IS BEING REFRESHED?**

The URL you have loaded have to be refreshed, which is why you might want to follow the redirection to get to the real web page.

If you get the redirection with refresh popup message then your answer is **YES – GO TO STEP 11.4**.

If you don't get any popup message stating that there is a redirection due to a META Refresh tag then your answer is **NO – GO TO STEP 11.8**.

The question will look something like that:



- IF YOUR ANSWER IS **NO** GO TO **STEP 11.8**
- IF YOUR ANSWER IS **YES** GO TO **STEP 11.4**

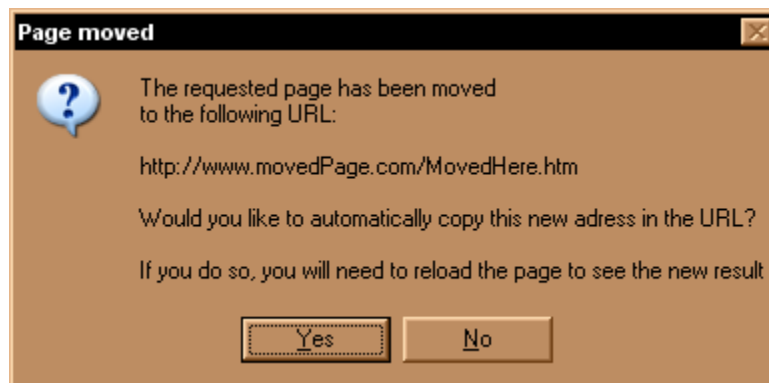
### 11.8 WEB PAGE IS BEING MOVED?

The URL you have loaded has moved, which is why you might want to follow the redirection to get to the real web page.

If you get the Page moved popup message then your answer is **YES – GO TO STEP 11.4**.

If you don't get any popup message stating that it has been moved then your answer is **NO – GO TO STEP 11.9**.

The question will look something like that:



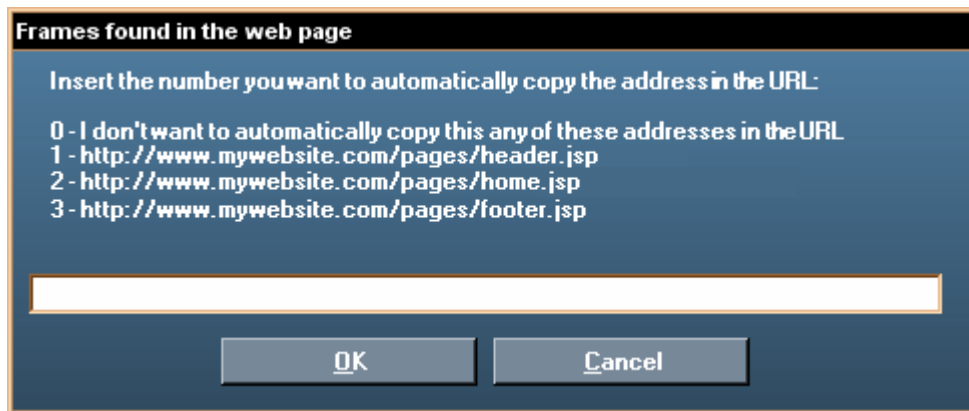
- IF YOUR ANSWER IS **NO** GO TO **STEP 11.9**
- IF YOUR ANSWER IS **YES** GO TO **STEP 11.4**

## 11.9 FRAMESET POPUP APPEARS?

If there is a frameset with several frames a popup message box will display asking to choose a number that corresponds to a frame URL. If it's the case then your answer is **YES – GO TO STEP 11.10**.

If you don't get any popup message asking to choose a frameset then your answer is **NO – GO TO STEP 11.11**.

In every case you will find the choice 0 that does exactly the same thing as to click on cancel.



### NOTE

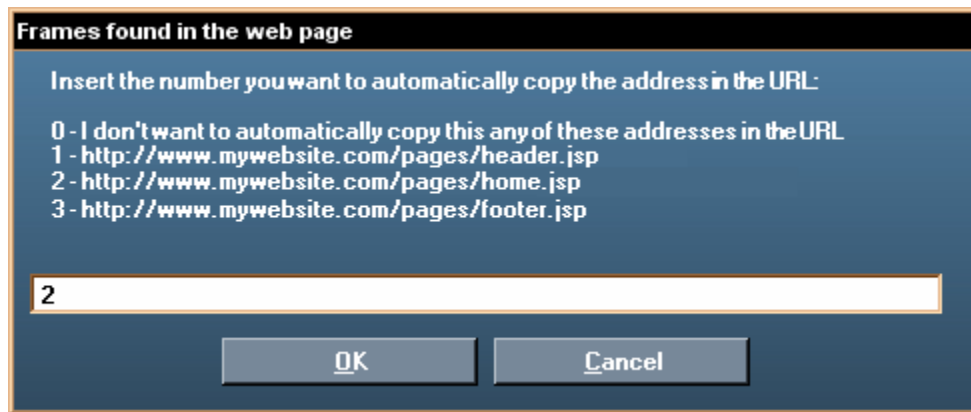
It is possible that you will run into several framesets. If it happens just choose the one you want to go until you reach a normal web page.

- IF YOUR ANSWER IS **NO** GO TO **STEP 11.11**
- IF YOUR ANSWER IS **YES** GO TO **STEP 11.10**

## 11.10 CHOOSE THE FRAMESET YOU WANT TO TEST BY INSERTING THE RIGHT NUMBER AND MODIFY URL, THEN GO BACK TO BEGINNING OF THE SCHEMA

Depending of the number of choice you will have to choose one frameset. A number associates each of them. It's that number that you need to insert in order to automatically copy the new URL in the URL textbox.





If you don't want to choose any of those, just click on cancel or insert 0 then click Ok. Once you have inserted the number, click ok and it will copy paste the URL in the URL textbox.

**NOTE**

Any value not included in the choice list will result as if you chose 0 or clicked on cancel.

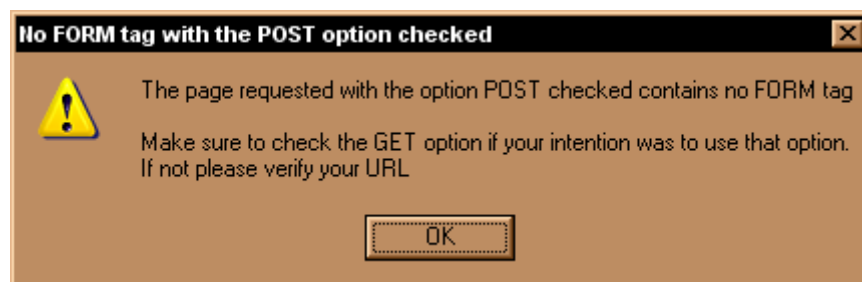
**GO TO [STEP 11](#) AT THE BEGINNING OF THE LOAD PAGE SCHEMA****11.11 WEB PAGE HAS NO FORM TAG?**

The URL you have loaded has no form tag and you selected the option POST.

If you get this message then your answer is **YES** – **GO TO [STEP 11.12](#)**.

If you don't get any popup message stating that no form tag has been found then your answer is **NO** – **GO TO [STEP 11.15](#)**.

The popup message will look something like that:



- IF YOUR ANSWER IS **NO** GO TO **[STEP 11.15](#)**
- IF YOUR ANSWER IS **YES** GO TO **[STEP 11.12](#)**

### 11.12 I WANT TO TEST WITH GET?

As the popup message stated in the **STEP 8.11** you might have wanted to load the page with GET method but you forgot to select it, if it is your case then your answer is **YES** – **GO TO STEP 11.13**.

If it was really the POST option you wanted perhaps you have made a mistake in the URL and should change it, in that case your answer is **NO** – **GO TO STEP 11.14**.

- IF YOUR ANSWER IS **NO** GO TO **STEP 11.14**
- IF YOUR ANSWER IS **YES** GO TO **STEP 11.13**

### 11.13 CHANGE OPTION TO GET AND GO BACK TO BEGINNING OF THE SCHEMA

You obviously forgot to select the GET option, just do so. After you're done you'll have to click again Click Load Page button.



After, you will need to go back to beginning of the “Load page” schema to continue the process (**STEP 11**).

**GO TO STEP 11 AT THE BEGINNING OF THE LOAD PAGE SCHEMA**

### 11.14 MODIFY URL AND GO BACK TO BEGINNING OF THE SCHEMA

Your URL might have some mistake in it, verify if it's the right one or need to be modified and if you need to modify it, do so. After you're done you'll have to click again Click Load Page button.

#### **NOTE**

If you're in GET method and get the popup message that there are no query string(s) it might mean that you have a malformed query pair name value. Verify that part and make sure it hasn't been cut in the middle of it.

It can happen when you copy paste a URL from an email.

After, you will need to go back to the beginning of the “Load page” schema to continue the process (**STEP 11**).

**GO TO STEP 11 AT THE BEGINNING OF THE LOAD PAGE SCHEMA**

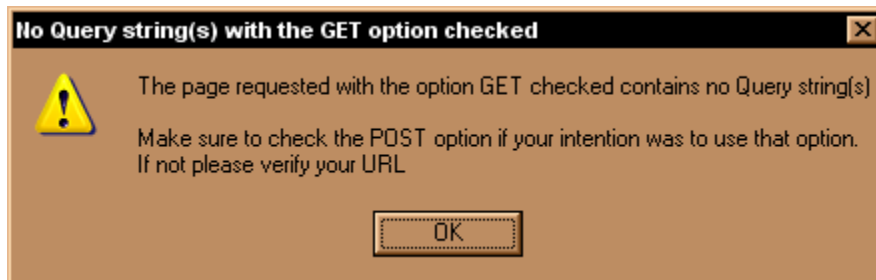
### 11.15 WEB PAGE HAS NO QUERY STRING(S)?

The URL you have loaded has no query string(s) and you selected the option GET.

If you get this message then your answer is **YES** – GO TO **STEP 11.14**.

If you don't get any popup message stating that no query string(s) have been found then your answer is **NO** – GO TO **STEP 11.2**.

The popup message will look something like that:



- IF YOUR ANSWER IS **NO** GO TO **STEP 11.2**
- IF YOUR ANSWER IS **YES** GO TO **STEP 11.14**

### 12. LOAD SESSION

You should be in that step if you had previously saved a session or have a session saved by someone else at your disposition and of course you want to use the same settings.

#### NOTE 1

You can always modify that saved session with your own settings either directly in the XML file prior to load it or inside the application.

#### HINT 1

You should go read the Hint 1 to 3 and the Note of the Step 1. This topic has already been partly discussed there.

#### HINT 2

You can add new parameters or even new forms using the XML formatting in the XML files. It's a nice way to test new parameters that were never loaded at first place or that you discovered later during the penetration testing.

In order to load a session you need to go on the menu File and click on Load Session.



From there the standard open file windows will open. At that moment you will need to search for the right XML file, then once found you select it and click on the button Open.

**NOTE 2**

By default the Load Session action will open in the Saved Session directory under your SQL Power Injector application directory.

**NOTE 3**

It is clear that if you have modified the XML file without respecting its format you would get an error.

**GO TO STEP 13**

**13. ARE THERE MORE THAN ONE FORM?**

It is possible that a web page contains more than one form. To see if you have more than one you will need to click on the Submit URL combo and at that moment you will see them all. If you see more than one then your answer is **YES – GO TO STEP 14**.

If you see only one then your answer is **NO – GO TO STEP 15**.

**NOTE 1**

If there are more than one form then you will see the detected parameters in the Datagrid only one form at the time. To see the other(s) you will need to go in the Submit URL Combobox and choose another one. At that moment, the values of that form will be displayed in the Datagrid.

**NOTE 2**

You might notice that the Submit URL has a different color depending of its method. If it's POST then it's light blue and if it's GET it's beige. The hover color is orange so do not be surprised if it's neither light blue nor beige. Also to make sure it is obvious there is a prefix [POST] or [GET] in front of the Submit URL.

**NOTE 3**

Even though you have chosen to load the page with one method does not mean that the method of that form will be the same. If it's different you will have to change the radio button for the right method prior to click on the start button to be sure to get the right behavior.

**HINT**

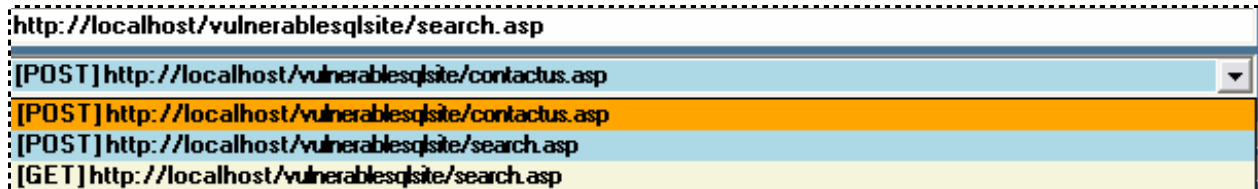
It could be interesting to test the detected method with a different one just to see how the application reacts.

- IF YOUR ANSWER IS **NO** GO TO **STEP 15**
- IF YOUR ANSWER IS **YES** GO TO **STEP 14**

**14. CHOOSE THE FORM YOU WANT TO TEST**

- Select the form you want to test

Here is an example of what you can find:

**NOTE**

By default the current page will be displayed if there is no action parameter in the form tag. This is anyway the default browser behavior.

**GO TO STEP 15****15. DO YOU WANT TO ADD, MODIFY OR REMOVE STRING OR COOKIE PARAMETER(S)?**

Once you loaded the web page it is now possible to add, modify or remove string or cookie parameters. In the case of the string parameters you might wish to play with them to do further tests. If it's what you want to do then your answer is **YES – GO TO STEP 16**.

As for the cookie parameters, you might add a cookie to keep the session context intact, such with a cookie keeping the current language. Or add an authenticated session cookie. If it's what you want to do then your answer is **YES – GO TO STEP 16**.

If for any other reasons you want to add, modify or remove string or/and cookie parameters your answer is **YES – GO TO STEP 16**.



If you don't want or don't need to add, modify or remove parameters or cookies your answer is **NO – GO TO STEP 17.**

#### NOTE

Before, the only way to modify parameters was to edit the XML file from the saved session.

#### HINT 1

Sometimes you will want to only test what happen if one string or cookie parameter is removed without entirely removing it. If it is your wish you just need to rename the parameter and it will be still be sent but with no meaning to web application. In the case of cookies, you can just uncheck the Use button.

#### HINT 2

With this new added features you can test secret parameters, such as admin=1 or debug=true.

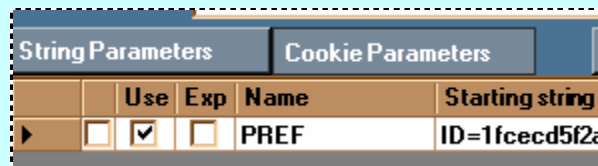
- IF YOUR ANSWER IS **NO** GO TO **STEP 17**
- IF YOUR ANSWER IS **YES** GO TO **STEP 16**

### 16. ADD, MODIFY OR DELETE STRING OR COOKIE PARAMETER(S)

New to this version, you can add, modify or delete string or/and cookie parameters. Both string and cookie parameters are handled in the same way. That is why there will be no need to explain it two times.

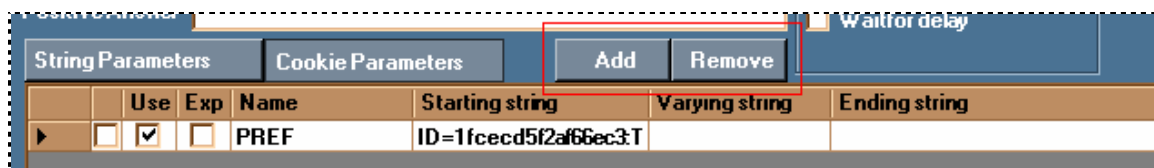
#### NOTE 1

In order to see the string or cookie parameters you just need to click on one of the button to toggle one from the other. You know it displays which parameter type by the fact that the button is pressed.



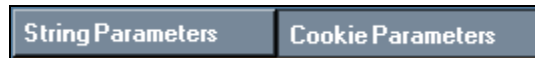
In that case it's the cookie parameters that are displayed.


Two new buttons have been added for this new feature.



### Add string or cookie parameter

1. First, select the type of parameter you want to add by clicking on String Parameters or Cookie Parameters



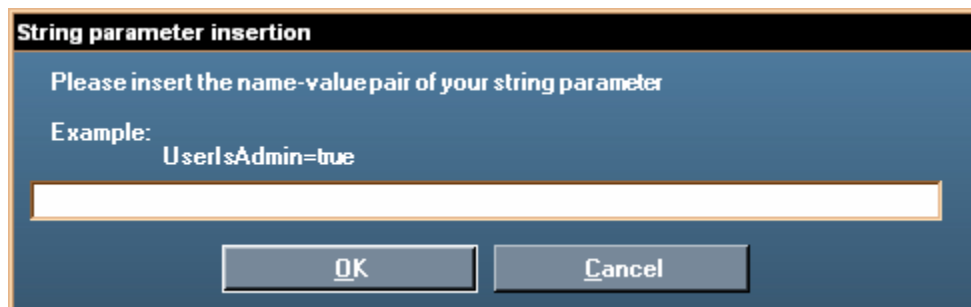
2. Then click on Add button  above the Datagrid.
3. An input box will be displayed, just paste the string or cookie value there and click ok.

The format should look like this in both cases: *name=value*

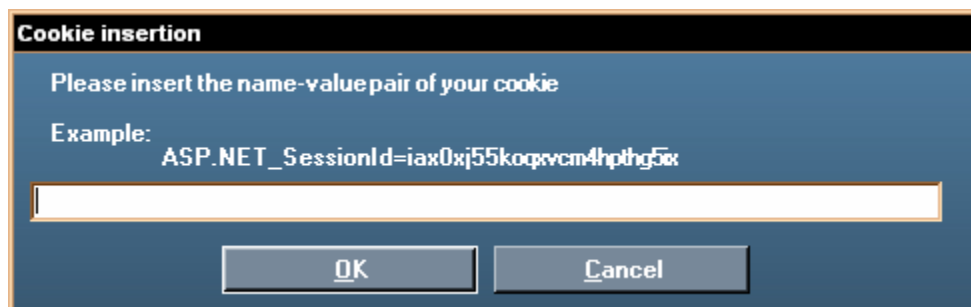
**Example 1:** Admin=true

**Example 2:** JSP session  
*JSESSIONID=D23TfhU3fdf7884HDSA45hfdGs*

String parameter input box:

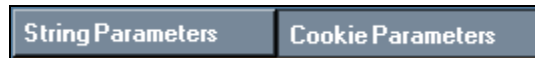
A dialog box titled 'String parameter insertion' with a black header bar. The main area has a blue gradient background. It contains the text 'Please insert the name-value pair of your string parameter' and 'Example: UserIsAdmin=true'. Below this is a white text input field. At the bottom are two buttons: 'OK' and 'Cancel'.

Cookie parameter input box:

A dialog box titled 'Cookie insertion' with a black header bar. The main area has a blue gradient background. It contains the text 'Please insert the name-value pair of your cookie' and 'Example: ASP.NET\_SessionId=iax0xj55koqxvcm4hpthg5x'. Below this is a white text input field. At the bottom are two buttons: 'OK' and 'Cancel'.

### Modify string or cookie parameter

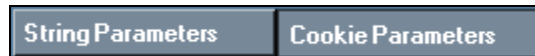
1. First, select the type of parameter you want to add by clicking on String Parameters or Cookie Parameters



2. Then directly edit inside the Datagrid the chosen name or value.

### Remove string or cookie parameters

1. First, select the type of parameter you want to add by clicking on String Parameters or Cookie Parameters



2. Then select one or more parameters by clicking at the left of the row. If you want to select more than one you can use the default multi Microsoft selection feature, that is to say maintaining Shift or Control when clicking on the selected parameter.

String Parameters			Cookie Parameters		Add	Remove	
	Use	Exp	Name	Starting string	Varying string	Ending string	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	toto	white' and	@servername	)--	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	ASPSESSIONIDS	HPMJNJOBPFLMOAJD			
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	test	asdasd	asds	dddd	

3. When you are ready, click on the Remove button



#### **NOTE 2**

Know that if you press Add or Remove button it will work in the context of the displayed Datagrid. Let's say that String Parameters has been pressed if you click on Add or Remove button it will add or remove a string parameter.

### **GO TO STEP 17**

### **17. DO I KNOW IF THE PAGE HAS SQL INJECTION?**

If you already know that there are SQL injections in that page then your answer is **YES**  
– **GO TO STEP 20.**

If you don't know then your answer is **NO** – **GO TO STEP 18**.

**HINT 1**

If you don't mind to be detected or blocked you can always use an automated tool to scan the whole web application to see if there are easy to find SQL injections. Usually, those scanners of vulnerabilities are not extremely good to find subtle SQL injections, but at least they can search several hundreds of page fairly easily for you. Where you might give up after ten or so, that application will do all of them.

However, I always rely on myself and it's where SQL Power Injector comes handy.

**HINT 2**

You can always make the rough job of searching the SQL injection inside Mozilla Firefox and once you ready you can use the SQL Power Injector Plugin to launch the vulnerable web page with SQL Power Injector to ease its exploitation.

- IF YOUR ANSWER IS **NO** GO TO **STEP 18**
- IF YOUR ANSWER IS **YES** GO TO **STEP 20**

**18. TEST LOADED PARAMETERS OR/AND COOKIES ONE BY ONE**

This step is one of the most important given that it's the one where you will actually search for the existence of SQL injection vulnerabilities.

There are several ways to find them and I'm not going to list all of them since the aim of this tutorial is not to have a crash course or white paper on SQL injection but more on how to use the application in order to optimize the search. Nonetheless, I will give some syntax examples in the hint boxes that could possibly help.

Here is what you need to know for how you can find them with the application. We stated that once the page has been loaded it will discover all the inputs that you can inject depending of the method (GET or POST) and fill up a Datagrid.

**NOTE**

You don't need to bother with the encoding of the strings if it's in the URL (for GET method) or in the input objects. SQL Power Injector will automatically encode them in the right format to be understood on the web server. Sometimes this encoding will even at the same time bypass IDS, Reverse Proxies or any filtering mechanisms!

In this example we have two parameters: login and pass. We can start to test for injection right now. What we need to do is to insert the value we want to test inside the Starting string (shown up by the red box in the next figure).

String Parameters				
	Name	Starting string	Varying string	Ending string
▶	<input type="checkbox"/> login			
	<input type="checkbox"/> pass			

In the search of SQL injection we don't need to bother with the two other columns (Varying String and Ending String) for now, those will be used only for the blind SQL injection technique (it will be fully explained in the **STEP 25** and **SUB-STEPS 25**). The Starting String represents the same thing than a normal textbox. All you will type there will be sent as if you modified the parameter in the URL or INPUTS fields.

In this case, we will use the Normal technique. Make sure that it's selected before to go on.

Technique	
<input checked="" type="radio"/> Normal	<input type="radio"/> Blind

To test the values it's as easy as to fill them one by one, some of them or all of them and look at the result in status section both in the current string bar and the mini-browser in the bottom section. Once the parameter(s) that you want to test are filled, just click on the Start button.

Start
-------

Here we tested both the values on the web site that we know that there are SQL injection vulnerabilities with quotes o' and test' and we've got interesting results:

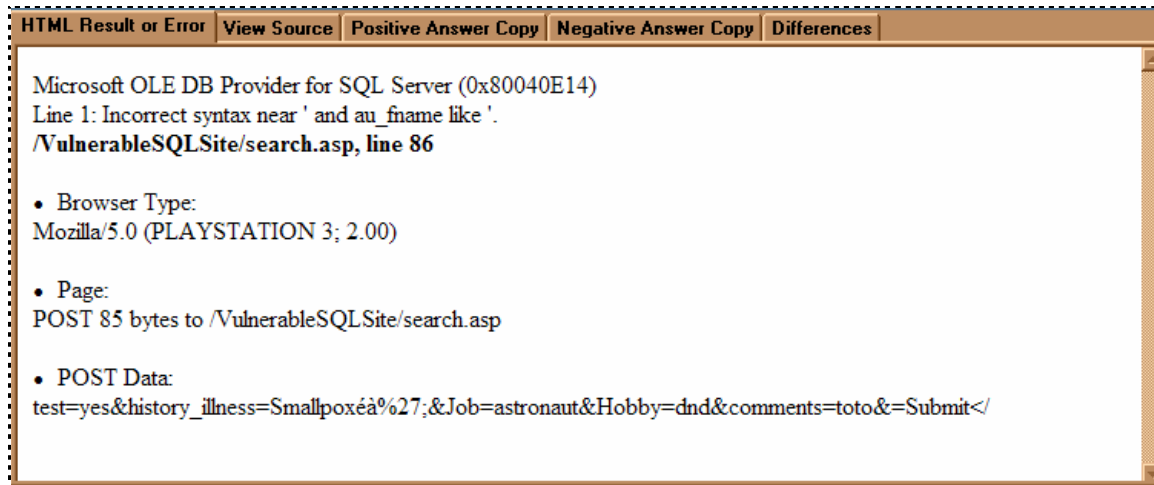
String Parameters		Cookie Parameters		Add	Remove
	Name	Starting string	Varying string	Ending string	
▶	<input type="checkbox"/> login	o'			
	<input type="checkbox"/> pass	test'			

First, in the current string result we can see there is an error 500. It's a good sign that there is something amiss and that SQL injection could be the cause. Notice that you have as well the exact syntax that is sent to the web server, it can be useful to know.

Status	
Current String	
▶	Error 500 - test=yes&history_illness=Smallpox&à%27;&Jo
HTML Result or Error View Source Positive Answer Conn	



Then the explicit result in the mini-browser:



We can plainly see that there is a SQL injection problem. That is one of the easiest ways to find out the existence of SQL injection since we have the explicit result in the response of the page. But it's not always like that...

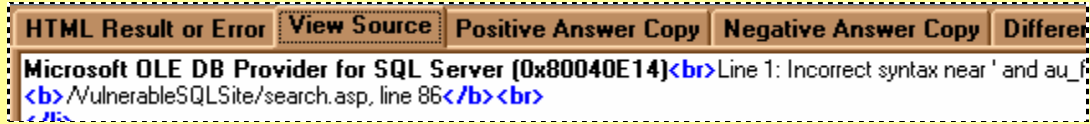
Sometimes you will get a nice page having a generic error message, such on the SQL Power Injector web site. ([www.sqlpowerinjector.com](http://www.sqlpowerinjector.com)) (see image below)



In that case you will need to do extra work and sometimes for nothing since the error could have been generated by a bad conversion of a string to an integer or because the modified value generates an application error. In some context it could be interesting for other kind of attacks, but here we cover the SQL injection.

**HINT 1**

Look also at the returned actual source text of the page in the View Source tab. Sometimes you will see information that will lead you to see that you succeed your injection.



In one of my pen test nothing was displayed in the returned page, but in the source code the ODBC error message was there between comment tags...

Other times the site might be vulnerable to SQL injections but some filtering mechanisms or any other kind of software or hardware will block them (IDS, Reverse Proxies, etc...) In those occasions you just need to be more clever and furtive.

**HINT 2**

Many web sites will rely their security on doubling the quote ( ' ) thus preventing the attack. It's pretty common in PHP since before the version 5 there were no ways to use the equivalent of the "stored procedure" or "prepared statement" in that technology.

The problem with that technique is that if you use a numerical parameter you don't even need to use the quote to succeed the injection! But even if we need one there are so many ways to bypass that problem, many great papers explain or at least hint at them.

Just a nice trick if you can't use quotes in the injection when comparing a string is to use its ASCII version for example:

```
SELECT Count(*) FROM MyUserTable WHERE login LIKE '%admin%'
```

Is the same thing than

```
SELECT Count(*) FROM MyUserTable WHERE login LIKE  
Char(37)+Char(97)+Char(100)+ Char(109)+Char(105)+Char(110)+Char(37)
```

Notice that there are no quotes anymore.

**HINT 3**

Some web sites will rely their security on really lousy rules such as to find some "dangerous" SQL command words and raise an error. (Union, Select, or 1=1, etc...)

Not to mention that some valid requests can be invalidated, considering that many of those words are usual English words, it's pretty easy to bypass those mechanisms when the attacker is aware of that feature.

The use of encoding or using comments /\* \*/ between the words or even putting more than one space can work pretty well. That's why there is an option to automatically put comments in the injection in the application.

☒ Replace Space by `' '`

#### HINT 4

Another nice trick is with a numerical value and to add 1 to it.

Let's say we have three parameters: Country, Language and NewsId. When we change the NewsId for 99 we get a different one. But what happen if we actually send to the server 99+1?

String Parameters		
	Name	Starting string
<input type="checkbox"/>	Country	England
<input type="checkbox"/>	Language	en-us
<input type="checkbox"/>	NewsId	99+1

If we get the same news as with the one with the NewsId=100 it means we found a SQL injection! Notice again that there is no use of quotes here...

**Note:** normally you would need to encode the + sign by %2B but the application, as explained earlier, encodes it for you.

#### HINT 5

A last trick is to use the **WAITFOR DELAY** technique. This technique will only work for SQL Server. So even if you're not sure at this point which DBMS it is you can always try it, it cannot harm and at the same deduce the server from it.

String Parameters		
	Name	Starting string
<input type="checkbox"/>	Country	England';WAITFOR DELAY '0:0:10'--
<input type="checkbox"/>	Language	en-us
<input type="checkbox"/>	NewsId	100

As soon as you clicked on Start button count the number of seconds elapsed and see if it's about the same than the one you inserted (10 seconds in this example).

If the web server respond after about the time set, voila! You found SQL injection!

**Note:** this technique will be further explained in the blind SQL injection section **STEP 25.6**

**GO TO STEP 19**

## 19. FIND SYNTAX

Once you have found the existence of SQL injection you can now work on the right syntax to get valuable information. However it is entirely possible you've got the right syntax in the previous **STEP 18**. A lot of examples can be found there in the hint boxes.

Depending of if you see the results in the returned page or not, the syntax will be completely different. Those conditions will be explained later and some hints of syntax will be explained accordingly to the situation.

**GO TO [STEP 23](#)**

## 20. DO I KNOW THE TYPE OF DBMS?

For now the application supports five DBMS: SQL Server, Oracle, MySQL, Sybase/Adaptive Server and DB2.

Although that it officially supports five DBMS, with the normal technique it's not really important. Because the values modified in the Datagrid are exactly the same sent (except for the encoding part of course) and nothing is added to make it dependent on a DBMS language. In other hand, with blind SQL technique we add some functions (Count, length, ASCII, substring functions) that are dependent to a DBMS language.

However, it's still good to know at this point which DBMS it is to be able to use the special commands or functions owned by each DBMS in your injection.

If you already know which DBMS it is then your answer is **YES – GO TO [STEP 21](#)**.

If you don't know then your answer is **NO – GO TO [STEP 22](#)**.

### HINT 1

Here is a pretty efficient trick to find out which DBMS it is. We assume that you found already a way to inject in the web page.

The **user** command: this command displays the current database user using the connection string

Try this: Select user in your injection (in a '**UNION SELECT user** or '**or 1 in (SELECT user)** for example)


If you don't get any error it means it's a SQL Server or Sybase. To see if it's either one just make the test with @@version and see what you get. At least at that point you have the choice between only two.

To see if it's MySQL use the same syntax but user with () after that is to say **user()** . If you don't get any errors then it's a MySQL server.

Finally to see if it's Oracle add **FROM DUAL** after your **SELECT USER** (without the parenthesis this time), if it works then it's an Oracle database. Oracle needs to always have the **FROM** statement in its **SELECT** statement.

## HINT 2

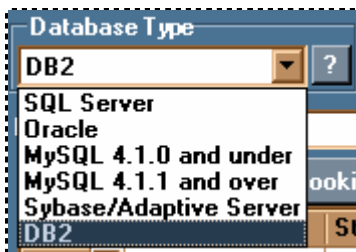
If you want to see more functions or system variables to help you to find out which DBMS it is you can use the new Databases "Aide Memoire" that contains specific information for each DBMS.

You can either choose one DBMS like **STEP 21** and then click on the question mark button  on its right or go on the ? menu and choose "Databases "Aide Memoire"".

- IF YOUR ANSWER IS **NO** GO TO **STEP 22**
- IF YOUR ANSWER IS **YES** GO TO **STEP 21**

## 21. CHANGE IT TO THE RIGHT ONE

This step is simply to change to the right one if you need so.



## NOTE

By default the application is set to SQL Server

GO TO **STEP 22**


## 22. DO I KNOW THE SYNTAX?

If you already know what is the syntax to inject then your answer is **YES** – GO TO **STEP 23**.

If you don't know then your answer is **NO** – GO TO **STEP 19**.

- IF YOUR ANSWER IS **NO** GO TO **STEP 19**
- IF YOUR ANSWER IS **YES** GO TO **STEP 23**

### 23. INSERT THE RIGHT SYNTAX IN LOADED PARAMETERS AND CLICK BUTTON START

- Insert the right syntax in the associated loaded parameters
- Click on Start button A rectangular button with a dark blue background and the word 'Start' in white text.

GO TO **STEP 24**

### 24. CAN I SEE THE RESULTS OR ERRORS IN THE WEB PAGE?

After you have clicked on the Start button, look at the resulting web page in the mini-browser. You might see a 500 error message like in the **STEP 18**. If it's the case then your answer is **YES – GO TO STEP 26**. Also see Hint 1 in the **STEP 18** to see if it's the case, if so then your answer is **YES – GO TO STEP 26**. Finally, the results can be seen in a **UNION** query, so that would make the answer **YES – GO TO STEP 26**.

If you don't see anything that could give you any clue on the data injected then your answer is **NO – GO TO STEP 25**.

#### **NOTE**

Perhaps the goal is not to see any data from the database and if it's the case you don't need to go forward. This scenario might happen if what you needed to do was to bypass the login page with a '**or 1=1--**' command.

But it's always interesting to see if there are hidden databases that could hold really sensitive data that are not linked to the current web application.

- IF YOUR ANSWER IS **NO** GO TO **STEP 25**
- IF YOUR ANSWER IS **YES** GO TO **STEP 26**

### 25. YOU NEED BLIND SQL INJECTION

This step has been drilled down to sub-steps to facilitate the understanding as you can notice as well in the schemas. It was created like this to alleviate the complexity of one step in a different schema where it can be exploded in a many more understandable sub-steps. Here we assume that the user will logically follow those steps in the reading order.

Unfortunately for you there are no results or errors that can give you any hint on the data injected. It means you'll have to do it in the hard way. It can be really time consuming and so many time spent to get nothing really valuable that many would abandon during the process. But now with SQL Power Injector the tedious repetitive tasks are automated.

As stated in good software design architecture, the menial tasks are done for you and you can concentrate on the business logic, in our case the SQL injection.

**NOTE**

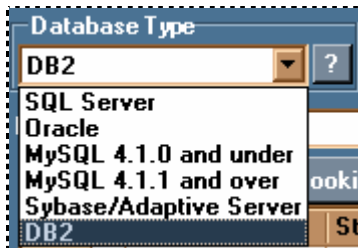
Once you selected the Blind technique you notice that most of the application changes with new textboxes and options. That's normal, it's just that I chose to show information accordingly to the context of what is used or not. I did improve this in the version 1.2 to group all blind SQL injection information together in a groupbox to make it easier to understand.

**GO TO [STEP 25.1](#)****25.1 FIND THE RIGHT DBMS THEN SELECT IT**

If you don't know which one it is you will need to find out. If in the **STEP 20** it wasn't that important to know here it is. Because the application will add all the SQL commands necessary to automate the injection with syntax that is proprietary to the DBMS.

You can use the same trick given in the Hint of **STEP 20**.

If you already know or just found out, change to the right one if you need to.

**GO TO [STEP 25.2](#)****25.2 IS GENERATING A REQUEST WITH ERRORS RISKS TO SEND AN EMAIL OR LOCK YOU OUT?**

Some web sites after a while will detect the attack attempts caused by the error and will block you out or slow you down. Or sometimes an Email will be sent out every time an error is generated. This occurrence will likely to happen when there is a nice web page stating that there is an error. (See my error page on **STEP 18**). But you never know, it could be any security software sending them. And of course, if an Email is sent every time you make an attempt you are more likely to have someone to react and block or slow you down after each time.

For any reasons, if you think it's risky to generate error on the web site your answer is **YES – GO TO [STEP 25.6](#)**.

If you don't see any risks then your answer is **NO** – **GO TO STEP 25.3**.

### IMPORTANT NOTE

Even if your answer is **YES** it doesn't mean using the Positive answer technique will end up generating errors every time. The odds are fairly high but if you can find a situation where the negative answer doesn't generate an error, well you can use that technique without fear, it is actually much faster than the Time Delay technique (it's what happens when you go to **STEP 25.6**). Those two techniques will be described more in details later.

- IF YOUR ANSWER IS **NO** GO TO **STEP 25.3**
- IF YOUR ANSWER IS **YES** GO TO **STEP 25.6**

## 25.3 TRY TO FIND A POSITIVE ANSWER WITH AN INJECTION

I will explain what I mean by positive answer. It's a special technique that I'm not quite sure it has been documented before but in the context of SQL Power Injector it comes really handy to master.

Your goal is to make the response of your injection to be different when a condition is true and when a condition is false. The difference can be in the text or can be in the cookie, it doesn't matter. First, I will demonstrate the theory and after I will show how to make your life much easier with a new feature that will find the differences for you.

Let's use this example. We have a search page that displays an author list (from the Pubs SQL Server database) depending on the search criteria. We found that there is an SQL injection error but unfortunately there is a nice page Error Page with no information whatsoever. And worse, there is no way to use the **UNION** command.

1. We try to find a way to SQL inject the condition without raising an error. We succeed with '**and 1=1--**'

String Parameters		Cookie Parameters
	Name	Starting string
▶	<input type="checkbox"/> txtSearch	'and 1=1--

And we get this information in the mini-browser:

HTML Result or Error	View Source	Positive Answer Copy	Negative Answer				
<h2>Recherche d'auteur par nom de</h2> <p>' and 1=1--</p> <p>Submit</p> <p>' and 1=1--</p> <p>No result found!</p> <table border="1"> <thead> <tr> <th>ID</th> <th>First Name</th> <th>Last Name</th> <th>Pho</th> </tr> </thead> </table>				ID	First Name	Last Name	Pho
ID	First Name	Last Name	Pho				



- We try to see what happen with ' and 1=2-- and search for the difference. We get this information:

HTML Result or Error
View Source
Positive Answer Copy
Negative Answer Copy

## Recherche d'auteur par nom de famille



' and 1=2--  
No result found!

ID	First Name	Last Name	Phone	City
----	------------	-----------	-------	------

The same information! Even in the View Source there are no differences.

- Let's try to inject it with a real value, let's use **white' and 1=1--**.

Status  
Current String  
▶ Is false - txtSearch=white%27+and+1%3D1--&Submit1=Submit [Cookies]ASPSESSIONIDAQCATBQB=NPBDFC

HTML Result or Error
View Source
Positive Answer Copy
Negative Answer Copy
Differences

## Recherche d'auteur par nom de famille



white' and 1=1--

ID	First Name	Last Name	Phone	City
172-32-1176	Johnson	White	408 496-7223	Menlo Park

We have two interesting information: the Current String states it is at "Is false" and we get a result with this injection.

4. Now we want to try with the negative condition **white' and 1=2--**. Let's see what we get:

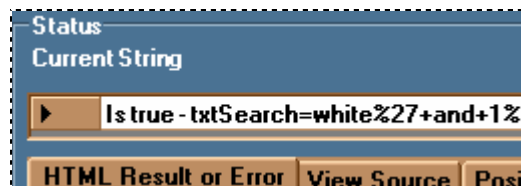


This time we have most definitely something different! Again notice that the Current String value states it is at "Is false"

5. We compare the two results and try to find text in the positive condition that you are sure that won't be on the negative one. We decided to use Johnson after we searched that word in the View Source section of the negative condition and found none.
6. We insert it in the Positive Answer textbox.



7. We click on Start again and now we can see that in the Current String we have "Is true"



You're done! You found one.

**NOTE 1**


There are various ways to find a positive condition and the '**and 1=1--**' is one of them. Sometimes I succeed with '**or 1=1--**' as well. Also you might have to add some parenthesis or extra special characters in order to succeed it. Every situation has its own syntax.

Here is the good news! There is a new feature that will search the differences for you as mentioned at the beginning of this step.

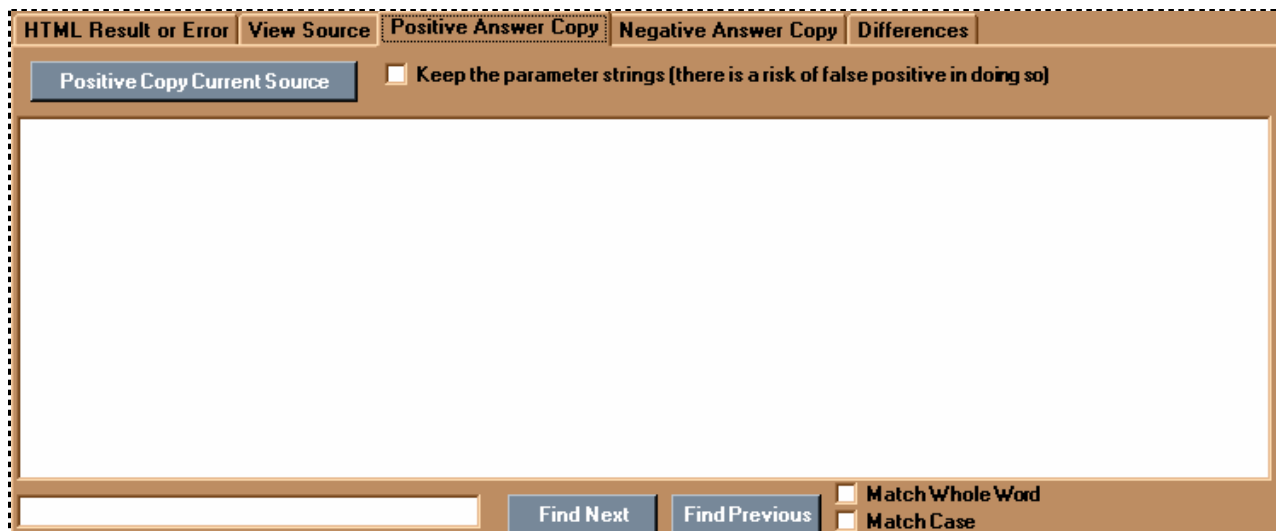
Here is how you need to do:


Let's keep the same example than before.

1. We skip step 1 and 2 and go directly to step 3 in the previous example and insert **white' and 1=1--**

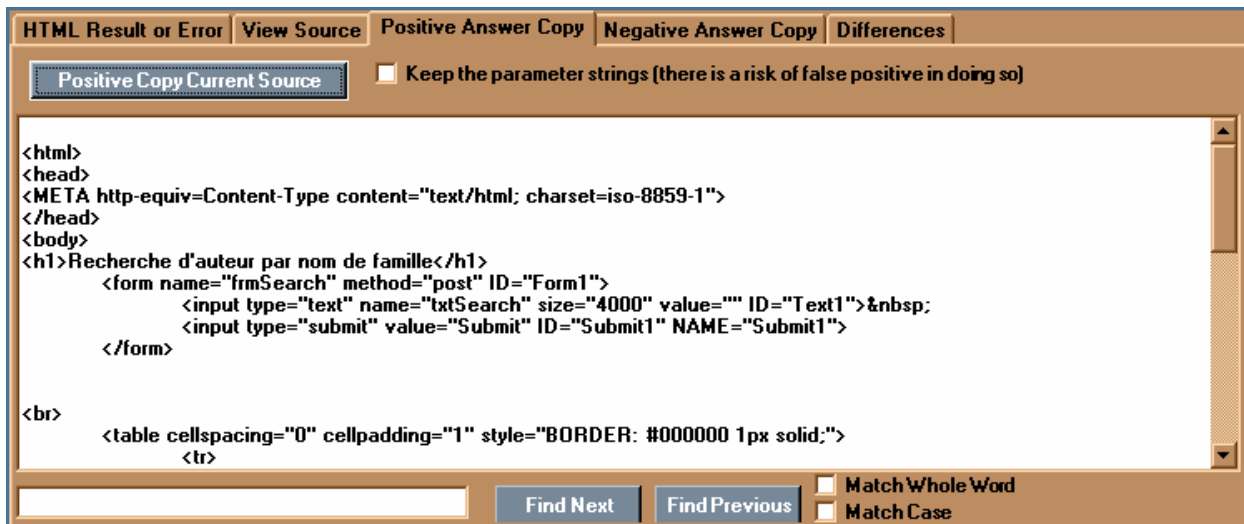
2. Click on Start button 



3. Then click on the Positive Answer Copy tab

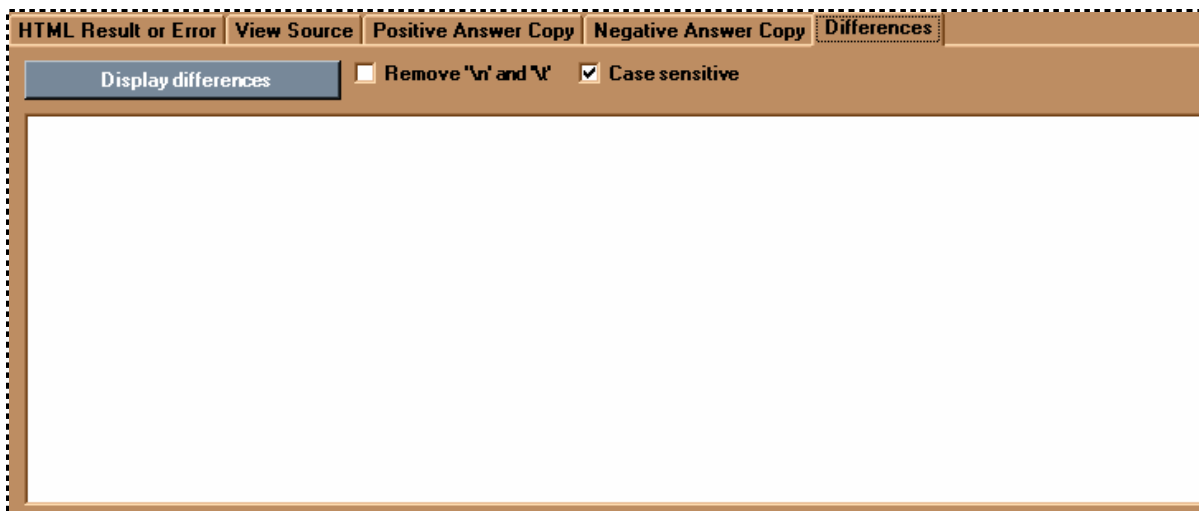



4. Click on Positive Copy Current Source button  to copy inside that box the source code of resulting page

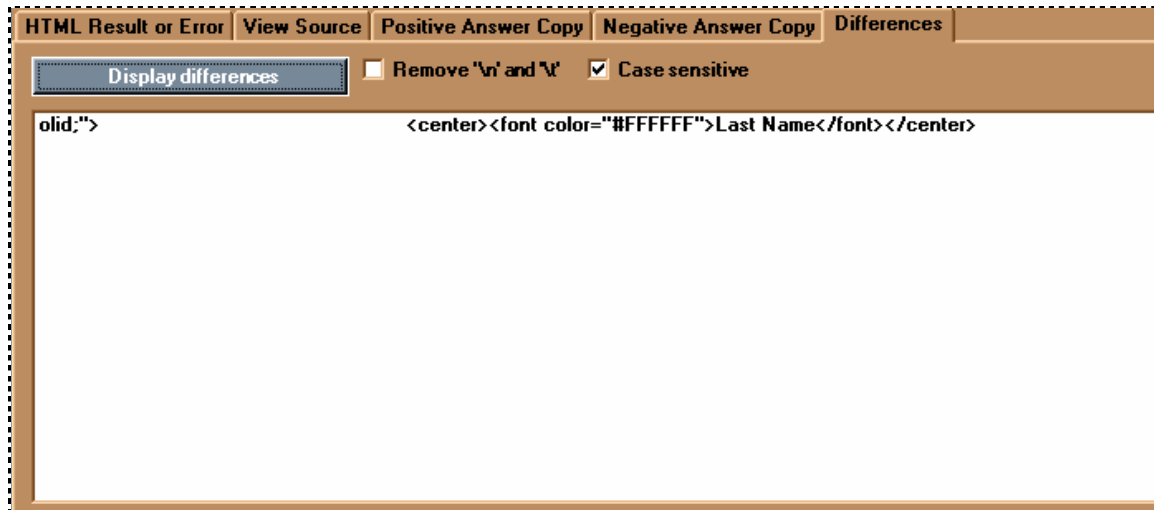
It should give you this result:



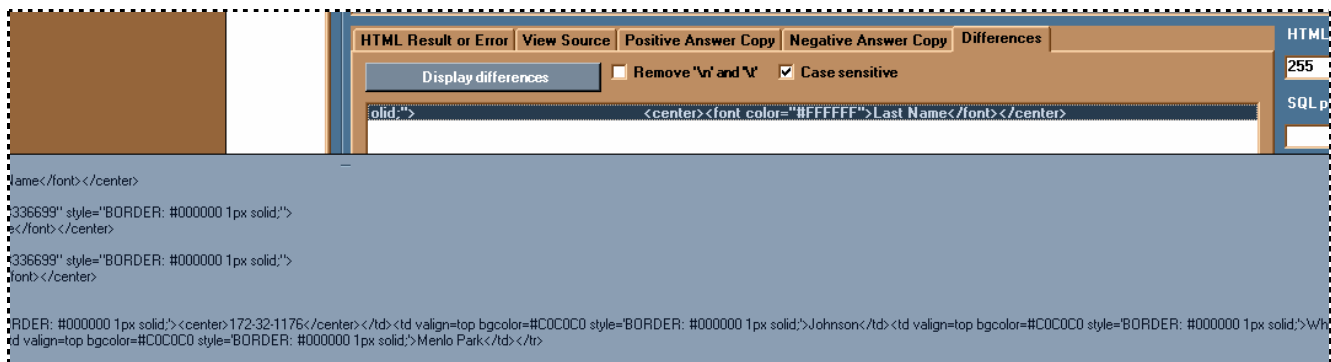
5. Now inject it with a negative answer, let's use **white' and 1=2--**
6. Click again on Start button 
7. This time click on Negative Answer Copy tab (No screen has been provided since it's exactly the same previous screen except that the Positive has been replaced by Negative)
8. From there click on Negative Copy Current button  to copy inside that box the source code of resulting page (No screen has been provided since it's exactly the same previous screen except that the Positive has been replaced by Negative)
9. After, click on Differences tab



10. Click on Display differences button  and wait for the message box telling you if it found differences or not. If it didn't find any try to modify your injections. If you did like in this example, go on the next step




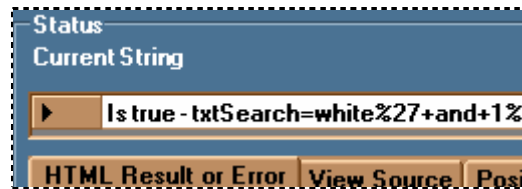
11. You should see one or several results (one by row). In that example there is only one, so it will be easy to test. If you want to see the whole results in that row just hover above it.




12. To test it just double click it and it will be automatically copied in the Positive Answer textbox



13. From there, you need to make sure it's not a false positive (unfortunately it might happen but hopefully you have enough results to find at least one that works). To do so, reinject **white' and 1=1--**, click on Start button  and look at the Current String. If it says "Is true" then you half way there.



14. Now, check with **white' and 1=2--**, then click on Start button  and look again at the Current String. If it says "Is false" then you found your string! If not try the step 12 to 14 again with the next in list (if any). If none, then you can always try to find it yourself. (Read the Important Note below)

## NOTE 2

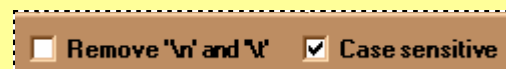
You might have noticed that there is a checkbox "Keep the parameters strings" inside the Positive Answer Copy and Negative Answer Copy tab. By default it is not checked for the simple reason that it might result in a false positive result when the algorithm is trying to find the differences.



Like in the first example the injection string **'1=1--** or **'1=2--** are displayed back to the screen. So, when it comes the time to find a difference this will obviously always find a difference...

## HINT

Inside the Differences tab you might have noticed that there are two checkboxes:



The first one, Remove '\n' and '\t', might be handy when you get a lot of false positives results. In that case, you can try to check and uncheck it and see if there is a difference in the results.

The second one, Case sensitive, is by default checked since that sometimes differences might dwell only by that small thread. However, it might result in many false positives and to uncheck it could make it more accurate.

**IMPORTANT NOTE**

The algorithm is not perfect and might not find all the subtlety that you can find. It's more to help you find some tedious hard to find differences that it was added. In some cases it might find differences that you wouldn't find and other cases you'll find some that it won't. I suggest to always use it by default since it's pretty quick and easy to use and then if it does not find anything look by yourself.

In any cases, the fact that you copied the positive and negative response in a different place with a search capability will help you anyway.

**GO TO [STEP 25.4](#)****25.4 DID YOU FIND A DIFFERENCE WITH THE TWO CONDITIONS?**

If you found your Positive Answer value then your answer is **YES** – **GO TO [STEP 25.11](#)**.

If you didn't find any difference in the text or cookie then your answer is **NO** – **GO TO [STEP 25.5](#)**.

- IF YOUR ANSWER IS **NO** GO TO **[STEP 25.5](#)**
- IF YOUR ANSWER IS **YES** GO TO **[STEP 25.11](#)**

**25.5 YOU MUST USE TIME DELAY TECHNIQUE**

You are here because you haven't found any Positive Answer in the **STEP 25.3**, but all is not lost because you can switch to Time Delay technique. The only thing is that it's much slower.

**GO TO [STEP 25.6](#)****25.6 TEST FOR TIME DELAY TECHNIQUE**

This technique as the name implies is to see if there are ways to inject time delay at the DBMS server side. It's a special technique that makes the current connection to the DBMS server to hang to the time set. You can see that you succeeded when the web page is displayed after the time you set as the delay.

**NOTE**

Most of the time it's roughly one second more than the time set because the page needs time for the round trip and the execution on the server side.

This technique is really useful when you don't have any result back in the return web page or when you want to make sure that you won't create any error on the Server side while in the automatic mode.

It can be really hard to exploit if you are not using SQL Server or Sybase. Because in Oracle you need to be inside a BEGIN and END clause in order to inject the semi-colon ( ; ) and then the sleep(sec) function. It happens really rarely.

MySQL is much subtler than Oracle and SQL Server, a method has been found with the function BENCHMARK that can create a delay depending on the number set. It is unfortunately not that reliable in SQL Power Injector, although it does work.

**HINT**

I will show an example for each DBMS for the time delay with a delay of 5 seconds.

**SQL Server or Sybase:**

*InjectedValue'; WAITFOR DELAY '0:0:5'--*

**ORACLE:**

*InjectedValue'; BEGIN DBMS\_LOCK.SLEEP(5); END;*

**MySQL:**

*InjectedValue OR IF (1=1, BENCHMARK(500000, MD5(CHAR(1))), null)*

**DB2:**

*No delay mechanism implemented*

**GO TO STEP 25.7**

**25.7 IS THE TIME DELAY TECHNIQUE WORKING?**

If you succeed to make it work then your answer is **YES** – **GO TO STEP 25.9**.

If you didn't make it work then your answer is **NO** – **GO TO STEP 25.8**.

- IF YOUR ANSWER IS **NO** GO TO **STEP 25.8**
- IF YOUR ANSWER IS **YES** GO TO **STEP 25.9**



## 25.8 YOU MUST USE POSITIVE ANSWER TECHNIQUE

You are here because you didn't make the time delay work in the **STEP 25.6**, but all is not lost because you can switch to Positive Answer technique. The good news is that it's much faster. However it is possible that you might end up generating errors.

**GO TO [STEP 25.3](#)**

## 25.9 INSERT THE TIME DELAY, THEN COUNT THE NUMBER OF SECONDS TAKEN

Chances are that you have already inserted the time delay in the Starting String to test it out. If you didn't it's time to do so.

Now right after you clicked the Start button count the number of seconds taken, yes it might seem odd since we are the one to set the delay but in some rare occasions the injection you're doing might be concatenated in several places. So, knowing that possibility we need to count and when we have the number we have to do a simple math calculation.

If the delay set is 5 seconds and it took around 15 seconds, how many concatenations occurred?

5 seconds for normally 1  
15 seconds for ?

**( 15 \* 1 ) / 5 = 3 places**

That information could be useful to know, but in our case only the number of seconds was enough. It's the number you need for the parameter "Delay in seconds".

**GO TO [STEP 25.10](#)**

## 25.10 CHECK WAITFOR DELAY OPTION AND INSERT THE NUMBER YOU GOT IN 25.9

- Check WAITFOR DELAY option



- Set the number of seconds counted (can be different of the one you set)



### HINT

You can fine tune the speed of the results by lowering the number in "Delay in seconds" by 1 each time and the one you set in the Ending String until you get a lot of false positive. Generally 3 seconds is the best.

## GO TO [STEP 25.11](#)

### 25.11 SELECT BLIND TECHNIQUE. INSERT VALUES TO INJECT IN PARAMETERS AND CHECK THE PARAMETER THAT HAS THE INJECTION

You have to select the Blind technique before to continue.



Now we are in a major piece of SQL Power Injector that is more complex than the rest of the application but once understood it's pretty easy to use.

The main part to be understood in that technique is the concept of the three strings portion. These three text insertion values are the main core of what makes the application powerful.

So far you have found SQL injection and you know which field or row will be used to inject SQL. It's the one you need to check and at that moment the application will use the three text insertion values that you inserted.

Let's say we have five parameters: Country, txtSearch, Language, jscrip and css. And we discovered that there is a SQL injection vulnerability with txtSearch. So we will check that parameter.

String Parameters	Cookie Parameters	Add	Remove	
Name	Starting string	Varying string	Ending string	
<input type="checkbox"/> Country	Canada			
<input checked="" type="checkbox"/> txtSearch	SQLInjectionFoundHere'			
<input type="checkbox"/> Language	en-us			
<input type="checkbox"/> iscript				

Checking this parameter will add according to the type chosen (Word, Length or Count) some strings after the Starting String (where it gets its name from) and before the Varying String and finally after the Varying String and before the Ending String (where it gets its name from as well).

## IMPORTANT NOTE

If you found SQL injection inside a cookie and this cookie is in fact a collection (**Ex:** CookieColName=Element1=valueEle1&Element2=valueEle2&Element3=valueEle3) then you might have to do an extra step before you click on start button.

If your injection is not at the end of the string, in the last example Element3, then you will have to reorder the cookie element to make the element with injected string at the end of the starting string. In doing so remember that the element separator in a cookie collection is the ampersand (&), make sure you separate the *name=value* with the ampersand (&)

### Example:

Found a SQL injection inside the ele2 in the cookie CookieCol. If we try to run this it will crash and no result will be obtained

String Parameters		Cookie Parameters		Add	Remove	FullSet	
Use	Exp	Name	Starting string	Varying string	Ending string		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	ASPSESSIONID	NAFNMFLCLFKIIDP00JKH8FOB				
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CookieCol	ele1=toto&ele2=white' and &ele3=lala	user	]--		

However, if we move this element to the end it will work perfectly

String Parameters		Cookie Parameters		Add	Remove	FullSet	
Use	Exp	Name	Starting string	Varying string	Ending string		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	ASPSESSIONID	NAFNMFLCLFKIIDP00JKH8FOB				
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CookieCol	ele1=toto&ele3=lala&ele2=white' and	user	]--		

Notice that the ampersands (&) separate ele1 from the ele3 and ele3 from ele2.

Why, we need to do this? The reason is because SQL Power Injector appends the rest of the crafted injection after the Starting String and is not able to make the difference between the elements in the cookie collection. So instead to append to ele2, it will append on the ele3.

Let's use the same parameters to demonstrate it with an example. For example we want to get the IP address of the server hosting an Oracle DB with the Positive Answer technique:

Options:

- **Database type:** Oracle

Database Type

- **Mode:** Blind

Technique

☐ Normal ☒ Blind

- **Type:** Word

Type

☒ Word ☐ Length ☐ Count

- **Starting length:** 30

Starting Length

- **Number of threads:** 8

Number of Threads

- **Positive answer textbox:** CLERK

Positive Answer

String Parameters	Cookie Parameters	Add	Remove	AlphabetLowerCa
Name	Starting string	Varying string	Ending string	
<input type="checkbox"/> Country	Canada			
<input checked="" type="checkbox"/> txtSearch	SMITH'AND	SYS_CONTEXT('USERENV','IP_ADDRESS')	FROM DUAL)--	
<input type="checkbox"/> Language	en-us			

Resulting in this SQL statement:

```
SMITH' AND 30 < (SELECT ASCII(SUBSTR(CAST(SYS_CONTEXT('USERENV',
'IP_ADDRESS') AS VARCHAR(4000)), 1, 1)) FROM DUAL)--
```

Legend:

Blue: Starting string  
Red and italic: Automatically added strings  
Purple and bold: Varying string  
Green: Ending string

As you can see many strings are added automatically and some value will change until it gets the full IP address value of the previous example.

You will have this result with all the requests sent:

Status
Current String
Thread 4: I: true - Country=Canada&txtSearch=SMITH%27+AND+46%3D%28select+ASCII%28SUBSTR%28CAST%28SYS_CONTEXT%28%27USERENV%27,%27P_ADDR
Thread 3: I: true - Country=Canada&txtSearch=SMITH%27+AND+57%28select+ASCII%28SUBSTR%28CAST%28SYS_CONTEXT%28%27USERENV%27,%27P_ADDR
Thread 2: I: false - Country=Canada&txtSearch=SMITH%27+AND+49%28select+ASCII%28SUBSTR%28CAST%28SYS_CONTEXT%28%27USERENV%27,%27P_ADDR
Thread 1: I: false - Country=Canada&txtSearch=SMITH%27+AND+49%28select+ASCII%28SUBSTR%28CAST%28SYS_CONTEXT%28%27USERENV%27,%27P_ADDR
Thread 8: I: false - Country=Canada&txtSearch=SMITH%27+AND+45%28select+ASCII%28SUBSTR%28CAST%28SYS_CONTEXT%28%27USERENV%27,%27P_ADDR
Thread 3: I: true - Country=Canada&txtSearch=SMITH%27+AND+55%3D%28select+ASCII%28SUBSTR%28CAST%28SYS_CONTEXT%28%27USERENV%27,%27P_ADDR
Thread 2: I: true - Country=Canada&txtSearch=SMITH%27+AND+50%3D%28select+ASCII%28SUBSTR%28CAST%28SYS_CONTEXT%28%27USERENV%27,%27P_ADDR
Thread 1: I: true - Country=Canada&txtSearch=SMITH%27+AND+49%3D%28select+ASCII%28SUBSTR%28CAST%28SYS_CONTEXT%28%27USERENV%27,%27P_ADDR
Thread 8: I: true - Country=Canada&txtSearch=SMITH%27+AND+46%3D%28select+ASCII%28SUBSTR%28CAST%28SYS_CONTEXT%28%27USERENV%27,%27P_ADDR
Thread 8: I: true - Country=Canada&txtSearch=SMITH%27+AND+255%28select+ASCII%28SUBSTR%28CAST%28SYS_CONTEXT%28%27USERENV%27,%27P_ADDR
Thread 8: I: true - Country=Canada&txtSearch=SMITH%27+AND+127%28select+ASCII%28SUBSTR%28CAST%28SYS_CONTEXT%28%27USERENV%27,%27P_ADDR

And now you want to get the current user name? Easy! You just need to change the Varying String to USER. As you can see once the syntax is found, the rest is just a matter to change some value.

Subsequently you can construct your SQL string to inject with the specific commands you want and it will fill in the information that is repetitive if you are in blind mode.

I discovered with time that when I was building my SQL strings to inject that the syntax could be broken in three main portions. And if some value can change as you have seen in the example, the one that always revolve around is the Varying String.

## HINT

You can use your mouse to hover a row to get the real value that will be sent to the web server. It is really useful when you are confused with what is automatically injected. Also, it's a good way to see in one glimpse your SQL statement.

String Parameters	Cookie Parameters	Add	Remove	AlphaGetLowerCa	Edit	Use
<input type="checkbox"/> Country	Canada					
<input checked="" type="checkbox"/> txtSearch	SMITH'AND			SYS_CONTEXT('USERENV','IP_ADDRESS')		FROM DUAL)--
<input type="checkbox"/> Language	en-us					
Results: txtSearch=SMITH'AND 100>(SELECT ASCII(SUBSTR(CAST(SYS_CONTEXT('USERENV','IP_ADDRESS') AS CHAR(4000)),1,1)) FROM DUAL)--						

## GO TO STEP 25.12

### 25.12 YOU WANT TO GET MORE THAN ONE VALUE AT THE TIME?

You keep using the same syntax for your injection except for a variable segment? Wouldn't it be nice if you could use a variable that could be replaced by a predetermined list of values? If that is your problem then your answer is **YES – GO TO STEP 25.13.**

If you just need to get a unique value at the time or it is impossible to get more than one value with a variable segment then your answer is **NO – GO TO STEP 25.14.**

Here are some examples of variable segments that can be useful to use with the multi-values feature:

**Example 1:** Getting each database value one by one with **dbid** changing each time

```
SELECT name FROM master..sysdatabases WHERE dbid=1
SELECT name FROM master..sysdatabases WHERE dbid=2
SELECT name FROM master..sysdatabases WHERE dbid=3
Etc...
```

**Example 2:** Getting different environment values from MySQL

```
SELECT DATABASE()
SELECT USER()
SELECT @@version_compile_os
SELECT @@basedir
SELECT @@version
```

**Example 4:** Brute force the sa password with OPENROWSET function

```
SELECT * FROM OPENROWSET('SQLOLEDB','localhost','sa';test','SELECT TOP 1 *
FROM pubs..authors')
SELECT * FROM OPENROWSET('SQLOLEDB','localhost','sa';admin','SELECT TOP 1
* FROM pubs..authors')
SELECT * FROM OPENROWSET('SQLOLEDB','localhost','sa';blank','SELECT TOP 1 *
FROM pubs..authors')
SELECT * FROM OPENROWSET('SQLOLEDB','localhost','sa';sa','SELECT TOP 1 *
FROM pubs..authors')
Etc...
```

- IF YOUR ANSWER IS **NO** GO TO **STEP 25.14**
- IF YOUR ANSWER IS **YES** GO TO **STEP 25.13**

## 25.13 USE MULTI-VALUES FEATURE

This new feature can not only save you time and automate tiresome copy-paste SQL syntaxes but also acquire information that you would not be able without it. For instance the example 4 in **STEP 25.12** is a good example of one of the many tricks you can achieve. After, it is entirely up to you to find original ways to optimize this feature.

Something else worth mentioning is the fact that the way you can interpret the results can give you some insights. For example, if you see the typical *"There is a problem with your injection!"* for a given result it could mean that it does not exist or you don't have the rights to access that data for example. With usage you will end up to understand and be able to optimize it. Also, the beauty of it is if a variable brings an error for any mentioned reason or any other the application will only make 2 requests to figure it out. So if you want to use a dictionary attack it will be only 2 requests for any wrong results. Not too bad...

Now let's explain how to use this feature:

1. First you need to find the emplacement where you want to replace the variables and insert this special string <<@>>

String Parameters			Cookie Parameters		Add	Remove		
	Use	Exp	Name	Starting string	Varying string	Ending string		
	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ASPSESSIONID	NAFNMFLCLFKIIDP00JKHBF0B				
	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Col	ele1=toto&ele3=lala&ele2=white' and	<<@>>	)--		

### NOTE 1

You can insert variable any places you want, it can be inside the Starting String, the Varying String or the Ending String.

You can also insert any number of variable (<<@>>) you want, again remember it will be replaced only by the current variable in the list. But remember in both cases to respect the SQL syntax.

2. After you need to check the Use Variable Range checkbox



3. Then, click on Edit button  to display the Variable Range Editor

**Variable Range Editor**

Generated Range

Numerical range
1
to
1
Add

Range from text file
Add

Range List

Number of values

0

Clear

Sort

Remove duplets

Note: Remove duplets might take several minutes to process

Close

4. Now you have to choose what kind of variable you want to use. If you need to use a numeric range to get each column from a given table you can use **Option A**.

If you want to get values from a dictionary to brute force a password for example, use **Option B**.

If you want to edit yourself the values use the **Option C**.

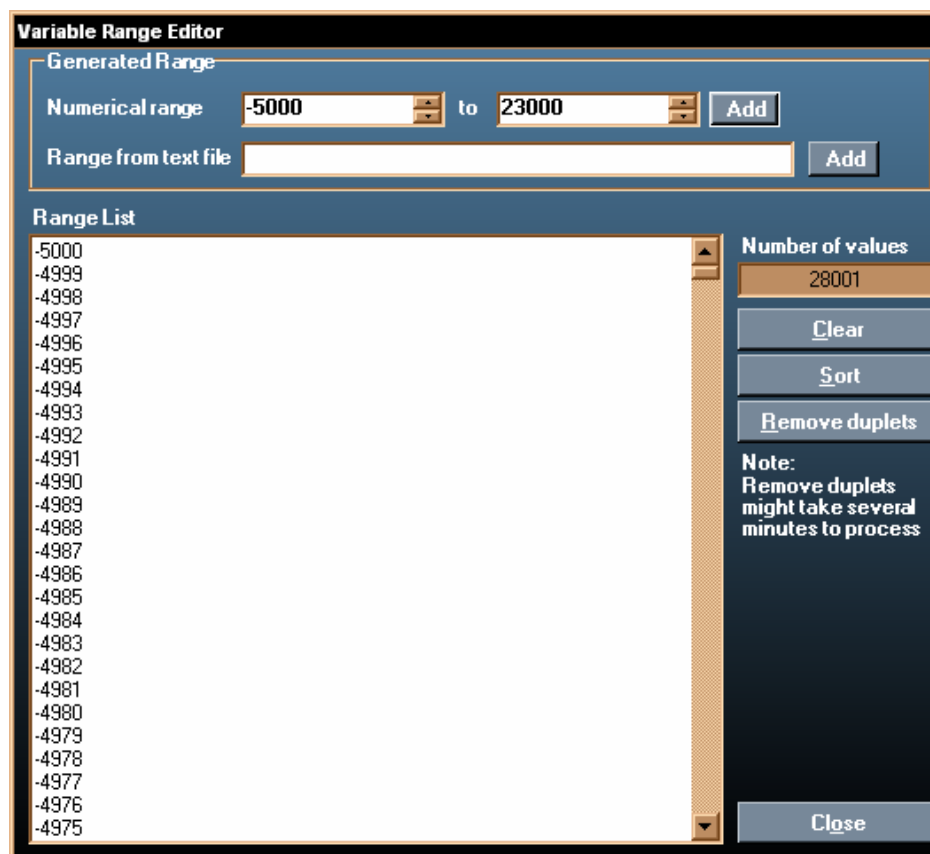
### Option A: Numeric Range



Generated Range

Numerical range  to

You can select the range with a *from* number to a *to* number. Those numbers can be negative. Then when you are ready just click the Add button



Variable Range Editor

Generated Range

Numerical range  to

Range from text file

Range List

-5000
-4999
-4998
-4997
-4996
-4995
-4994
-4993
-4992
-4991
-4990
-4989
-4988
-4987
-4986
-4985
-4984
-4983
-4982
-4981
-4980
-4979
-4978
-4977
-4976
-4975

Number of values

Note:  
Remove duplets  
might take several  
minutes to process



### Option B: Range from a text file

Range from text file
Z:\SPInj\Version 1.2\EnvVariablesMSSQL.txt
Add

You can select the file by clicking Add button **Add** and it will automatically add it to the list

**Variable Range Editor**

Generated Range

Numerical range
1
to
1
Add

Range from text file
Z:\SPInj\Version 1.2\EnvVariablesMSSQL.txt
Add

Range List

db\_name()  
IS\_MEMBER('dbo')  
user  
APP\_NAME()  
@@SERVERNAME  
HOST\_NAME()  
@@SERVICENAME  
SERVERPROPERTY('ProductVersion')  
SERVERPROPERTY('Edition')  
@@options  
@@language

Number of values  
11  
Clear  
Sort  
Remove duplets  
Note:  
Remove duplets  
might take several  
minutes to process  
Close

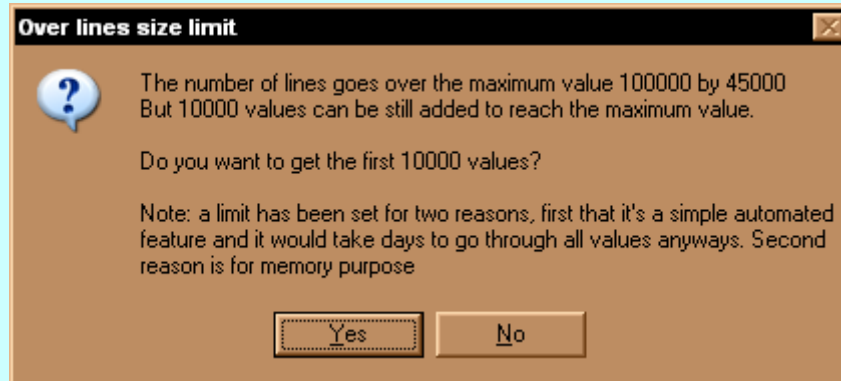
### Option C: Manually

Just edit the Range List window with the values you want

## NOTE 2


The number of variables in the list is limited to 100000. It should be sufficient for most of all the use you can get. The reason why there is a limit is simple, it is a memory limitation and besides will you really request at least 200000 requests on a Web Server?

In case you go beyond that point the application will nicely tell you that you have reached the maximum of 100000 and by how many. And if you didn't reach that maximum before it will offer you to add those that could starting with by the firsts.



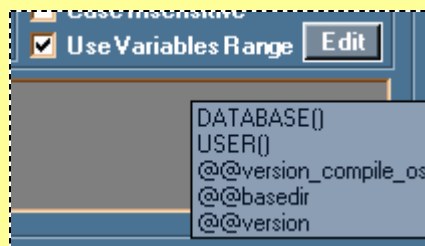
## NOTE 3



You can always edit manually even if you used **Option A** or **B**. In the same way, you can mix all the options together. It has been separated in three options just for the sake of ease of explanation

- Click Close button  when you are done. At that moment you are back to the main window and the variables are in memory

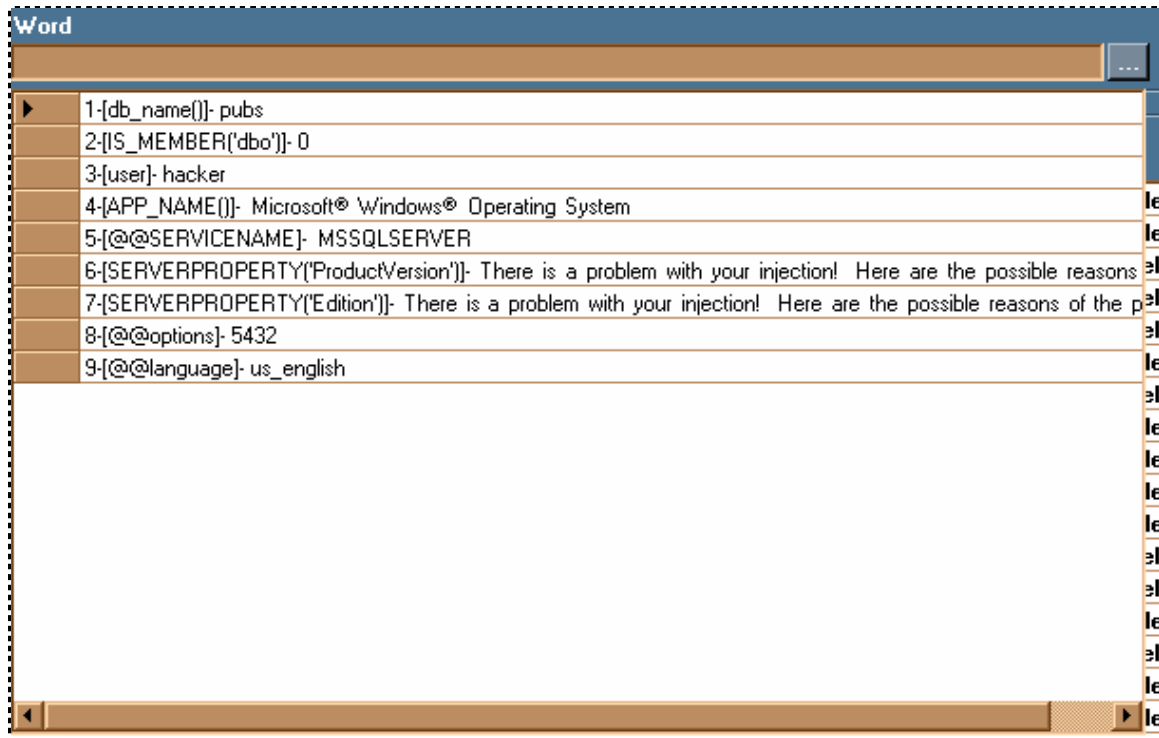
## HINT 1

You can see the first 25 values with the last one if you leave your cursor over Use Variable Range checkbox.




- Now you set! Click on Start button  and get the values. In order to visualize the results you will need to click on the button  situated on the right of the chosen type (Word, Length or Count)

On this example, it's on the right of the Word since we wanted to get the word value from the variable list.



First you see the number of the line on the left and between the brackets you see the variable value used in the injection. Then you find the result itself after the dash.

## HINT 2

You can click on the button  before you start or during the injection to see the results appear in real time.

## HINT 3

Not only the feature Range list from file is convenient to load existing list of words but it can be very handy to save lists of environment variables belong to each DBMS. Therefore, you will be able to use it each time you are doing a Pen-Test to gather basic information without any efforts. I'll provide saved files containing those lists with SQL Power Injector.

A good starting point is to get environment variables from the Information section of each DBMS in the "Aide Memoire" help file.

#### NOTE 4

You might get this special sun character (☼) instead of a character. What it means is that that character for any reason has not been found (server timed out, too many threads at the time, to name a few). I decided to replace it with this character, first because it's almost never used in real situation and second that I preferred to get an invalid character than to stop the flow of the process. Especially with this **STEP** when you want to get multi-values.

### GO TO [STEP 25.14](#)

## 25.14 YOU WANT THE VALUE OF A WORD?

By word it means that it's the value in the Varying String you want to get. If it's a user then it will be one word, if it is a stored procedure then it will be as many words it has, and so on.

You will need to select the type Word if it's not already selected.



#### NOTE 1

Word is selected by default.

As an example in the next figure, you want to get the name of the first database.

String Parameters		Cookie Parameters		Add	Remove
	Name	Starting string	Varying string	Ending string	
▶	<input checked="" type="checkbox"/> txtSearch	white' and	name	from master..sysdatabases where dbid=1)-	

You will get the result displayed like the next figure:

Results				
Current Char	Length	Word	Time taken	Total Requests
t	6	master	0 s 687 ms	43

**Note:** the Word part has been cut to make it possible to fit in the Tutorial

In the last figure, the **Current Char** as the name implies is the current character being sought, the **Length** is the length of the Varying String value, **Word** is the actual value of the Varying String, in this case the name of the first database, **Time taken** is the time that actually took since it started its search for the word, and finally, **Total Requests** is the number of requests that has been done to the web server in order to get the value.

**NOTE 2**

In order to find the value of the Varying String the application needs to find out the length first. It will use it to go fetch character by character until the size is found.

Besides it's always good to know in advance to have a general idea of how long it will take.

**NOTE 3**

You will see in the **STEP 25.19** that it can be handy to know that value take in order to optimize the number of requests made on the web server.

**NOTE 4**

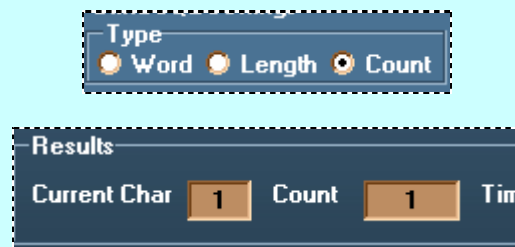
You might get this special sun character (☀) instead of a character. What is mean is that that character for any reason has not been found (server timed out, too many threads at the time, to name a few). I decided to replace it with this character, first because it's almost never used in real situation and second that I preferred to get an invalid character than to stop the flow of the process.

If you want to see the value of a word then your answer is **YES – GO TO STEP 25.18**.

If you don't want to see the value of a word then your answer is **NO – GO TO STEP 25.15**.

**IMPORTANT NOTE**

You absolutely need to have only one value count if not you will end up to have an error message. To see if you have more than one value, switch the type to Count option and look if you get a count of 1.



The screenshot shows a software interface with two main sections. The top section is titled 'Type' and contains three radio buttons: 'Word', 'Length', and 'Count'. The 'Count' radio button is selected. The bottom section is titled 'Results' and contains a table with three columns: 'Current Char', 'Count', and 'Time'. The 'Current Char' column has a value of '1' in a text box, and the 'Count' column has a value of '1' in a text box. The 'Time' column is empty.

If you don't get a count of 1 try to use **WHERE** clause to make it go down to 1.

- IF YOUR ANSWER IS **NO** GO TO **STEP 25.15**
- IF YOUR ANSWER IS **YES** GO TO **STEP 25.18**

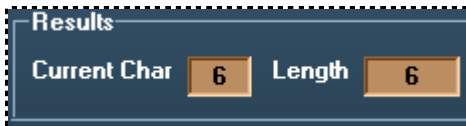
**25.15 YOU WANT THE LENGTH OF THE VALUE?**

This time it means that it's the length of the value in the Varying String you want to get.

You will need to select the type Length if it's not already selected.



This time you will get the result displayed like the next figure:



In the last figure, the **Current Char** as the name implies is the current character being sought and the **Length** is the length of the Varying String value.

If you want to see the length of the Varying String then your answer is **YES – GO TO STEP 25.20**.

If you don't want to see the length of the Varying String then your answer is **NO – GO TO STEP 25.16**.

- IF YOUR ANSWER IS **NO** GO TO **STEP 25.16**
- IF YOUR ANSWER IS **YES** GO TO **STEP 25.20**

### 25.16 YOU WANT THE NUMBER OF RESULTS WITH THIS INJECTION?

Now it means it's the number of occurrence that the Varying String with the conditions if any (**WHERE CLAUSE**) will have.

You will need to select the type Count if it's not already selected.



This time you will get the result displayed like the next figure:



In the last figure, the **Current Char** as the name implies is the current character being sought and the **Count** is the number of results you've got with the injection.

If you want to see the number of results you've got with the injection then your answer is **YES – GO TO STEP 25.17**.

**NOTE**

In this case there is no “**NO**” answer because you should be there after answering no to Word and to Length leaving you with no other possible option.

- IF YOUR ANSWER IS **YES** GO TO **STEP 25.17**

**25.17 CHOOSE THE NUMBER OF RESULTS CLOSEST TO WHAT YOU EXPECT AND CLICK START**

In this step you need to find the number of results closest to what you expect to have. Why? You can always leave it by default but you will get an error message if it is over it, or if you expect a value of 2 it will make more call to the web site by default than if you set the starting Count value to 5 for example.

It's a really nice way to fine-tune the application and it can on the long run save you a lot of time.

**NOTE 1**

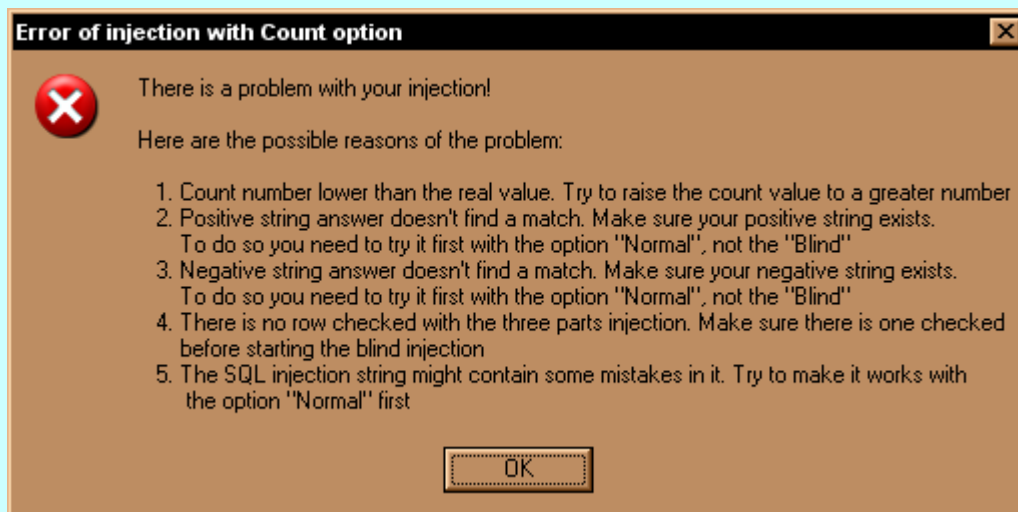
The value by default is 100.

**NOTE 2**

You should refer to the question “*Why can I change the number of the starting length and starting count?*” in the FAQ section of the web site to have a nice discussion about the choice of the Length and Count.

**IMPORTANT NOTE**

You will get a popup error message if your Count is set to a lower number than it is. If it happens you just need to raise the Count number to a level where you won't get popup error message anymore. The popup error message should look like this:



You can change the value in this textbox:



Once you are done, click Start button  to launch the process.

**GO TO [STEP 25.21](#)**

### **25.18 CHOOSE THE NUMBER OF THREADS EXECUTED**

Here you will need to decide the number of threads you want to use to find the value of the Varying String. It is one of the powerful features of the application since you can cut literally in half the time taken to get a word. In some cases it can save you half an hour to several hours on really long string (See the Statistics on the web Site).

#### **NOTE**

The value by default is 1.

You can change the value in this textbox:



#### **HINT**

If you know ahead the length of the value you should try to get the highest divisible value for the number of threads. In average all threads will finish at the same time so if they have the same number of characters to find it will all end at the same time. What happens sometimes it's that the number is not equal for each thread and one will have to finish after, rising thus the time taken.

Of course, the optimization is more complex than that, many things can affect the numbers, such as the CPU of the computer doing the injections, the capacity of the attacked web server to answer many requests at the same time to name a few.

**GO TO [STEP 25.19](#)**

### **25.19 CHOOSE OR OPTIMIZE DICHOTOMY**

Again this is a new feature that came with version 1.2. In fact, it's a novelty that as far as I know no one has ever made any paper with this optimization. Well most definitely it has never been added in a security tool before.

Perhaps you already asked yourself: how could I possibly avoid making 8 requests for each character in the word I'm seeking. For instance, let's say you want to get the full version information for SQL Server, which contains around 185 characters.



It would take 1480 requests! (In SQL Power Injector it's not 1480 but 1665 requests, see note 1 below)

**NOTE 1**

You might have notice that by default it's not 8 requests by character that is made by SQL Power Injector but 9.

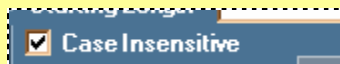
The reason is that the last character is deduced by the dichotomy algorithm but never verified. If in theory it's great, in the practical world it might end up with false positive since during the process something weird might have happened during the comparison process. And it does happen more often you can imagine...

So to be sure that it does not induce errors or even crash the whole process the application is making an extra check to see if it really equals to that ASCII number.

A bit taxing isn't? Well I added a new way to optimize the dichotomy. Why use the Full-Set of ASCII character (256) when you already know that it is most likely be alphabet characters? That would reduce the set of characters to 52 if you care about the case sensitivity.

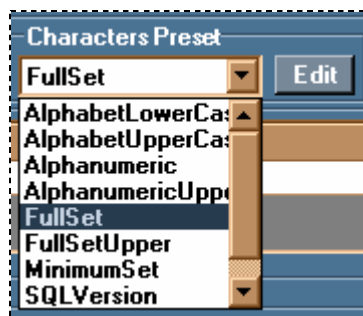
**HINT**


If you don't care about case sensitivity, and most of the time you don't since the rare time you need it it's for passwords, then there is yet another way to optimize it. Just check the Case Insensitive checkbox and choose a preset that does not contain any lower case alphabet characters or edit the preset, if it is not read-only, and remove them by hand. (By default it is set to UPPER the characters)



Here are the ways you can use that new feature (please note that some of the buttons and details will be explained after at the end of this **STEP**):

Choose one of the characters preset that fits more to the expected result if it exists

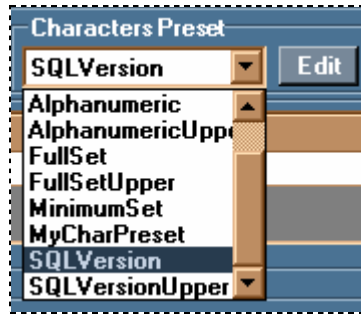



After you selected it you just need to click on Start button  to launch the process. As you can see in the list, there is the FullSetUpper and

AlphabetUpperCase that can be chosen with the Case Insensitive option checked in order to save some requests as explained in the previous hint.

Choose an existing characters preset and modify it

First choose it




Then, click on Edit button  to display the characters preset with the characters selected in the ASCII table.

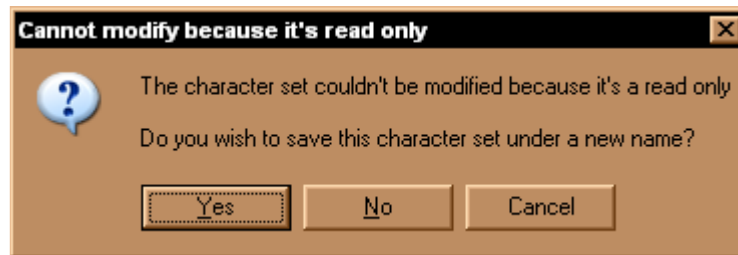
Ascii Table															
SQLVersion		Save	Save As	Remove	Select All	Clear	Estimated requests by char if char is inside array 7 to 8				Close				
NUL	DLE	SPACE	0	@	P	`	p	€		NBSP	*	À	Ð	à	ä
SOH	DC1	!	1	A	Q	a	q		‘	ı	±	Á	Ñ	á	ñ
STX	DC2	"	2	B	R	b	r	,	’	ç	²	Â	Ò	â	ò
ETX	DC3	#	3	C	S	c	s	f	“	£	³	Ã	Ó	ã	ó
EOT	DC4	\$	4	D	T	d	t	.	”	¤	´	Ä	Ô	ä	ô
ENQ	NAK	%	5	E	U	e	u	_	•	¥	µ	Å	Õ	å	õ
ACK	SYN	&	6	F	V	f	v	†	—	¦	¶	Æ	Ö	æ	ö
BEL	ETB	'	7	G	W	g	w	‡	—	§	·	Ç	×	ç	÷
BS	CAN	[	8	H	X	h	x	~	—	-	˙	È	Ø	è	ø
TAB	EM	]	9	I	Y	i	y	%	™	©	˘	É	Ù	é	ù
LF	SUB	*	:	J	Z	j	z	\$	§	®	˚	Ê	Ú	ê	ú
VT	ESC	+	;	K	[	k	{	<	>	«	»	Ë	Û	ë	û
FF	FS	,	<	L	\	l		œ	æ	–	¼	Ì	Ü	ì	ü
CR	GS	-	=	M	]	m	}			-	½	Í	Ý	í	ý
SD	RS	.	>	N	^	n	~	Ž	ž	•	¾	Î	Þ	î	þ
SI	US	/	?	O	_	o	DEL		Ÿ	–	¿	Ï	ß	ï	ÿ

As you can see the selected characters are in beige. I truly believe that it is a real convenient way to choose the characters by displaying this ASCII table.

To add a character or remove it, just click on it to toggle it on or off.

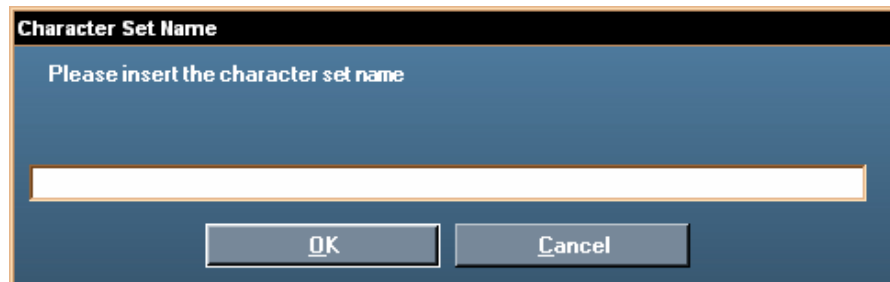
So to continue the explanation, modify the preset as explained in the previous sentence.


Then click on the Save button . If it's not a read-only preset then it will display a message stating that it has been successfully saved. If it's read-only preset you will see this message:




You will have the choice to save it under a new name. If you don't, you will lose that modified preset.



In the case you do save it under a new name you will get this window asking the new name.





Once you provided the name and clicked OK you will get a new preset. After this, you just need to click on the Close button  to return to the main window. From there you will be ready to launch the SQL injection since it will automatically select that new preset for you.

### Create a brand new characters preset

Directly click on Edit button  and it will open the characters preset with the selected characters preset from the drop down menu with the characters selected in the ASCII table (see image of the window in page 62). Don't mind this, since you are creating a brand new one anyways.

There, you face two choices. You can either remove and add new characters like it was explained in the last use case or click on Clear button  to start over or on Select All button  and from there remove the ones you don't need.

Either way, once you are ready to save it you need to click on Save As button  since your goal is to create a new preset.

And from there you are done. Just need to click on Close button  to return to the main window. From there you will be ready to launch the SQL injection since it will automatically select that new preset for you.

Now let's explain some details that have not yet been discussed.

Remove button  :

If you need to remove one of the preset you just need to click on this button and the selected character preset in the drop down menu will be deleted. Make sure that you have the right one in the drop down menu before to remove it.



You cannot remove the read-only presets.

Estimated requests by char if char is inside array:



It is simply the number of requests that the application will need to do in order to get the value if that value has been selected in this characters preset. Sometimes you will get x to y requests. It only means that depending on the exact position of that character within the selected characters it might take more and less one request to get that value.

## NOTE 2

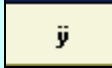
Even though you use a character preset that contains fewer characters than the Full-Set (256 characters, normal dichotomy) you might end up having more requests. The reason is that if it does not find it in available characters at its disposal, the application will have to make a second pass with the missing characters.

So let's imagine the Word you are seeking contains a dash (-), it's not in the characters preset and it takes around 6 requests to figure it out. It will have to do a second pass and might have to requests potentially 6 times (it will never call the Full-Set again, see why below) making a total of 12 requests when you tried to optimize your SQL injection and that if you would have used the default Full-set it would have taken only 8 calls. Like I said it's optimization, but believe me it's worth the effort since most of the time the average of these special characters are really low.

The good news is that if the application didn't find it on the first pass it will not call all the characters again. Why? Because it already knows which set of characters was available and not only that but it knows as well the approximate position in the ASCII table where it can be found. (For more details about the optimization please refer to the source code)

**NOTE 3**

As you might have noticed when you display the characters preset ASCII table the



character is always selected. It's because in order for the optimized algorithm to work it needs the upper boundary set to the character ASCII 256. It might seem weird but so far I haven't been able to fix that problem. If anyone can help on that one I will credit him for it in the next version.

**NOTE 4**

If you need to make one of your created characters preset read-only or for any mysterious reason you want to make the built characters preset writable it is possible. You need to edit the XML file `characterSavedSets.xml` in Settings folder and change the value of **ReadOnly** to true or false.

**GO TO [STEP 25.20](#)**

**25.20 CHOOSE THE LENGTH THAT YOU EXPECT THE VALUE TO BE AND CLICK START**

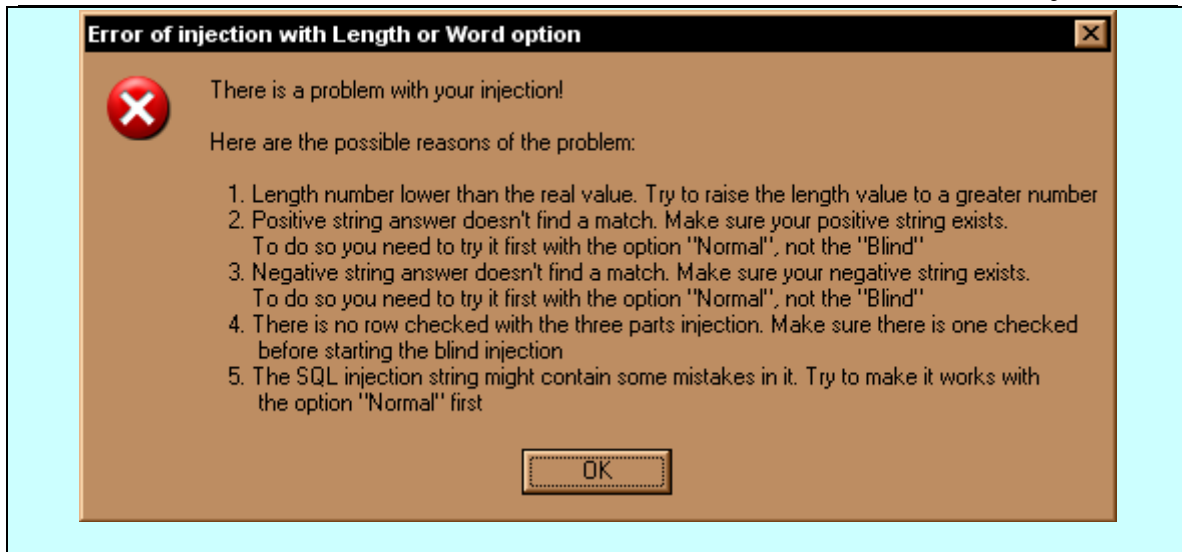
This time you will need to decide what will be the Length of the Varying String and set it up. Again it's a question of optimization and the reasons are the same as for the **STEP 25.17** so please refer to that step to have more details.

**NOTE 1**

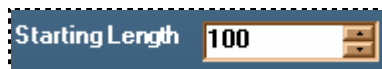
The value by default is 50.

**IMPORTANT NOTE**

You will get a popup error message if your Length is set to a lower number than the length of the value in Varying String is. If it happens you just need to raise the Length number to a level where you won't get popup error message anymore. The popup error message should look like this:



You can change the value in this textbox:



#### NOTE 2

You should refer to the question *"Why can I change the number of the starting length and starting count?"* in the FAQ section of the web site to have a nice discussion about the choice of the Length and Count.

**GO TO [STEP 25.21](#)**

### 25.21 READ THE RESULT

You are done! At that moment you should have the value you requested (Word, Length or Count)

**CONGRATULATION YOU FINISHED THE BLIND SQL INJECTION TUTORIAL!!**

### 26. CAN I USE UNION?

The UNION command is one of the most powerful commands you can inject when you only want to read data in the database. The ironical thing is that you will use the normal display feature of the web application to display your data. Normally it's going to be displayed in a table but be alert because many times I found it in the source code or even in the path of the **SRC** of an image!

If you can inject a UNION command then your answer is **YES – GO TO [STEP 28](#)**.

If you are not able to inject a UNION command then your answer is **NO** – GO TO **STEP 27**.

**HINT**

If when using the **UNION** you get a lot of undesirable data from the normal result you can easily get rid of them in inserting before the injection (most of the times a quote) weird data, like zxxzxxz. It needs of course to correspond to the data type of the **WHERE** clause.

This way, the normal **SELECT** part won't find anything and you will only get your data.

- IF YOUR ANSWER IS **NO** GO TO **STEP 27**
- IF YOUR ANSWER IS **YES** GO TO **STEP 28**

**27. USE TECHNIQUE THAT WILL DISPLAY ERRORS THEN CLICK START BUTTON**

There are many ways to get information from an error page and will most definitely not tell all of them, if such thing is possible at first place. They are just too many and I'm sure that I don't know half of them.

But I will try to give the basic ones.

- Or 1 in (select
- Having 1=1
- Putting a ) or (
- Putting a ; or --
- Putting ' again somewhere else,
- Union select 1
- Etc

When you are done click Start button

A rectangular button with a dark grey background and the word 'Start' in a light blue, sans-serif font.

GO TO **STEP 29**

**28. MODIFY PARAMETERS FOR UNION VALUES THEN CLICK START BUTTON**

We can assume that at this time you have already tested it to make it works so you have the real syntax, if not try to find the real syntax.

**HINT**

The main problem in **UNION** injection is to find the right number of columns that must correspond to the number of columns in the left **SELECT** (the one you use the **UNION** with).

A nice trick is to add a null value until you stop to get error, once done replace the first one with your value, let's say **@@servername**, and if you don't get any error the type is alright. If you do get an error replace it with null and try the next column and so on until it works.

Once your parameters are all set, click Start button

A rectangular button with a dark blue background and the word 'Start' in white text.

**GO TO [STEP 29](#)**

**29. READ THE RESULTS**

You are done! At that moment you should have got the value you requested.

**CONGRATULATION YOU FINISHED THE TUTORIAL!!**