# TESTING

Web testing is a software testing practice to test the websites or web applications for potential bugs. It's a complete testing of web-based applications before making live. A web-based system needs to be checked completely from end-to-end before it goes live for end users. By performing website testing, an organization can make sure that the web-based system is functioning properly and can be accepted by real-time users.

The UI design and functionality are the captains of website testing.

### Web testing checklists

* **Functionality Testing**
* **Usability testing**
* **Compatibility testing**

**5.1 Functionality Testing:**

Test for – all the links in web pages, database connection, forms used for submitting or getting information from the user in the web pages, Cookie testing etc.

**Check all the links:**

* Test the outgoing links from all the pages to the specific domain under test.
* Test all internal links.
* Test links jumping on the same pages.
* Test links used to send email to admin or other users from web pages.
* Test to check if there are any orphan pages.
* Finally, link checking includes, check for broken links in all above-mentioned links.

## 5.1.1 TEST CASES

**Table 5.1.1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SIN** | **SCENARIOS** | **EXPECTED**  **RESULT** | **ACTUAL RESULT** | **STATUS** |
| **1** | Test the outgoing links from all the pages to the specific domain under test | Check successful | Check successful | Success |
| **2** | Test all internal links | Check successful | Check successful | Success |
| **3** | Test links jumping on the same pages | Check successful | Check successful | Success |
| **4** | Test links used to send email to admin or other users from web pages | Email successful | Email successful | Success |
| **4** | Test to check if there are any orphan pages | Not found | Not found | Success |
| **5** | link checking includes, check for broken links in all above-mentioned links | Not found | Not found | Success |

**Test forms on all pages:**

Forms are an integral part of any website. Forms are used for receiving information from users and to interact with them. So, what should be checked in these forms?

* First, check all the validations on each field.
* Check for default values of the fields.
* Wrong inputs in the forms to the fields in the forms.

## 5.1.2 TEST CASES

**Table 5.1.2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SIN** | **SCENARIOS** | **EXPECTED**  **RESULT** | **ACTUAL RESULT** | **STATUS** |
| **1** | check all the validations on each field. | Check successful | Check successful | Success |
| **2** | Check for default values of the fields | Check successful | Check successful | Success |
| **3** | Wrong inputs in the forms to the fields in the forms | Right Validate | Right Validate | Success |

**Cookies Testing:**

Cookies are small files stored on the user machine. These are basically used to maintain the session- mainly the login sessions. Test the application by enabling or disabling the cookies in your browser options.

cookies are encrypted before writing to the user machine. testing the session cookies (i.e. cookies that expire after the session ends) check for login sessions and user stats after the session ends. Check effect on application security by deleting the cookies.

**5.2 Usability Testing**

Usability testing is the process by which the human-computer interaction characteristics of a system are measured, and weaknesses are identified for correction.

• Ease of learning  
• Navigation  
• Subjective user satisfaction  
• General appearance

**Test for navigation:**

Navigation means how a user surfs the web pages, different controls like buttons, boxes or how the user uses the links on the pages to surf different pages.

**Usability testing includes the following:**

* The website should be easy to use.
* Instructions provided should be very clear.
* The main menu should be provided on each page.

## 5.2.1 TEST CASES

**Table 5.2.1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SIN** | **SCENARIOS** | **EXPECTED**  **RESULT** | **ACTUAL RESULT** | **STATUS** |
| **1** | The website should be easy to use. | Check successful | Check successful | Success |
| **2** | Instructions provided should be very clear | provided successful | provided successful | Success |
| **3** | Wrong inputs in the forms to the fields in the forms | Right Validate | Right Validate | Success |
| **4** | The main menu should be provided on each page | Check successful | Check successful | Success |

**Content checking:**

Content should be logical and easy to understand. Check for spelling errors. Usage of dark colours annoys the users and should not be used in the site theme.

follow some standard colors that are used for web page and content building. These are the commonly accepted standards mentioned above about annoying colours, fonts, frames etc.

Content should be meaningful. All the anchor text links should be working properly. Images should be placed properly with proper sizes.

* 1. **Compatibility Testing:**

Compatibility of your website is a very important testing aspect. See which compatibility test to be executed:

* **Browser compatibility**
* **Operating system compatibility**
* **Mobile browsing**

**Browser compatibility:**

In my web-testing career, I have experienced this as the most influencing part of website testing.  
Some applications are very dependent on browsers. Different browsers have different configurations and settings that your web page should be compatible with.

Your website coding should be a cross-browser platform compatible. If you are using java scripts or AJAX calls for UI functionality, performing security checks or validations then give more stress on browser compatibility testing of your web application.

Test web application on different browsers like Internet Explorer, Firefox, Netscape Navigator, AOL, Safari, Opera browsers with different versions.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SIN** | **BROWSER SCENARIOS** | **EXPECTED**  **RESULT** | **ACTUAL RESULT** | **STATUS** |
| **1** | Google Chrome v73  (latest) | compatible | compatible | Success |
| **2** | Firefox v66 (latest) | compatible | compatible | Success |
| **3** | IE v11 (latest) | compatible | compatible | Success |
| **4** | EDGE v44 (latest) | compatible | compatible | Success |

**Table 5.3.1**

**OS compatibility:**

Some functionality in your web application is that it may not be compatible with all operating systems. All new technologies used in web development like graphic designs, interface calls like different API’s may not be available in all Operating Systems.

**Table 5.3.2**

Hence test your web application on different operating systems like Windows, OS with different OS flavours.

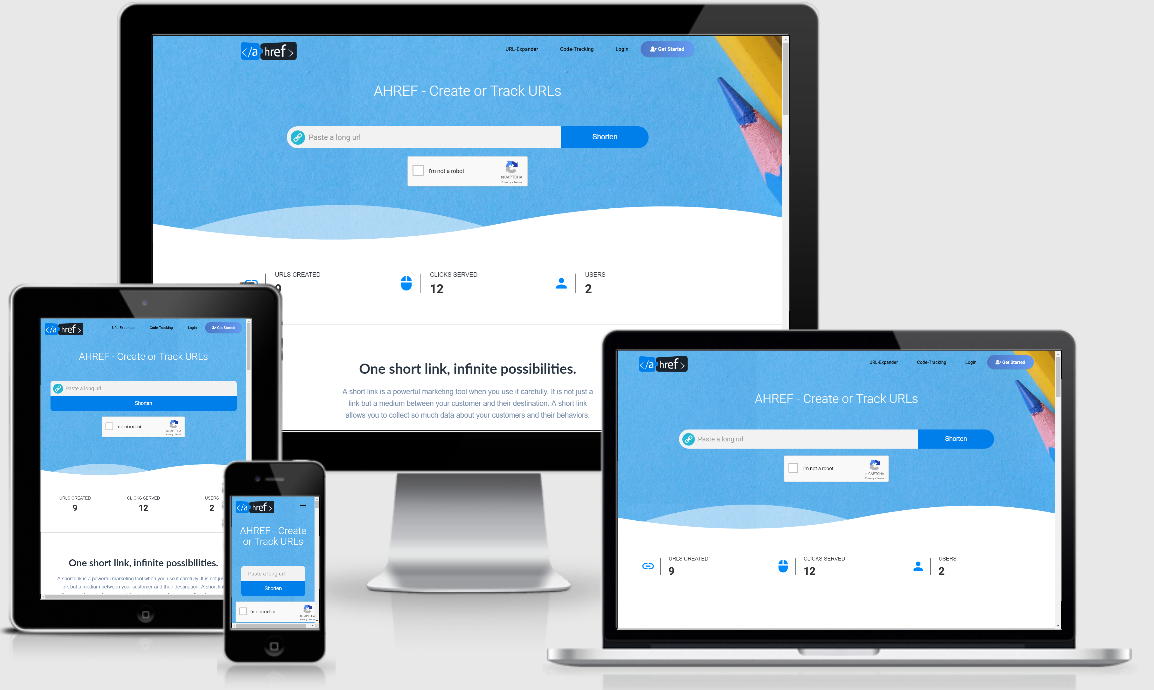
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SIN** | **O.S SCENARIOS** | **EXPECTED**  **RESULT** | **ACTUAL RESULT** | **STATUS** |
| **1** | WIN 10 Home singles | compatible | compatible | Success |
| **2** | Win 8.0 | compatible | compatible | Success |
| **3** | Win 7 | compatible | compatible | Success |

**Mobile browsing:**

We are in the new technology era. So, in future Mobile browsing will rock. Test your web pages on mobile browsers. Compatibility issues may be there on mobile devices as well.

**Table 5.3.3**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SIN** | **BROWSER SCENARIOS** | **EXPECTED**  **RESULT** | **ACTUAL RESULT** | **STATUS** |
| **1** | Chrome | compatible | compatible | Success |
| **2** | UC Browser | compatible | compatible | Success |
| **3** | Firefox | compatible | compatible | Success |
| **4** | Android Default Browser | compatible | compatible | Success |



**Fig 5.3.1 Cross Browser Platform**

* 1. **Security Testing:**

Website security is critical component to protect and secure websites and servers. Websites are scanned for any possible vulnerabilities and malware through website security software. This software can scan for backdoor hacks, redirect hacks, Trojans, and many other threats. A website security software notifies the user if the website has any issue and provides solutions to address them.

Enterprise Networks are always at high risk of vulnerability and ensuring [website security](https://cwatch.comodo.com/) is vital. If the Network gets compromised, the server and the website get compromised as well – this would let the malware infiltrate through the enterprise network and introduce malware activities.

# SQL Injection

SQL Injection (SQLi) is a type of an [injection attack](https://www.acunetix.com/blog/articles/injection-attacks/) that makes it possible to execute malicious SQL statements. These statements control a database server behind a web application. Attackers can use SQL Injection vulnerabilities to bypass application security measures. They can go around authentication and authorization of a web page or web application and retrieve the content of the entire SQL database. They can also use SQL Injection to add, modify, and delete records in the database.

An SQL Injection vulnerability may affect any website or web application that uses an SQL database such as MySQL, Oracle, SQL Server, or others. Criminals may use it to gain unauthorized access to your sensitive data: customer information, personal data, trade secrets, intellectual property, and more. SQL Injection attacks are one of the oldest, most prevalent, and most dangerous web application vulnerabilities. The OWASP organization (Open Web Application Security Project) lists injections in their[OWASP Top 10](https://www.acunetix.com/vulnerability-scanner/owasp-top-10-compliance/) 2017 document as the number one threat to web application security.

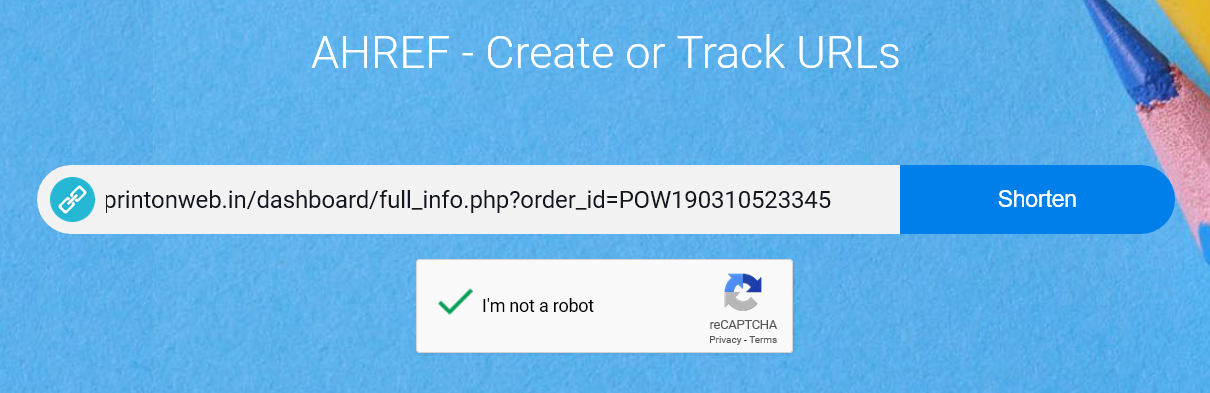
## SQL Injection Performed

To make an SQL Injection attack, an attacker must first find vulnerable user inputs within the web page or web application. A web page or web application that has an SQL Injection vulnerability uses such user input directly in an SQL query. The attacker can create input content. Such content is often called a malicious payload and is the key part of the attack. After the attacker sends this content, malicious SQL commands are executed in the database.

# 5.5 Testing SQL Injection using SQLMAP

# sqlmap is an open source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database servers. It comes with a powerful detection engine, many niche features for the ultimate penetration tester and a broad range of switches lasting from database fingerprinting, over data fetching from the database, to accessing the underlying file system and executing commands on the operating system via out-of-band connections.

* **Step 1: Inserting random URL in Ahref.tech**

****

**Fig 5.5.1 Inserting random URL**

For start testing, we need to first create a payload that need to act as sql injection. Here we are trying to set the main input field. While inserting the random URL inside the field we will start capturing its https request.

* **step 2: Capturing request using Burp suit**

Burp Suite is a Java based [Web Penetration Testing](https://www.pentestgeek.com/web-penetration-testing) framework. It has become an industry standard suite of tools used by information security professionals. Burp Suite helps you identify vulnerabilities and verify attack vectors that are affecting web applications. Because of its popularity and breadth as well as depth of features.

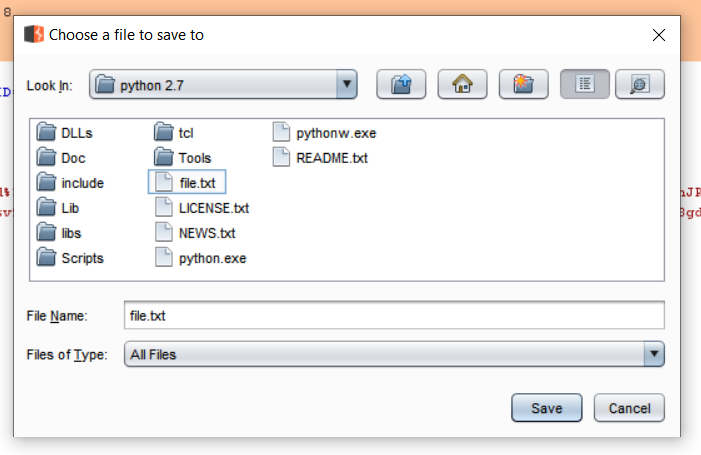
In its simplest form, Burp Suite can be classified as an Interception Proxy. While browsing their target application, a penetration tester can configure their internet browser to route traffic through the Burp Suite proxy server. Burp Suite then acts as a (sort of) [Man In The Middle](https://www.pentestgeek.com/man-in-the-middle) by capturing and analysing each request to and from the target web application so that they can be analysed. Penetration testers can pause, manipulate and replay individual HTTP requests in order to analyse potential parameters or injection points. Injection points can be specified for manual as well as automated fuzzing attacks to discover potentially unintended application behaviours, crashes and error messages.

****

**Fig 5.5.2 Capture POST request**

Once user hit the shorten button the request is captured using burp suit include all the parameters that users enter in the fields. This fields further can be update and modified as per the attacker choice. We are going to capture the request and save it into the txt format in order to test this URL field

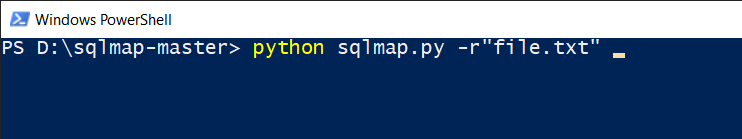
* **step 3: Saving the Response in file.txt**



**Fig 5.5.3 save POST request into txt file**

Copy the request into the txt format under file.txt this particular act as log which can be access any time by the attacker. This is also done because of main POST request which is due to secure in nature.

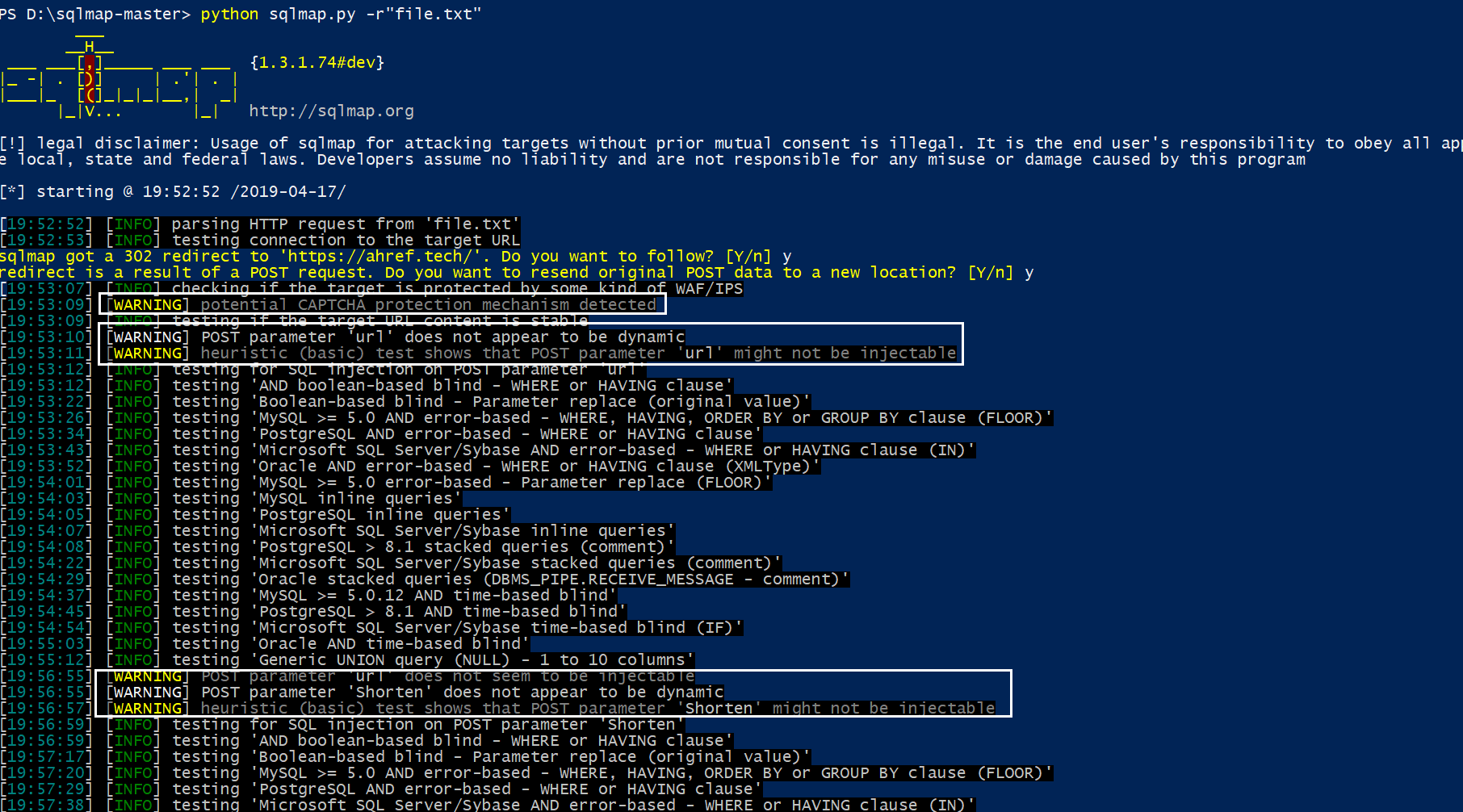
* **step 4: Open Sqlmap by using PowerShell in win10 && enter this command:**

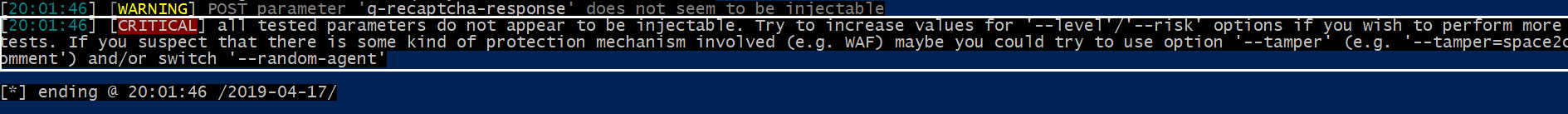


**Fig 5.5.4 Uploading payload to sqlmap**

Open the sqlmap by pressing shift+left click. **NOTE**: python v2.7 must be installed

* **step 5: Open Sqlmap by using PowerShell in win10 && enter this command:**





**Fig 5.5.5 Testing**

**legal disclaimer:** Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

By proceeding with the request, it will scan all the possible combination of attacks on the target field. But unfortunately, no variability was found in this input field.

# 5.6 Testing SQL Injection on Login & signup and contact page

**Table 5.6.1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SIN** | **Pages** | **SCENARIOS** | **EXPECTED**  **RESULT** | **ACTUAL RESULT** | **STATUS** |
| **1** | Login | Performing VATP & SQLMAP | NO VA | NO VA | Success |
| **2** | register | Performing VATP & SQLMAP | NO VA | NO VA | Success |
| **3** | contact | Performing VATP & SQLMAP | NO VA | NO VA | Success |

Above test cases prove that the Ahref is not injectable with SQL injection.