5. 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.

5.3 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.

Table 5.3.1

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

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.

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

Table 5.3.2

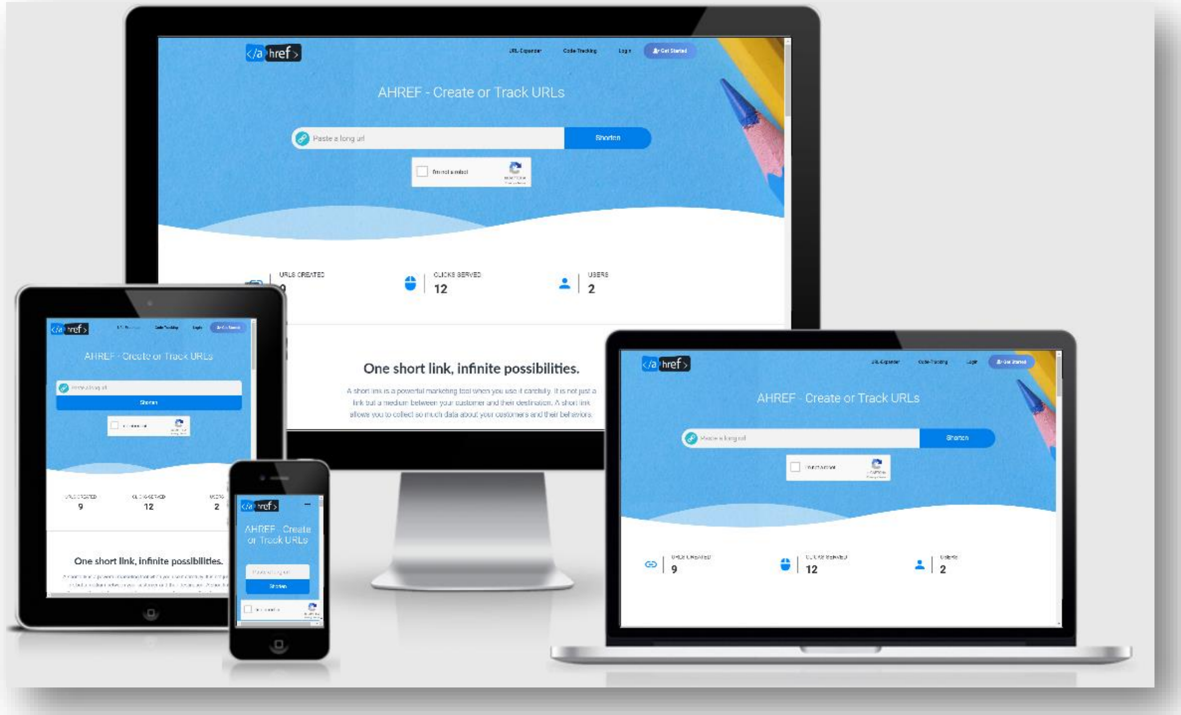
SIN	O.S SCENARIOS	EXPECTED RESULT	ACTUAL RESULT	STATUS
1	WIN 10 Home singles	compatible	compatible	Success
2	Win 8.0	compatible	compatible	Success
3	Win 7	compatible	compatible	Success

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



</a href>
AHREF - Create or Track URLs
Paste a long url
Shorten
URLS CREATED
9
CLICKS SERVED
12
USERS
2
One short link, infinite possibilities.

Fig 5.3.1 Cross Browser Platform

5.4 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 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 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 2017 document as the number one threat to web application security.

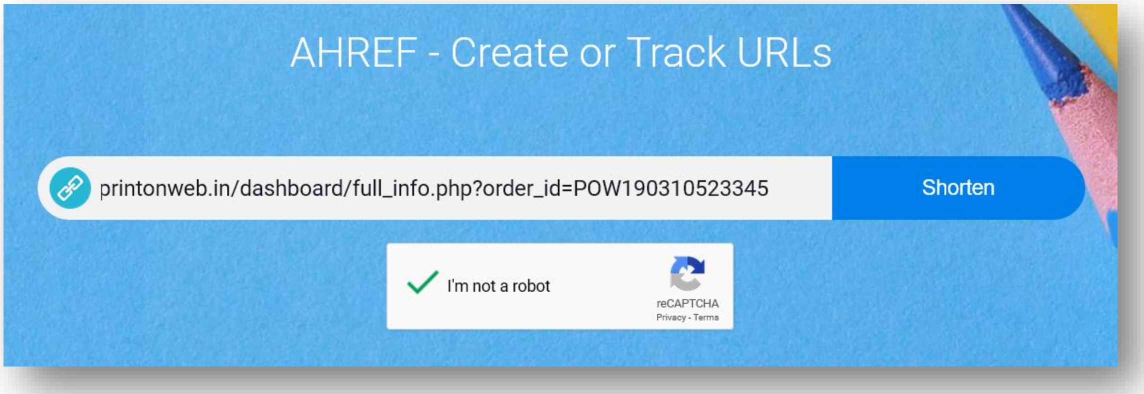
SQL Injection Performed

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**



AHREF - Create or Track URLs

printonweb.in/dashboard/full_info.php?order_id=POW190310523345

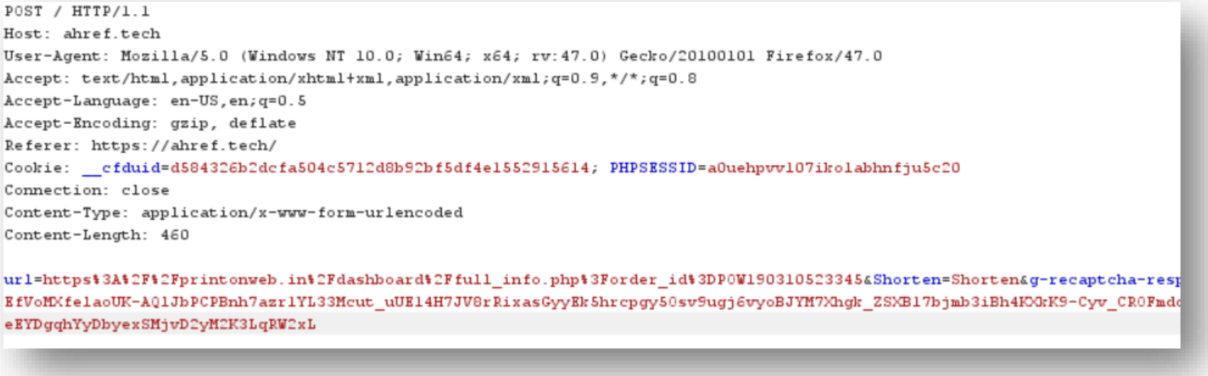
Fig 5.5.1 Inserting random URL

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 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 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.



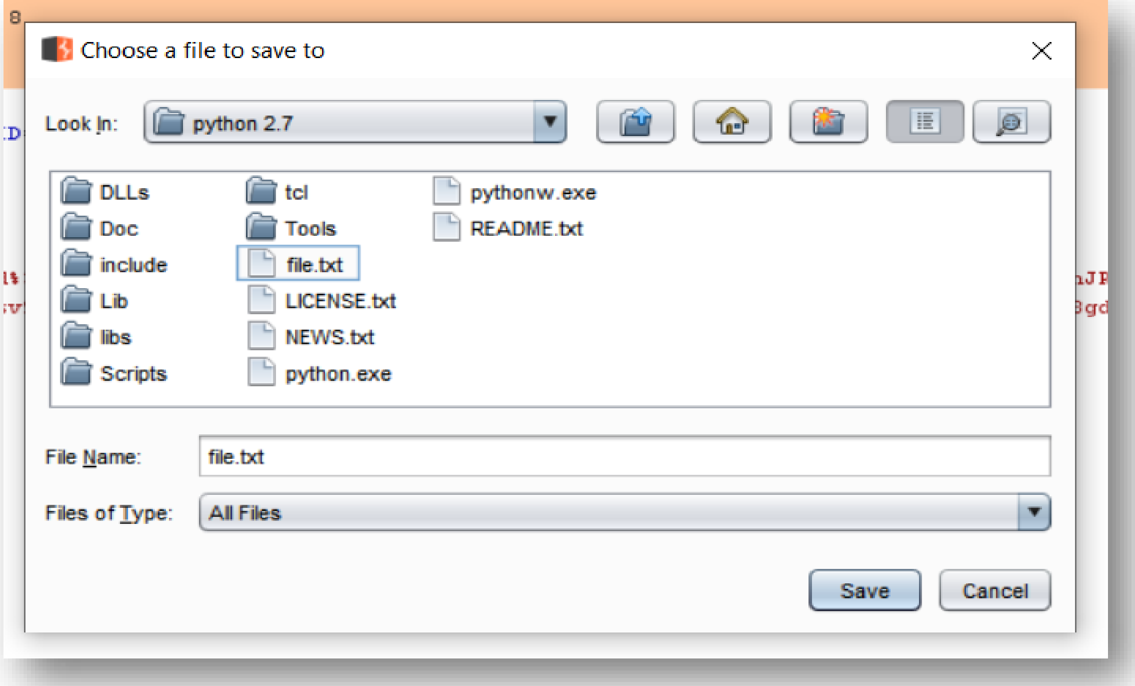
```
POST / HTTP/1.1
Host: ahref.tech
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:47.0) Gecko/20100101 Firefox/47.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://ahref.tech/
Cookie: __cfduid=d584326b2dcfa504c5712d8b92bf5df4e1552915614; PHPSESSID=a0uehpvv107ikolabhnfju5c20
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 460

url=https%3A%2F%2Fprintonweb.in%2Fdashboard%2Ffull_info.php%3Forder_id%3DP0W190310523345&Shorten=Shorten&g-recaptcha-resp
EfVoMXfelaoUK-AQlJbPCPBnh7azrlYL33Mcut_uUEl4H7JV8rRixasGyyEk5hrcpgy50sv9ugj6vyoBJYM7Xhgk_ZSXB17bjmb3iBh4KXkK9-Cyv_CR0Fmdc
eEYDgqhYyDbyexSMjvD2yM2K3LqRW2xL
```

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**



Choose a file to save to

Look In: python 2.7

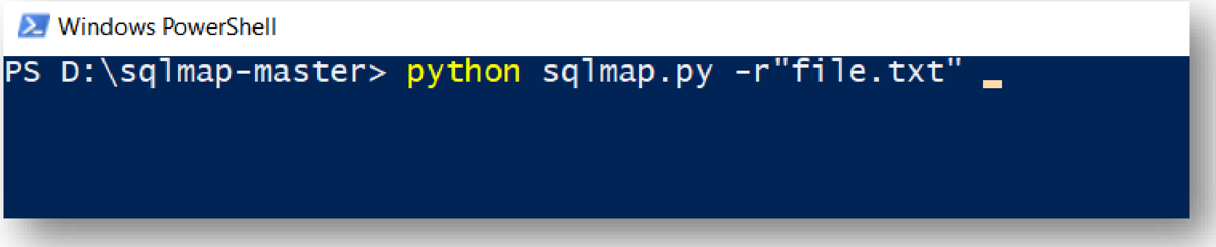
File Name: file.txt

Files of Type: All Files

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:**



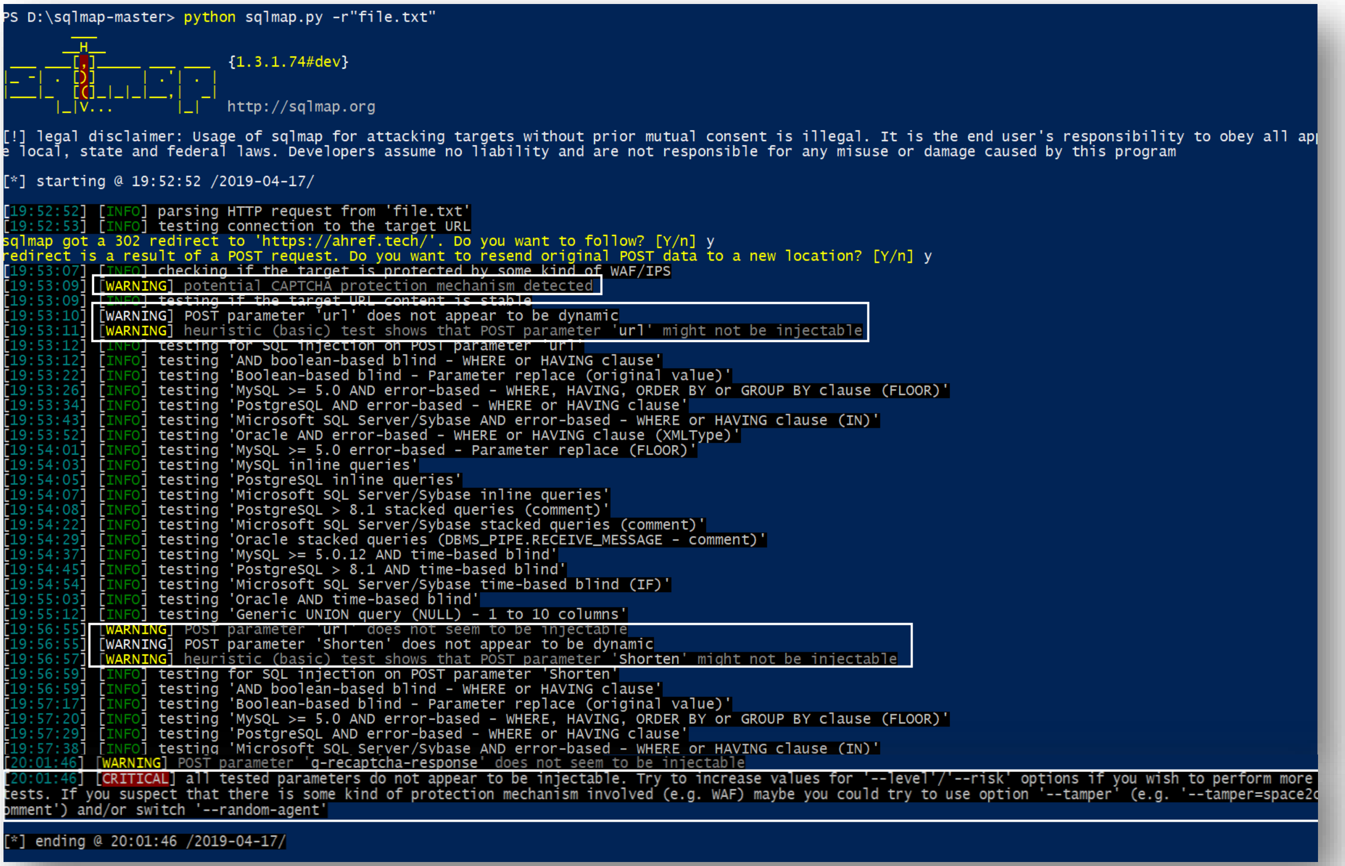
Windows PowerShell

```
PS D:\sqlmap-master> python sqlmap.py -r"file.txt"
```

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:**



```
PS D:\sqlmap-master> python sqlmap.py -r"file.txt"
        ___
       __H__
 ___ ___[,]_____ ___ ___  {1.3.1.74#dev}
|_ -| . [)]     | .'| . |
|___|_  [(]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

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

[*] starting @ 19:52:52 /2019-04-17/

[19:52:52] [INFO] parsing HTTP request from 'file.txt'
[19:52:53] [INFO] testing connection to the target URL
sqlmap got a 302 redirect to 'https://ahref.tech/'. Do you want to follow? [Y/n] y
redirect is a result of a POST request. Do you want to resend original POST data to a new location? [Y/n] y
[19:53:07] [INFO] checking if the target is protected by some kind of WAF/IPS
[19:53:09] [WARNING] potential CAPTCHA protection mechanism detected
[19:53:09] [INFO] testing if the target URL content is stable
[19:53:10] [WARNING] POST parameter 'url' does not appear to be dynamic
[19:53:11] [WARNING] heuristic (basic) test shows that POST parameter 'url' might not be injectable
[19:53:12] [INFO] testing for SQL injection on POST parameter 'url'
[19:53:12] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[19:53:22] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[19:53:26] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[19:53:34] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[19:53:43] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[19:53:52] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[19:54:01] [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace (FLOOR)'
[19:54:03] [INFO] testing 'MySQL inline queries'
[19:54:05] [INFO] testing 'PostgreSQL inline queries'
[19:54:07] [INFO] testing 'Microsoft SQL Server/Sybase inline queries'
[19:54:08] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[19:54:22] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[19:54:29] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[19:54:37] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind'
[19:54:45] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[19:54:54] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[19:55:03] [INFO] testing 'Oracle AND time-based blind'
[19:55:12] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[19:56:55] [WARNING] POST parameter 'url' does not seem to be injectable
[19:56:55] [WARNING] POST parameter 'Shorten' does not appear to be dynamic
[19:56:57] [WARNING] heuristic (basic) test shows that POST parameter 'Shorten' might not be injectable
[19:56:59] [INFO] testing for SQL injection on POST parameter 'Shorten'
[19:56:59] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[19:57:17] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[19:57:20] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[19:57:29] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[19:57:38] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[20:01:46] [WARNING] POST parameter 'g-recaptcha-response' does not seem to be injectable
[20:01:46] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options if you wish to perform more
tests. If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=space2c
omment') and/or switch '--random-agent'

[*] ending @ 20:01:46 /2019-04-17/
```

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.