HACKLAB PIZZA WAPT

Michael Awoyemi

CMP319: Web Application Penetration Testing

BSc Ethical Hacking Year 3

2023/2024

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

Abstract

It is a challenge for all organisations to keep up with all the flaws and vulnerabilities in systems, considering the rapidly evolving world of technology. The pace of technological advancements often outpaces the ability of companies to keep up, creating an environment where potential vulnerabilities may go unnoticed or unaddressed. Moreover, companies don't have a fully qualified team or the funds and resources to fix vulnerabilities even after they have been detected and reported.

Web applications are not exempted. Especially with the increasing prevalence of online procedures, user's sensible information (name, username, password, email, and address) poses a significant risk in the event of criminals exploiting the web applications.

This report presents a web security test performed on Hacklab Pizza online systems. The approach that will be used in this report to identify flaws and vulnerabilities is the OWASP web application testing methodology. After following the OWASP guide, the conclusion is that the Hacklab Pizza web is very unsecure. It has been discovered the use of outdated services, in addition to other misconfigurations that can lead to several vulnerabilities. Additionally, suggested countermeasures will be found in this report. By presenting both the discovered vulnerabilities and the suggested countermeasures, this report aims to guide Hacklab Pizza in fortifying its web applications against potential threats.

Contents

1 Introduction

1.1 Background

Hacklab Pizza recently purchased a web application from a web developer company. It has been brought to the new owner that the application has some bugs, but it is mostly functional. The owner of the company is concerned that there may be some bugs that could be used to hack into the application and has requested to perform a web application testing.

Web application tests are conducted to determine the level of security of a web page. Simulated attacks are attempted against the application as an attacker would do. Any flaws in administration management, system vulnerabilities, or user configuration errors can be identified from the findings. These tests aim to detect, report, and mitigate the flaws before criminals use them to their advantage. Depending on the degree of the flaws, criminals can obtain total control to vandalise or disrupt the original functionality of web applications.

In the year 2022, the most common web application attacks are SQL injections, store Cross-site scripting (XSS) and malicious file upload. A list of the OWASP top 10 can be found in Appendix A.

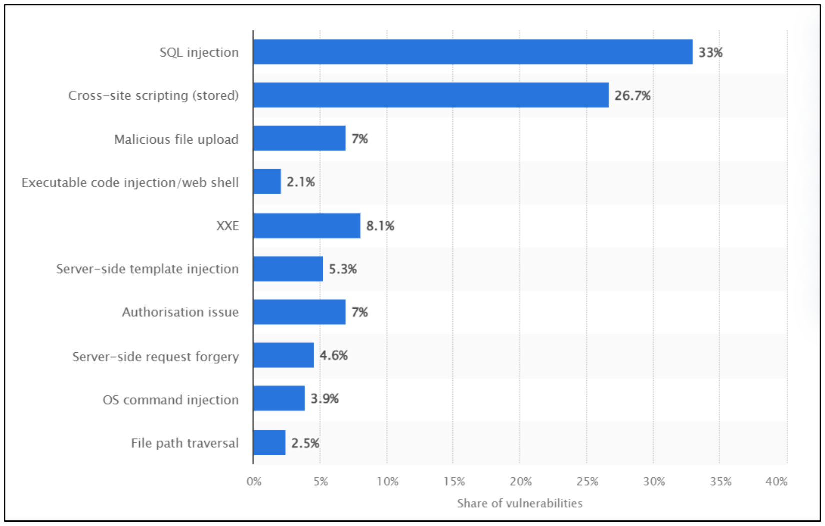


Figure 1. Distribution of web app vulnerabilities as of 2022

To conduct the web application testing on Hacklab Pizza, the following information provided to the penetration tester includes the URL (http://192.168.1.10) and the credentials of the account of a normal user named Mr Rick Astley (username *hacklab@hacklab.com* and password *hacklab*). After the testing, it has been discovered that the Hacklab pizza web application is very unsecure.

1.2 AIMS

The aim of the report is to conduct web application testing using the OWASP methodology to find vulnerabilities or flaws within the pizza online store.

In the event of these being found, they will be documented and further actions such as exploitation and privileged escalations to disrupt the normal functionality of the web page will be taken to raise awareness of the risk of not mitigating them. Lastly countermeasures will also be provided to have a secure web application.

2 METHODOLOGY

The Open Web Application Security Project (OWASP) Web Security Testing Guide is the methodology used for this report. Being the most used methodology used in by many penetration testers it has proven its value with time.

With a well-known documented process that provides penetration testers with a structured guide on how to perform tests efficiently. The guide also emphasises on the TOP 10 which are the most common web application vulnerabilities. The OWASP Top 10 can be found in Appendix A.

The following list is the OWASP Web Security Testing Guide (version 4.2) structure:

I. Introduction and Objectives
II. Information Gathering
III. Configuration and Deployment Management Testing
IV. Identity Management Testing
V. Authentication Testing
VI. Authorization Testing
VII. Session Management Testing
VIII. Input Validation Testing
IX. Testing for Error Handling
X. Testing for Weak Cryptography
XI. Business Logic Testing
XII. Client-side Testing
XIII. API Testing

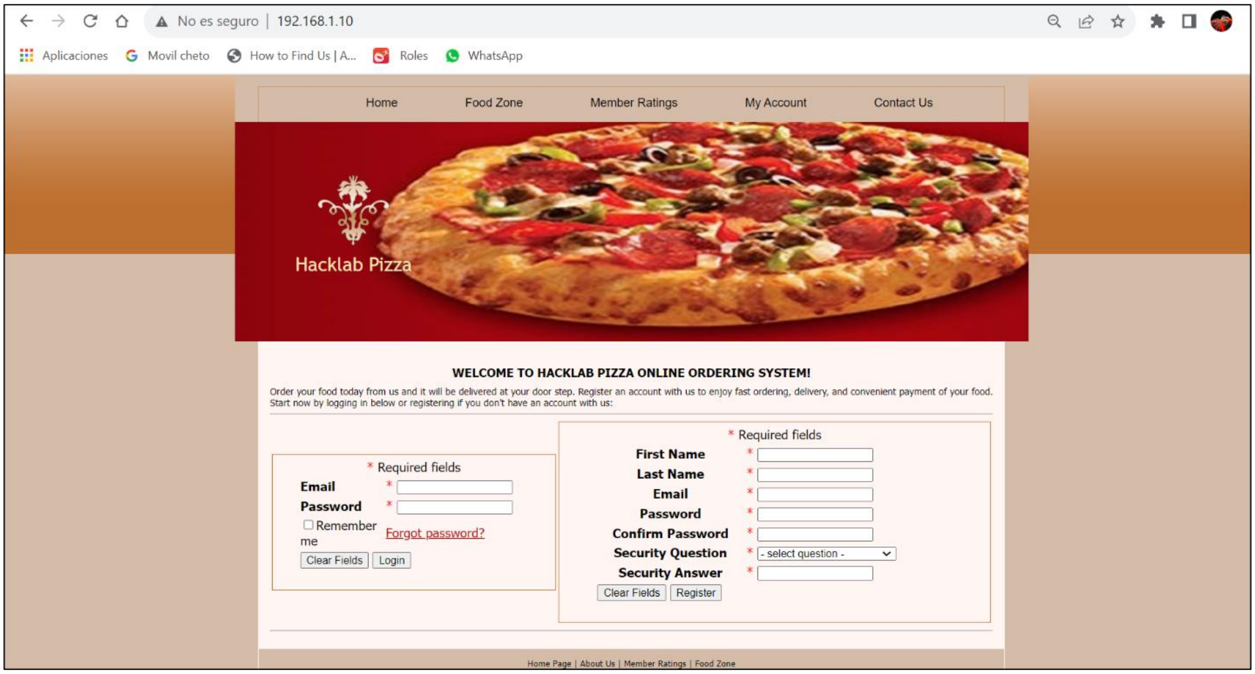
*Hacklab Pizza being a simulated web application some steps in the structure provided by the web security guide will not apply

3 Procedure and Results

3.1 Information Gathering

3.1.1 Passive Information Gathering

During the passive reconnaissance, the home page and linked pages were explored to have a better understanding of the structure of the web application.



WELCOME TO HACKLAB PIZZA ONLINE ORDERING SYSTEM!

Order your food today from us and it will be delivered at your door step. Register an account with us to enjoy fast ordering, delivery, and convenient payment of your food. Start now by logging in below or registering if you don't have an account with us:

* Required fields

Email *

Password *

□ Remember me Forgot password?

* Required fields

First Name *

Last Name *

Email *

Password *

Confirm Password *

Security Question * - select question -

Security Answer *

Figure 2: Hacklab pizza main page

3.1.2 Active Information Gathering

The following step was active reconnaissance. This was done with the following tools:

Nmap: Network map. With the appropriate command information about open ports, banner information and service versions.

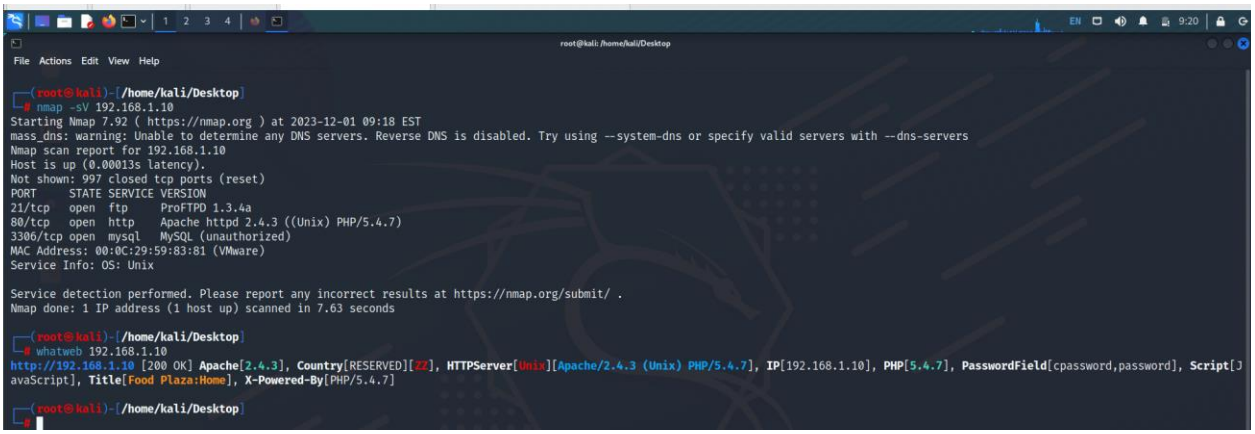
WhatWeb: a tool to identify technologies on a run web.

Dirbuster: an application designed to brute force directories and file names on web application servers using a wordlist.

OWASP Zap: a penetration testing tool that detects and finds vulnerabilities in web applications and has a spider function tool that is used to automatically discover new resources (URLs).

Nessus and Nikto are vulnerability scanners that scan web servers to find unwanted or sensitive files, outdated services and other problems.

The Nmap scan discovered the following open ports and services and versions: port 21 FTP (ProFTPD 1.3.4a), port 80 HTTP (Apache httpd 2.4.3 & PHP/5.4.7), port 3396 running MYSQL.

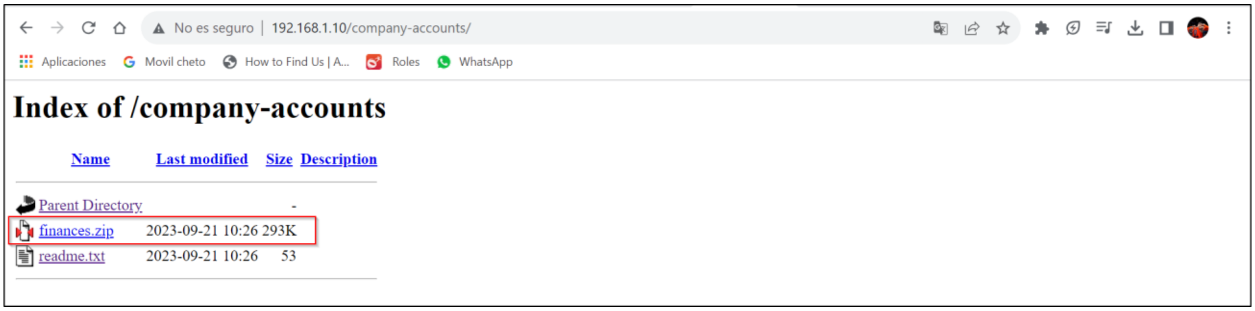


```
┌──(root㉿kali)-[/home/kali/Desktop]
└─# nmap -sV 192.168.1.10
Starting Nmap 7.92 ( https://nmap.org ) at 2023-12-01 09:18 EST
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.1.10
Host is up (0.00013s latency).
Not shown: 997 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
21/tcp   open  ftp     ProFTPD 1.3.4a
80/tcp   open  http    Apache httpd 2.4.3 ((Unix) PHP/5.4.7)
3306/tcp open  mysql   MySQL (unauthorized)
MAC Address: 00:0C:29:59:83:81 (VMware)
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 7.63 seconds

┌──(root㉿kali)-[/home/kali/Desktop]
└─# whatweb 192.168.1.10
http://192.168.1.10 [200 OK] Apache[2.4.3], Country[RESERVED][ZZ], HTTPServer[Unix][Apache/2.4.3 (Unix) PHP/5.4.7], IP[192.168.1.10], PHP[5.4.7], PasswordField[cpassword,password], Script[J
avaScript], Title[Food Plaza:Home], X-Powered-By[PHP/5.4.7]

┌──(root㉿kali)-[/home/kali/Desktop]
└─#
```

Figure 3: Nmap scan and Whatweb scan

The scan that followed was the OWASP Zap. It injects payloads to discovered vulnerabilities, it also has a spider crawl function that returns URLs that we useful to compromise the web application. The Dirburster tool was run using a medium wordlist to find directories more directories that Zap could not find. The directories returned from both scans can be found in Appendix B.1.

After analysing the URLs returned from both scans the following links *http://192.168.1.10/company-accounts/* contained a zip file with sensitive information. The zip file was downloaded as it was available to the public.

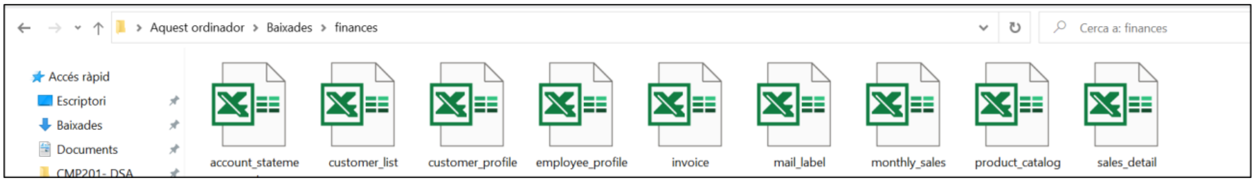


Index of /company-accounts

Name	Last modified	Size	Description
Parent Directory		-	
finances.zip	2023-09-21 10:26	293K	
readme.txt	2023-09-21 10:26	53	

Figure 4: http://192.168.1.10/company-accounts

The zip contained spreadsheets with information about account statements, customer lists, customer profiles, employee profiles, invoices, monthly sales, product catalogues, and sale details. Refer to Appendix B.2 for detailed information.

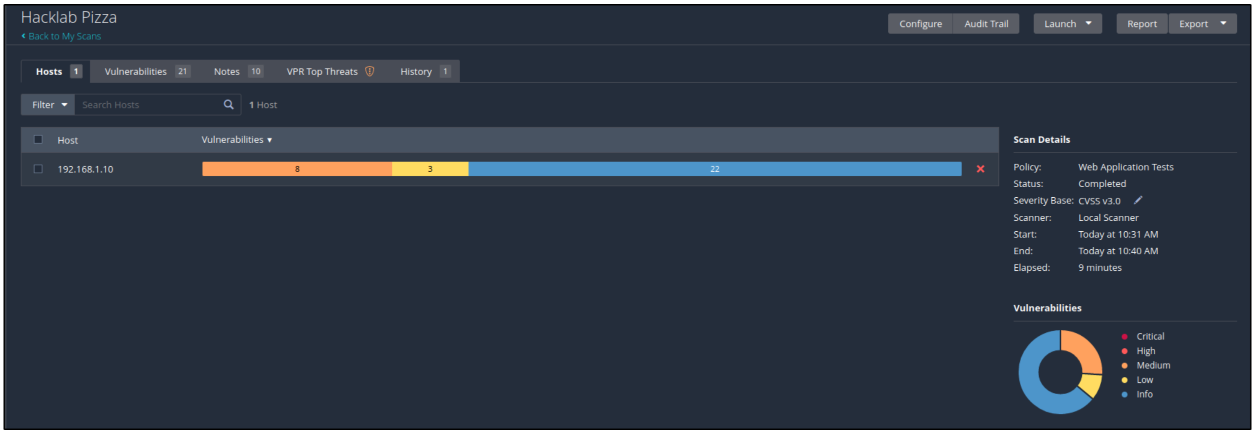


account_stateme
customer_list
customer_profile
employee_profile
invoice
mail_label
monthly_sales
product_catalog
sales_detail

Figure 5: Zip content

The login page to the admin account was also discovered by both Zap and Dirbuster scans. The admin login page was attacked like an attacker would and broke into. This will be discussed in future sections.

A Nessus scan was run to the end of the active information gathering phase as seen below in Figure 6. The result from the scan came back with vulnerabilities and 10 notes. The full report of the scan can be found in Appendix D.



□	Host	Vulnerabilities ▾
□	192.168.1.10	8 3 22

Scan Details

Policy: Web Application Tests
Status: Completed
Severity Base: CVSS v3.0
Scanner: Local Scanner
Start: Today at 10:31 AM
End: Today at 10:40 AM
Elapsed: 9 minutes

Figure 6: Nessus web application test scan

3.2 Configuration and Deployment Management Testing

Configuration and deployment management testing must be performed to verify the functionality and security of web applications. Security misconfigurations involve insecure settings, configurations, or deployments that can be exploited by attackers. No user should be able to access sensitive information directly by searching the path of a directory in the URL. Allowing direct access to sensitive files like configuration files, server-side scripts, or even stylesheets (CSS) can be a security risk. If users can access these files directly, they might gain insights into the inner workings of your application, potentially discovering vulnerabilities or sensitive information. 90% of web applications suffer from security misconfiguration as per OWASP's Top 10 security risk list.

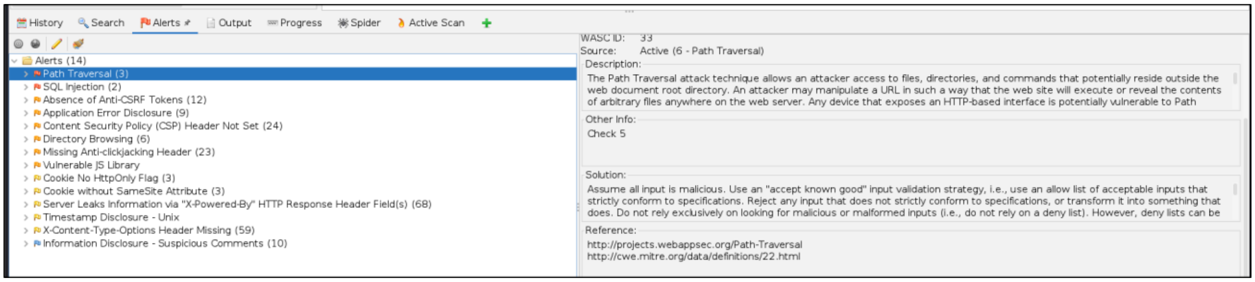
The first security misconfiguration that was encountered was the zip file containing company information. It was not restricted to the public or had any type of permission before opening the files. The content of the zip file can be found in Figure 5 and Appendix B.2

To find more security misconfiguration the tool Nikto was used. Nikto is a web server scanner that performs an exhaustive scan. From the scan, there can be observed a lot of directives that should not be accessible to the public by just typing in the URL as it contains sensitive information. Here are the interesting links. The full Nikto scan result can be found in Appendix C.

```
+ /phpinfo.php: Output from the phpinfo() function was found.
+ Cookie PHPSESSID created without the httponly flag.
+ OSVDB-3268: /css/: Directory indexing found.
+ OSVDB-3092: /css/: This might be interesting...
+ OSVDB-3268: /install/: Directory indexing found.
+ OSVDB-3092: /install/: This might be interesting...
+ OSVDB-3268: /stylesheets/: Directory indexing found.
+ OSVDB-3092: /stylesheets/: This might be interesting...
```

+ OSVDB-3233: /cgi-bin/printenv: Apache 2.0 default script is executable and gives server environment variables. All default scripts should be removed. It may also allow XSS types of attacks.
+ OSVDB-3233: /cgi-bin/test-cgi: Apache 2.0 default script is executable and reveals system information. All default scripts should be removed.
+ 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.

OWASP Zap was also used to gain more security misconfiguration alerts.



Alerts (14)
- Path Traversal (3)
- SQL Injection (2)
- Absence of Anti-CSRF Tokens (12)
- Application Error Disclosure (9)
- Content Security Policy (CSP) Header Not Set (24)
- Directory Browsing (6)
- Missing Anti-clickjacking Header (23)
- Vulnerable JS Library
- Cookie No HttpOnly Flag (3)
- Cookie without SameSite Attribute (3)
- Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s) (68)
- Timestamp Disclosure - Unix
- X-Content-Type-Options Header Missing (59)
- Information Disclosure - Suspicious Comments (10)

WASC ID: 33
Source: Active (6 - Path Traversal)

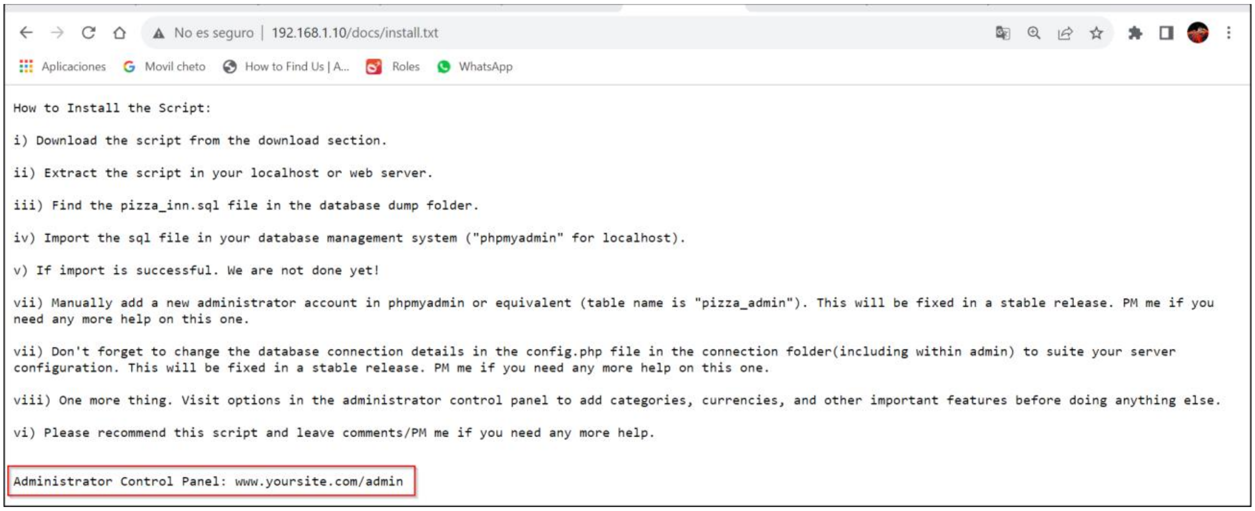
Description:
The Path Traversal attack technique allows an attacker access to files, directories, and commands that potentially reside outside the web document root directory. An attacker may manipulate a URL in such a way that the web site will execute or reveal the contents of arbitrary files anywhere on the web server. Any device that exposes an HTTP-based interface is potentially vulnerable to Path

Other Info:
Check 5

Solution:
Assume all input is malicious. Use an "accept known good" input validation strategy, i.e., use an allow list 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 deny list). However, deny lists can be

Reference:
http://projects.webappsec.org/Path-Traversal
http://cwe.mitre.org/data/definitions/22.html

Figure 7: OWASP Zap Alerts

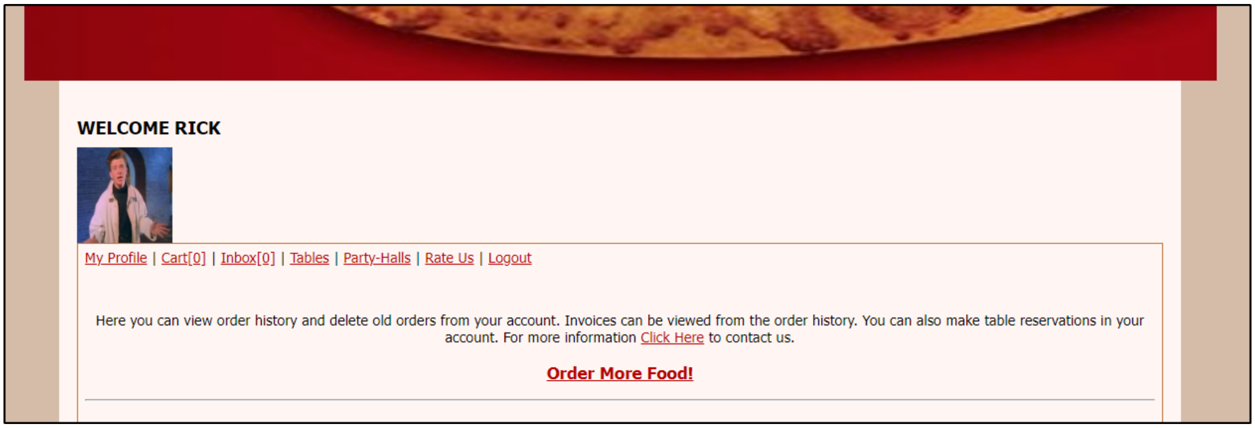


```
How to Install the Script:

i) Download the script from the download section.

ii) Extract the script in your localhost or web server.

iii) Find the pizza_inn.sql file in the database dump folder.

iv) Import the sql file in your database management system ("phpmyadmin" for localhost).

v) If import is successful. We are not done yet!

vii) Manually add a new administrator account in phpmyadmin or equivalent (table name is "pizza_admin"). This will be fixed in a stable release. PM me if you
need any more help on this one.

vii) Don't forget to change the database connection details in the config.php file in the connection folder(including within admin) to suite your server
configuration. This will be fixed in a stable release. PM me if you need any more help on this one.

viii) One more thing. Visit options in the administrator control panel to add categories, currencies, and other important features before doing anything else.

vi) Please recommend this script and leave comments/PM me if you need any more help.


Administrator Control Panel: www.yoursite.com/admin
```

Figure 8: relevant information

3.3 Authentication Testing

The main page contains user logging fields and user registration fields as seen in the previous image Figure 2. Upon seeing this, Mr Rick Astley's credentials previously provided (username hacklab@hacklab.com and password hacklab) were used to log in and further exploration was conducted to the user welcome page and all the linked pages. Inside My Profile, the user can change their password, picture and add their delivery address.



WELCOME RICK

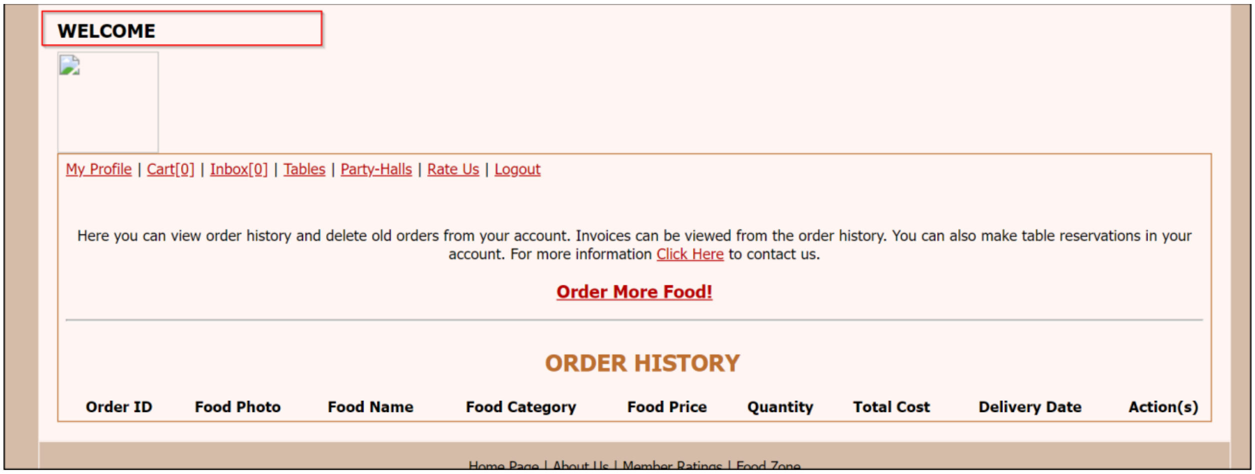
My Profile | Cart[0] | Inbox[0] | Tables | Party-Halls | Rate Us | Logout

Here you can view order history and delete old orders from your account. Invoices can be viewed from the order history. You can also make table reservations in your account. For more information Click Here to contact us.

Order More Food!

Figure 9: Mr Rick Astley's account

In search of a logging error message, it was discovered that access is granted when a user clicks the log-in button without entering a username and password.



WELCOME

My Profile | Cart[0] | Inbox[0] | Tables | Party-Halls | Rate Us | Logout

Here you can view order history and delete old orders from your account. Invoices can be viewed from the order history. You can also make table reservations in your account. For more information Click Here to contact us.

Order More Food!

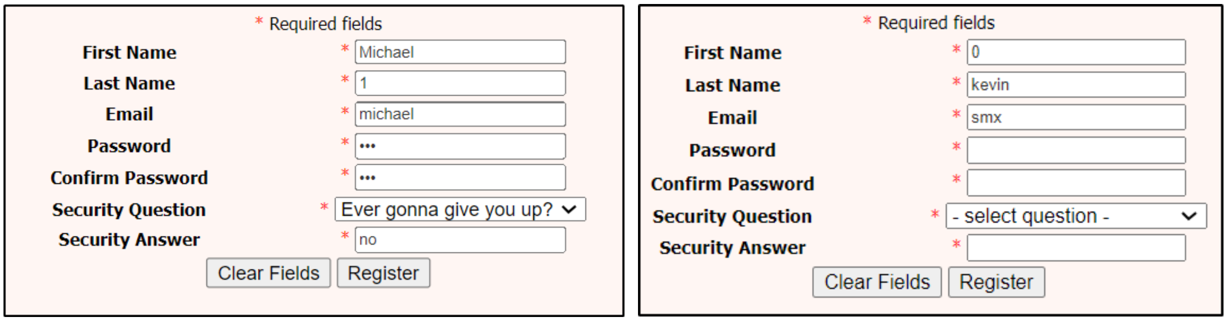
ORDER HISTORY

Order ID	Food Photo	Food Name	Food Category	Food Price	Quantity	Total Cost	Delivery Date	Action(s)

Home Page | About Us | Member Ratings | Food Zone

Figure 10: Blank username and password access granted.

Once the user's home page has been explored, the session was closed. Back to the main home page, several new users were created. The first user's details are the following: last name was a number; a non-valid email and a very weak password were used. The user was created successfully, and access was quickly gained to the user's home page without any type of verification. The second user had a number as a name, non-valid email and password were not provided and did not have a security question or answer. Evidence of the access granted to the profile of the new 2 users can be found in Appendix E.



* Required fields

First Name	* Michael
Last Name	* 1
Email	* michael
Password	* •••
Confirm Password	* •••
Security Question	* Ever gonna give you up?
Security Answer	* no

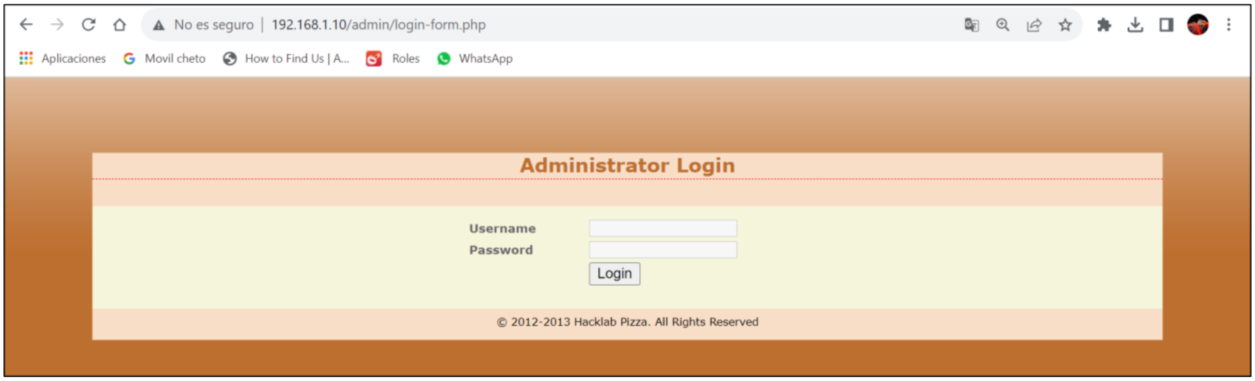
* Required fields

First Name	* 0
Last Name	* kevin
Email	* smx
Password	*
Confirm Password	*
Security Question	* - select question -
Security Answer	*

Figures 11 & 12: New users created with the parameter mentioned above.

One of the findings that Dirburster and OWASP zap returned was the Administrator login form page. Upon navigating to the URL, the login page required both username and password input as shown in Figure 13. The username "Administrator" and password "p4$$w0rd" were tried. The error message that was displayed in Figure 14, said Username not found. This message was useful to the tester, instead of trying to check another password, they can focus on finding the user and brute force the password once the username is correct.

After trying a couple of usernames, it appeared to be "admin". The password for this was wrong but the error message in Figure 15 displayed a Login message failed. The error message went from Username not found to login failed which gave the attacker the insight that the username exits on the database and has been guessed.

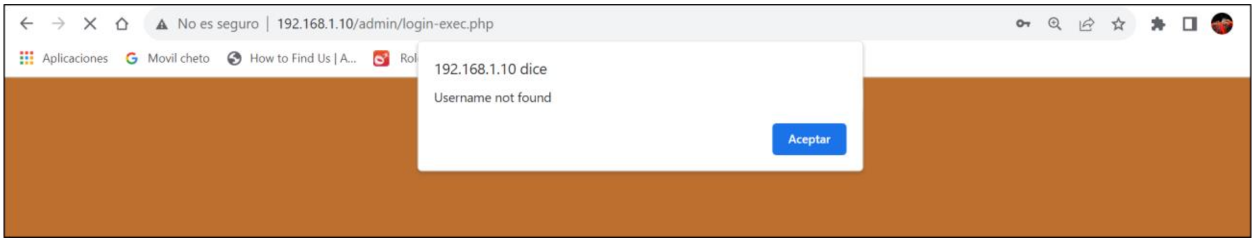


Administrator Login

Username	
Password	

© 2012-2013 Hacklab Pizza. All Rights Reserved

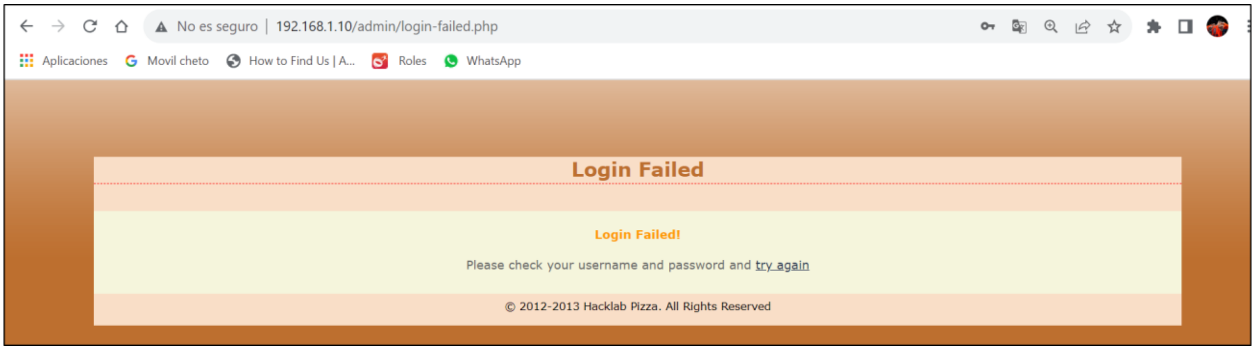
Figure 13: Administrator Login form.



192.168.1.10 dice

Username not found

Figure 14: Administrator error message.



Login Failed

Login Failed!

Please check your username and password and try again

© 2012-2013 Hacklab Pizza. All Rights Reserved

Figure 15: Administrator Login failed.

After 20 tries to guess the administrator password there were not any type of lock-out attempts which means it can be subjected to a brute force attack.

This made the attacker go back to the normal user login page and try an incorrect password to the Mr Rick Astley account previously provided (username *hacklab@hacklab.com*). This process consisted of trying the wrong password to the user 10 times in search of any user lock-out attempts that could prevent the account and the web application from a brute force account and several other attacks.

3.4 Authorization Testing

Owasp refers to authorization testing as the assessment of access control on a web application. This test determines what actions or resources a determined type of user is permitted to access. Authorization testing identified vulnerabilities and weaknesses in the access control list that may lead a user to unauthorized sensitive information. An attacker gaining unauthorized access to sensitive information can lead to data manipulation or complete compromisation of the web application.

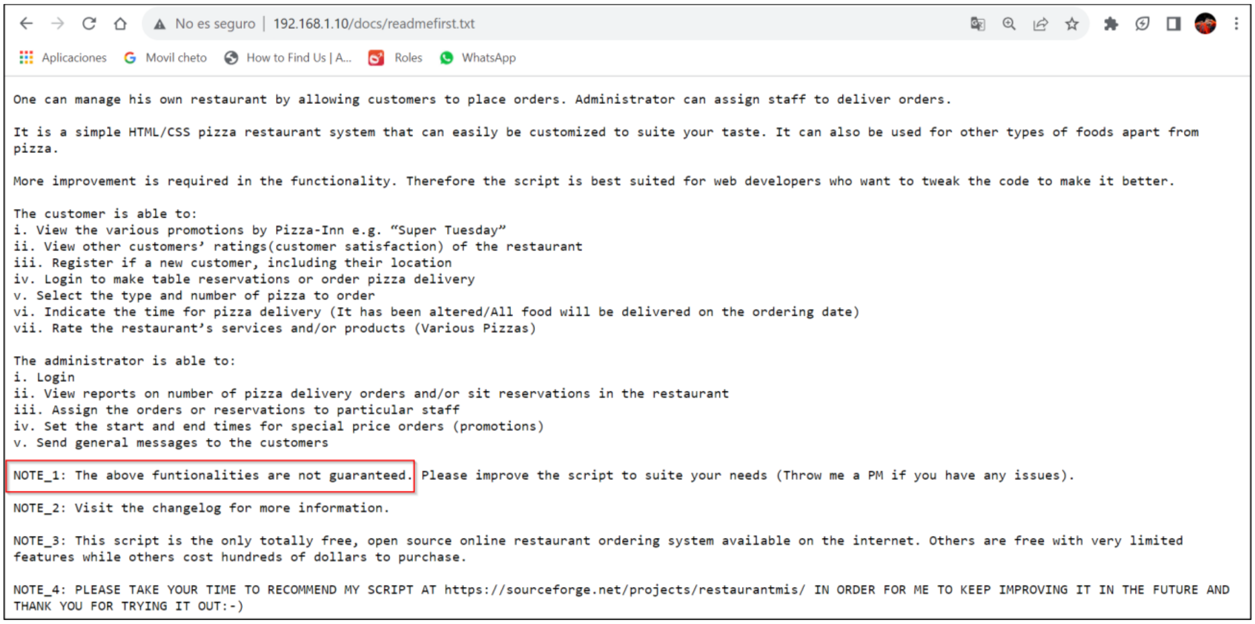
Inside the Dirburster scan that can be found in Appendix B.1, there is a directory called /docs/readmefirst.txt. This was accessed and the information of each account role is displayed and available to the public.

Here is what the customer (normal user) can do on the web application:

The customer can:
I. View the various promotions by Pizza-Inn e.g. “Super Tuesday”.
ii. View other customers’ ratings (customer satisfaction) of the restaurant
iii. Register if a new customer, including their location.
iv. Login to make table reservations or order pizza delivery.
v. Select the type and number of pizzas to order.
vi. Indicate the time for pizza delivery (It has been altered/All food will be delivered on the ordering date).
vii. Rate the restaurant’s services and/or products (Various Pizzas).

Here is what an Administrator can do on the web application:

The administrator can:
I. Login
ii. View reports on number of pizza delivery orders and/or sit reservations in the restaurant.
iii. Assign the orders or reservations to particular staff.
iv. Set the start and end times for special price orders (promotions).
v. Send general messages to the customers.



```
One can manage his own restaurant by allowing customers to place orders. Administrator can assign staff to deliver orders.

It is a simple HTML/CSS pizza restaurant system that can easily be customized to suite your taste. It can also be used for other types of foods apart from
pizza.

More improvement is required in the functionality. Therefore the script is best suited for web developers who want to tweak the code to make it better.

The customer is able to:
i. View the various promotions by Pizza-Inn e.g. "Super Tuesday"
ii. View other customers' ratings(customer satisfaction) of the restaurant
iii. Register if a new customer, including their location
iv. Login to make table reservations or order pizza delivery
v. Select the type and number of pizza to order
vi. Indicate the time for pizza delivery (It has been altered/All food will be delivered on the ordering date)
vii. Rate the restaurant's services and/or products (Various Pizzas)

The administrator is able to:
i. Login
ii. View reports on number of pizza delivery orders and/or sit reservations in the restaurant
iii. Assign the orders or reservations to particular staff
iv. Set the start and end times for special price orders (promotions)
v. Send general messages to the customers

NOTE_1: The above funtionalities are not guaranteed. Please improve the script to suite your needs (Throw me a PM if you have any issues).

NOTE_2: Visit the changelog for more information.

NOTE_3: This script is the only totally free, open source online restaurant ordering system available on the internet. Others are free with very limited
features while others cost hundreds of dollars to purchase.

NOTE_4: PLEASE TAKE YOUR TIME TO RECOMMEND MY SCRIPT AT https://sourceforge.net/projects/restaurantmis/ IN ORDER FOR ME TO KEEP IMPROVING IT IN THE FUTURE AND
THANK YOU FOR TRYING IT OUT:-)
```

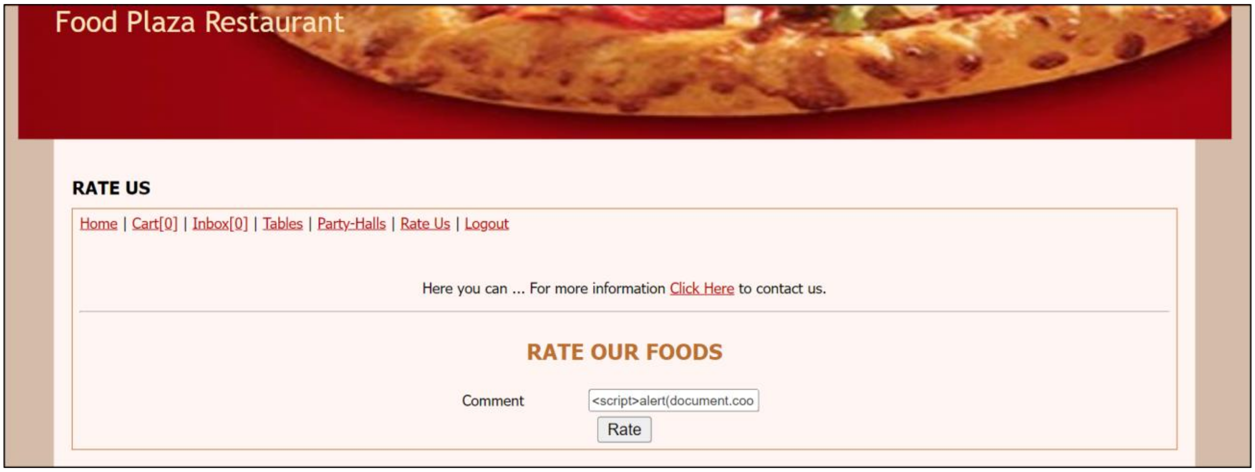
Figure 16: Information about roles

3.5 Input Validation Testing

As seen in the Owasp top 10, cross-site scripting is a very well-known vulnerability and is categorised as an injection. 94% of the applications were tested for some form of injection. Cross-site scripting is a type of malicious injection where the attacker injects a script into a trusted website, and this is then executed when the browser is accessed.

The script that was used is the following: *<script>alert(document.coockie)<script>.* This script was inserted into a comment field to rate the food. As user input was not properly sanitised, the script was stored on the database.

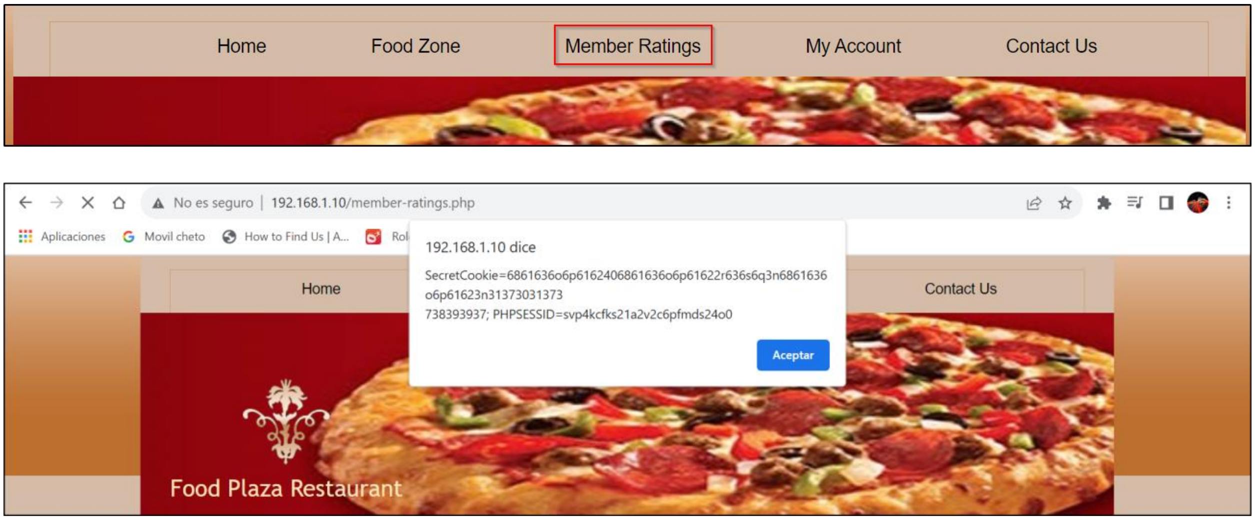
At first, it appeared that nothing happened, but when the link to the member rating was accessed, the script pulled out an alert with the Secret Cookie information and the PHPSESSID. PHPSESSID is always one of every attacker's main target to find.


Food Plaza Restaurant
RATE US
Home | Cart[0] | Inbox[0] | Tables | Party-Halls | Rate Us | Logout

Here you can ... For more information Click Here to contact us.

Comment <script>alert(document.coo

Figure 17: Script to alert cookie information



192.168.1.10 dice

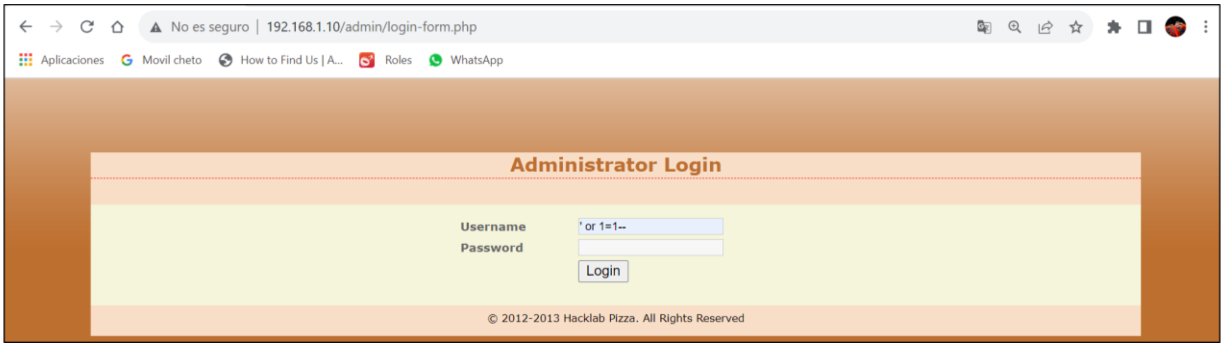
```
SecretCookie=6861636o6p6162406861636o6p61622r636s6q3n6861636
o6p61623n31373031373
738393937; PHPSESSID=svp4kcfks21a2v2c6pfmds24o0
```

Figure 18: Secret information and phpsessid information

Another type of input validation flaw that Hacklab Pizza is subject to is an SQLi. Structure query language injection is when an attacker modifies the structure of a request by including code to the original or excluding a line from the original. The main types of SQLi are username' or 1=1— and username '1=1''.

The first one that was tried was username' or 1=1- -. This line is intended to comment out the query where a password is required and to add a query where if username OR 1=1 is correct access should be granted. Upon entering this line into the username field an error message showed up, the error message can be found in Figure 21. The error message leads the attacker to use a second approach.

The second approach was to use the username '1=1''. This line is intended to add a query that verifies if the username and password are correct or verifies if 1=1 is correct. As 1=1 access was granted to the administrator who was able to create a special offer and check all the information available on the administrator page.

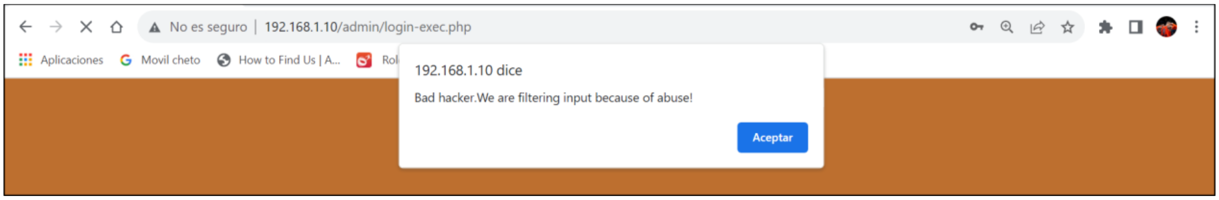


Administrator Login

Username ' or 1=1--
Password
Login

© 2012-2013 Hacklab Pizza. All Rights Reserved

Figure 19: SQLi attempt n°1

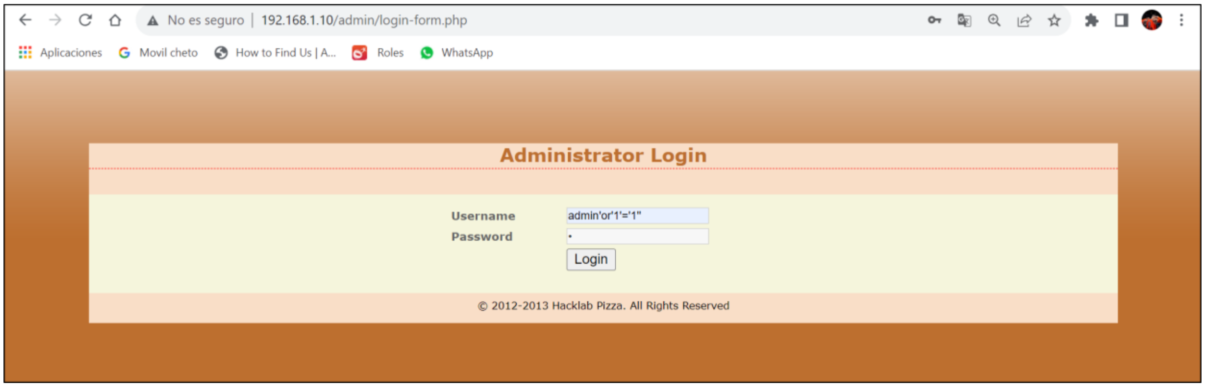


192.168.1.10 dice

Bad hacker.We are filtering input because of abuse!

Aceptar

Figure 20: Error message from attempt n°1

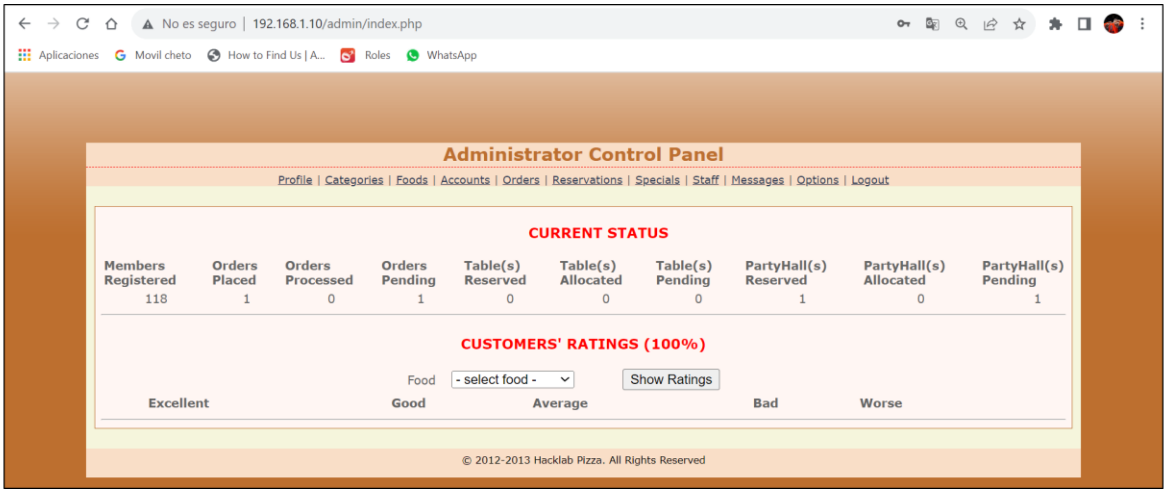


Administrator Login

Username admin'or'1'='1"
Password •
Login

© 2012-2013 Hacklab Pizza. All Rights Reserved

Figure 21: SQL attempt n°2



Administrator Control Panel

Profile | Categories | Foods | Accounts | Orders | Reservations | Specials | Staff | Messages | Options | Logout

CURRENT STATUS

Members Registered	Orders Placed	Orders Processed	Orders Pending	Table(s) Reserved	Table(s) Allocated	Table(s) Pending	PartyHall(s) Reserved	PartyHall(s) Allocated	PartyHall(s) Pending
118	1	0	1	0	0	0	1	0	1

CUSTOMERS' RATINGS (100%)

Food - select food - Show Ratings

Excellent	Good	Average	Bad	Worse

© 2012-2013 Hacklab Pizza. All Rights Reserved

Figure 22: SQL attempt n°2 result

4 DISCUSSION

4.1 OVERALL DISCUSSION

After all the testing phases it has been proved that the Hacklab Pizza web application is too vulnerable. Suffering from misconfiguration, outdated versions of services, input validation attacks and loose or non-existent password policy.

4.1.1 Information Gathering Discussion

By visiting the Hacklab Pizza web page it could be observed that it is HTTP. This means that packets travelling are not encrypted and could be susceptible to a man-in-the-middle, session hijacking, attack, or sniffing. HTTPS should be used to avoid the attacks mentioned previously.

During the active information gathering reported that the service is used on the web server out of date. After searching service versions, it appears they are at the end of life. End of life is the stage where a service version is not being supported and updated anymore by the provider.

PHP 5.4 was released over 8 years and three months ago it is now an unsupported branch, with the PHP website recommending upgrading to the current version 8.3.0. Using old versions can lead to multiple vulnerabilities.

The Apache version also could use an upgrade from 2.4.3 to 2.4.58 to enhance security and minimise possible attacks against the web application.

4.1.2 Configuration and Deployment Management Testing Discussion

As seen in the OWASP top 10, security misconfiguration is a problem for the majority of companies. Company sensitive information should be well protected as this is a breach of the GDPR and lead to sanctions.

Hacklab Pizza is not an exception, from having sensitive information public to everyone to having directories with key information for an attacker, Hacklab Pizza could benefit from implementing the following countermeasures.

To enhance security:

1. Configure Server Permissions: Ensure that server configurations are set to deny direct access to sensitive files. This is typically done through server configurations like Apache's `.htaccess`

2. Use Proper Authentication and Authorization: Implement proper user authentication and authorization mechanisms. Users should only be able to access resources to which they have the appropriate permissions.

3. Secure Files: Ensure that configuration files and other sensitive information are stored outside or in a location where they can't be accessed directly via a URL.

4. Minimize Information Exposure: Avoid exposing unnecessary information about the server or application in error messages or other responses. This helps in preventing attackers from gathering useful information for potential attacks.

4.1.3 Authentication Testing Discussion

In this phase, the tester found out that there was a user with a blank name and blank password. The tester also created various accounts. One of them had a last name as a number; a non-valid email and a very weak password were used. The second user had a number as a name, non-valid email and password were not provided and did not have a security question or answer.

Numbers or special characters shouldn't be included in name fields. The e-mail field should verify that it is an appropriate e-mail and should also send a verification request to the e-mail inbox.

There was a very loose or non-existent password policy. One of the users that was created did not input a password and was able to create the account the other user that was created added a three-character password.

A lock out policy was missing; the tester tried a wrong password on the administrator log in page 20 times. This means the administrator log in would be vulnerable to a brute-force attack.

Hacklab Pizza could benefit from the following password policy:

1. Minimum password length of 12 characters.

2. Require a combination of uppercase and lowercase letters.

3. Mandate the use of at least one numerical digit.

4. Enforce the inclusion of special characters in passwords.

5. Implement a password history policy to prevent the reuse of last passwords.

6. Set a password expiration period.

7. Enable account lockout after 5 consecutive failed login attempts.

8. Implement two-factor authentication (2FA) for all user accounts.

4.1.4 Input Validation Testing Discussion

Due to user input not being validated and sanitised, SQLi and XSS attacks were used to gain both secret cookies and phpsessid and access to the administrator's home page.

Unchecked user input to the database should not be allowed. Every variable that passes into the application should be sanitized and validated. The user input that is passed into the database should be quoted and tagged for control purposes.

Here are countermeasures to mitigate or minimise input validation vulnerabilities:

1. Whitelisting: Allow only specified characters or patterns of input and reject everything else.

2. Blacklisting: Identify and block known malicious input patterns, such as common SQL injection or cross-site scripting (XSS) strings.

3. Regular Expressions: Use regular expressions to define and validate the expected format of input data.

4. Use parameterized queries or prepared statements to mitigate SQL injection vulnerabilities.

5. Check the length of input data to ensure it falls within acceptable limits.

6. Validate that the input matches the expected data type to prevent type-related vulnerabilities.

7. When dealing with file uploads, verify the file type and size to prevent potential security risks associated with malicious files.

4.1.5 Testing for Error Handling Discussion

Error messages also provide the attacker with more information to guide them toward their goal of exploiting the web application. This should be prevented.

Implementing proper error handling to avoid revealing sensitive information and provide meaningful error messages to users. Instead of the error message “username not found” it should display “login failed” or “incorrect credentials”.

4.2 FUTURE WORK

In future work, in other to find more vulnerabilities the tester would have cracked the cookie using the tool chefcyber. The tester would have also tried to brute-force the administrator password using tools such as Hydra and OWASP Zap.

In the future, the outdated services from the server could have been taken advantage of. Using known exploits to gain access and raise awareness as to why they need to be updated regularly.

Other vulnerabilities that were not found and covered during the testing would have been found using other tools or methodologies and assessing how weak the web application is.

Finally, countermeasures to the new finding would have been provided to help prevent or fully mitigate possible attacks.

REFERENCES PART 1

Statista - Most common vulnerabilities in 2022 [Accessed 27 November 2023] https://www.statista.com/statistics/806081/worldwide-application-vulnerability-taxonomy/

OWASP top 10. 2021. [online]. https://owasp.org/Top10/ [Accessed 27 November 2023]

OWASP web Security Testing Guide [online guide] https://owasp.org/www-project-web-security-testing-guide/assets/archive/OWASP_Testing_Guide_v4.pdf [Accessed 27 November 2023]

OWASP web Security Testing Guide v4.2 [online guide] https://owasp.org/www-project-web-security-testing-guide/v42/

Lionel Sujay Vailshery 2023. Distribution of web application critical vulnerabilities [online image]Available from: https://www.statista.com/statistics/806081/worldwide-application-vulnerability-taxonomy/ [Accessed 3 December 2023]

PHP End of life list [online] https://www.php.net/eol.php [Accessed 10 December 2023]

Apache server vulnerabilities update [online] https://httpd.apache.org/security/vulnerabilities_24.html [Accessed 10 December 2023]

APPENDICES

APPENDIX A - OWASP TOP 10

OWASP top 10	Description
Broken Access Control (94% of applications were tested for some form of broken access control)	This occurs when ACL are not properly implemented, allowing unauthorized users to access sensitive information or perform actions they shouldn't be able to.
Cryptographic Failures	This refers to weaknesses in the implementation of cryptographic algorithms, which can lead to data breaches or unauthorized access to sensitive information.
Injection (94% of the applications were tested for some form of injection)	Injection vulnerabilities occur when untrusted data is sent to an interpreter as part of a command or query, to execute malicious code or manipulate the application's behaviour.
Insecure Design	Insecure design refers to security flaws that are inherent in the design of an application, such as using weak encryption algorithms or not properly validating user input.
Security Misconfiguration (90% of applications were tested for some form of misconfiguration)	Security misconfiguration happens when security settings are not properly configured, leaving the application vulnerable to attacks or unauthorized access.
Vulnerable and Outdated Components	refer to potential outdated software or libraries used to build a web application.
Identification and Authentication Failures	occurs when the application fails to properly verify the identity of a user.
Software and Data Integrity Failures	focusing on making assumptions related to software updates, critical data, and CI/CD pipelines without verifying integrity
Security Logging and Monitoring Failures	Occurs when an application is unable to effectively record and analyse security events
Server-Side Request Forgery	When the attacker manipulates the server request into sending it to external sources.

APPENDIX B – INFORMATION GATHERING

APPENDIX B.1 – ACTIVE INFORMATION GATHERING

OWASP ZAP SPIDER URLS

```
http://192.168.1.10
http://192.168.1.10/
http://192.168.1.10/aboutus.php
http://192.168.1.10/access-denied.php
http://192.168.1.10/affix.php?type=terms.php
http://192.168.1.10/cart-exec.php?id=1
http://192.168.1.10/company-accounts
http://192.168.1.10/company-accounts/
http://192.168.1.10/company-accounts/?C=D;O=D
http://192.168.1.10/company-accounts/finances.zip
http://192.168.1.10/company-accounts/readme.txt
http://192.168.1.10/contactus.php
http://192.168.1.10/foodzone.php
http://192.168.1.10/icons
http://192.168.1.10/icons/
http://192.168.1.10/icons/back.gif
http://192.168.1.10/icons/blank.gif
http://192.168.1.10/icons/compressed.gif
http://192.168.1.10/icons/text.gif
http://192.168.1.10/images
http://192.168.1.10/images/
http://192.168.1.10/images/img001.png
http://192.168.1.10/images/img002.png
http://192.168.1.10/images/img003.png
http://192.168.1.10/images/img004.png
http://192.168.1.10/images/img005.png
http://192.168.1.10/images/img006.png
http://192.168.1.10/images/img007.png
http://192.168.1.10/images/img008.png
http://192.168.1.10/images/img009.png
http://192.168.1.10/images/img010.png
http://192.168.1.10/images/img011.png
http://192.168.1.10/images/img012.png
http://192.168.1.10/images/img013.png
http://192.168.1.10/images/img014.png
http://192.168.1.10/images/img015.png
http://192.168.1.10/images/img016.png
http://192.168.1.10/images/img017.png
http://192.168.1.10/images/img018.png
http://192.168.1.10/images/img019.png
http://192.168.1.10/images/img020.png
http://192.168.1.10/images/img021.png
http://192.168.1.10/images/img022.png
http://192.168.1.10/images/img023.png
http://192.168.1.10/images/img024.png
http://192.168.1.10/images/img025.png
http://192.168.1.10/images/pizza-inn-map4-mombasa-road.png
http://192.168.1.10/index.php
http://192.168.1.10/login-exec.php
http://192.168.1.10/login-register.php
http://192.168.1.10/member-index.php
http://192.168.1.10/member-ratings.php
http://192.168.1.10/register-exec.php
http://192.168.1.10/register-success.php
http://192.168.1.10/robots.txt
http://192.168.1.10/sitemap.xml
http://192.168.1.10/stylesheets
http://192.168.1.10/stylesheets/
http://192.168.1.10/stylesheets/user_styles.css
http://192.168.1.10/swf
http://192.168.1.10/swf/
http://192.168.1.10/swf/swfobject.js
http://192.168.1.10/validation
http://192.168.1.10/validation/
http://192.168.1.10/validation/user.js
```

DIRBUSTER

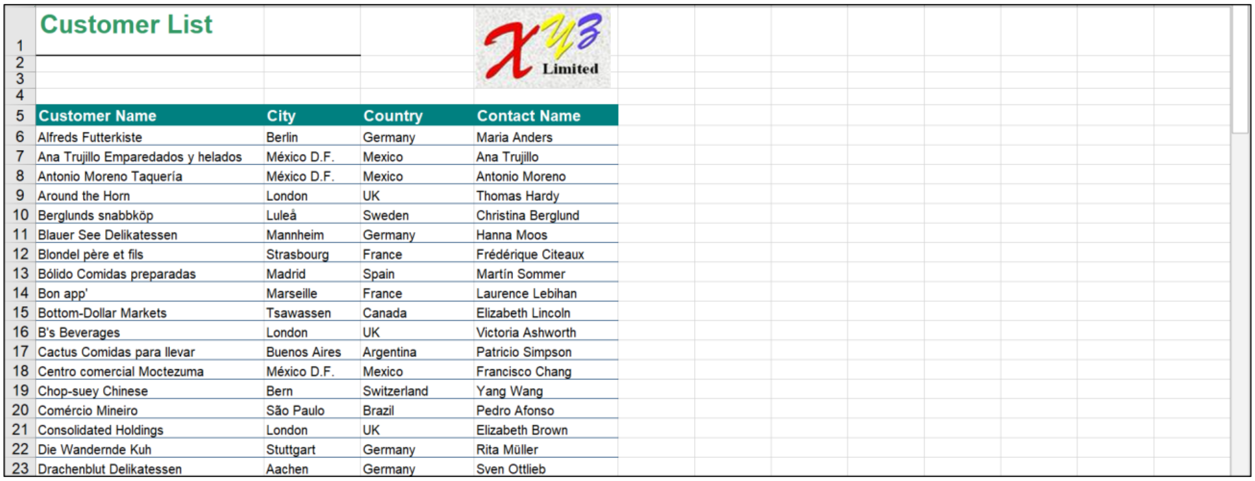
DirBuster 1.0-RC1 - Report

http://www.owasp.org/index.php/Category:OWASP_DirBuster_Project http://192.168.1.10:80

Directories found during testing:

```
/
/images/
/cgi-bin/
/icons/
/docs/
/admin/
/validation/
/swf/
/videos/
/pictures/
/css/
/install/
/js/
/icons/small/
/js/google-code-prettify/
/js/holder/
/admin/validation/
/error/
/error/include/
/images/pizza/
/phpmyadmin/
/connection/
/admin/connection/
/stylesheets/
/admin/stylesheets/
/index.php
/gallery.php
/footer.php
/cart.php
/logout.php
/ratings.php
/cookie.php
/auth.php
/aboutus.php
/foodzone.php
/member-ratings.php
/member-index.php
/affix.php
/contactus.php
/login-register.php
/login-exec.php
/register-exec.php
/validation/user.js
/admin/index.php
/info.php
/admin/profile.php
/swf/swfobject.js
/docs/changelog.txt
/docs/install.txt
/docs/readmefirst.txt
/terms.php
/docs/support.txt
/tables.php
/docs/~$C%20409%20TERM%20PROJECTJan%202012.doc
/username.php
/swf/Carousel.swf
/admin/categories.php
/swf/default.xml
/inbox.php
/partyhalls.php
/reserve-exec.php
/css/DT_bootstrap.css
/css/bootstrap-responsive.css
/css/bootstrap.css
/css/datepicker.css
/css/demo.css
/css/diapo.css
/admin/specials.php
/css/docs.css
/css/font-awesome.css
/install/coming_soon.txt
/css/normalize.css
/css/style.css
/js/DT_bootstrap.js
/js/application.js
/js/bootstrap-affix.js
/js/bootstrap-alert.js
/js/bootstrap-button.js
/js/bootstrap-carousel.js
/admin/messages.php
/js/bootstrap-collapse.js
/js/bootstrap-dropdown.js
/js/bootstrap-modal.js
/js/bootstrap-popover.js
/js/bootstrap-scrollspy.js
/js/bootstrap-tab.js
/js/bootstrap-tooltip.js
/js/bootstrap-transition.js
/js/bootstrap-typeahead.js
/js/bootstrap.js
/js/bootstrap.min.js
/js/datepicker.js
/js/html5shiv.js
/js/jquery-1.7.2.min.js
/js/jquery-1.10.2.min.js
/js/jquery.dataTables.js
/admin/logout.php
/js/jquery.easing.1.3.js
/js/jquery.hoverIntent.minified.js
/js/jquery.hoverdir.js
/js/jquery.js
/js/google-code-prettify/prettify.css
/js/jquery.mobile-1.0rc2.customized.min.js
/js/holder/holder.js
/js/google-code-prettify/prettify.js
/admin/login-form.php
/instructions.php
/admin/accounts.php
/admin/options.php
/admin/login-exec.php
/admin/validation/admin.js
/admin/orders.php
/admin/auth.php
/admin/reservations.php
/images/pizza/Romans.xcf
/connection/config.php
/admin/connection/config.php
/stylesheets/user_styles.css
/admin/stylesheets/admin_styles.css
/phpinfo.php
/specialdeals.php
/admin/foods.php
```

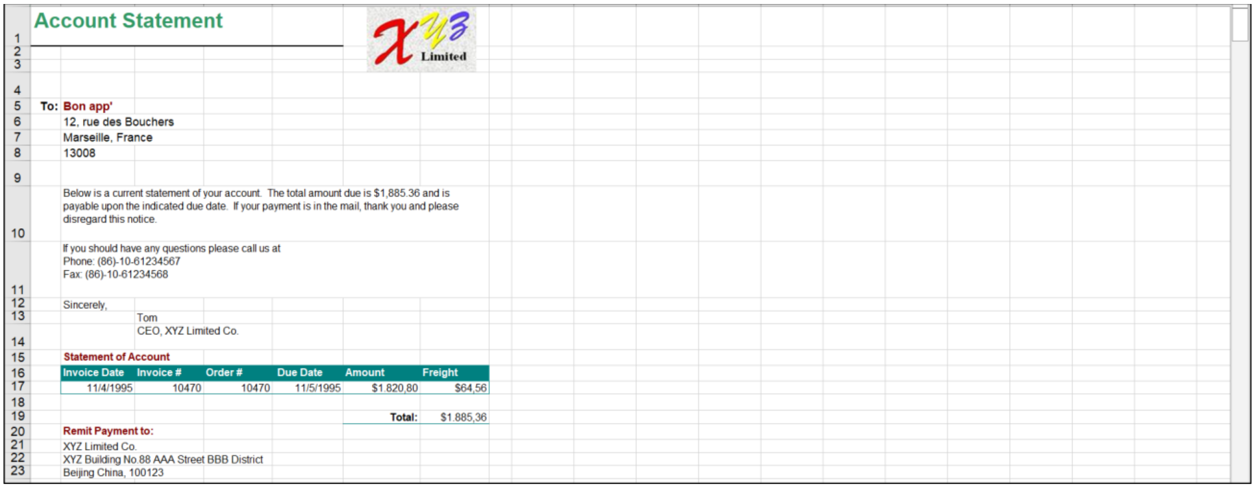
APPENDIX B.2. – COMPANY ACCOUNT (ZIP FILE)



Customer List

Customer Name	City	Country	Contact Name
Alfreds Futterkiste	Berlin	Germany	Maria Anders
Ana Trujillo Emparedados y helados	México D.F.	Mexico	Ana Trujillo
Antonio Moreno Taquería	México D.F.	Mexico	Antonio Moreno
Around the Horn	London	UK	Thomas Hardy
Berglunds snabbköp	Luleå	Sweden	Christina Berglund
Blauer See Delikatessen	Mannheim	Germany	Hanna Moos
Blondel père et fils	Strasbourg	France	Frédérique Citeaux
Bólido Comidas preparadas	Madrid	Spain	Martín Sommer
Bon app'	Marseille	France	Laurence Lebihan
Bottom-Dollar Markets	Tsawassen	Canada	Elizabeth Lincoln
B's Beverages	London	UK	Victoria Ashworth
Cactus Comidas para llevar	Buenos Aires	Argentina	Patricio Simpson
Centro comercial Moctezuma	México D.F.	Mexico	Francisco Chang
Chop-suey Chinese	Bern	Switzerland	Yang Wang
Comércio Mineiro	São Paulo	Brazil	Pedro Afonso
Consolidated Holdings	London	UK	Elizabeth Brown
Die Wandernde Kuh	Stuttgart	Germany	Rita Müller
Drachenblut Delikatessen	Aachen	Germany	Sven Ottlieb

Figure B.2.1: Customer list spreadsheet from zip file



Account Statement

To: Bon app'
12, rue des Bouchers
Marseille, France
13008

Below is a current statement of your account. The total amount due is $1,885.36 and is payable upon the indicated due date. If your payment is in the mail, thank you and please disregard this notice.

If you should have any questions please call us at
Phone: (86)-10-61234567
Fax: (86)-10-61234568

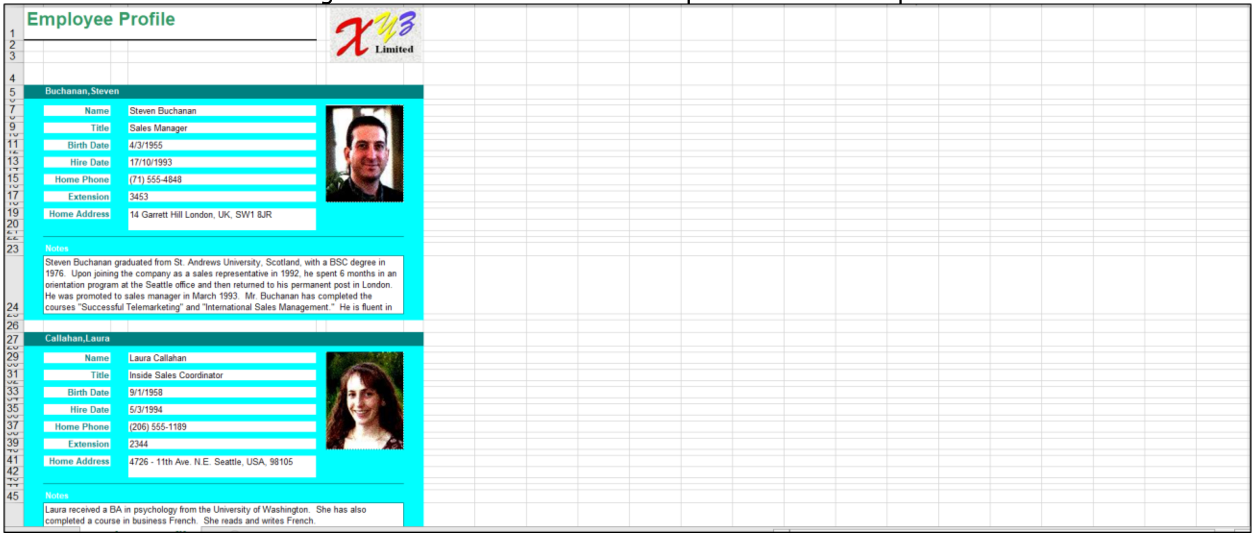
Sincerely,
Tom
CEO, XYZ Limited Co.

Statement of Account

Invoice Date	Invoice #	Order #	Due Date	Amount	Freight
11/4/1995	10470	10470	11/5/1995	$1.820,80	$64,56
				Total:	$1.885,36

Remit Payment to:
XYZ Limited Co.
XYZ Building No.88 AAA Street BBB District
Beijing China, 100123

Figure B.2.2: Account statements spreadsheet from zip file



Employee Profile

Buchanan,Steven

Name	Steven Buchanan
Title	Sales Manager
Birth Date	4/3/1955
Hire Date	17/10/1993
Home Phone	(71) 555-4848
Extension	3453
Home Address	14 Garrett Hill London, UK, SW1 8JR

Notes

Steven Buchanan graduated from St. Andrews University, Scotland, with a BSC degree in 1976. Upon joining the company as a sales representative in 1992, he spent 6 months in an orientation program at the Seattle office and then returned to his permanent post in London. He was promoted to sales manager in March 1993. Mr. Buchanan has completed the courses "Successful Telemarketing" and "International Sales Management." He is fluent in

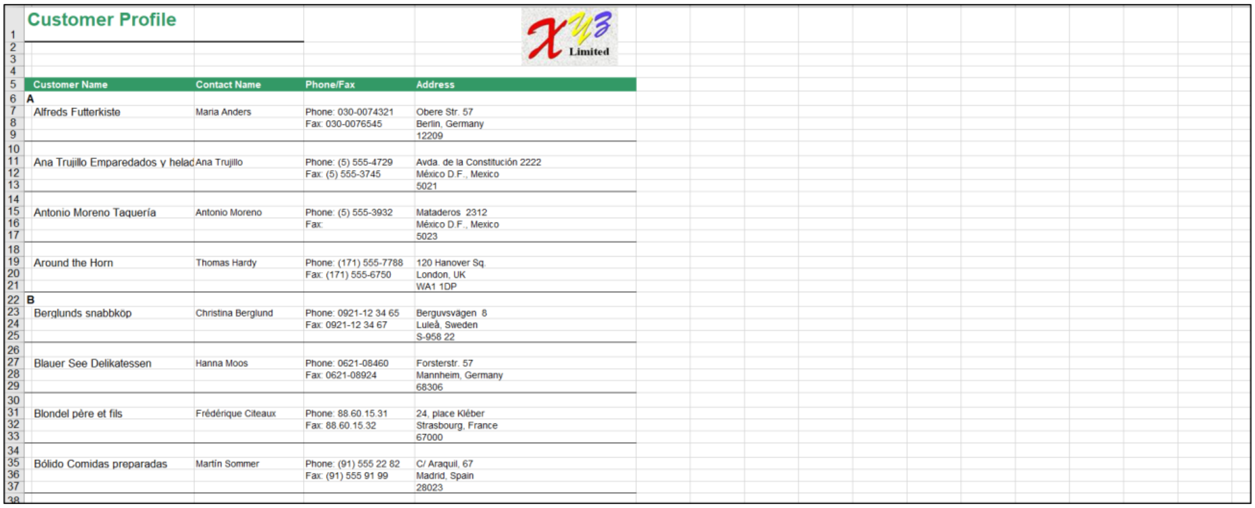
Callahan,Laura

Name	Laura Callahan
Title	Inside Sales Coordinator
Birth Date	9/1/1958
Hire Date	5/3/1994
Home Phone	(206) 555-1189
Extension	2344
Home Address	4726 - 11th Ave. N.E. Seattle, USA, 98105

Notes

Laura received a BA in psychology from the University of Washington. She has also completed a course in business French. She reads and writes French.

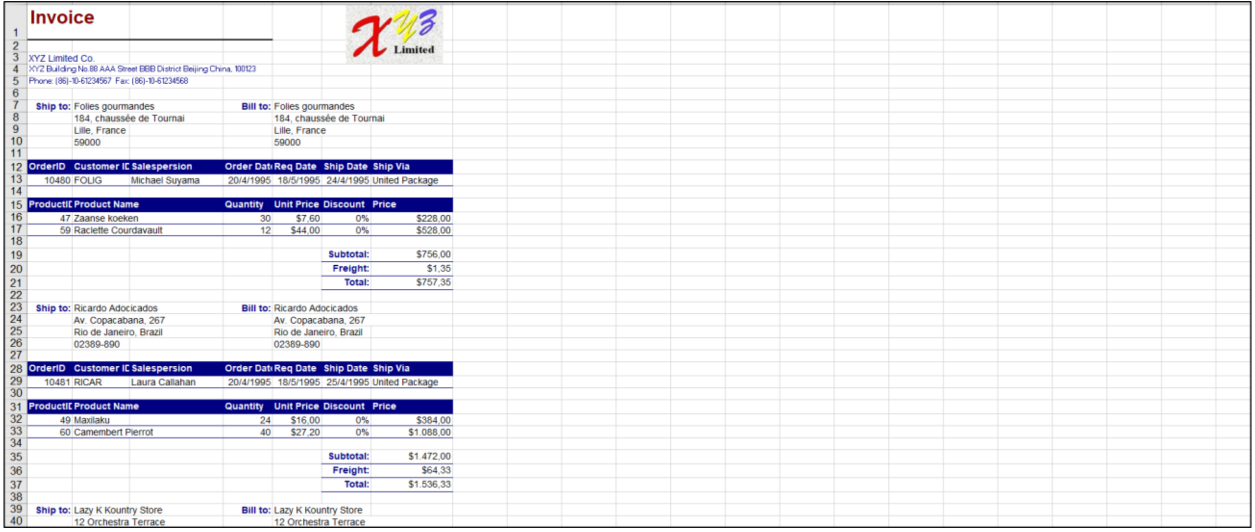
Figure B.2.3: Employee profile spreadsheet from zip file



Customer Profile

Customer Name	Contact Name	Phone/Fax	Address
A			
Alfreds Futterkiste	Maria Anders	Phone: 030-0074321 Fax: 030-0076545	Obere Str. 57 Berlin, Germany 12209
Ana Trujillo Emparedados y helad	Ana Trujillo	Phone: (5) 555-4729 Fax: (5) 555-3745	Avda. de la Constitución 2222 México D.F., Mexico 5021
Antonio Moreno Taquería	Antonio Moreno	Phone: (5) 555-3932 Fax:	Mataderos 2312 México D.F., Mexico 5023
Around the Horn	Thomas Hardy	Phone: (171) 555-7788 Fax: (171) 555-6750	120 Hanover Sq. London, UK WA1 1DP
B			
Berglunds snabbköp	Christina Berglund	Phone: 0921-12 34 65 Fax: 0921-12 34 67	Berguvsvägen 8 Luleå, Sweden S-958 22
Blauer See Delikatessen	Hanna Moos	Phone: 0621-08460 Fax: 0621-08924	Forsterstr. 57 Mannheim, Germany 68306
Blondel père et fils	Frédérique Citeaux	Phone: 88.60.15.31 Fax: 88.60.15.32	24, place Kléber Strasbourg, France 67000
Bólido Comidas preparadas	Martín Sommer	Phone: (91) 555 22 82 Fax: (91) 555 91 99	C/ Araquil, 67 Madrid, Spain 28023

Figure B.2.4: Customer profile spreadsheet from zip file



Invoice

XYZ Limited Co.
XYZ Building No.88 AAA Street BBB District Beijing China, 100123
Phone: (86)-10-61234567 Fax: (86)-10-61234568

Ship to:	Folies gourmandes 184, chaussée de Tournai Lille, France 59000	**Bill to:**	Folies gourmandes 184, chaussée de Tournai Lille, France 59000

OrderID	Customer IC	Salespersion	Order Dat	Req Date	Ship Date	Ship Via
10480	FOLIG	Michael Suyama	20/4/1995	18/5/1995	24/4/1995	United Package

ProductIC	Product Name	Quantity	Unit Price	Discount	Price
47	Zaanse koeken	30	$7,60	0%	$228,00
59	Raclette Courdavault	12	$44,00	0%	$528,00
				Subtotal:	$756,00
				Freight:	$1,35
				Total:	$757,35

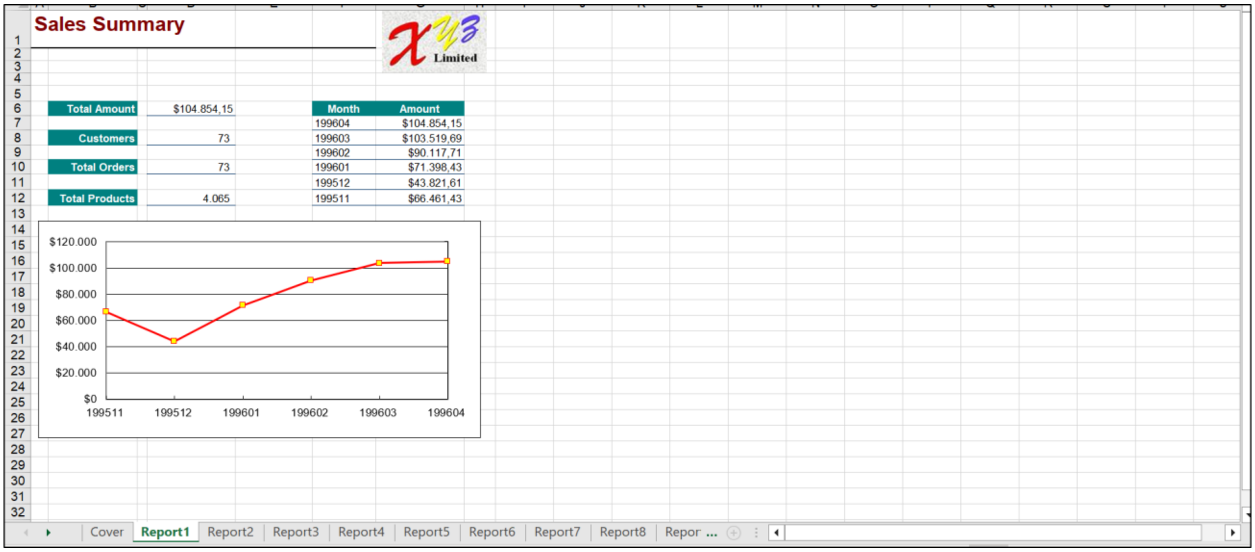
Ship to:	Ricardo Adocicados Av. Copacabana, 267 Rio de Janeiro, Brazil 02389-890	**Bill to:**	Ricardo Adocicados Av. Copacabana, 267 Rio de Janeiro, Brazil 02389-890

OrderID	Customer IC	Salespersion	Order Dat	Req Date	Ship Date	Ship Via
10481	RICAR	Laura Callahan	20/4/1995	18/5/1995	25/4/1995	United Package

ProductIC	Product Name	Quantity	Unit Price	Discount	Price
49	Maxilaku	24	$16,00	0%	$384,00
60	Camembert Pierrot	40	$27,20	0%	$1.088,00
				Subtotal:	$1.472,00
				Freight:	$64,33
				Total:	$1.536,33

Ship to:	Lazy K Kountry Store 12 Orchestra Terrace	**Bill to:**	Lazy K Kountry Store 12 Orchestra Terrace

Figure B.2.5: Invoice spreadsheet from zip file

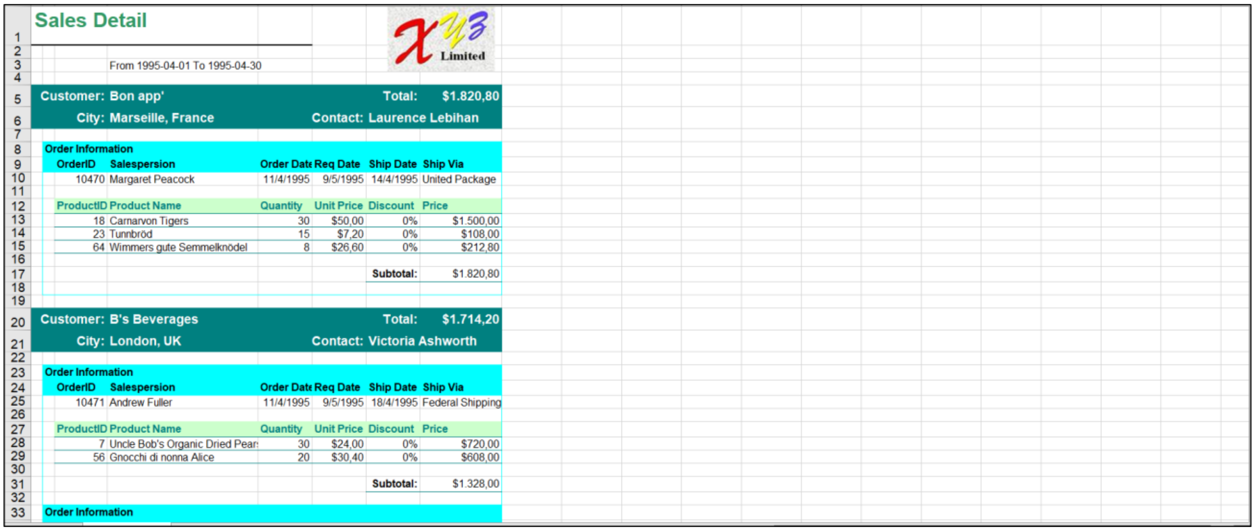


Sales Summary

Total Amount	$104.854,15
Customers	73
Total Orders	73
Total Products	4.065

Month	Amount
199604	$104.854,15
199603	$103.519,69
199602	$90.117,71
199601	$71.398,43
199512	$43.821,61
199511	$66.461,43

Figure B.2.6: Sales Summary spreadsheet from zip file



Sales Detail

From 1995-04-01 To 1995-04-30

Customer: Bon app' Total: $1.820,80

City: Marseille, France Contact: Laurence Lebihan

Order Information

OrderID	Salespersion	Order Date	Req Date	Ship Date	Ship Via
10470	Margaret Peacock	11/4/1995	9/5/1995	14/4/1995	United Package

ProductID	Product Name	Quantity	Unit Price	Discount	Price
18	Carnarvon Tigers	30	$50,00	0%	$1.500,00
23	Tunnbröd	15	$7,20	0%	$108,00
64	Wimmers gute Semmelknödel	8	$26,60	0%	$212,80
				Subtotal:	$1.820,80

Customer: B's Beverages Total: $1.714,20

City: London, UK Contact: Victoria Ashworth

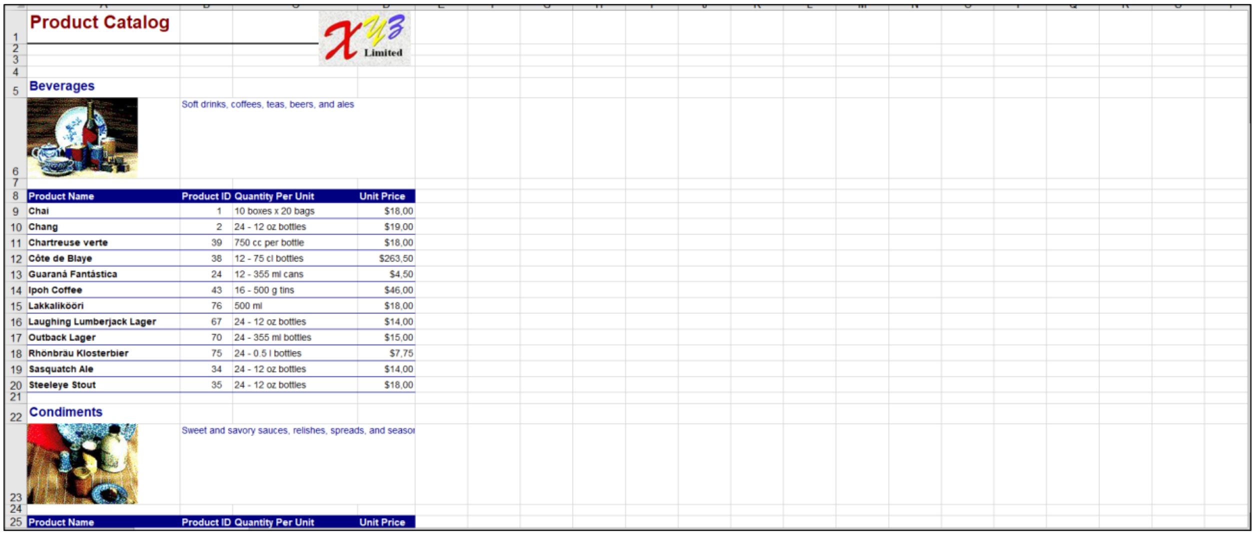
Order Information

OrderID	Salespersion	Order Date	Req Date	Ship Date	Ship Via
10471	Andrew Fuller	11/4/1995	9/5/1995	18/4/1995	Federal Shipping

ProductID	Product Name	Quantity	Unit Price	Discount	Price
7	Uncle Bob's Organic Dried Pear:	30	$24,00	0%	$720,00
56	Gnocchi di nonna Alice	20	$30,40	0%	$608,00
				Subtotal:	$1.328,00

Order Information

Figure B.2.7: Sales details spreadsheet from zip file



Product Catalog

Beverages

Soft drinks, coffees, teas, beers, and ales

Product Name	Product ID	Quantity Per Unit	Unit Price
Chai	1	10 boxes x 20 bags	$18,00
Chang	2	24 - 12 oz bottles	$19,00
Chartreuse verte	39	750 cc per bottle	$18,00
Côte de Blaye	38	12 - 75 cl bottles	$263,50
Guaraná Fantástica	24	12 - 355 ml cans	$4,50
Ipoh Coffee	43	16 - 500 g tins	$46,00
Lakkalikööri	76	500 ml	$18,00
Laughing Lumberjack Lager	67	24 - 12 oz bottles	$14,00
Outback Lager	70	24 - 355 ml bottles	$15,00
Rhönbräu Klosterbier	75	24 - 0.5 l bottles	$7,75
Sasquatch Ale	34	24 - 12 oz bottles	$14,00
Steeleye Stout	35	24 - 12 oz bottles	$18,00

Condiments

Sweet and savory sauces, relishes, spreads, and seaso

Product Name	Product ID	Quantity Per Unit	Unit Price

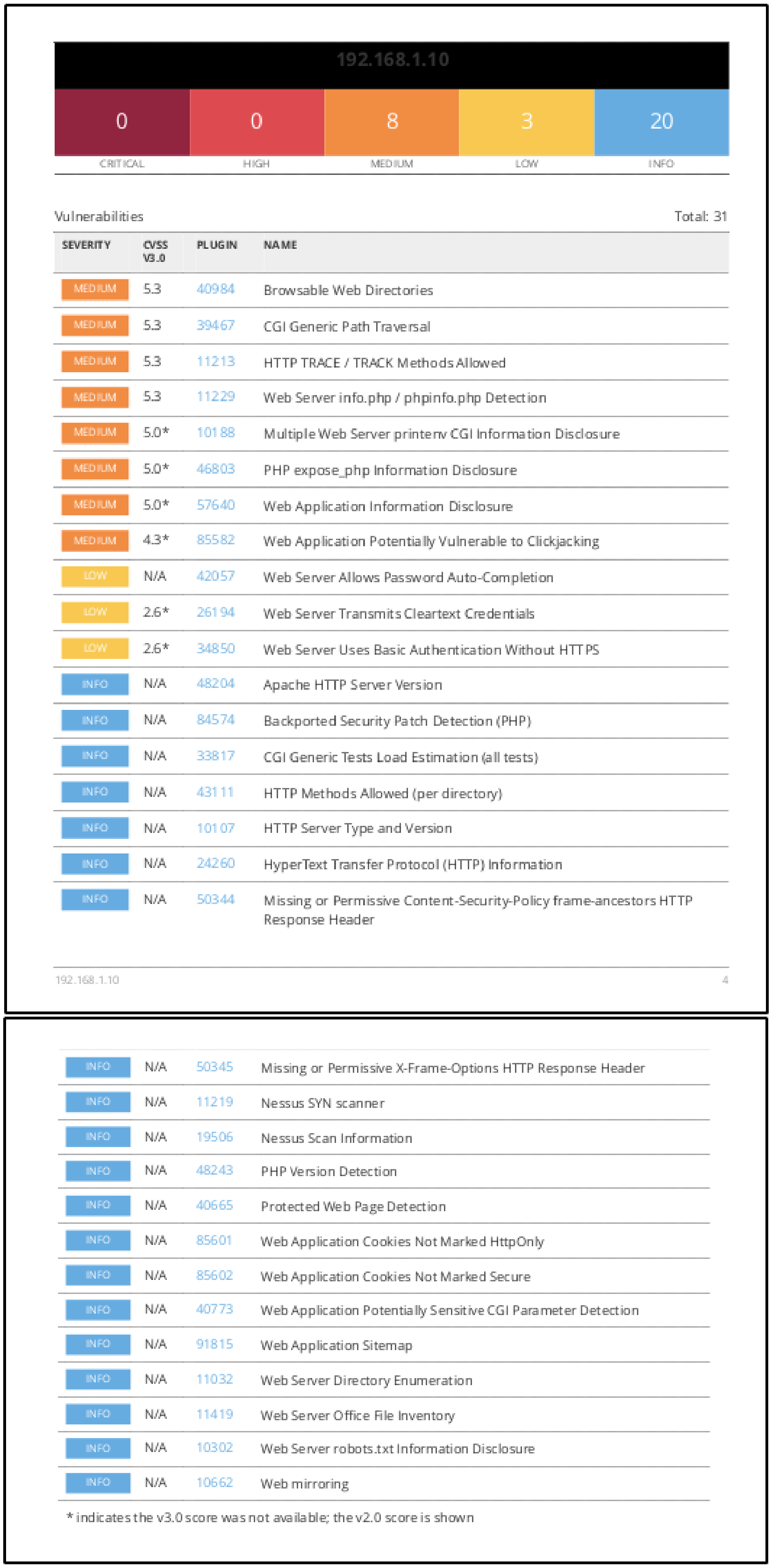
Figure B.2.8: product catalogue spreadsheet from zip file

APPENDIX C – NIKTO SCAN

```
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.1.10
+ Target Hostname:    192.168.1.10
+ Target Port:        80
+ Start Time:         2023-12-07 18:53:24 (GMT-5)
---------------------------------------------------------------------------
+ Server: Apache/2.4.3 (Unix) PHP/5.4.7
+ Retrieved x-powered-by header: PHP/5.4.7
+ 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
+ OSVDB-3268: /company-accounts/: Directory indexing found.
+ Entry '/company-accounts/' in robots.txt returned a non-forbidden or redirect HTTP code
(200)
+ "robots.txt" contains 1 entry which should be manually viewed.
+ 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
+ Apache/2.4.3 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is
the EOL for the 2.x branch.
+ PHP/5.4.7 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.
+ OSVDB-112004: /cgi-bin/printenv: Site appears vulnerable to the 'shellshock' vulnerability
(http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6271).
+ OSVDB-112004: /cgi-bin/printenv: Site appears vulnerable to the 'shellshock' vulnerability
(http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6278).
+ 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
+ /phpinfo.php: Output from the phpinfo() function was found.
+ Cookie PHPSESSID created without the httponly flag
+ OSVDB-12184: /?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially
sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?=PHPE9568F36-D428-11d2-A769-00AA001ACF42: PHP reveals potentially
sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?=PHPE9568F34-D428-11d2-A769-00AA001ACF42: PHP reveals potentially
sensitive information via certain HTTP requests that contain specific QUERY strings.
```

```
+ OSVDB-12184: /?=PHPE9568F35-D428-11d2-A769-00AA001ACF42: PHP reveals potentially
sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-3268: /css/: Directory indexing found.
+ OSVDB-3092: /css/: This might be interesting...
+ OSVDB-3268: /install/: Directory indexing found.
+ OSVDB-3092: /install/: This might be interesting...
+ OSVDB-3268: /stylesheets/: Directory indexing found.
+ OSVDB-3092: /stylesheets/: This might be interesting...
+ OSVDB-3233: /cgi-bin/printenv: Apache 2.0 default script is executable and gives server
environment variables. All default scripts should be removed. It may also allow XSS types of
attacks. http://www.securityfocus.com/bid/4431.
+ OSVDB-3233: /cgi-bin/test-cgi: Apache 2.0 default script is executable and reveals system
information. All default scripts should be removed.
+ 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: /images/: Directory indexing found.
+ OSVDB-3268: /docs/: Directory indexing found.
+ 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/
+ 9687 requests: 0 error(s) and 35 item(s) reported on remote host
+ End Time:           2023-12-07 18:55:03 (GMT-5) (99 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

APPENDIX D –NESSUS SCAN



192.168.1.10

0	0	8	3	20
CRITICAL	HIGH	MEDIUM	LOW	INFO

Vulnerabilities Total: 31

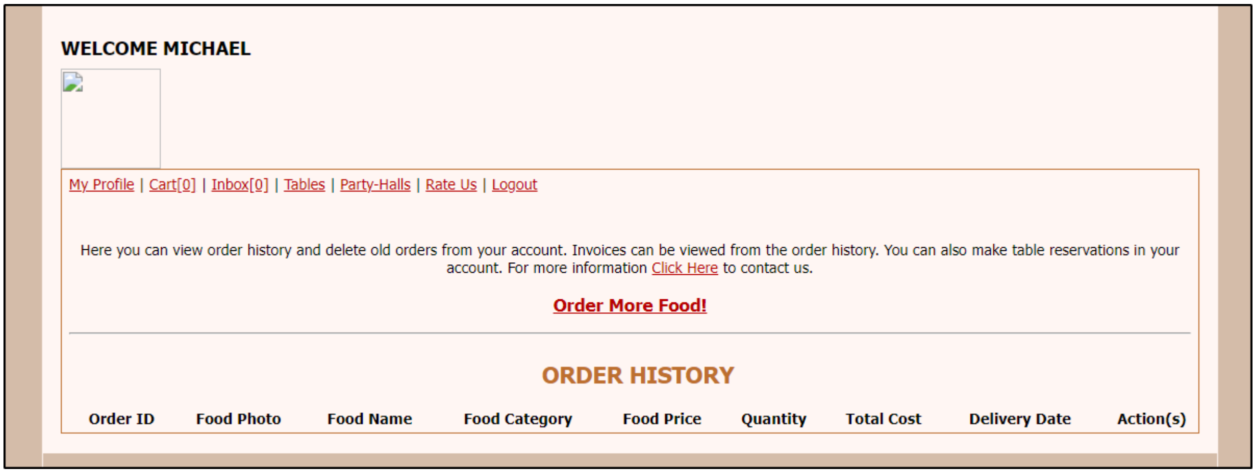
SEVERITY	CVSS V3.0	PLUGIN	NAME
MEDIUM	5.3	40984	Browsable Web Directories
MEDIUM	5.3	39467	CGI Generic Path Traversal
MEDIUM	5.3	11213	HTTP TRACE / TRACK Methods Allowed
MEDIUM	5.3	11229	Web Server info.php / phpinfo.php Detection
MEDIUM	5.0*	10188	Multiple Web Server printenv CGI Information Disclosure
MEDIUM	5.0*	46803	PHP expose_php Information Disclosure
MEDIUM	5.0*	57640	Web Application Information Disclosure
MEDIUM	4.3*	85582	Web Application Potentially Vulnerable to Clickjacking
LOW	N/A	42057	Web Server Allows Password Auto-Completion
LOW	2.6*	26194	Web Server Transmits Cleartext Credentials
LOW	2.6*	34850	Web Server Uses Basic Authentication Without HTTPS
INFO	N/A	48204	Apache HTTP Server Version
INFO	N/A	84574	Backported Security Patch Detection (PHP)
INFO	N/A	33817	CGI Generic Tests Load Estimation (all tests)
INFO	N/A	43111	HTTP Methods Allowed (per directory)
INFO	N/A	10107	HTTP Server Type and Version
INFO	N/A	24260	HyperText Transfer Protocol (HTTP) Information
INFO	N/A	50344	Missing or Permissive Content-Security-Policy frame-ancestors HTTP Response Header
INFO	N/A	50345	Missing or Permissive X-Frame-Options HTTP Response Header
INFO	N/A	11219	Nessus SYN scanner
INFO	N/A	19506	Nessus Scan Information
INFO	N/A	48243	PHP Version Detection
INFO	N/A	40665	Protected Web Page Detection
INFO	N/A	85601	Web Application Cookies Not Marked HttpOnly
INFO	N/A	85602	Web Application Cookies Not Marked Secure
INFO	N/A	40773	Web Application Potentially Sensitive CGI Parameter Detection
INFO	N/A	91815	Web Application Sitemap
INFO	N/A	11032	Web Server Directory Enumeration
INFO	N/A	11419	Web Server Office File Inventory
INFO	N/A	10302	Web Server robots.txt Information Disclosure
INFO	N/A	10662	Web mirroring

192.168.1.10 4

* indicates the v3.0 score was not available; the v2.0 score is shown

Figure D.1&2 Nessus scan result

APPENDIX E – AUTHENTICATION TESTING



WELCOME MICHAEL

My Profile | Cart[0] | Inbox[0] | Tables | Party-Halls | Rate Us | Logout

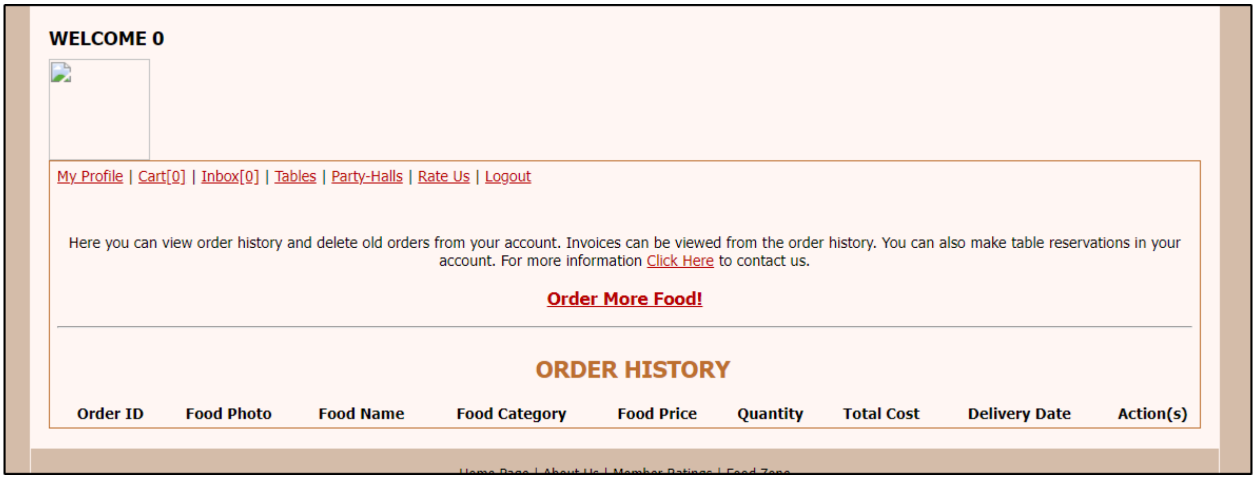
Here you can view order history and delete old orders from your account. Invoices can be viewed from the order history. You can also make table reservations in your account. For more information Click Here to contact us.

Order More Food!

ORDER HISTORY

Order ID	Food Photo	Food Name	Food Category	Food Price	Quantity	Total Cost	Delivery Date	Action(s)

Figure E.1: New user Michael's welcome page



WELCOME 0

My Profile | Cart[0] | Inbox[0] | Tables | Party-Halls | Rate Us | Logout

Here you can view order history and delete old orders from your account. Invoices can be viewed from the order history. You can also make table reservations in your account. For more information Click Here to contact us.

Order More Food!

ORDER HISTORY

Order ID	Food Photo	Food Name	Food Category	Food Price	Quantity	Total Cost	Delivery Date	Action(s)

Home Page | About Us | Member Ratings | Food Zone

Figure E.2: New user 0´s welcome page