



National
College of
Ireland

National College of Ireland
BSc (Honours) in Computing & Cyber Security
2017/2018

Project Title: Penetration-Testing
Final Project Report

Dawid Trojanowski

x14321651

x14321651@student.ncirl.ie

Table of contents

Executive Summary	5
1 Introduction	6
1.1 Background.....	6
1.2 Aims	6
1.3 Technologies	6
1.4 Structure	6
2 System	8
2.1 Requirements	8
2.1.1 Functional requirements	8
2.1.2 Data requirements.....	8
2.1.3 User requirements.....	8
2.1.4 Environmental requirements.....	9
2.1.5 Usability requirements.....	9
2.2 Design and Architecture.....	10
2.3 Implementation	10
2.4 Graphical User Interface (GUI) Layout	11
2.5 Testing	11
3 Conclusions	12
4 Further development or research	13
5 References.....	14
6 Implementation & Design	15
6.1 Database Connection	15
6.2 Registration & Login.....	16
6.3 Navigation Menu	19
6.4 Dashboard	21
6.5 Chat.....	22
6.6 Offers & Subscription	23
6.7 About & Contact pages	25
6.8 Scanners.....	26
7 Vulnerability Testing	29
7.1 Scans performed:.....	29

7.2	Using Sparta / Nikto	29
7.3	Nmap / Sparta	31
7.4	OWASP-ZAP / Burpsuite	31
7.5	Injection.....	32
7.6	RIPS.....	33
8	Appendix	39
8.1	Project Proposal	39
8.2	Objectives	39
8.3	Background.....	39
8.4	Technical Approach	40
8.5	Special resources required	40
8.6	Project Plan	40
8.7	Technical Details	41
8.8	Evaluation	41
8.8.1	Project Plan	41
8.8.2	Purpose.....	41
8.8.3	Project Scope	42
8.9	User Requirements Definition.....	42
8.10	Requirements Specification	42
8.11	Functional requirements	42
8.11.1	Use Case Diagram	43
8.11.2	Requirement 1: Registration	44
8.11.3	Requirement 2: Login System	45
8.11.4	Requirement 3: Credits	47
8.11.5	Requirement 4: Chat	48
8.11.6	Requirement 5: Report errors	49
8.11.7	Requirement 6: Tools.....	50
8.11.8	Requirement 7: Solution	52
8.12	Non-Functional Requirements.....	53
8.12.1	Performance/Response time requirement	53
8.12.2	Availability requirement.....	53
8.12.3	Recover requirement.....	54
8.12.4	Robustness requirement.....	54

8.12.5	Security requirement	54
8.12.6	Reliability requirement.....	54
8.12.7	Maintainability requirement.....	55
8.12.8	Portability requirement	55
8.12.9	Reusability requirement	55
8.12.10	Resource utilization requirement	55
9	Interface requirements	56
9.1	GUI.....	56
9.2	Application Programming Interfaces (API).....	56
10	System Architecture	57
11	System Evolution.....	58
12	Monthly Journals	59
12.1	Reflective Journal	59
12.2	Reflective Journal.....	60
12.3	Reflective Journal.....	61
12.4	Reflective Journal.....	62
12.5	Reflective Journal.....	63
12.6	Reflective Journal.....	65

Executive Summary

The objective of the penetration testing website is to allow the users to test their own website and see how secure the site is. User will have an easy to use website, with simple instruction and layout so that everyone can understand it. The main purpose of the penetration testing website is to make people aware of how security in their own products is important. Penetration testing website allows the user to test their own website using tools like or like SQLMAP, NIKTO. The site will allow users also to communicate with each other on a chat or blog to ask questions and other things. Also, the website will contain a payment method, it will require to user to purchase subscription to the site for the user to be allowed to test their own website. Penetration testing website is developed in PHP, JavaScript, HTML, CSS, jQuery and many others might come during the development path.

1 Introduction

The purpose of my project is to allow the users to test their own websites and see if they are secure, also they will be allowed to chat with each other or have a blog on which they can paste comments and talk with each other to get help. My project will provide few tools that allow to scan/test/find vulnerabilities of the website they own or can test. The aim of my project is to get people more aware of security problems, target group would be adults that own/maintain the website and want to have it secure.

1.1 Background

I am currently in progress of developing my 4th year project, I picked this project because I'm interested in scanning, testing of websites, the whole security aspect. The project is perfect for me because I can learn more about testing tools as it self about the security that's fallows behind it.

1.2 Aims

The aim of this project is to contain a stable system that is going to work for everyone. System that will contain the software's for testing, chat for communication, results and hits given to users for solving their security problems and a "contact us" page to ask for help in relation of something.

1.3 Technologies

The technologies behind it would be: PHP, JavaScript, HTML, CSS, jQuery and many depending on how the project will evolve. The system will be sitting on Linux environment, running from it so that the penetration testing tools can work without problems.

1.4 Structure

The report structure is so that the first section is allowing the user to understand what the project is about, basically what is going on.

In the second section the document will be describing in detail the main background and main aims of the project that might be important to the users, it will describe how will the technologies be implemented.

The third section will be describing the advantages, disadvantages and limits of the project, will explain the strong and weak points of the project.

Fourth section will describe and tell where this project lead to could, how far can It go in next few years, is it going to die or develop to be big.

Section five, references of all used website, tutorials and others that had been used for developing purpose of the site.

Section six are appendix that will contain Project proposal, monthly reports and project specification report.

2 System

2.1 Requirements

This section will be like your original requirements specification. Requirements have probably evolved somewhat since. Where this is the case explain what changed and why.

2.1.1 Functional requirements

The requirements that the project is developing are:

- Registration
- Login system
- Buy subscription
- Chat
- Report errors
- Use tools on the website (penetration-testing tools)
- Getting solution on how to solve the problem.

None of the function requirements have change during the development.

2.1.2 Data requirements

The important aspect will be a backup and recovery, like in any system a backup is a crucial part of functionality. If we don't have back, we can't have a website since when something happens to it, it's all over. The security needs to be strict, data needs to be kept in a secure place away from hackers and people that try to exploit it. Maintaining/reusing of the data space, database should be cleaned or formatted to keep a low storage size.

2.1.3 User requirements

User requirements, the user will be allowed to log in and register, they will have to get the access to the website and the tools. After registration and login, the user will be moved to the main page and will be able to see and use tools (depending on the amount of credit they have). During the movement on the website they will learn on which of the penetration tests they should focus, site

will give them hints in relation of what test to use for what purpose, so that the user won't be just guessing what they should try out. Also, the users will have a chat on which they can talk with each other without any problems. Clients will be able to purchase the site currency by their real money, the use of payment is for allowing the users to test and review their websites, also after testing if they use more of the currency they can get the result on how to solve the problems that occurred in the test.

2.1.4 Environmental requirements

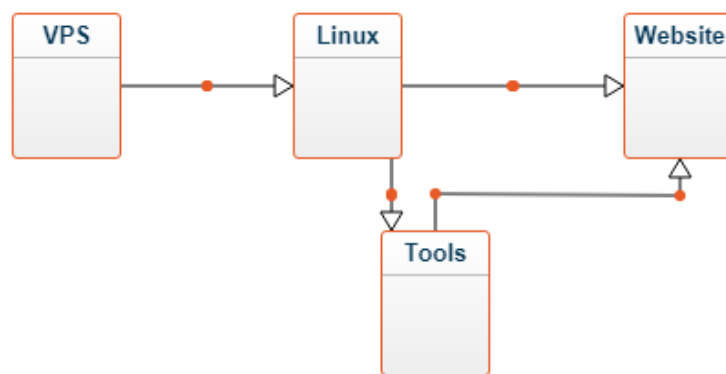
The site needs to be portable and needs to have good scalability, the website needs to work on unlimited number of devices at the same time being able to handle many requests without getting into an error on falling. At the same time, it needs to be well scaled, the site needs to work on any type of device with any size of screen or software that's being used, iOS, android, windows, Linux or whatever the people will use.

2.1.5 Usability requirements

The application should have a good performance, no lags, lost packets ext. It needs to give the users real time feedback about the test that's being made. Also, it needs to be easy to navigate around the site to use it effectively, the site needs to respond fast and clear to commands that users will input in the text fields.

2.2 Design and Architecture

Basically, the system will consist of VPS – virtual private server, a Linux running on it that will be containing the tools and site running, then the tools will be integrated with the website and running on the site so that people can use it.

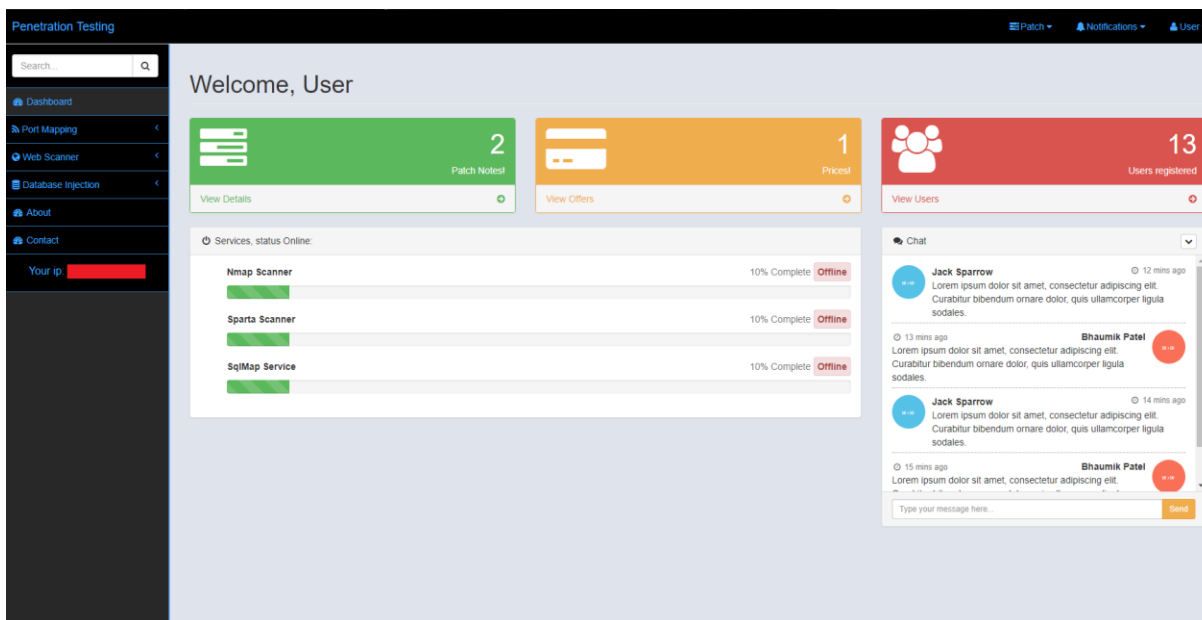


Design will be simple, tools on the left side menu and logout, credits and status of the tools on top right corner of the site.

2.3 Implementation

The design and functionality of the website is quite simple until it comes to tools. Design will be having simple 2 menus, left side of the screen and top menu. Both menus will have different functions. In relation to go on site the user will have to register into the site and then login where he will be moved into the pen testing site but if he won't be using the site for some time, his session will expire, and he will be logged out of the site. The registration system will contain userId, userName, userEmail and userPass.

2.4 Graphical User Interface (GUI) Layout



The design will be simple, still to be changed, display of the scanners status (offline or online), navigation menu, IP address ext. Your IP address is displaying the real time, IP address that the user has when he is on the site, during the login the IP address will go into the database. (I have hidden my IP address, security reasons.)

2.5 Testing

The site will be used for penetration testing, in the same way it can be tested, I obtained Kali Linux, software used for penetration testing. I will test my own website in many ways, from port mapping, brute-force, ftp exploiting, SQL injections, http exploit ext.

3 Conclusions

This project has good advantage, it will be a service needed for huge amount of time, security is getting bigger and bigger, people need to understand on how to keep them and people that access their software always safe. The need for security will always be there so software like this will be helpful always. The disadvantage is that the only thing that can change over time is just the amount or type of tools that will be developed and added to the site over time but other than that this site has potential to be big in penetration testing world.

4 Further development or research

During the development I thought about the feature of this project, I thought that it can get quite big, if everything works out it can be just like a business to make money and help people. People need to test their websites, if the site I do will be cheap enough and good then users will be willing to use it and pay for the tools that are on the site, also due to the contact us page the users will be able to contact the administrator (me) in relation to any problems with the payments or whatever problem they will face on the site. To expand the site, I will need much more time and a good machine, machine that will be able stable, cheap and with good parameters to function correctly.

5 References

- 5.1.1.1 GUI, S. (2015). SQLMap Web GUI. *[online] Cyberpunk. Available at: <https://n0where.net/sqlmap-web-gui/> [Accessed 22 Nov. 2017].*

6 Implementation & Design

6.1 Database Connection

Connecting to the database needs to be secure to keep attackers away from our database, tables and information inside them. To secure the database, I changed the MySQL statements into the PDO version of the PHP that is more secured. This is how my DB connection looks like in PDO:

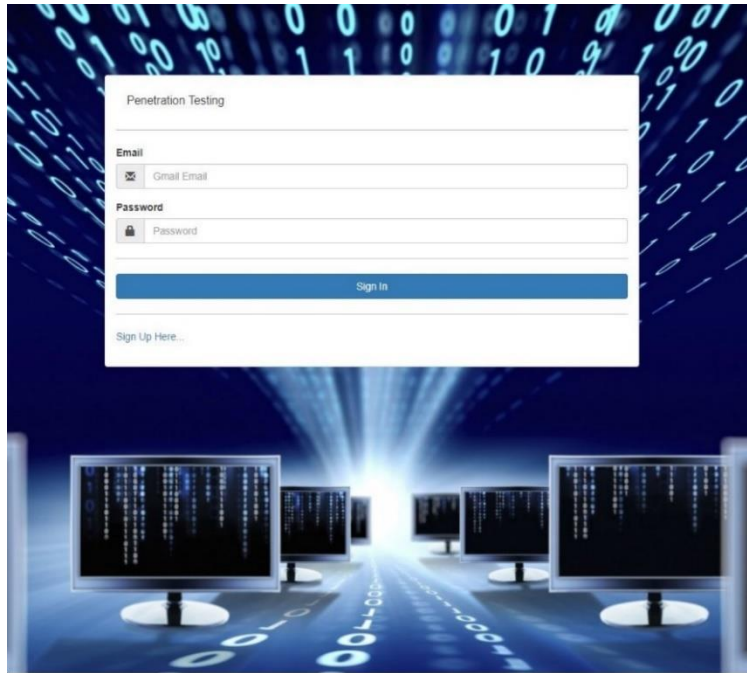
```
<?php
    $host = "localhost";
    $user = "root";
    $pass = "";
    $db = "pentesting";
    $charset = 'utf8mb4';

    $dsn = "mysql:host=$host;dbname=$db;charset=$charset";
    try {
        $opt = [ PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION, PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC, PDO::ATTR_EMULATE_PREPARES => false, ];
        $conn = new PDO($dsn, $user, $pass, $opt);
    }
    catch(PDOException $e) {
        echo "Error: " . $e->getMessage();
    }
?>
```

6.2 Registration & Login

The design of the website is quite simple and clear to users so that they can understand it easily. First, user must face the login or registration relating to what they need to do first on the site. When the user tries to login they have to enter their Gmail email address and their password to get logged in.

During the registration user must fill out his Name, a Gmail email address, repeat the email, then



type in twice password that must be at least 8 characters in length, one upper case letter, one number and one lower case letter. At the end the user must retype the captcha but only the 3 black letters/digits depending what will be displayed since the captcha changes on each refresh of the page or new entrance.

After correct registration and login the user is being redirected to the Dashboard from which he can move forward.

Registration and Login are coded in PHP, all code is created in secure PDO to prevent SQL Injections or any type of take over. Each request to and from database is being protect by prepared statements and most of the time with using bindParam. This is how the functional login page code looks like:

```
// If there's no error this code allows the user to log in:
if (!$error) {

    $password = hash('sha256', $pass); // Password hashing using SHA256 it is encrypting of the password
    $res = $conn->prepare('SELECT userId, userName, userPass FROM users WHERE userEmail = ? LIMIT 1');
    $res->execute([$email]);
    $row = $res->fetch(PDO::FETCH_ASSOC);
    $count = $res->rowCount();

    // EMAIL VALIDATION
    if(!preg_match('/^[a-zA-Z0-9.]+@gmail\.com$/i', $email)){ // forcing exact email
        // Return Error - Invalid Email
        $error = true;
        $emailError = 'The email you have entered is invalid, please try again.';
    }
    else{
        // check email exist or not
        $res = $conn->prepare("SELECT userEmail FROM users WHERE userEmail = ? ");
        $res -> execute([$email]);
        $row = $res->fetch(PDO::FETCH_ASSOC);
        $count = $res->rowCount();

        if($count != 0){
            $error = true;
            $emailError = "Provided Email is already in use.";
        }
    }
}
```

For the registration functionality there are 2 times that the PDO had to be implemented, first was the checking if the email is already taken by the user. Using the prepared statement, I have selected the “userEmail from users WHERE userEmail =?” and then executed the result of email. This is how the code looks like:

The second part of registration was the insertion into the database of records that the user has input into the input fields on the page. For that I have used PDO with bindParam to make it more secured. This is how the insertion code looks like:

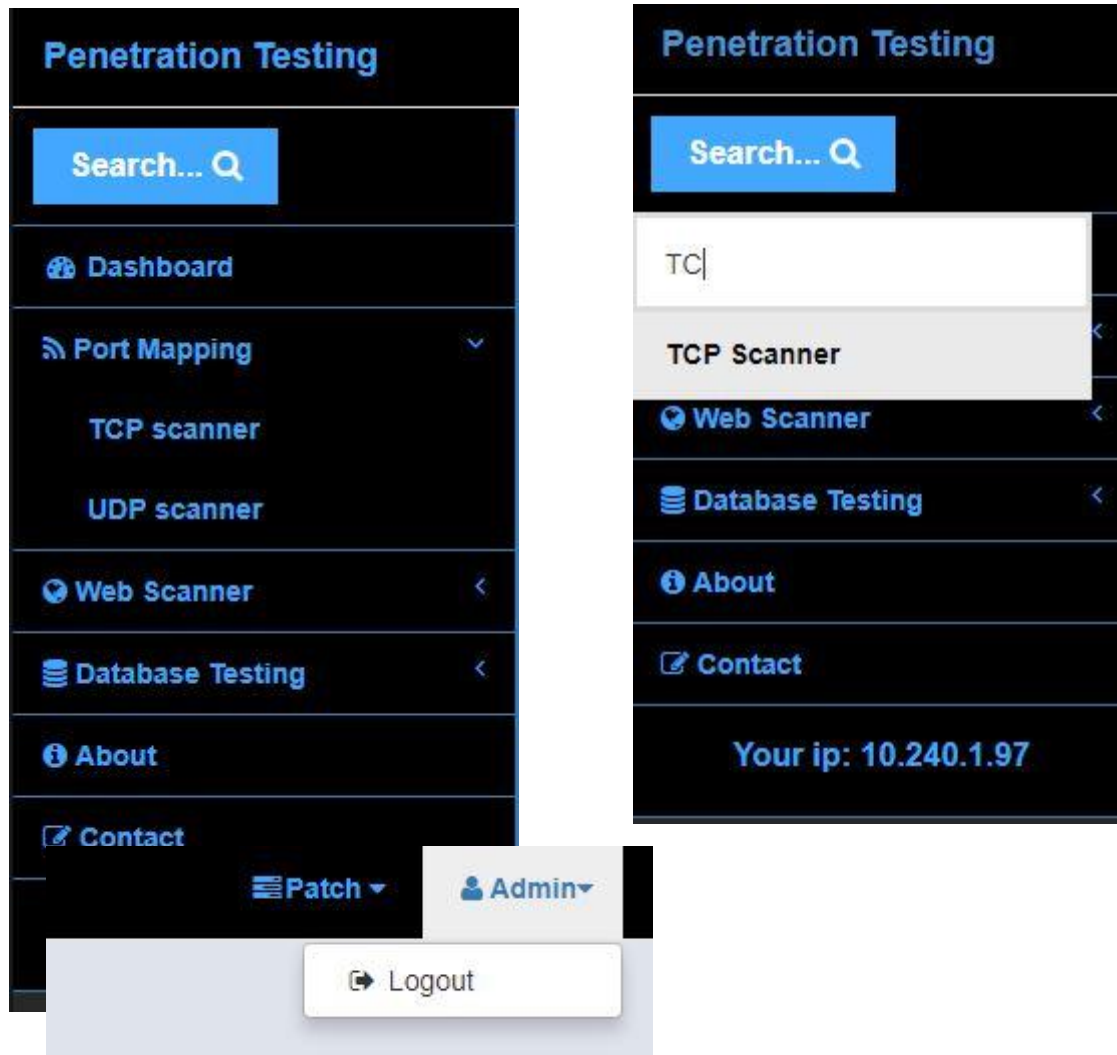
```
// if there's no error, continue to signup
if( !$error ){

    $stmt = $conn->prepare("INSERT INTO users (userName, userEmail, userPass) VALUES (:userName, :userEmail, :userPass)");
    $stmt->bindParam(':userName', $name);
    $stmt->bindParam(':userEmail', $email);
    $stmt->bindParam(':userPass', $password);
    $stmt->execute();

    if ($res) {
        $errTyp = "success";
        $errMSG = "Successfully registered, you may login now";
        unset($name);
        unset($email);
        unset($email2);
        unset($pass);
        unset($pass2);
        unset($captcha);
    }
    else{
        $errTyp = "danger";
        $errMSG = "Something went wrong, try again later...";
    }
}
```

6.3 Navigation Menu

I have implemented the top and left side navigation menu, top navigation menu is used for users to see if the services are online, if the maintains of website is completed and a logout option which is killing the session that is live. This is how the navigation looks like:



All navigation is implemented in one file and then it is requested into each page using `php require_once("");` Header includes the session that lasts 10*60 which is 10minutes, after 10 minutes inactivity the user's needs to login again. This is how the session management looks like with PDO:

```
//start of session time out
if( $_SESSION['last_activity'] < time()-$_SESSION['expire_time'] ) { //have we expired?
    //redirect to logout
    header('Location: ../../pages/logout.php?logout');
} else{
    $_SESSION['last_activity'] = time(); //this was the moment of last activity.
}
$_SESSION['logged_in'] = true; //set you've logged in
$_SESSION['last_activity'] = time(); //your last activity was now, having logged in.
$_SESSION['expire_time'] = 10*60; //expire time in seconds
//end of session time out

// if session is not set this will redirect to login page
if( !isset($_SESSION['user']) ) {
    header("Location: ../../Login");
    exit;
}

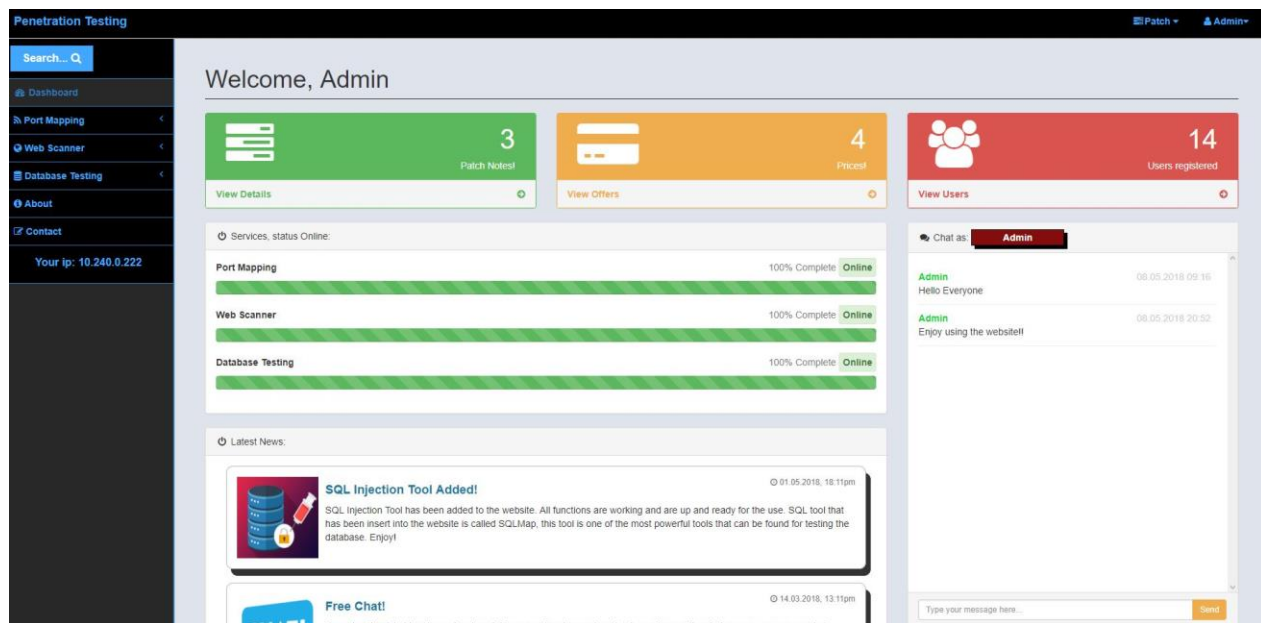
$session_user = $_SESSION['user'];
$query_session = $conn->prepare('SELECT * FROM users WHERE userId = ?');
if ($query_session->execute([$session_user])) {
    $row_session = $query_session->fetch();
}

$res = $conn->prepare('SELECT * FROM users WHERE userId = ?');
if ($res->execute([$SESSION['user']])) {
    $userRow = $res->fetch();
}

// Function to get the client ip address
$IP = $_SERVER['REMOTE_ADDR'];
}
catch(PDOException $e) {
    echo "Error: " . $e->getMessage();
}
}
```

6.4 Dashboard

For dashboard I have tried to make the look as simple as it can be to make it easy for any user to use the site. I have implemented information that show if the vulnerability scanners are working, a simple chat for users to communicate with each other live and ask for help if needed. Dashboard contains also few latest news contents at the bottom as also navigation menu and 3 special things, number of users that are currently registered as also list of latest 15 registered users, subscription offers as also the patch notes that consists of information about the changes that were made to the site in latest maintenance. This is how the dashboard looks like:



6.5 Chat

Chat has been implemented in most of the pages on the website to allow the users to contact other people live at any moment wherever they are on the site. The chat is using PDO just like other parts of the site since it is requesting message + names from the database of the users that post a new message. Also, the chat blocks itself for 30s each time someone posts the message so that they can't just spam random number of

```
try {
    $dbh = new PDO("mysql:host=$hostname;dbname=$dbname", $username, $password);

    if($_POST['name']) {
        $name = $_POST['name'];
        $message = $_POST['message'];

        /** set all errors to exceptions */
        $dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

        $sql = "INSERT INTO chat (date_time, name, message)
                VALUES (NOW(), :name, :message)";
        /** prepare the statement */
        $stmt = $dbh->prepare($sql);

        /** bind the params */
        $stmt->bindParam(':name', $name, PDO::PARAM_STR);
        $stmt->bindParam(':message', $message, PDO::PARAM_STR);

        /** run the sql statement */
        if ($stmt->execute()) {
            populate_shoutbox();
        }
    }
} catch(PDOException $e) {
    echo $e->getMessage();
}

if($_POST['refresh']) {
    populate_shoutbox();
}





function populate_shoutbox() {
    global $dbh;
    $sql = "select * from chat order by date_time desc limit 15";
    echo '<design>';
    foreach ($dbh->query($sql) as $row) {
        echo '<li>';
        echo '<span class="date">'.date("d.m.Y H:i", strtotime($row['date_time'])).'</span>';
        echo '<span class="name">'. $row['name']. '</span><br>';
        echo '<span class="message">'. $row['message']. '</span>';
        echo '</li>';
    }
    echo '</design>';
}
?>
```


messages. The input is validated, and empty messages are not taken by the chat. This is how the code for chat looks like:

6.6 Offers & Subscription

Offers & Subscription page contains a simple and clear layout so that people can understand what is going as also a credit card payment menu on the right side of the website. The menu validates the credit card number input to check if its real and as also displays the type of card if its Visa, Mastercard... ext. The menu checks if the date is

correct and if the date is being the current time then it displays an error, on success the payment menu looks like this:

Payment Details

CARD NUMBER unionpay

EXPIRATION DATE

CV CODE

CHOOSE YOUR SUBSCRIPTION

Start Subscription

Validation passed but at the moment we don't charge for service.

The code responsible for the chat to work is a JavaScript:

```
<script>
jQuery(function($) {
  $('[data-numeric]').payment('restrictNumeric');
  $('.cc-number').payment('formatCardNumber');
  $('.cc-exp').payment('formatCardExpiry');
  $('.cc-cvc').payment('formatCardCVC');
  $.fn.toggleInputError = function(errred) {
    this.parent('.form-group').toggleClass('has-error', errred);
    return this;
  };
  $('form').submit(function(e) {
    e.preventDefault();
    var cardType = $.payment.cardType($('.cc-number').val());
    $('.cc-number').toggleInputError(!$.payment.validateCardNumber($('.cc-number').val()));
    $('.cc-exp').toggleInputError(!$.payment.validateCardExpiry($('.cc-exp').payment('cardExpiryVal')));
    $('.cc-cvc').toggleInputError(!$.payment.validateCardCVC($('.cc-cvc').val(), cardType));
    $('.cc-brand').text(cardType);
    $('.validation').removeClass('text-danger text-success');
    $('.validation').addClass($('.has-error').length ? 'text-danger' : 'text-success');
  });
});
</script>
```

6.7 About & Contact pages

Both About & Contact pages are simple in design and very clear, the main functions of about page are to describe how the website is constructed, in what languages, by who and a short description. This is how about page looks like:



For the contact us page, the main function is presenting where the company base in as also a contact form so that users can send email if they have any problems.

Search... Q

Dashboard

Port Mapping

Web Scanner

Database Testing

About

Contact

Your ip: 10.240.0.239

Get in Touch with us.

Please contact us if you would like to know anything about the services or if you have any questions.

Details:

Name: Admin

Email Address: admin@gmail.com

Phone Number:

Message:

Submit

National College of Ireland

IFSC, Mayor Street, North Dock, Dublin 1, D01 Y300

4.3 ★★★★★ 62 reviews

View larger map

Directions

Save

Docklands

Park Rite IFSC Car Park

Jeanie Johnston Tall Ship

N Wall Quay

Samuel Beckett Bridge

River Liffey

The Convention Dublin (The CCL)

Map data ©2018 Google

Terms of Use

Report a map error

Copyright © PenTesting by Trojan 2017

6.8 Scanners

There are 4 scanners on the website, they are as follows: TCP Scanner, UDP Scanner, WHOis Scanner as also SQL Injection scanner. TCP and UDP scanners are used to find open ports on most used ports by people as also the most important ports. WHOis scanner is used to find the information about a target, servers hosted, contact ext. SQL injection is used to find vulnerabilities that can allow the attacker to attack to take advantage of the software and retrieve sensitive information from the database.

Scanners look like keep the layout the same.

Here is how UDP Scanner looks like:

The screenshot shows the UDP Scanner web interface. At the top, there are two numbered instructions: 3. Restrict the access to ports like 3306 that is responsible for database. 4. Never have a free open port that is no needed since the attacker can take advantage of that and use it. Below these is a 'Tool:' section with a blue header that reads 'UDP SCANNER' and a subtitle 'AUTOMATIC UDP SCANNER, WILL FIND OPEN AND CLOSED PORTS FOR YOU!'. A welcome message follows: 'Welcome to the UDP Port Scanner! Use the allocated space to input the IP address or the URL of the target. Then simply click on the button at the bottom when done to launch a new scan!'. A red warning note states: 'If you input incorrect URL, or IP address all ports might come up as Closed, make sure that you type everything correctly'. There is a text input field for 'Domain or IP Address:' containing 'jiri: website.com/100.164.231.0'. Below this is a large blue button labeled 'Run UDP port scan'. At the bottom, there is a checkbox for 'I am authorized to scan this target and I accept the Terms and Conditions' and a copyright notice: 'Copyright © PenTesting by Trojan 2017'.

SQL injection scanner is the only one that is more advanced and requires more tabs since the testing must be adjusted more to test what the user exactly wants. The main page look of SQL injection scanner is:

The screenshot shows the SQLMAP Web GUI interface. It features a blue header with 'SQLMAP' and the subtitle 'AUTOMATIC SQL INJECTION TESTER, WILL TEST AND PRESENT ATTACK POINTS TO YOU!'. A welcome message reads: 'Welcome to the SQLMAP Web GUI! Use the tabs below to configure your scan settings. Then simply click on the button at the bottom when done to launch a new scan!'. A red hint note says: 'A small hint! Navigate to Enumeration tab and chose what you want to be looked for, example: All available Databases & All Database Users'. Below this is a tabbed interface with tabs for 'Basic', 'Request', 'Technique', 'Detection', 'Enumeration', and 'Access'. The 'Basic' tab is active. It contains a 'Target URL:' field with 'http://site.com/vuln.php?id=1', an 'HTTP Method:' dropdown set to 'Default (GET)', and a 'Flush Any Existing Session Info:' dropdown set to 'No'. There are four radio button options: 'Marking Injection', 'Known Vulnerable Parameter', 'Unknown, Fuzz All Parameters!' (which is selected), and 'Unknown, Fuzz Forms on Page'. Below these is an 'Optional Parameter Name(s) to Skip:' field with the example 'i.e. paramName,to,skip'. A large blue button labeled 'Run SQLMAP Web Scan' is at the bottom. At the very bottom, there is a checkbox for 'I am authorized to scan this target and I accept the Terms and Conditions' and a copyright notice: 'Copyright © PenTesting by Trojan 2017'.

Final Project Report

If the user navigates to different tabs he will get different options for the tools like if he goes into Enumeration he will be able to choose what the tool should be looking for, the interface looks like this:

The screenshot displays the SQLMAP Web GUI interface. At the top, a welcome message reads: "Welcome to the SQLMAP Web GUI! Use the tabs below to configure your scan settings. Then simply click on the button at the bottom when done to launch a new scan! A small hint! Navigate to Enumeration tab and chose what you want to be looked for, example: All available Databases & All Database Users". Below this, there are six tabs: Basic, Request, Technique, Detection, Enumeration (selected), and Access. The Enumeration tab contains several configuration sections:

- Database(s) to Dump or Enumerate:** A text input field with the placeholder "i.e. database,names,here".
- Table(s) to Dump or Enumerate:** A text input field with the placeholder "i.e. table,names,here".
- Column(s) to Dump or Enumerate:** A text input field with the placeholder "i.e. juicy,columns,here".
- Column(s) to Exclude or NOT Enumerate:** A text input field with the placeholder "i.e. useless,columns,here".
- Specific Database User to Enumerate:** A text input field with the placeholder "i.e. username".
- Where Condition to Filter Dump Results:** A text input field with the placeholder "i.e. group='admin'".
- Select Enumeration Options to Enable:** A scrollable list of options. The selected options are: "Enumerate ALL the Things!", "Version or Banner Info", "Extensive DBMS Fingerprint", "Database Server Hostname", "Current Active Database", "All Available Databases", "Current Database User", "All Database Users", and "Dump Database & Table Schema".
- Row Start:** A dropdown menu with "Dis:" selected.
- Row Stop:** A dropdown menu with "Dis:" selected.
- SQL Statement to Execute:** A text input field with the placeholder "i.e. SELECT version()".

At the bottom, there is a large blue button labeled "Run SQLMAP Web Scan". Below the button, there is a checkbox labeled "I am authorized to scan this target and i accept the [Terms and Conditions](#)".

7 Vulnerability Testing

7.1 Scans performed:

Scanning was performed using tools: SQLMAP, NMAP, SPARTA, OWASP-ZED, COMMIX, BURPSUITE, NIKTO, COMMIX, RIPS php tester.

7.2 Using Sparta / Nikto

During the scans performed using tools Sparta and Nikto I was able to find few vulnerabilities that were putting my site to danger, one of them was not declaring the HTTP only flag and having insecure session management which I had to format and secure it. Also using does tools I was able to display all the folders and subfolders of the website, example: pages/Tools/sqlmap.php. In order to secure all does outputs I had to use htaccess to secure the access to the site. My htaccess looks like this:

```
RewriteEngine On

RewriteRule ^TCP-Port-Scanning/?$ pages/Tools/tcpscan.php [L,NC]
RewriteRule ^Whois-Scanner/?$ pages/Tools/whoistest.php [L,NC]
RewriteRule ^SQL-Injection/?$ pages/Tools/sqlmap.php [L,NC]
RewriteRule ^UDP-Port-Scanning/?$ pages/Tools/udpscan.php [L,NC]
RewriteRule ^SQL-Injection-Scan/?$ pages/Tools/sqlmap/scans.php [L,NC]

RewriteRule ^About-Us/?$ pages/about.php [L,NC]
RewriteRule ^Contact-Us/?$ pages/contact.php [L,NC]
RewriteRule ^Dashboard/?$ pages/mainpage.php [L,NC]
RewriteRule ^Patch-News/?$ pages/patchdetails.php [L,NC]
RewriteRule ^List-Of-Users/?$ pages/viewusers.php [L,NC]
RewriteRule ^Logout-from-the-site/?$ pages/logout.php [L,NC]
RewriteRule ^View-&-Buy/?$ pages/Offers.php [L,NC]

RewriteRule ^Login/?$ index.php [L,NC]
RewriteRule ^Register/?$ registration.php [L,NC]

php_value session.cookie_httponly 1
php_value session.cookie_secure 1
Options -Indexes

RedirectMatch 403 ^/register/?$
RedirectMatch 403 ^/phpmyadmin/?$
```


Using this htaccess I was able to restrict the access to .php files, set the session cookies to secure, set the HTTP only flag as also block the access to any subdirectories.

The result before implementing of htaccess:

```
-----
+ Target IP:      146.148.16.117
+ Target Hostname: software-project-dev-nastoprocent.c9users.io
+ Target Port:    80
+ Start Time:     2018-05-12 15:46:00 (GMT1)
-----
+ Server: Apache/2.4.7 (Ubuntu)
+ Retrieved x-powered-by header: PHP/5.5.9-1ubuntu4.22
+ 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
+ Uncommon header 'x-backend' found, with contents: apps-proxy
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ Cookie XDEBUG_SESSION created without the httponly flag
+ Cookie PHPSESSID created without the httponly flag
+ Root page / redirects to: Logout-from-the-site
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Apache/2.4.7 appears to be outdated (current is at least Apache/2.4.12). Apache 2.0.65 (final release) and 2.2.29 are also current.
+ OSVDB-3268: /img/: Directory indexing found.
+ OSVDB-3092: /img/: This might be interesting...
+ OSVDB-3268: /pages/: Directory indexing found.
+ OSVDB-3092: /pages/: This might be interesting...
+ Uncommon header 'x-ob_mode' found, with contents: 0
+ OSVDB-3092: /register/: This might be interesting...
```

Result after implementing htaccess:

```
root@kali:~# nikto -port 80 -host software-project-dev-nastoprocent.c9users.io
- Nikto v2.1.6
-----
+ Target IP:      146.148.16.117
+ Target Hostname: software-project-dev-nastoprocent.c9users.io
+ Target Port:    80
+ Start Time:     2018-05-12 16:16:02 (GMT1)
-----
+ Server: Apache/2.4.7 (Ubuntu)
+ Retrieved x-powered-by header: PHP/5.5.9-1ubuntu4.22
+ 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
+ Uncommon header 'x-backend' found, with contents: apps-proxy
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ Cookie XDEBUG_SESSION created without the httponly flag
+ Root page / redirects to: Logout-from-the-site
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Apache/2.4.7 appears to be outdated (current is at least Apache/2.4.12). Apache 2.0.65 (final release) and 2.2.29 are also current.
+ Uncommon header 'x-ob_mode' found, with contents: 0
+ Server leaks inodes via ETags, header found with file /icons/README, fields: 0x13f4 0x438c034968a80
+ OSVDB-3233: /icons/README: Apache default file found.
+ /phpmyadmin/: phpMyAdmin directory found
+ 7653 requests: 0 error(s) and 11 item(s) reported on remote host
+ End Time:       2018-05-12 16:28:26 (GMT1) (744 seconds)
-----
+ 1 host(s) tested
```

From all of the results after implementing htaccess, non are real threat, phpMyAdmin directory was found but user name and password are set as also they are not default password and user name. Root page redirected to the logout-from-the-site because there

is no session for the scanner which means that he is not logged in. Cookie xdebug_session cannot be change since the c9 doesn't allow access to files like .ini.

7.3 Nmap / Sparta

Using Nmap and Sparta I wasn't able to find anything that could potentially harm the website during an attack. None of my scans brought any harmful result, all not needed ports were closed, the scripts and tools that I have implemented do not open any unnecessary ports that should be closed.

```
Starting Nmap 7.40 ( https://nmap.org ) at 2018-05-12 14:59 IST
Nmap scan report for software-project-dev-nastoprocent.c9users.cba
(0)
Host is up (0.079s latency).
rDNS record for 35.195.221.140: 140.221.195.35.bc.googleusercontent.com
Not shown: 993 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https
1234/tcp  open  hotline
8000/tcp  open  http-alt
8080/tcp  open  http-proxy
8081/tcp  open  blackice-icecap
8082/tcp  open  blackice-alerts

Nmap done: 1 IP address (1 host up) scanned in 15.37 seconds
root@Kali:~#
```

All ports that are open are basically responsible for website like 80/443/8000/8080 which are for users in order to run the website and be able to open it. 1234/8081/8082 ports are mandatory from c9 which doesn't allow me to close them up.

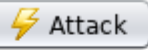

7.4 OWASP-ZAP / Burpsuite

Using owasp-zap and Burpsuite I wasn't able to find any vulnerabilities, Burpsuite couldn't upload and delete any files from the server since none of the http methods were allowed

as also OWASP-ZAP was unable to even get a hold on the site, trying to scan for http and https gave no results:

to quickly test an application, enter its URL below and press 'Attack'.

URL to attack:

Progress: Failed to attack the URL: Maximum redirects (100) exceeded

7.5 Injection

Using SQLmap and Commix I have scanned the website for any injection points but before of that I secured all site by using PDO secure PHP coding as also I had validated each input so that it returns error if any of unwanted characters are inputted into the fields. Any field either if it is search bar, chat or login, each input is validated fully so that no injection using SQL or XSS could be performed. Using the tools I have tested and got no vulnerable points, output from the SQL test:

```
[*] starting at 15:02:47
[15:02:48] [WARNING] you've provided target URL without any GET parameters (e.g. 'http://www.site.com/article.php?id=1') and without providing any POST parameters through option '--data'
do you want to try URI injections in the target URL itself? [Y/n/q] y
[15:02:50] [INFO] testing connection to the target URL
sqlmap got a 302 redirect to 'http://software-project-dev-nastoprocent.c9users.io:80/Logout-from-the-site'. Do you want to follow? [Y/n] n
[15:02:52] [INFO] checking if the target is protected by some kind of WAF/IPS/IDS
[15:02:53] [INFO] testing if the target URL is stable
[15:02:54] [WARNING] URI parameter '#1*' does not appear to be dynamic
[15:02:54] [WARNING] heuristic (basic) test shows that URI parameter '#1*' might not be injectable
[15:02:54] [INFO] testing for SQL injection on URI parameter '#1*'
[15:02:54] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[15:02:55] [WARNING] reflective value(s) found and filtering out
[15:02:56] [INFO] testing 'MySQL >= 5.0 boolean-based blind - Parameter replace'
[15:02:56] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[15:02:57] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[15:02:58] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[15:03:00] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[15:03:00] [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace (FLOOR)'
[15:03:00] [INFO] testing 'MySQL inline queries'
[15:03:01] [INFO] testing 'PostgreSQL inline queries'
[15:03:01] [INFO] testing 'Microsoft SQL Server/Sybase inline queries'
[15:03:01] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[15:03:01] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[15:03:02] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[15:03:03] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind'
[15:03:04] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[15:03:05] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[15:03:06] [INFO] testing 'Oracle AND time-based blind'
[15:03:06] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[15:03:06] [WARNING] using unescaped version of the test because of zero knowledge of the back-end DBMS. You can try to explicitly set it with option '--dbms'
[15:03:13] [WARNING] URI parameter '#1*' does not seem to be injectable
[15:03:13] [CRITICAL] all tested parameters appear to be not injectable. Try to increase '--level'/'--risk' values to perform more tests. Also, you can try to rerun by providing either a valid value for option '--string' (or '--regexp'). If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could retry with an option '--tamper' (e.g. '--tamper=space2comment')
[15:03:13] [WARNING] HTTP error codes detected during run:
403 (Forbidden) - 5 times, 404 (Not Found) - 144 times
[*] shutting down at 15:03:13
root@Kali:~#
```


7.6 RIPS

Using rips I checked every php with function file to search for any type of vulnerabilities and each test on pages like login, registration, chat, db. Connection. Using the test on my old registration page I had a lot of misconfigurations which brought me 2 SQL injections and 2 XSS injections that work on my site just in registration. My old registration php code was based on mysql_query which is the insecure way. After scanning this code I got this output:

File: /home/ubuntu/workspace/oldregistration.php

```

SQL Injection
Userinput reaches sensitive sink. For more information, press the help icon on the left side.
65: mysql_query $res = mysql_query($query);
64: $query = "SELECT userEmail FROM users WHERE userEmail='$email'";
50: $email = $_POST['userEmail'];

requires:
17: if(isset($_POST['btn-signup']))
62: if(!filter_var($email, FILTER_VALIDATE_EMAIL)) else

SQL Injection
Userinput reaches sensitive sink. For more information, press the help icon on the left side.
100: mysql_query $res = mysql_query($query);
90: $query = "INSERT INTO users(userName,userPass,userEmail,userDate,userLevel) VALUES('$name', '$pass', '$email', NOW(), 0)";
22: $name = htmlspecialchars($name);
21: $name = strip_tags($name);
20: $name = trim($_POST['userName']);
30: $pass = htmlspecialchars($pass);
29: $pass = strip_tags($pass);
28: $pass = trim($_POST['userPass']);
50: $email = $_POST['userEmail'];

requires:
17: if(isset($_POST['btn-signup']))
88: if(!$error)

Cross-Site Scripting
Userinput reaches sensitive sink. For more information, press the help icon on the left side.
168: echo echo $name;
22: $name = htmlspecialchars($name); // if(isset($_POST)),
21: $name = strip_tags($name); // if(isset($_POST)),
20: $name = trim($_POST['userName']); // if(isset($_POST)),
20: $name = trim($_POST['userName']); // if(isset($_POST)),
21: $name = strip_tags($name); // if(isset($_POST)),
20: $name = trim($_POST['userName']); // if(isset($_POST)),

Cross-Site Scripting
Userinput reaches sensitive sink. For more information, press the help icon on the left side.
176: echo echo $email;
50: $email = $_POST['userEmail']; // if(isset($_POST)),

```

The php scanner found that I have many misconfigurations and I need to change my mysql_query into MySQLi or even better to PDO. After changing all of my registration into PDO version I have received 0 misconfigurations and injection points:

path / file: ☒ subdirs
verbosity level: 1. user tainted only vuln type: All scan
code style: phps bottom-up /regex: search

Nothing vulnerable found. Change the verbosity level or vulnerability type and try again.

Result
No vulnerabilities found.
Scanned files: 1
Include success: 2/2 (100%)
Considered sinks: 298
User-defined functions: 0
Unique sources: 8
Sensitive sinks: 14
Info: uses sessions
Get the next generation of **RIPS** with state-of-the-art code analysis!
Scan time: 0.013 seconds

Because of changing registration into PDO I had to change the rest also. I have scanned each file that contained some connection with DB or was taking input from user, each file was scanned one by one. The results of each scan are as follows:

DB configuration file:

path / file: ☒ subdirs
verbosity level: 1. user tainted only vuln type: All scan
code style: phps bottom-up /regex: search

Nothing vulnerable found. Change the verbosity level or vulnerability type and try again.

Result
No vulnerabilities found.
Scanned files: 1
Include success: No includes.
Considered sinks: 298
User-defined functions: 0
Unique sources: 0
Sensitive sinks: 1
Get the next generation of **RIPS** with state-of-the-art code analysis!
Scan time: 0.002 seconds

Login file:

path / file: ☒ subdirs
verbosity level: vuln type:
code style: /regex:

Nothing vulnerable found. Change the verbosity level or vulnerability type and try again.

Result

No vulnerabilities found.

Scanned files:	1
Include success:	1/1 (100%)
Considered sinks:	298
User-defined functions:	0
Unique sources:	3
Sensitive sinks:	7

Info: uses sessions

Get the next generation of **RIPS** with state-of-the-art code analysis!

Scan time: 0.005 seconds

Header file:

path / file: ☒ subdirs
verbosity level: vuln type:
code style: /regex:

Nothing vulnerable found. Change the verbosity level or vulnerability type and try again.

Result

No vulnerabilities found.

Scanned files:	1
Include success:	No includes.
Considered sinks:	298
User-defined functions:	0
Unique sources:	1
Sensitive sinks:	7

Info: uses sessions

Get the next generation of **RIPS** with state-of-the-art code analysis!

Scan time: 0.004 seconds

Chat file:

path / file: ☒ subdirs
verbosity level: vuln type:
code style: /regex:

Nothing vulnerable found. Change the verbosity level or vulnerability type and try again.

Result
No vulnerabilities found.
Scanned files: 1
Include success: No includes.
Considered sinks: 298
User-defined functions: 1
Unique sources: 3
Sensitive sinks: 8
Get the next generation of **RIPS**
with state-of-the-art code analysis!
Scan time: 0.004 seconds


Inserting into DB the scan records:

path / file: ☒ subdirs
verbosity level: vuln type:
code style: /regex:

Nothing vulnerable found. Change the verbosity level or vulnerability type and try again.

Result
No vulnerabilities found.
Scanned files: 1
Include success: No includes.
Considered sinks: 298
User-defined functions: 0
Unique sources: 4
Sensitive sinks: 5
Info: uses sessions
Get the next generation of **RIPS**
with state-of-the-art code analysis!
Scan time: 0.006 seconds

Main page file:



path / file: ☒ subdirs

verbosity level: vuln type:

code style: /regex/:

Nothing vulnerable found. Change the verbosity level or vulnerability type and try again.

Result

No vulnerabilities found.

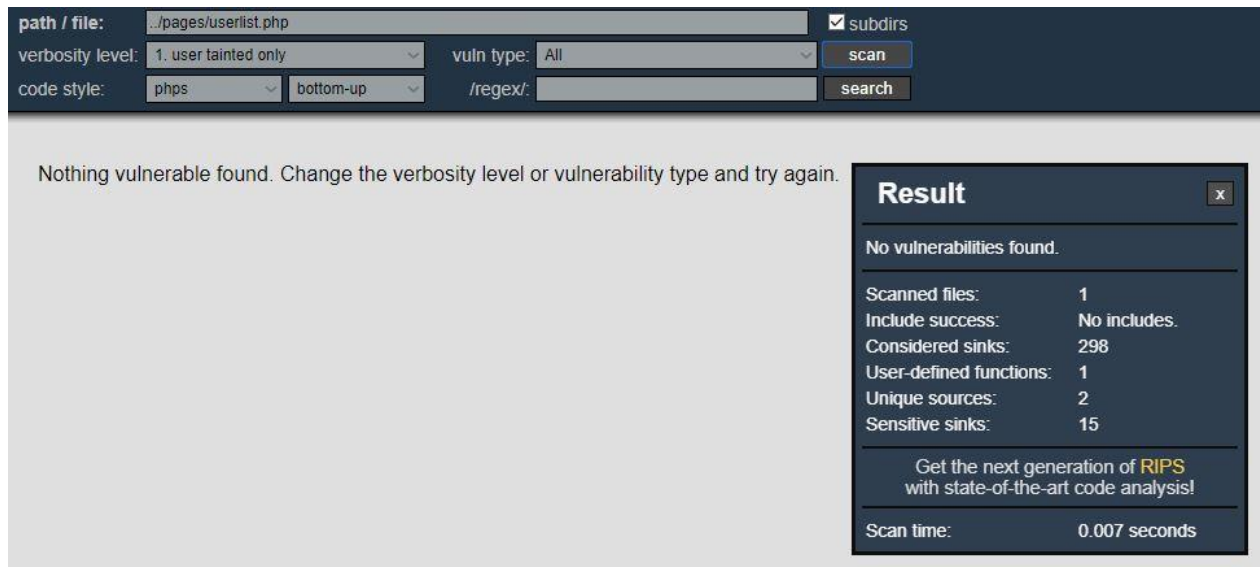
Scanned files:	1
Include success:	2/2 (100%)
Considered sinks:	298
User-defined functions:	0
Unique sources:	1
Sensitive sinks:	13

Info: uses sessions

Get the next generation of **RIPS** with state-of-the-art code analysis!

Scan time: 0.008 seconds

User list:



path / file: ☒ subdirs

verbosity level: vuln type:

code style: /regex/:

Nothing vulnerable found. Change the verbosity level or vulnerability type and try again.

Result

No vulnerabilities found.

Scanned files:	1
Include success:	No includes.
Considered sinks:	298
User-defined functions:	1
Unique sources:	2
Sensitive sinks:	15

Get the next generation of **RIPS** with state-of-the-art code analysis!

Scan time: 0.007 seconds

All files that are responsible for any type of php functionality in the website are secured from any input from an user, no vulnerabilities were found in the files that users could use. The tester is only complaining on code sinks for doing things like:

Include_once ("header"); or require_once ("header");

Sensitive sinks displayed by the scanner are for things that could be done when the user would have an access to my php files and just play around with the code, that he could take advantage of header/DB imports ext. Since the code is secure, website is secure the user wont have any access to my code.

8 Appendix

8.1 *Project Proposal*

This section will contain the project proposal as also the reflective journals.

8.2 Objectives

For my project I am creating a website that is going to allow the users to test their own website using my website. Basically, there will be few penetration-testing tools on the website that will require the user to buy “Credits” or something similar that will allow them to use one of the tests that are on the website. The website will provide testing service for other web users that want to see if their websites are secure, from finding vulnerabilities to DDOS and SQL injections, port mapping ext.

8.3 Background

At the start the user will have to log in into the webpage to be able to use any of the content of the site, he will be able to view it but to use it, he will have to login into the page. After login in the user will be brought back to the main index page where he will be able to navigate around the page, on the left side he will be able to see the types of services that there will be provided from him to use on his website. The website as also the programs that will be used for testing will have to be installed on VPS that will probably be a Linux system.

I will have to research how to install Linux or Kali Linux on VPS setting it up for a service on a website. The user won't have any profile but there will be his name displayed in the right top corner. The user will be allowed to log out. On the website there will be his IP displayed but only to himself, he will be able to see what his IP is by just going into the website.

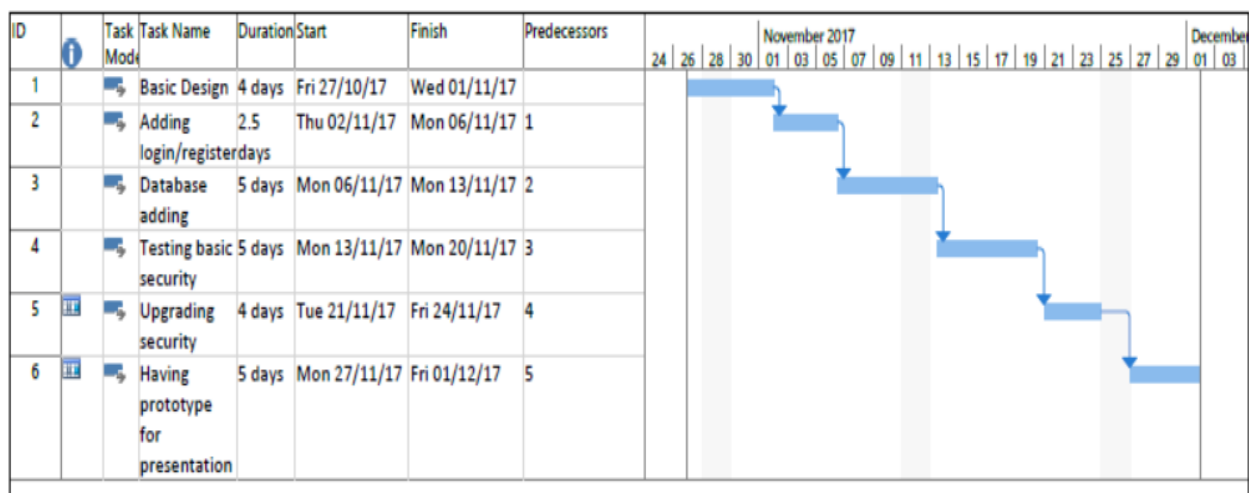
8.4 Technical Approach

For the languages used I will use PHP, HTML, CSS for sure, I will try to add JAVA into the project where I will be able to. I will set a database either on the same repository or on another one to have it separated from the main files of the website. Also, I will need to have Kali Linux or any other type of Linux and then run a website on the system. When I will get that working I will have to get the connection between Linux and website, basically I will have to make the Kali using scripts to respond to websites requests, let's say if user is going to use NMAP command then I will need Kali to turn on terminal and run command "nmap -pn "address of the website" and then display all of this back to the user.

8.5 Special resources required

I will need to get a VPS that will allow me to get Kali + Website running on it with good access to it. Also, I will need some good articles, books or notes on web to get the information's I need for development of my project

8.6 Project Plan



8.7 Technical Details

The languages that I will implement for sure are PHP, HTML, CSS and probably JAVA. I will try to do as much as I can in PHP since my login/registration will be done in it. My SQL database will be protected from any SQL injections or exploitation, I will need to look what coding way will be the most correct for DB when it comes to security because simple encryption or hashing won't be enough to prevent exploit.

8.8 Evaluation

In the project I'm going to create few users but I'm not going to populate hugely the database. The system will contain sensitive data, which will be the details of credit card that I need to secure hugely. When it comes to integration, scalability I will test the website from Desktop PC, Laptop, Tablet, Phone (Android and iOS) which will give me the main view of how the website is prospering to different screen and operating systems. Regarding the security testing I will test my website myself using Kali Linux that I have installed some time ago on my laptop, I'm going to test my database from SQL injections as from exploitation, then I will be testing the website if it can be exploited and taken over in any way. The system needs to be easy to use so that any user can work from it and test their website, also always when someone will try to use the testing tools he will need to tick a box saying that he can do so, that the website belongs to him or the administrator of website allows him to perform various of tests on his website.

8.8.1 Project Plan

Dawid Trojanowski 11/21/2017 – requirements Specification

Introduction

8.8.2 Purpose

The purpose of this document is to set out the requirements for my project. The purpose of my project is to allow the users to test their own websites and see if they are secure, also they will be

allowed to chat with each other or have a blog on which they can paste comments and talk with each other to get help. My project will provide few tools that allow to scan/test/find vulnerabilities of the website they own or can perform the given test. The aim of my project is to get people more aware of security problems, target group would be adults that own/maintain the website and want to have it secure.

8.8.3 Project Scope

The scope of the project is to develop good looking and easy to use website that will allow the user to test their own website. Project should contain at least 3 types of testing, SQL / port mapping / website vulnerabilities scanner. The website will be running on a VPS that will contain Linux / trying to get kali Linux on it. Then I will try to run the website on the VPS, I will connect the Linux with website in the way that the user using the website will send requests to the Linux and tell him to run some tool, then return the data result and close the terminal after completing.

8.9 User Requirements Definition

All clients using any type of website want to be safe, they want their data to be safe and not exploited and used in some or other way. Every user has a right to privacy, so their data should be kept in top security. Security today is hard since it's easier and easier to exploit and hack websites which contain any type of miss configuration or bugs in code. The users will be able to understand how important it is to secure their website, they will get told how to do it, and in relation of the result they will get told what they need to do to secure the website.

8.10 Requirements Specification

All requirements should be verifiable. For example, experienced controllers shall be able to use all the system functions after a total of two hours training. After this training, the average number of errors made by experienced users shall not exceed two per day.

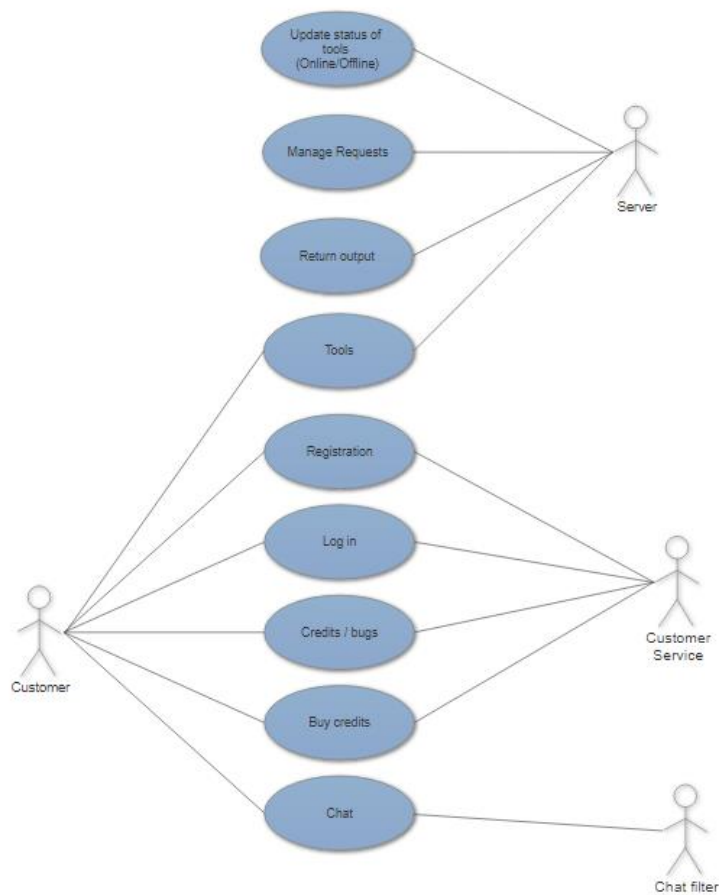
8.11 Functional requirements

The requirements that the project will have are:

- Registration

- Login system
- Buy credits
- Chat
- Report errors
- Use tools on the website (penetration-testing tools)
- Getting solution on how to solve the problem.

8.11.1 Use Case Diagram



8.11.2 Requirement 1: Registration

Description & Priority

Registration is one of the most important things, the user needs to be able to register. When user registers he needs to confirm that any website they test is owned by them or they can do so (Disclaimer). Without registration user won't be able to use anything on the site.

Use Case

Sequence 1

Scope

The scope of this use case is to allow the users to register into the webpage and allow them to use the site.

Description

This use case describes how the user will access the website, basically the user needs to register into the website, after registration the user will need to log in, after logging in the user will be send to the main page where they will be able to access the content/tools of the website.

Flow Description

Precondition

The system is in initialisation mode, wait for the user to register and connected to the site.

Activation

This use case starts when a user presses button registers and registers his correct details into the site where the system encrypts and saves all his data into the database.

Main flow

1. The system identifies the request of registration.
2. The user types in his details in registration (See A1).
3. The system checks if the details don't contain not allowed fragments (See E1).
4. The system sends back message regarding the details.
5. The user has account, needs to login.

Alternate flow

A1: Registration

1. The system allows user to register.
2. The user puts in details.
3. The system checks details.
4. The user is registered.

Exceptional flow

E1: Wrong/Not allowed details

1. The system allows user to type in details.
2. The user enters details.
3. The system checks details.
4. The system returns that details mismatch/are not allowed/something missing.
5. The system resets the form.
6. The user must put his details again.

Termination

After completing of registration, the user will be send back to the login page where he can login to the page.

Post condition

The system goes into a wait state until another registration request comes.

8.11.3 Requirement 2: Login System

Description & Priority

This use case describes how the login system works.

Use Case

Sequence 2

Scope

The scope of this use case is to allow the user to login into the page.

Description

After completing of the registration, the user must be allowed to login to the page with their details, Login system will check if the characters entered are correct to the ones saved in database and if so the system will let the user login into the site and then redirect him into the main page, allowing him to chat with other people and use tools that are available on the site.

Flow Description

Precondition

The system is in initialisation mode waiting for someone to login.

Activation

This use case starts when a user types in their correct details and presses button to login.

Main flow

1. The system identifies the details typed in.
2. The user waits for the system to check the details.
3. The system checked the details and lets the user login (See E1).
4. The system returns "login into the site".
5. The user is moving around the website.

Exceptional flow

E1: Login error

1. The system waits for the user to login.
2. The user types in their details.
3. The system gets the details.
4. The system checks the details.
5. The system returns the details were typed in incorrect.

Termination

The system presents the main page if the login was correct if not it resets the page and waits for user to login again

Post condition

The system goes into a wait state and waits for the user logging in again.

8.11.4 Requirement 3: Credits

Description & Priority

This use case is used to allow the user to buy credits on the webpage and then use the tools that are on the page.

Use Case

Sequence 3

Scope

The scope of this use case is to allow the user to buy credits and use the tools on the page.

Description

This use case describes the user process of buying credits.

Flow Description

Precondition

The system is in initialisation mode.

Activation

This use case starts when a user wants to use a tool and needs to buy credits to do so.

Main flow

1. The system identifies the request of user to get credits
2. The system sends user to a credit page and allows the user to purchase them.
3. The user is typing in his correct details of credit card ext. (A1)
4. The system verifies if the details are correct.
5. The system sends the user correct amount of credits.
6. The user got the credits and can use tools.

Alternate flow

A1: Saved details

1. The system checks the account and verifies if the user has any credit card details saved.
2. The system retrieves 4 digits from the credit card number and asks if the user wants to use it.
3. The user takes the saved details and uses them.
4. The system gives user credits on the account.

Exceptional flow

E1: Wrong details.

1. The system takes details from user.
2. The system checks the details.
3. The system returns error message saying that the details are incorrect.
4. The system sends user back to the main page.
5. The user must go into the credit page again.

Termination

The system adds credit to user if all the details are correct.

Post condition

The system goes into a wait state.

8.11.5 Requirement 4: Chat

Description & Priority

This use case is used to allow the users to communicate with other users on the page.

Use Case

Sequence 4

Scope

The scope of this use case is to allow the users to communicate easily with other users.

Description

This use case describes the user being able to chat with other users.

Flow Description

Precondition

The system is in initialisation mode.

Activation

This use case starts when a user sends a message on the chat.

Main flow

1. The system connects the chat with a database.
2. The system waits for a user to send a message.

3. The user sends a message.
4. The system displays the message in the chat window.

Exceptional flow

E1: Chat filter

1. The system connects the chat with a database.
2. The system waits for a user to send a message.
3. The user sends a message.
4. The system checks the words used.
5. The system displays message saying that there are not allowed words.

Termination

The system displays messages in the chat.

Post condition

The system goes into a wait state until a new message comes.

8.11.6 Requirement 5: Report errors

Description & Priority

This use case is used to allow the users to report any type of errors the face while using the page.

Use Case

Sequence 5

Scope

The scope of this use case is to allow the users to report errors/bugs.

Description

This use case describes the user communication with the customer service.

Flow Description

Precondition

The system is in initialisation mode.

Activation

This use case starts when a user sends a help request to the IT.

Main flow

1. The system waits for a user to create a request.
2. The user creates a help request.
3. The system verifies the message.
4. The system sends the message
5. The system returns send message to the user.

Exceptional flow

E1: Chat filter/email check

1. The system checks the message and email address of the user.
2. The system returns error message saying email incorrect or not allowed content.
3. The system waits for the user to change his message.
4. The user types the message again or edits the message.

Termination

The system sends the message on support email.

Post condition

The system goes into a wait state until a new request comes.

8.11.7 Requirement 6: Tools

Description & Priority

This use case is used to allow the users to use the tools that are provided on the website.

Use Case

Sequence 6

Scope

The scope of this use case is to allow the users to test their own website.

Description

This use case describes the user being able to use tools provided on the website.

Flow Description

Precondition

The system is in initialisation mode.

Activation

This use case starts when a user types in “IP” or the “URL” of the website that he wants to test (his own website).

Main flow

1. The system waits for the user to type in IP or URL.
2. The user types in the correct IP or URL.
3. The system verifies if he can connect to the given IP or URL.
4. The system takes credit off from the user.
5. The system tests the IP or URL of webpage.
6. The system displays the result.
7. The user reads the result.

Exceptional flow

E1: Error in URL/IP or accepting the disclaimer

1. The system waits for the user to type in IP or URL.
2. The user types in the IP or URL.
3. The system verifies if he can connect to the given IP or URL.
4. The system returns message saying the IP or URL is incorrect.
5. The user types in the correct IP.
6. The system returns message saying that disclaimer must be ticked for the test to be followed on.

Termination

The system displays result of the test.

Post condition

The system goes into a wait state until a new test is requested.

8.11.8 Requirement 7: Solution

Description & Priority

This use case is used to allows users to understand how to block their website from the tests they performed.

Use Case

Sequence 7

Scope

The scope of this use case is to allow the users to secure their website.

Description

This use case describes the user being able to use secure their website with code.

Flow Description

Precondition

The system is in initialisation mode.

Activation

This use case starts when a user types in “IP” or the “URL” of the website that he wants to test (his own website) and then gets the result of the test, clicks on “get solution” and he must pay some credits to get the tips on how to block it.

Main flow

1. The system waits for the user to type in IP or URL.
2. The user types in the correct IP or URL.
3. The system verifies if he can connect to the given IP or URL.
4. The system takes credit off from the user.
5. The system tests the IP or URL of webpage.
6. The system displays the result.
7. The user presses on “Get security tips”.
8. The system takes credit off from the user.
9. The system displays possible solutions to the problems.
10. The user gets all result with solution.

Exceptional flow

E1: Error in URL/IP or accepting the disclaimer

1. The system waits for the user to type in IP or URL.
2. The user types in the IP or URL.
3. The system verifies if he can connect to the given IP or URL.
4. The system tests the IP or URL, but it can't detect anything.
5. The system displays "No weakness found".

Termination

The system displays result of the test.

Post condition

The system goes into a wait state until a new solution is requested.

8.12 Non-Functional Requirements

Specifies any other non-functional attributes required by the system. Examples are provided below. **Remove the requirement headings that are not appropriate to your project.**

8.12.1 Performance/Response time requirement

The performance and response time will be crucial, it will need to respond and react in real time. Basically, whenever user does anything on the site it has to do it as fast as the service can especially when it comes to tools, some tools will require time to work, some test the sites for few sec or even min, but other than that the site needs to run smooth and fast.

8.12.2 Availability requirement

The penetration testing website will be available to anyone, anyone can use it to test their own website in many ways. The only thing that user will need to agree to is term saying that they own the website, or they can test it, just a disclaimer.

8.12.3 Recover requirement

I'm planning to have everything on c9 basically I will keep a backup of the full site with a database on second machine that will be just for storage purpose. The backup is always needed, anything can happen even to the best services in the world, system failure, machine going down, hackers ext.

8.12.4 Robustness requirement

The only few errors that can occur on the website might be with user not having the enough amount of credits, not having funds in his bank account, not accepting the disclaimer or a simple connection loss of the internet.

8.12.5 Security requirement

My project is a security type, basically it will be for testing, in other words it will need to be secure since the purpose of it is security, ports will have to be closed, everything to be done over https, ftp access blocked after few tries, SQL injections blocked, any other type of attacks will be blocked to all my knowledge.

8.12.6 Reliability requirement

The website needs to be reliable, it goes in money making way and needs to be up and running for any user that wants to use it in good purpose. The site will need to be up all the time because if it will go down, people will go to other sites for testing.

8.12.7 Maintainability requirement

The maintenance of database shouldn't be hard, basically the database will hold credit cards, login details, chat messages (some not all like 15 at max at a time).

8.12.8 Portability requirement

My site will be portable, you will be able to run it on anything with any software android, windows, iOS ext.

8.12.9 Reusability requirement

The site can't be reused, if the website would be reused, same software would come up or with just different testing tools.

8.12.10 Resource utilization requirement

All users can contact the support through the email on the site or if there will be admin online in the chat window.

9 Interface requirements

9.1 GUI

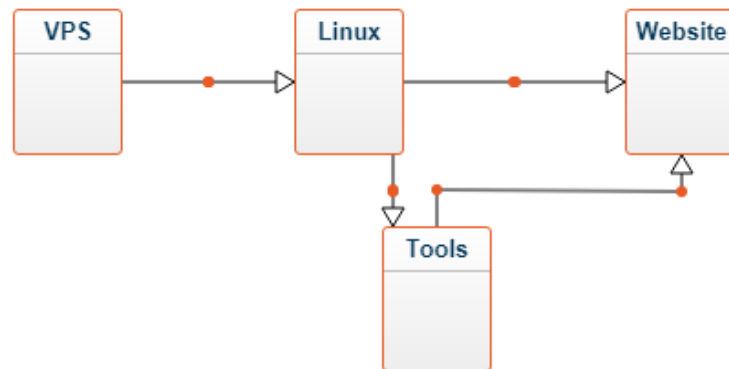
The site will have a simple navigation, left side bar for dashboard, tools, seeing the IP of user, or even contacting the support. The right top menu will contain the logout button, the progress of tools (checking if they are online) and a name of the person that's logged in as also the credit number they have on the account. The site will contain around 12 pages, each tool will get 1 page, all tools will be locked that a person has to login to the site to use any type of tests, left side menu will be linked to around 8 sites, while the top menu to about 2-3.

9.2 Application Programming Interfaces (API)

My project will be providing a chat service, learning place for users and a testing environment for the users. It will provide a simple left and top right menu, clear to use. Left menu will contain dashboard, tools and contact pages while top right will contain the logout and users name as also the online status of tools.

10 System Architecture

Basically, the system will consist of VPS/c9 server that will run the application in a Linux



environment, the tools won't need to be installed to run on the site. The tools will be integrated into the website using python and php scripts to get them to function.

11 System Evolution

Penetration testing web can evolve in few ways, especially one, provide a huge range of tools which can be used for testing, provide a service of helping to test or secure a site, giving 1 on 1 help tutorials via skype ext. There are huge ways in which it can evolve especially in the number of tools.

12 Monthly Journals

12.1 Reflective Journal

Student name: Dawid Trojanowski

Program: BSc (Honours) in Computing & Cyber Security

Month: September

My Achievements

This month,

- I was able to get my project idea sorted.
- I have all my goals set and ready to go.
- Know the design of website.
- Content.
- Security Aspects of website.

My Reflection

During the researches and time spend to think of good idea, I found many things that have been already out there. It was hard to find something unique and good, but I managed to do so. My idea has changed quite a bit from the start but it's going well.

During the researches I couldn't do other projects that I thought about because they were already there, this made me lose more time in thinking what could be good to do. The meeting on Monday 02.10.17 has helped me a lot to realise how to make my project much better and more advanced.

Intended Changes

Next month I want to have basic visuality of website done and some basic functions like login page with registration. I would like to get the main report going, it's quite big and it's going to take a

lot of time, so I would like to start it as soon as possible. Also, I want to have a steady idea, mostly finish all ideas I have in my head and then to think of new that can expand my project.

During my work and researches I have realised that the projects scale is huge, I need to put my head into it and do my best. I need to put as much work as I can into the main document that is in the project.

Supervisor Meetings

Date of Meeting: To be arranged

Items discussed: To be arranged

Action Items: To be arranged

12.2 Reflective Journal

Student name: Dawid Trojanowski

Program: BSc (Honours) in Computing & Cyber Security

Month: October

My Achievements

During past month I managed to create the basic design of the website, I have the main page ready, just changing the content and creating the login with registration. I did well with design, was quite simple. The web page consists most of the design I wanted, with the menu, basic login without functionality, also with the drop-down table.

My Reflection

It did well in creating of my website design and a login page (Without functionality) but on the other hand I'm not happy that I can't find any good details and information on how to connect kali Linux with a webpage so that it will execute requests of the users that they will input into the website.

Intended Changes

To the end of the month I will try to finish up the design with some content and focus on database and login form, also I need to find some good sources on how to connect a kali Linux that is going to be installed on VPS to a webpage. Also, I want to fix up the presentation of the website and prepare a prototype since the mid presentation is getting closer and closer. Supervisor Meetings

Date of Meeting: 24/10/2017

Items discussed: I have talked with my supervisor, we talked about my project and we concluded, basically my old idea was not bad, but it didn't feel that complex and I would have to go just for a pass. During the meeting my supervisor was great, he told me what are the main parts that bring high marks, he explained to me what is needed to aim high, he couldn't give me an idea, but he had put me on track and I had come up with a good idea which I'm working on now.

12.3 Reflective Journal

Student name: Dawid Trojanowski

Program: BSc (Honours) in Computing & Cyber Security

Month: November

My Achievements

This month,

- I have completed the design for most pages,
- Made one of the tools to work,
- Created login and registration.

My Reflection

It worked well during the one-month period to get one tool working, SQL testing tool to run and test the website. But on the other hand, I couldn't get more than just one tool working. It took a lot of time for me to just get one working without implementing it into the website.

Intended Changes

Next month I want to have more than just one tool ready as also the look full ready, just the tools needed to be completed.

Supervisor Meetings

Date of Meeting: 27/11/2017

Items discussed: All project going right way, keeping the track and getting things done, tools discussed, more need to be added.

12.4 Reflective Journal

Student name: Dawid Trojanowski

Program: BSc (Honours) in Computing & Cyber Security

Month: January

My Achievements

This month,

- Know what tools/scripts are going to be on the site,
- Added more functionality to the tool implemented,
- Started to secure the website

My Reflection

It worked well during the one-month period to implement more functionality and play around with design as also adding more security into the application

Intended Changes

Next month I want to have more tools implemented or get the design fully to what I would want it to be.

Supervisor Meetings

Date of Meeting: 24/01/2018

Items discussed: Scripts/tools need to be implemented, at least 3 functional scripts/tools to work and display results.

12.5 Reflective Journal

Student name: Dawid Trojanowski

Program: BSc (Honours) in Computing & Cyber Security

Month: February

My Achievements

This month,

- Implementing of a script to scan the ports,
- Added more functionality to the script and tool implemented,
- Half way through the securing

My Reflection

It worked well during the one-month period to fix up few main security breach points that I could find and evolve the php mysql_query to PDO version with prepared statements and bind parameters.

Intended Changes

Next month I want to have all the security finished with 0 errors or misconfigurations

Supervisor Meetings

Date of Meeting: 23/02/2018

Items discussed: All going right way, working well, need to fix all the security and focus on the scripts/tools to have at least 2-3 scripts/tools

12.6 Reflective Journal

Student name: Dawid Trojanowski

Program: BSc (Honours) in Computing & Cyber Security

Month: March

My Achievements

This month,

- Fixing up the scripts for port testing
- Implementing of TCP scanner.
- Finished the security

My Reflection

It worked well during the one-month period to implement 2 scripts (SQL and TCP scanners) and secure the website as also fix up the session misconfiguration

Supervisor Meetings

Date of Meeting: 26/03/2018

Items discussed: All going right way, working well, need to fix all the security and focus on the scripts/tools to have at least 2-3 scripts/tools

