198735

Introduction to computer Security Coursework – G6077

Contents

Task1 5marks

	RAT	Botnets
1	Compromises one system.	Consist of a large number of compromised systems.
2	The target is this comprised machine.	Used to target another machine.
3	Controlled via keyboard and mouse through the internet	Use of Command and Control variations to control the compromised system.
4	Use of non-standard network protocols to communicate	Use of standard network protocols to communicate
5	Can be used legally for various purposes	Are used for malicious purposes

Task2 5marks

	Weakness	**Improvement**
1	Unencrypted Password in database, if database cracked passwords exposed.	Use of a hash function when storing and retrieving passwords.
2	Sends password in email, if intercepted allows the hacker in.	Don’t send emails with password in, make them reset the password using the website.
3	No checks on password when inputted, could input script.	Check the password only contains a predetermined list of characters for example letters numbers and special characters (!,@,-,….).
4	Default passwords concatenated from an unencrypted text document, easy to crack if text document found.	Randomly generate the newly created account passwords, or even better make the users enter a password on creation of account.
5	If attacker has email and password they can login as you.	Implement a third party authentication system so the user has to verify the account is theirs.

Task3 5marks

p = 13 q = 31 e = 19 m = 2

N = p * q = 13 * 31 = 403

We know e is 19.

Encryption:

So public key = (19, 403)

Plaintext: B → 2 index

$C = m^e \bmod N$

m is 2, N is 403 and e is 19 so:

$C = 2^{19} \bmod 403 = 388$

Ciphertext: D → 388

Decryption:

To find d:

$d * e \bmod \varphi(N) = 1$

$\varphi(N) = (p - 1)(q - 1) = (13 - 1)(31 - 1) = 12 * 30 = 360$

$19d \bmod 360 = 1$

We find any number that matches this:

If d = 19 then:

19 (19) mod 360 = 1

But this is the same as the public key so this isn't allowed.

If d = 379 then:

19 (379) mod 360 = 1

This is allowed so we choose d to be 379.

Therefore: private key = (379, 403)

Ciphertext: D → 388

$M = C^d \bmod N$

C is 388, d is 379, and N is 403 so:

$M = 388^{379} \bmod 403 = 2$

So 2 is our decrypted message.

Task 4

i Implement your own algorithm that perform single or other levels of frequency analysis [5 marks]

```java
/**
 *
 * @author 198735
 */
public class Task4 {


    public Task4 (String encoded_msg){

        // The encoded message is inputted as a string
        // and used as a parameter in the function
        double[] char_freq = singleCharacterFrequencies(encoded_msg);
        System.out.println("Character Frequencies: ");
        for(int i = 0; i < char_freq.length; i++){
            System.out.println((char)(i+65) + ":" + char_freq[i] + " ");
        }
        System.out.println("");
    }

    // Outputs the single character frequency analysis in the form of
    // a double array
    private double[] singleCharacterFrequencies(String encoded_msg){
        // The array storing the frequncies of characters
        double[] char_freq = new double[26];
        for(int i =0; i < encoded_msg.length(); i++){
            // 65 decimal is ascii code for A
            // taking 65 away gives us the place in the double array
            char_freq[encoded_msg.charAt(i)-65] += 1;
        }
        // Works out the frequency of the letters
        // based on the length of the encoded message
        for (int i = 0; i < char_freq.length; i++){
            char_freq[i] = (double) (char_freq[i] / encoded_msg.length()) * 100.0;
        }
        return char_freq;
    }

    public static void main(String[] args) {

        new Task4(args[0]);

    }

}
```

ii Algorithm works [2 marks]

```
[joe@Work-PC java]$ javac Task4.java
[joe@Work-PC java]$ java Task4 PBFPVYFBQXZTYFPBFEQJHDXXQ
VAPTPQJKTOYQWIPBVWLXTOXBTFXQWAXBVCXQWAXFQJVWLEQNTOZQGGQL
FXQWAKVWLXQWAEBIPBFXFQVXGTVJVWLBTPQWAEBFPBFHCVLXBQUFEVWL
XGDPEQVPQGVPPBFTIXPFHXZHVFAGFOTHFEFBQUFTDHZBQPOTHXTYFTOD
XQHFTDPTOGHFQPBQWAQJJTODXQHFOQPWTBDHHIXQVAPBFZQHCFWPFHPB
FIPBQWKFABVYYDZBOTHPBQPQJTQOTOGHFQAPBFEQJHDXXQVAVXEBQPEF
ZBVFOJIWFFACFCCFHQWAUVWFLQHGFXVAFXQHFUFHILTTAVWAFFAWTEVO
ITDHFHFQAITIXPFHXAFQHEFZQWGFLVWPTOFFA
Character Frequencies:
A:5.276381909547738
B:6.532663316582915
C:1.507537688442211
D:2.512562814070352
E:3.015075376884422
F:12.814070351758794
G:2.512562814070352
H:6.281407035175879
I:2.512562814070352
J:2.261306532663317
K:0.7537688442211055
L:2.512562814070352
M:0.0
N:0.25125628140703515
O:3.7688442211055273
P:7.035175879396985
Q:10.552763819095476
R:0.0
S:0.0
T:6.78391959798995
U:1.0050251256281406
V:6.030150753768844
W:5.527638190954774
X:7.035175879396985
Y:1.507537688442211
Z:2.0100502512562812
```

iii At-least 8 words have been decrypted [4 marks]

thanked, the, and, at, hand, enter, are, reread, need

Half decrypted, capitals are ciphertext:

thetoYehaXZTYetheEaJrDXXaodtTtaJkTOYanIthonLXTOXhTeXandXhoCXandXeaJon
LEaNTOZaGGaLeXandkonLXandEhItheXeaoXGToJonLhTtandEhetherCoLXhaUeEo
nLXGDtEaotaGottheTIXterXZroedGeOTreEehaUeTDrZhatOTrXTYeTODXareTDtTO
GreathandaJJTODXareOatnThDrrIXaodtheZarCentertheIthankedhoYYDZhOTrthata
JTaOTOGreadtheEaJrDXXaodoXEhatEeZhoeOJIneedCeCCerandUoneLarGeXodeXar
eUerILTTdondeednTEoOITDrereadITIXterXdearEeZanGeLontTOeed

iv Algorithm is structured well [4 marks]

Task 5

list screenshots of output and code below

i Registration form and login

a Registration Form 1marks

```php
<?php
// registeration form
// 198735

include_once "../src/Functions.php";
session_start();

echo "<form action='registeration.php' method='POST'>";
echo "<pre>";
echo "Name:          ";
echo "    <input name='txtName' type='text'/>";
echo "<br/>";

echo "Email:         ";
echo "    <input name='txtEmail' type='email'/>";
echo "<br/>";

echo "Password:      ";
echo "    <input name='txtPassword' type='password'/>";
echo "<br/>";

echo "Date of Birth:";
echo "    <input name='txtDOB' type='date'/>";
echo "<br/>";

echo "Address:       ";
echo "    <input name='txtAddress' type='text'/>";
echo "<br/>";

echo "<input type='hidden' name='token' value='<?php echo tokenGen(); ?>'/>";

echo "<br/>";
echo "<input type='submit' name='register' value='Register'>  ";
echo "<input type='reset'>";

echo "</pre>";
echo "</form>";
```

```php
//functions.php
//198735
function tokenGen(){

    if(!isset($_SESSION["token"])){
        //gen new
        $token = md5(uniqid(rand(), true));
        $_SESSION['token'] = $token;
    }
    else{
        //reuse
        $token = $_SESSION["token"];
    }
    return $token;
}
```

b Registration Check 1 marks

```php
<!DOCTYPE html>
<!-- registration webpage 198735 -->
<?php
    include '../src/Functions.php';
    headerCSP();
    session_start();
    tokenCheck();
?>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Pizza Restaurant</title>
    </head>
    <body>
        <h1>Pizza Restaurant</h1>
        <?php
        // put your code here
            include "../src/registerationCheck.php";
        ?>
    </body>
</html>
```

```php
<?php
//registerationCheck 198735
include_once '../src/Functions.php';

list($conn,$table) = getConnection(1);
echo $table;

//Values from form
$name= filterData($_POST['txtName']);
$email = filterData($_POST['txtEmail']);
$dob = filterData($_POST['txtDOB']);
$address = filterData($_POST['txtAddress']);
//hashes and salts password with automatic random salt using default php crypt -> blowfish
$password = password_hash(filterData($_POST['txtPassword']), PASSWORD_BCRYPT);

//  INSERT query   , check hash variable in the Values statement
$query = "INSERT INTO ". $table . " (customerName, customerPass, customerEmailAddress, dateOfBirth, Address) VALUES(?,?,?,?,?)";

if (!$statement = $conn->prepare($query)){

    echo "prepare not successful";
    echo "</br> " . $conn->errno . " : " . $conn->error;

}
if(!$statement->bind_param("sssss", $name, $password, $email, $dob, $address)){
    echo "bind para not successful";
    echo "</br> " . $statement->errno . " : " . $statement->error;

}
if(!$statement->execute()){
    echo "execution not successful";
    echo "</br> " . $statement->errno . " : " . $statement->error;
}
$statement->close();
```

```php
<?php
// Functions.php
// 198735

function getData($check){
    $file = fopen('../src/Data.txt', "r") or die("File unavailable!");
    $server = preg_replace('/\s+/', '', fgets($file));
    $db = preg_replace('/\s+/', '', fgets($file));
    $table = preg_replace('/\s+/', '', fgets($file));
    while(!feof($file)){
        $user = preg_replace('/\s+/', '', fgets($file));
        $pass = preg_replace('/\s+/', '', fgets($file));
        if ($check == 1){
            break;
        }
        else if ($check == 2){
            $user = preg_replace('/\s+/', '', fgets($file));
            $pass = preg_replace('/\s+/', '', fgets($file));
            break;
        }
    }
    return array($server, $db, $table, $user, $pass);
}

function getConnection($check){
    $Data = getData($check);
    //echo $Data[0] . $Data[1] . $Data[2] .$Data[3] .$Data[4];
    $connect = new mysqli($Data[0], $Data[3], $Data[4], $Data[1]);
    //new mysqli($host, $username, $passwd, $dbname)
    //check connection
    if ($connect->connect_error){
        die("Connection Failed! -> " . $connect->connect_error);
    }
    return array($connect, $Data[2]);
}

function filterData($data){
    return htmlspecialchars(stripslashes(trim($data)));
}
```

```php
//Functions.php
//198735
function headerCSP(){
    header("Content-Security-Policy: default-src 'self';");
    header("Set-Cookie: samesite=; path=/; domain=localhost; HttpOnly; SameSite=Lax");

}

function tokenGen(){

    if(!isset($_SESSION["token"])){
        //gen new
        $token = md5(uniqid(rand(), true));
        $_SESSION['token'] = $token;
    }
    else{
        //reuse
        $token = $_SESSION["token"];
    }
    return $token;
}

function tokenCheck(){
    if (isset($_SESSION['token']) && ($_POST["token"] != $_SESSION["token"])){
        //reset token
        unset($_SESSION["token"]);
        die("token failed!");
    }
}
```

c Database table 1 mark

Max. rows: 100 Fetched Rows: 1

#	customer_ID	customerName	customerPass	customerEmailAddress	dateOfBirth	Address
1	1	Abi	$2y$10$4i7r.0b6sTXIY6Hc0c/aaeiFKudtK8FJFqwJM6Ec5jPjAqrIdGzr6	abi@abi.com	2020-12-03	20 dmmf fjf

d Screenshot of table before and after successful user registration

Max. rows: 100 Fetched Rows: 1

#	customer_ID	customerName	customerPass	customerEmailAddress	dateOfBirth	Address
1	1	Abi	$2y$10$4i7r.0b6sTXIY6Hc0c/aaeiFKudtK8FJFqwJM6Ec5jPjAqrIdGzr6	abi@abi.com	2020-12-03	20 dmmf fjf

SELECT * FROM Customer LI... ×

Max. rows: 100 Fetched Rows: 2

#	customer_ID	customerName	customerPass	customerEmailAddress	dateOfBirth	Address
1	1	Abi	$2y$10$4i7r.0b6sTXIY6Hc0c/aaeiFKudtK...	abi@abi.com	2020-12-03	20 dmmf fjf
2	2	Dom	$2y$10$pAkyhrMdVArEZO2YCHQQzOk4j...	Dom@dom.com	2020-12-02	234 dom road

e Login form and check scripts 3 marks

```php
<?php
//login form 198735
include_once "../src/Functions.php";
session_start();

echo "<form action='login.php' method='POST'>";
echo "<pre>";
echo "Email:     ";
// name here corosponds to checks input
echo "<input name='txtEmail' type='email' />";
echo "<br/>Password:  ";
// name here corosponds to checks input
echo "<input name='txtPass' type='password' />";

echo "<input type='hidden' name='token' value='<?php echo tokenGen(); ?>'/>";

echo "<br/><input type='submit' name ='login' class='button'  value='Login'>  ";
echo "<input type='reset'>";
echo "</pre>";
echo "</form>";
```

```php
<!DOCTYPE html>
<!-- login webpage -->
<?php
    include '../src/Functions.php';
    headerCSP();
    session_start();
    tokenCheck();
?>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Pizza Restaurant</title>
    </head>
    <body>
        <h1>Pizza Restaurant</h1>
        <?php
        // put your code here
            include "../src/loginCheck.php";
        ?>
    </body>
</html>
```

```php
<?php
// login check 198735
include_once '../src/Functions.php';

//server and db connectioon values
list($conn,$table) = getConnection(2);

// values come from user entry in webform
$email = filterData($_POST['txtEmail']);
$password = filterData($_POST['txtPass']);

//query
$query = "SELECT customerEmailAddress, customerPass FROM " . $table;
$result = $conn->query($query);

//flag type variable - boolean to see if we find user
$userFound = 0;

if ($result->num_rows > 0) {

    while ($userRow = $result->fetch_assoc()) {

        if ($userRow['customerEmailAddress'] == $email) {
            $userFound = 1;
            // verifies password is same as hash
            if (password_verify($password, $userRow['customerPass'])) {

                echo "Hi " . $username . "!";
                echo "<br/> Welcome to our website!";
            } else {

                echo "Wrong password";
            }
        }
    }
}

if ($userFound == 0) {
    echo "This user was not found in our Database";
}
```

```php
<?php
// Functions.php
// 198735

function getData($check){
    $file = fopen('../src/Data.txt', "r") or die("File unavailable!");
    $server = preg_replace('/\s+/', '', fgets($file));
    $db = preg_replace('/\s+/', '', fgets($file));
    $table = preg_replace('/\s+/', '', fgets($file));
    while(!feof($file)){
        $user = preg_replace('/\s+/', '', fgets($file));
        $pass = preg_replace('/\s+/', '', fgets($file));
        if ($check == 1){
            break;
        }
        else if ($check == 2){
            $user = preg_replace('/\s+/', '', fgets($file));
            $pass = preg_replace('/\s+/', '', fgets($file));
            break;
        }
    }
    return array($server, $db, $table, $user, $pass);
}

function getConnection($check){
    $Data = getData($check);
    //echo $Data[0] . $Data[1] . $Data[2] .$Data[3] .$Data[4];
    $connect = new mysqli($Data[0], $Data[3], $Data[4], $Data[1]);
    //new mysqli($host, $username, $passwd, $dbname)
    //check connection
    if ($connect->connect_error){
        die("Connection Failed! -> " . $connect->connect_error);
    }
    return array($connect, $Data[2]);
}

function filterData($data){
    return htmlspecialchars(stripslashes(trim($data)));
}
```

```php
//Functions.php
//198735
function headerCSP(){
    header("Content-Security-Policy: default-src 'self';");
    header("Set-Cookie: samesite=; path=/; domain=localhost; HttpOnly; SameSite=Lax");

}

function tokenGen(){

    if(!isset($_SESSION["token"])){
        //gen new
        $token = md5(uniqid(rand(), true));
        $_SESSION['token'] = $token;
    }
    else{
        //reuse
        $token = $_SESSION["token"];
    }
    return $token;
}

function tokenCheck(){
    if (isset($_SESSION['token']) && ($_POST["token"] != $_SESSION["token"])){
        //reset token
        unset($_SESSION["token"]);
        die("token failed!");
    }
}
```

f Test screen shot of login

g Security. It will be checked from your screenshots of code [9 marks]

ii Updating password

a Updating password form and check. [4 marks]

```
<!DOCTYPE html>
<!-- update Password webpage 198735 -->
<?php
    include '../src/Functions.php';
    headerCSP();
    session_start();
    tokenCheck();
?>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Pizza Restaurant</title>
    </head>
    <body>
        <h1>Pizza Restaurant</h1>
        <?php
        // put your code here
            include "../src/updatePasswordCheck.php";

        ?>
    </body>
</html>
```

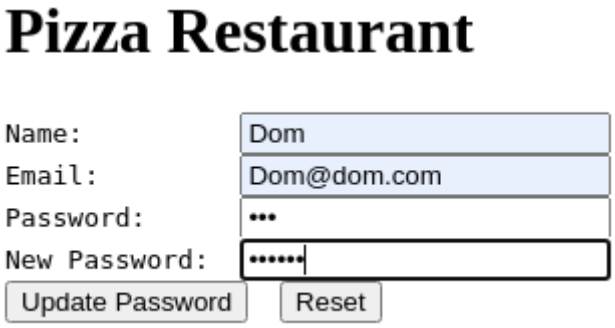
```php
<?php
//updatepasswordform 198735
include_once "../src/Functions.php";
session_start();

echo "<form action='updatePassword.php' method='POST'>";
echo "<pre>";
echo "Name:          ";
// name here corosponds to checks input
echo "<input name='txtName' type='text' />";
echo "<br/>Email:         ";
// name here corosponds to checks input
echo "<input name='txtEmail' type='email' />";
echo "<br/>Password:      ";
// name here corosponds to checks input
echo "<input name='txtPass' type='password' />";
echo "<br/>New Password:  ";
// name here corosponds to checks input
echo "<input name='txtNewPass' type='password' />";

echo "<input type='hidden' name='token' value='<?php echo tokenGen(); ?>'/>";

echo "<br/><input type='submit' name ='update' class='button'  value='Update Password'>  ";
echo "<input type='reset'>";
echo "</pre>";
echo "</form>";
```

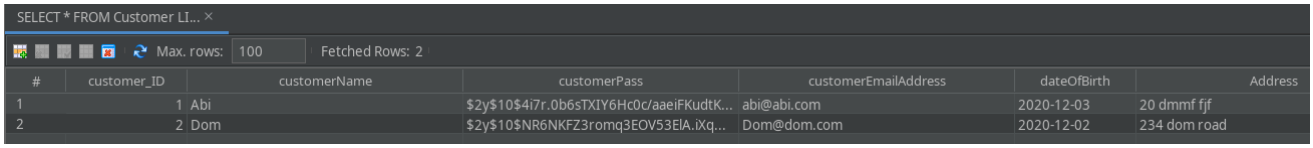
```php
<?php
//updatepasswordcheck 198735
include_once '../src/Functions.php';
//server and db connectioon values
list($conn,$table) = getConnection(3);
// values come from user entry in webform
$name = filterData($_POST['txtName']);
$email = filterData($_POST['txtEmail']);
$password = filterData($_POST['txtPass']);
$newpassword = password_hash(filterData($_POST['txtNewPass']), PASSWORD_BCRYPT);
//query
$passQuery = "SELECT customerEmailAddress, customerPass FROM " . $table;
$result = $conn->query($passQuery);
//flag type variable - boolean to see if we find user
$userFound = 0;
if ($result->num_rows > 0) {

    while ($userRow = $result->fetch_assoc()) {

        if ($userRow['customerEmailAddress'] == $email) {
            $userFound = 1;
            // verifies password is same as hash
            if (password_verify($password, $userRow['customerPass'])) {

                $result->close();
                $query = "UPDATE " . $table . " SET customerPass =? "
                 . "WHERE customerName =? AND customerEmailAddress =?";

                if (!$statement = $conn->prepare($query)){

                    echo "prepare not successful";
                    echo "</br> " . $conn->errno . " : " . $conn->error;
                }
                if(!$statement->bind_param("sss", $newpassword, $name, $email)){
                    echo "bind para not successful";
                    echo "</br> " . $statement->errno . " : " . $statement->error;
                }
                if(!$statement->execute()){
                    echo "execution not successful";
                    echo "</br> " . $statement->errno . " : " . $statement->error;
                } else {
                    echo "Update Password Successful";
                }
                $statement->close();
            } else {
                echo "Wrong password";
            }
        }
    }
}
if ($userFound == 0) {
    echo "This user was not found in our Database";
}
```

```php
<?php
// Functions.php
// 198735

function getData($check){
    $file = fopen('../src/Data.txt', "r") or die("File unavailable!");
    $server = preg_replace('/\s+/', '', fgets($file));
    $db = preg_replace('/\s+/', '', fgets($file));
    $table = preg_replace('/\s+/', '', fgets($file));
    while(!feof($file)){
        $user = preg_replace('/\s+/', '', fgets($file));
        $pass = preg_replace('/\s+/', '', fgets($file));
        if ($check == 1){
            break;
        }
        else if ($check == 2){
            $user = preg_replace('/\s+/', '', fgets($file));
            $pass = preg_replace('/\s+/', '', fgets($file));
            break;
        }
    }
    return array($server, $db, $table, $user, $pass);
}

function getConnection($check){
    $Data = getData($check);
    //echo $Data[0] . $Data[1] . $Data[2] .$Data[3] .$Data[4];
    $connect = new mysqli($Data[0], $Data[3], $Data[4], $Data[1]);
    //new mysqli($host, $username, $passwd, $dbname)
    //check connection
    if ($connect->connect_error){
        die("Connection Failed! -> " . $connect->connect_error);
    }
    return array($connect, $Data[2]);
}

function filterData($data){
    return htmlspecialchars(stripslashes(trim($data)));
}
```

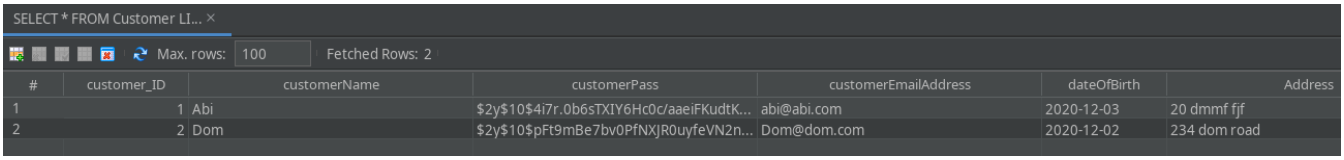
```php
//Functions.php
//198735
function headerCSP(){
    header("Content-Security-Policy: default-src 'self';");
    header("Set-Cookie: samesite=; path=/; domain=localhost; HttpOnly; SameSite=Lax");

}

function tokenGen(){

    if(!isset($_SESSION["token"])){
        //gen new
        $token = md5(uniqid(rand(), true));
        $_SESSION['token'] = $token;
    }
    else{
        //reuse
        $token = $_SESSION["token"];
    }
    return $token;
}

function tokenCheck(){
    if (isset($_SESSION['token']) && ($_POST["token"] != $_SESSION["token"])){
        //reset token
        unset($_SESSION["token"]);
        die("token failed!");
    }
}
```

b Test screen shots of updating password before and after.





#	customer_ID	customerName	customerPass	customerEmailAddress	dateOfBirth	Address
1	1	Abi	$2y$10$4i7r.0b6sTXIY6Hc0c/aaeiFKudtK...	abi@abi.com	2020-12-03	20 dmmf fjf
2	2	Dom	$2y$10$NR6NKFZ3romq3EOV53ElA.iXq...	Dom@dom.com	2020-12-02	234 dom road



#	customer_ID	customerName	customerPass	customerEmailAddress	dateOfBirth	Address
1	1	Abi	$2y$10$4i7r.0b6sTXIY6Hc0c/aaeiFKudtK...	abi@abi.com	2020-12-03	20 dmmf fjf
2	2	Dom	$2y$10$pFt9mBe7bv0PfNXJR0uyfeVN2n...	Dom@dom.com	2020-12-02	234 dom road

c Security. It will be checked from your screenshots of code [11 marks]

iii Reasons of i and ii are secure.

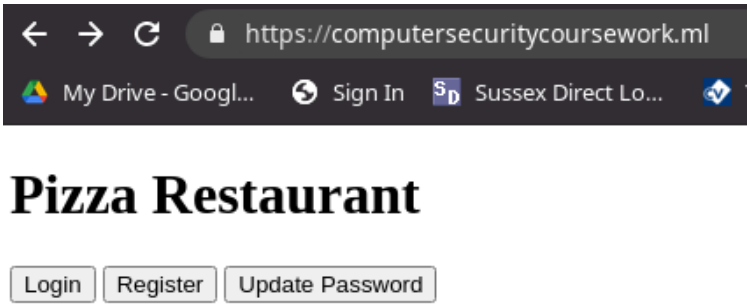
- No php code checks or forms are stored in the web root folder
- Use of CSRF tokens.
- Prepared MySQL Statements.
- Use of header CSP for only self.
- Specific Users used for MySQL server with limited permissions that allow only the prepared statement type.
- Encrypted passwords in Server.
- Passwords are updated and not edited.
- Name and Email Address are needed to update Password
- All data inputted by forms are filtered using trim() stripslashes() and htmlspecialchars().
- All passwords and database details are stored in a file outside the web root and are read into php.

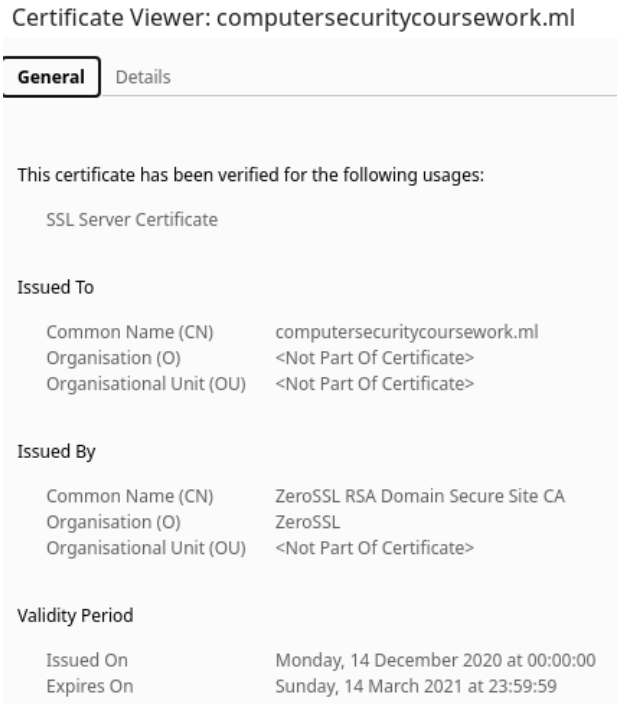
iv Deeper understanding and beyond

For any area to be consider for marking, it needs to make application secure, must be significant and student should have extended the boundary of knowledge to do it themselves.

a) HTTPS

Implementation evidence must be provided in full and in proper order + Video will be checked as well



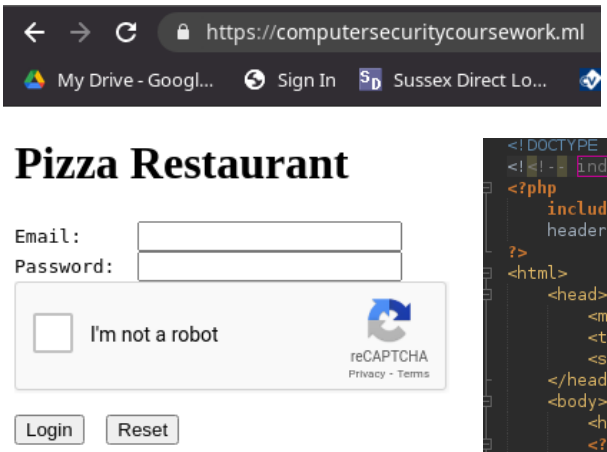


Reasons + attacks

- i Reason 1
  - i.a secure bi directional encrypted transmission across a computer network
- ii Reason 2
  - ii.a protects against man in middle attacks
- iii Reason 3
  - iii.a attackers can only know there is a connection between two parties along with their domain names and IP addresses but nothing else.

b) reCaptcha

Implementation evidence must be provided in full and in proper order + Video will be checked as well




```php
<!DOCTYPE html>
<!<!- index 198735 ->
<?php
    include '../src/Functions.php';
    headerCSP();
?>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Pizza Restaurant</title>
        <script src="https://www.google.com/recaptcha/api.js" async defer></script>
    </head>
    <body>
        <h1>Pizza Restaurant</h1>
        <?php

            if (isset($_POST['form1'])){
                include '../src/loginForm.php';
            }
            else if (isset($_POST['form2'])){
                include '../src/registerationForm.php';
            }
            else if (isset($_POST['form3'])){
                include '../src/updatePasswordForm.php';
            }
            else {

                echo "<form method = 'post'>";
                echo "<input type='submit' name='form1' class='button' value='Login'/>  ";
                echo "<input type='submit' name='form2' class='button' value='Register'/> ";
                echo "<input type='submit' name='form3' class='button' value='Update Password'/>";
                echo "</form>";
            }

        ?>
    </body>
</html>
```

```php
<?php
//login form 198735
include_once "../src/Functions.php";
session_start();

echo "<form action='login.php' method='POST'>";
echo "<pre>";
echo "Email:     ";
// name here corosponds to checks input
echo "<input name='txtEmail' type='email' />";
echo "<br/>Password:  ";
// name here corosponds to checks input
echo "<input name='txtPass' type='password' />";
//<?php echo getreCaptcha(1);

echo "<br/><div class='g-recaptcha' data-sitekey='6LdDDQUaAAAAAHnV7gyHJ_3LCiTzRbe56XDEU8E6'></div>";
echo "<input type='hidden' name='token' value='<?php echo tokenGen(); ?>'/>";

echo "<br/><input type='submit' name ='login' class='button'  value='Login'>  ";
echo "<input type='reset'>";
echo "</pre>";
echo "</form>";
```

```php
function getreCaptcha($key){
    $file = fopen('../src/reCaptcha.txt', "r") or die("File unavailable!");
    $captcha = preg_replace('/\s+/', '', fgets($file));
    if ($key == 2){
        $captcha = preg_replace('/\s+/', '', fgets($file));
    }
    return $captcha;
}

function checkreCaptcha(){
    if (isset($_POST['g-recaptcha-response']) && !empty($_POST['g-recaptcha-response'])){

    $verify = file_get_contents('https://www.google.com/recaptcha/api/siteverify?secret='.
            getreCaptcha(2).'&response='.$_POST['g-recaptcha-response']);
    if(json_decode($verify)->success){
        echo 'Verification Success <br/>';
    }
    else{
        die('reCaptcha verification failed');
    }
}
else{
    die('reCaptcha verification needs to be checked');
}
}
```

```php
<?php
// login check 198735
include_once '../src/Functions.php';

checkreCaptcha();

//server and db connectioon values
list($conn,$table) = getConnection(2);

// values come from user entry in webform
$email = filterData($_POST['txtEmail']);
$password = filterData($_POST['txtPass']);
```

Reasons + attacks

i Reason 1
  i.a combat internet bots, i.e. botnets, from trying to login to the website
ii Reason 2
  ii.a distinguish between human and robot so only customers can login to the site
iii Reason 3
  iii.a protects the emails of the customers that are on the website from web crawlers

c) Two Factor Authentication

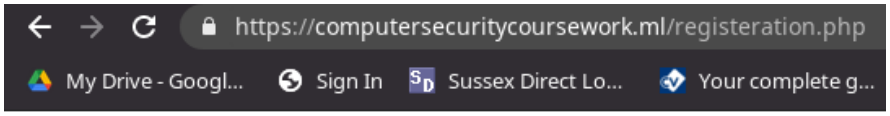
Implementation evidence must be provided in full and in proper order + Video will be checked as well

```php
<?php
//registerationCheck 198735
include_once '../src/Functions.php';
require_once '../src/vendor/autoload.php';
$google2fa = new PragmaRX\Google2FAQRCode\Google2FA();
checkreCaptcha();

list($conn,$table) = getConnection(1);

//Values from form
$name= filterData($_POST['txtName']);
$email = filterData($_POST['txtEmail']);
$dob = filterData($_POST['txtDOB']);
$address = filterData($_POST['txtAddress']);
//hashes and salts password with automatic random salt using default php crypt -> blowfish
$password = password_hash(filterData($_POST['txtPassword']), PASSWORD_BCRYPT);

$key = $google2fa->generateSecretKey();

//  INSERT query   , check hash variable in the Values statement
$query = "INSERT INTO ". $table
        . " (customerName, customerPass, customerEmailAddress, dateOfBirth, Address, secretKey) "
        . "VALUES(?,?,?,?,?,?)";

if (!$statement = $conn->prepare($query)){

    echo "prepare not successful";
    die("</br> " . $conn->errno . " : " . $conn->error);

}
if(!$statement->bind_param("ssssss", $name, $password, $email, $dob, $address, $key)){
    echo "bind para not successful";
    die("</br> " . $statement->errno . " : " . $statement->error);

}
if(!$statement->execute()){
    echo "execution not successful";
    die("</br> " . $statement->errno . " : " . $statement->error);
}
$statement->close();
echo "Successful! Scan QRCode for two factor authentication: <br/>";
$Url = $google2fa->getQRCodeInline(
        'Computer Security Pizza',
         $email,
         $key
);
echo $Url;
```

#	customer_ID	customerName	customerPass	customerEmailAddress	dateOfBirth	Address	secretKey
1	17	Abi	$2y$10$ml./6L096KBWq70OFbELQ.jTHiS...	abi@abi.com	2020-12-04	20 dmmf fjf	2CRN6IZ3RIN3HJC4

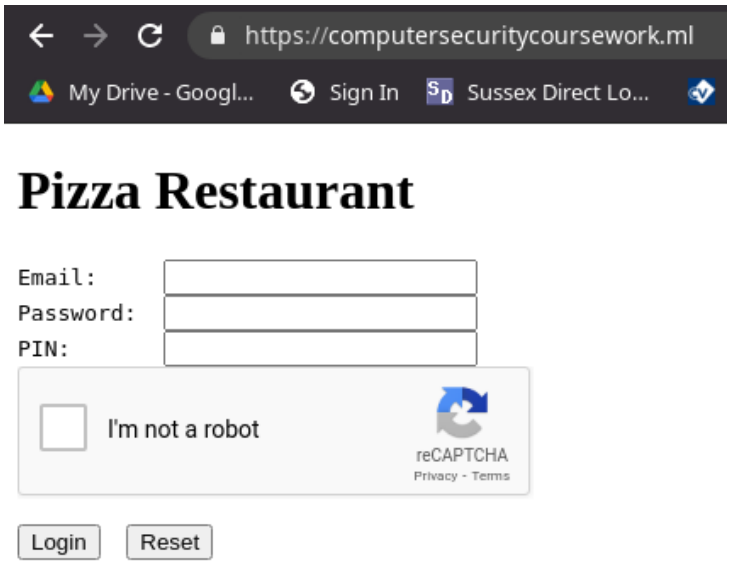
```php
<?php
//login form 198735
include_once "../src/Functions.php";
session_start();

echo "<form action='login.php' method='POST'>";
echo "<pre>";
echo "Email:     ";
// name here corosponds to checks input
echo "<input name='txtEmail' type='email' />";
echo "<br/>Password:  ";
// name here corosponds to checks input
echo "<input name='txtPass' type='password' />";
//<?php echo getreCaptcha(1);

echo "<br/>PIN:       ";
echo "<input name='txtSecret' type='text' />";

echo "<br/><div class='g-recaptcha' data-sitekey='6LdDDQUaAAAAAHnV7gyHJ_3LCiTzRbe56XDEU8E6'></div>";
echo "<input type='hidden' name='token' value='<?php echo tokenGen(); ?>'/>";

echo "<br/><input type='submit' name ='login' class='button'  value='Login'>  ";
echo "<input type='reset'>";
echo "</pre>";
echo "</form>";
```

```php
<?php
// login check 198735
include_once '../src/Functions.php';
require_once '../src/vendor/autoload.php';
$google2fa = new PragmaRX\Google2FAQRCode\Google2FA();
checkreCaptcha();

//server and db connectioon values
list($conn,$table) = getConnection(2);

// values come from user entry in webform
$email = filterData($_POST['txtEmail']);
$password = filterData($_POST['txtPass']);
$secret = filterData($_POST['txtSecret']);

//query
$query = "SELECT customerEmailAddress, customerPass, secretKey FROM " . $table;
$result = $conn->query($query);

//flag type variable - boolean to see if we find user
$userFound = 0;

if ($result->num_rows > 0) {

    while ($userRow = $result->fetch_assoc()) {

        if ($userRow['customerEmailAddress'] == $email) {
            $userFound = 1;
            // verifies password is same as hash
            if (password_verify($password, $userRow['customerPass'])) {

                if ($google2fa->verifyKey($userRow['secretKey'],$secret)){
                    echo "Hi " . $username . "!";
                    echo "<br/> Welcome to our website!";
                }
                else {
                    echo "Wrong PIN";
                }
            } else {
                echo "Wrong password";
            }
        }
    }
}

if ($userFound == 0) {
    echo "This user was not found in our Database";
}
```



Pizza Restaurant

Verification Success
Successful! Scan QRCode for two factor authentication:



Reasons + attacks

i Reason 1
  i.a Even if attack gets password they need the secret Key to access the PIN codes needed to login to the website.

ii Reason 2
  ii.a Attacker only has a 2 min window to secure a PIN to input into login

iii Reason 3
  iii.a PIN codes are forever changing and are never the same.

Task 6) Documentation [5 marks]

i) Using template correctly [1 mark]

ii) Recording video of task 5 explaining all aspects [2 marks]

iii) Fill in the self-assessment column in the form below [2 marks]

	Evidence and marking scheme list	**Self-assessment marks/ Mark your own effort.**
Task 1/5marks	Five significance differences are listed. [1 mark for each]	5
Task 2/5marks	Five important weaknesses and five improvements. [1/2 mark for each]	5
Task 3/5marks	All the steps shown, and correct values are given. [1 mark for correct answer while 4 for the steps]	5
Task 4 / 15 marks	I Implement your own algorithm that perform single or other levels of frequency analysis [5 marks] II Algorithm works [2 marks] III At-least 8 words have been used [4 marks] IV Algorithm is structured well [4 marks]	13
Task 5 / I = 15 marks II= 15marks III 5marks IV 30 marks	I a Registration Form [1mark] b Registration Check [1 mark] c Database table [1 mark] d Screenshot of table before and after successful user registration e Login form and check scripts [3 marks] f Test screen shot of login g Security. It will be checked from your screenshots of code [9 marks] II Updating password a Updating password form and check. [4 marks] b Test screen shots of updating password before and after. c Security. It will be checked from your screenshots of code [11 marks] III Reasons of i and ii are secure. a Reasons for (i) 3 marks b Reasons for (ii) 2 marks IV Deeper understanding and beyond For any area to be consider for marking in this section, it needs to make a application secure b must be significant c student should have extended the boundary of knowledge beyond.	15 15 5 27
Task 6 / 5marks		5
	Total self-assessed marks	