

Configuration Manual

SecureWeb : Elevating Web Authentication with PCCP and Email- Driven OTP

MSc Research Project
Cyber Security

Sriram Kalyanasundaram
Student ID: X21246734

School of Computing
National College of Ireland

Supervisor: Rohit Verma

National College of Ireland MSc
Project Submission Sheet
School
of Computing



Student Name: Sriram Kalyanasundaram
Student ID: X21246734
Programme: MSc. in CyberSecurity **Year:** 2023
Module: MSc in Research project
Lecturer: Rohit Verma
Submission Due Date: 31/1/2024
Project Title: SecureWeb : Elevating Web Authentication with PCCP and Email-Driven OTP

Word Count: 1064 **Page Count:** 9

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Sriram Kalyanasundaram
Date: 31/1/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

| | |
|---|--------------------------|
| Attach a completed copy of this sheet to each project (including multiple copies) | <input type="checkbox"/> |
| Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies). | <input type="checkbox"/> |
| You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. | <input type="checkbox"/> |

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

| | |
|----------------------------------|--|
| Office Use Only | |
| Signature: | |
| Date: | |
| Penalty Applied (if applicable): | |

Configuration Manual

Sriram Kalyanasundaram
Student ID: x21246734

1 Introduction:

To download and execute the SecureWeb: Elevating Web Authentication with PCCP and Email-Driven OTP prototype successfully, it is important to meet specific software and hardware requirements. The software stack used to develop this prototype was chosen with careful consideration, as explained further below. In addition, some important functionality of the prototype is mentioned as code for discussion.

2 Uploading Image:

The images uploaded to the system do not necessarily meet any requirements. It is preferable to use images with more subject matter so that the click-able points are more, and it will drastically increase the security.

The images shown in the prototype are random images downloaded across the internet and used for predefined image authentication purposes.

3 Hardware Requirements:

- RAM: 8.00 GB
- The system type used for developing the application was **64-bit operating system, x64-based processor.**
- The application was developed and tested in a window 10 pro 22h2 version.
- Good internet connection is required as the application is hosted online on Azure.

4 Software Requirements:

- Windows 7,8 or the above versions of Operating Systems
- MySQL was used for storing the data for the application.
- The entire project was built in the Visual Studio Code 2022 development environment. The front-end design was built with a combination of HTML, Bootstrap, and JavaScript, while the server-side logic and data management were handled efficiently with the .NET framework.
- Smtplib protocol was used to configure an email trigger when the user requests a login.
- An email was created for this sole purpose “sriram.leodas@gmail.com” and was integrated with .NET smoothly and the functionality was achieved.

- As a web application, it was hosted on Azure account using the college email account, and SQL server studio management studio 19 was used to visualize the data saved in the application. (The IP address must be saved in the Azure account to visualize the data that is stored)

5 Key functions of the project:

The function used to collect the click points when the points are clicked during registration process.

(File name: RegM2.aspx.cs)

```

98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
    if (x != "" && y != "")
    {
        using (SqlCommand cmd4 = new SqlCommand("select * from image_master_m2 where uid=@id", con))
        {
            cmd4.Parameters.AddWithValue("@id", Session["uid"].ToString());
            using (SqlDataAdapter adp4 = new SqlDataAdapter(cmd4))
            {
                DataTable dt4 = new DataTable();
                adp4.Fill(dt4);
                if (dt4.Rows.Count == 0)
                {
                    Response.Redirect("Registration.aspx");
                }
                else
                {
                    using (SqlCommand cmd2 = new SqlCommand("select top 3 * from image_master order by newid()", con))
                    {
                        using (SqlDataAdapter adp = new SqlDataAdapter(cmd2))
                        {
                            DataTable dt1 = new DataTable();
                            adp.Fill(dt1);
                            if (dt1.Rows.Count > 3)
                            {
                                Response.Redirect("Index.aspx?reg=1");
                            }
                            else
                            {
                                if (i == 1)
                                {
                                    Image1.ImageUrl = "data:image;base64," + Convert.ToBase64String(b1);
                                    bytes1 = b;
                                }
                                else if (i == 2)

```

Figure 1(a): Pushing image into database

```

    {
        Image1.ImageUrl = "data:image;base64," + Convert.ToBase64String(b2);
        bytes1 = b1;
    }
    else if (i == 3)
    {
        bytes1 = b2;
    }
}

using (SqlCommand cmd = new SqlCommand("insert into image_master_m2(uid,image_startx,starty) values(@id,@image,@x,@y) ", con))
{
    cmd.Parameters.AddWithValue("@id", Session["uid"].ToString());
    cmd.Parameters.AddWithValue("@x", x);
    cmd.Parameters.AddWithValue("@y", y);

    SqlParameter p1 = new SqlParameter("@image", SqlDbType.Image);
    p1.Value = bytes1;
    if (bytes1.Length > 20)
    {
        i = i;
    }
    else
    {
        cmd.Parameters.Add(p1);
        con.Open();
        cmd.ExecuteNonQuery();
        con.Close();
        i++;
    }
}

using (SqlCommand cmd1 = new SqlCommand("select * from image_master_m2 where uid=@id", con))
{
    cmd1.Parameters.AddWithValue("@id", Session["uid"].ToString());
    using (SqlDataAdapter adp3 = new SqlDataAdapter(cmd1))
    {
        DataTable dt2 = new DataTable();
        adp3.Fill(dt2);
        if (dt2.Rows.Count >= 3)

```

Figure 1(b): Pushing image into database

Generating OTP:

Here a random binary of character 0 & 1 otp of three digits is created randomly for each trigger and they are stored in the database. While triggering the email, the generated data stored in the OTP table from the database based on the id is sent to the user who has requested a login.

```
114         if (dt.Rows.Count > 0)
115         {
116             Session["id"] = dt.Rows[0]["id"].ToString();
117             string useremail = dt.Rows[0]["email"].ToString();
118             //Session["UId"] = dt.Rows[0]["id"].ToString();
119
120             var chars = "01";
121             var stringChars = new char[3];
122             var random = new Random();
123
124             for (int i = 0; i < stringChars.Length; i++)
125             {
126                 stringChars[i] = chars[random.Next(chars.Length)];
127             }
128
129             var finalString = new String(stringChars);
130
131             using (SqlCommand cmd2 = new SqlCommand("select * from otpdetails where userid=@id", con))
132             {
133                 cmd2.Parameters.AddWithValue("@id", Session["id"].ToString());
134                 using (SqlDataAdapter adp2 = new SqlDataAdapter(cmd2))
135                 {
136                     DataTable dt2 = new DataTable();
137                     adp2.Fill(dt2);
138                     if (dt2.Rows.Count > 0)
139                     {
140                         using (SqlCommand cmd3 = new SqlCommand("update otpdetails set otp=@otp where userid=@id", con))
141                         {
142                             cmd3.Parameters.AddWithValue("@id", Session["id"].ToString());
143                             cmd3.Parameters.AddWithValue("@otp", finalString);
144                             con.Open();
145                             cmd3.ExecuteNonQuery();
146                             con.Close();
147                         }
148                     }
149                 }
150             }
151         }
152     }
153 }
```

Figure 2: Generating an OTP for the SMTP trigger

Configuring SMTP protocol code Snippet:

The mail Id used here is sriram.leodas@gmail.com especially for the thesis mail triggering functionality. Line 116 contains the code generated by Google for the SMTP protocol to use as a password to access the email each time an email is triggered. The triggered email delivers an OTP to the user from which the inputs for the image are selected.

```
114         SmtplibClient SmtServer = new SmtplibClient();
115         MailMessage mail = new MailMessage();
116         SmtServer.Credentials = new System.Net.NetworkCredential("sriram.leodas@gmail.com", "zhplqjbdhtjshtwa");
117         SmtServer.Port = 587;
118         SmtServer.EnableSsl = true;
119         SmtServer.Host = "smtp.gmail.com";
120         mail = new MailMessage();
121         mail.From = new MailAddress("sriram.leodas@gmail.com");
122
123         mail.To.Add(useremail);
124         mail.Subject = "Otp";
125         mail.Body = "Your OTP=" + finalString + "(0 is for Wrong point and 1 is for Correct Point)";
126         SmtServer.Send(mail);
127
128
129
130         Response.Redirect("Login3.aspx?log=1");
131     }
132     else
133     {
134         Response.Redirect("Login.aspx?log=error");
135     }
136 }
137 }
138 }
```

Figure 3: SMTP Protocol Configuration

After installing the Visual Studio Code and importing the file into Visual Studio code the Solution Explorer looks as shown in figure 4.

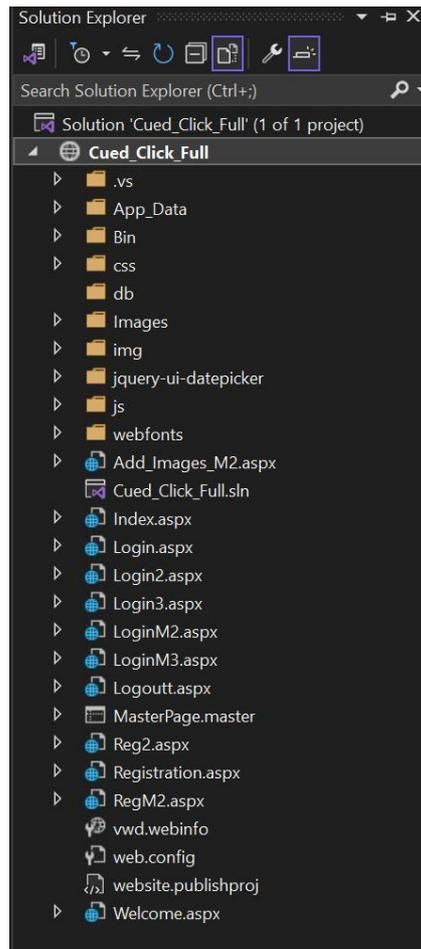


Figure 4: Solution Explorer of the code file when inserted in Visual Studio

6 Prototype flow:

Homepage:

The following options are available on the prototype's home page: To Register and Login. If you are a new user, you must register; if you are an existing user, you can login using the methods listed below. M1, M2, M3.

M1 - Traditional text-based password login.

M2 - To login with images uploaded by the user via an email.

M3 - Login using predefined images provided by the system via email.

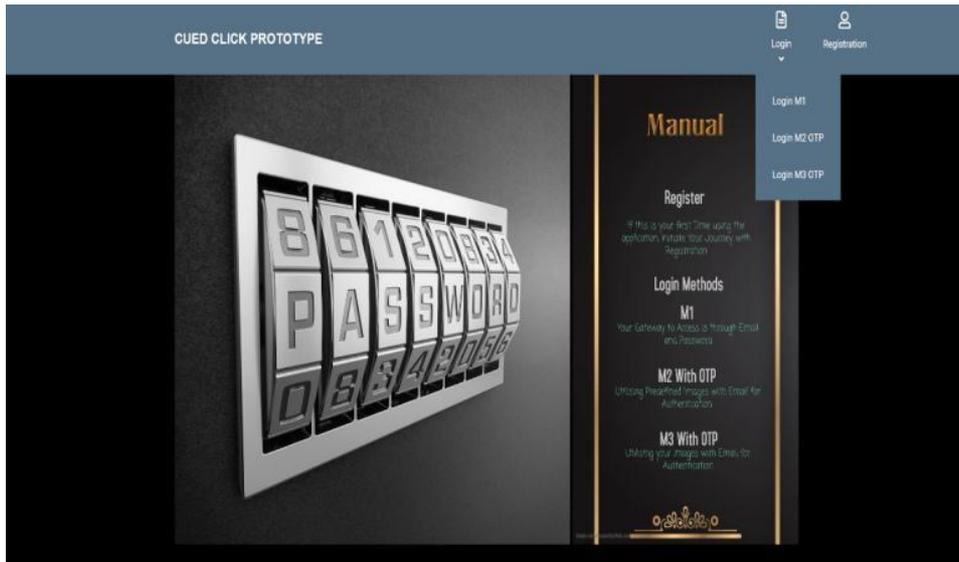


Figure 5: Home Page

Registration:

Users must provide the required information to begin the registration process as shown in Figure 6. They will then proceed to the next step, which will involve the creation of a visual password.

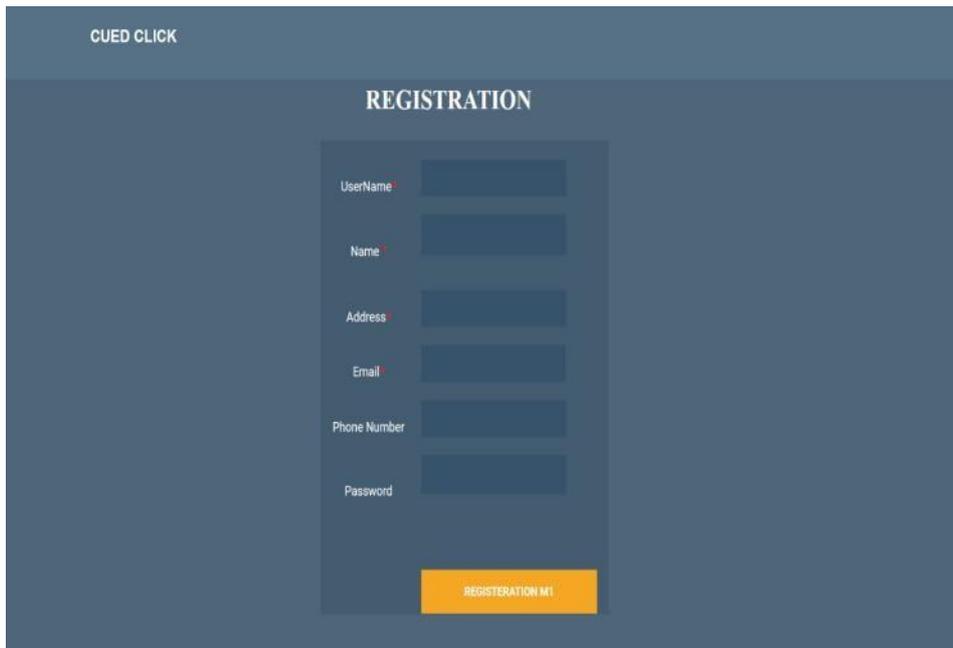


Figure 6: Registration Page

Selecting images:

After entering the required information, you will be redirected to this page where you will be requested to upload an image of your choice following that a point on the image must be selected which will be considered as the visual password as shown in the Figure 7. After an image is uploaded and a point is selected it will be acknowledged on the screen that the point is selected now the next button must be pressed as shown in Figure 8. This process is repeated until three images are uploaded.

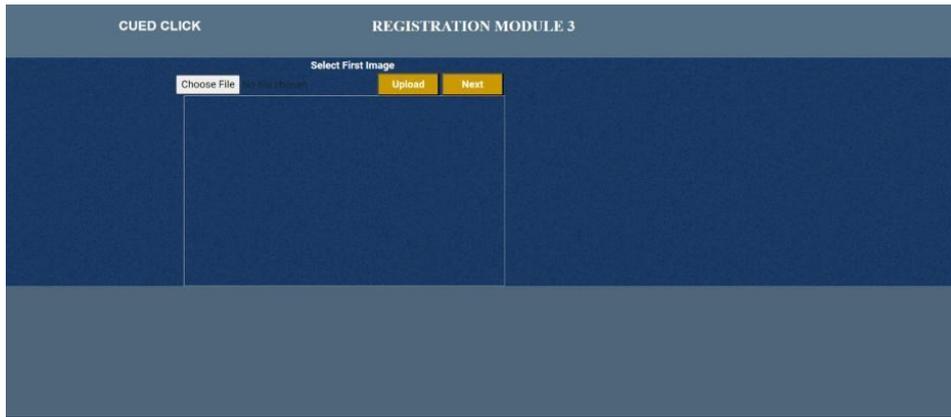


Figure 7: Selecting Image window

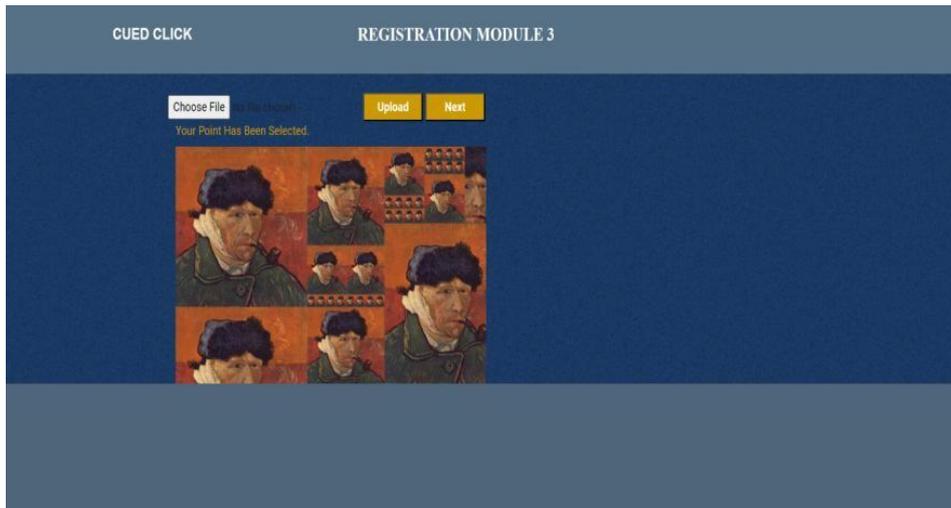


Figure 8: Acknowledgment

Logging in:

When the user enters their username (as shown in Figure 9), they will receive an email with a three-digit binary code (as shown in Figure 10) which will be the source of selecting the correct or wrong point that was previously selected during the registration phase.

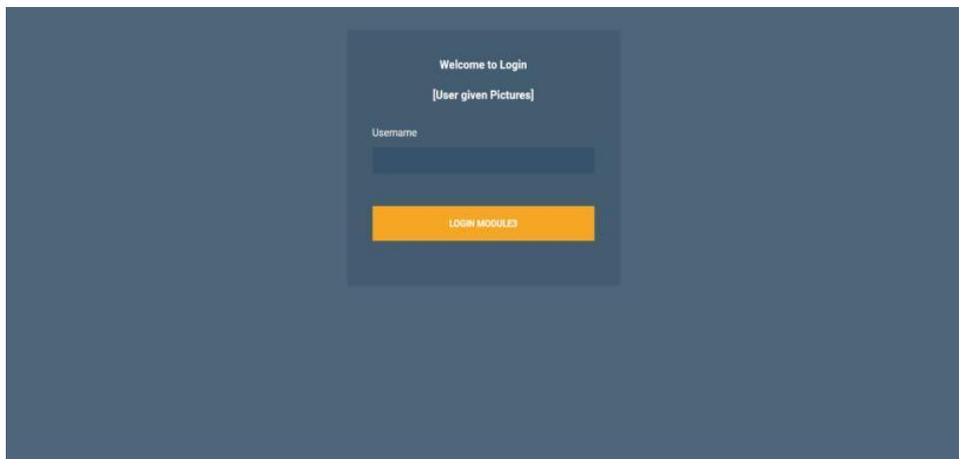


Figure 9: Login Page



sriram.leodas@gmail.com

to me ▼

Your OTP=110(0 is for Wrong point and 1 is for Correct Point)

Figure 10: OTP