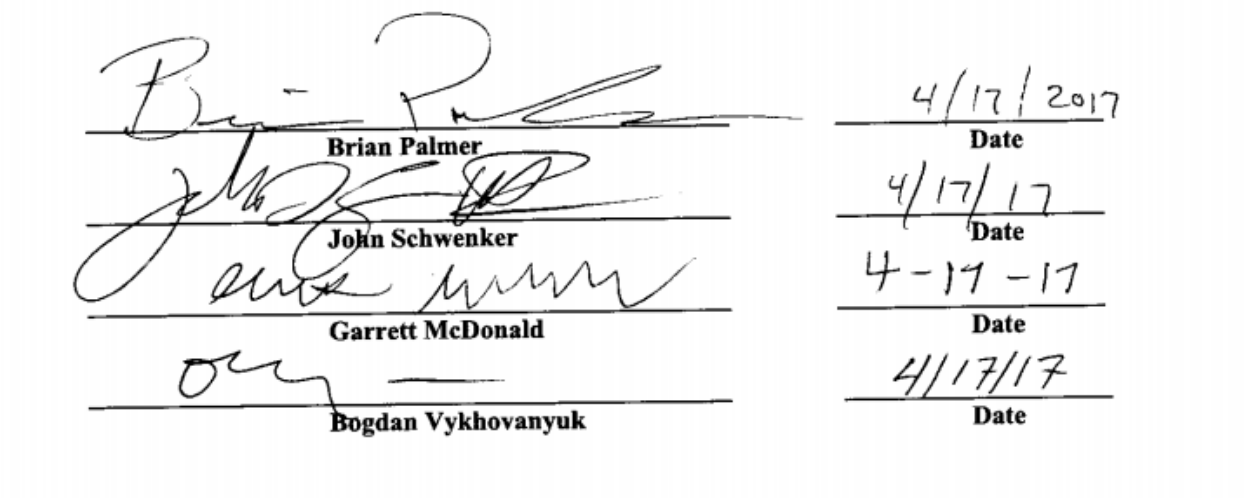
T4 Authentication

By

Brian Palmer, John Schwenker, and Garrett McDonald

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology



Brian Palmer	4/17/2017 Date
John Schwenker	4/17/17 Date
Garrett McDonald	4-17-17 Date
Bogdan Vykhovanyuk	4/17/17 Date

University of Cincinnati
College of
Education, Criminal Justice, and Human Services
April 2017

Contents

Tables and Figures

Abstract

In this day and age, everything from banking to health services are accessible online 24/7, anywhere in the world. With the accessibility and prevalence of this information, there is an inherent risk of your data being compromised. T4 Authentication presents a method of multifactor authentication that curbs this risk by eliminating the shortcomings of traditional multifactor authentication such as enterable text fields. Using something you have, something you know, and pseudo-random generation, we present a stronger method of multifactor authentication. The way in which these components are combined sets T4 Authentication apart. Through the synchronization of patterns on both a mobile device and a web client, the user is able to discreetly enter their pin without fear of compromisation. When you use T4 Authentication, you can rest easy knowing that only you will be able to access your data.

1. Problem Statement

1.1 Introduction

T4 Authentication is a startup group consisting of Brian Palmer, John Schwenker and Garrett McDonald. The three members are all Information Technology majors at The University of Cincinnati. The group unifies as a senior design group, with an interest in developing an improved method of multi-factor authentication.

As a cyber security major, Garrett McDonald has learned numerous ways of performing multi-factor authentication and, with their programming skills, Brian and John have developed a way to implement them.

1.2 Project Description

The project will contain both a phone app and a Web client, backed by a database. Upon login on the phone app, a diagram will be generated that resembles a tic-tac-toe board, with the numbers 0 - 9 randomly positioned and shown in the different sections. The user will have a 4 digit pin, only known by them that must be entered in order to correctly log into the site. Upon login of the username and password on the Web client, the same hash board will be shown, however the randomly generated numbers will be replaced by dots. By looking at the number placement on the phone app, the user must correctly click the corresponding squares of where the correct pin numbers are. This new method of authentication is important because even if there is a breach in

security, the pin is never typed, removing the potential of a key logger. The second advantage is that number positions change on a timed basis, the pin location will not be the same for the next time a criminal attempts to log in. Also, there is no way for an outside user to determine the correct click pattern without knowing the pin placement. The figure below shows an outline of how the layout of the grid will work on both the mobile application as well as the Web client.

Figure 1 Project Description

1.3 Problem

The problem that we face in this day and age is safely authenticating users to make sure they are who they say they are. While traditional two factor authentication is a better known way of authentication, there are still loopholes. Due to keyloggers and other forms of malicious activity, a way of authentication needs to be developed to work around these shortcomings.

1.4 Solution

The solution to the problem is a separate application allowing the user to have a unique key, linked only to their person device that other people cannot log on to. Once they have a display of various items only the user knows, as opposed to an exact key that someone else could visually see and obtain. To keep this secure, the combination of numbers changes on regular intervals, to ensure that if security was compromised, it gets a regular change to reduce those threats. The user then inputs the proper key without using a typing combination or other ways that malicious personnel can see.

1.5 User Profile

1.5.1 Application Title

T4 Authentication

1.5.2 Potential Users

Any user who authenticates to a site remotely, including any company whose employees or customers will access data remotely. Designed for those whose data integrity and confidentiality demand the highest levels of security, beyond what traditional multi-factor authentication grants.

1.5.3 Experience with similar applications

Because the project utilizes a mobile application, the user must be familiar with mobile devices. Furthermore, the user must know how to successfully download and log into applications on a smartphone. Last, the user must also be able to access the Internet and remember a username and password to authenticate.

1.5.4 Software and Interface Experience

During a login attempt with the Web client the user should have the mobile application already open or, at least, ready to open to ease the process. From the mobile application, the user must locate the numbers of their pin. Finally, the user will click the corresponding boxes on the Web client display in the same order of their pin number.

1.5.5 Task Experience

Users will be assured that access to their account is secure, that only they have access to the authentication material provided to their mobile phone, and that data integrity is maintained throughout the login process.

1.5.6 Frequency of Use

After initial setup, requiring an Internet connection to login and sync the mobile device, the user will interact with the authentication system at every login attempt with the intended site.

1.5.7 Key Interface Design Requirements that the Profile Suggests

Simplistic GUI to maintain ease of use, which in turn does not hinder the user from implementing this form of multi-factor authentication. It is of utmost importance that this app is not considered a hindrance, but a small step that adds a superb level of security to user information.

2. Project Management

2.1 Budget

Figure 2: The estimated budget for which it would cost to build this application in a real world scenario. Due to T4 Authentication being developed as a senior project, the actual cost of this application in $0.

Project Budget				
No.	**Item**	**$/unit**	**Number**	**Total**
1	AWS Service	$100	1	$100
2	Visual Studio Licenses	$499	3	$1,497
3	Labor	$75	165	$12,375
			TOTAL:	$13,972

2.2 Project Milestones

Figure 3: *The deliverables set out for T4 Authentication*

Project Deliverables		
No.	**Name**	**Completed Date**
1	Completed Database Milestone	10/01/2016
2	Completed Mobile Application Milestone	11/19/2016
3	Completed Web Client Milestone	11/26/2016
4	Completed Testing Milestone	12/10/2016
5	Completed Draft Milestone	12/20/2016

2.3 Project Schedule

Figure 4: The following chart is the planned Gantt chart for the spring 2017 semester.

Task	Duration	Week 1 (1/8 - 1/14)	Week 2 (1/15 - 1/21)	Week 3 (1/22 - 1/28)	Week 4 (1/29 - 2/4)	Week5 (2/5 - 2/11)	Week 6 (2/12 - 2/18)	Week 7 (2/19 - 2/25)	Week 8 (2/26 - 3/1)	Week 9 (3/2 - 3/8)	Week 10 (3/9 - 3/15)	Week 11 (3/16 - 3/22)	Week 12 (3/23 - 3/29)	Week 13 (3/30 - 4/6)
Paperwork	**13**													
Gantt Chart	1													
Update Deliverables	1													
Misc. Admin Tasks	3													
Draft Paper	1													
Draft Poster	1													
Final Paper	1													
Final Poster	1													
Technological	**12**													
Random generation code	2													
Generation code testing	4													
Phone GUI Design	3													
Phone GUI Testing	2													
Database integration	3													
Database population	2													
Database testing	3													
Database injection testing	3													
Pen Testing	6													
Beta Testing	4													
UI Cleanup	3													
3rd Party Code Review	6													

2.3.1 Fall Semester Deliverables

The following is an outline of the deliverables for the fall semester. These deliverables are elements that need to be accomplished in order for the application to work properly, and be completed to the specifications required.

1. Web Client
    a. GUI
    b. Value generator
        i. Block out numbers
        ii. Accept clickable values
    c. Ability to login/User Authentication
    d. Database communication
2. Phone App
    a. GUI
    b. Pattern generator
    c. Secure logon/biometrics
    d. Ability to use on multiple platforms

3. Security
    a. Pen Testing
        i. SQL Injection through Web client
        ii. Password attacks on mobile application
    b. Secure Data transfer
        i. Security auditing to check that data is being sent
    c. Input control measures
4. Database
    a. User credentials

3. Technical Elements

3.1 Database

T4 Authentication will be using DynamoDB provided by Amazon Web Services (AWS) for our database. AWS fully manages and provides all of the technical aspects of the database. On our end, we only need to control the information passed to and from the database. DynamoDB is expandable through AWS, so storage may be purchased as necessary. AWS also provides a backup at all times, to refrain from a server going down or crashing, resulting in loss of data.

3.2 Software

The development team will be using Microsoft Visual Studio 2015 with Xamarin for the creation of both the Web client as well as the mobile application. The mobile application will be shared to the appropriate operating systems using the multi-platform program Xamarin which compiles for native performance on android and iOS through the C# language.

4. Application

4.1 Web Client

The Web client handles input to verify the user as well as communication with the DynamoDB database. This will be the buffer between end users and Website they are attempting to access.

4.1.3 Web Client UI - Main page

This is the first page the user will see on the Web client. Request username and password in order to identify the user. The Web Client will then communicate this input with the server in preparation for user authentication.

4.1.3 Web Client UI - Pattern view

This page will display an empty pattern of clickable boxes. This will prevent anything on the local machine from attempting to track the user inputs. Once the user selects the boxes in the corresponding order their personal pin is displayed on the phone app, access will be granted to the user to continue on to the main site.

4.2 Mobile Application

The mobile application handles the user account creation, and generates the digit filled pattern for the user to use for authentication. This is the pattern the user will reference when accessing the Web client sign on page.

4.2.1 Phone App UI - Main screen view

The login page is the initial page the user sees when using the mobile application. This page will have a request for the user to input a username and password. Once the user authenticates they will then be directed to the Pattern View.

4.2.2 Phone App UI - Pattern view

Upon successful login, the user will be redirected to the phone pattern screen. Here the user will see a generated pattern filled with numbers 0 through 9. This is the pattern that the user will reference in pairing with their personally known pin when attempting to authenticate.

4.3 Database

DynamoDB will store all user credentials and accounts using the application. This information will be used to authenticate the user and collect any additional metadata the client may need.

5. Testing

5.1 Overview

This section will explain both routine and specific testing methods for T4 Authentication. Testing will be used on the database, the web client and the mobile application (Android).

5.2 Scope

The scope of testing covers the operation and security of T4 Authentication's Android, Web server, and database portions. The test will be organized based on requirements of the application.

5.3 Objective

To ensure that the application works as expected, on all levels, individually and together. Security tests will be performed to verify the application is secure on all fronts. Developers will run the unit tests for their components. Security testers will run tests over the entire project.

5.4 Software Tests

Software tests will be implemented to ensure the quality of the application throughout additions to the codebase.

5.4.1 Unit Testing

Unit tests will be programmed as needed to verify the functionality of specific code. Unit tests will be designed in a way that allows them to execute swiftly and presented in a way that is simple to understand for someone unfamiliar with the code.

5.4.1 Installation and Compatibility

Installation tests will occur on various mobile devices of differing Android version to ensure device compatibility. Android Developer lists the most used Android operating systems as Marshmallow, Lollipop, KitKat, and Jellybean. The most up to date OS is Nougat and should be included as well.

5.4.3 Systems Testing

5.4.3.1Account Creation and Log-in

Creation of an account must be tested on both Android and the Web server application. This will involve a transfer of data from the mobile device to the Web application and its database. The account information for the user must also be stored in the mobile device. Once a user has successfully logged-in for the first time, the user should be able to access the application without logging-in again.

5.4.3.2 Random Pattern Generation

The random pattern generation should be confirmed to match on the mobile and Web application at several different intervals to ensure the application is working as intended. A matching pattern will be verified by demonstrating that only the pattern presented on the mobile application will result in a successful log-in on the Web application.

5.5 Security Tests

Security tests are designed as though an attacker has full knowledge of the T4 Authentication system in order to create an all-encompassing testing environment.

5.5.1 Web Application

Part one of the security testing of T4 Authentication will be based around the server and Web GUI.

5.5.1.1 Injection

This test will focus on gaining information from the server through the web client via code injections.

5.5.1.2 Brute Force

This test will focus on attempts to gain a false positive authenticated session by applying RNG to the sign on page.

5.5.2 Mobile

Part two of the security testing for T4 Authentication will be based around the phone application on every user's local device.

5.5.2.1 Man-in-the-Middle

This test will focus on the account creation portion of the phone application and will test whether or not an attacker could steal user credentials.

5.5.2.2 Application Data Storage

This testing will involve searching the locally stored content for the T4 Authentication application on the user's device in an attempt to find any confidential information on the user that may be used to gain access through the Web application.

5.2 Schedule of Testing

The schedule of application testing followed the schedule on figure 4. Phone GUI testing was completed on February 11th, 2017. Database testing was completed on March 1st, 2017. Database injection testing was completed on March 15th, 2017. Pen testing and beta testing occurred simultaneously and were completed on March 29th, 2017.

6. Conclusion

6.1 Fall Semester 2016

In the fall semester of 2016, the T4 Authentication group began with creating a storyboard and initial design of the app. This allowed them to have a concrete understanding of the purpose, direction, and teamwork for the project. From there, the group created and initialized the database using AWS, as well as began with the actual coding and functional use of both the mobile application and Web client.

6.2 Spring Semester 2017

In the spring semester T4 authentication focused on UI cleanup as well as functional and security testing of the application. We begin with preliminary alpha/beta testing, which included various types of penetration testing as well as security auditing, ensuring that all code and data integrity is maintained in order to guarantee the user is authenticated.

6. References