E-ZeePass: A web-based username and password hash

Cadrian Chan, cadrianchan@gmail.com and **Liwen Zhang**, lz6@cec.wustl.edu (A project report written under the guidance of Prof. Raj Jain)

Download 

Abstract

Internet users use different login passwords to prevent against the vulnerability of having one password for all existing accounts. Managing multiple passwords is often troublesome and even insecure for users who do not handle their passwords well. E-ZeePass is a web based username and password hash service. It facilitates users by requiring them to remember a single master password, which generates different secure passwords for different login accounts by secure hashing algorithms. Passwords generated for different web addresses will be unique, while the ones generated within same domain will be identical for each login site. The goal of this project is to provide a convenient password management system for users even when they are communicating across public networks. Users can be assured of the security of our website by the Secure Sockets Layer (SSL) cryptographic protocol.

Keywords

Password management, hash, encryption, security, MD5, SHA, web application, tools

Description

A web based password management system facilitating users by requiring them to remember a single master password, which generates different secure passwords for different login accounts by secure hashing algorithms.

Table of Contents

- 1 Introduction
  - 1.1 Theory
  - 1.2 Hash Algorithm
  - 1.3 Project Features
  - 1.4 Password Concepts
- 2 Project Organization
  - 2.1 Software Process Model
  - 2.2 Environment and Tools
- 3 Project Management Plan
  - 3.1 Tasks
  - 3.2 Milestone
  - 3.3 Resource Needed
  - 3.4 Users' Concerns and Solutions
  - 3.5 Timeline
- 4 User Manual
  - 4.1 User Requirements
  - 4.2 Procedures
- 5 Summary and Future Improvements
  - 5.1 Summary
  - 5.2 Future Improvements
  - 5.3 Related Products
- References
- User Manual and Source Code
- Acronyms

1 Introduction

Internet users often have different login passwords to prevent against the vulnerability of having one password for all existing accounts. This is recommended because when a user's password from a less secured site is compromised, other accounts will be in danger of being compromised as well if all the login passwords are the same. However, writing down your passwords (let alone different usernames) or encrypting them in some "secure" files is not very secure, and people try to recite the passwords in their minds and forget about them later. Some open new accounts every time they revisit the site thus wasting resources. Putting a file of passwords in a computer is also inconvenient when you are using public computers away from your own machine. As a solution, we provide you with E-ZeePass, a web based username and password hash. A user only needs to remember one single password, and paste the address of the login page to our program. We provide a hashed password from the two inputs, and the hashed value is different every time with different login addresses. Passwords generated for the same domain will be identical for each login site. In addition, users have the capability to choose the length of

the passwords they desire. The goal of this project is to provide a convenient password management system for users even when they are using public computers.

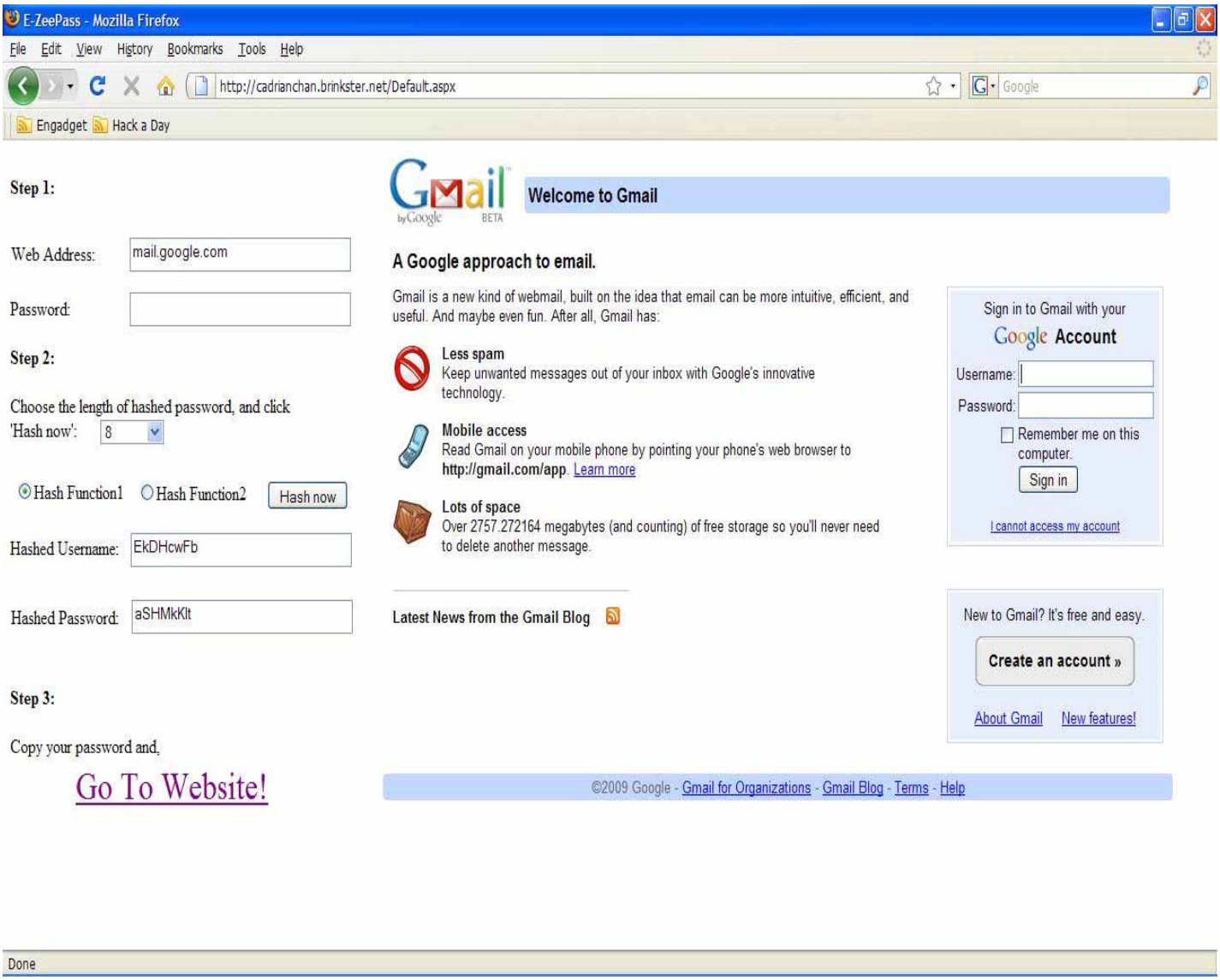

Figure 1 Logging in user account using E-ZeePass

1.1 Theory

The main propellant of our project is the secure hash algorithms that are used to generate our hashed passwords and user names. We have implemented 7 hash algorithms, including 2 which are our own, and 5 which are modified from industry standards: Message-Digest algorithm 5 (MD5), Secure Hash Algorithm (SHA-1, SHA-256, SHA-384, SHA-512). In this class project, we have decided to use four of these algorithms, two for both the hashed passwords and hashed usernames. They are MD5, SHA-1, SHA-384 and SHA-512. The four chosen algorithms are all industry standards, so users can rest assured the provided hashed outputs are safe and non-reversible.

There are two inputs to each hash algorithm: the desired login address and the master password. We concatenate these two inputs, which gets fed into the chosen algorithm. The output is a non-reversible hashed value. From this value, we extract the output hash for the user according to their desired length of password characters. As we mentioned above, the user can choose their desired hash algorithm, and the hashed password will be available to the user with an extra option of a hashed user name.

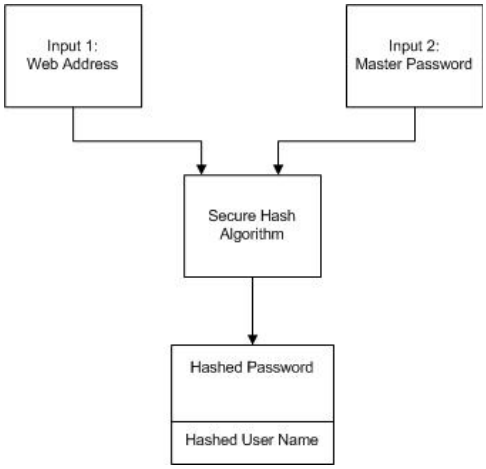

Figure 2 Flow diagram of hashed password generation

In addition, for the user's convenience, we have a function which lets a user navigate to his desired login website within our iframe. This is made possible because the domain name of the web address input is recorded while the user enters his desired login address.

1.2 Hash Algorithms

A hash algorithm is a function that takes inputs and converts them into a fix sized bit string, which is called the cryptographic hash value or message digest. There are several properties of hash algorithms which make it ideal to use with our service [Kaufmann02]:

1) It is easy to compute the hash value from any given inputs.

2) It is a one-way function which is not practical to figure out what input corresponds to a given output.

3) It is computationally infeasible to find a message that has a given pre-specified message digest.

4) It is computationally infeasible to find two messages that have the same message digest.

The four hash algorithms chosen for implementation in our website are all industry standards, and are cryptographically secure. In addition, through the use of hash algorithms, the output hash passwords generated are random enough for security concerns.

1.3 Project Features

The project will be implemented using Microsoft Active Server Pages .NET (ASP.NET) technology with the Visual Basic .NET (VB.NET) language.

Basic Functionalities

- User remembers one master password for logging into different websites using different hashed passwords.
- The per-website password is the output of a secure hash algorithm, using the master password and the web address as inputs.
- Passwords generated for the same domain will be identical for each login site.
- Portability: web based service allows users to access our service on-the-go.
- Variable password length.
- Security: Hashed passwords are randomly generated to provide maximum security.
- Extra option of giving user a 2nd hash value as the username.
- Convenience: User can navigate to their login accounts using our iframe design and copy & paste their password easily.
- Protection against keylogger: users are not required to type in password by hand.

Advanced functionalities

- Compatibility with mobile phone/device.
- Logging on to websites automatically.
- Security and data integrity provided by implementing SSL certificates.

These advanced features are not part of the class project due to money/time constraints, but will be featured in the commercial version.

1.4 Password Concepts

Over time, passwords may be compromised in many ways. E-ZeePass recommends users to change their master password regularly, at most every 90 days. For the same reasons, users should not reuse old passwords, as they may already have been compromised. We recommend users to use at least two different passwords when using this service, for different level of security importance of their login accounts.

When a user changes his master password, all the per-website passwords will be updated to new ones, since the master password is one of the inputs of the hash algorithms. This is one of the conveniences of using our password management system.

---

2 Project Organization

Our group consists of two members: Cadrian Chan and Liwen Zhang. Our plan was to implement our project according to the following model and techniques:

2.1 Software Process Model

We have implemented our design according to a sequential development process, which consists of the following milestones:

1) Requirement Specification

- The environment and required software were identified.

2) Design

- The design of our web application, architecture, algorithms and web site were chosen.

3) Implementation

- Our code was written in the ASP.NET environment using the Visual Basic language.

4) Integration

- Code and design from the two members was integrated. Documents were tracked using a simple wiki.

5) Testing and debugging

- Extensive testing was done inside the .NET framework, using the built in server from Visual Studio 2005. Additional testing was done using different browsers and Operating Systems.

6) Deployment

- The code was migrated to a free ASP.NET hosting service for further testing.

7) Maintenance

- Feedback was gathered from professor, TA, classmates and friends. Updates were done according the opinions gathered.

2.2 Environment and Tools

The following environment and tools were used during the implementation of our project:

Microsoft Visual Studio 2005:

- Microsoft Visual Studio 2005 Professional Edition is a complete environment for individual developers building Microsoft Windows, Web, or mobile solutions.

- Users are able to easily create and deploy client applications. Automatically publish and maintain applications and their dependencies with integrated ClickOnce support [Microsoft09].

Microsoft .NET Framework Version 2.0.50727

- The Microsoft .NET Framework is a software framework available with several Microsoft Windows operating systems. It includes a large library of coded solutions to prevent common programming problems and a virtual machine that manages the execution of programs written specifically for the framework. The .NET Framework is a key Microsoft offering and is intended to be used by most new applications created for the Windows platform [Wikipedia09].

ASP.NET

- We used ASP.NET as a web application framework to assist us in building the web application, web form and services.

Visual Basic

- Visual Basic is an active scripting language which is designed to provide functionalities for easy web form and application design. Our internal hash algorithms are designed in Visual Basic, including the two hash algorithms designed by our own members.

Dreamweaver

- Dreamweaver was used to design the appearance of our main webpage, including the designing of our logo.

---

3 Project Management Plan

Here is the breakdown of our workflow for the project:

3.1 Tasks

- Decide what kind of programming language and integrated development environment (IDE) to use for the project. Architecture of the web application (database VS stateless).

- Select hash functions (our own design VS industrial standards). Studies on the properties of different hash algorithms and their security.

- Write program using ASP.NET in Visual Studio 2005. Develop a prototype for our web application with basic functionalities.

- Integrate separately developed code by our members. Graphical user interface (GUI) for users to enter their inputs added to website.

- Test and debug all functionalities and combinations of the application.

- Deploy our code from .NET framework to actual website.

- Gather ideas from professor and friends who tried using the actual website.

- Update application according to user opinions.

- Release of final version.

3.2 Milestone

Table 1 Milestone

Milestone	**Status**
Brainstorming	Completed
Building prototype	Completed
Integration & GUI for users	Completed
Beta version in Visual Studio with built-in server through .NET framework	Completed
Deployment to actual website	Completed
User review, update and release of final version	Completed

3.3 Resources needed for development

Software resources:

Microsoft Visual Studio 2005

.NET framework 2.0

Dreamweaver

Windows 98 or newer OS

Browser that supports asp service

Hardware resources:

Pentium III or newer CPU

512MB RAM or above

1GB hard disk space

3.4 User's concerns and solutions

The following concerns are gotten from surveying users and peer reviews from class:

Q: What if an attacker tries to intercept your packet while the server sends you the hashed password?

Ans: This project is a solution for password management, rather than solving the problem of establishing a secure connection. We can solve this problem by adding a SSL/Transport Layer Security (TLS) certificate on our web site if we are going to commercialize our project (since adding the certificate requires payment for the service). We can also improve the security by using browser side scripting like JavaScript. If the user's internet connection is insecure, his password might be intercepted during the login anyway. (Although websites such as Bank of America has SSL certificate, majority of other websites like Hotmail and Facebook etc do not).

Q: Is your service going to record my entered master password/domain info (and use or sell it later)?

Ans: Our service does not log user data of any kind, and this can be verified from the source code that we have uploaded with this document. We agree there is a need to address this issue on our website and convince users on our policy if we want to make it a popular service.

Q: Some login sites require special character mixes?

Ans: Currently we let the users choose the length of their hashed passwords, and options for special character requirements will be a future improvement. On the side note, special characters generated by our hash algorithms, Eg. ~!@#$%^&* etc. are accepted by most common email accounts like Gmail and Yahoo! Mail, and generating "regular" password characters will be added as an extra option.

3.5 Timeline

Table 2 Timeline

Milestone	**Deadline**	**Status**
Project outline and evaluation	3/2	Completed
Prototype	3/15	Completed
Integration and GUI	3/20	Completed
Beta version in Visual Studio with built-in server through .NET framework	3/25	Completed
Deployment to actual website	3/27	Completed
User review, update and Finalize code	4/4	Completed

---

4 User Manual

The user manual walks a user through all the necessary steps, procedures and requirements for using the E-ZeePass service.

4.1 User Requirements

ASP compatible browsers, such as Internet Explorer, Firefox, Chrome.

E-ZeePass is a web-based application. The required software and hardware specifications are minimal.

4.2 Procedures

Step 1)

a) Go to the E-ZeePass website from your favourite browser.
b) Enter the web address of your login account. It will be automatically converted to its domain name.

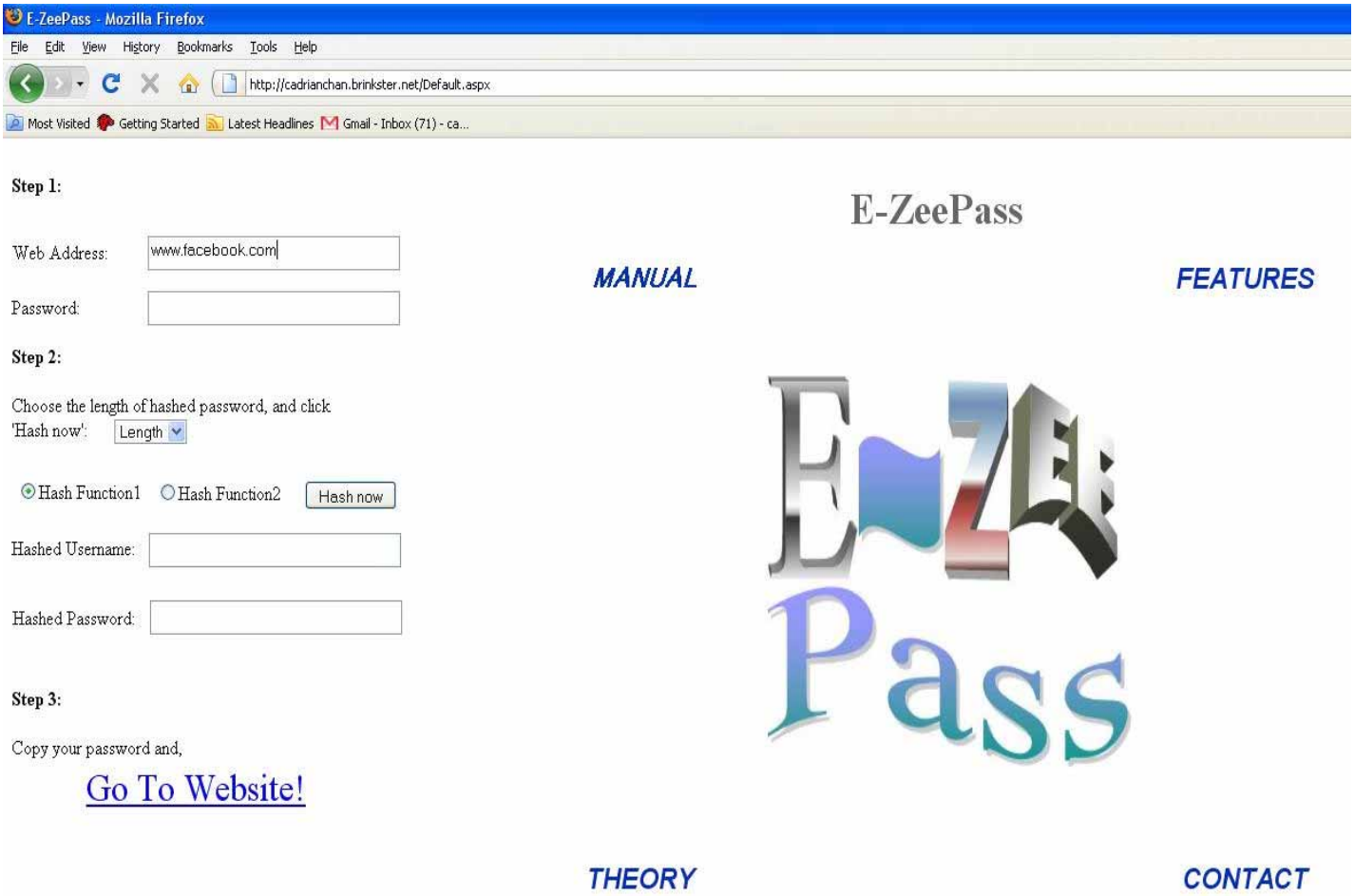

Figure 3 Entering web address

c) The following example shows the address www.facebook.com is converted to facebook.com

d) Create and enter your master password. This is the single password you need to remember from now on.

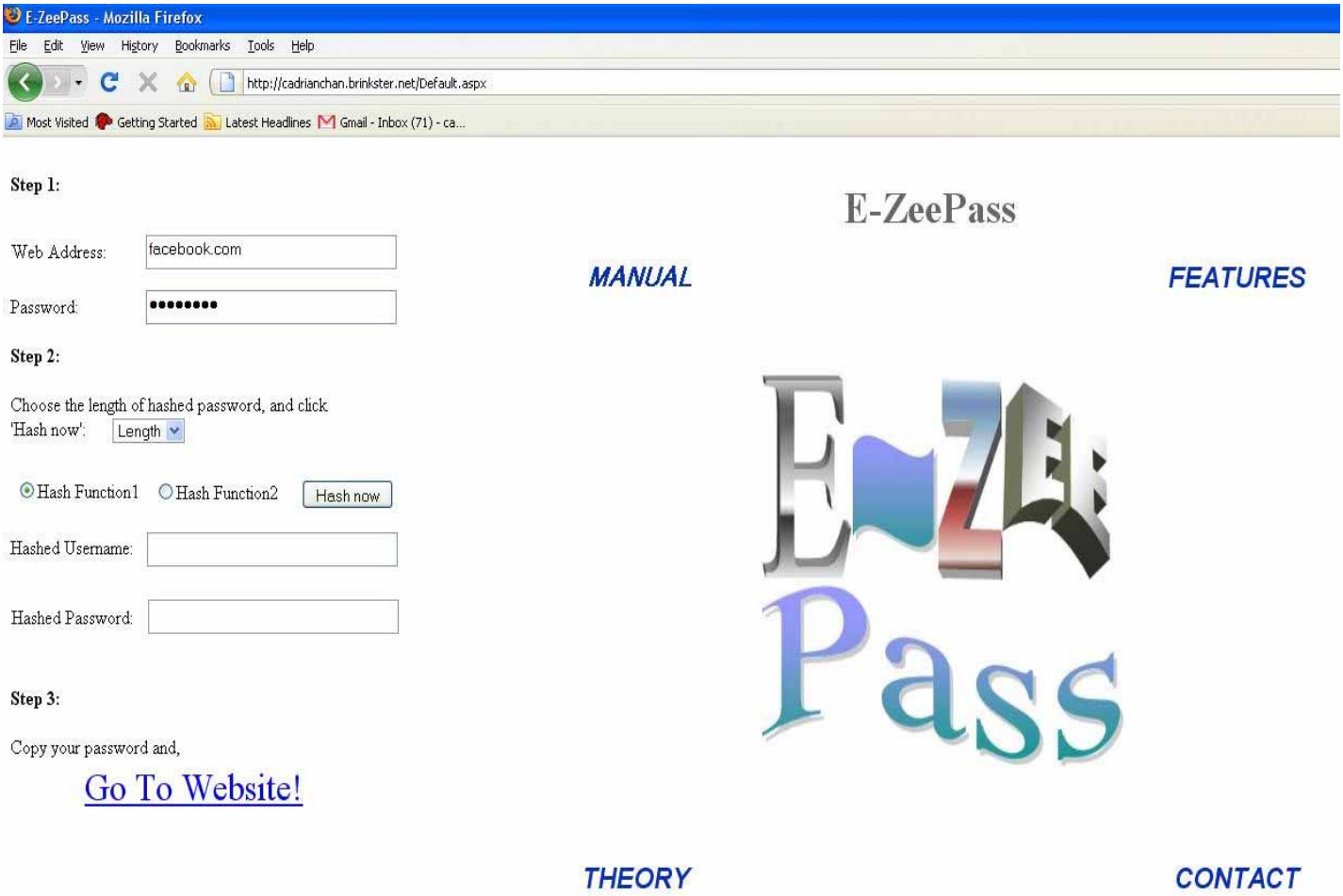

Figure 4 Entering password

Step 2)

a) Variable password length: Use the dropdown list to choose the desired length of your hashed password, such as "10" in the following example.

b) Select your hash function - "Hash Function 1" uses MD5 and "Hash Function 2" uses SHA-512. The actual algorithm is not displayed for security reasons.

c) Click the "Hash now" button, and the hashed username and password will be generated as shown:

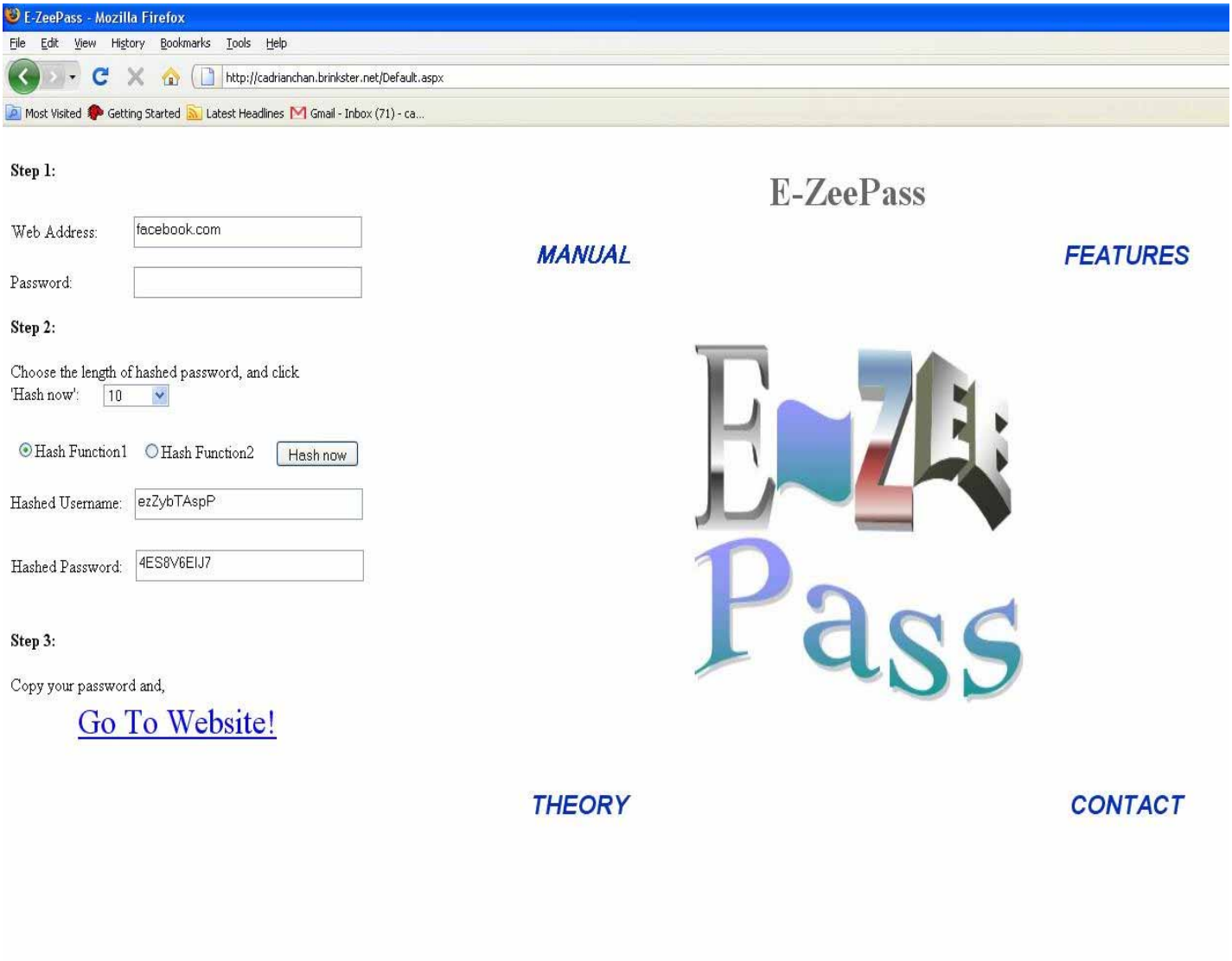

Figure 5 Generating hashed password and username

Step 3)

a) Click on "Go to Website!" to navigate to your desired login page using the frame on the right hand side.

b) Copy and paste your password into the corresponding field of the login site:

Figure 6 Navigating to login page

That is all there is to using E-ZeePass! It is meant to be a quick and easy way for users to manage their passwords. User can navigate their login account within the iframe, or in a new tab/window if they prefer.

---

5 Summary and Future Improvements

In this section we will talk about the summary of our findings and achievements, and also our future work planned to improve our service based on user opinions.

5.1 Summary

Infosecurity professionals continually fight against evolving threats, but one problem that just won't go away is the vulnerabilities that arise through using passwords for authentication. In this project, we hope to provide a simple and elegant password management system for the public. There is similar software out there in the market, but they seem to lack portability when the user is away from his own machine. Our main objective is to provide mobile users a quick and reliable web based solution on-the-go when they are on a public network.

5.2 Future Improvements

The following advanced features are not part of the class project due to money/time constraints, but will be featured in the commercial version:

- Compatibility with mobile phone/device will be implemented, eg. Some mobile phone browsers do not support the copy and paste function.

- SSL certificates will be added to our website to provide security and data integrity when we publish the website in public, since this feature requires payment for the service, and our own domain name.

- Options for special password character requirements.

- Automatic login of websites.

5.3 Related Products

As mentioned above, there is similar software in the market related to password management, with some of the popular ones being RoboForm [RoboForm09] and KeePass [KeePass09]. E-ZeePass is different from these products in that it provides a portable and simple to use solution which can be accessed everywhere through internet. The use of cryptographic hash algorithms is another major difference between E-Ze?Pass and the above products. The current password management programs in the market generally use cryptographic algorithms for encrypting stored passwords, and generate randomized passwords for users. E-ZeePass make use of hash algorithms to achieve both of these features.

As we are not as established as the above products, the biggest obstacle we face are gaining trust from the users and assuring them their password information is not being logged. We plan to include this paper, related documents and source code in the E-ZeePass website to convince users regarding this issue.

References

[Kaufmann02] Kaufmann, Charlie et al. "Network Security, Private Communication in a Public World, 2nd Edition" Upper Saddle River, N.J. : Prentice Hall 2002. pp. 117

[Microsoft09] Microsoft Faculty Connection. "Visual Studio 2005 Professional (English Version)" 2009
http://www.microsoft.com/education/facultyconnection/software/softwaredetails.aspx?cid=1&c1=en-us&c2=0

[Wikipedia09] Wikepedia. ".NET Framework" 2009
http://en.wikipedia.org/wiki/.NET_Framework

[RoboForm09] RoboForm "Product Features" 2009
http://www.roboform.com/

[KeePass09] KeePass "KeePass Password Safe" 2009
http://keepass.info/

User Manual and Source Code

User Manual (doc)
Source Code (zip)

Acronyms

ASP	Active Server Pages
GUI	Graphical User Interface
IDE	Integrated Development Environment
MD5	Message-Digest algorithm 5
SHA	Secure Hash Algorithm
SSL	Secure Sockets Layer
TLS	Transport Layer Security
VB	Visual Basic
XML	Extensible Markup Language
XSLT	Extensible Stylesheet Language Transformation

Last Modified: April 19, 2009
This and other papers on latest advances in network security are available on line at http://www.cse.wustl.edu/~jain/cse571-09/index.html
 Back to Raj Jain's Home Page