Andrew Hamilton
5 Period

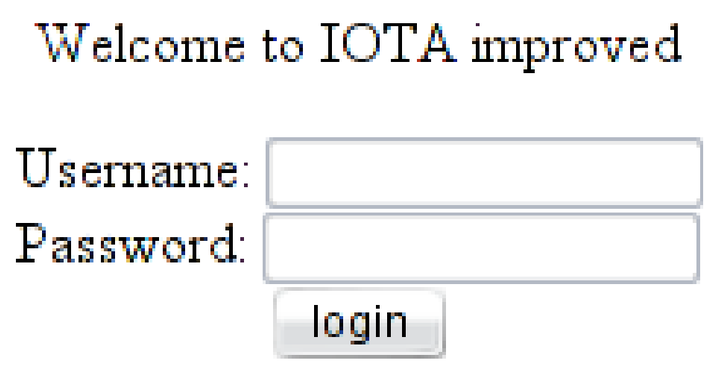
IOTA Improved

Design and Implementation of a Modular and Extensible Website Framework

Abstract

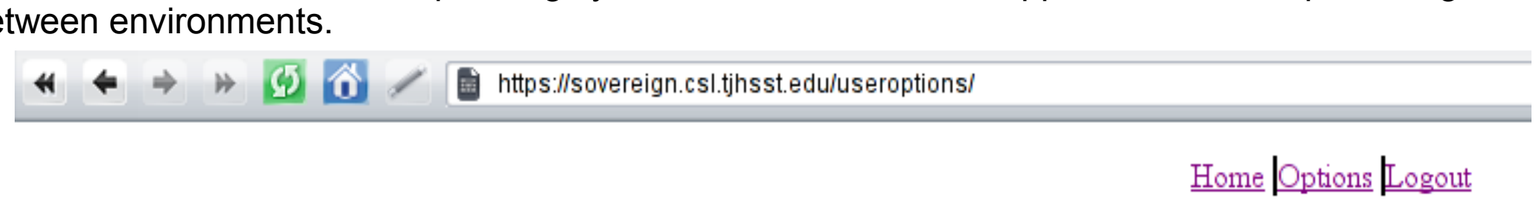
The web today is fast becoming what the phone book used to be, except on a global scale. Almost every company is expected to have a website, ranging from a few words on the homepage to massive and dynamic websites like that of Intel (intel.com). However, there is a problem, many websites, especially those run by smaller companies or individuals, suffer from numerous design and security flaws due to the complexity of designing a functional and fashionable website today. Blackboard, FCPS's Course Management System, is one such example of an application that is sorely out of date with regard to security standards as well as functionality and design. The goal of this project is to develop a basic web design framework that will handle the basic security and design concerns while being easily extensible with modules in order to be customized for almost any purpose.

Keywords: software design, software engineering, application, web-based



Introduction

For many people today, the Internet is the first place they go for information and many companies are trying to make as much information available online as possible. However many websites today suffer from a lack of common-sense security measures as well as basic accessibility standards. The goal of this project is to provide a framework with which to build such sites which will help the end user to easily build their site through the use of simple add-in modules while transparently handling the security concerns. The ideal web framework would be easy and intuitive to use, extensible, and secure, the last becoming more important every day. I will describe the design of an alternative, web-based, extensible web framework running on a standard LAMP (Linux, Apache, MySQL, PHP) server. There are many choices that can be made in web applications including language (PHP, ASP, Ruby, Python) and data storage system (LDAP, Oracle, MySQL, Postgre SQL). All have their advantages and disadvantages. PHP is the current standard for web pages although Ruby and Python are growing. ASP is a proprietary Microsoft language which makes it less than ideal for web site development. The current database standard is MySQL, though the others are popular for various purposes. PHP and MySQL are available for both Windows and UNIX operating systems which means that applications developed using them are relatively easy to transport between environments.



Background

Many websites today are designed to "look cool" incorporating lots of Javascript and flash in order to try and "stand out" in the crowded Internet Unfortunately, these sites often have security and accessibility problems that are in many cases left unfixed in the name of flashiness. For instances, Intel's site looks incredibly cool with the animated menus on top of a background image, however, the colors can sometimes be hard to read and separating the image from the menus can be difficult for some people. Also, traversing the site with a screen reader takes much longer because the textual representation of the flash is incredibly lengthy. Another example, Sun Microsystems, implements a login system wherein your user-name and password (entered on one page) are passed to the next page through the URL such that your password can be seen by anyone looking at the computer and is saved in your browser's history in clear text. This enables anyone who knows you have a sun account to check your history and easily determine the password. This is a bug that is easy to fix but which someone has thus far apparently not considered.

3 Testing and Analysis

Currently I have Apache2 with mod ssl and mod rewrite, PHP5, and MySQL5 running on a Debian Linux server in the lab. This is all that my project requires to run. The currently working version of my code presents a login page and denies access to the site without a valid login. Once a user logs in, their IP and browser agent are recorded for security, and they are given a PHP session which is used throughout the site. Before allowing them to access each page, the kernel module checks that their IP and browser agent are the same (to protect against session hijacking) and that they have not been idle for a long time (to protect users who forget to logout). All of these features work with multiple accounts and the main page currently uses information from the database to display the user's name and "rank" (teacher, student, or administrator).

Most of the framework is now complete and so I am working on building a set of modules to enable this framework to be used for the TJ cubesat project website. As part of this project, I have been working on a python middleman script since their website needs to talk to an amateur radio and PHP itself has no serial functions (because it's a web scripting language).