Forgot Password Cheat Sheet

From OWASP

Contents

Introduction

This article provides a simple model to follow when implementing a "forgot password" web application feature.

The Problem

There is no industry standard for implementing a Forgot Password feature. The result is that you see applications forcing users to jump through myriad hoops involving emails, special URLs, temporary passwords, personal security questions, and so on. With some applications you can recover your existing password. In others you have to reset it to a new value.

The recommendations presented here for implementing Forgot Password are most appropriate for organizations that have a business relationship with users. Web applications that target the general public (social networking, free email sites, etc.) are fundamentally different and some concepts presented may not be feasible in those situations.

Steps

Step 1) Gather Identity Data

The first page of a secure Forgot Password feature asks the user for multiple pieces of hard data. A single HTML form should be used for all of the inputs.

A minimum of three inputs is recommended, but the more you require, the more secure it will be. One of the inputs, preferably listed first, should be the username. Others can be selected depending on the nature of the data available to the application. Examples include:

- email address
- last name
- date of birth
- account number
- customer number
- social security number
- zip code for address on file
- street number for address on file

Step 2) Verify Security Questions

After the form on Step 1 is submitted, the application verifies that each piece of data is correct for the given username. If anything is incorrect, or if the username is not recognized, the second page displays a generic error message such as “Sorry, invalid data”. If all submitted data is correct, Step 2 should display at least two of the user’s pre-established personal security questions, along with input fields for the answers. It’s important that the answer fields are part of a single HTML form.

Do not provide a drop-down list for the user to select the questions he wants to answer. Avoid sending the username as a parameter (hidden or otherwise) when the form on this page is submitted. The username should be stored in the server-side session where it can be retrieved as needed.

Step 3) Send a Token Over a Side-Channel

After step 2, email or SMS the user a randomly-generated code having 8 or more characters. This introduces an “out of band” communication channel and would be extremely tough for a hacker to overcome. If the bad guy has somehow managed to successfully get past steps 1 and 2, he is unlikely to have

compromised the side channel.

Step 4) Allow user to change password

Step 4 requires input of the code sent in step 3 and allows the user to reset his password. Display a simple HTML form with one input field for the code, one for the new password, and one to confirm the new password. Verify the correct code is provided and be sure to enforce all password complexity requirements that exist in other areas of the application. As before, avoid sending the username as a parameter when the form is submitted. Finally, it's critical to have a check to prevent a user from accessing this last step without first completing steps 1 and 2 correctly. Otherwise, a forced browsing attack may be possible.

Related Articles

FishNet Security White Paper - Best Practices for a Secure "Forgot Password" Feature (http://www.fishnetsecurity.com/Resource_/PageResource/White_Papers/FishNetSecurity_SecureForgotPassword.pdf)

Other Articles in the OWASP Prevention Cheat Sheet Series

- Authentication Cheat Sheet
- Cross-Site Request Forgery (CSRF) Prevention Cheat Sheet
- **Forgot Password Cheat Sheet**
- Cryptographic Storage Cheat Sheet
- SQL Injection Prevention Cheat Sheet
- Transport Layer Protection Cheat Sheet
- XSS (Cross Site Scripting) Prevention Cheat Sheet
- DOM based XSS Prevention Cheat Sheet

Authors and Primary Editors

Dave Ferguson - Dave.Ferguson[at]fishnetsecurity com
Jim Manico - jim[at]owasp.org

Categories: Cheatsheets | OWASP Document