Softwaretest and Security Assignment

2. januar 2024 13:24

Forfatter: Mads Søndergaard
Fag: Softwaretest og Sikkerhed

PRODUCT OWNER KRAV:
Jeg ønsker en C#.NET server-side web app med høj sikkerhed som skal " deploy" på vores Linux setup.Jeg vil gerne have en 2 faktor bruger registrering OG login hvor bruger registrering og login info gemmes i en bruger database. Forresten skal login implementerer en stærk password krav. Vores firma har en SSL certifikat som applikationen kan bruges.Til at starte med, ønsker vi til den første sprint, at applikationen også kan opret en simpel tekst fil på serveren så vi kan gå i gang med at afprøve vores installeret Linux setup.

Product Backlog:

Requirement	Priority
C# .NET Server-side web app	1
Configure User database compatible with Linux	2
Design user registration interface	3
Implement two-factor authentication.	4
Implement strong password requirements.	4
Implement password validation.	4
Create pages (2, 3, 4) for testing access/authentication	5
Implement roles to allow for administrator	5
Implement SSL certificate capability.	6
Make the server capable of storing a textfile.	6
Store file locally for user and generally on server.	7

Sprint Backlog:

Story:	Sprint	Requirement	Priority
Customer wants a C#.NET server-side Web APP	1	Configure C# .NET server-side web (Blazor)	1
Customer wants to deploy it on Linux.		Configure User database that is linux compatible	1
Customer wants the ability to generate a textfile on the server.		Implement ability to generate textfile (access_log.txt) on the server.	2
Customer wants 2FA registration	2	Implement 2FA (Authenticator) on registration	3
Customer wants 2FA login		Implement 2FA (Authenticator) on login	4
User accomplishes this 2FA via QR Code		Add QR code capability to allow for easier 2FA	4
Customer wants to implement strong password		Implement strong password requirements	4
		Create roles to allow for administration	4
		Create pages to test access	5
Customer wants to store user info in Database		Save user info in database (SQLite)	5
Customer has an SSL Certificate for use.		Add SSL capability.	6
Customer wants a local-specific version of the textfile (access_log.txt) stored in the users own personal folder(s), for both Windows and Linux	3	Add second access_log.txt which only shows local access info (the user themselves), and store in their personal folder, irrespective of OS.	7
Customer wants an overview of users that accessed Page3, if the user is Admin		Print out contents of access_log.txt on the page if user has Admin role.	8

Test Plan:

Test Cases:

Environment Testing:

- **Windows/Linux:**
  - Ensure that registration/login/file write is functional on either OS.
- **Database:**
  - Ensure new users are saved in database (can login afterward).

Functionality Testing:

- **2FA:**
  - Verify new users are automatically required to utilize 2FA
  - Verify successful login with valid credentials and 2FA.
- **Password:**
  - Validate password requirements during registration (length, complexity).
    - 8 length, upper and lower case, digit

Usability Testing:

- **User Registration/Login:**
  - Test registration process flow.
  - Validate login flow; check if 2FA setup is intuitive for users.
- **Page Navigation and Access:**
  - Navigate to Page2 without login; verify redirection to login page.
  - Attempt to access Page4 without Admin role; ensure access denial/block.

Conclusion:

- Registration should be operational, and 2FA should be applied automatically and required during first registration.
  - Password requirments shold be set to strong (8 minimum length, including upper and lower case, and digit)
- Login - using 2FA - should be possible following Registration, thus confirming users are stored in database.
- Page2 should redirect to login if user is not logged in.
- Page3 should inform the user they are authorized or not, depending on whether or not they are logged in.
  - Page3 should show access_log in textarea, if the user has Admin role; otherwise empty.
- Page4 should be hidden for non-Admin users, visible for Admin users.
  - If accessed manually, notify non-Admin users they do not have the Admin role.
  - Will inform Admin users they have the Admin role.

Unittests:

- ComponentViewTest
  - TestViewAlone
    - Sanity test of contents of simple page (Page2) is shown as expected, regardless of authorization or authentication.

- AccessTest
  - TestNotAuthorizedCode
    - Tests if the user DOES NOT have Admin role (checked via code)
  - TestAuthenticatedCode
    - Tests if the user IS authenticated (checked via code)
  - TestViewAuthorized
    - Tests if the user is authorized to see expected content when viewing Page3.
  - TestViewNotAuthorized
    - Tests if the user who isn't authorized sees expected content when viewing Page3.
  - TestViewRoleAdmin
    - Tests that a user with the Admin role sees the expected content on Page4.

- FileTest
  - TestAccessLogWrite
    - Verifies local- and server-side logs are identical after writing when a user visits Page3.
  - TestAccessLogPrint
    - Verifies that log printed on web (Page3, shown only for Admin-role users) is identical to the one stored server-side.