pitney bowes

Connect

ICP AutoOpen

Auto-Authentication

Tech Note

v2.0 - December 2017

Tech Note Summary:

This document describes Connect's AutoOpen/AutoAuthentication Feature: a revolutionary method to open sensitive (secure) documents with the least amount of clicks/credentials interaction by the end user - yet maintain security at the highest level.

Use Cases:

This Tech Note describes Connect's way of allowing Connect statement recipients to have an "AutoAuthentication" / "AutoOpen" functionality, using the following scenarios:

End User opts in to the AutoAuthentication:

- When opening an e-document (typically clicking on the document/statement link from within the email) the first time it is received on a specific device, the end user will have the option to enroll/register to Connect AutoOpen capability by enabling the feature (e.g. selecting the "Remember me" checkbox) and successfully authenticate to view the e-document (entering correct credentials.)

End User opens a subsequent e-document on the same device:

- If the end user had been enrolled via Connect solution, he/she will not need to reauthenticate and will bypass the authentication step automatically. No further steps will be needed to view the e-document securely.

End User wants to opt out from the AutoAuthentication capabilities or disable the AutoAuthentication feature on a specific device:

- The end user will need to Turn-Off AutoOpen from the e-document currently viewed. Cookie will be revoked/expired and the user will be asked to authenticate again the next time she opens the e document.

Enterprise control center for end users AutoAuthentication:

- The enterprise admin has a control center that allows revoking (due to a security breach, users opt outs, abusive usage or any other concern) specific end user or a group of users' AutoAuthentication feature.
- The enterprise control center also allows to setup AutoAuthentication auto-expiration based on time, number of login attempts and other advanced controls.

Technical description:

Workflow:

1. A login cookie is issued/created by the Connect Server side.

2. The login cookie will contain an identifier, a token and creation time. The identifier and the token are two different GUIDs, which are totally unique between any end-point devices.
3. Both identifier and token are stored together on the Connect back-end repository and the token is hashed to adhere to a high security policy.
4. At the first time, when a non-registered/non-logged-in user visits the site and presents a login cookie, the specific identifier is looked up in the repository.
    - If the identifier exists and the hash of the token matches the hash for that identifier, the user is considered authenticated.
5. At that time a new token is generated, a new hash for the token is stored over the old record, and a new login cookie is issued to the user (at this time we take the assumption that it's okay to re-use the identifier.
    - This is to avoid a case of abuse where someone has copied the token and trying to re-use it on a different device.)
6. If the identifier is present but the token does not match, an intrusion or abuse is assumed.
    - The user receives an alert and all sessions remembered for that user are deleted/revoked. This approach provides comprehensive defense.
    - If someone manages to leak/steel the database table, it does not leave the attacker an open door for impersonating users

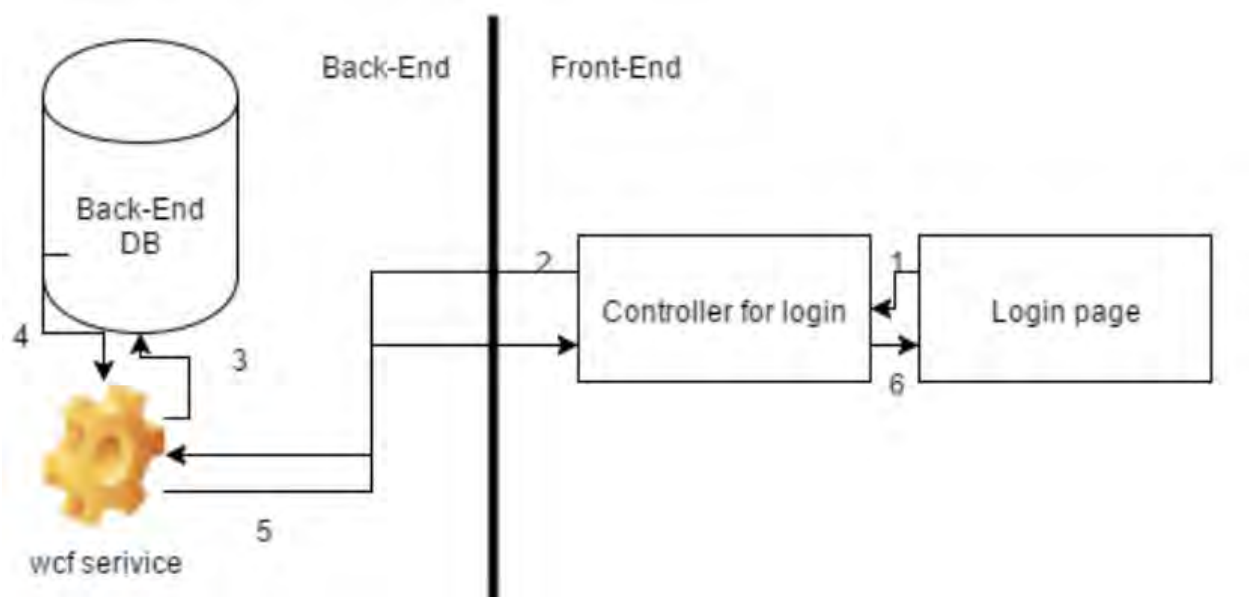
Tech Notes:

1. The hash is Pbkdf2 with SHA/512 algorithm.
2. The repository is placed under the secure LAN (e.g. back-end).
3. Optional: The cookie can be encrypted with HMACSHA512/256 (this will depend on the Enterprise policy)
4. The encryption is compliant with the United States Federal Information Processing Standards (FIPS).

The above approach gives our customers the capability to:

1. Delete any cookie, all users or a single user.
2. Know when the user is going to have the auto-open option expired and will have the option to notify him by email or SMS to "Re-connect".
3. Have tracking information on how much times the user has connect to the system.

Diagram



1. **1** User authenticates to the e-document received, and selects the “AutoOpen On” option.
2. **2** Token and identifier are generated by the controller (Connect Backend)
3. **3-4.** Token procedure
    a. In case of first login or token expiration - Token is inserted to Connect repository
    b. In case of user coming back to the system – token is validated against the current token in the repository. Following the scenario described above.
4. 5-6. Validation and success of insertion goes back to the end user for proper presentation of the process

v2.0 - December 2017