Transitioning from Passwords to Passwordless

The State of Passwords

Passwords present a significant challenge in the cybersecurity landscape today. While they are widely reviled as the least secure method of protecting accounts and assets, their very ease of use makes it very hard to transition to more secure options. WebAuthn provides a easy to use and immensely more secure authentication option then the common password, but making the jump from one to the other is far from easy, both from a technical standpoint to building a smooth user experience.

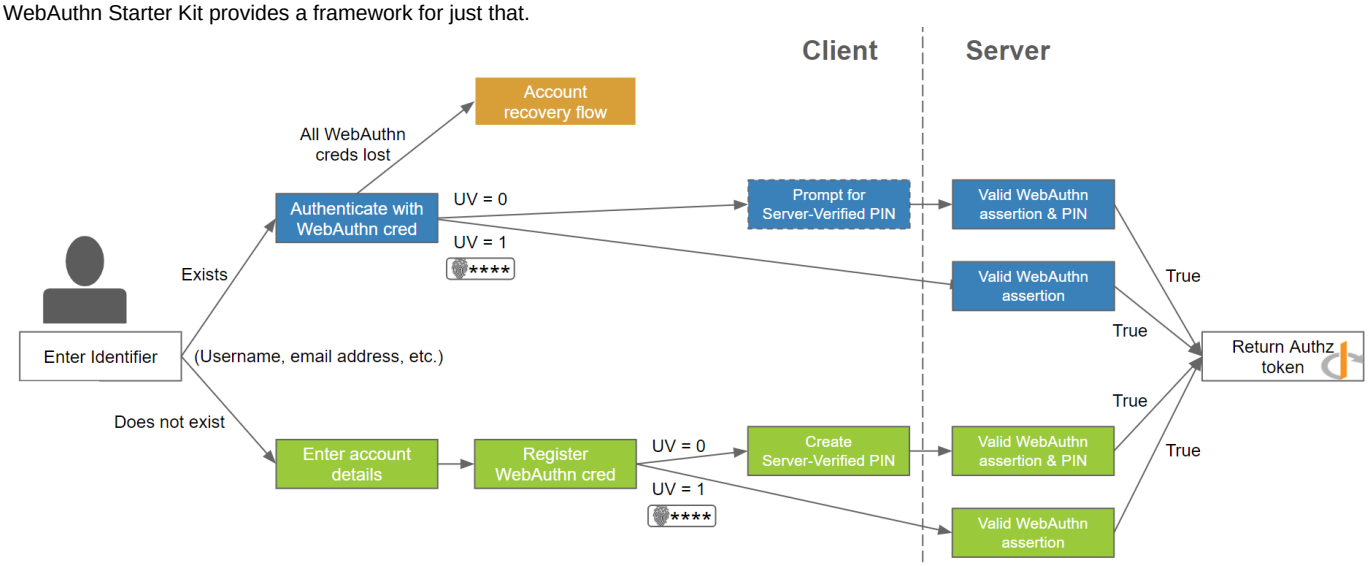
The Yubico WebAuthn Starter Kit looks to address both of those concerns. It provides a working reference to WebAuthn to address technical questions and understanding, de-mystifying the process of connecting a WebAuthn Server to existing frameworks. It also provides a working example of a transitory step from usernames and passwords to a truly passwordless and secure experience.

The FIDO2 Authenticator

At it's core, moving from a traditional password to a passwordless experience is about changing what is considered the root of authentication. A username and password system focuses on the password being the root of authentication - which is a logical step, as the password can leverage standard keyboards or other ubiquitous input methods to be provided. This universal support and ease of use has made the password hard to replace.

WebAuthn and FIDO2 address this in a number of ways. First, native support for WebAuthn is present in to a degree across all major operating systems and web browsers - the WebAuthn Stater kit has support for every significantly used combination. This allows FIDO2 authentication devices to leverage commonly supported interfaces, such as USB or NFC, to work with systems without requiring users to install software or custom clients. Further, mobile devices now can be used as authenticators without additional hardware when accessing services locally. With the widespread use of smart phones of both iOS and Android, most people have a FIDO2 authenticator even if they are not aware of it.

The last major element to overcome is the ease of use of FIDO2 vs passwords. Users can be slow to adopt new technology if they are presented with too many options - most people want to have just what is relevant to their needs presented to them. The Identity first flows used in the WebAuthn Starter Kit provides a framework for just that.



Identity First Flows

The key architectural aspect of the WebAuthn Starter kit is the Identifier first flow, where a user is only provided options which are relevant to their account and the environment they are currently accessing. This leads to very smooth user experiences customized to the endpoint they are currently accessing - a mobile device with a built in FIDO2 Authenticator may only need to enter their username then tap the fingerprint reader, while a laptop using a security key connected to it will require the user to enter a PIN and touch their device.

This automatic customization of the user's experience is possible due to the breath of support FIDO2 and WebAuthn have for dissimilar environments. In some situations where the host environment does not support more modern authentication, such as a device PIN or Biometric, the WebAuthn Starter kit provides a comparable flow with a PIN verified by the server. In doing so, a similar level of security is preserved.

In an identity first flow customized for a specific service, an architect should take into account the following:

- What is the method used by most accounts to login?
- What is the most secure method you want accounts to use to login?
- What is the method for gaining access you want to use should the primary method be unavailable?

The design of an identifier first flow can be based around allowing the user to continue to use their primary method to start, but then transition them to the more secure method as they enable it on their account - not as a secondary multifactor authentication step after the primary login, but replacing it entirely. Further logic can be applied, with different environments presenting varying options based on what is available.

For most, the most commonly used method is username and passwords, while the most secure method would be accounts secured with WebAuthn. The WebAuthn Starter kit demonstrates this flow - for more details, refer to the High Level Architecture documentation.

Transitioning from Passwords to Passwordless

An Identity first flow should also be considered a path to transition between the traditional username and password to a more secure passwordless deployment using modern authentication options.

Assuming a traditional authentication flow using username and password, the first step can be to add support for WebAuthn as an traditional login flow. Many solutions offer WebAuthn as an optional multifactor authentication option today - the difference here is based on the if a user has a FIDO2 authenticator associated with their account, that option is presented instead of the username and password. As the user is required to provide their identifier in the form of their username at the start of the process, the framework for transitioning to an identifier first flow is already in place.

Once support for WebAuthn is in place as a second factor authentication solution, the next step is to move it from an optional component to a required one in the authentication flow. With the introduction of support for WebAuthn platform authentication on desktop and mobile devices, this becomes less of a burden on the end user. For a successful transition, there are three important changes which must be made to the User flow:

- Existing users must be required to register a WebAuthn Authenticator to their account as part of the login flow.
- New users must be required to register a WebAuthn Authenticator on account creation.
- All users must have the option to add or remove additional authenticators in their account.

Passwords vs PINs & Server PINs

Moving from always requiring a password to only requesting a PIN if the user is not required to provide one to an authenticator may feel like moving to a less secure security model. In actuality, by implementing such a flow the major vulnerability of a password is removed. Instead of relying on a password to verify the identity of a user, the authenticator is. The password is only used to verify the authenticator is being used by an authorized party. The threat plane transitions from anyone who can connect to the service to just those who can physically take control of the authenticator device. This is the same model as used with debit cards at ATMs.

Likewise, since a password is no longer used to access an account, the traditional complexity requirements for a PIN can be relaxed. PINs should still be random, but can be shorter, not changed as often, if at all, and not have the same character complexity requirements. Overall, this provides a better user experience with a higher level of user security.