High Level WebAuthn Authentication flow

WebAuthn authentication process overview

Pre-requisites

The WebAuthn authentication process described in this section is first and foremost based on the authentication specification in the W3C WebAuthn standard. The WebAuthn authentication process is equivalent to the WebAuthn Get Assertion procedure. All WebAuthn authentication parameters, JSON objects, and generic WebAuthn flows are based on the W3C WebAuthn standard.

In addition to the W3C WebAuthn standard, the specific authentication flow described in this section adheres to the User Login section of the Identifier First Flow document. The Identifier First Flow user login description explains the WebAuthn authentication process as well as Account Recovery.

Furthermore, the Custom Authentication Flow in the High Level Architecture document describes the technical environment in more depth. This document describes the AWS architecture, Lambda components and the SQL database that need to be deployed for hosting a WebAuthn Relying Party for the WebAuthn Starter Kit.

It is recommended to study the documents mentioned above as a pre-requisite.

Finally, a user account and corresponding FIDO authenticator must be registered before, as described in the page High Level WebAuthn Registration Flow.

Purpose of this section

Now, this section describes how to perform the WebAuthn authentication process by using Microsoft Windows, an Apple iOS smartphone with the Safari browser, and an Apple iOS smartphone with an app that implements WebAuthn.

As described in the Identifier First Flow page, there are a number of routes the authentication flow can take, depending of what clients, parameters and selections that are used during the authentication process. For each section, the client’s configuration is described, as well as the authentication scenario.

WebAuthn authentication using Microsoft Windows

Client configuration for Microsoft Windows

The client configuration used in this section is the following:

- Operating system: Microsoft Windows 10 Pro (edition 2004)
- Web browser: Google Chrome (version 84.0.4147.105)
- FIDO2 implementation: Microsoft's Web Authentication API, which is a Win32 API that exposes the W3C WebAuthn functions to Windows 10 applications (including Google Chrome), and Microsoft's CTAP2 stack

Authentication with UV=1 on Windows

Authentication selections for UV=1 on Windows

The FIDO2 authenticator used in this section is the following:

- A YubiKey 5 NFC (version 5.2.6) is used as FIDO2 authenticator. The YubiKey 5 is configured with FIDO2 credentials and a PIN-code according to section High Level WebAuthn Registration flow.

The authentication route described in this section is derived from the following parameters and selections according to the Identifier First Flow page:

- The WebAuthn GetAssertion parameter UserVerification is set to 'Preferred', which resolves to the CTAP2 parameter UV=1 for Google Chrome used on Windows 10.
- A "network PIN" is not set for this account as part of the registration process (since the CTAP2 parameter UV=1 is set).
- Backup codes are not used for restoring the account.

Authentication flow and UX design for UV=1 on Windows

The selections for the authentication process, described in the pre-requisites above, result in the authentication flow and UX described in this section.

Step 1.1: The YubiKey 5 is selected as authenticator. (In other words, a platform authenticator is not used.)

Step 1.2: The user visits the Login page https://localhost:8080/login. The user enters the username that was created in the registration section for Windows.

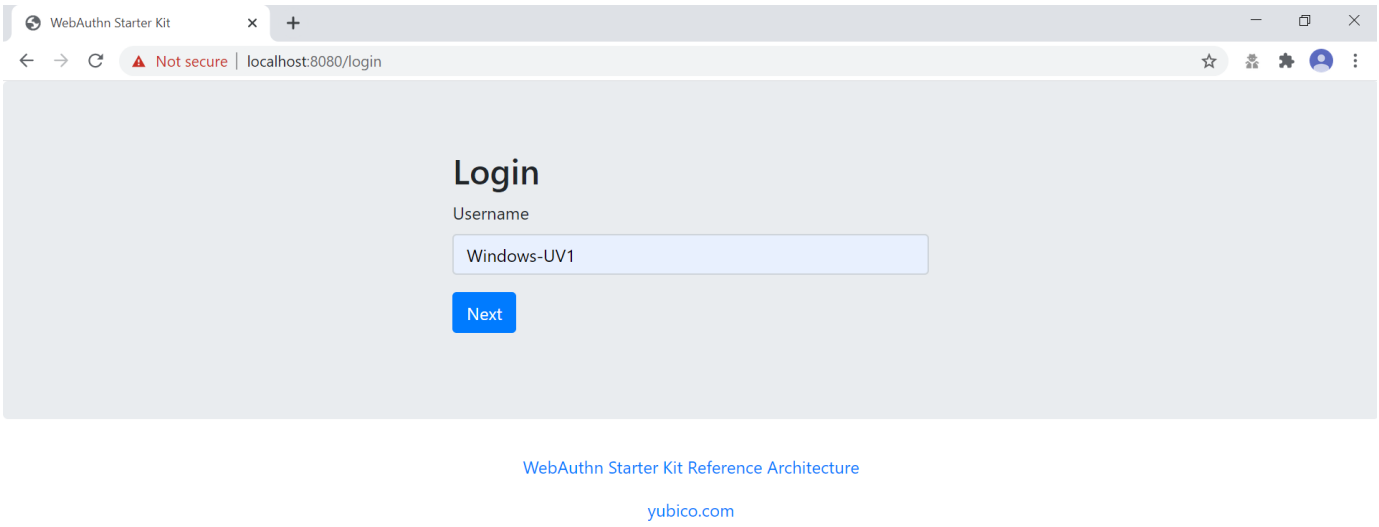


Figure 1 - *Login page for WebAuthn authentication*

Step 1.3: The user presses “Next” in the window above, and the user gets the option to login with a security key. The WebAuthn parameter UserVerification is set to 'Preferred' (CTAP2 UV=1) for this authentication process. The user inserts the YubiKey 5 into the computer and presses the button “Login with Security Key”.

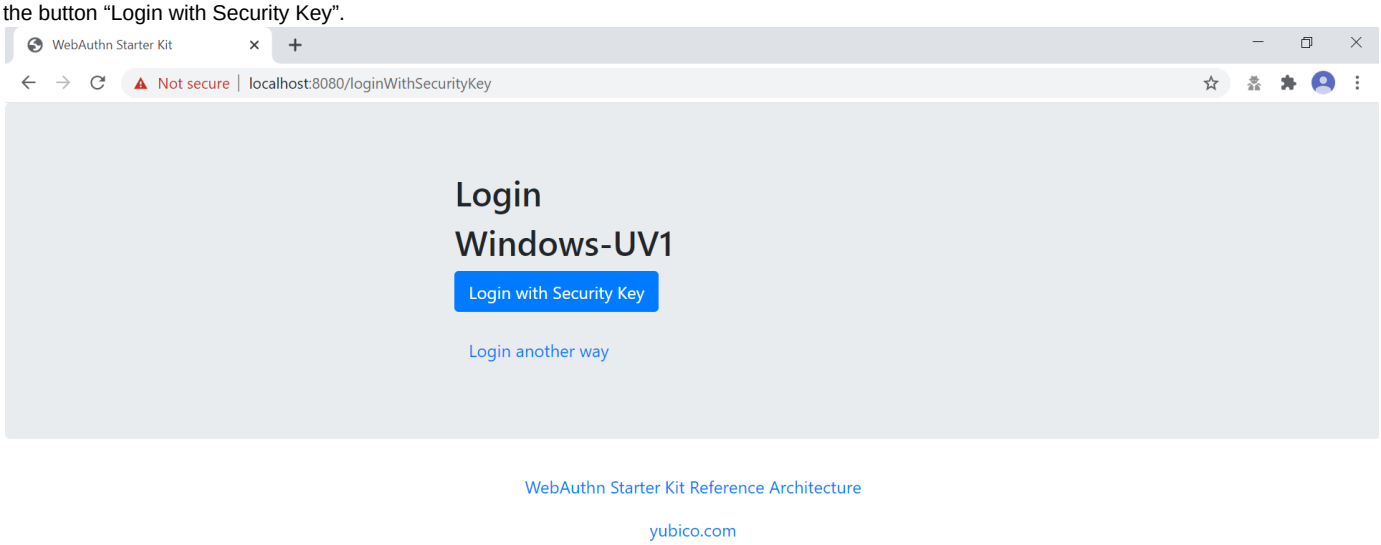


Figure 2 - *Login page for WebAuthn authentication with a security key*

Step 1.4: Windows displays a security dialog box, in which the user enters the PIN-code for the YubiKey.

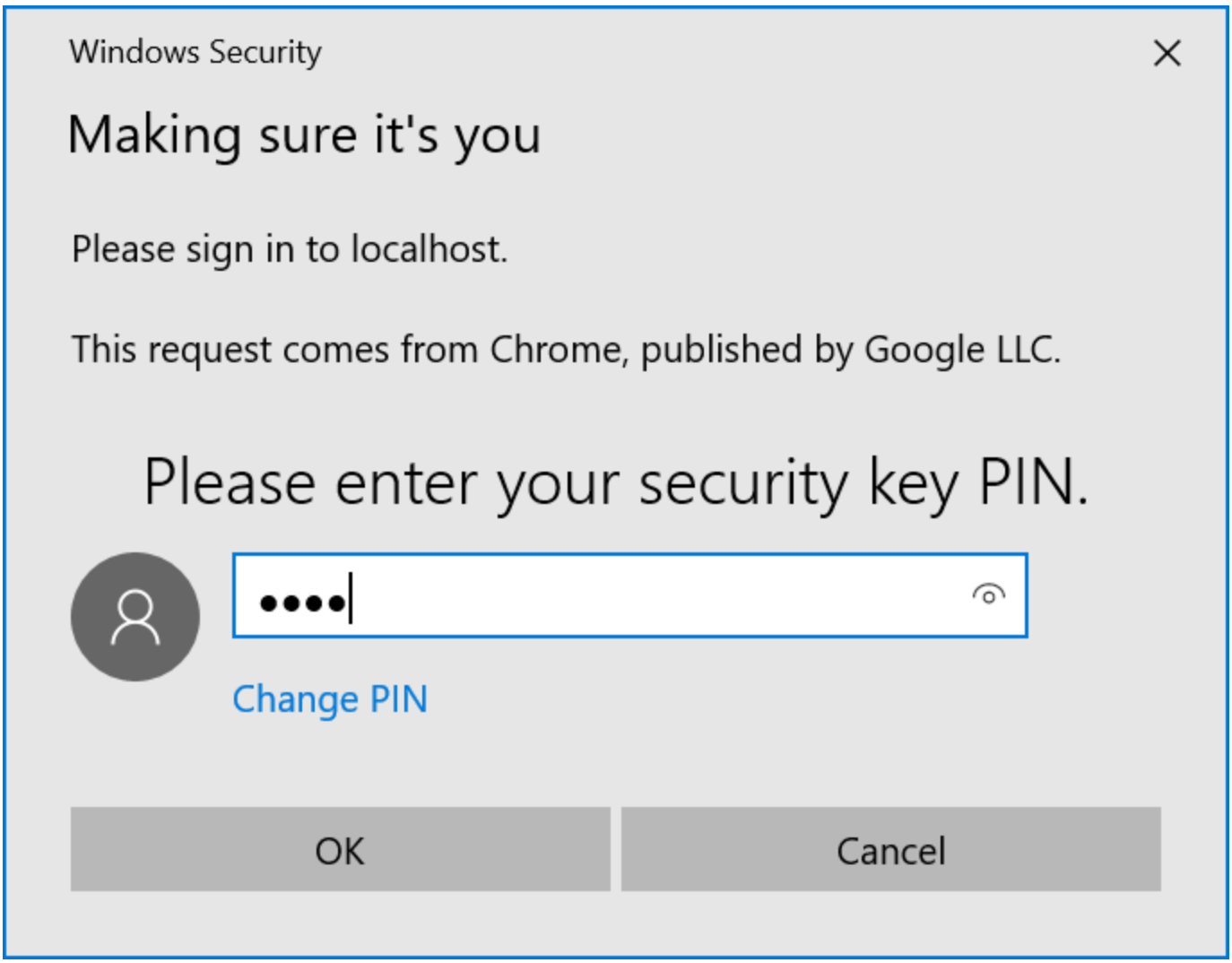


Figure 3 - *Enter PIN-code for the YubiKey's FIDO application*

Step 1.5: Windows displays a security dialog box with instructions for the user to touch the security key. The user touches the sensor on the YubiKey.

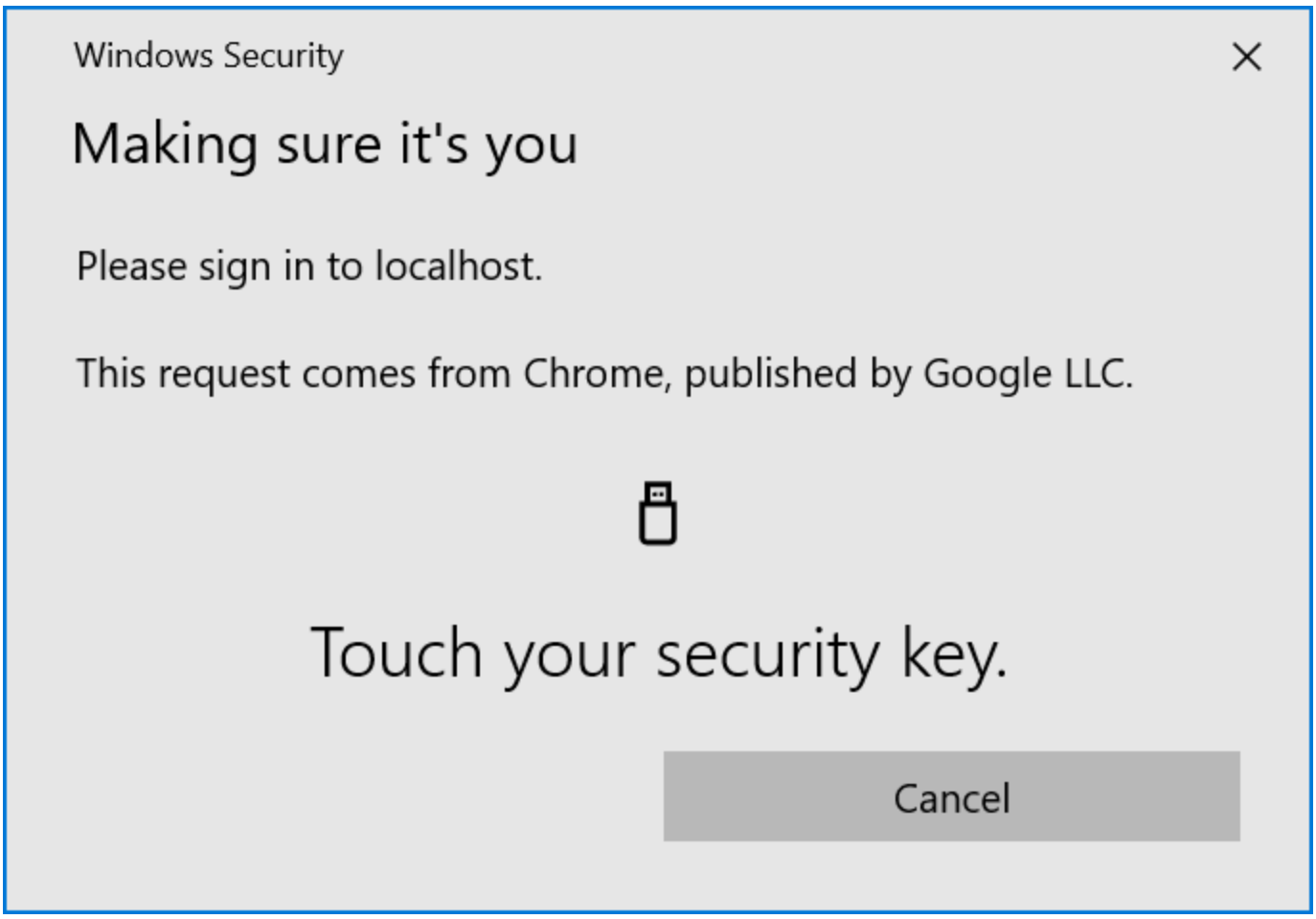


Figure 4 - *Touch the YubiKey for FIDO2 authentication*

Step 1.6: The user is logged in, and is presented with the options presented in the dialog box shown below.

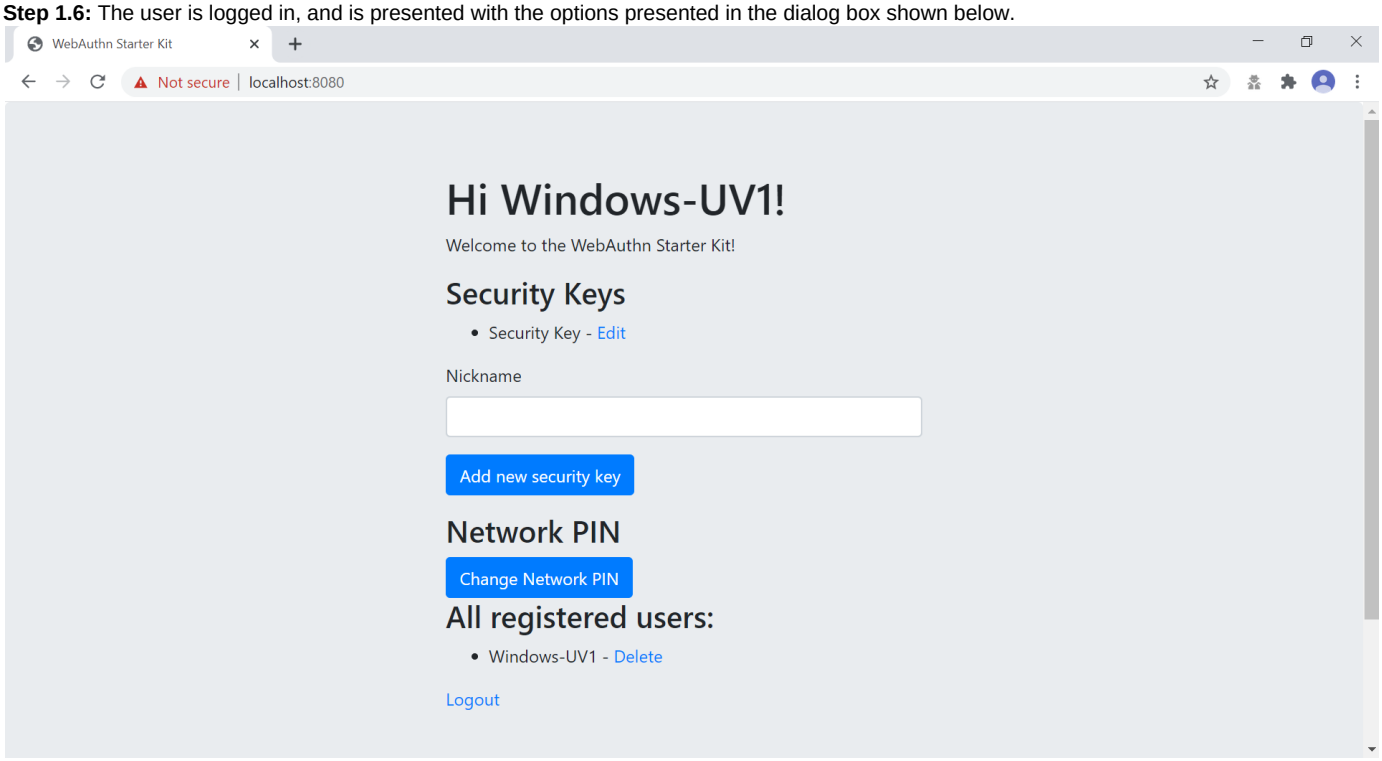


Figure 5 - *Successful login*

Authentication with UV=0 on Windows

Authentication selections for UV=0 on Windows

The FIDO authenticator used in this section is the following:

- A YubiKey 5 NFC (version 5.2.6) is used as FIDO authenticator. The FIDO2 application on the YubiKey is **deactivated**. The YubiKey 5 is configured with FIDO U2F credentials and no PIN-code according to section High Level WebAuthn Registration flow.

The authentication route described in this section is derived from the following parameters and selections according to the Identifier First Flow page:

- The WebAuthn GetAssertion parameter UserVerification is set to 'Preferred', which resolves to the CTAP2 parameter UV=0 for a FIDO2 disabled YubiKey used with Google Chrome on Windows 10. The backward compatible FIDO U2F flow of WebAuthn is therefore used. This behaviour is equivalent to setting the WebAuthn GetAssertion parameter UserVerification to 'Discouraged'.
- A “network PIN” is set for this account as part of the registration process (since the CTAP2 parameter UV=0 is set).
- Backup codes are not generated.

Authentication flow and UX design for UV=0 on Windows

The authentication process for UV=0 is identical to the authentication process for UV=1 with one notable exception:

The FIDO authenticator will not require a PIN-code (as shown in figure 1.3); the FIDO authenticator will only require the user to touch the FIDO authenticator. Instead, a the user must enter a Network PIN when accessing the account. The Network PIN is used as first factor authentication to protect the account.

An example of how to enter a Network PIN is shown in the screenshot below.

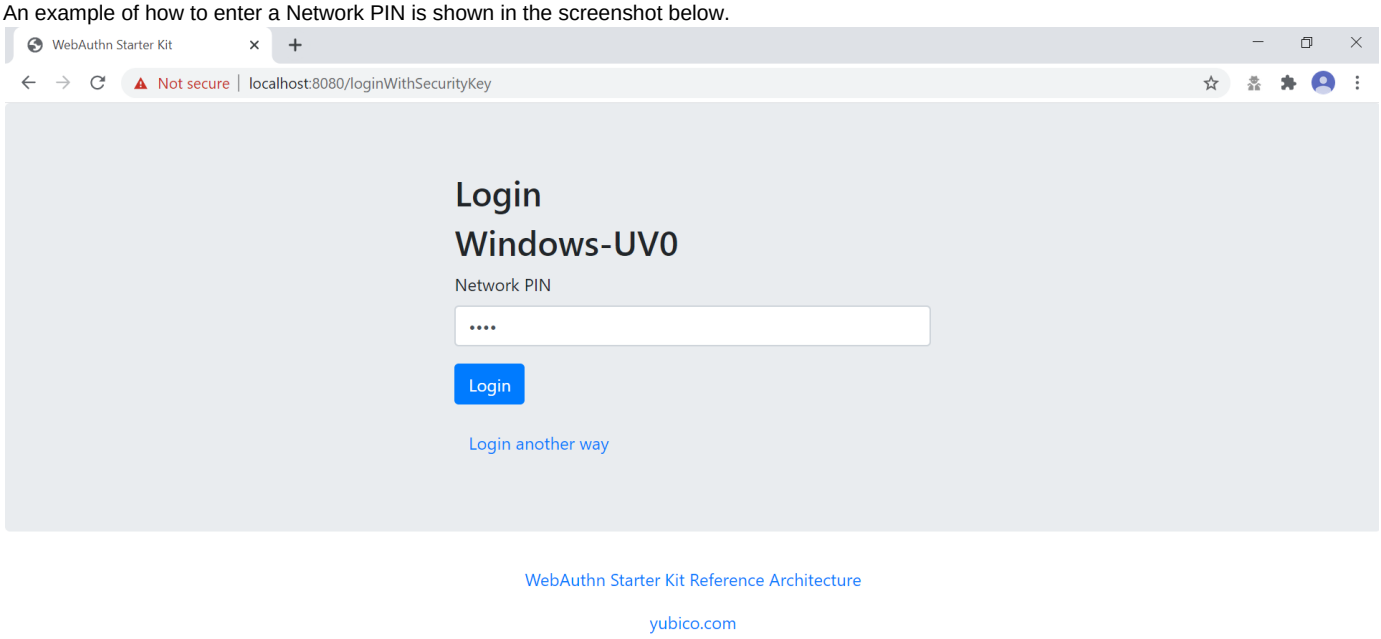


Figure 6 - *The user enters a Network PIN*

WebAuthn authentication using Apple MacOS

Client configuration for Apple MacOS

The client configuration used in this section is the following:

- Operating system: Apple MacOS Catalina 10.15.6
- Web browser: Google Chrome (version 84.0.4147.125)

- FIDO2 implementation: Google Chrome's Web Authentication API, which is an API that exposes the W3C WebAuthn functions to MacOS applications (including Google Chrome), and Google's CTAP2 stack

Authentication with UV=1 on MacOS

Authentication selections for UV=1 on MacOS

The FIDO authenticator used in this section is the following:

- A YubiKey 5 NFC (version 5.1.2) is used as FIDO authenticator. The FIDO2 application on the YubiKey is **activated**. The YubiKey 5 has a PIN-code set and FIDO2 credentials enrolled according to section High Level WebAuthn Registration flow.

The authentication route described in this section is derived from the following parameters and selections according to the Identifier First Flow page:

- The WebAuthn GetAssertion parameter UserVerification is set to 'Preferred', which resolves to the CTAP2 parameter UV=1 for a FIDO2 enabled YubiKey with PIN used with Google Chrome on MacOS. This behaviour is equivalent to setting the WebAuthn GetAssertion parameter UserVerification to 'Required'.
- A "network PIN" is not set for this account as part of the registration process (since the CTAP2 parameter UV=1 is set).
- Backup codes are not generated.

Authentication flow and UX design for UV=1 on MacOS

The selections for the authentication process, described in the pre-requisites above, result in the authentication flow and UX described in this section.

Step 2.1: The YubiKey 5 is selected as authenticator. (In other words, a platform authenticator is not used.)

Step 2.2: The user visits the Login page https://localhost:8080/login. The user enters the username that was created in the registration section for MacOS.

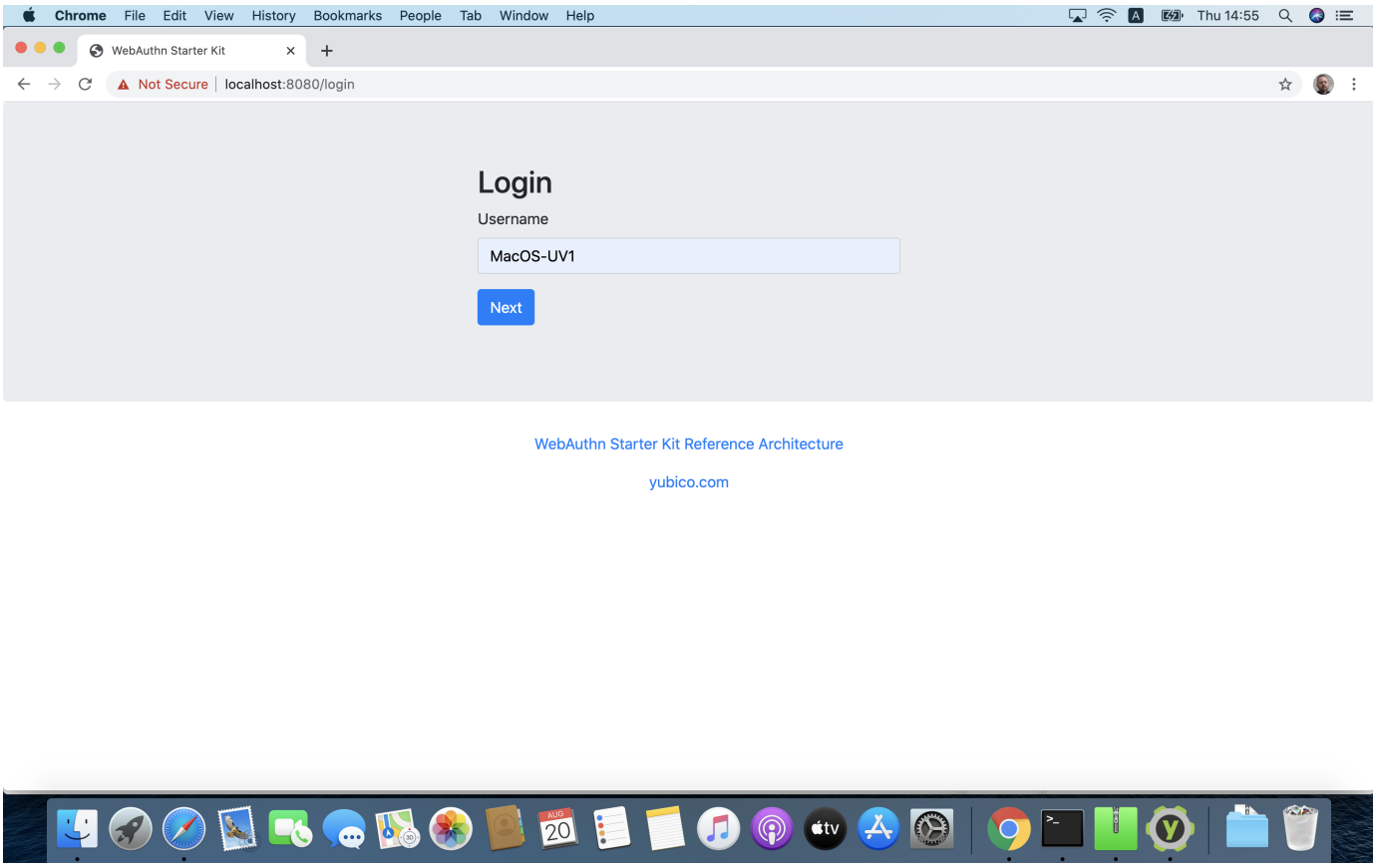


Figure 7 - *Login page for WebAuthn authentication*

Step 2.3: The user presses “Next” in the window above, and the user gets the option to login with a security key. The WebAuthn parameter UserVerification is set to 'Preferred' (CTAP2 UV=1) for this authentication process. The user inserts the YubiKey 5 into the computer and presses the button “Login with Security Key”.

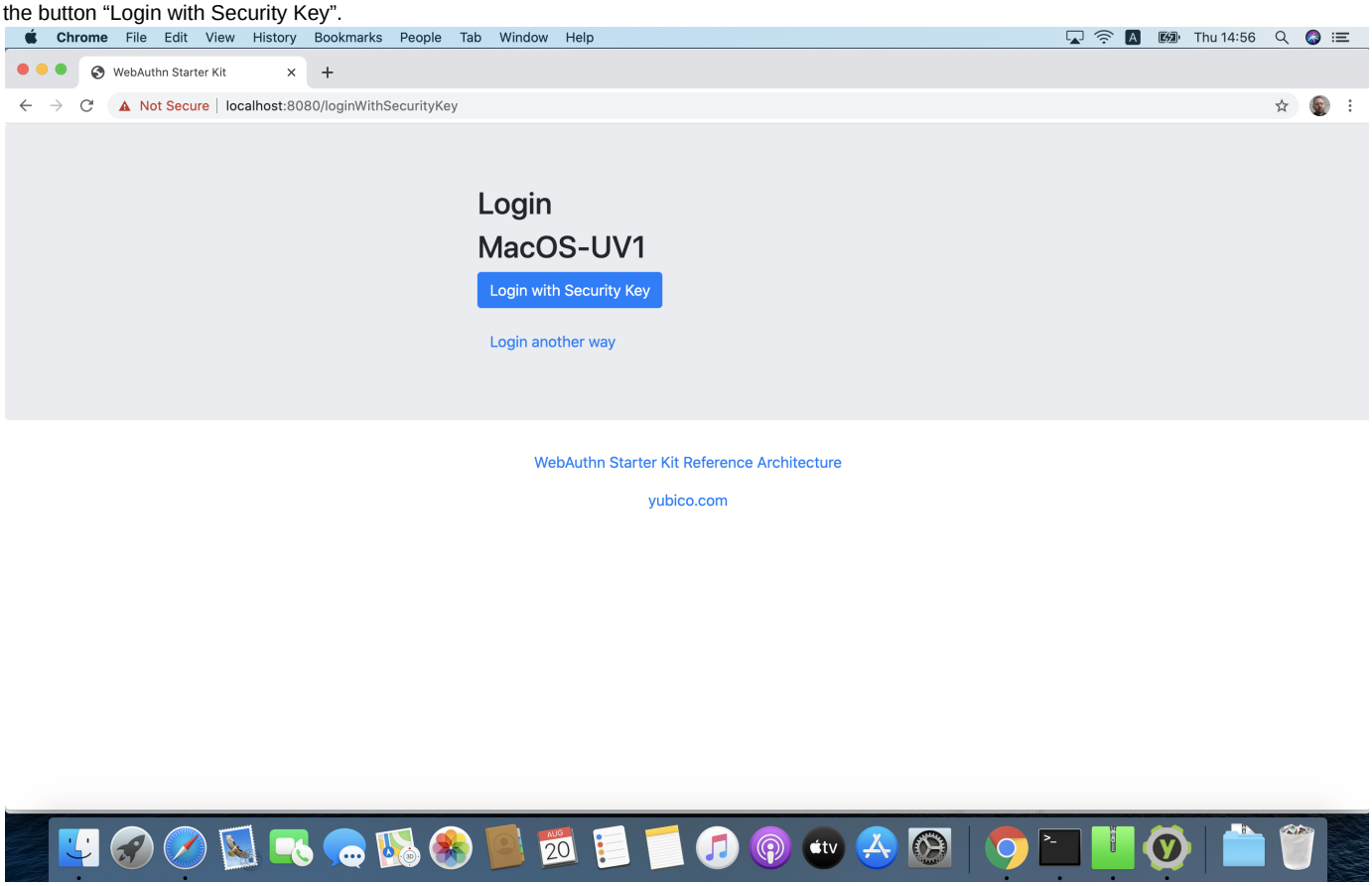


Figure 8 - *Login page for WebAuthn authentication with a security key*

Step 2.4: Google Chrome on MacOS displays a security dialog box, and the user selects the security key and touches it.

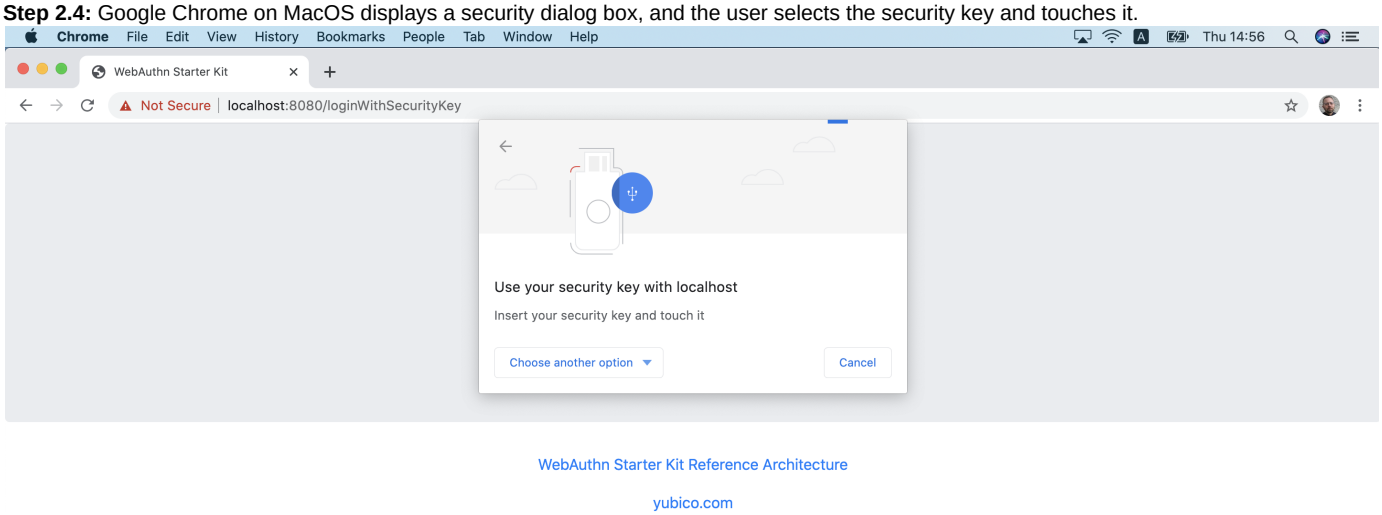
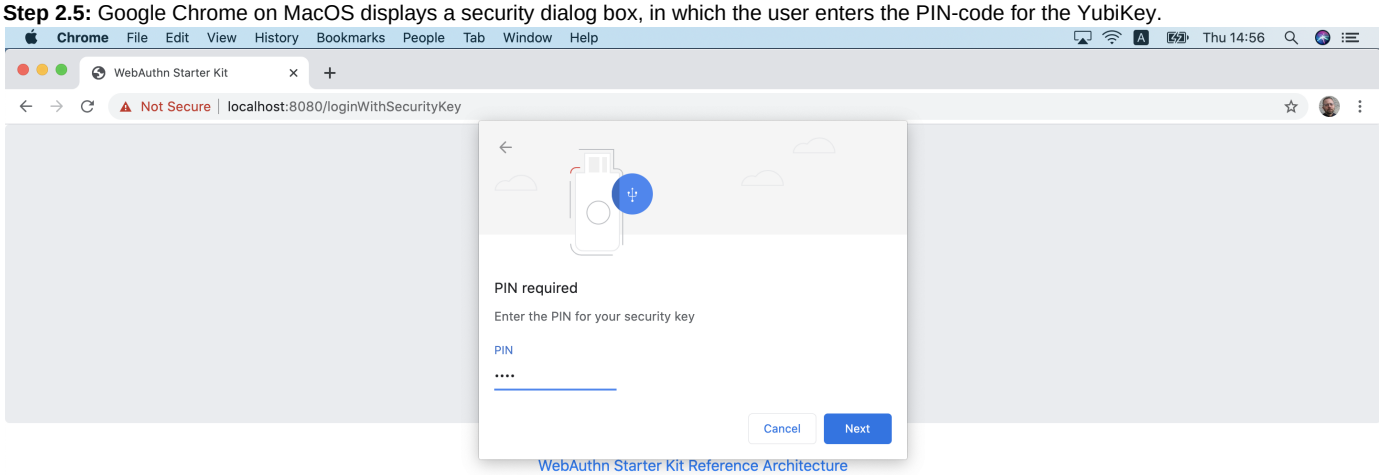


Figure 9 - *Select security key for authentication*

Step 2.5: Google Chrome on MacOS displays a security dialog box, in which the user enters the PIN-code for the YubiKey.



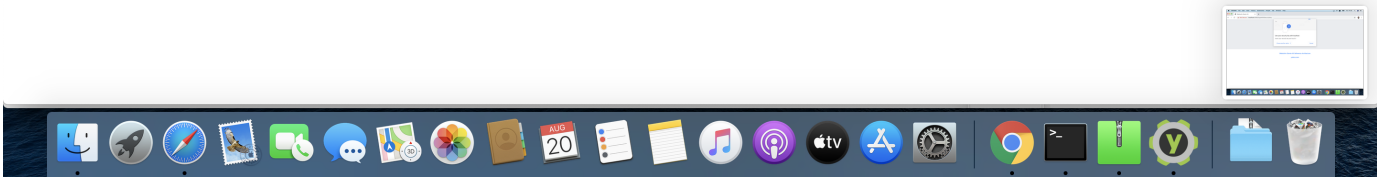


Figure 10 - *Enter PIN to the security key*

Step 2.6: Google Chrome on MacOS displays a security dialog box with instructions for the user to touch the security key again. The user touches the sensor on the YubiKey once more.

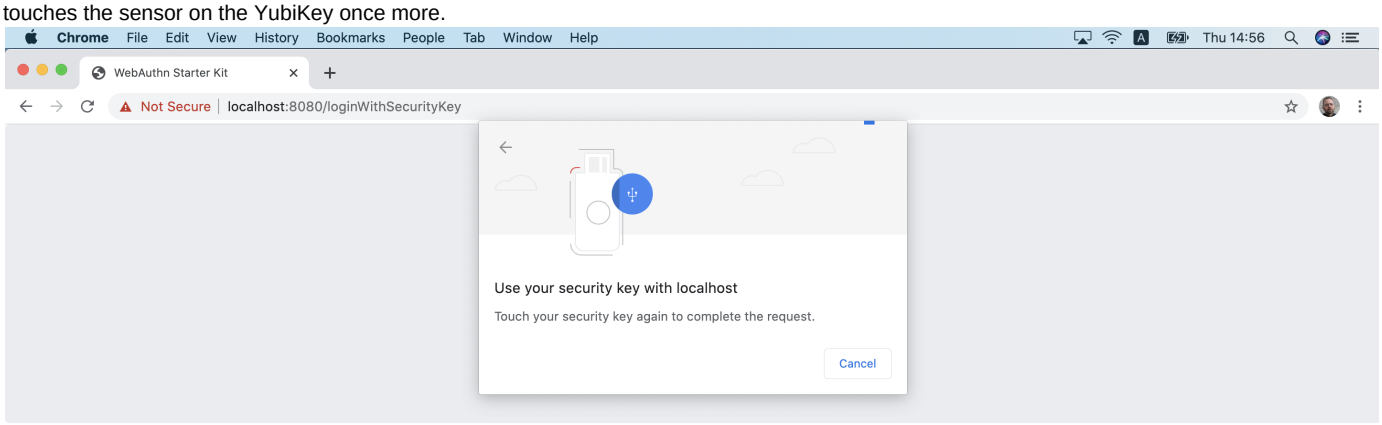


Figure 11 - *Touch the YubiKey for FIDO2 authentication*

Step 2.7: The user is logged in, and is presented with the options presented in the dialog box shown below.

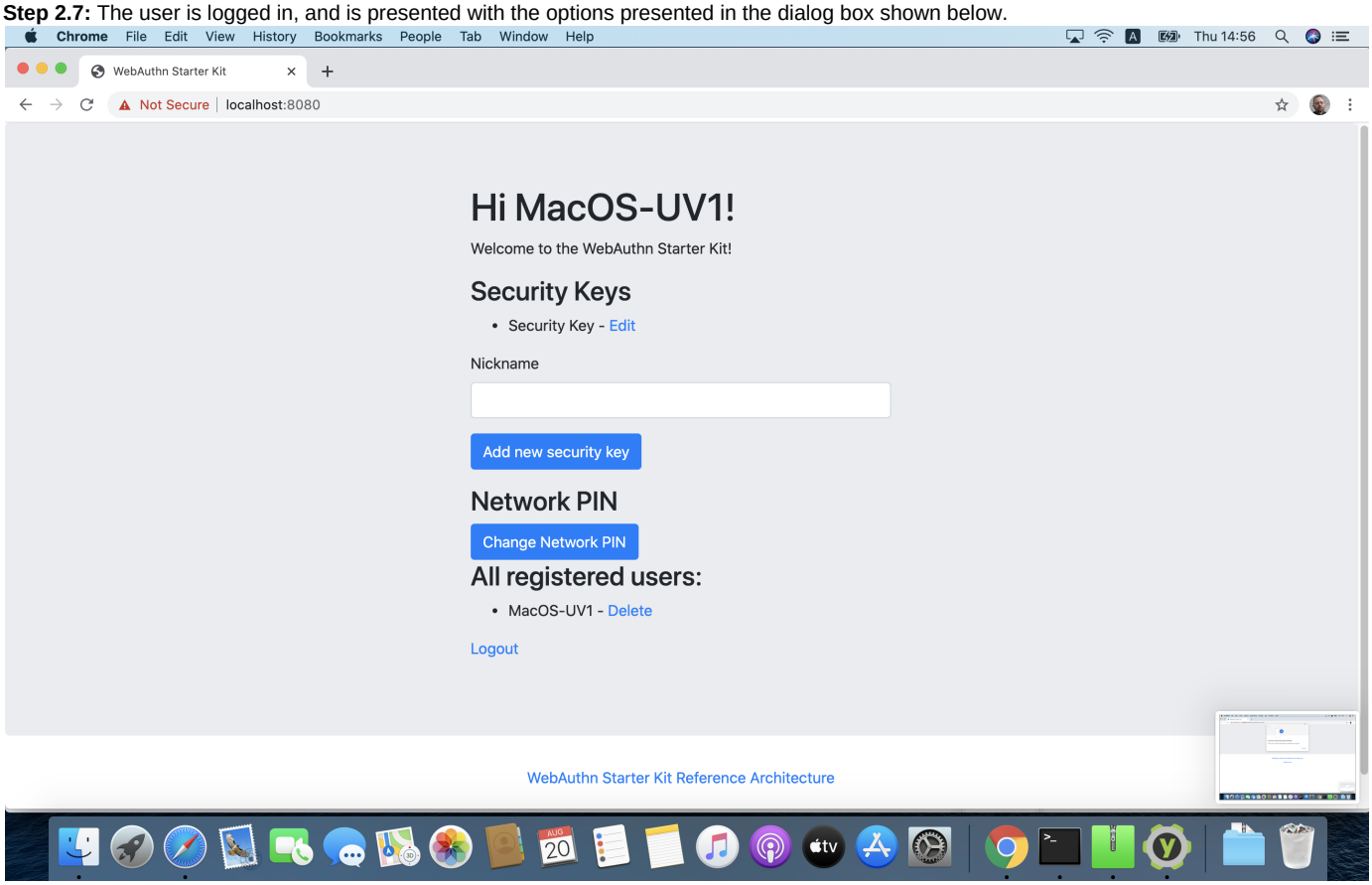


Figure 12 - *Successful login*

Authentication with UV=0 on MacOS

Authentication selections for UV=0 on MacOS

The FIDO authenticator used in this section is the following:

- A YubiKey 5 NFC (version 5.1.2) is used as FIDO authenticator. The FIDO2 application on the YubiKey is **deactivated**, which triggers the UV=0 behaviour on MacOS. Also a YubiKey with the FIDO2 application activated, but with no PIN-code set, will trigger the UV=0 flow on MacOS (which is a significant difference from Windows that will prompt the user for setting a PIN and activate the UV=1 process).

The authentication route described in this section is derived from the following parameters and selections according to the Identifier First Flow page:

- The WebAuthn GetAssertion parameter UserVerification is set to 'Preferred', which resolves to the CTAP2 parameter UV=0 for a FIDO2 disabled YubiKey used with Google Chrome on MacOS. The backward compatible FIDO U2F flow of WebAuthn is therefore used. This behaviour is equivalent to setting the WebAuthn GetAssertion parameter UserVerification to 'Discouraged'.
- A “network PIN” is set for this account as part of the registration process (since the CTAP2 parameter UV=0 is set).
- Backup codes are not generated.

Authentication flow and UX design for UV=0 on MacOS

The authentication process for UV=0 is identical to the authentication process for UV=1 with one notable exception:

The FIDO authenticator will not require a PIN-code (as shown in figure 10); the FIDO authenticator will only require the user to touch the FIDO authenticator. Instead, a the user must set a Network PIN when creating the account. The Network PIN is used as first factor authentication to protect the account.

An example of how to set a Network PIN is shown in the screenshot below.

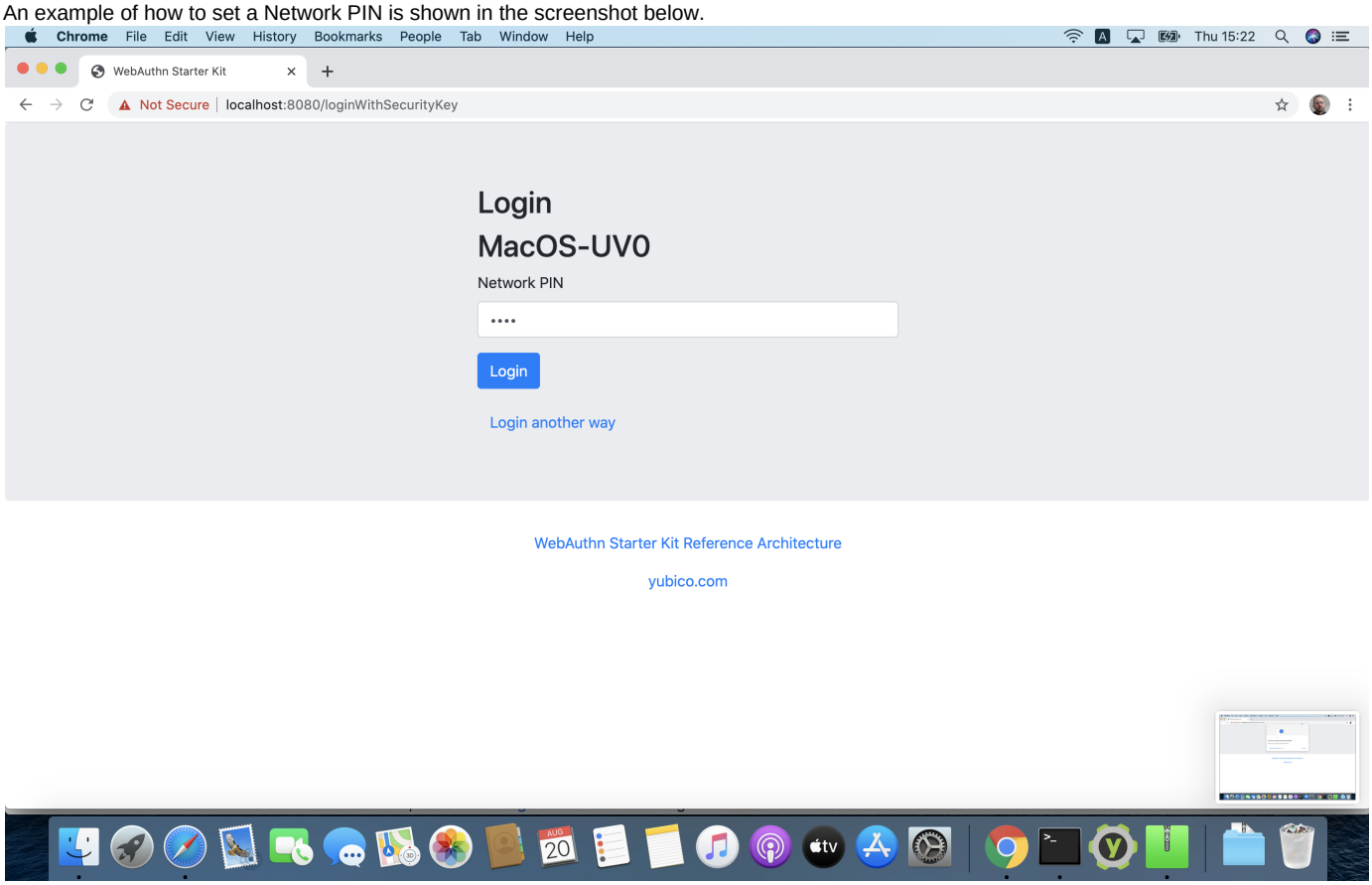


Figure 13 - *The user enters a Network PIN*

WebAuthn authentication using the Apple iOS Safari browser

Client configuration for Apple iOS Safari

The client configuration used in this section is the following:

- Operating system: Apple iPhone iOS 14 developer beta 6
- Web browser: Apple iPhone Safari 14 developer beta 6
- FIDO2 implementation: Apple iPhone iOS 14 developer beta 6, which is an API that exposes the W3C WebAuthn functions to iOS applications (including Safari)

Authentication with UV=1 on Apple iOS Safari

Authentication selections for UV=1 on Apple iOS Safari

The FIDO authenticators used in this section is the following:

- A YubiKey 5Ci (version 5.2.7) is used as FIDO authenticator, which is plugged into the iPhone's lightning port. The FIDO2 application on the YubiKey is **activated**. The YubiKey 5Ci has a PIN-code set and FIDO2 credentials enrolled according to section High Level WebAuthn Registration flow..
- A YubiKey 5 NFC (version 5.2.6) is also used as FIDO authenticator, which is tapped to the iPhone's NFC receiver. The FIDO2 application on the YubiKey is **activated**. The YubiKey 5 has a PIN-code set and FIDO2 credentials enrolled according to section High Level WebAuthn Registration flow. The Yubico OTP application over NFC is deactivated, to avoid the NFC tag pop-up window.

The WebAuthn authentication results are identical when using both YubiKeys.

The authentication route described in this section is derived from the following parameters and selections according to the Identifier First Flow page:

- The WebAuthn GetAssertion parameter UserVerification is set to 'Preferred', which resolves to the CTAP2 parameter UV=1 for a FIDO2 enabled YubiKey with PIN used with Safari on Apple iOS. This behaviour is equivalent to setting the WebAuthn GetAssertion parameter UserVerification to 'Required'.
- A “network PIN” is not set for this account as part of the registration process (since the CTAP2 parameter UV=1 is set).
- Backup codes are not generated.

Authentication flow and UX design for UV=1 on Apple iOS with Safari

The selections for the authentication process, described in the pre-requisites above, result in the authentication flow and UX described in this section.

Step 3.1: The YubiKey is selected as authenticator. (In other words, a platform authenticator is not used.)

Step 3.2: The user visits the Login page ttps://dev.d3sohe8raj3jl8.amplifyapp.com. The user enters the username that was created in the registrat ion section for Apple iOS with Safari.

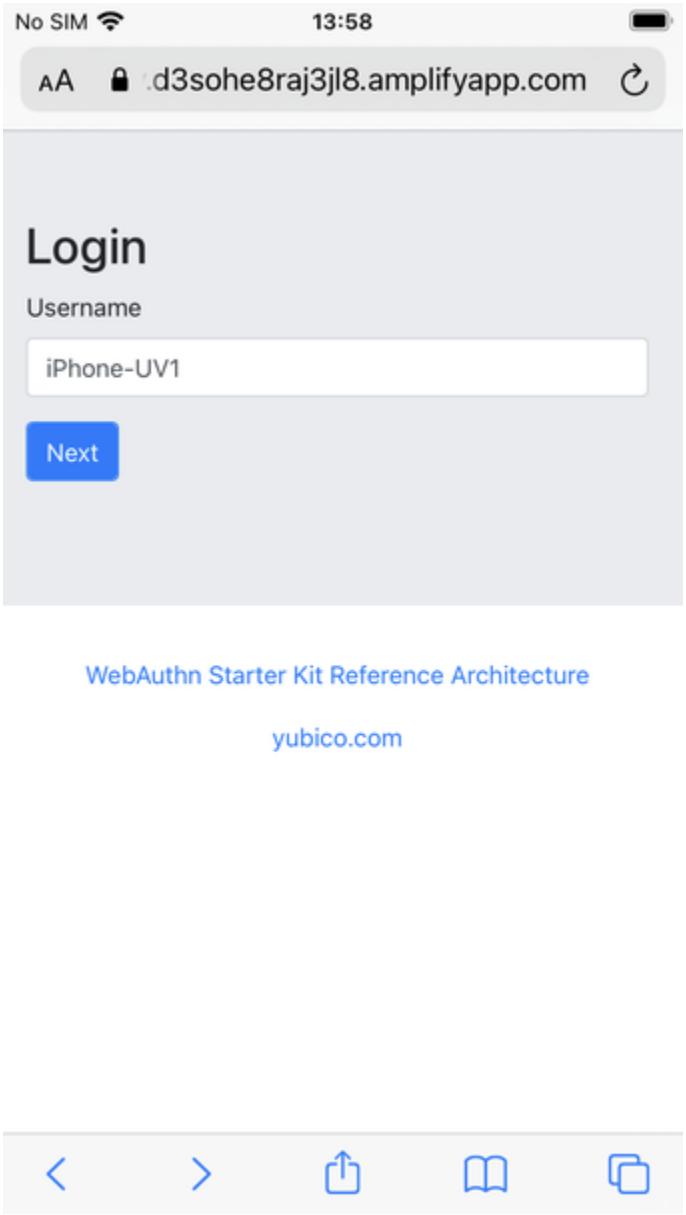


Figure 14 - *Login page for WebAuthn authentication*

Step 3.3: The user presses “Next” in the window above, and the user gets the option to login with a security key. The WebAuthn parameter UserVerification is set to 'Preferred' (CTAP2 UV=1) for this authentication process. The user inserts or taps the YubiKey to the iPhone.

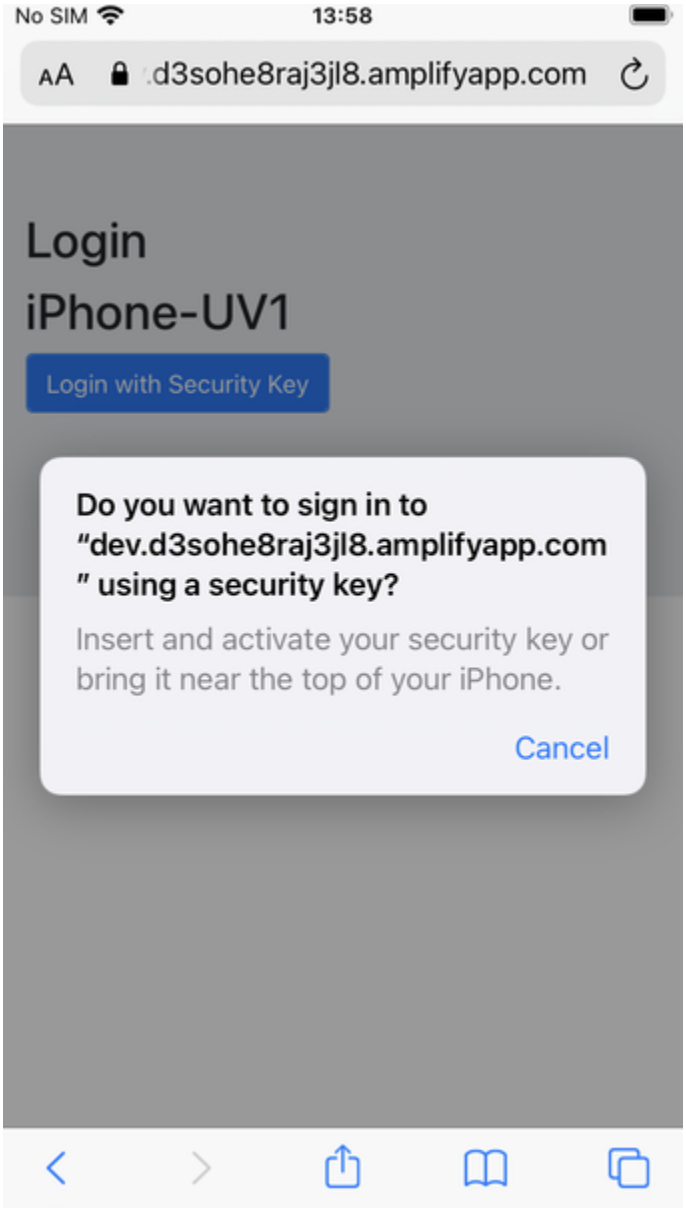
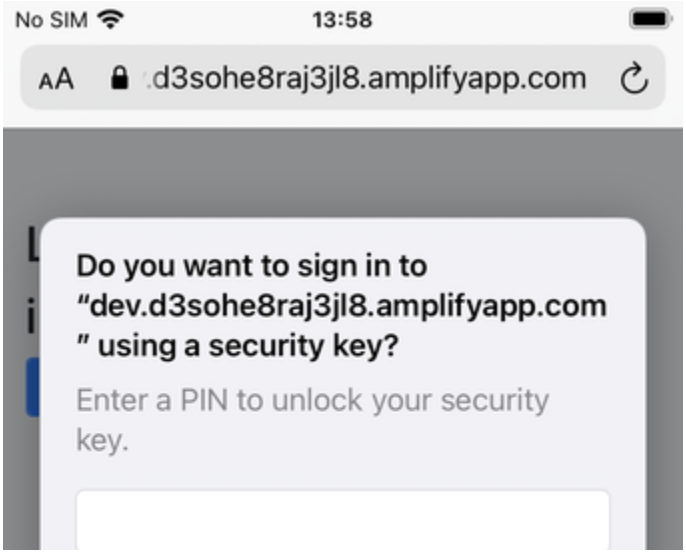


Figure 15 - *Login page for WebAuthn authentication with a security key*

Step 3.4: Safari on Apple iOS displays a security dialog box, in which the user enters the PIN-code for the YubiKey.



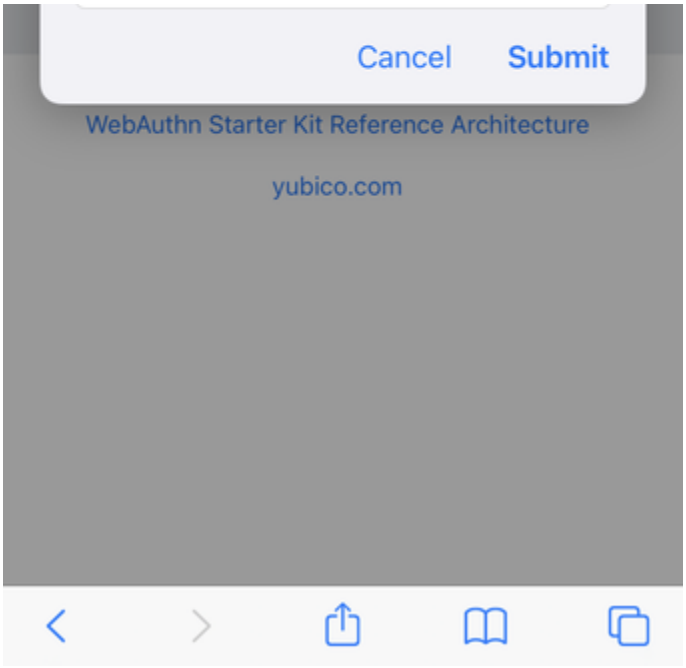
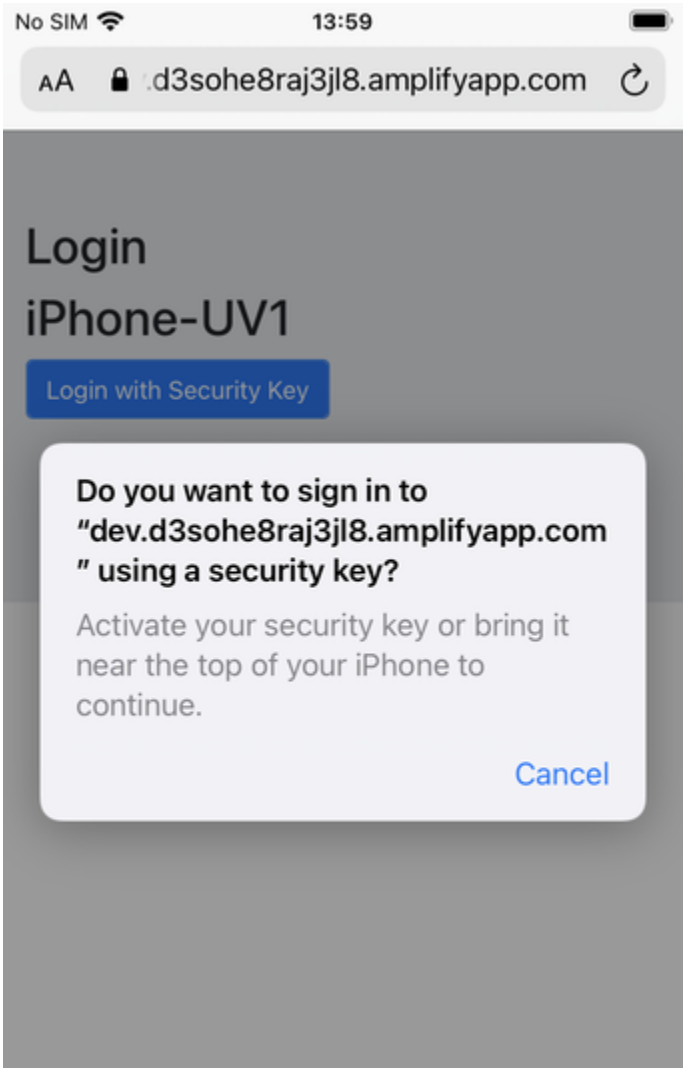


Figure 16 - *Enter PIN to the security key*

Step 3.5: Safari on Apple iOS displays a security dialog box with instructions for the user to touch the security key again. The user touches the sensor on the YubiKey once more.



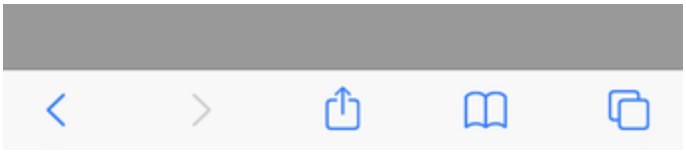


Figure 17 - *Touch the YubiKey for FIDO2 authentication*

Step 3.6: The user is logged in, and is presented with the options presented in the dialog box shown below.

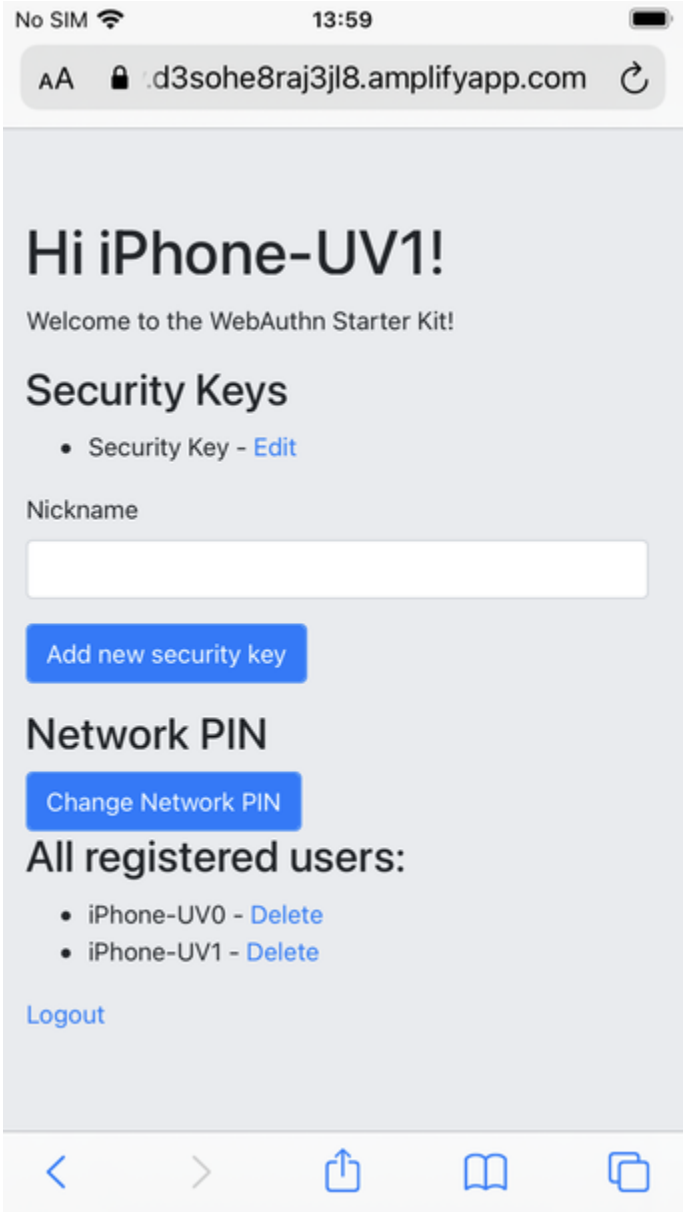


Figure 18 - *Successful login*

Authentication with UV=0 on Apple iOS with Safari

Authentication selections for UV=0 on Apple iOS with Safari

The FIDO authenticators used in this section are the following:

- A YubiKey 5Ci (version 5.2.7) is used as FIDO authenticator, which is plugged into the iPhone's lightning port. The FIDO2 application on the YubiKey is **deactivated**, which triggers the UV=0 behaviour on iOS. Also a YubiKey with the FIDO2 application activated, but with no

PIN-code set, will trigger the UV=0 flow on MacOS (which is a significant difference from Windows that will prompt the user for setting a PIN and activate the UV=1 process). The YubiKey 5Ci has no PIN-code set and FIDO2 credentials enrolled according to section High Level WebAuthn Registration flow.

- A YubiKey 5 NFC (version 5.2.6) is also used as FIDO authenticator, which is tapped to the iPhone's NFC receiver. The FIDO2 application on the YubiKey is **deactivated**, which triggers the UV=0 behaviour on iOS (which is a significant difference from Windows that will prompt the user for setting a PIN and activate the UV=1 process). The YubiKey 5 has no PIN-code set and FIDO2 credentials enrolled according to section High Level WebAuthn Registration flow. The Yubico OTP application over NFC is deactivated, to avoid the NFC tag pop-up window.

The WebAuthn authentication results are identical when using both YubiKeys.

The authentication route described in this section is derived from the following parameters and selections according to the Identifier First Flow page:

- The WebAuthn GetAssertion parameter UserVerification is set to 'Preferred', which resolves to the CTAP2 parameter UV=0 for a FIDO2 disabled YubiKey used with Safari on Apple iOS. The backward compatible FIDO U2F flow of WebAuthn is therefore used. This behaviour is equivalent to setting the WebAuthn GetAssertion parameter UserVerification to 'Discouraged'.
- A "network PIN" is set for this account as part of the registration process (since the CTAP2 parameter UV=0 is set).
- Backup codes are not generated.

Authentication flow and UX design for UV=0 on Apple iOS with Safari

The authentication process for UV=0 is identical to the authentication process for UV=1 with one notable exception:

The FIDO authenticator will not require a PIN-code (as shown in figure 16); the FIDO authenticator will only require the user to touch the FIDO authenticator. Instead, a the user must set a Network PIN when creating the account. The Network PIN is used as first factor authentication to protect the account.

An example of how to set a Network PIN is shown in the screenshot below.

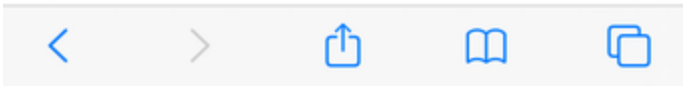


Figure 19 - *The user enters a Network PIN*

WebAuthn authentication using an Apple iOS app

Notes for this section:

- Describe the WebAuthn authentication process by using Apple iOS and the app as client, with the AWS demo application as backend.
  - Describe both UV=0 and UV=1 scenarios.
  - Describe how to recover the account by using backup codes.
- Describe how to use the iOS WebAuthn API?
- Take screenshots of the WebAuthn authentication process, by using Apple iOS and the app, with the AWS Relying Party as the end-point.