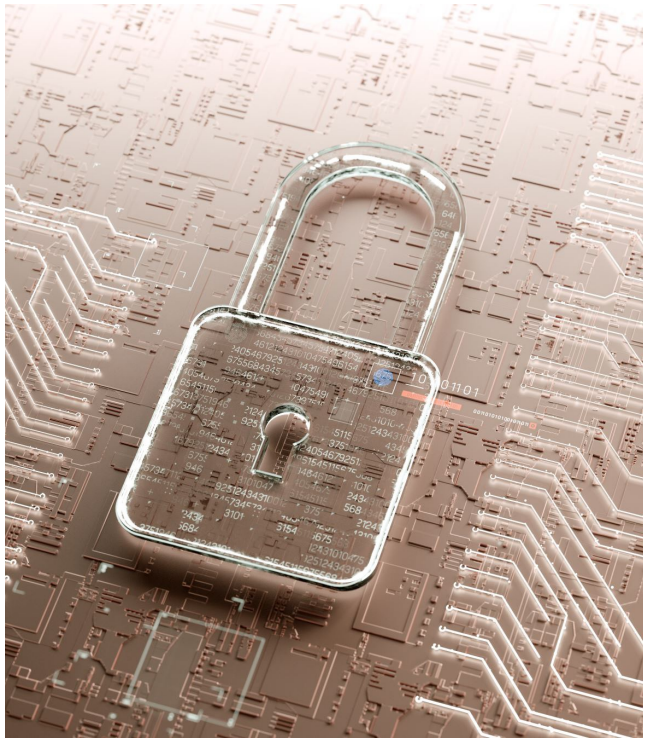
Implementing Passwordless logins using Passkey, WebAuthn protocols & Spring Authorization Server

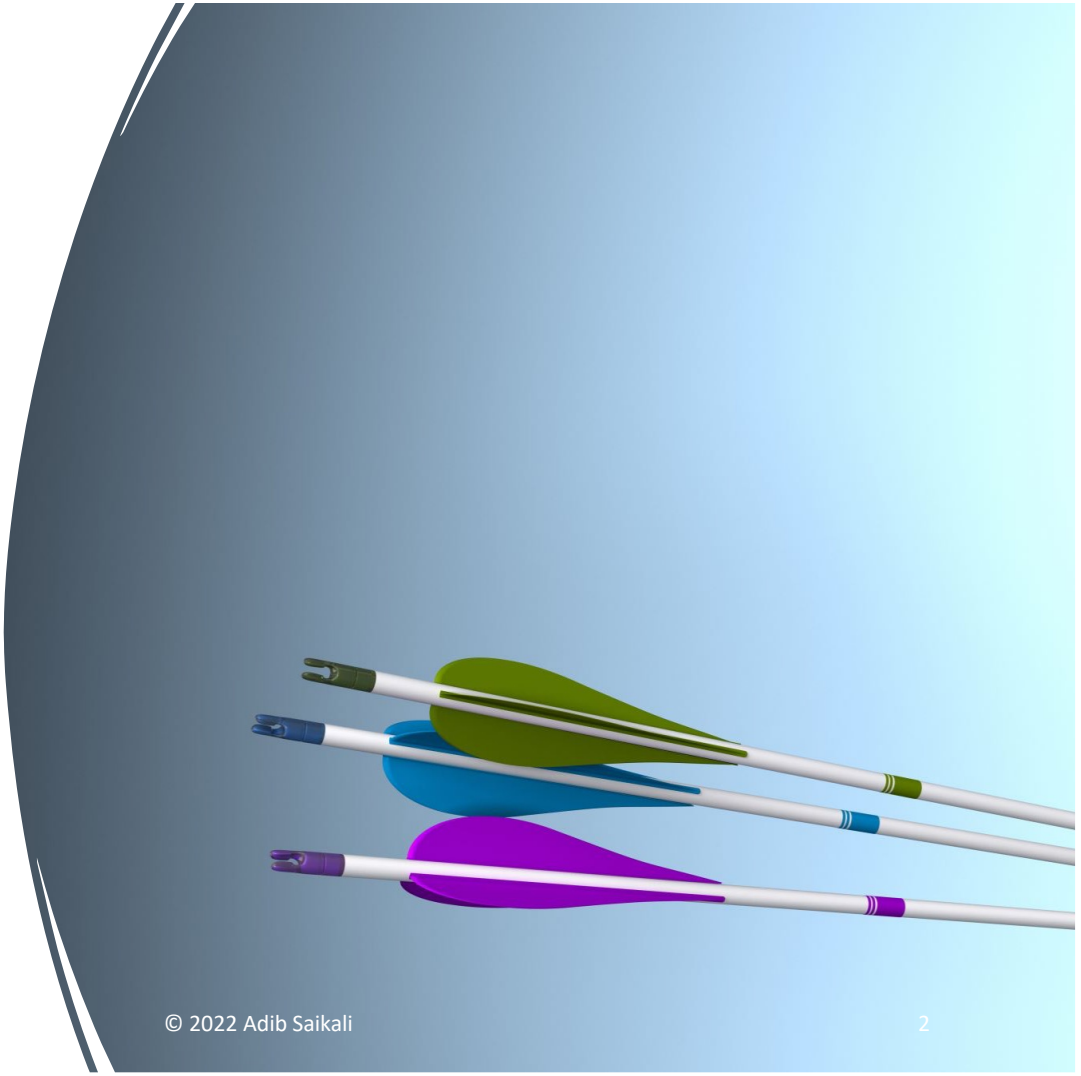
Adib Saikali @asaikali

Joe Grandja @joe_grandja



The Goal

Learn how to add Passkey support to your existing applications using the Spring Authorization Server



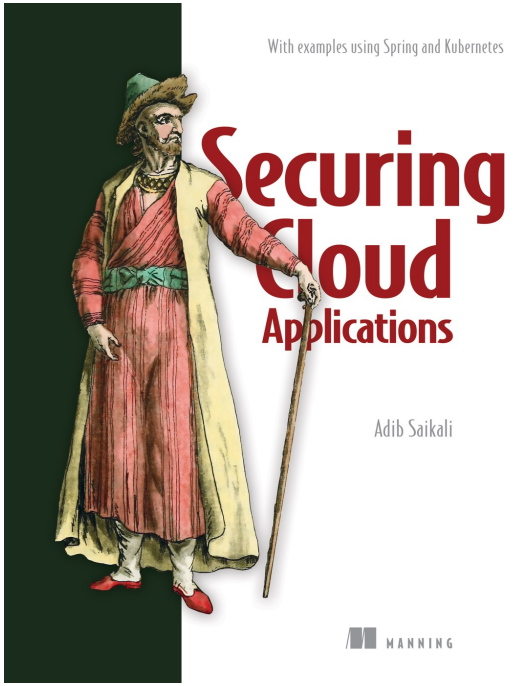
The plan

1. Try the Passkey User Experience
2. Learn how Passkey works by adding it to a Spring Boot application
3. Learn how Spring Authorization Works
4. Add Passkey to Spring Authorization Server

Adib Saikali – @asaikali on twitter

Download book for free at https://tanzu.vmware.com/content/ebooks/securing-cloud-applications

- Software developer since 1995
- Code Janitor since 2014
- Global Field Principal @ VMware Tanzu
  - Kubernetes, Cloud Foundry, Spring
  - Cloud Native Application Architecture (Modular Monoliths & Microservices)
  - Application modernization refactoring patterns
  - Applications security patterns
- Authoring “Securing Cloud Applications” for Manning Publications

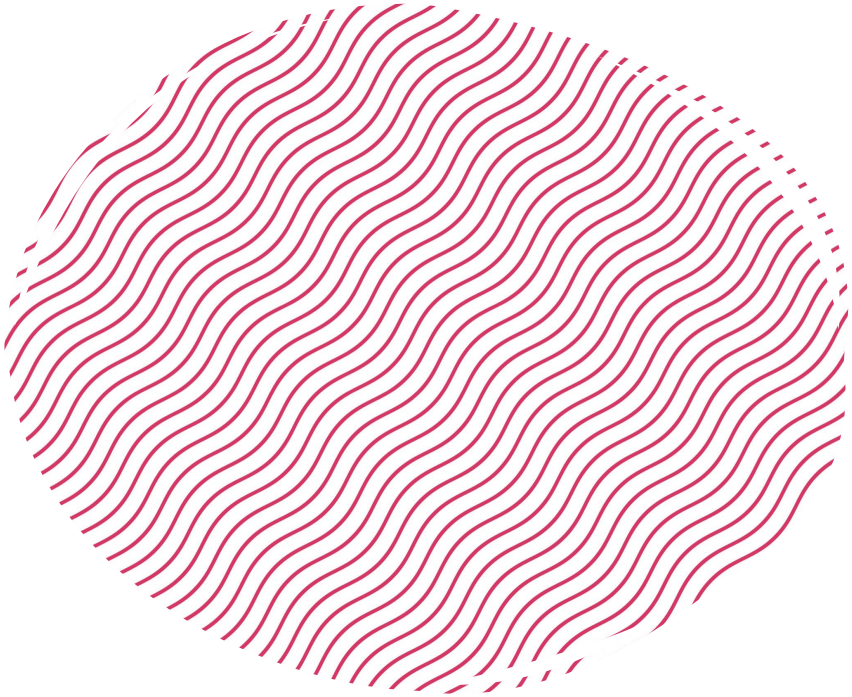


https://www.manning.com/books/securing-cloud-applications

Joe Grandja

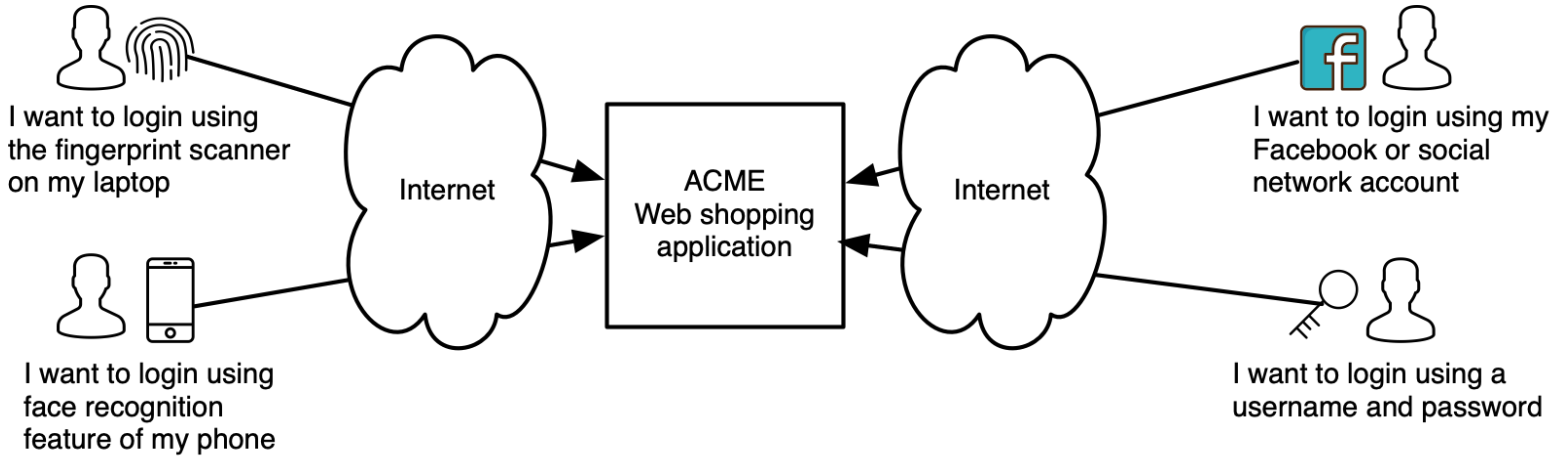
Spring Security Senior Engineer | Toronto, Canada

Joe has been in the Software Industry for over 20 years. He has successfully designed, built and delivered enterprise grade software in the financial services and health sector. He has been using Spring for over 10 years and is very excited to have joined the Spring Security engineering team, in early 2016. Outside of his passion for crafty software, Joe continues to travel the world with his family, snowboarding the most challenging mountains, exploring nature on foot and doing his best to enjoy what life brings.

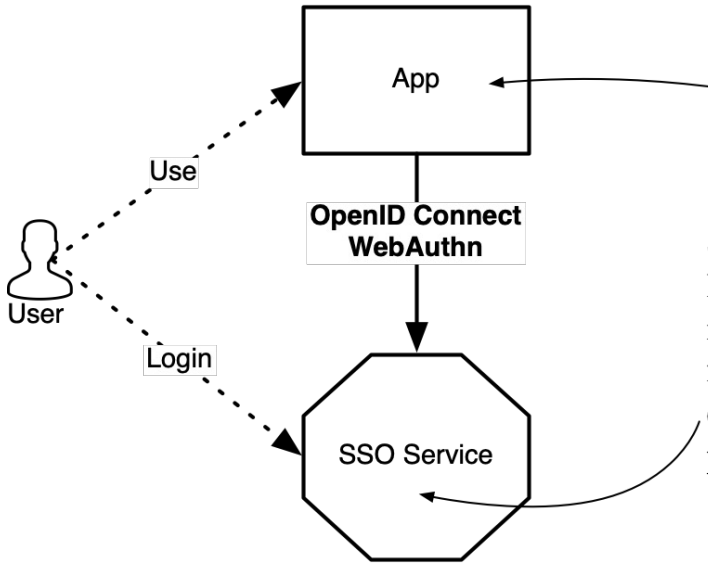


User Identity Patterns & Protocols

User Authentication



Speak OpenID Connect delegate your authentication troubles away



You write your application to interact with an SSO service using the OpenID Connect & WebAuthn protocols

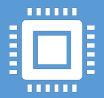
Support all login technologies such as social network login, biometric Face ID or fingerprint, along with multi-factor password authentication. Can be federated with other SSO services to enable collaboration between customers, employees, and partners.

Thought Experiment … Imagine that you

Built an application following secure application coding practices

Hired a security consulting firm to audit your code looking for security vulnerabilities

Trained your users to pick strong passwords and follow password management best practices



The application requires users to login with a password and a second factor such as one-time password from an authenticator app

How can you break into this application?

Attack the human

Build	Build a website login screen that looks like the real app
Trick	Trick the human into going to the fake site
Trick	Trick the human into entering their password and one time token into the fake site
Use	Use the stolen password and OTP code to access the real site
Steal	Steal user data or perform unauthorized operations such as buying stuff or stealing money

All humans can be phished, we need a phishing resistant authentication technology

A phishing resistant authenticator can validate that the user is accessing the real site and refuse to authenticate against a phishing site

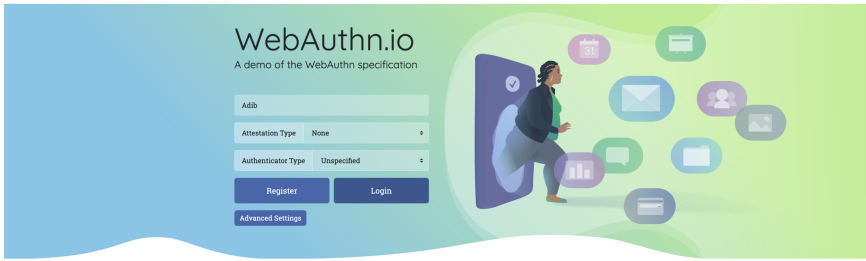
Cross-Platform FIDO2 Phishing Resistant Authenticators

Platform Phishing Resistant Authenticators

- Leverages features of user's devices such as
  - Facial recognition
  - Thumbprint scanner
- Widely supported
- Most users already have a device that can be used as a platform authenticator

WebAuthn User Experience

Try it out at https://webauthn.io/



What is WebAuthn?

Welcome to webauthn.io! This site is designed by Duo Labs to test the new W3C Specification Web Authentication. WebAuthn is supported in the Chrome, Firefox, and Edge browsers to different degrees, but support for credential creation and assertion using a U2F Token, like those provided by Yubico and Feitian, is supported by all of them. The code for this demo can be found here on GitHub. To read more about WebAuthn and what is does, check out webauthn.guide for an introduction.

Read more at webauthn.guide

How do Phishing Resistant Authenticators Work?

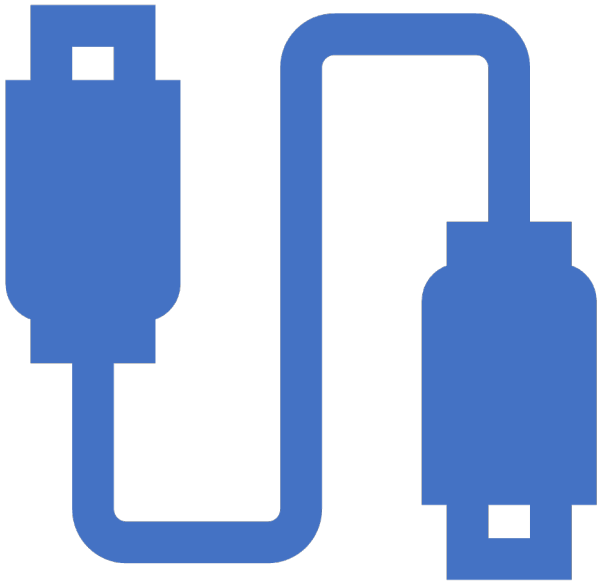
The authenticator generates a unique private-public key pair to identify a user on a specific website

The private key never leaves the authenticator

The public key is shared with the website

The authenticator is passed the website url and userid as part of the authentication flow the authentication will only work if the userid and website url match what is

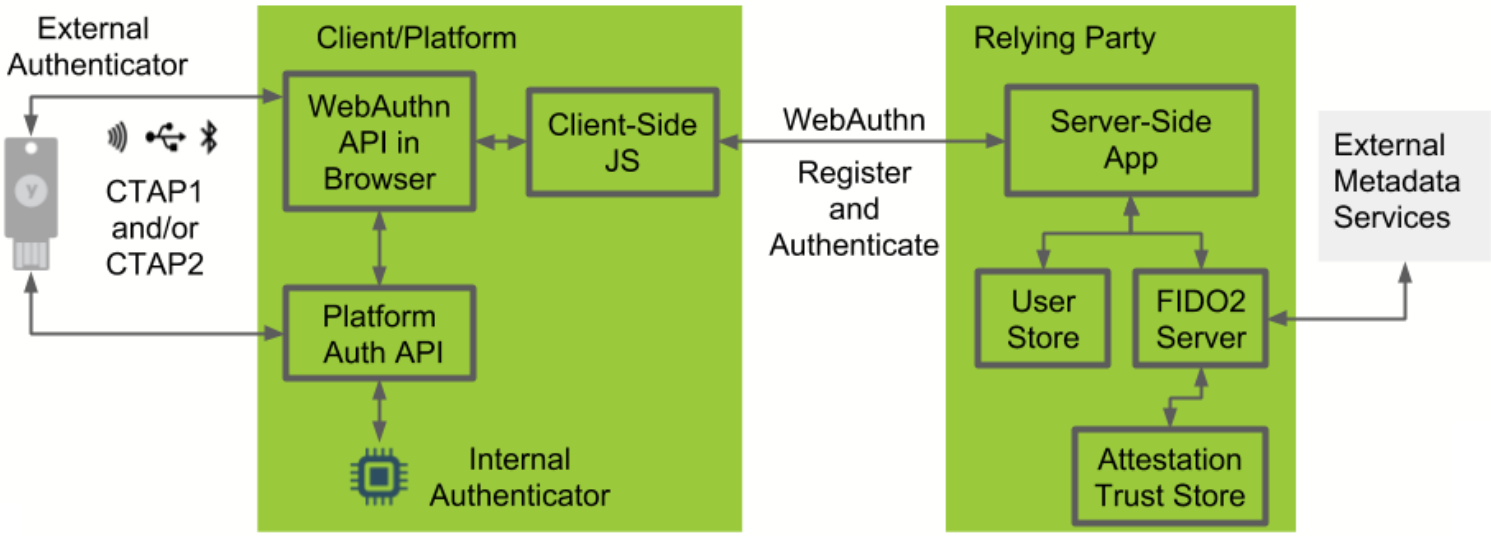
You can fool the human into thinking they are on the real website but you can't fool the browser and the authenticator



Industry standards that power the phishing resistant authenticators

- **CTAP2** (Client to Authenticator Protocol v2), a protocol for interacting with authenticator over USB, Bluetooth, and NFC is defined by the fido alliance an industry consortium
  - https://fidoalliance.org/
- **WebAuthn API -** a browser-based JavaScript API that your web application can use to interact with authenticators
  - 90.4% of browsers support WebAuthn https://caniuse.com/?search=webauthn
  - Your code calls WebAuthn API, the browser uses the CTAP2 protocol to talk to the authenticators

FIDO2 Application Architecture



Source: https://developers.yubico.com/WebAuthn/WebAuthn_Developer_Guide/fido2_app_architecture.png

Demo code for adding WebAuthn to Spring Boot

https://github.com/asaikali/devoxx-2023-passkey