qAuth: Mutual 2 Factor Authentication Using Trusted Services

David Patrzeba, David Karivalis, Jesse Ziegler
Advisor: Prof. Janne Lindqvist

Goal

- Provide a mutual 2 Factor Authentication scheme using current and emerging technologies and protocols.

Motivations and Objectives

Motivations
- Password reuse; a password compromised on one website potentially compromises a user across multiple sites.[1]
- Phishing attacks are sophisticated, taking advantage of human psychology in order to compromise user’s credentials.[3]
- Offline attacks on secure passwords (salted + hashed) are increasing successful: i.e., username/password is broken.[2]

Objectives
- Take advantage of existing and emerging technologies and protocols in order to deliver a usable 2FA scheme.
- Authenticate both the user to the service provider as well as the service provider to the end user.

Challenges

- Create a scalable client-server architecture.
- Provide a seamless 2nd factor of authentication with “something you have”: a Bluetooth connected accessory.
- Develop for Android Wear, a new platform released March 2014.

Methodology

- The Go Programming Language was used to rapidly prototype servers for the qAuth application.
- Android and Android Wear ecosystems were used to develop the proof of concept applications.
- TLS sessions are used to communicate securely between the qAuth client, server, and third party providers.
- RSA public key cryptography is used for identity binding.
- Trusted Bluetooth securely communicates between the qAuth client and Bluetooth accessory.

Results

- Built robust protocol to provide two factor authentication to existing third party services.
- The resulting product is a no-effort solution to consumers who want increased security.
- Design authenticates both the user and the service providing additional assurances to both sides.
- The design is modular.
  - The qAuth client could be a laptop instead of Android device.
  - The secure Bluetooth connection could come from a custom-made Bluetooth device instead of an Android watch.

Our Implementation

References