Users and Authentication

Daniel Zappala

CS 360 Internet Programming
Brigham Young University

Passwords

Passwords

- requirements
  - registration form
  - help user select good password
  - email recovery
  - secure password storage

Secure Password Storage

- assume the attacker WILL get your password database
  - do NOT store in plaintext
  - do NOT store with just a hash
    - rainbow table attack
    - huge table of precomputed password hashes
- concatenate password with a salt (random data that is unique for that user), then hash
- examples
  - ▸ node.js
  - ▸ flask
  - ▸ Rails

Why You Should Not Use Passwords

- users are bad at choosing passwords
- passwords have poor usability
  - hard to choose a good one and remember it (can use LastPass or 1Password)
  - users will often repeat passwords over multiple sites
  - designing a secure system is hard, attackers always getting better
  - security challenges are easily broken
  - only as secure as email, since email used for resetting passwords

Password Alternatives

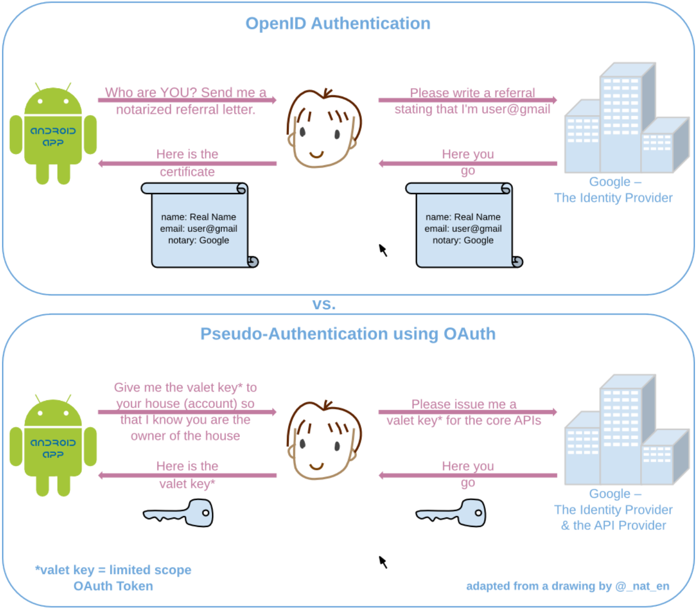
Email Authentication

- since it used for password recovery, why not just use email authentication?
    - email or text the user a token
    - ask them to supply token to login
    - can keep them logged in for several weeks using cookies
- see ▸ Citizen Budget for a simple example
- for a more secure version, see ▸ Simple Authentication for the Web, from BYU ISRL

OAuth

- trust your logins to a third-party service
- leverage trust of your users with these sites
- providers
  - Facebook, Twitter, Google, DropBox, FamilySearch, GitHub, LinkedIn, MailChimp, Steam, Tumblr, Yahoo...
- tools
  - ▸ Passport for node.js
  - ▸ OmniAuth for Rails, Sinatra
  - ▸ Flask-OAuthlib for Flask
  - ▸ Google+ Sign-In
  - ▸ Facebook Login

OAuth



OpenID Authentication

Who are YOU? Send me a notarized referral letter.

Please write a referral stating that I'm user@gmail

Here is the certificate

Here you go

name: Real Name
email: user@gmail
notary: Google

name: Real Name
email: user@gmail
notary: Google

Google –
The Identity Provider

vs.

Pseudo-Authentication using OAuth

Give me the valet key* to your house (account) so that I know you are the owner of the house

Please issue me a valet key* for the core APIs

Here is the valet key*

Here you go

Google –
The Identity Provider
& the API Provider

*valet key = limited scope
OAuth Token

adapted from a drawing by @_nat_en

Additional Reading

Additional Reading

- ▸ Authentication at Scale, Eric Grosse and Mayank Upadhyay, Google
  - device-centric authentication
  - smartcard-like USB token
- ▸ A Research Agenda Acknowledging the Persistence of Passwords, Cormac Herley and Paul C. van Oorschot, Microsoft Research and Carleton University
  - no silver bullet
  - passwords will be with us for a long time
  - need for research making them more secure