Users and Authentication

Daniel Zappala

CS 360 Internet Programming
Brigham Young University

Passwords

Passwords

- requirements
  - registration form
  - login form
  - email recovery
  - password policies and meters
  - 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
- see listomatic or node.js

Usability of Passwords

- users are bad at choosing good passwords
- users have to remember too many passwords
- users will often repeat passwords over multiple sites

Password Policies and Meters

- policies
  - length
  - types of characters (lowercase, uppercase, numbers, symbols)
  - avoid words in dictionary
  - expiration dates

An Administrator's Guide to Internet Password Research

An Administrator's Guide to Internet Password Research

An Administrators Guide to Internet Password Research, by Dinei Florencio, Cormac Herley, and Paul C. van Oorschot Microsoft Research published in Usenix LISA, November 2014

- examines the research literature on passwords and identifies what works, what does not work, and what remains unknown
- offers practical advice for system administrators

Categorizing Accounts

- **don't care**: no impact
  - one-time email, nuisance accounts for free articles
  - don't bother users about security of these passwords
- **low consequence**: minimal impact or easily repaired
  - social networks (infrequent users), discussion groups (infrequent users), online newspapers, accounts not storing credit cards
  - users may just rely on password reset
- **medium consequence**: limited loss (e.g. $50 cap on credit card loss)
  - secondary email account, online shopping sites, social network accounts (casual users), human resource sites
  - user effort resisting online attacks is well spent

Categorizing Accounts

- **high consequence**: critical accounts related to employment, finance, or important documents
  - primary or professional email accounts, social networks (heavy users and celebrities), online banking, SSH and VPN access, corporate databases
  - spend user effort securing passwords, provide two-factor authentication
- **ultra-sensitive**: major, life-altering, irreversible damage
  - multi-million dollar banking transactions, authorization to launch military weapons, encryption of national secrets
  - use something better than a password

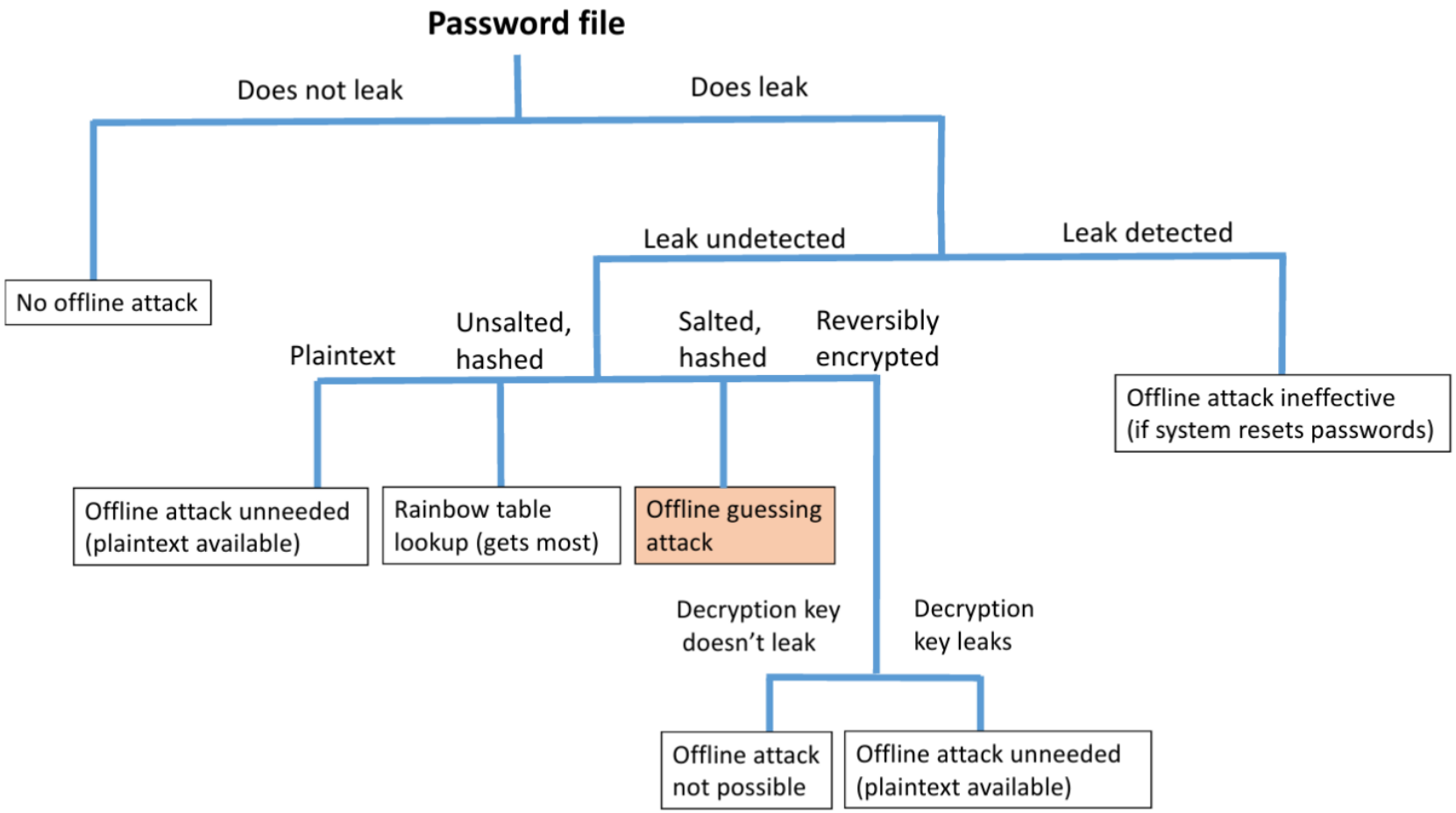
Password Strength

- examined leaked datasets from Rockyou, Gawker, Tianya, eHarmony, LinkedIn, Evernote, Adobe, Cupid Media
  - only Gawker and Evernote were hashed and salted
- ideally, users choose passwords randomly
- in practice, users choose common words (password, monkey, princess), proper nounds (julie, snoopy), and predictible sequences (abcdefg, asdfgh, 123456)
- metrics such as entropy are misleading
  - $L * log_2(C)$, L = length, C = size of alphabet
  - P@sswOrd is far more common than gunpyo, but has higher entropy
- *guessing resistance*: estimate of how many guesses needed to crack password

Online and Offline Guessing

- attacks on client don't involve guessing: malware, phishing, sniffing
- attacks on server's public facing web site: online guessing
- attacks on server's back end web site: offline guessing
  - gain access to system
  - be undetected (sysadmin can otherwise force system-wide password resets)
  - test passwords against hashes and salts

When an Offline Attack is Needed



How Many Guesses?

- determines how difficult a password must be to guess
- attackers can't make as many online guesses: need to be indistinguishable from ordinary traffic

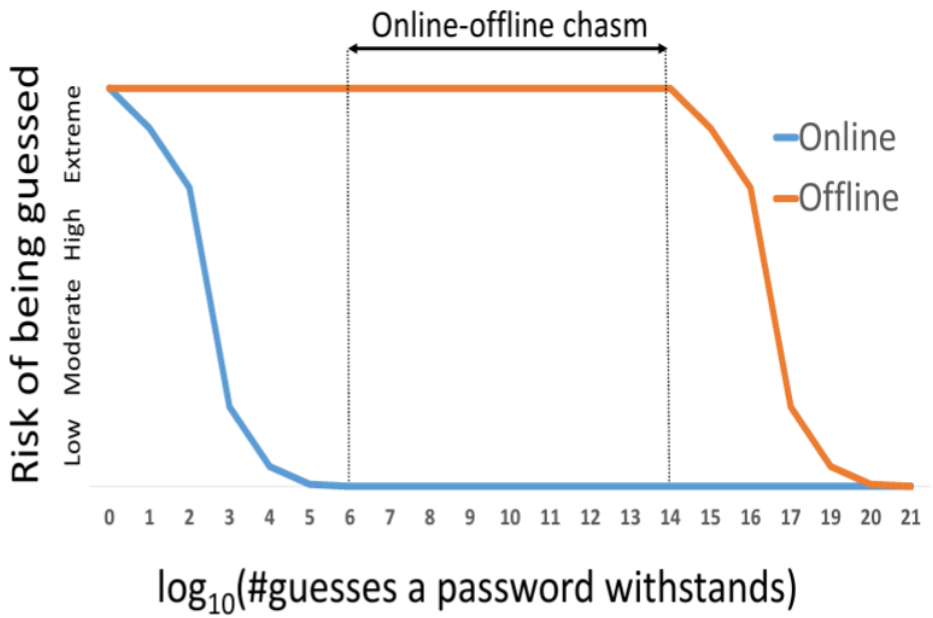


Figure 2: Conceptualized risk from online and offline guessing as a function of the number of guesses a password will withstand over a 4-month campaign. In the region from 10^6 to about 10^{14}, improved guessing-resistance has little effect on outcome (online or offline).

Policies

- composition and length
  - e.g. at least 8 characters, some uppercase and numbers
  - users respond with minimally compliant choices
  - overall, policies help to protect against online attacks, but not offline ones
  - users dislike them strongly
  - authors feel there are better alternatives

Policies

- blacklists: common passwords or leaked passwords
  - protects users most at risk
  - can ban most popular passwords used at your site
- expiration
  - only helpful for offline attacks
  - users choose highly predictable variants

Policies

- rate-limiting and lockout
  - lockout can be abused for denial of service attack
  - rate-limiting effective against online attacks
  - can require CAPTCHA for new IPs
- password meters
  - many JavaScript libraries are flawed and useless
  - need a stringent meter to have significant effect
  - do change user behavior for important accounts

Policies

- backup questions and reset
  - evidence shows that in practice the guessing space of security questions is far too small or can be looked up online
  - generally regarded a bad idea until more research done

Advice

- store passwords with salt and iterated hashing: slows offline guessing
  - detect leak and reset all user passwords
- rate limiting and lockout: reduces online guessing
- blacklisting: eliminates most-probable passwords
- length rules: use ≥ 8
- password meters: marginal gain
- password aging: more harm than good
- character-set rules: often bad return on user effort

Conclusions

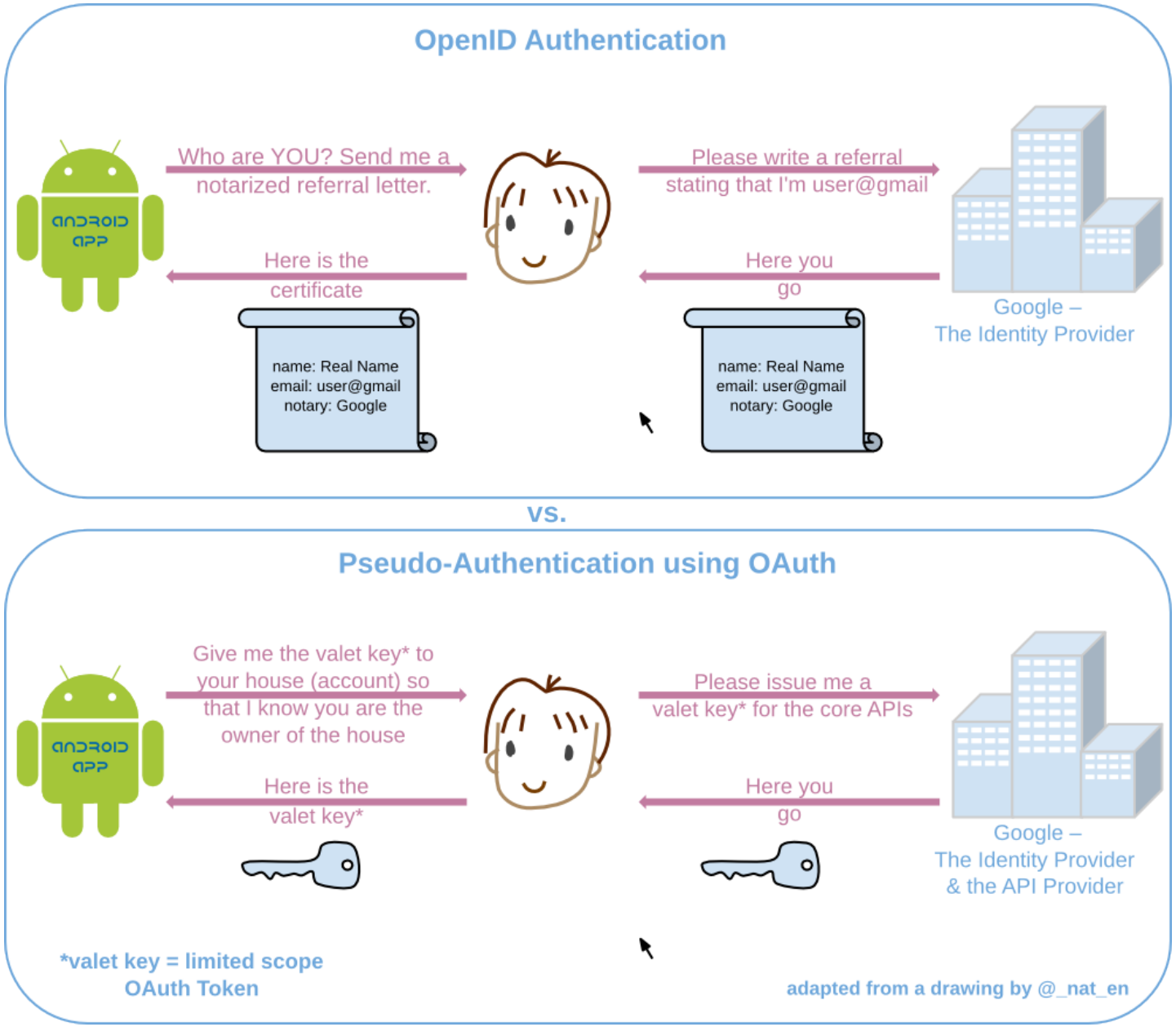
- we don't know how to help users resist offline password attacks – all attempts so far are failures
- failed attempts waste a lot of use effort
- the task gets harder every year
- zero-user-burden mechanisms that largely or entirely eliminate online attacks are rarely used

Password Alternatives

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...
- Passport for node.js
- 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

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

Authentication at Scale

Authentication at Scale, by Eric Grosse and Mayank Upadhyay, Google, published in IEEE Security and Privacy, 2013

- smartcard-like USB token
  - no software installation, simple and free registration, open standards
  - registration: browser calls an API to generate a public/private keypair for each site
  - login: browser calls an API to challenge device to prove it has the private key
  - USB drive, keychain, jewelry, smartphone
- channel bindings
  - would like to use client certificates, but users have not adopted these
  - with channel binding, generate a key pair for each new site
  - server binds cookies to client's public key
  - no user interaction, implemented in Chrome 24

The Quest To Replace Passwords

The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes, by Joseph Bonneau, Cormac herley, Paul C. van Oorschot, and Frank Stajano, published in IEEE S&P, 2012.

- examined various password replacements for usability, deployability, and security
  - password managers, one-time codes, single sign-on, graphical passwords, cognitive login, paper tokens, hardware tokens, mobile phones, and biometrics
- see table 1, page 11, in the paper
  - most schemes do better than passwords on security
  - some schemes do better and some worse on usability
  - every scheme does worse than passwords on deployability
- passwords will be hard to displace