Case Study: Building an Authentication System

Let’s Build an Authentication System

- Actually, let’s build two
- The first is for a social network; the second is for a bank
- We’ll do the social network first
- How should we authenticate?

What Should it Look Like?

Wrong!

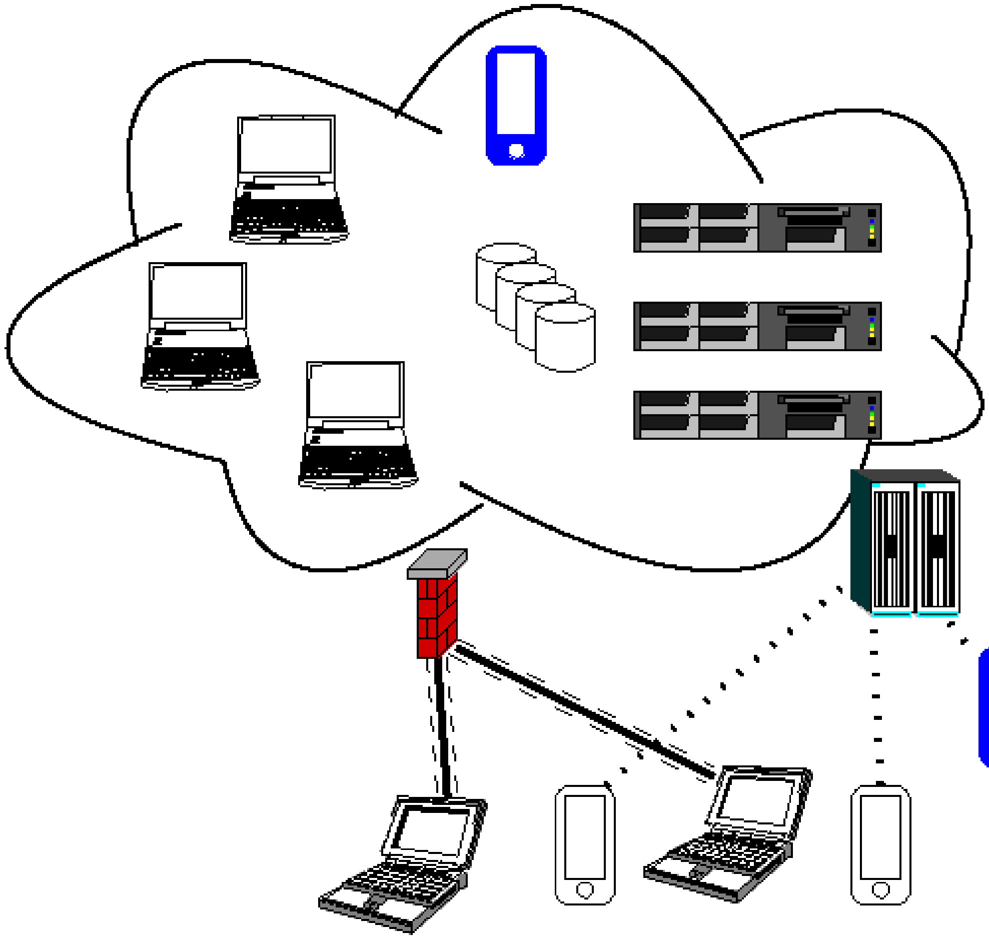
- We haven't answered our first question yet: what are we trying to protect, and against whom?
- We can't answer those until we know more about the service

The Social Network

Inside Developers, servers, database, social network admin, sysadmins

Border Firewall, web server

Outside Remote developers via VPN, users, social network admin



What Are the Resources to Protect?

What Are the Resources to Protect?

- User interaction database
- Code base
- Developers machines?
- Social network admin’s machines?
- Users’ machines?

Who Are the Attackers?

Who Are the Attackers?

- Joy hackers?
- Targeted mischief makers?
- Intelligence services?

Attackers

- Probably all of the above!
- Imagine what a hostile intelligence agency could do with the account of an important politician
- Just building the social graph is useful to intelligence agencies

☞ Do spooks socialize more with other spooks? Probably. . .

Choices

- Passwords or multi-factor?
☞ What is a good second factor?
- One authentication service or two?

Execution Environments for Different Actors

Developers Can view and modify the codebase

Sysadmins Can control more or less anything

Social network admins Can view and modify the user interaction database

Users Can interact, and can control own posts

Where Does Good Site Authentication Help?

	Interaction DB	Code Base	Developer Machines	Social Admin Machines	User Machines
Developers	?	✓?	✓	-	-
Rem. Devels	?	✓	?	-	-
Sysadmins	✓	✓	✓	✓	-
Social Admins	✓	-	-	✗	-
Users	✓	-	-	-	✗

✓ Strong site administration helps

? It might help, but there are other good attacker paths

✗ It doesn't help

- Site authentication is irrelevant

The Limits of Authentication

- The site doesn't control all authentication issues, e.g., users to their phones
- There is always the potential for hacking—and for some situations, it's more of a threat than stolen credentials
- Sysadmins have a lot of power (which we'll talk about later in the term)

Where Should Authentication Servers Live?

- Developer and sysadmins need a server inside the firewall
- Remote developers authenticate first to the firewall, and are then regular developers, i.e., inside
- Sysadmins are inside
- Users authenticate outside
- Social network administrators could be inside or outside

One Authentication Server or Two?

One Authentication Server or Two?

- You do not want outsiders consulting an internal authentication database—it's too sensitive
- It's safer for inside social network administrators to call out through the firewall

☞ They have to do that anyway, to reach the web server

- Conclusion: two authentication databases is probably the right path

Design Decisions

1. We will use two authentication systems

Simple Passwords or Multifactor Authentication?

Simple Passwords or Multifactor Authentication?

- Multifactor. . .
- We’ve seen far too many attacks on user social network accounts
- Other resources are even more important
- But—will users accept it?

Other Issues

- There are a number of other issues to consider, with varying tradeoffs
- Having two different authentication databases lets us make different tradeoffs for the different databases

Acceptance

- Who likes using MFA to log in to, e.g., Courseworks?
- Answer: no one, especially when it's in addition to a password
- Then why do we all use it?

☞ Because we have to!

- Will users accept a social network that *requires* multifactor authentication? Or will they go elsewhere?
- What is the *cost* of dealing with compromised user accounts? What is the *cost* of lost business?

Do Some Users Want MFA?

- Absolutely—they recognize the threat
- Many high-profile Twitter accounts have been compromised
- Example: when the Associated Press account was hacked and a fake tweet was sent about a bomb at the White House and Obama being injured, the DowJones average dropped 1% in seconds

There May Be Regulatory Issues

- A Twitter administrative account was once hacked because they didn't use MFA and an admin stored his Twitter password in his gmail account—and that fell to password-guessing
- Another time, an administrative account was brute-forced via online guessing
- Fake tweets were sent from a variety of user accounts, including President Obama and Fox News
- MFA would have prevented these attacks
- The Federal Trade Commission acted. . .

The FTC: (Some) Problems at Twitter

- Require employees to use hard-to-guess administrative passwords that they did not use for other programs, websites, or networks;
- Suspend or disable administrative passwords after a reasonable number of unsuccessful login attempts;
- Provide an administrative login webpage that is made known only to authorized persons and is separate from the login page for users;
- Enforce periodic changes of administrative passwords
- Impose other reasonable restrictions on administrative access, such as by restricting access to specified IP addresses.

Multifactor Authentication

- There's a strong argument for making it mandatory for employees
- There's a strong argument for making it available, but perhaps not mandatory, for users

Design Decisions

Developers

1. MFA use should be required

Social Network Users

1. MFA should be available. Note that social network admins are employees, another reason for that interface to support it

Types of Authentication: Employees

- Should we use passwords as one factor? Probably.
- What’s second, biometrics or some sort of token?
- And what sort of biometric or token?

What Type of Biometric?

What Type of Biometric?

- Note: I’m talking about a login biometric, not a phone-unlock one
- Only two types appear vaguely suitable, facial recognition (using laptop cameras) or fingerprints
- But laptop fingerprint readers are rare, and when present are used for device unlock
- And what about remote users?

☞ Biometrics do not appear suitable

- Conclusion: we need a token as a second factor

What Type of Token?

- There are dedicated TOTP tokens—but do they offer enough security advantages over a phone-based soft token?
- U2F—Universal Second Factor
- Text message

What is U2F?

- Supports the industry-standard FIDO2 authentication protocol
- Supported by all major browsers
- Usable for login on MacOS, Linux, Windows 10
- Provides cryptographic authentication and prevents MitM attacks
- But—users have to carry extra hardware with them

TOTP Tokens

- Pretty good, but doesn't prevent MitM attacks
- But soft tokens are cheap, and almost everyone has a smartphone (and probably *every* developer)
- A good choice, but not as good as U2F

Text Messages

- Rapidly falling out of favor
- Issue: SIM-jacking
- Issue: SS7 hijacking
- Both attacks seen in the real world (especially if governments are the attackers)

Phone Cameras

- Have the login screen display a QR code
- A phone app reads the code and generates a TOTP
- Cheap!
- Like TOTP, but without MitM issues
- But—doesn't authenticate phone logins; U2F can

Cost, Writ Large

- What do these solutions cost?
- U2F tokens cost money, but not very much compared to the loaded cost of a developer
- Soft tokens cost even less
- Camera-based TOTP should be low, too
- But—what does the software cost?

Software Costs

- What does the company-side software cost?
- Look at the full picture: support for all platforms, administration platform, provisioning, database backup, logging and auditing, and more
- Remember that people cost much more than hardware
- The big reason that the SecurID card did so well in the market is that they provided a full software suite

☞ Software costs suggest that supporting more than one scheme may be infeasible

Which Second Factor for Users?

- U2F seems very secure, but you probably can't afford to buy them for your users
- You could add support for users who buy their own
- You could also add TOTP support
- Camera plus TOTP? Maybe, but you might have to write the app
- Text message? Far better than just a password, but not strong enough against serious attackers
- Do you have to pay twice for your software package?
- If your users log in from their phones, how is a phone a second factor?

Conclusion: Tokens for Employees

- U2F is the strongest; TOTP the cheapest, and is secure enough for local logins
- (Some U2F tokens even work with phones)
- Need to balance cost against threats
- The rapid growth of host and browser support for U2F is encouraging, but watch out for the support software costs

Conclusion: Tokens for Users

- If you think users may be targeted by serious actors, support U2F
- Btw: you probably need U2F support anyway, for your social network admins
- TOTP is pretty good, but is vulnerable to phishing attacks

Design Decisions

Developers

1. MFA use should be required, including for social network admins
2. U2F is probably the best choice

Social Network Users

1. MFA should be available
2. U2F support is needed for employees; TOTP with soft tokens is more accessible to most users

Lost Credential Recovery

- What about recovery from lost credentials: forgotten passwords, lost U2F tokens, phone problems?
- You have to recover, and recovery *securely*
- Local employees are easy: have them show their badge to the help desk (or equivalent), or let their manager request/authorize a replacement
- What about remote employees? Users?

Remote Employees

- A *very* difficult problem
- Can you overnight a new token to them? What if they're on the road?
- What is the tradeoff between cost and lost productivity?
- Do you fall back to secondary authentication? Not very secure!
- TOTP via a phone app might be a good fallback: use secondary authentication to the help desk to enable that for that employee for a limited time
- But that's more software complexity
- There are no perfect answers!

Lost Credential Recovery: Users

- *Very* difficult and *utterly* mandatory
- Common fallback: email for password recovery
- But what about a lost U2F token?
- A password plus email? Many people will use the same password for email and your service
- Or maybe their password for your service is stored in their email account
- Worst of all: you can't afford to spend much on recovery

Bootstrapping Authentication

- Ultimately, it’s a cost/benefit tradeoff
- Email plus password is probably the best you can do for a lost U2F or TOTP credential
- Possibly—though setting it up and administering it is expensive—have a paid support tier (but except for popular services, very few will take advantage of it)

Authentication Architecture

- Where do we put authentication databases?
- Developers: inside the firewall
- But—*really* lock down the machine
- What about the user authentication database?

User Authentication Database

- Clearly, inside the firewall
- There is also a database of user profiles
- Do we put authentication data in the same database?
- Remember: if you have two databases, they *will* get out of sync

Separate Databases!

- The user database has far more kinds of activity and hence has a larger attack surface
- There are many activities that will write to it
- ☞ It is in the execution environment of more processes
- So: put authentication data in a separate database
- (More design details later in the term)

Design Decisions

Developers

1. MFA use should be required, including for social network admins
2. U2F is probably the best choice
3. Internal, locked-down database
4. Recovery via management chain and overnight shipping

Social Network Users

1. MFA should be available
2. U2F support is needed for employees; TOTP with soft tokens is more accessible to most users
3. Separate database for authentication only
4. Recovery via email, plus password for token loss

What About Banks?

- Most of the analyses are the same
- However: more is at risk, but there's more money to spend solving the problems
- Should MFA be mandatory?
- Hard—it's a competitive market
- Query: who is liable for financial losses, the customer or the bank?

Liability

- Under US law, consumers are generally not liable; businesses are
- But—consumers generally have far less money, so the bank's losses are limited
- Always offer strong MFA—remember that phishing is a serious problem, so MitM resistance is important
- Perhaps given U2F tokens to high-value customers and business customers
- Let people visit a branch for replacement tokens
- Or: mail or overnight replacements—you always have physical addresses
- (Might a serious attacker—or disgruntled former spouse/partner—stake out the mailbox?)

Summary

- Authentication is a systems problem
- It’s much more complex than just “use passwords” or “use MFA”
- There are tradeoffs, and there are problems with no great solutions