Last time

- User Authentication
  - Authentication Factors
  - Passwords
  - Attacks on Passwords

This time

- User Authentication
  - Beyond passwords
  - Biometrics
- Security Policies and Models
  - Trusted Operating Systems and Software
  - Military and Commercial Security Policies

Interception Attacks

- Attacker intercepts password while it is being transmitted to website
- One-time passwords make intercepted password useless for later logins
  - In a challenge-response protocol, the server sends a random challenge to the client
  - Client uses challenge and password as an input to a function and computes a one-time password
  - Client sends one-time password to server
  - Server checks whether client’s response is valid
  - Given intercepted challenge and response, attacker might be able to brute-force password
- Cryptographic protocols (e.g., SRP) make intercepted information useless to an attacker

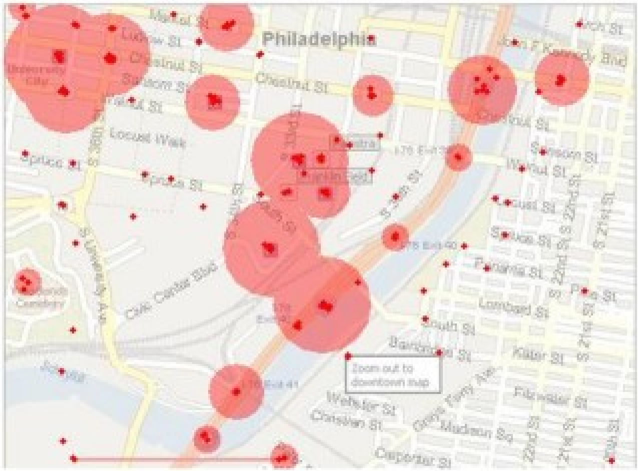
Interception Attacks

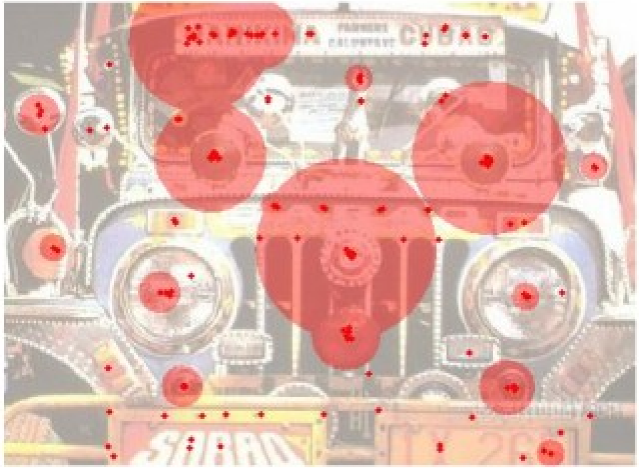
- Proposed solutions are difficult to deploy
  - Patent issues
  - Changes to HTTP protocol required (i.e., every browser and many servers would have to be changed)
  - Challenge-response functions need to be irreversible, but also computable by humans for easy deployment, which makes them rare

Graphical Passwords

- Graphical passwords are an alternative to text-based passwords
- Multiple techniques, e.g.,
  - User chooses a picture; to log in, user has to re-identify this picture in a set of pictures
  - User chooses set of places in a picture; to log in, user has to click on each place
- Issues similar to text-based passwords arise
  - E.g., choice of places is not necessarily random
- Shoulder surfing becomes a problem
- Ongoing research

Graphical Passwords





Server authentication

- With the help of a password, system authenticates user (client)
- But user should also authenticate system (server) so that password does not end up with attacker instead!
- Classic attack:
  - In a computing lab, have a program display a fake login screen
  - When user "logs in", programs prints error message, sends captured user ID and password to attacker and ends current session (which will start actual login screen)
  - That's why Windows requires you to press <CTRL-ALT-DELETE> for login. Always gives login window and cannot be overridden
- Today's attack:
  - Phishing

Biometrics

- Biometrics have been hailed as a way to get rid of the problems with password and token-based authentication
- Unfortunately, they have their own problems
- Idea: Authenticate user based on physical characteristics
  - Fingerprints, iris scan, voice, handwriting, typing pattern,…
- If observed trait is sufficiently close to previously stored trait, accept user
  - Observed fingerprint will never be completely identical to a previously stored fingerprint of the same user
- Biometrics work well for local authentication, but are less suited for remote authentication or for identification

Local vs. Remote Authentication

- In local authentication, a guard can ensure that:
  - I put my own finger on a fingerprint scanner, not one made out of gelatin
    - Watch corresponding MythBusters episode on YouTube
  - I stand in front of a camera and don’t just hold up a picture of somebody else
- In remote authentication, this is much more difficult

Authentication vs. Identification

- Authentication: Does a captured trait correspond to a particular stored trait?
- Identification: Does a captured trait correspond to any of the stored traits?
  - Identification is an (expensive) search problem, which is made worse by the fact that in biometrics, matches are based on closeness, not on equality (as for passwords)
- False positives can make biometrics-based identification useless
  - False positive: Alice is accepted as Bob
  - False negative: Alice is incorrectly rejected as Alice

Biometrics-based Identification

- Example (from Bruce Schneier's "Beyond Fear"):
  - Face-recognition software with (unrealistic) accuracy of 99.9% is used in a football stadium to detect terrorists
    - 1-in-1,000 chance that a terrorist is not detected
    - 1-in-1,000 chance that innocent person is flagged as terrorist
  - If one in 10 million stadium attendees is a known terrorist, there will be 10,000 false alarms for every real terrorist
  - Remember "The Boy Who Cried Wolf"?
- After pilot study, German FBI recently concluded that this kind of surveillance is useless
  - Average detection accuracy was 30%

Other Problems with Biometrics

- Privacy concerns
  - Why should my employer (or a website) have information about my fingerprints, iris,..?
    - Aside: Why should a website know my date of birth, my mother's maiden name,… for "secret questions"?
  - What if this information leaks? Getting a new password is easy, but much more difficult for biometrics
- Accuracy: False negatives are annoying
  - What if there is no other way to authenticate?
  - What if I grow a beard, hurt my finger,…?

Trusted Operating Systems

- Trusting an entity means that if this entity misbehaves, the security of the system fails
- We trust an OS if we have confidence that it provides security services, i.e.,
  - Memory and file protection
  - Access control and user authentication
- Typically a trusted operating system builds on four factors:
  - Policy: A set of rules outlining what is secured and how
  - Model: A model that implements the policy and that can be used for reasoning about the policy
  - Design: A specification of how the OS implements the model
  - Trust: Assurance that the OS is implemented according to design

Trusted Software

- Software that has been rigorously developed and analyzed, giving us reason to trust that the code does what it is expected to do and nothing more
- Functional correctness
  - Software works correctly
- Enforcement of integrity
  - Wrong inputs don't impact correctness of data
- Limited privilege
  - Access rights are minimized and not passed to others
- Appropriate confidence level
  - Software has been rated as required by environment
- Trust can change over time, e.g., based on experience

Security Policies

- Many OS security policies have their roots in military security policies
  - That’s where lots of research funding came from
- Each object/subject has a sensitivity/clearance level
  - “Top Secret” > “Secret” > “Confidential” > “Unclassified” where “>” means “more sensitive”
- Each object/subject might also be assigned to one or more compartments
  - E.g., “Soviet Union”, “East Germany”
  - Need-to-know rule
- Subject s can access object o iff level(s) $\geq$ level(o) and compartments(s) $\supseteq$ compartments(o)
  - s dominates o, short “s $\geq$ o”

Example

- Secret agent James Bond has clearance “Top Secret” and is assigned to compartment “East Germany”
- Can he read a document with sensitivity level “Secret” and compartments “East Germany” and “Soviet Union”?
- Which documents can he read?

Commercial Security Policies

- Rooted in military security policies
- Different classification levels for information
  - E.g., external vs. internal
- Different departments/projects can call for need-to-know restrictions
- Assignment of people to clearance levels typically not as formally defined as in military
  - Maybe on a temporary/ad hoc basis

Other Security Policies

- So far we've looked only at confidentiality policies
- Integrity of information can be as or even more important than its confidentiality
  - E.g., Clark-Wilson Security Policy
  - Based on well-formed transactions that transition system from a consistent state to another one
  - Also supports Separation of Duty (see RBAC slides)
- Another issue is dealing with conflicts of interests
  - Chinese Wall Security Policy
  - Once you've decided for a side of the wall, there is no easy way to get to the other side

Chinese Wall Security Policy

- Once you have been able to access information about a particular kind of company, you will no longer be able to access information about other companies of the same kind
  - Useful for consulting, legal or accounting firms
  - Need history of accessed objects
  - Access rights change over time
- ss-property: Subject s can access object o iff each object previously accessed by s either belongs to the same company as o or belongs to a different kind of company than o does
- *-property: For a write access, we also need to ensure that all objects readable by s either belong to the same company as o or have been sanitized

Example

- Fast Food Companies = {McDonalds, Wendy’s}
- Book Stores = {Chapters, Amazon}
- Alice has accessed information about McDonalds
- Bob has accessed information about Wendy’s
- ss-property prevents Alice from accessing information about Wendy’s, but not about Chapters or Amazon
  - Similar for Bob
- Alice could write information about McDonalds to Chapters and Bob could read this information from Chapters
  - Indirect information flow violates Chinese Wall Policy
  - *-property forbids this kind of write

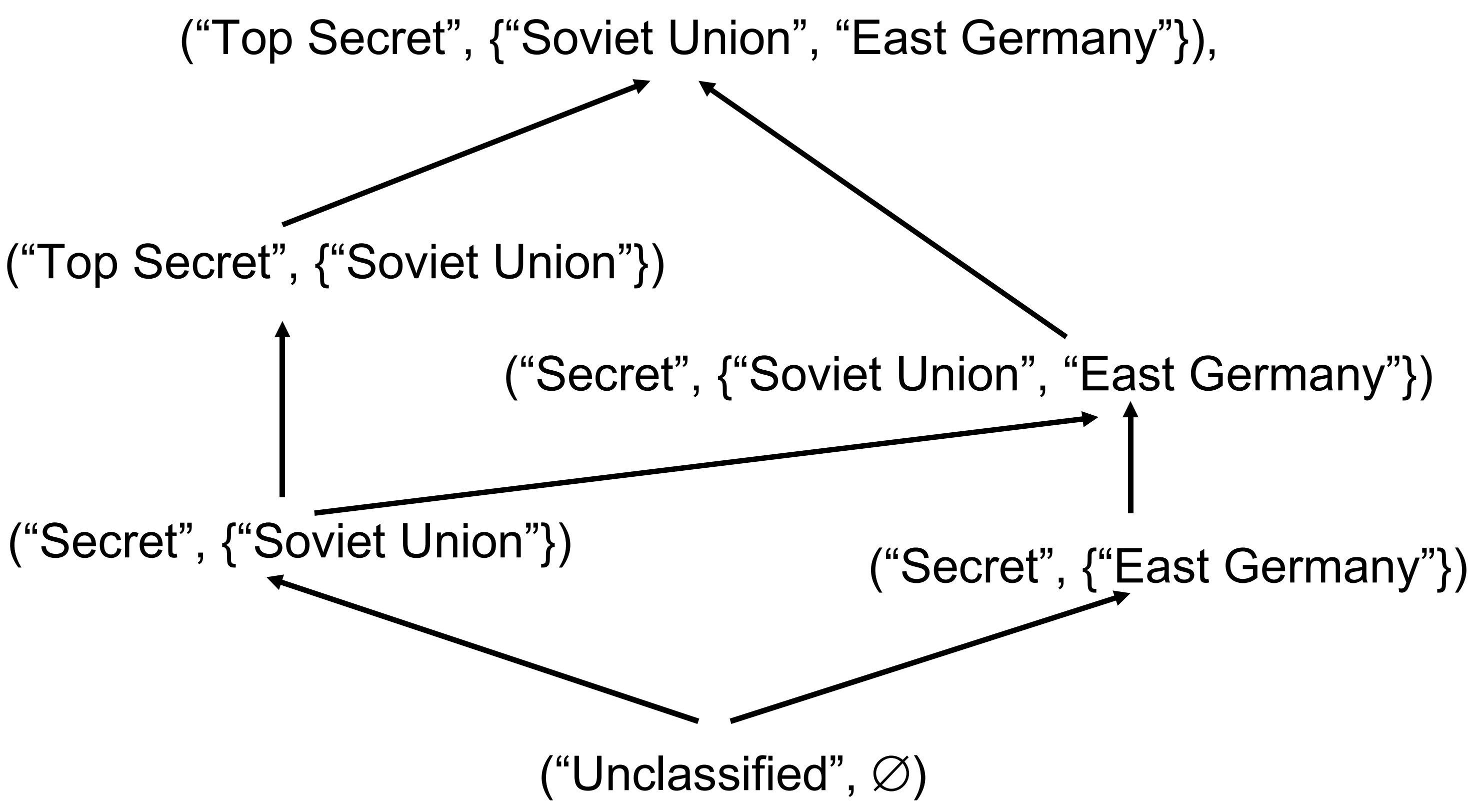
Security Models

- Many security models have been defined and interesting properties about them have been proved
- Unfortunately, for many models, their relevance to practically used security policies is not clear
- We'll focus on two prominent models
  - Bell-La Padula Confidentiality Model
  - Biba Integrity Model
  - See text for others
- Targeted at Multilevel Security (MLS) policies, where subjects/objects have clearance/classification levels

Lattices

- Dominance relationship ≥ defined in military security model is transitive and antisymmetric
- Therefore, it defines a lattice
- For two levels a and b, neither a ≥ b nor b ≥ a might hold
- However, for every a and b, there is a lowest upper bound u for which u ≥ a and u ≥ b and a greatest lower bound l for which a ≥ l and b ≥ l
- There are also two elements U and L that dominate/are dominated by all levels
  - In example,
    U = (“Top Secret”, {“Soviet Union”, “East Germany”})
    L = (“Unclassified”, ∅)

Example Lattice



Recap

- User Authentication
  - Beyond passwords
  - Biometrics
- Security Policies and Models
  - Trusted Operating Systems and Software
  - Military and Commercial Security Policies

Next time

- Security Policies and Models
  - Bell La-Padula and Biba Security Models
  - Information Flow Control

- Trusted Operating System Design
  - Design Elements
  - Security Features
  - Trusted Computing Base
  - Least Privilege in Popular OSs
  - Assurance