CSE 410/565 Computer Security
Spring 2022

Lecture 7: Authentication

Department of Computer Science and Engineering
University at Buffalo

Lecture Outline

- Definition of entity authentication
- Solutions
  - password-based authentication
  - token-based authentication
  - biometric-based authentication
- Stronger forms of secure authentication

Entity Authentication

- Authentication is a broad term and is normally referred to mechanisms of ensuring that
  - entities are who they claim to be
  - data has not been manipulated by unauthorized parties
- Entity authentication or identification refers to the means of verifying user identity
  - if such verification is successful, the user is granted appropriate privileges
- The need for user authentication in early computer systems arose once it became possible to support multi-user environments

Entity Authentication

- During an authentication protocol:
  - one party, the verifier, gathers evidence that the identity of another party, the claimant, is as claimed
- Goals of authentication protocols:
  - honest parties should be able to successfully finish the protocol with their identity accepted as authentic
  - it should be difficult for dishonest parties to impersonate an identity of another user
  - impersonation must remain difficult even after observing a large number of successful authentications by other parties
- User registration is required prior to an authentication protocol

Entity Authentication

- Identification mechanisms are often divided into 3 types based on how the identity evidence is gathered
  - user knows a secret
    - examples include passwords, personal identification numbers (PINs), secret keys, mother's maiden name, etc.
  - user possesses a token
    - these are normally hardware tokens such as magnetic-striped cards or custom-designed devices for time-variant passwords
  - user has a physical attribute
    - characteristics inherent to the user such as biometrics, handwritten signatures, keystroke dynamics, facial and hand geometries, voice, etc.

Entity Authentication

- Often, different types can be combined together
  - e.g., PIN-based authentication is often used with a physical device (user ID, credit card)
  - biometric-based authentication is often used in combination with a password or a physical token
- Many identification mechanisms used in practice are not secure
  - calling cards
  - credit card purchases
  - passwords
- Ideally we want solutions against which replay attacks don't work

Password-Based Authentication

- A password is a string of (normally 8 or more) characters associated with a certain user
  - it serves the purpose of a shared secret between the user and the system
- During the identification protocol:
  - a user sends (*userid, password*) pair
    - *userid* identifies the user
    - *password* provides the necessary evidence that the user possesses the secret
  - the system compares that information with its has stored
  - if the check succeeds, access is granted

Password-Based Authentication

- Storage of passwords
  - the most straightforward way of storing passwords is in clear text
    - there is a problem with such approach
  - to mitigate it, most systems apply a one-way hash function to a password and store the hash
    - the password itself cannot be recovered, but there are other concerns
- Attacks on passwords
  - replay of passwords: an attacker reuses a captured password
    - an attacker can capture a password by seeing a user type it, using a keylogger program or obtaining it in transit

Password-Based Authentication

- Attacks on passwords (cont.)
  - exhaustive search: an attacker attempts to guess a user password by trying all possible strings
    - this can be done on the verifier itself or by obtaining a copy of the password file and performing the attack off-line
    - often the attack is infeasible if the password space is large enough
      - but it is still possible to exhaust all short passwords
  - dictionary attack: an attacker tries to guess a password using words from a dictionary and variations thereof
    - can have a high probability of success
    - dictionary attacks become increasingly sophisticated

Password-Based Authentication

- Is there a way to decrease the vulnerability of the system to such attacks?
- Additional measures are normally employed, some of which are:
  - salting passwords
    - this technique makes guessing attacks less effective
    - a password is augmented with a random string, called salt, prior to hashing
    - the salt is stored in cleartext in the password file

$$uid_1, salt_1, h(salt_1||pwd_1)$$
$$uid_2, salt_2, h(salt_2||pwd_2)$$

    - how does it improve security?

Password-Based Authentication

- Measures for improving security of passwords (cont.)
  - slowing down password verification
    - the hash function for password verification is made more computationally extensive
    - this can be done, e.g., by iterating the computation n times
    - what is its drawback?
  - limiting the number of unsuccessful password guesses
    - a user account is locked after the number of successive unsuccessful authentication attempts exceeds the threshold
  - employing password rules
    - additional rules on password choices are imposed
    - this often strengthens password choices but limits the search space

Password-Based Authentication

- Measures for improving security of passwords (cont.)
  - preventing direct access to password file
    - the file/database with hashed passwords is kept inaccessible by ordinary users
- Another technique that aims at improving security of passwords is called password aging
- It is always a challenge to find a balance between memorability of passwords and their resistance to dictionary attacks
  - do users make acceptable password choices?
  - can we help them with choosing strong passwords?

Password-Based Authentication

- Password strength has been studied since 1990s
  - a significant portion of used passwords is guessable
    - passwords of short length can be cracked using brute force search
    - account-related or dictionary-derived passwords are common
  - password crackers today are increasingly complex
- How can we help users to select stronger passwords?
  - systems are much better at helping users than before
  - a variety of tools exist

Password-Based Authentication

- Tools for choosing stronger passwords
  - computer-generated passwords
    - selecting less predictable passwords which users can remember can be done by using computer-generated pronounceable passwords
    - for example: heloberi, hoparmah, ulensoev, atonitim
  - password checking
    - a proactive password checker rates password strength at the time of password selection
  - other types of passwords
    - techniques for using images and graphical interfaces for authentication have been developed

Password-Based Authentication

- Tools for choosing stronger passwords (cont.)
  - image-based passwords and graphical interfaces
    - displaying a sequence of images
    - drawing patterns on a grid
    - choosing points using an image
    - their unpredictability is often not as great as desired
- Unpredictability and usability of passwords is hard to achieve simultaneously
  - passwords can provide only a weak form of security

Best Password Practices

- NIST's Special Publication 800-63 provides authentication guidelines for organizations including password-based authentication
  - the latest version is dated by June 2017
- In general, you want to
  - use strong passwords
  - not reuse passwords across different services
  - not share your passwords with anyone else
- Password managers are of great help in dealing with password explosion

Remote Authentication

- Now assume we want to use passwords for remote authentication
  - will it work?
- Passwords observed on the network are trivially susceptible to replay
  - initially remote login and file transfer programs, such as `telnet`, communicated passwords in the clear
  - now encryption is used (`ssh`, `scp`, etc.)
- Authentication based on time-invariant passwords is therefore a weak form of authentication
  - this form of authentication is nevertheless the most common
- A natural way to improve security is to use one-time passwords

One-Time Passwords

- In authentication based on one-time passwords each password is used only once
- Such authentication can be realized in the following ways:
  - the user and the system initially agree on a sequence of passwords
    - simple solution but requires maintenance of the shared list
  - the user updates her password with each instance of the authentication protocol
    - e.g., the user might send the new password encrypted under a key derived from the current password
    - this method crucially relies on the correct communication of the new password to the system

One-Time Passwords

- One-time password authentication mechanisms (cont.)
  - the new password is derived with each instance of the authentication protocol using a one-way hash function
    - the system based on hash chains is called S/Key and is due to Lamport
    - a user begins with secret k and produces a sequence of values $k, h(k), h(h(k)), \ldots, h^t(k)$
    - password for ith identification session is $k_i = h^{t-i}(k)$
    - when user authenticates $(i+1)$st time with k_{i+1}, the server checks whether $h(k_{i+1}) = k_i$
    - if h is infeasible to invert, this convinces the server that the user is legitimate

One-Time Passwords

- Example of S/Key
  - suppose $t = 5$
  - at setup stage
    - user chooses k and computes $h(k), h(h(k)), h^3(k), h^4(k), h^5(k)$
    - uses gives $h^5(k)$ to the verifier
  - during authentication
    - at session 1:
    - at session 2:
    - at session 5:

Entity Authentication

- An even stronger form of authentication is one where the user doesn't have to send the secret to the verifier
  - ideally you want to convince the verifier without leaking information about your secret
  - such solutions exist and often involve the verifier sending a random challenge to the claimant
  - the claimant uses the challenge and the secret to compute the response
  - anyone who monitors the channel, cannot deduce information about the secret

Challenge-Response Techniques

- The goal of challenge-response techniques is to
  - use a single secret for authentication
  - provide evidence of the secret without leaking information about it
  - proving possession of a secret without leaking information about it is called a zero-knowledge proof of knowledge
- Challenge-response protocols can be built
  - from simple cryptographic primitives (e.g, MACs and signature schemes)
  - from scratch (Schnorr, Okamoto, and Guillou-Quisquater schemes)

Challenge-Response Techniques

- The basic form of such protocols is normally as follows:
  - suppose Alice is authenticating to Bob
  - Alice has a secret s and Bob has a verification value v
  - Bob sends to Alice a challenge c (chosen or computed anew)
  - Alice computes a response $r = f(s, c)$ and sends it to Bob
  - Bob verifies r using c and v
- Building a secure challenge-response protocol is non-trivial
  - must be secure against active adversaries
    - parallel session attack
    - man-in-the-middle attack

Authentication based on Secrets

- If passwords are such a poor way of authenticating, why are they so popular?

Token-Based Authentication

- Authentication based on what you possess can be done using different types of tokens
  - memory cards
    - data is passively stored on a medium
    - a card reader can retrieve information stored on the card
    - e.g., magnetic stripe credit cards, ATM cards, hotel keys
    - memory cards provide a limited level of security (i.e., card contents can be read by any reader and copied to another card)
    - memory cards are often combined with a password or PIN
    - using memory cards with computers requires special readers

Token-Based Authentication

- Types of authentication tokens (cont.)
  - smart cards
    - such cards have a built-in microprocessor, programmable read-only memory and random-access memory (RAM)
    - they can engage in different types of authentication protocols including challenge-response
    - such tokens can also be used to generate dynamic passwords
      - each minute the device generates a new password
      - the device and the verifier must be synchronized
    - tamper-resistance of such tokens must be addressed
      - it's been shown in the past that key material can be recovered with relatively inexpensive equipment

Token-Based Authentication

- Types of authentication tokens (cont.)
  - USB dongle
    - USB tokens can also be used for authentication
    - they can store static data as well as code
      - recent dongles also include non-volatile memory
    - no additional hardware such a special-purpose reader is necessary
    - USB dongles are commonly used for copy protection of copyrighted material
    - dongle products often don't provide enough security to be used in rigid security requirement environments

Biometric Authentication

- Biometric authentication systems authenticate an individual based her physical characteristic

- Types of biometric used in authentication
  - face
  - palm geometry
  - fingerprint
  - iris
  - signature
  - voice

- Most common uses of biometric authentication is for specific applications rather than computer authentication

Biometric Authentication

- Like other authentication mechanisms, biometric authentication includes an enrollment phase during which a biometric is captured
  - the initial reading is often called a template
  - at authentication time, a new biometric reading is performed and is compared to the stored template
- Unlike other authentication mechanisms, biometric matching is approximate
  - each reading can be influenced by a variety of factors
    - e.g., light conditions, facial expressions, hair style, glasses, etc. for face recognition
  - some types of biometrics can match more accurately than others
    - e.g., iris vs. face or palm

Biometric Authentication

- Biometric matching can be used to perform
  - verification
    - user's biometric scan is used to match her own template only
  - identification
    - user's biometric scan is used to match a database of templates
- Identification might not always be possible
- Biometric systems attempt to minimize
  - false reject rate: authentic biometric is rejected
  - false accept rate: imposter biometric is accepted
- Depending on the environment, minimizing one of them might be more important than minimizing both

Biometric Authentication

- New types of biometrics are being explored
  - brain waves, heart beats, etc.
- Many forms of traditional biometrics can be stolen
- Static biometrics can be replayed

Biometric Authentication

- Current research direction: biometric key generation
  - the idea: a biometric can be used to generate a cryptographic key
  - the key can be reproduced using another biometric close enough to the original
    - no need to remember any information such as a password
  - the key can be used for authentication or encryption
  - key generation algorithm produces a helper data that can later aid in recovering the same key from a noisy version of the biometric
  - security requirements are strict
    - the helper data must leak minimal information about the biometric
    - compromise of the key must not lead to recovery of the biometric

Summary

- Entity authentication is an important topic with the main application in access control
- Various techniques exist ranging from time-invariant passwords to provably secure identification schemes
- Despite the weak security password-base authentication provides, it is the most widely used authentication mechanism
  - ease of use, user familiarity, no infrastructure requirements
- Next time
  - access control mechanisms