Lecture 12
October 23, 2023
Challenge-Response

User, system share a secret function $f$ (in practice, $f$ is a known function with unknown parameters, such as a cryptographic key)

- User requests to authenticate
- System sends a random message $r$ (the challenge)
- User sends $f(r)$ (the response)

Slide 2
One-Time Passwords

• Password that can be used exactly once
  • After use, it is immediately invalidated

• Challenge-response mechanism
  • Challenge is number of authentications; response is password for that particular number

• Problems
  • Synchronization of user, system
  • Generation of good random passwords
  • Password distribution problem
S/Key

• One-time password scheme based on idea of Lamport
• $h$ one-way hash function (SHA-256, for example)
• User chooses initial seed $k$
• System calculates:
  \[ h(k) = k_1, \ h(k_1) = k_2, \ldots, \ h(k_{n-1}) = k_n \]
• Passwords are reverse order:
  \[ p_1 = k_n, \ p_2 = k_{n-1}, \ldots, p_{n-1} = k_2, \ p_n = k_1 \]
S/Key Protocol

System stores maximum number of authentications $n$, number of next authentication $i$, last correctly supplied password $p_{i-1}$.

System computes $h(p_i) = h(k_{n-i+1}) = k_{n-i} = p_{i-1}$. If match with what is stored, system replaces $p_{i-1}$ with $p_i$ and increments $i$. 
Hardware Support

• Token-based
  • Used to compute response to challenge
    • May encipher or hash challenge
    • May require PIN from user

• Temporally-based
  • Every minute (or so) different number shown
    • Computer knows what number to expect when
  • User enters number and fixed password
Biometrics

• Automated measurement of biological, behavioral features that identify a person
  • Fingerprints: optical or electrical techniques
  • Voices: speaker verification or recognition
  • Eyes: patterns in irises unique
  • Faces: image, or specific characteristics like distance from nose to chin
  • Keystroke dynamics: believed to be unique
Location

• If you know where user is, validate identity by seeing if person is where the user is
  • Requires a device saying where the user is, like a smart phone
Multi-Factor Authentication

• Example: “where you are” also requires entity to have LSS and GPS, so also “what you have”

• Can assign different methods to different tasks
  • As users perform more and more sensitive tasks, must authenticate in more and more ways (presumably, more stringently) File describes authentication required
    • Also includes controls on access (time of day, etc.), resources, and requests to change passwords
  • Pluggable Authentication Modules
Multi-Factor Authentication

• Example: “where you are” also requires entity to have LSS and GPS, so also “what you have”
• Can assign different methods to different tasks
  • As users perform more and more sensitive tasks, must authenticate in more and more ways (presumably, more stringently) File describes authentication required
    • Also includes controls on access (time of day, etc.), resources, and requests to change passwords
  • Pluggable Authentication Modules
Identity

• *Principal*: a unique entity
• *Identity*: specifies a principal
• *Authentication*: binding of a principal to a representation of identity internal to the system
  • All access, resource allocation decisions assume binding is correct
Files and Objects

• Identity depends on system containing object
• Different names for one object
  • Human use, *eg.* file name
  • Process use, *eg.* file descriptor or handle
  • Kernel use, *eg.* file allocation table entry, inode
More Names

• Different names for one context
  • Human: aliases, relative vs. absolute path names
  • Kernel: deleting a file identified by name can mean two things:
    • Delete the object that the name identifies
    • Delete the name given, and do not delete actual object until all names have been deleted

• Semantics of names may differ
Example: Names and Descriptors

- Interpretation of UNIX file name
  - Kernel maps name into an inode using iterative procedure
  - Same name can refer to different objects at different times without being deallocated
    - Causes race conditions
- Interpretation of UNIX file descriptor
  - Refers to a specific inode
  - Refers to same inode from creation to deallocation
Example: Different Systems

• Object name must encode location or pointer to location
  • SSH style: host:object
  • URLs: protocol://host/object

• Need not name actual object
  • SSH style may name pointer (link) to actual object
  • URL may forward to another host
Users

• Exact representation tied to system

• Example: UNIX/Linux systems
  • Login name: used to log in to system
    • Logging usually uses this name
  • User identification number (UID): unique integer assigned to user
    • Kernel uses UID to identify users
    • One UID per login name, but multiple login names may have a common UID
Multiple Identities

• UNIX systems again
  • Real UID: user identity at login, but changeable
  • Effective UID: user identity used for access control
    • Setuid changes effective UID
  • Saved UID: UID before last change of UID
    • Used to implement least privilege
    • Work with privileges, drop them, reclaim them later
  • Audit/Login UID: user identity used to track original UID
    • Cannot be altered; used to tie actions to login identity
Groups

• Used to share access privileges

• First model: alias for set of principals
  • Processes assigned to groups
  • Processes stay in those groups for their lifetime

• Second model: principals can change groups
  • Rights due to old group discarded; rights due to new group added
Roles

• Group with membership tied to function
  • Rights given are consistent with rights needed to perform function
• Uses second model of groups
• Example: DG/UX
  • User *root* does not have administration functionality
  • System administrator privileges are in *sysadmin* role
  • Network administration privileges are in *netadmin* role
  • Users can assume either role as needed
Naming and Certificates

• Certificates issued to a principal
  • Principal uniquely identified to avoid confusion

• Problem: names may be ambiguous
  • Does the name “Matt Bishop” refer to:
    • The author of this book?
    • A programmer in Australia?
    • A stock car driver in Muncie, Indiana?
    • Someone else who was named “Matt Bishop”
Disambiguating Identity

• Include ancillary information in names
  • Enough to identify principal uniquely
  • X.509v4 Distinguished Names do this

• Example: X.509v4 Distinguished Names
  • /O=University of California/OU=Davis campus/OU=Department of Computer Science/CN=Matt Bishop/

  refers to the Matt Bishop (CN is common name) in the Department of Computer Science (OU is organizational unit) on the Davis Campus of the University of California (O is organization)
CAs and Policies

• Matt Bishop wants a certificate from Certs-from-Us
  • How does Certs-from-Us know this is “Matt Bishop”?
    • CA’s authentication policy says what type and strength of authentication is needed to identify Matt Bishop to satisfy the CA that this is, in fact, Matt Bishop
  • Will Certs-from-Us issue this “Matt Bishop” a certificate once he is suitably authenticated?
    • CA’s issuance policy says to which principals the CA will issue certificates
Example: Verisign CAs

• Class 1 CA issued certificates to individuals
  • Authenticated principal by email address
    • Idea: certificate used for sending, receiving email with various security services at that address

• Class 2 CA issued certificates to individuals
  • Authenticated by verifying user-supplied real name and address through an online database
    • Idea: certificate used for online purchasing
Example: Verisign CAs

• Class 3 CA issued certificates to individuals
  • Authenticated by background check from investigative service
    • Idea: higher level of assurance of identity than Class 1 and Class 2 CAs

• Fourth CA issued certificates to web servers
  • Same authentication policy as Class 3 CA
    • Idea: consumers using these sites had high degree of assurance the web site was not spoofed