SE-4920: Lecture 12 Authentication

- Reading
  - Chapters 9 and 10
- Today's Outcomes
  - Explain the difference between authorization and authentication
  - Critique authentication methods using password and/or address-based methods
  - Discuss eavesdropping and server database reading and how various authentication methods deal with them
  - Explain the general use of trusted intermediaries for both secret and private key based systems
  - Discuss issues specific to authenticating people, including the three main approaches to doing so

1

Authentication

- Reliably verifying the identity of someone or something
  - People recognize voice / appearance
  - ID badge
  - Knowledge of CVC2 (Card Verification Code)
- Computer to computer
  - Can use high-quality cryptographic systems
- User to computer
  - Limitations to human memory of secrets
- Contrast with authorization
  - Deals with allowing access by someone/something to data, service, etc.

2

Password-based authentication

- Major problem is eavesdropping
  - Transmit password over network to prove identity
  - Cell phone cloning (transmit number/password)
    - Has been addressed, but escalation continues...
- Why not use cryptography?
  - When person is involved, there is still an initial entry point
  - With computers...
    - Perhaps system built on previously human-computer method
    - Cryptography too expensive to implement or compute
    - Legal issues

3

Off- vs. on-line password guessing

- Online guessing
  - System enforces a particular interface for verifying password
  - *E.g.*, PIN number at ATM
  - Easy to limit number and/or speed of attempts
    - Linux uses an increasing delay
  - Can respond to repeated attempts
- Offline guessing
  - Capture quantity X derived from password (*e.g.*, hash)
    - Try various passwords until you get X
      - Easy to automate, no evidence on system that an attack is ongoing
  - Also called "dictionary attack" (dictionary words are common passwords)

4

Storing user passwords: concepts

- Where should authentication information be stored?
  - On every server?
  - On single server that provides the information to other servers when needed?
    - Authentication storage node
  - On a single server that processes requests from services on behalf of the user and responds "yes" or "no"?
    - Authentication facilitator node
- In 2 and 3, servers must authenticate the node
- Storing unencrypted passwords escalates the severity of a compromise in any case

5

Storing user passwords: solutions

- Hashing
  - Node compromise does not compromise password,
    - but enables offline attack
- Encrypt passwords?
  - Good for backups, but where is the key?
  - ...compromise of node may also compromise key
- *E.g.*, Sun's NIS (Network Information Services)
  - Uses an authentication storage node with hashed passwords
  - But, anything can claim to need to see the information
    - Can be fixed with configuration changes in newer NIS+
  - And, the node doesn't need to prove its identity

6

Address-based authentication

- Infer the identity using network location
  - Trust the network for source information
- UNIX rtools implement this
  - "Equivalent machines" – access if username matches (**`/etc/hosts.equiv`**)
  - Mappings from remote machine, account name pairs to local accounts (per-user **`.rhosts`**)
- Safe from eavesdropping
  - Passwords not sent

7

Problems with address-based authentication

- Compromise of one system not isolated
- Impersonation/spoofing of network addresses
  - Generally easy to lie about the source address
    - May be caught by firewalls or routers, depending upon scope and configuration
  - Harder to get traffic back than to send it
    - MAC flooding
      - Switch fail open, limited scope
    - ARP spoofing
      - Change MAC address for a given IP, limited scope
    - Source routing
      - Specify IP route, put intruder in middle
      - Firewalls, etc., may block

8

Cryptographic authentication protocols

- Improvement on password-based and address-based authentication
- Prove identity by performing cryptographic operation on supplied data
  - *E.g.*, based on shared secret
- We wish to authenticate both systems and people

9

Passwords as cryptographic keys

- A function of the password (*e.g.*, hash) to get a secret key (*e.g.*, for DES)
- What about specially-constructed keys (RSA, PGP, ...)?
  - Encrypt the private key with a secret-key algorithm based on the password
    - Strength reduced to password strength if not careful (§12.4 has a solution)
  - Seed a psueorandom number generator based on the password and use it to generate the key
    - Not used much
    - A lot of computation
    - Public key also derived from generator, allows offline guessing

10

Eavesdropping and server database reading

- Public key algorithms allow security from...
  - Eavesdropping
    - Private key unknown
  - And compromise of the server database
    - Private key not stored there
- Authentication is done by signing with private key

11

Eavesdropping and server database reading

- Hard to protect against both without PK
- Transmit clear passwords but compare to hash
  - Vulnerable to eavesdropping
- Challenge to encrypt a random number
  - Vulnerable to server database reading
- Can do both with Lamport's hash (§12.2), with some limitations

12

Trusted intermediaries: secret key

- Infeasible for all pairs to share keys
- Key Distribution Center (KDC)
  - Has a shared secret with each node
  - A requests key for B from KDC
  - KDC makes random K_{AB}
    - Encrypts with K_A and K_B
    - Returns these to A (the 2nd one is a "ticket")
- Problems
  - Compromise of KDC allows complete impersonation
  - Single point of failure / bottleneck
- Can chain together (*e.g.*, one for each company, protocol on page 231)

13

Trusted intermediaries: public key

- Problem: too many public keys
  - Solution: trust a small number of signers to vouch for validity of public keys
    - Only need to know this public key *a priori*
- Certification Authority (CA)
  - Generates certificates
    - Signed message specifying a valid user / public key association
- Can chain together (CAs sign keys for other CAs)
- Advantages
  - Can be isolated from network: secure, offline signing
  - Failure does not bring down system, just prevents adding / invalidating certificates
  - Certificates cannot be forged without the key, and require no special protection
  - Compromised KDC cannot decrypt conversations

14

Certificate revocation

- Charge card good for 2 years, and then you cancel it…
- Similar solution with CAs
  - Revocation list
    - All unexpired certificates that are no longer valid

15

Authentication of people

- Humans cannot feasibly
  - Remember very high-quality secrets
  - Perform cryptographic calculations
- Solution: combine multiple techniques
  - What you know (*e.g.*, password)
  - What you have (*e.g.*, key, ID card)
  - What you are (biometrics; *e.g.*, fingerprint, voice, iris, retina)

16

Off-line password guessing: dictionary attack

- Consider a large number of users and their captured, hashed passwords
- Encrypt every word in dictionary and other likely passwords; search for matches
- Overhead is typically in the hashing of each word
  - Not searching through the long list of users

17

Foiling the dictionary attack: salting the hash

- Put the complexity back in favor of the good guys
- Concatenate passwords with a random “salt” value before hashing
  - Salt stored in the clear in the password system
- For N accounts
  - the work of off-line guessing was increased by about N
  - but the increase is negligible for legitimate use

18

Eavesdropping

- Issues with wiretapping, key loggers, etc.
- Use of one-time passwords
  - A printed list of passwords used in order
  - Generate new list when they run out
- Virtually eliminates standard eavesdropping attacks
  - More tedious

19

Passwords and careless users

- Users tend to circumvent hard to follow policies
  - Required resets tend to encourage simple passwords
  - Requiring fresh passwords leads to adding digits, etc.
    - Plus old passwords/hashes need to be stored
- The ultimate circumvention is writing the password down
  - Perhaps in a fairly public place

20

Authentication tokens

- Something you have, like a door key
- Drawbacks
  - Generally require custom hardware
  - Can be stolen
- Solution: smart card
  - PIN protected information storage
  - Cryptographic challenge/response
    - Generates response only after entering PIN
  - Cryptographic calculator / readerless smart card
    - Advantage: no reader, common for remote employee access
    - System presents numeric challenge
    - User enters PIN and numeric challenge
    - Card displays response
    - User enters response to system

21

Biometrics

- Retinal scanner – laser, psychologically threatening
- Fingerprint – long history, catching on?
- Face recognition
- Iris scanner – less intimidating than retinal scan
- Handprint reader
- Voiceprint – area of research: how to work around tape recording vulnerability
- Keystroke timing
- Signature – UPS delivery

22