

CS 4803

Computer and Network Security

Alexandra (Sasha) Boldyreva
Authentication

1

Authentication

- Verifying the identity of another entity
 - Computer authenticating to another computer
 - Person authenticating to a local computer
 - Person authenticating to a remote computer
- Two issues:
 - How authentication information is stored (at both ends)
 - Authentication protocol itself

2

Overview

- Authentication may be based on
 - What you know
 - What you have
 - What you are
 - Examples?
- Mutual authentication vs. unidirectional authentication

3

Attack taxonomy

- Passive attacks
- Active attacks
 - Impersonation
 - Man-in-the-middle
- Server compromise
- Different attacks may be easier/more difficult in different settings

4

Address-based authentication

- Is sometimes used (e.g., unix)
- This is generally not very secure
 - Relatively easy to forge source addresses of network packets

5

Password-based protocols

- Password-based authentication
 - Any system based on *low-entropy* shared secret (note: different from book definitions!)

6

Password selection

- User selection of passwords is typically very weak
 - Lower entropy password makes dictionary attacks easier
- Typical passwords:
 - Derived from account names or usernames
 - Dictionary words, reversed dictionary words, or small modifications of dictionary words
 - Etc.

7

Better password selection

- Non-alphanumeric characters
- Longer phrases
- Can try to enforce good password selection...
- ...but these types of passwords are difficult for people to memorize and type!

8

From passwords to keys?

- Can potentially use passwords to derive symmetric or public keys
- What is the entropy of the resulting key?
- Often allows off-line dictionary attacks on the password

9

Password-based protocols

- Any password-based protocol is vulnerable to an “on-line” dictionary attack
 - On-line attacks can be detected and limited
 - How?
- Any password-based protocol is vulnerable to off-line attack if server is compromised

10

Password-based protocols

- Best: Use a password-based protocol which is secure against off-line attacks when server is *not* compromised
 - Unfortunately, this has not been the case in practice (e.g., telnet, cell phones, etc.)
- This is a difficult problem!

11

Password storage

- In the clear..
- Hash of password (done correctly)
 - Doesn't always achieve anything!
 - Makes adversary's job harder
 - Potentially protects users who choose good passwords
- “Salt”-ed hash of password
 - Makes bulk dictionary attacks harder, but no harder to attack a particular password
- Centralized server stores password
- Threshold password storage

12

Centralized password storage

- Authentication storage node
 - Central server stores password; servers request the password to authenticate user
- Auth. facilitator node
 - Central server stores password; servers send information from user to be authenticated by the central server
- Note that central server must be authenticated!


13

Basic authentication protocols...

- Server stores $H(\text{pw})$; user sends pw
 - "Secure" against server compromise, but not eavesdropping (or replay attacks)
- Server stores pw, sends R; user sends $H(\text{pw}, R)$
 - Secure against eavesdropping, but not server compromise (or dictionary attack)
 - What if the user sends R also...?
- Can we achieve security against both?

14

Other techniques for human auth.

- Tokens
 - Magnetic stripe cards
 - Smartcards
 - "Standalone" tokens:

- Still need a secure auth. protocol!

15

Biometrics

- Various possibilities...
- Drawbacks
 - Entropy?
 - Are biometric data secret?
 - Revocation?
- Difficult to use securely!
 - Non-uniform
 - Errors
 - Still need a secure protocol...

16

Public-key protocols

- Server stores pk ; user stores sk
- Server sends R ; user signs R
 - Using a secure signature scheme...
- Is this secure?
 - Potential weaknesses
 - What if we had used encryption instead?
- Can we achieve security against server compromise and eavesdropping without using public-key crypto?

17

Lamport's hashing protocol

- Server stores $H^n(pw)$; user sends $H^{n-1}(pw)$
 - Server updates user's entry...
- Can also add "salt" to hash
 - Can use same password on different sites
 - Protects against off-line attacks
- Can use same password (but different salt) when password "expires"

18

Some attacks...

- Secret expires...
- No mutual authentication
 - "Small n " attack

19

Session key establishment

- There are very few applications for which authentication alone is sufficient!
 - What do you do once you are authenticated?
- Generally, need to establish a *session key*
 - Efficiency advantages to using symmetric-key techniques if public-key auth. is used
 - Advantages even if a symmetric key is already shared
 - ...

20

Session keys

- Reduces effectiveness of cryptanalysis
- If a key is compromised, only one conversation is affected
- Prevents replay of messages from other conversations
- Better security from un-trusted host

21

KDCs

- Key Distribution Centers
- Advantages of symmetric-key crypto, without $O(n^2)$ keys
 - But requires a trusted intermediary
 - Single point of failure/attack

22

Multiple intermediaries

- Can use multiple KDCs...
 - Can have all pairs of KDCs share a key
 - More likely, there will be a hierarchy of KDCs

23

Basic key exchange

- Public-key based...
- Diffie-Hellman key exchange
 - Not authenticated (yet)!

24