Authentication and Password

CS166 Introduction to Computer Security
Confidentiality
• Prevent disclosure of information to unauthorized parties

Integrity
• Detect data tampering

Availability
• Guarantee access to data
Beyond CIA...

Who are you?

Prove it!

Here’s your stuff...

Identification

Authentication

Authorization
Identification

- Humans are generally indistinguishable in front of a computer
- A subject should provide an identity
- The system will verify if you have the proof to claim an identity
  - This process is called Authentication
Authentication

• Authentication is the act of confirming the truth of an attribute of a datum or entity

• There are three authentication factors:
  – Knowledge: Something you know
  – Ownership: Something you have
  – Inherence: Something you are
## Knowledge

<table>
<thead>
<tr>
<th>Strengths</th>
<th>Weaknesses</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Easy to transport</td>
<td>• Can be forgotten</td>
</tr>
<tr>
<td>• Can be changed</td>
<td>• Easy to duplicate</td>
</tr>
<tr>
<td>• Easily transferrable/easy to duplicate</td>
<td>• Verifier often learns the secret</td>
</tr>
</tbody>
</table>
Ownership

- Something the user has (e.g., phone number, ID card, security token, etc.)

**Strengths**
- Easily transferable
- More difficult to clone
- That what you know

**Weaknesses**
- Easily transferable
- Can be lost or stolen
- Can be forged
  - e.g. a key can be made from photos
Inherence

• Something the user is or does (e.g., fingerprint, retinal pattern, DNA sequence, voice, etc.).

Strengths
• Non-transferable
• Usually identifies individual

Weaknesses
• Forgeable (ie, fingerprint from picture or from a glass)
• Can be lost (ie, loss or degradation)
• Can’t be changed
Authentication Devices

Time-Varying Codes
(One Time Password)
- Physical/Tamper proof
- Precise clock
- Hash chain inside

Biometric
- physiological or behavioral characteristic
- Irises, fingerprints, etc.
More authentication factors?

• Location factor
  – Where you are (ie. Gps, Mobile Cell, etc.)

• Ability factor
  – What you can do (ie. Keystroke Dynamics, mouse tracking, etc.)

• …

• Usually they are classified in the inherence factor, It is an open problem
  – NIST SP 800-63-1
If you need more security?

• Could you use more authentication factors to verify the identity of a user?

  — **Multi Factor Authentication** is born

  • To increase the level of security, many systems will require a user to provide different types of authentication factors

  **2-factor authentication**
  • ATM card + PIN
  • Credit card + signature
  • Passport + fingerprint
  • ...

  **3-Factor authentication**
  • ATM Card + Pin + Fingerprint
  • Passport + fingerprint + face
Multi-step Authentication

- User submits two or more authentication tokens
- ATM bank card (Two Factor)
  - Physical card (something you have)
  - PIN (something you know)
- Password + code sent to the phone (Two Step) Brown Authentication
  - Enter password (something you know)
  - Enter code (something you know)
Authorization

• Once a subject is Authenticated, access should be authorized

• **Authorization** is the function of specifying access rights to resources (**access control**)

• More formally, "to authorize" is to define access policy: permissions, rights, etc.
AAA and more...

Identification, **Authentication, Authorization, Accounting**, Auditing

– AAA Working Group, IETF

Identification

Authentication

Authorization

Access Control

Security Reference

Monitor

Access

Accounting

Auditing

Credentials, UserID, etc.

Factors, Proofs, etc.

Rights, Permissions, Privileges, ACL, etc.
Summary

• Often broken into three steps
  – Identification, Authentication, and Authorization
• Three ways to prove authentication
  – Something you have/are/know
• Multifactor authentication is generally more secure
Model of Web Access Control

• Authentication
  – Password, multi factors, etc.

• Session Management
  – Keep track of authenticated users

• Authorization
  – Given authentication, check permissions
In BROWSER we trust...

• Most of our trust on web security relies on information stored in the Browser:
  – A Browser should be updated since Bugs in the browser implementation can lead to various attacks
  – Add-ons too are dangerous
    – Hacking Team flash exploits - goo.gl/syVwiD
  – Executing a browser with low privileges helps
HTTP Headers

Every http packet comes with a header
Key/Value pairs storing metadata about the request/response
A user does not want to authenticate for every single request
Http is stateless

• Ideas?
  – Log in once
  – Generate a Session token/id
    • a temporary token used to both identify and authenticate user
How to implement a session token?

• Usually a random long string (not always)
• A shared secret between client and server
• Stored in a Cookie with an expiration time and a configured host
  – Automatically sent with every request
• DEMOS (No Cookies - No Sessions)
1. Removing cookies erases authentication
   – Server makes us log in again
2. ‘Remember me’ checkbox on the login
   – Cookie does not expire in the browser but also on the server
3. Logout and session cookie removed on client and server
4. If we disable cookies, can not sign in to most websites
Client-Side Controls

• Web security problems arises because clients can submit arbitrary input
  – OWASP TOP 10 comparison (goo.gl/oxyqDe)

• What about using client side controls to check the input?

• Which kind of controls?
Web client tool

• Web inspection tool:
  – Firefox Firebug or Chrome web developer:
    • powerful tools that allow you to edit HTML, CSS and view the coding behind any website: CSS, HTML, DOM and JavaScript

• Web Proxy:
  – Burp, OWASP ZAP, Tamper Data etc.
    • Allow to modify GET or POST requests
HTTP Proxy

• An intercepting Proxy:
  – inspect and modify traffic between your browser and the target application
  – Burp Intruder, OWASP ZAP, Tamper Data etc.
1. Session cookie theft (chrome -> firefox – tamperdata)
2. Session id randomness measure (burp)
   – goo.gl/7xtppW (randomness description)
Same Origin Policy (SOP)

• The SOP restricts how a document or script loaded from one origin can interact with a resource from another origin

• One origin is permitted to send information to another origin, but one origin is not permitted to receive information from another origin

• Without "allow sending”, there would be no "web" at all because each origin would be allowed to link only to itself
Browser add-ons: Request Policy (RP)

• RP gives you a default deny policy for cross-site requests.

• Cross-site requests are:
  – requests that your browser is told to make by a website you are visiting to a completely different website.

• RP allows you to whitelist cross-site requests you trust.
Is this site Secure?

• Web Technology introduced a new range of vulnerabilities not present in earlier Client Applications

• Unfortunately it is not enough to say:
  – We use **Https** or **SSL -128 bit** so this site is absolutely secure

• Main problem, **users** can:
  – Submit **arbitrary inputs**
  – **Interfere** with any piece of data transmitted
  – **Forge** requests or parameters with tools external to the browser

• In the modern web application frameworks (e.g. Spring, Struts, Hibernate, etc.) or CMS (e.g. wordpress, joomla, etc.) are commonly used and they can have bugs.
Summary

• Maintaining a session is a problem on the web
• Pay attention to session cookies
  - Expiration
  - Randomness
  - Tracking
• Log out is more secure than closing the page
• Input validation server side