Security for Web-based Applications

Jeff Offutt

http://www.cs.gmu.edu/~offutt/

SWE 642
Software Engineering for the World Wide Web
Help from Drs. Nick Duan and Ye Wu

Security Through Time

• 100 BC Rome : magic charms
• 1400s England : not much worth stealing, armed guards for the rich
• 1600s America : no doors
• 1800s USA : doors
• 1900s USA : better lock than your neighbor
• 21st Century : keys, PINs, passwords, biometrics, …
Passwords for Web Sites

• I have over 135 passwords
  – 5 bank & financial
  – 4 credit cards
  – 6 or 7 PINS
  – 4 computers
  – 9 accounts at GMU
  – 7 email
  – 14+ Other
  – 28+ Commercial
  – 5 Conferences
  – 3+ Home & Utilities
  – 27 Research & Professional
  – 11 My website
  – …

• Most of you probably have fewer … but still a lot
• How can we …
  – Keep all these passwords secure … AND
  – Remember all of them?

That is … balance security and usability

Usability and Passwords

• When users are forced to change their passwords frequently, they must come up with schemes to remember
• If change is too frequent, users’ schemes subvert security, making it easier to crack their passwords
• The dividing line is about six months
  – When users have to change their passwords twice a year or more, security goes down
• Designing memorable passwords is easy
• Designing secure passwords is easy

It is very hard to design passwords that are both easy to remember and secure!
User Strategies

• Have one password for all sites
• Have a simple scheme
  – april09april
  – offutt1 offutt2, …
• Have 150 passwords and a good notebook
  Frankly, none of these are very clever …
  • One password invites theft
  • Simple schemes can be broken
  • Nobody can remember 150 passwords …
    and notebooks get lost

Use a multilayer approach …

Multilayer Approach to Passwords
(thanks to Ravi Sandhu)
Users vs. Software

- This is the user’s level
  - Important principle – poorly designed security can reduce usability, but better usability increases security!

- How about the software?
  - Important principle – security can mean more work for the programmer!

Security Requirements for Web Apps

1. **Authentication**
   - Verify the identity of the parties involved
2. **Authorization**
   - Limit access to resources to users
3. **Confidentiality**
   - Ensure that information is given only to authenticated parties
4. **Integrity**
   - Ensure that information is not changed or tampered with
Where to Apply?

• Security can be applied at three levels:
  1. Web server (Apache)
  2. Web container (Tomcat)
  3. Web application (your servlet)

• If implemented in a Web application, that is sometimes considered being through the container

Security Application Methods

1. Secure web applications using a Web server
   - HTTP authentication
   - Authorization of users/groups
   - Authorization of domains
   - Secure HTTP, an extension of HTTP
   - SSL capabilities

2. Secure web applications using a servlet container
   - HTTP authentication (basic, digest)
   - Form-based authentication
   - Authorization of users/groups
   - SSL capabilities

3. Securing web applications by programming
   - Authorization of users
   - User information kept on the server in a session
User-level Passwords with Apache (1. web server security)

- In the directory: create .htaccess:
  
  AuthUserFile /home/student/gburdell/lib/users — passwd file, readable
  AuthGroupFile /dev/null — for groups, forget it
  AuthName swe642 — name of directory, part of prompt
  AuthType Basic — the only one allowed
  <Limit GET> — most common access control
    require user gburdell
  </Limit>
- Create password:
  
  htpasswd -c /home/student/gburdell/lib/users gburdell
  Will prompt for the password
- Adding additional users:
  
  1. htpasswd /home/student/gburdell/lib/users george
  2. add to .htaccess: require user george

Securing Web Apps by a Program

- Use the password input field
  
  - `<input name="password" type="password" id="password">`
- Validate the username and password on the server
- Store whether the user has been authenticated in the session object
  
  - *Never* pass this information back to the client!
- Don’t forget to lock the back doors
  
  - Check authentication in every software component
  - If the user is not authenticated, go back to the login screen
  - This is one of the most common vulnerabilities in web applications
Secure Socket Layer (SSL) based Authentication (*https*)

- Invented by Netscape in the mid 90's
- Encrypt every HTTP message to and from the web server using standard PKI technology
- De-facto standard used for secure web-based transactions
- Default URL https://some.domain.com with default port number 443
- Don’t get confused with S-HTTP
  - S-HTTP encrypts only the http message body

Applicability of Client SSL Authentication

- Highest level of security
- Can integrate with smart-card and biometric technologies
- Now supported by most browsers
  - Plug-ins are sometimes required
- Additional server module required to validate clients, usually using a vendor-specific security server
- Very expensive because large PKI resources (hardware / software / personnel) needed to create, maintain, distribute user certificates
Security in Web Applications

- Web applications require proper security at various levels for different purposes
  - HTTP Authentication, Basic/Digest (lowest level)
  - Form-based authentication
  - Customized authentication
  - SSL server authentication
  - SSL client authentication (highest level)

- Other security concerns
  - Database security
  - Network security
  - Human factors
    - Users must remember passwords
    - Changing passwords more twice a year or more decreases security

- This is an intro—more sophisticated techniques available

Security Over the Years

- Web applications open up many avenues for security threats
- In the 1980s, security was all math …
- In the 1990s, security revolved around the database …
- In the 2000s, security moved to the network …
- Now most security vulnerabilities are due to software faults
- For more information, take SWE 681, Secure Software Design and Programming
Summary

In 2007, Symantec reported that most security vulnerabilities were now due to software faults.

In a house: your lock should be better than your neighbor's.
This principle does not work with Web apps.