Attacking Session Management

Professor Larry Heimann
Web Application Security
Information Systems
Lab 2 Review

• Fastest crackers
  • Carlos Diaz-Padron
  • Miko Bautista
  • Jonathan Lazarowicz
  • Ryan Flood
  • Jake Kushner

• Key points to remember
  • plenty of attack vectors for XSS attacks
  • some XSS is just annoying, others can be quite painful
  • users have a great deal of control over what gets submitted
  • good hackers often start with things you think are invulnerable
  • FIEO -- in lab there was filtering of input but no escaping of output
Session basics

- HTTP is inherently stateless
- Need for sessions
- How data is saved, accessed

Session ID is the key to matching session data with a particular user
Session IDs and entropy

• Many languages like PHP or frameworks like Rails automatically generate session IDs for you

• May choose to customize session IDs for a variety of reasons

• If an attacker gets access to session ID or can brute force a session ID, major trouble will follow

• Measure the security of session IDs with concept of entropy; essentially a measure of randomness
IN WASHINGTON, THE JUSTICE DEPARTMENT TODAY CHARACTERIZED THE FIRST, FOURTH, FIFTH AND SIXTH AMENDMENTS TO THE CONSTITUTION AS "TYPOS."

IN SPORTS, THE NCAA HAS DETERMINED THAT ITS ANNUAL MEN'S BASKETBALL TOURNAMENT IS DISRUPTIVE TO ACADEMIC SCHEDULES AND WILL BE ABOLISHED.

IN TECH NEWS, MICROSOFT CHAIRMAN BILL GATES HAS CHALLENGED LINUX CREATOR LINUS TORVALDS TO A WINNER-TAKE-ALL STEEL-CAE JUDO DEATHMATCH.

AND ON THE BUSINESS FRONT, ANALYSTS ARE PREDICTING A MAJOR SPIKE IN THE ECONOMY AS PAIGE FOX PURCHASES HER BACK-TO-SCHOOL PIMPLE CREAM.

WILL YOU STOP HACKING THE CNN TELEPROMPTER?!

WOLF BLITZER'S ABOUT TO SAY "FEAR ME" IN KLINGON.
Securing sessions

**Generate strong tokens**

- Any finite item can in principle be guessed given sufficient time and resources.

- Objective of generating robust tokens is to make a successful prediction attack extremely unlikely during the lifespan of any given token.

- Most web application platforms contain built-in session handling mechanisms that are tried-and-tested.

- If you create your own tokens, use strong sources for random numbers (e.g. java.security.SecureRandom, not java.util.Random), and use recognized crypto libraries such as SHA1.
Securing sessions

**Protect tokens throughout their lifecycle**

- Tokens should only be transmitted over HTTPS – set the “secure” flag if HTTP cookies are used.

- Tokens should not be transmitted in the URL – use a hidden field in a form that uses the POST method.

- Logout functionality should be implemented.

- Session expiry should be implemented on the server side.

- Concurrent logins should be prevented. After a successful login, existing sessions for the same user should be expired.

- Any administrative functionality that displays current session tokens should have effective access controls.
Securing sessions

**Protect tokens throughout their lifecycle**

- If HTTP cookies are used to transmit tokens, their domain scope should be set as restrictively as possible. Do NOT rely on path scope to defend against malicious code on the same domain (this cannot be achieved).

- The HttpOnly flag can be set to prevent client-side scripts from trivially accessing HTTP cookies (to prevent some XSS attack payloads).

- When a user transitions to a higher trust level, they should be issued with a fresh token, to stop session fixation attacks.

- Unrecognized tokens should be rejected, and a fresh valid token issued instead.

- Secondary authentication can be used to protect critical actions.

- Per-page tokens can be implemented to limit opportunities for session hijacking, even if other flaws exist.
Securing sessions

Protect tokens throughout their lifecycle

• If HTTP cookies are used to transmit tokens, their domain scope should be set as restrictively as possible. Do NOT rely on path scope to defend against malicious code on the same domain (this cannot be achieved).

• The HttpOnly flag can be set to prevent client-side scripts from trivially accessing HTTP cookies (to prevent some XSS attack payloads).

• When a user transitions to a higher trust level, they should be issued with a fresh token, to stop session fixation attacks.

• Unrecognized tokens should be rejected, and a fresh valid token issued instead.

• Secondary authentication can be used to protect critical actions.

• Per-page tokens can be implemented to limit opportunities for session hijacking, even if other flaws exist.
Next Class:
Class Demonstrations