Web Security

Presented by Pete Freitag
ActivSoftware, Inc.

Agenda

- Security Principles
- Types of Security Attacks
- Coding for Security
- Lunch
- Web Server Security
- Q & A

Who are the bad guys?

- Hackers... Black Hat / White Hat
- Script Kiddies
- Con Artists / Phishers
- Spammers

How do hackers hack?

- Hackers exploit software flaws
  - The flaw usually unchecked inputs
- Snooping and sniffing
- Spoofing
- Bruit Force

Your Responsibility

- The security of your web application is your responsibility
- ChoicePoint is being sued for:
  - "negligence in protecting the private data of consumers"
- If your supervisors don't give you time to ensure your application is secure - keep a paper record of it.

Be Proactive

- Keep up on security it always changes
  - mailing lists
  - web sites
  - blogs

Security Principles

- You are only as secure as your weakest point
- Security by Obscurity is not security at all
- It is difficult to cover all possible attacks

Common Types of attacks

- SQL Injection Attacks
- URL Hacking
- Session Hi-jacking
- Cross Site Scripting (CSS or XSS)
- Cross Site Request Forgery (CSRF)

URL Hacks

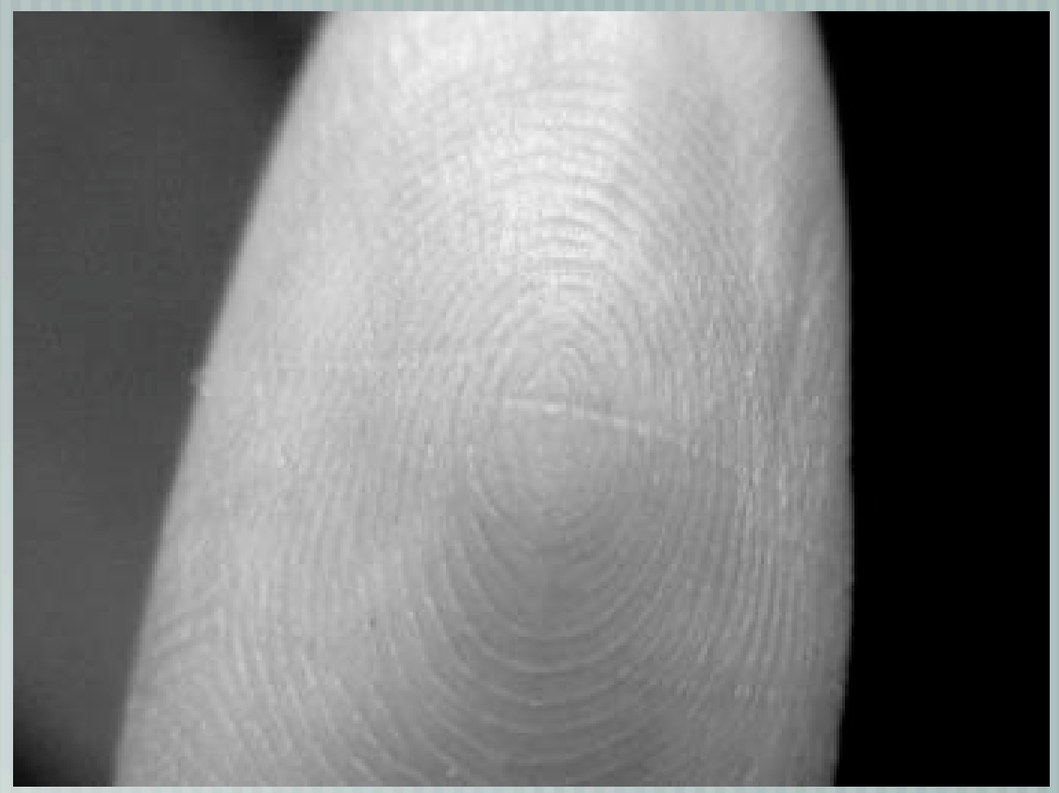
- /view.cfm?file=readme.txt
- <cfinclude template="#url.file#">

Ooops!

- You just gave the world read permission on your server
  - In this specific example cfm files will be executed, not displayed - which is also a problem
- /view.cfm?file=../../../etc/passwd

Rule of Thumb

- Don’t put file names in variables



URL Hacking (continued)

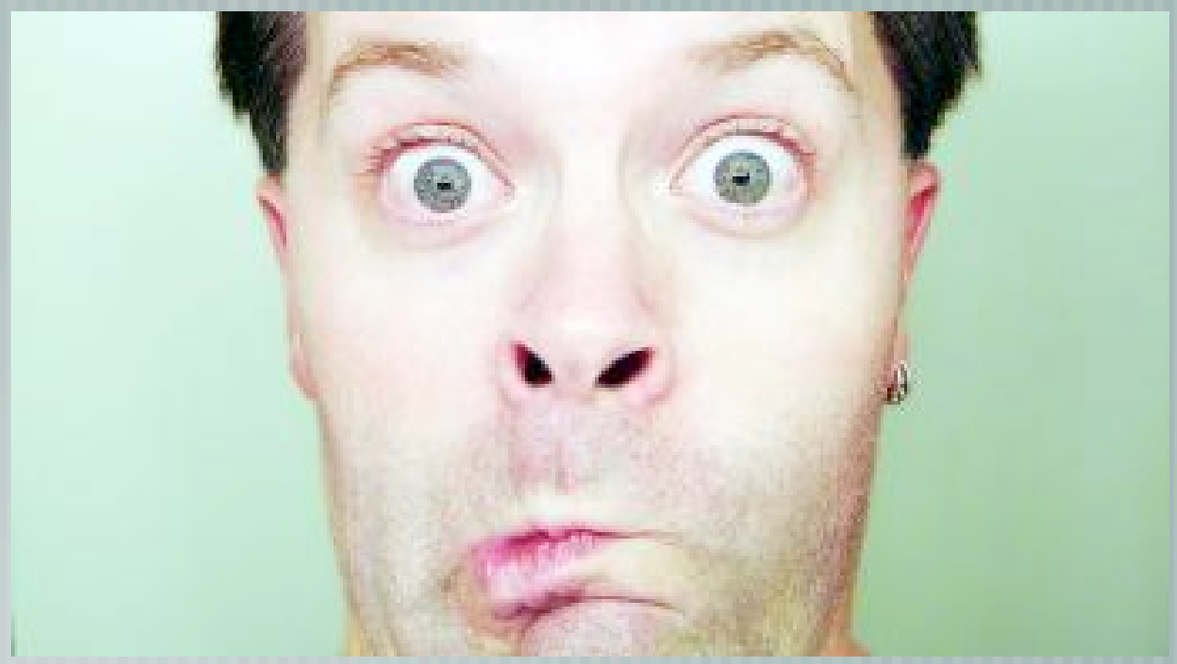
- /members/deleteSomething.cfm?id=24
- Never trust the input
  - Validate user permissions
  - DELETE FROM widgets WHERE id = ? AND user_id = ?

SQL Injection Hacking

- /page.cfm?id=123
- SELECT stuff FROM things WHERE id = #url.id#

Ooops!

- /page.cfm?id=123;DELETE FROM content
- SELECT stuff FROM things WHERE id=123;DELETE FROM content



Stopping SQL Injection

- Validate the input!
- CFQUERYPARAM / Prepared Statements is a good solution:
  - SELECT stuff FROM things
    WHERE id = <cfqueryparam cfsqltype="cf_sql_integer"
    value="#url.id#">

Stopping SQL Injection (CF)

- Type Checking - IsValid() / <CFPARAM>
- Length Checking - Make sure length is not exceeded Len()
- Ensure single quotes and semi-colons are escaped
- Use regular expressions, eg:
- <cfif ReFind("[^0-9]", url.id)> Not an Integer </cfif>

CFQUERYPARAM

- Prevents SQL Injection attacks
- Adds Performance similar to a stored procedure
- Can be used in all types of SQL statements, on all databases
- maxlength Attribute for length checking
- Can be used to insert NULL values into the DB null="true"
- Used with IN statement lists, with list="true"

CFPARAM

- Also useful for validating input types (especially in CFMX 7)
- Can be used to set defaults for inputs:
- <cfparam name="url.id" type="numeric" default="0">
- The numeric type accepts decimal numbers

Stopping SQL Injection (PHP)

- Same idea as with CFML - validate the input
- Prepared Statements
  - PHP 5's mysqli interface (mysql 4.x+)
  - PEAR:DB
- String Escaping Functions

mysqli Example

```
$db_connection = new mysqli("host", "user", "pass", "db");
$statement = $db_connection->prepare("SELECT x FROM y WHERE id = ?");
$statement->bind_param("i", $id);
$statement->execute();
```

Validating Integers in PHP

- is_int function
  - if (!is_int($id)) { ... }

Regex in PHP

- ereg() eregi() functions
  - Match any character that is not a letter
    - if (eregi("[^a-z]", $type)) { ... }

MySQL Escaping

- mysql_real_escape_string()
  - Escapes any character that is a special character in MySQL
  - PHP Version 4 and up

PostgreSQL Escape

- pg_escape_string()
  - PHP version 4.2 and up
  - PostgreSQL 7.2 or later

More SQL Escaping

- addslashes()
  - escapes ', ", \ and NULL
  - PHP Version 3 and up

Protect the Database

- Make sure your database user has minimal privileges
  - if you don't ever DROP or CREATE or execute stored procedures DISABLE them for your user

Cross Site Scripting XSS

XSS

- Attackers pass scripts in as variables
- /greetings.cfm?name=Pete
- <cfoutput>#url.say#</cfoutput>

Attackers will pass scripts

- /greeting.cfm?name=<script>...
- The script will be executed on your domain
- Inputs include URL, FORM, CGI, and Cookie variables

Why is it bad?

- It allows anyone to create a page on your site
- Can be used to trick your users into entering passwords, or simply to deceive them.
- Content exists even on a HTTPS connection
- Attackers can read cookies, and then hijack sessions

Preventing XSS

- Input validation
- Escape any HTML before outputting a variable.

Escaping HTML

- CF - Use HTMLEditFormat(url.name)
- PHP - Use htmlentities($name)

Check input for <

- <cfif Find("<", url.name)> <cfabort> </cfif>

ColdFusion 7 ScriptProtect

- <cfapplication scriptprotect="all" ...>
  - object, embed, script, applet, and meta tags replaced with InvalidTag
    - What about iframe?
    - What about javascript:

Cross Site Request Forgery

What is XSRF?

- Suppose your logged in to PayPal.com
- You get an email
- Tricked into clicking on
- https://paypal.com/transfer.cfm?to=hacker@xsrf.org&amount=500

How to prevent it

- Require HTTP Post variables
- Require passing a hash of the user's id
  - <input type="hidden" name="hash" value="#Hash (session.userid & salt)#" />

Session Hijacking

- Attacker spoofs login cookies
- In coldfusion it may be the cfid and cftoken cookies
- In PHP it may be the PHPSESSIONID cookie
- Or a home brew cookies

Session Hijacking

- Make sure attackers can't guess cookie values
- Encrypt your cookies
  - don't set a cookie with user id
- Use HTTPS
- Use UUID for CFTOKEN

Session Hijacking

- Don’t put session id’s in the URL
- Ask users to enter password before performing sensitive operations

Secure Cookies

- Can only be set over a HTTPS connection
- <cfcookie secure="true" ...>
- PHP set secure argument to TRUE
  - setcookie (name, value, expire, path, domain, secure);

HTTPS and Sniffing

- With HTTPS everything in the request and response are encrypted except for the request URL
  - So URL Variables are in plain text over HTTPS
  - Always use method="post" for sensitive data
- Lots of users use Public WiFi HotSpots

File Uploading

- Upload files into directories that don't have execute permissions.

File Uploading

- Require valid mime types
  - <cffile accept="image/jpg,image/jpeg,image/gif" ...>

File Uploads

- Finally double check the file extension
  - <cfif cffile.ClientFileExt IS NOT “pdf”> error </cfif>
  - MIME Type can be spoofed

Denial of Service Attacks

- Often called DOS attacks
- Malicious make repeated requests to your application
- Typically targets the page that requires the most processing
  - Search Forms

Preventing DOS Attacks

- No sure fire way to do so
- Limit the amount of processing a user do without authentication
- Implement Caching

Encryption

- If an attacker does compromise your system you need to ensure that sensitive data is encrypted

CFMX 7 Strong Encryption

- CFMX 7 Added support for AES, Blowfish, DES, and Triple DES
  - Pluggable security providers
- <cfset key = GenerateSecretKey("AES")>
- <cfset enc = Encrypt(str, key, algorithm, encoding)>
- <cfset dec = Decrypt(enc, key, algorithm, encoding)>

PHP Encryption

- mcrypt
- mcrypt_encrypt()
- DES, TripleDES, Blowfish (default), 3-WAY, SAFER-SK64, SAFER-SK128, TWOFISH, TEA, RC2 and GOST in CBC, OFB, CFB and ECB cipher modes. Additionally, it supports RC6 and IDEA which are considered "non-free"

Hashing

- A Hash function takes string and and encrypts it such that it cannot be decrypted.
- Often used for storing passwords
- Hash Function in ColdFusion
- Crypt Function in PHP

Password Hashing

- When the user sets their password, you run a Hash (password) and store the result in your DB.
- When they login, you Hash(form.password) and compare it to the Hash in your DB.
- If they match, the user entered the correct password.

Password Salt

- One technique to increase security is to create a random string (called a salt string) and append that to the password before hashing it.

Server Security

PHP Security Settings

- Register Globals should be turned off
- Don't display exception information
  - log_errors = On
  - display_errors = Off

PHP Security Settings

- memory_limit = 8M
  - compile with –enable-memory-limit
- post_max_size = 8M
- max_input_time = 60
- max_execution_time = 30

PHP Security Settings

- file_uploads = Off
- upload_max_filesize = 2M
- safe_mode = On

ColdFusion Security Settings

- Turn off Robust Exception Information
- Enable Use UUID for cftoken
- Create a global error handler
- Timeout long running requests (DOS)
- Make sure CF Administrator is hidden and pwd protected
  - only accessible over SSL

ColdFusion Security Settings

Setup Sandbox Security

- Disable unused tags
- Disable unused datasources
- Disable unused functions
- Disable external network access

Web Server Security

Web Server Security

- Step 1 - Install the latest security patches
- Step 2 - Disable all features your not using
  - Front Page Extensions
  - ASP / ASP.NET, Perl CGI, PHP?

Web Servers

- Run as a non privileged user
  - mitigates the possible damage done

Server Information

- Don't give away info / versions about your server
- Apache
  - ServerSignature Off
  - ServerTokens ProductOnly
- IIS
  - Get an ISAPI Filter, URLScan can do this
  - Y P [illegible] A S P . N E T

Put Apache in Jail

- You can create a jail on unix servers using chroot
- This limits the apache process to its own filesystem

mod_security

- An optional module for Apache that performs security filtering and auditing
- Create filtters, audits and rules based on any HTTP request variable
  - Form Post data, url, headers, cookies, etc.
- Detect / Block common attacks

IIS - Lockdown tool

- A wizard built by microsoft to help you secure your IIS web server.
  - Highly recommended
- URLScan - intrusion detection tool for IIS also by Microsoft

Scanning Tools

- Nessus
- Nikto

Security Best Practices

- Pay extra close attention to inputs
  - especially when the variable is used to manipulate files, databases, authentication, or the environment in general
- Ensure that error messages do not give away system details
  - file name, table names, internal variables, etc.

Security Best Practices

- Learn Regular Expressions
  - Very Useful for validating input in web apps
- Always use server side validation
  - Client side validation can be bypassed

Resources

- http://www.petefreitag.com/presentations/security/

?!

Thanks.