Set 12: Web Servers
(configuration and security)
(Chapter 21)
Key Questions

• What does a web server do?

• How can I control it?
  – URL re-writing / re-direction (and why do I care?)

• Access control and security
  – Developers
  – Users
Popular Web Servers

- One server to rule them all?

Web Server Basics

- http://www.example.com/products/widget.html
- What happens? Where does it come from?

- Are we sure?
Web Server Basics

- http://www.example.com/cgi-bin/search.pl?q=widgets
- What happens? Where does it come from?

- Are we sure?

- What’s not so good about this?
• Things to avoid

• Things to do

• How to do this?
How to control a web server?

• Apache – two primary config locations:
  – httpd.conf
    • Whole site
    • Must be root user
    • Requires restart
  – .htaccess
    • Per directory
    • Possibly each user (depends on config)
    • Re-read for each request
Content Control

1. Redirection

2. Rewriting

3. Content negotiation
Apache Modules

• The Apache web server uses modules to handle URL requests
  – “module” -> “mod”

• Examples:
  – mod_alias: redirect a request to another URL
  – mod_rewrite: rewrite behind the scenes
Redirection using mod_alias

# NOTE: this file (.htaccess) is in the 'change' directory

# Load from a different directory on the server.
Alias /image /ftp/pub/image

# Redirect file somewhere else (target MUST be absolute URL; root ok)
Redirect permanent /Users/nchamber/change/oldfile1.txt http://newplace734.com/test1.txt

# Redirect whole directory
Redirect permanent /Users/nchamber/change/olddir http://newplace734.com/newdir
Redirection OR Rewriting (using mod_rewrite)

# NOTE: continuation of .htaccess file, still in ‘change’ directory

# Using mod_rewrite - **first must turn on**
RewriteEngine On

# Sets the URL parent of the target (Default is /home/username/public_html)
RewriteBase /~username

# Rules use the directory paths, and redirect to same server
RewriteRule ^oldfile3.txt$ /change/test3.txt [R,L]
RewriteRule ^oldfile*.*txt$ change/catchOldFiles.txt [R,L]

# Behind the scenes change
RewriteRule ^oldfile5.txt$ /change/test5.txt [L]

# More complex
# redirect change/stuff/dogs to change/query.pl?q=dogs
# 302 = temp change
RewriteRule ^stuff/([^/]+)/?$ change/query.pl?q=$1 [R=302,L]
Exercise

• Create a rewrite rule:
  – People visit your site: **www.burritos.com/filling/beef**
  – Turn all possible fillings into search terms that are sent to your script: **www.burritos.com/search/fillings.pl?type=beef**
  – Make it silent so the user doesn’t see the new URL.
  – It should **not** redirect a longer URL from the user like: **www.burritos.com/filling/beef/salsa**
Apache Access Control – Options

1. Domain/IP restrictions

2. Password protection: “Basic”
   1. Much relegated to browser – can’t control
   2. Passed in plain text! (okay if using SSL)
   3. Password passed every time!
   4. Okay if using SSL

3. Password protection: “Digest”
   1. Sends “digest” rather than plain password
   2. But hacker could re-use digest!


5. Alternative?
1. Access control: IP-based

<LIMIT GET>
order deny,allow
deny from all
allow from .nadn.navy.mil
allow from .usna.navy.mil
allow from .usna.edu
allow from .naps.edu   # Naval Academy Prep School
allow from 192.190.228.  # test bench
allow from 192.190.229.  # test bench
allow from 192.31.8   # test bench
allow from 207.86.40.42  # NAPS
allow from 131.158.248.  # Navy Medical
allow from 131.158.247.  # Navy Medical
allow from 137.225.250.  # Joint Spectrum Command
allow from 12.110.116.250  # Alumni Association
allow from 128.56.
allow from 131.121.
allow from 131.122.
</LIMIT>
2. Access Control: “Basic”

- Whole directory
  AuthType Basic
  AuthUserFile /home/mXXX/public_html/.htpasswd
  AuthName "Members Only"
  require valid-user

- Per file
  <Files somefile.html>
  AuthType Basic
  AuthUserFile /home/mXXX/public_html/.htpasswd
  AuthName "Members Only"
  require valid-user
  </Files>
3. Access Control: “Digest”

- Whole directory
  AuthType Digest
  AuthName "myrealm"
  AuthUserFile .../.htpasswddigest
  Require valid-user

- Per file
  - Use <Files>

- Specific user (also applies to “Basic”)
  - Require user nchamber needham

- Groups of users
  - See documentation
Making the password file

htpasswd -c c:/wamp/.htpasswd username

htdigest -c c:/wamp/.htpasswd.digest realm username

Notes:
-c makes new file – omit to just add new entry (or update)
Substitute in actual path to the file
Don’t store password file in the web space!
Where to get more info

• Textbook (some in Chapter 21)
• Redirection/rewriting
  – Simple overview
    http://www.yourhtmlsource.com/sitemanagement/urlrewriting.html
  – Not-so-simple details
    http://httpd.apache.org/docs/1.3/mod/mod_rewrite.html#RewriteRule
• Access control
  http://httpd.apache.org/docs/1.3/howto/auth.html
Users and Passwords

• Don’t save passwords in plain text!
• Encryption: md5
  – Basic approach, ok for normal sites
  – *Not collision resistant
  – Online databases can lookup common passwords!

• Perl requirements:
  – Use Digest::MD5 qw(md5 md5_hex)
  – my $hashed = md5_hex($password)