

Faculty of Computer Science, Dalhousie University

16-Jan-2025

DGIN 5201 — Digital Transformation

Lecture 5: Lec 4: Password Protection

Location: LSC C236 Instructor: Vlado Keselj
Time: 13:05–14:25

Previous Lecture

- Notes: no copy-and-paste, touch typing
- SSH connection, elements of public-key cryptography
- Example 2: Applicant registration, printable form
- .htaccess file and directory index

Notices etc.

- A1: e1 and e2 postponed
- Some Emacs important Emacs commands:
 - C-x C-s — save
 - C-x C-c — quit
 - C-z — suspend to the command line
 - fg — go back from the command line
 - A useful tutorial: <https://www2.lib.uchicago.edu/keith/tcl-course/emacs-tutorial.html>

Hands-on e3: Password Protection

Example e3: Password Protection

- Let us make a copy of our e2 site
- First, go back to the directory above e2:


```
cd ../../
```
- Use command `pwd` to check your directory
- Copy e2 to e3 as an exact copy:


```
rsync -av e2/ e3/
```
- Check the new site e3 in the browser
- `rsync` is a very useful utility for copying directory structures
 - it works locally as well as over ssh
 - it copies incrementally differences, which is important if two sites are large and mostly equal
 - it may preserve permissions if we use option `-a`

Example 3: Simple Password Protection

- `cd` to e3 directory and let us prepare a password
- In a locally-only readable file `pw (rw-----)` we can save a password for our reference: `dt dt5201`

- Prepare the password for the site using the command:

```
htpasswd -bc .htpasswd dt dt5201
```

- Make the file `.htpasswd` all-readable and check its contents
- Prepare the file `.htaccess` and make it all readable:

```
AuthType Basic
AuthName dgin5201
AuthUserFile /users/webhome/<your.csid>/dgin5201/e3/.htpasswd
AuthGroupFile /dev/null
<Limit GET POST>
require user dt
</Limit>
```

- Check that site is password-protected

Summary of e3

- Files and permissions copied from e2
- pw file with permissions `rw-----`
- `.htpasswd` file with permissions `rw-r--r--` and appropriate content set up with the `htpasswd` command
- `.htpasswd` file with permissions `rw-r--r--` and content set up for password protection as given in class

Concepts Review: Example 3

- `rsync` command, `-av` options
- `htpasswd` command, password saved as hash
- Using `.htaccess` for password-controlled access

Unix-style Customization

Slide notes:

Unix-style Customization

- Unix-style customization is typically text-based
- Example: bash customization
 - aliases: `rm`, `mv`, `cp`, `em`
 - `.profile` and `.bashrc` files
- Example: Emacs customization
 - `.emacs` file
- Earlier example: Apache customization
 - `.htaccess`, `.htpasswd` files

The Unix-style operating systems, such as Linux, mostly use text-based configuration in various scenarios, and it is useful to be able to edit those plain-text files to customize the system and different software utilities, and for solving different technical problems.

Bash shell customization: The Bash shell customization is one such example. For example, the command `rm` is used to remove a file. Its default behaviour is usually such that after a command such as `rm file1` it will immediately remove the file named `file1`. In practice, this may easily lead to a mistake of accidentally removing a file that we did not want to be removed. This can be fixed by creating an alias for the command `rm` in Bash using the command:

```
alias rm='rm -i'
```

This command would create the alias `rm`, which when typed into the shell causes the actual command `'rm -i'` to be executed, which uses the option `-i` of the program `rm` to always ask user for a confirmation if they really want to remove the given file.

Hands-on e4: Introducing a Form

Example e4: Introducing a Form

- With `rsync` copy e3 to e4, update `.htaccess` file
- Change the table part of `index.html` to:

```
<form>
<table>
<tr><th align=right>First and last name:</th>
<td><input type="text"></td></tr>
<tr><th align=right>Email:</th>
<td><input type="text"></td></tr>
<tr><th>Area of Interest (DB, HI, DS):</th>
<td><select><option>DB</option><option>HI</option>
<option>DS</option></select></td></tr>
</table>
</form>
```

- Check the page and see that this is usable fillable form, which can be printed

Concepts Review: Example 4

- Creating fillable form in HTML: `<form>...</form>`
- `<input type="text">`
- `<select><option>opt1</option>...</select>`

Summary of e4

- Files set up as in e3
- `index.html` modified to make a usable fillable form