Securing a web site

- 3 viewpoints:
  - Server admin
  - Web site admin
  - Web site programmer/designer
- 3 levels
  - Server level
  - Whole web site level
  - Directory level

2 strategies

- Forbid access
  - Firewall, etc.
  - Login/passwd and ACL
- Encryption
  - Global
  - Local
- Both mixed ;-)

FORBIDDEN !

- Server admin side
  - Firewalling, etc.
- Web admin side (if no access to config)
  - Global server config file (apache.conf)
- Web programmer/designer side
  - Login/logout scripts
  - Session and cookies

OR

  - ACL using .htaccess/.htpasswd files
  - .htaccess protect 1directory
  - .htpasswd gives the ACL

.htaccess file ?

- Text file (no extension)
- Contains a set of directives
- One per protected folder (recursive)
- Should be readable for the server (windows compatibility issues;-)
- Directives allowed if global conf allows with :

```
AllowOverride All
```

.htaccess example

- **To be put in the private folder**

```
AuthUserFile /path/to/.htpasswd
AuthName "Enter your login/passwd"
AuthType Basic
require valid-user
```

- **Create the ACL file = .htpasswd**

```
htpasswd -c .htpasswd username1
htpasswd .htpasswd username2
```

http://www.kxs.net/support/htaccess_pw.html for Windows (not very effective – try htpasswd command – try IndexOptions +ShowForbidden to show the protected folder with Windows server)

- **Tell the server to accept local directives for this particular folder (conf file of the server)**

```
AllowOverride All
```

.htaccess pro/cons

PRO

- **Flexible (per directory basis)**
- **Simple**
- **Keep track of the session (passwd required only once in the same tree)**

- **CONS**
  - **ACL not flexible (no registering process)**
  - **Server efficiency+windows compat.**
  - **No log off process (close the browser)**

Conclusion : DO NOT USE IT;-) (if you can)

Global configuration

- ACL : .htpasswd
- same method to build up this file
- In the global config file
  - Create `directory` directive for target folder
  - Directives similar to the .htaccess local file
- PRO :
  - No local .htaccess
  - Server efficiency (read at start time)
  - The best option (apart from login/pass script)
- CONS
  - No registering process

Encryption howto

- Using SSL on top of HTTP = HTTPS
- Can be done on top of login/passwd
- Main process
  - Create private + public + certificate
  - Configure the main files
  - Configure ssl virtual host
  - Restart the server;-)

Ubuntu example

- **Enable ssl protocol (loading a module)**

  `a2enmod ssl (necessary)`

- **Create keys+certif**

  `apache2-ssl-certificate`

- **Server to listen 80 and 443 ports by adding**

  `Listen 443` **(in `ports.conf file`)**

- **Configure virtual host with root directory**
- **Deny access with HTTP (in HTTP Virtual host)**

  `SSLRequireSSL`

Ubuntu example

- **Enable ssl protocol (loading a module)**

  `a2enmod ssl`

- **Create keys+certif**

  `apache2-ssl-certificate`

- **Server to listen 80 and 443 ports by adding**

  `Listen 443` **(in `ports.conf` file)**

- **Configure virtual host with root directory**
- **Deny access with HTTP (in HTTP Virtual host)**

  `SSLRequireSSL`

Code example

- **In main virtual host**

```
 <Directory /path/to/protectedFolder/>
SSLRequireSSL
</Directory>
```

- **In SSL virtual host**

```
DocumentRoot /path/to/protectedFolder
SSLEngine on
SSLCertificateFile    /path/to/certifFile
SSLCertificateKeyFile /path/to/privateKeyFile
```

Conclusion

- SSL necessary for sensitive info
  - Credit cards
  - Addresses
  - HeartBleed April 2014:-)
- ACL
  - Necessary for private part (facebook, etc.)
  - Best option : login/passwd scripts
  - .htaccess : the worse option when nothing else possible

DO IT YOURSELF … NOW;-)