Activitat sm8act01: Breu estudi del protocol HTTP

A) Busca informació i respon les següents preguntes

1. Quin és el número i tipus de port utilitzat per defecte per un servidor web?. A on trobarem aquesta informació?.

2. Quin són els documents dins dels quals està definit el funcionament del protocol HTTP. Indica qui crea aquests document. el tipus i número dels documents.

3. Respon les següents preguntes:
   - **a)** Que significa que el protocol HTTP sigui un protocol client-servidor?
   - **b)** Què significa que HTTP és un protocol sense estat (stateless)?

4. Respon les següents preguntes:
   - **a)** Quines són les 4 parts d'un missatge HTTP?
   - **b)** Quina és la informació que pots trobar a la línia inicial d'un missatge HTTP?. Quins són els 2 tipus de línies inicials existents?. Quin tipus de programa cadascun dels tipus de línies incials?.
   - **c)** Com comença una lína de petició?. I una lína resposta?
   - **d)** Què és la capçalera HTTP?. En quines parts està dividida?
   - **e)** Quina informació dóna el camp de capçalera User-Agent?. Aquest camp el trobaràs dins d'un missatge de petició o de resposta?.
   - **f)** Quina informació dóna el camp de capçalera Host?. Aquest camp el trobaràs dins d'un missatge de petició o de resposta?.
   - **g)** Quina informació dóna el camp de capçalera Connection?. Aquest camp el trobaràs dins d'un missatge de petició o de resposta?.
   - **h)** Quina informació dóna el camp de capçalera Server?. Aquest camp el trobaràs dins d'un missatge de petició o de resposta?.
   - **i)** Quina informació dóna el camp de capçalera Strict-Transport-Security?. Aquest camp el trobaràs dins d'un missatge de petició o de resposta?.
   - **j)** Quina informació dóna el camp de capçalera Date?. Aquest camp el trobaràs dins d'un missatge de petició o de resposta?.

5. Indica de quina manera el client faria una petició de la pàgina estàtica index3.html que és troba a la carpeta "exam" que penja de l'arrel de l'arbre de directoris del servidor web. S'ha d'indicar mètode, fitxer, protocol i versió del protocol.

6. Indica l'esquema URI per accedir a un fitxer que s'anomena ex1m08uf1.pdf que es troba a la carpeta /home/daw2/examen d'un servidor FTP al qual s'ha d'accedir amb el login daw i password m08uf1pr1. El nom del servidor és ftp.fjeclot.edu i escolta pel port 21.

7. Indica quines són les diferències existents entre utilitzar un mètode POST i un mètode PUT

8. Indica quines són les parts de les quals es composa una entitat i el propòsit de cadascuna d'elles.

9. Indica quins són els 5 grups de missatges que pot enviar un servidor al client i quins són els propòsits de cadascun d'ells.

10. Indica quins mètodes farian que el servidor respongués amb el missatge 201.

11. Respon amb verdader o fals les següents afirmacions:
    - **a)** El mètode POST és menys segur perquè envia dades dins de la URL. Raona la resposta.
    - **b)** El mètode GET reenvia la informació quan es recarrega una web i el navegador hauria d'avisar-nos d'aquest fet. Raona la resposta.
    - **c)** Les peticions de tipus POST no poden ser desades a les adreces d'interès (bookmark)
    - **d)** Una petició de tipus POST envia les dades dins del cos del missatge
    - **e)** Les peticions de tipus GET no admeten qualsevol tipus de dades (texte, imatge, video, etc..)

12. Quin és el significat de la següen resposta d'un servidor:

```
HTTP/1.1 302 Found
Location: http://www.example.com/test/index2.php
```

B) Peticions i respostes HTTP treballant amb Telnet i wireshark

1. Posa en marxa **Wireshark** i filtra el contingut de les captures de manera que només es mostrin els continguts dels missatges HTTP. Fes la següent petició HTTP utilitzant: ***telnet www.google.cat 80***

```
GET / HTTP/1.1
Host: www.google.cat
User-Agent: telnet
```

Mostra el resultat obtingut amb **Wireshark**, i troba:
a) La capçalera IP, l'adreça IP del teu ordinador i la del servidor a la capçalera IP del missatge enviat pel programa telnet.
b) El port (tipus i valor) utilitzat pel programa telnet i pel servidor web
c) Quin codi de resposta dóna la capçalera HTTP enviada pel servidor? Què significa?
d) Que indica la capçalera *Server* de la resposta donada pel servidor? Explica el resultat que has obtingut.
e) Que indica la capçalera *Content-Type* de la resposta donada pel servidor? Explica el resultat que has obtingut.
f) Comprova si la connexió es tanca immediatament després de rebre la resposta del servidor. Hi ha alguna manera de tancar la connexió?

2. Fes la següent petició HTTP utilitzant: ***telnet www.google.com 80***

```
GET / HTTP/1.1
Host: www.google.es
User-Agent: telnet
Connection: close
```

Mostra el resultat obtingut amb **Wireshark**, i respon a les següens preguntes:

a) Per què serveix la capçalera *Connection: close*? Què passa al executar la petició amb *Connection:close* a diferència de l'exercici anterior?
b) Fixa't que a la petició el *Host* ara és www.google.es. Això ha fet que la resposta sigui diferent? Per què? Mostra la resposta i comenta el resultat obtingut i les diferències amb l'exercici anterior.
c) Què significat pel navegador que el Expires valgui -1 i que el camp max-age de Cache-Control sigui igual a 0?

3. Posa en marxa **Wireshark** i amb l'eina **telnet** connectat al port **80/tcp** del servidor **xtec.gencat.cat** i realitza la següent petició:

```
HEAD /ca/curriculum/professionals/ HTTP/1.1
Host: xtec.gencat.cat
User-Agent: telnet
Connection: close
```

Amb wireshark comprova:

a) Quina reposta dóna el servidor. Indica el significat del codi de la reposta i del camp Location.
b) Indica quina és la nova adreça del servidor.
c) Indica el programa servidor de pàgines web utilitzat per xtec.gencat.cat.
d) Quina és la diferència entre haver utilitzat HEAD i no GET?
e) A quina hora s'ha generat el missatge HTTP en el servidor?

4. Analitza la següent "conversa" HTTP entre un client i un servidor i descriu allò que està passant:

```
GET /private/index2.html HTTP/1.1
Host: localhost
```

```
HTTP/1.1 401 Authorization Required
Server: HTTPd/1.0
Date: Sat, 27 Nov 2006 10:18:15 GMT
WWW-Authenticate: Basic realm="Secure Area"
Content-Type: text/html
Content-Length: 311
```

```
GET /private/index2.html HTTP/1.1
Host: localhost
Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
```

```
HTTP/1.1 200 OK
Server: HTTPd/1.0
Date: Sat, 27 Nov 2006 10:19:07 GMT
Content-Type: text/html
Content-Length: 10476
```

C) Peticions i respostes HTTP treballant amb Firefox i wireshark

1.- Treballant amb camps de capçalera - I

a) Descarrega la captura de wireshark **http://www.collados.org/daw2/sm8/httpwireshark** i fes un filtre per visualitzar només missatges HTTP i descartar missatge OSCP.
b) Obre el missatge amb la petició inicial realitzada pel navegador al servidor.
c) Quin és el propòsit del valor **q**?. Quin és l'efecte del valor en aquest cas concret?
d) Quin navegador i versió s'ha utilitzat?
e) A quin host virtual hem accedit?
f) De quin tipus era la petició?. Era de lectura o escriptura?
g) Obre el segon missatge de petició del navegador al servidor i comprova que demana?.

2.- Treballant amb camps de capçalera - II

a) Descarrega la captura de wireshark **http://www.collados.org/daw2/sm8/httpwireshark** i fes un filtre per visualitzar només missatges HTTP i descartar missatge OSCP.
b) Obre el missatge amb la resposta del servidor a la petició inicial realitzada pel navegador.
c) Quina és la data de creació del missatge HTTP?
d) El missatge ha trobat el recurs demanat?. Com ho saps?.
d) Quin tipus de document s'han enviat i quina és la seva codificació?. Com ho saps?
f) Què s'envia dins del camp de dades del missatge HTTP?
e) Temps que recordarà el navegador que el servidor només és accessible via HTTPS?. Com ho saps?
f) La connexió es tancarà després d'envia el missatge?. Com ho saps?.

3.- Comprovant el funcionament de les peticions POST i GET

a) Connectat a **http://www.collados.org/daw2/m08/uf1/php_html_post_get/form.html**. Comprova el valor del camp User-Agent del primer missatge HTTP enviat des del client.
b) Accedeix a la web Mètode Post. Omple el formulari i tramet la consulta. Comprova:
- Amb wireshark que s'ha creat una petició de tipus POST. Demostra-ho.
- Comprova que les dades s'envien dins del cos del missatge. Demostra-ho.
- Que les dades no són visibles a la barra d'adreces del navegador.
- Demostra que el resultat no es pot afegir a les adreces d'interès
- Comprova des de wireshark que quan recarreguem là pàgina es torna a enviar la petició i que el navegador dóna un missatge d'avís.

c) Accedeix a la web Mètode Get. Omple el formulari i tramet la consulta. Comprova:
- Amb wireshark que s'ha creat una petició de tipus GET. Demostra-ho.
- Comprova que les dades s'envien dins de la capçalera del missatge. Indica dins de quin camp es troben aquestes dades. Demostra-ho.
- Que les dades són visibles a la barra d'adreces del navegador.
- Demostra que el resultat es pot afegir a les adreces d'interès

4.- Múltiples connexions

a) Treballa amb Firefox. Connectat a **http://www.binefa.net/electronica/tutorial04/**. Comprova quantes peticions GET s'han generat per part del client i i quantes respostes respostes ha enviat el servidor.
b) Indica el motiu pel qual, a part de la petició inicial del client i la resposta inicial del servidor, s'han generat noves peticions i respostes.
c)Comprova que cada petició ha implicat la utilització d'un nou port per part del client. Per què?

D) Taula comparativa de mètodes

a) Què significa que un mètode sigui segur?. Quins mètodes són segurs?
b) Què significa que un mètode sigui "Cacheable"?. Quins mètodes són "Cacheables"?
c) Fes una taula comparativa dels mètodes GET, PUT, POST, HEAD i DELETE indicant per cada mètode si té les següents propietats:
- La petició envia dades al cos del missatge?
- La resposta a la petició té dades al cos del missatge?
- És un mètode segur?
- És un mètode idempotent?
- És un mètode cacheable?