Seguretat i Hacking: Monitorització de tràfic

Un cop activat el sniffer ethereal en una xarxa commutada visitem la plana web “www.google.es”, primerament capta els paquets de la resolució del domini:

```
No.     Time        Source                Destination           Protocol Info
      1 0.000000    192.168.49.5          192.168.49.4          DNS      Standard query AAAA www.google.es
      2 0.624625    192.168.49.4          192.168.49.5          DNS      Standard query response CNAME
www.google.com CNAME www.google.akadns.net
      3 0.624839    192.168.49.5          192.168.49.4          DNS      Standard query A www.google.es
      4 1.088554    192.168.49.4          192.168.49.5          DNS      Standard query response CNAME
www.google.com CNAME www.google.akadns.net A 216.239.59.147 A 216.239.59.99 A 216.239.59.104
```

El servidor DNS a la xarxa té la IP 192.168.49.4 i la NIC del nostre ordinador té la IP 192.168.49.5.

Al paquet número 1 es fa una consulta al registre AAAA del domini www.google.es, el qual s'utilitza si usem el protocol IPv6. El servidor no respon retornant una IP sinó un llistat d'alias “www.google.es” i “www.google.com”, interpreto que això significa que google no soporta encara IPv6. Tampoc arribo a entendre perquè el meu navegador Mozilla Firefox realitza aquest intent de resolució quan la xarxa local treballa també amb IPv4, es possible que estigui relacionat amb alguna opció de compilació i els desenvolupadors de la distribució GNU/Linux Ubuntu han decidit activar per facilitar el funcionament en xarxes IPv6.

Al paquet 3 es torna a fer una consulta al DNS, al registre A del domini “www.google.es”. Ara si, el servidor DNS respon donant diferents IPs pel domini. Molt probablement Google utilitza Round Robin DNS per repartir la carrega entre tots els seus servidors.

A continuació estableix la connexió amb el servidor (Three-way Handshake):

```
No.     Time        Source                Destination           Protocol Info
      5 1.091571    192.168.49.5          216.239.59.147        TCP      33223 > www [SYN] Seq=0 Ack=0 Win=5840
Len=0 MSS=1460
      6 1.091707    216.239.59.147        192.168.49.5          TCP      www > 33223 [SYN, ACK] Seq=0 Ack=1
Win=5840 Len=0 MSS=1460
      7 1.091831    192.168.49.5          216.239.59.147        TCP      33223 > www [ACK] Seq=1 Ack=1 Win=5840
Len=0
```

El nostre ordinador vol establir una connexió amb la IP 216.239.59.147 que correspon a google així que envia un paquet amb el flag SYN activat per sincronitzar els números de seqüència, ethereal mostra aquest número com 0 però es relatiu ja que el número inicial real es aleatori. També s'indica el tamany de la finestra utilitzat per la tècnica sliding window de TCP.

Google ens confirma l'establiment de la connexió amb el flag ACK i també activa el flag SYN per sincronitzar el seu número de seqüència. Y finalment confirmem amb un ACK.

Seguidament es realitza la transmissió de dades:

```
No.     Time        Source                Destination           Protocol Info
      8 1.091969    192.168.49.5          216.239.59.147        HTTP     GET / HTTP/1.1
      9 1.092154    216.239.59.147        192.168.49.5          TCP      www > 33223 [ACK] Seq=1 Ack=502 Win=6432
Len=0
     10 1.507879    216.239.59.147        192.168.49.5          HTTP     HTTP/1.0 200 OK (text/html)
     11 1.507957    192.168.49.5          216.239.59.147        TCP      33223 > www [ACK] Seq=502 Ack=1461
Win=8760 Len=0
     12 1.507993    216.239.59.147        192.168.49.5          HTTP     Continuation
     13 1.508012    192.168.49.5          216.239.59.147        TCP      33223 > www [ACK] Seq=502 Ack=1507
Win=8760 Len=0
     14 1.515075    216.239.59.147        192.168.49.5          HTTP     Continuation
     15 1.515162    192.168.49.5          216.239.59.147        TCP      33223 > www [ACK] Seq=502 Ack=1626
Win=8760 Len=0
     16 1.584284    192.168.49.5          216.239.59.147        HTTP     GET /images/hp0.gif HTTP/1.1
```

El paquet 8 es la petició del nostre navegador per obtindre la plana principal de google:

```
GET / HTTP/1.1
Host: www.google.es
User-Agent: Mozilla/5.0 (X11; U; Linux i686; rv:1.7.3) Gecko/20041005 Firefox/0.10.1 (Ubuntu)
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: es,ca;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Cookie: PREF=ID=3f80c54a4922cd26:LD=es:TM=1096056840:LM=1096119940:S=p-6Jy7q9lfBbOmc0
```

Y a continuació google ens retorna la plana principal amb la següent capçalera:

```
HTTP/1.0 200 OK
Cache-Control: private
Content-Type: text/html
Content-Encoding: gzip
Server: GWS/2.1
Content-Length: 1307
Date: Thu, 07 Oct 2004 21:13:54 GMT
Age: 0
X-Cache: MISS from saturno.marblestation.homeip.net
X-Cache-Lookup: MISS from saturno.marblestation.homeip.net:3128
Connection: keep-alive

..........V.r.6.}.W.L.....d+q,..&u..d.L..C........(...x./.'......o.....3.A....=..,s..g..8......+.0/.%..
V....e..iW.sp...qJx...G?..qp..3'.......Y..f.-......H..#..........$,.ry..`..2e...H.7....k....!~8.GA0......<.YnD...
=).wB+b.^.:...1`?..~.........\.&.*6..5#}...R.Jk...H..h...lV....F.5.$. .z.YL.0........H.. e..X.....-.
oV.61.:D..>.PSE.H3G..E...y8.yJ..T.,..d..OEB."v..MNI...}z.(X..&X...5.Cx.5..p.............u.M.\2kiv_.N.X)
4.v._........si J...*}.V.O...X...`T.q..;TN......rXO(.....u.h..V+.z..fx...r.I.)yS.s...'..Bo!T...[H..
rxH........W/t^h.jC.Kf......F[`.%...<H?3...
W.$=oN..W.*...hgc...CO..4.T...y....z......qo.H.+...;l^Mon...(2..%.V..a.K.;.d.....u....[+.F-.~._c."[L..hU.
(l..~...1.....[..(.......!.N.....B
~A.6K.....O..U~.w
```

Aquí es pot observar clarament que el gateway de la xarxa (192.168.49.4) té en marxa un proxy transparent ja que el navegador no esta configurat per passar per cap proxy. Totes les peticions que arriben al gateway des de la xarxa local i que vaguin dirigides al port 80, es redireccionen al port 3128 del propi gateway a on esta escoltant el proxy Squid. Segons les capçaleres, el proxy no tenia en cache la plana del google (ha fet un miss com indiquen els camps X-Cache) i s'encarrega de demanar la plana directament a google i donar-la al nostre navegador de forma totalment transparent.

A més veiem que el contingut esta comprimit amb gzip per obtindre un major aprofitament de l'ample de banda (s'indica al camp Content-Encoding), per aquest motiu no es veu en text clar el HTML de la plana web.

Al paquet 16 es veu com el navegador torna a fer una nova petició, aquest cop demana una imatge que ha extret del codi HTML després d'analitzar-ho. La resta de paquets que no he afegit es corresponent a la transmissió de totes les imatges que resten a la plana web de google.

Es pot observar que encara que HTTP es un protocol que per defecte no manté la connexió, si utilitzem el camp “Connection: keep-alive” podem mantenir la connexió oberta i d'aquesta forma demanar tots els elements que componen una plana web sense haver de obrir diverses connexions. D'aquesta forma s'optimitza la transmissió de dades ja que ens estalviem fer diversos three-way handshakes.

Autor: Sergio Blanco Cuaresma