The Pirate Bay un-SSL

Author

Stanislaw Pusep (creaktive@gmail.com)
Date: July 31, 2008
Homepage: http://sysd.org/stas/node/220

Theory

Recently, the world saw The Pirate Bay offering SSL encryption on their server. This means that your ISP won't know anymore which torrent you are downloading, right? Wrong. HTTPS is quite useless for protecting static **and** public content. By *static*, I do mean the `.torrent` file itself. It is always the same. By *public*, I do mean than one doesn't need any kind of authentication to pick up the content. It's always the same, for everyone. For crawlers, too.

So, one could easily index (a portion of) The Pirate Bay torrent database by the Content-Length. Then, one could intercept some encrypted traffic between some machine(s) within his/her network and the `torrents.thepiratebay.org` server. Knowing both (encrypted) request and response lengths, it is possible to get a quite reliable list of matches from the previously indexed torrent list.

Practice

1. Use Wireshark to capture some torrent downloads. Torrents are hosted on the separate server, which makes the task easier yet. Use the following capture filter: `"tcp and port 443 and host torrents.thepiratebay.org"`
2. Now, just go with the stream (*"Follow TCP Stream"* for the packet you suspect belongs to the torrent download. This will create another filter, like `"(ip.addr eq 192.168.0.10 and ip.addr eq 83.140.176.156) and (tcp.port eq 2157 and tcp.port eq 443)"`)
3. Save the displayed stream anywhere else (`pcap1.pcap` for example)
4. Use my TPB-TLSlen.pl Perl script to get the request/response lengths:

   ```
   perl TPB-TLSlen.pl pcap1.pcap
   ```

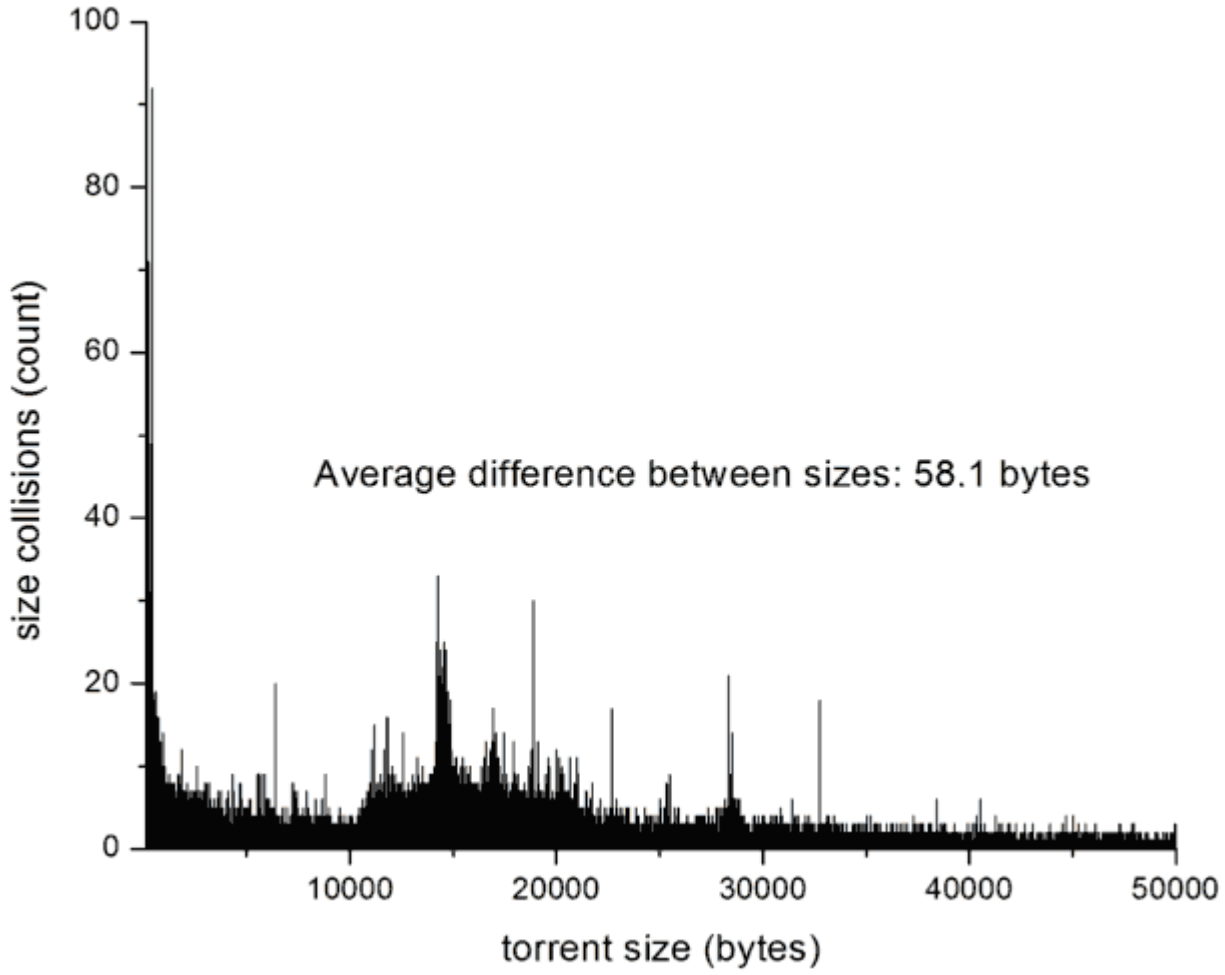
   This time, it only supports the TLS cipher. And it simply calls the `tshark` (the command line version of Wireshark) to parse output from.

5. Paste the **REQ** and **RES** values here (http://sysd.org/stas/node/220). (note that the **REQ** value is optional, setting it to 0 simply ignores the request size for matching)

Note that you are able to fine-tune the maximum and minimum header sizes. For the response, the headers are almost the same all the time. The only thing that varies is the decimal representation of the file length and age. (Un)fortunately, the request headers do vary for different browsers and referring pages. However, knowing the request size still helps a bit, especially if the torrent's filename was huge.

Precision

The following size distribution chart was generated using the database with ~80K torrents:



The most common torrent size is ~14 KB, and it's easy to figure out that such torrents represent the shared 700 MB files. There's also a major peak for the 454 bytes torrents. However, bigger torrents are less common; thus, the size detection technique becomes more precise. Now, the average "distance" between torrent sizes is ~60 bytes (at least for the sample I've collected). So, adding a cookie with the random size up to 128 bytes will disrupt the size matching detection a lot. The request size disruption is even easier: the largest torrent URI I've found was 150 bytes-wide. Thus, padding every request URI to match 150 characters is enough to make the requests completely indistinguishable. Joining the pieces (the padding add-on strings are **bold**):

```
GET /4319199/[a4e]Ghost_in_the_Shell_TV_01-
26.4319199.TPB.torrent?nVM2UGfcG533un4ym70eT29r0WwBLYdmFCNN+UTV/hiJ7EAXdFU5KfdWHpkB
5lXaCmITsACKOPVyjmpbaOB+CrI5 HTTP/1.1
Host: torrents.thepiratebay.org
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.1)
Gecko/2008070208
 Firefox/3.0.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: https://thepiratebay.org/recent
Cookie: language=pt_BR; country=BR; PHPSESSID=ad6cb7e414c8dc88e0c2444f6215165a

HTTP/1.1 200 OK
Content-Type: application/x-bittorrent
Etag: "2198642509"
Last-Modified: Mon, 28 Jul 2008 22:28:59 GMT
Server: lighttpd
Content-Length: 91601
Date: Mon, 28 Jul 2008 22:37:56 GMT
X-Varnish: 108010229 107999438
Age: 253
Via: 1.1 varnish
Connection: keep-alive
Set-Cookie:
p=68eOfxOC7JwBYcMe1RJWC4Z5PV/lJzqJORW8KROPMH9zQhszSjFnRp2tsNWEoyabWAloneUaoz
MxYtx4hoM9MZUKE/7wGzC3ZKLEZdppG4og3W; expires=Mon, 28-Jul-2008 22:37:56 GMT;
path=/; domain=torrents.thepiratebay.org

(binary torrent data)
```

Solution

1. Use a constant padding in the `.torrent` files. This messes things a bit, but stills ineffective. The only advantage is **not** reconfiguring the server software.
2. Patch the lighttpd server so it sends a non-lasting cookie with a random size.

Thanks

- MEGA Hospedagem, for the network resources provided for this little research.
- http://www.warchalking.com.br, for the inspiration.