Headshoot First to solve this challenge!

Web - 200 points

# Soal

|  |
| --- |
| <http://139.59.245.67:8000> |

# Solusi

Di halaman soal disebutkan bahwa flag ada di alamat <http://139.59.245.67:8000/flag.txt>, tapi tidak bisa pakai GET/POST. Maka kita coba request dengan metode OPTIONS untuk mendapatkan request yang diizinkan:

|  |
| --- |
| $ curl -v -X OPTIONS http://139.59.245.67:8000/flag.txt  \* Hostname was NOT found in DNS cache  \* Trying 139.59.245.67...  \* Connected to 139.59.245.67 (139.59.245.67) port 8000 (#0)  > OPTIONS /flag.txt HTTP/1.1  > User-Agent: curl/7.35.0  > Host: 139.59.245.67:8000  > Accept: \*/\*  >  \* HTTP 1.0, assume close after body  < HTTP/1.0 200 OK  < Content-Type: text/html; charset=utf-8  < Allow: **HEAD, OPTIONS**  < Content-Length: 0  < Server: Werkzeug/0.11.10 Python/2.7.6  < Date: Sun, 28 Aug 2016 01:59:39 GMT  <  \* Closing connection 0 |

Jadi bisa pakai HEAD (ada hubungannya sama judul soal?). Mencoba request HEAD:

|  |
| --- |
| $ curl -v -X HEAD http://139.59.245.67:8000/flag.txt  \* Hostname was NOT found in DNS cache  \* Trying 139.59.245.67...  \* Connected to 139.59.245.67 (139.59.245.67) port 8000 (#0)  > HEAD /flag.txt HTTP/1.1  > User-Agent: curl/7.35.0  > Host: 139.59.245.67:8000  > Accept: \*/\*  >  \* HTTP 1.0, assume close after body  < HTTP/1.0 200 OK  < Content-Type: text/html; charset=utf-8  < **Content-MD5: T/JeCTIbbkLdlRS/YqTvHw==**  < X-Compatible-With: Jigsaw  < **Content-Length: 48**  < Server: Werkzeug/0.11.10 Python/2.7.6  < Date: Sun, 28 Aug 2016 02:00:31 GMT  <  \* transfer closed with 48 bytes remaining to read  \* Closing connection 0  curl: (18) transfer closed with 48 bytes remaining to read |

Ada hal yang menarik dari hasil request tersebut:

* MD5 dari content (dalam bentuk base64-encoded, bukan hex-encoded)
* Panjang dari content
* Karena HEAD, content-nya sendiri tidak dikasih (inilah challenge-nya)

MD5 tersebut bisa dijadikan bentuk hex di Python:

|  |
| --- |
| >>> from base64 import b64decode  >>> print b64decode("T/JeCTIbbkLdlRS/YqTvHw==").encode("hex")  4ff25e09321b6e42dd9514bf62a4ef1f |

Hash tersebut dicari di tools database hash online ternyata tidak ditemukan :(

Kemudian terpikirkan ide, gimana kalau kita request satu byte pertama saja dari konten, dengan header “Range: bytes=0-1”?

|  |
| --- |
| $ curl -v -X HEAD --header "Range: bytes=0-1" http://139.59.245.67:8000/flag.txt  \* Hostname was NOT found in DNS cache  \* Trying 139.59.245.67...  \* Connected to 139.59.245.67 (139.59.245.67) port 8000 (#0)  > HEAD /flag.txt HTTP/1.1  > User-Agent: curl/7.35.0  > Host: 139.59.245.67:8000  > Accept: \*/\*  > Range: bytes=0-1  >  \* HTTP 1.0, assume close after body  < HTTP/1.0 200 OK  < Content-Type: text/html; charset=utf-8  < Content-Range: bytes=0-1  < **Content-MD5: j6FM3XVPkcxlVMnnGSnM5w==**  < X-Compatible-With: Jigsaw  < **Content-Length: 1**  < Server: Werkzeug/0.11.10 Python/2.7.6  < Date: Sun, 28 Aug 2016 02:04:52 GMT  <  \* transfer closed with 1 bytes remaining to read  \* Closing connection 0  curl: (18) transfer closed with 1 bytes remaining to read |

Didapatkan MD5 dari kontennya adalah j6FM3XVPkcxlVMnnGSnM5w== atau 8fa14cdd754f91cc6554c9e71929cce7 dalam hex. Dicari di database hash online ternyata itu adalah hash untuk huruf “f”. Voila, kita sudah dapat satu huruf pertama flag.

Maka kita bisa lanjutkan untuk karakter selanjutnya (Range: bytes=1-2), dapatkan MD5-nya, “decrypt” MD5 tersebut, dan seterusnya sampai (48-1) kali. Bisa manual, bisa otomatis. Saya buat script untuk mengotomatisasi langkah-langkah tersebut. Untuk tahu suatu MD5 adalah hash dari huruf apa, saya buat database sendiri (karena pasti 1 huruf, jadi bisa dibuat dengan mudah).

|  |
| --- |
| **import** **md5** **import** **httplib** **from** **base64** **import** b64decode  *# buat database MD5 dari setiap huruf yang mungkin* *# bentuknya: hashdict[suatunilaiMD5] = hurufnya*  hashdict = {}  **for** i **in** range(256):  m = md5.new()  c = chr(i)  m.update(c)  h = m.hexdigest()  hashdict[h] = c  *# mulai cari satu per satu isi flagnya*  k = ""  **for** i **in** range(47):  **print** i  conn = httplib.HTTPConnection("139.59.245.67", 8000)  conn.request("HEAD", "/flag.txt", "", {"Range" : "bytes=" + str(i) + "-" + str(i+1)})  res = conn.getresponse()    serverhash = b64decode(res.getheaders()[5][1]).encode('hex')   k += hashdict[serverhash]   **print** k |

Didapatkan flagnya:

**flag{HaveYouEverSeenMD5ContentHeaderBeforeThis?**