Like our articles? Signup for our Mailing List

HTTP(Hypertext Transfer Protocol) Request and Response

Submitted by Sarath Pillai on Thu, 02/21/2013 - 14:35

0 17
g+1 Share

In this article we will be discussing the foundational protocol of the internet. During 1990 Tim Berners-Lee, was the first one to propose a project called world wide web. The original purpose was to have a fast and reliable information exchange between scientists working in different parts of the world.
HTTP(Hyper Text Transfer Protocol), was implemented first by Tim Berners-Lee in

Web2PDF
converted by Web2PDFConvert.com

CERN.
HTTP protocol works in a client and server model like many other protocol. A web browser using which a request is initiated is called as a client and a web server software which respond's to that request is called as a server.
Below mentioned are some key points & terms to note about HTTP protocol, before we go ahead and understand a complete http request and response in practical.

- HTTP is an application layer protocol
- The default port if not mentioned in the request, is assumed as 80
- The hostname in the request is case insensitive
- World Wide Web Consortium and the Internet Engineering Task Force, both coordinates in the standardization of the HTTP protocol
- HTTP allows the improvement of its request and response with the help of intermediates in between(for example a gateway, a proxy, or a tunnel)
- The resources that can be requested by using HTTP protocol is made available with the help of a type of URI(Uniform Resource Identifier) called URL(Uniform Resource Locator).
- TCP (Transmission Control Protocol), is used to establish a connection to the application layer port 80 used by HTTP.(its not at all necessary to use port 80 for http connections, but if not explicitly mentioned in the URL, port 80 is assumed)
- A series of request and response in http is called as a session in HTTP
- HTTP version 0.9 was the first documented version of HTTP
- HTTP is a stateless protocol(which means each and every connection is independant of each other.)

For this tutorial i have setup two Machine's. One will act as a client, and the other will act as a server, containing apache version 2.2.3-22.
A default page which will show a **"hello"** message for this demosnstration is setup.
We will initiate connection from the client to the server, and will capture all the traffic on the server that's coming from the client and see what's happening.
Below shown is the request i initiated from the client (a centos machine), with the help of wget.

Web2PDF
converted by Web2PDFConvert.com

```
[root@slashroot1 ~]# wge
t http://192.168.0.103
--15:51:09--  http://192
.168.0.103/
Connecting to 192.168.0.
103:80... connected.
HTTP request sent, await
ing response... 200 OK
Length: 6 [text/html]
Saving to: `index.html.7
'
100%[====================
====================>] 6
           --.-K/s   in
0s
15:51:09 (283 KB/s) - `i
ndex.html.7' saved [6/6]
```

Let's see what happened during the http request initiated by the client on the server side by capturing all network traffic using tcpdump.

Web2PDF
converted by Web2PDFConvert.com

```
[root@slashroot2 ~]# tcpdump -i eth
0 -s0 -n -A host 192.168.0.104
tcpdump: verbose output suppressed,
 use -v or -vv for full protocol de
code
listening on eth0, link-type EN10MB
 (Ethernet), capture size 65535 byt
es
05:36:32.781066 IP 192.168.0.104.52
802 > 192.168.0.103.http: S 3151905
201:3151905201(0) win 5840 <mss 146
0,sackOK,timestamp 6458684 0,nop,ws
cale 4>
E..<..@.@......h...g.B.P..A........
.UR.........
.b.<........
05:36:32.781957 IP 192.168.0.103.ht
tp > 192.168.0.104.52802: S 3165795
006:3165795006(0) ack 3151905202 wi
n 5792 <mss 1460,sackOK,timestamp 6
465077 6458684,nop,wscale 4>
E..<..@.@......g...h.P.B..2...A....
..h.........
.b.5.b.<....
05:36:32.783047 IP 192.168.0.104.52
802 > 192.168.0.103.http: . ack 1 w
in 365 <nop,nop,timestamp 6458685 6
465077>
E..4..@.@......h...g.B.P..A...2....
m.d.....
.b.=.b.5
05:36:32.784509 IP 192.168.0.104.52
802 > 192.168.0.103.http: P 1:121(1
20) ack 1 win 365 <nop,nop,timestam
p 6458685 6465077>
E.....@.@......h...g.B.P..A...2....
m.......
.b.=.b.5GET / HTTP/1.0
User-Agent: Wget/1.10.2 (Red Hat mo
dified)
Accept: */*
Host: 192.168.0.103
Connection: Keep-Alive


05:36:32.784551 IP 192.168.0.103.ht
tp > 192.168.0.104.52802: . ack 121
 win 362 <nop,nop,timestamp 6465080
 6458685>
E..4..@.@.$....g...h.P.B..2...B*...
j.......
.b.8.b.=
05:36:32.786377 IP 192.168.0.103.ht
tp > 192.168.0.104.52802: P 1:268(2
67) ack 121 win 362 <nop,nop,timest
amp 6465082 6458685>
E..?..@.@.#....g...h.P.B..2...B*...
j.Q.....
.b.:.b.=HTTP/1.1 200 OK
Date: Thu, 21 Feb 2013 10:36:32 GMT
Server: Apache/2.2.3 (Red Hat)
Last-Modified: Thu, 21 Feb 2013 09:
47:39 GMT
ETag: "377709-6-f71984c0"
Accept-Ranges: bytes
Content-Length: 6
Connection: close
Content-Type: text/html; charset=UT
F-8

hello

05:36:32.787006 IP 192.168.0.103.ht
tp > 192.168.0.104.52802: F 268:268
(0) ack 121 win 362 <nop,nop,timest
amp 6465082 6458685>
E..4..@.@.$....g...h.P.B..3...B*...
j.......
.b.:.b.=
05:36:32.787622 IP 192.168.0.104.52
802 > 192.168.0.103.http: . ack 268
 win 432 <nop,nop,timestamp 6458689
 6465082>
```

Let's examine the tcpdump output by a close examination. I have devided the tcpdump output of our sample http request to different steps below, for easy

Web2PDF
converted by Web2PDFConvert.com

understanding.

Step 1: Establishing a TCP connection to the server by the client

Like any other application layer protocol, HTTP also works over TCP. So the first step done by the client in initiating a http connection is establishing a TCP connection with the http server.

You can clearly see that from the **first three lines of our tcpdump output above**. Those first three lines indicate's the initial TCP Handshake.

Read: TCP Three way handshake

If you see the tcpdump output, the first line is a connection initiate request with **"S"** flag set(so its a syn request). The second is a reply from the server to the client with "**S & ACK**" flag set. The third is an "**ACK**" reply from the client to complete the three way handshake.

Keep in mind that, we only completed the TCP connection establishment till here.

Step 2: Initiating a HTTP GET Request by the client to the server

The fourth line in the tcpdump output is from the client that initiates a http "GET" request.

Get requests are used by the clients for retrieving data from the server. If you closely see the GET request from the client, it reveals some information as below.

1. what is the client requesting. For example in our case, its a simple **"/"**. **GET "/"**, such type of request tell's the server to retrieve the root directory(the default page of the website.)
2. It also tells the http version the client is using. In our case its **HTTP/1.0**.
3. The client inform's the server what type of user agent is being used for the request(type of browser, its version, whether its an application etc. This part is send by the client software used to initiate the connection. In our case its "**Wget/1.10.2 (Red Hat modified)**")
4. Another important detail that the client

Web2PDF
converted by Web2PDFConvert.com

informs the server is the type of data that the client is ready to accept. HTTP allows all MIME media types. In our example our client is ready to accept any kind of data hence its */*

5. **HOST** field tells the hostname of the server, from which client is requesting the data.
6. Most of the clients request a keep-alive type of connection from the server. Keep-alive is used to keep the tcp connection made by the client, so that the overhead of creating a tcp connection is reduced for subsequent requests. Although the client requests a keep-alive kind of connection from the server, its the server who decides whether to keep tcp connection active or not(based upon the server configuration).

All the above mentioned GET request parameter's are shown below (which is also shown in the tcpdump output above.)

```
05:36:32.784509 IP 192.168.0.104.52
802 > 192.168.0.103.http: P 1:121(1
20) ack 1 win 365 <nop,nop,timestam
p 6458685 6465077>
E.....@.@......h...g.B.P..A...2....
m.......
.b.=.b.5GET / HTTP/1.0
User-Agent: Wget/1.10.2 (Red Hat mo
dified)
Accept: */*
Host: 192.168.0.103
Connection: Keep-Alive
```

Web server logs will also report the same information that we saw above, Lets confirm it by seeing the get request from our web server logs.

```
[root@slashroot2 ~]# tai
l -f /var/log/httpd/acce
ss_log
192.168.0.104 - - [21/Fe
b/2013:05:34:14 -0500] "
GET / HTTP/1.0" 200 6 "-
" "Wget/1.10.2 (Red Hat
modified)"
```

Step 3: HTTP Server Response to a HTTP GET Request

On getting the GET request from the client, the server responds, by revealing some information about itself, and metadata about the data asked by the client, along with the data.

1. The server sends a status code(this status code informs the client software

Web2PDF
converted by Web2PDFConvert.com

about the result of the request). There are many status codes used by HTTP. We will see some common status codes in HTTP later in this post. Discussing all the status codes used HTTP is beyond the scope of this article.

The server replies to the client with a status code of 200 and the http version it is using. In our case its **HTTP /1.1**.

Status code of 200 means the request has succeeded.

2. Date specifies the date and time at which the response was originated

3. **Server** filed in the response tell's the the server software that's being used(this field is modifiable in almost all web server's. This at times is a security breach as it reveals the server information to the client.)

In our case its **Apache 2.2.3 (Red Hat)**

4. Last modified field informs the client, the modified date and time of the data requested.

5. Etag is a string that identifies the modification of the data requested. This is used for caching. This improves the quality of web caching.

Etag is an identifier assigned by the web server to a specific version of the resource requested.

6. Another important factor that makes http a wonderful protocol is that you can request your required bytes of a resource.

For example, if you are downloading a file of 100M and in between the connection dropped, you can later resume the download by specifying the range of bytes from where to start the download in the GET request.

In our case the server informs the client in its response that it supports bytes range.

7. Content-Length specifies the size of the resource requested in bytes. In our case its just a "hello" printed. Hence its 6 bytes.

8. **Connection: close ,** in our case indicates that the connection will be closed after completion of this particular request.

9. Content-Type specifies the type of the content. In our example its a simple Text/HTML.

Web2PDF
converted by Web2PDFConvert.com

10. finally comes the data after these header's.

All the above mentioned details from the tcpdump output in an http get response is shown below.

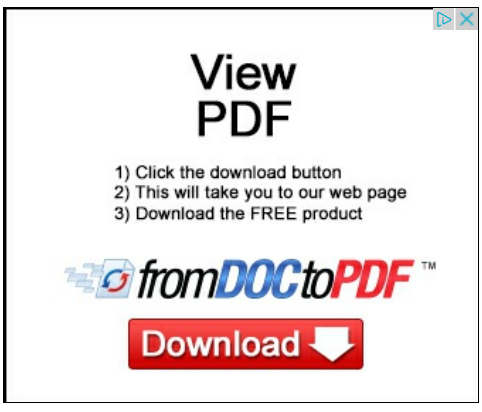
```
05:36:32.786377 IP 192.168.0.103.ht
tp > 192.168.0.104.52802: P 1:268(2
67) ack 121 win 362 <nop,nop,timest
amp 6465082 6458685>
E..?..@.@.#....g...h.P.B..2...B*...
j.Q.....
.b.:.b.=HTTP/1.1 200 OK
Date: Thu, 21 Feb 2013 10:36:32 GMT
Server: Apache/2.2.3 (Red Hat)
Last-Modified: Thu, 21 Feb 2013 09:
47:39 GMT
ETag: "377709-6-f71984c0"
Accept-Ranges: bytes
Content-Length: 6
Connection: close
Content-Type: text/html; charset=UT
F-8

hello
```

HTTP Request Types:

We have already seen the **GET** request type in the above example, let's see some more types of HTTP request types.

HTTP Head Request

HTTP head request is very much similar to the GET request. Its the easiest method to know the complete details of the resource available on a particular URL, without downloading the entire data.



For example, if we use HEAD request in our above example, we will get all the header's in the response except our page containing the "Hello" message.

Web2PDF
converted by Web2PDFConvert.com

This kind of a request is used only to retrieve attributes of the data without the data. This can give you information about the resource by saving your bandwidth.

HTTP Post Request

POST request is mostly used to send data from the client to the server. Let's see an example of HTTP post request from the client to the server.

```
POST /my_data_send.php HTTP/1.1
Host: 192.168.0.105
User-Agent: ELinks/0.11.1 (textmode
; Linux; 80x25-2)
Referer: http://192.168.0.105/
Accept: */*
Accept-Encoding: gzip
Accept-Language: en
Connection: Keep-Alive
Content-Type: application/x-www-for
m-urlencoded
Content-Length: 62
name=sarath&last=pillai&email=&tele
phone=&comments=
```

In the above shown post request the request is being send to **"my_data-send.php"**. This data being send to the server will contain other header's as well that we saw during our GET request example.

The last line sends the exact data to the server.

HTTP Put request:

HTTP put request is very much similar to the post request. PUT request sends or creates a resource in the specified URI.

IF the resource is already present in that specified URI, it will update that URI, otherwise it will create the resource.

HTTP Delete Request:

HTTP delete request deletes a specified resource on a specified URI.

It's not at all advisable to configure a webserver for HTTP delete operation. However if you want to enable such functionalities, its better to that with a http POST operation using a web form, which intern will delete a resource.

HTTP Trace Request

HTTP trace request is used to trouble shoot http web pages.

Web2PDF
converted by Web2PDFConvert.com

For example, if suppose a web page is not getting loaded the way you want in your browser. Then in such cases http trace request can be used to retrieve the complete request that the server got from the client back to the client itself.

This kind of request kind is mostly disabled in most of the web server's. The main reason is that its very much similar to viewing the web server log of the request you send.

Some Famous Error Codes

Some of the widely know HTTP status codes are mentioned below.

400

Http 400 error code stands for **"Bad request Type"**. the server responds with 400 error when the server receives a malformed request of any kind.

403

This status code error is shown by the server when the requested resource is forbidden.

In other words it can be said that you do not have access to the resource you are trying to access.

404

The most famous error code in http is 404. It indicates a not found error. The requested resource is not found on the server.

500

This error code of 500, is used to inform the client that some internal error happened in the server. But this error does not specify a specific problem in the web server.

Or say an unknown error in the web server.

Note: The main purpose of this article was to go through the steps involved in completing a HTTP request.

For more information on HTTP i will

Web2PDF
converted by Web2PDFConvert.com

recommend reading the below links.

HTTP Wikipedia Article

HTTP 1.1 RFC(Request For Comments)

Rate this article: ☆☆☆☆☆
Average: 1.2 (2932 votes)

Add new comment

Comments

Simple and Effective article

Permalink Submitted by Omprakash on Thu, 01/09/2014 - 14:51

This article has helped me in understanding the HTTP functioning in simple and efficient terms. Happy that I could now let people know about HTTP in a better and easy terms.

Many Thanks to the author

reply

Nice Detailed article

Permalink Submitted by Hozze on Sun, 08/10/2014 - 17:02

Explains the concept in a detailed way.

reply

Thank you this has been very

Permalink Submitted by Roger Sicore on Sat, 11/01/2014 - 00:07

Thank you this has been very useful information and has helped me understand alot more.

reply

Good article

Permalink Submitted by Gangadhara Rao on Thu, 06/04/2015 - 15:57

Very informative and simple article.

reply

Superb ! well explained.....

Web2PDF
converted by Web2PDFConvert.com

Permalink Submitted by Neeraj on Tue, 06/23/2015 - 09:48

Superb ! well explained.....

reply

Add new comment

Your name

Subject

Comment *

More information about text formats

- No HTML tags allowed.
- Web page addresses and e-mail addresses turn into links automatically.
- Lines and paragraphs break automatically.

Word verification *

(verify using audio)

Type the characters you see in the picture above; if you can't read them, submit the form and a new image will be generated. Not case sensitive.

Save Preview

Related articles

- Dockerfile Tutorial - Building Docker Images for Containers
- How To Run A Docker Container
- vgcreate vgchange vgconvert vgreduce vgextend vgmerge vgsplit vgrename and vgremove linux command
- Amazing Facts About Linux You Didn't Know
- Docker Tutorial: How to Install and Use Containers

Search Articles ...

Subscribe Our Mailing List

* indicates required

Email Address

Web2PDF
converted by Web2PDFConvert.com

*

First Name

Last Name

Subscribe

Today's Most Popular

vgcreate vgchange vgconvert vgreduce vgextend vgmerge vgsplit vgrename and vgremove linux command

Archives - 1 comment(s)

IPERF: How to test network Speed,Performance,Bandwidth

Archives - 9 comment(s)

CURL command Tutorial in Linux with Example Usage

Archives - 10 comment(s)

MX Record in DNS Explained with Example Configurations

Archives - 0 comment(s)

Most Commented

How Does Traceroute Work and Example's of using traceroute command

Networking - 44 comment(s)

Logstash Tutorial: Linux Central logging server

Archives - 31 comment(s)

Web2PDF
converted by Web2PDFConvert.com

difference between iterative and recursive dns query

Archives - 28 comment(s)

SAN vs NAS - Difference between a Storage Area Network and Network Attached Storage

Archives - 26 comment(s)

Top Rated Articles

Find Command Usage and Examples in linux

Average: 5 (2 votes)

Modify your swap space by configuring your Linux machine to use LVM as Swap Space.

Average: 5 (6 votes)

Linux Antivirus and Virus Scanning

Average: 5 (24 votes)

NMAP in Linux Examples

Average: 5 (1 vote)

static routing configuration

Average: 5 (4 votes)

website vulnerability scanner

Average: 5 (1 vote)

how to install and configure git

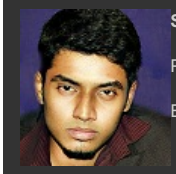
Average: 5 (1 vote)

Amazing Facts About Linux You Didn't Know

Average: 5 (1 vote)

Jump back to navigation

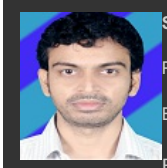
Get in touch with The Authors



Sarath Pillai

Ph: +917303074400

Email: sarath@slashroot.in



Satish Tiwary

Ph: +919509452488

Email: satish@slashroot.in

Follow Us

Subscribe to our RSS Feed

Web2PDF
converted by Web2PDFConvert.com

Follow us on Twitter

Be a fan on Facebook

Our News Letter

* indicates required

Email Address *

Subscribe

Recent Posts

Dockerfile Tutorial - Building Docker Images for Containers

Archives - *2 months 1 week* ago

How To Run A Docker Container

Archives - *4 months 3 weeks* ago

vgcreate vgchange vgconvert vgreduce vgextend vgmerge vgsplit vgrename and vgremove linux command

Archives - *6 months 1 week* ago

Amazing Facts About Linux You Didn't Know

Archives - *6 months 2 weeks* ago

Last Viewed

How to configure a Router to work as a DHCP Server.

Networking - last view *8 sec* ago

SAN vs NAS - Difference between a Storage Area Network and Network Attached Storage

Archives - last view *16 sec* ago

Linux grub(Grand Unified Bootloader) tutorial

Archives - last view *17 sec* ago

How are passwords stored in Linux (Understanding hashing with shadow utils)

Security Notes - last view *31 sec* ago

Copyright © Slashroot.in.All rights reserved.Privacy Policy|Terms Of Service|Sitemap

Web2PDF
converted by Web2PDFConvert.com