Hands-on:
HTTP Request Challenge

Internet Engineering July 17th 2019
By Teaching Assistant

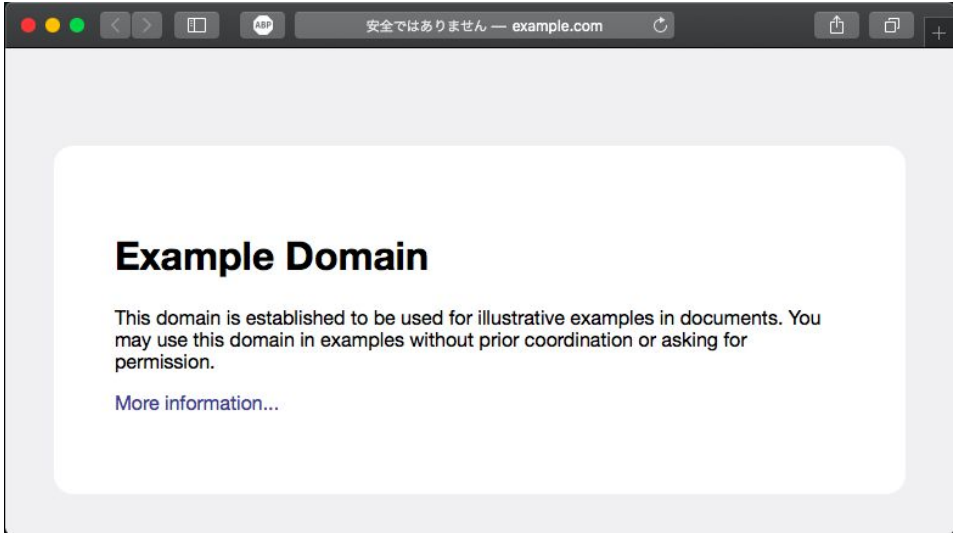
Agenda

1. Resolve hostname
2. Access server using by browser
3. Access server using by telnet
4. Compare the HTTP header

Query http request packet by Browser

Run wireshark and capture the packets.
After that, access `http://example.com/` by your browser.

NOT `https://example.com`



Observe the packet

It might be a lot of packets on your wireshark.
To show only the packets of example.com, use `ip.addr` filter.

```
ip.addr == <target ip>
```

You can't use `example.com` because it's domain name.
Therefore, we need to get ip address.

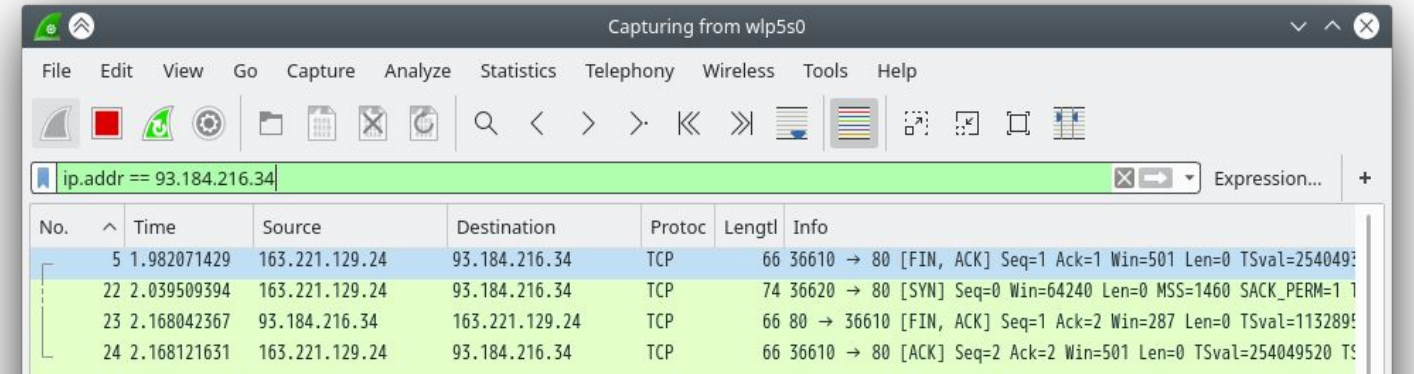
Get IP Address from Domain Name

To get IP address from domain name, dig command is useful like as follows:

```
$ dig example.com -t A +short
93.184.216.34
```

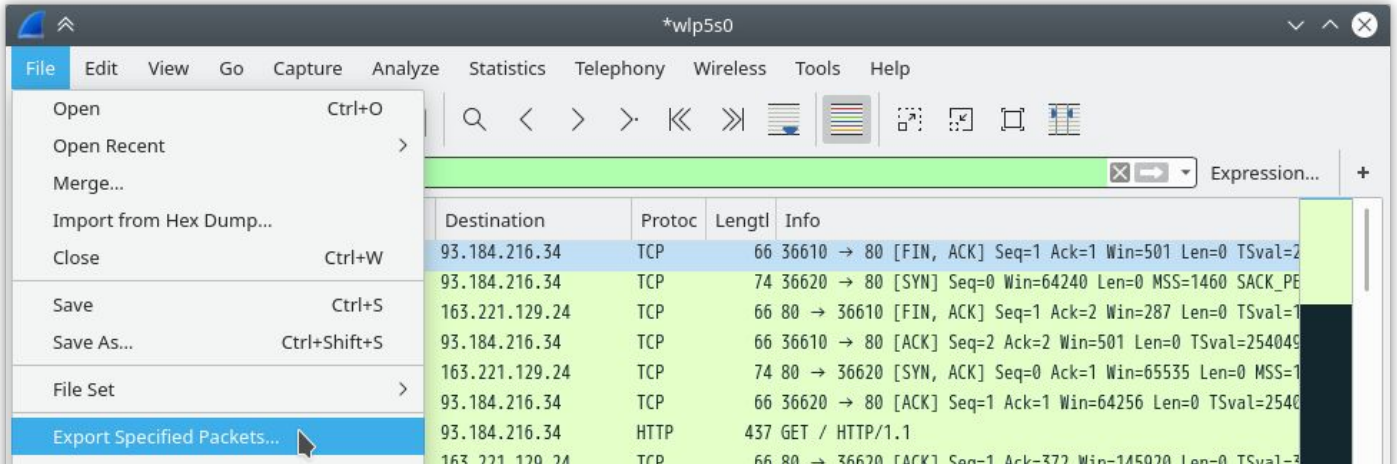
Filter the packets

You got the ip address.
Let's filter the packets regarding with `example.com`



Save the packets

Stop capturing packets and click `[File] -> [Export Specified Pakcets…]`.
NOT `[Save]`.



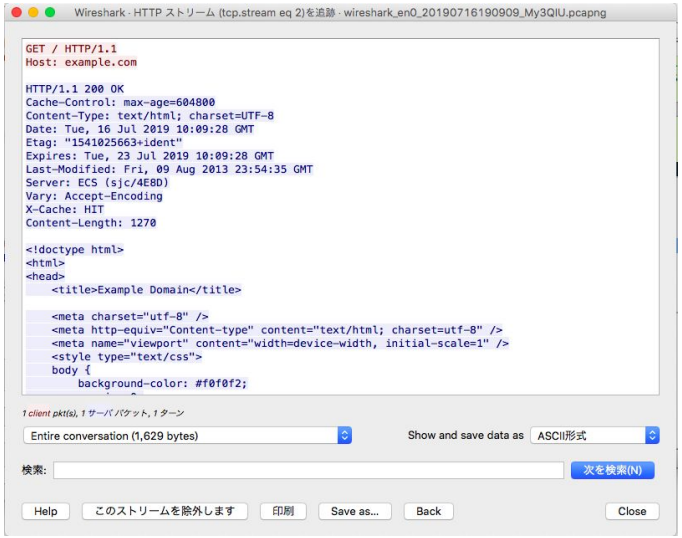
Query http request packet by telnet

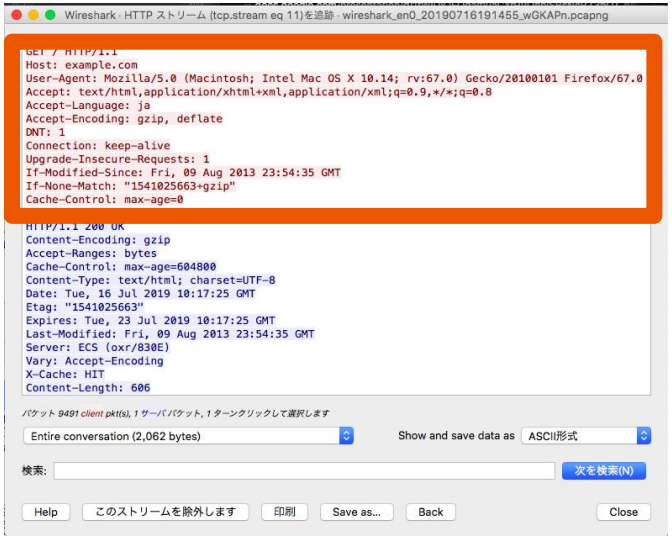
Run wireshark and capture the packets.
You can access server by telnet as well, like as follows:

```
$ telnet example.com 80
Trying 93.184.216.34…
Connected to example.com.
Escape character is '^]'.
GET / HTTP/1.1
Host: example.com
```

```
root@docker:~# telnet example.com 80
Trying 93.184.216.34...
Connected to example.com.
Escape character is '^]'.
GET / HTTP/1.1
Host: example.com

HTTP/1.1 200 OK
Cache-Control: max-age=604800
Content-Type: text/html; charset=UTF-8
Date: Tue, 16 Jul 2019 10:00:30 GMT
Etag: "1541025663+ident"
Expires: Tue, 23 Jul 2019 10:00:30 GMT
Last-Modified: Fri, 09 Aug 2013 23:54:35 GMT
Server: ECS (oxr/8316)
Vary: Accept-Encoding
X-Cache: HIT
Content-Length: 1270

<!doctype html>
<html>
<head>
    <title>Example Domain</title>

    <meta charset="utf-8" />
    <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
```

Comparison of HTTP header





Why does the browser add these header?

Think the meaning of these header info

- What's the meaning of `GET / HTTP/1.1`?
- Describe the differences between `HTTP/1.0` and `HTTP/1.1`.
- What's the purpose of `keep-alive`?
- What's the meaning of `200 OK`?