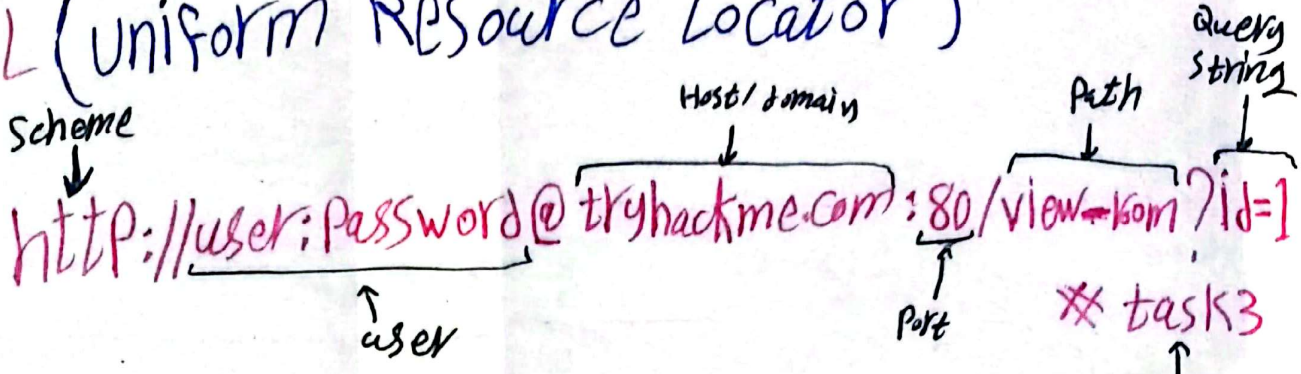
* عبارة عن بروتوكول يستخدم لنقل البيانات عبر الإنترنت، وهو الأساس الذي يعتمد عليه تصفح الإنترنت، لما بتدخل عنوان URL يتم استخدام HTTP لإرسال طلب (Request) إلى الـ server ومن ثم يستجيب السيرفر بـ Response.

* Stateless protocol مش بيحتفظ بمعلومات بتاعت الـ Requests بين بعض يعني لو انا دخلت على الفيسبوك مرة بيه وجيت ادخل الفيسبوك تاني هيفضل أكتب الأيميل والباسورد تاني

* كمان بيدعم أكتر من Request في نفس الوقت في نفس الـ connection وده يزود سرعة التصفح ويقلل الوقت اللي محتاجه لعمل connection

URL (uniform Resource Locator)



① Scheme: this instructs on what protocol to use for accessing the resource such a HTTP, HTTPS, FTP.

② User: some services require authentication to log in, you can put a user name and password into the URL to log in.

بس حاليا معدش الـ user ده بيتكتب في الـ URL لأنك بتكتب الباسورد فيه فممكن حد يخترقك

③ Host: the domain name or IP of the server you wish to access.

④ Port: the port that you are going to connect to, usually 80 for HTTP and 443 for HTTPS.

أوقات مش بيتكتب

⑤ Path: the file name or location of the resource you are trying to access.

الصفحة اللي عاوز تفتحها على السيرفر ده

⑥ Query string: Extra bits of information that can be sent to the requested path

بيتكتب بعد ؟ وده معتبرانا بقوله يبحث عن ايه في المسار ده

زي مثلا https://www.google.com/search?q=cats

يبحث عن cats

⑦ Fragment: ※ بعد الـ

بيقوله انه يوديني عند مكان معين في الصفحة يعني هنا بيقوله وديني في task3 باستخدمه لو محتوى الصفحة كبير

Requests and Responses

Request?

```
GET / HTTP/1.1
Host: tryhackme.com
User-Agent: Mozilla/5.0 Firefox/87.0
Referer: https://tryhackme.com/
```

Line 1: this Request is sending the GET method, request the home page with / and telling the webserver we are using HTTP protocol version 1.1

Line 2: we tell the webserver we want the website tryhackme.com

Line 3: we tell the web server we are using firefox version 87 browser

Line 4: https://tryhackme.com/ يقول للسيرفر الصفحة اللي عاوزه يجيبهالي

* آخر حاجه في الـ Request بيبقا سطر فارغ عشان اعرف السيرفر ان الـ Request خلص

Response?

```
HTTP/1.1 200 OK
Server: nginx/1.15.8
Date: Fri, 09 Apr 2021 13:34:03 GMT
Content-Type: text/html
Content-Length: 98

<html>
<head>
</head>
</html>
```

HTTP/1.1 نسخة HTTP السيرفر بيقول للبراوزر اني هو مستخدم

200 OK يعني بيقوله إن الـ Request بتاعك تم بنجاح

Server: nginx/1.15.8 ← الـ Software بتاع السيرفر والـ Version بتاعته

Date: Fri, 09 Apr 2021 13:34:03 GMT

Content-Type: text/html

* الـ Content-Type عبارة عن header بيوضح الـ client ايه نوع البيانات اللي جايلك سواء كانت html أو صورة أو فيديو وهكذا

Content-Length: 98

* الـ Content-Length بيعرفك حجم الداتا اللي هيتبعتلك

<html> سطر فارغ

<head>

* قبل ما الصفحه تجيلك في سطر فارغ عشان يعرفك إن دا نهاية الـ Response بعد ما المعلومات اللي طلبها يتجيلك زي إن الموقع فتح صفحه والصفحة بتاعته ظهرت

</html>

http methods

3

ال methods دي عشان اقول للـ server انا عاوز اي بالظبط
هل عاوزك تعرضلي محتوى بس ولا عاوز اسجل دخول ولا عاوز امسح داتا

1- GET Request: this is used for getting information from a web server
الحصول على معلومات فقط ممكن نوصل بيها بياناته بس مش أمان لانك بتبعت البيانات في الـ (url)

2- POST Request: this is used for submitting data to the web server and creating new records
لارسال بيانات أو عمل سجلات جديدة

3- PUT Request: this is used for submitting data to a web server to update information.
لتحديث البيانات = الموجودة فعلا على الـ server ذي اللي تغير صورة أو شكل

4- DELETE Request :- this is used for deleting information /records from a web server.

HTTP status codes

بيبقا في أول سطر من الـ Response ذي الـ 200 كدا ولكن ممكن انه يبعت
الـ client نتيجة الـ Request بتاعته أي هل اتقبل ولا لا ولا ايه اللي حصله

100-199 → information Response عشان يعرفه لو الـ client بيتبعت
أول جزء من الطلب بتاعه اتقبل وانه المفروض يبعت بقيت طلبه

200-299 → success أداء طلب معين تم اقبل بس لسه بيتعمل عليه شوية عمليات
عشان يعرف لو الـ Request نجح

300-399 → Redirection:- these are used to redirect the client's request to another resource. this can be either to a different webpage or a different website altogether.

400-499 → Client Errors

500-599 → server errors

Headers

common Request headers

دول الـ headers اللي بتتبعت من الـ client لـ server

Host:. Some web server host multiple websites so by providing the host headers you can tell it which one you require, otherwise you will just receive the default website for the server.

User-Agent :- This is your browser software and version number
لو انت بعت داتا للسيرفر فالـ header دا بيعرفه انت بعتلو اي

content-Length:- عشان السيرفر يتأكد لو الداتا وصلت ومحصلش فقد فيها

Scanned with CamScanner

Accept-Encoding :- Tells the web server what types of compression methods the browser supports so the data can be made smaller for transmitting over the internet.

Cookie:- Data sent to the server to help remember your information

Common Response Headers

* عبارة عن headers اللي بترجع لل client من السيرفر

set-Cookie : بيعرفك انه خزن المعلومات عشان لو جيت تدخل تاني يكون عارفك

Cache-control : يستخدم للتحكم في كيفية تخزين البيانات في ذاكرة المتصفح أو ال cache الخاصة بيه ، و تحديد متى يجب طلبها مرة أخرى من ال server بحيث بعد المدة متخلص يرجع ال Request لل browser تاني وهكذا

Content-type :- نوع الداتا اللي راجعه لل client

Content-Encoding :- ايه هي الطريقة اللي بيعرف ال client بيها ال server استخدمها في ضغط البيانات عشان يعرف يفكها على الانترنت

HTTP is Stateless

* every Request is completely independent

يعني لو انت فتحت صفحة وجيت طلب معينه فبتعمل طلب لو جيت تفتح صفحه تانيه بتقدم طلب من الأول تاني ولو جيت تفتح نفس الصفحه تاني اللي كنت فاتحها بتقدم طلب من الأول برضك

* بيتعامل مع كل Request لوحده (independent)

Subject : Date : SAT SUN MON TUE WED THU FRI

Curl بيعمل عملية تحويل الـ URL و data بتاعة الـ Request وبيعمل Request لـ URL وكمان بيحدد الـ method بتاع الـ Request سواء GET أو POST

* Apache2 ده عبارة عن web server

* curl عبارة عن commands بيستخدم لعمل Requests للسيرفرات على الإنترنت باستخدام بروتوكولات مثل الـ HTTP و HTTPS و FTP كمان ممكن استخدمه لتحميل الملفات وارسال بيانات و استقبال ردود من السيرفرات

curl www.google.com ده كده بيبعت Request بـ GET وبطلب يجيلي صفحة جوجل فهيرد بصفحة جوجل بالـ HTML

* لو عاوز كمان يطبعلي الـ Headers بتاعت الـ Request ضيف -i

(زي الـ set-cookie و content-type)

curl -i www.google.com

curl -O myfile.zip https://example.com/File.zip

كده انا بنزل ملف اسمه file.zip من على موقع example.com وبقوله سميه لما ينزل على الجهاز myFile.zip

* الـ O small ← الـ O capital

* لو مش عاوز تسميه حاجة -O ← الـ O capital

curl -O https://example.com/file.zip

* نوع الـ method اللي بيستخدمها curl default هي GET

لو عاوز اغيرها اعمل -X (اسم الـ method)

curl -X POST -d "name=abd&age=20" https://example.com/

-X capital

-d عشان هبعت داتا

Scanned with CamScanner ممسوح ضوئيا بـ CamScanner

استخدام CURL للاتصال بموقع يستخدم HTTPS

* عند محاولة الاتصال بموقع يستخدم HTTPS يقوم CURL بالتحقق من صحة شهادة SSL لضمان أن الاتصال آمن، إذا كانت الشهادة غير صالحة أو منتهية الصلاحية أو لم يتم التحقق منها من جهة موثوقة، فإن CURL يمنع الاتصال افتراضياً لحمايتك من هجمات MITM (Man-in-the-Middle)

كيفية تجاوز مشكلة SSL في CURL؟ استخدم -k

curl -k https://

وهكذا يتم تعطيل التحقق من شهادة الـ SSL

Scanned with CamScanner
تم المسح ضوئيا بـ CamScanner