ABOUT

Yoann "fuzzy" Lamouroux

- Sysadmin/Technical Leader at @nbs-system
- Security enthusiast
- Suze advocate (not the Linux RedHat-ish distro)
- Likes: python and vim

trolls: @xoxopowo (twitter), legreffier (freenode)

ylamouroux@ubuntu.com
TRIVIA

Started in 1996, by Daniel Stenberg ... for an IRC bot project

- Available almost everywhere
- >20 years old, still getting updates
- Integrated with many tools
- MIT/X License (quite close to BSD)

means: "Client URL Request Library"
silly one and only curl-ing pun in this presentation
QUITE A LOT OF USE-CASE
it is misused a lot too
You usually need it for a quick check.

Manpage = 2700 lines

Therefore: this mini-talk
HTTP

• What the web is built upon (but you probably knew that)
• It's (usually) how your apps will talk
  ▪ API!
  ▪ REST!
  ▪ IoT!
• It's a TCP protocol (it's reliable, it needs an IP)
CENSORSHIP

I won't say anything about:
DNS

how a domain is matched to one or more IPs

Just don't mix up the:
• domain name:
  the mechanism to get an IP from a domain name

• and the "Host" header
  the actual site you'll request to an IP

(more on "Host" later)
HTTPS

how the http gets wrapped in a ssl-encrypted tunnel

Just patch your things

(ʃ °□°)ʃ︵┻━┻
YOUR COMPUTER SAYS TO THAT IP:

1. I want '/'
2. on the site named: example.com
3. Some more info (about your browser and what it can do)
4. Even more info (if you were already there)
5. [nothing] (it will actually send an empty line)

Only 1st, 2nd and 5th steps are mandatory in an HTTP request
Connected to example.com (93.184.216.34) port 80 (#0)
> GET / HTTP/1.1
> Host: example.com
> User-Agent: curl/7.58.0
> Accept: */*
>
THE SERVER ANSWERS:

1. HTTP Return code (200, 404, 50x (oh sh...))
2. Some infos about the datas (size, type, taste)
3. Some infos for you (your browser) in case you come back 🍪
4. Stuffs (html... if you're on the web)
CLIENT ← SERVER

< HTTP/1.1 200 OK
< Cache-Control: max-age=604800
< Content-Type: text/html
< Date: Thu, 28 Jun 2018 17:03:42 GMT
< Etag: "1541025663+ident"
< Expires: Thu, 05 Jul 2018 17:03:42 GMT
< Last-Modified: Fri, 09 Aug 2013 23:54:35 GMT
< Server: ECS (dca/532C)
< Vary: Accept-Encoding
< X-Cache: HIT
< Content-Length: 1270
<
IT'S WHAT YOUR BROWSER DO:

🕺 🕺 🕺 🕺 🕺

 waypoints

inb4 epic tcpdump capture coming your way.
tts fuzz ylmrx $ sudo tcpdump -qA -i any "dst 93.184.216.34 and tcp port 80 and (((ip[2:2] - ((ip[0]&0xf)<<2)) - ((tcp[12]&0xf)>>2)) != 0)"
REQUEST HTTPS

OMG FAEK

DUMPS ON HTTP
HOW NOT TO USE CURL
curl -vI https://example.com/
• Sometimes HEAD is not allowed
• It will only get metadata (Headers)
• **this is not a reliable test** (you're not issuing the right request)
You actually want:

curl -v https://www.example.com > /dev/null

Show your 1337-skills, omg. OMG!, /dev pseudo-file and stream redirection 😻
Despite you can use ":-o [FILE]:" to output to any file instead of stdin, there's no direct option to disable output.
You're testing (locally?) some website, you need to have a resolution to an IP you and trick DNS for whatever reason.

you usually don't need to edit /etc/hosts
DO YOU EVEN RESOLVE?

curl -v --resolve www.example.com:80:127.0.0.1 http://www.example

See mom? No sudo vim in /etc!
DO YOU EVEN .CURLRC?

You don't want to type this long command every time!

Edit ~/.curlrc, add those options:

```
--resolve www.example.com:80:127.0.0.1
```

(you can add many off these "--resolve" or whatever curl option)
DO YOU SCRIPT?

I saw this, once:

curl -v https://anothercoolsite.com/ 2>&1 | grep -v "HTTP/1.1 200" 
echo "Something was wrong"

- curl has many exit codes.
- just echo that "$?" bad boy!
- too long to list, read the man.
A bad HTTP code (404, 504, ...) is not considered as an error by curl by default (it succeeded at making a request, which failed).

Use "-f", so curl actually crashes.

curl -vf https://anothercoolsite.com/
[ $? -eq 22 ] && echo "something was wrong"
Or even:

curl -f https://anothercoolsite.com/ || echo "something was wrong"
X-Forwarded-*, Host, User-Agent, know the basics
No matter what, -H got you covered. (= "--header")
"-A 'Opera 4.0'" = "-H 'User-Agent: Opera 4.0'"
"That's cool for the trivial work, but I live in a real world. With real things."
(Twilight Sparkle, in My Little Pony, S4E08)
Now you can POST, PUT (and MORE)!

You can now auth to some services and post your useful original opinion on a blog.
You'll need the `--data` option for those to be useful.

YOU CAN RECORD COOKIES!
Usually goes something like:

- curl -X POST --data "login=admin;password=wowmuchsecure" --cookie-jar myjar.txt https://website.com
- get the auth cookie in myjar.txt
- curl --cookie "auth=1234567890abcdef" https://website.com/
TIMINGS!

... and several other nice infos

- option is: --write-out "FMT_STRING"
- FMT_STRING: "foo bar %{var-name}"
- var-name:
  - time_total, time_connect, ...
  - size_download, ...
  - So many moar 😄
Firefox

You can have this automagically from Firefox:

- Developer Tools > Network > GET /
- Right click : Copy as Curl command!
- And work from there
You have a similar feature in BurpSuite which is a nice tool.
Shouldn't have talked about it... it ain't FOSS.
There's a ton more features:

- FTP
- HTTPS
- http/2 (if it's recent enough)
- SMTP/POP
SMTP, REALLY
IT WRITES YOUR C CODE!

Introducing: --libcurl
I barely scratched the surface. Use the talk as-is, or go check the manpage.
Thank you.