How to go to Google.com

- `auto webpage = HttpGet(“www.google.com“);`

Who am I?

Who am I?

- **DHCP Discover**
  - From: Nobody @ Spartacus‘ NIC
  - To: Everybody @ Everywhere
  - Does anybody have a name for me?

Who am I?

- **DHCP Discover**
  - From: DHCP Server
  - To: Nobody @ Spartacus NIC
  - I can offer you the address 192.168.2.5 (== Spartacus). Let me know if you want it.

Who am I?

- **DHCP Request**
  - From: Nobody @ Spartacus‘ NIC
  - To: DHCP Server
  - I‘d like to be spartacus, thanks!

Who am I?

- **DHCP Request**
  - From: DHCP Server
  - To: Nobody @ Spartacus NIC
  - You‘re now Spartacus for me. You have the lease for a week.

DHCP tells you

- **Your IP address (192.168.2.5 == Spartacus)**
- **Your netmask (255.255.255.0)**
  - how big is the local area
- **Your gateway IP address (192.168.2.1)**
  - Gateway has to be local

- **Your DNS server IP address**
  - May not be local

Does anybody else claim to be me?

- **Send out an ARP**
  - Is Spartacus here?

- **Send out a reverse ARP**
  - I‘m Spartacus

Now to find Google

- **Send DNS server a DNS request on port 53**

Now to find Google

- **Send DNS server a DNS request on port 53**
  - Where is the DNS server? Assume it‘s not local
- **Ask the gateway to send a message to the DNS server**

Now to find Google

- **Send DNS server a DNS request on port 53**
  - Where is the DNS server? Assume it‘s not local
- **Ask the gateway to send a message to the DNS server**
  - But where is the gateway?
- **Send ARP message asking where to find the gateway**

We found google!

- **Now to connect. Messages through gateway on IP:**
- **TCP SYN – I‘d like to connect to you.**

We found google!

- **Now to connect. Messages through gateway on IP:**
- **TCP SYN – I‘d like to connect to you.**
- **Receive TCP SYN+ACK: I heard you. I‘d like to connect back too.**

We found google!

- Now to connect. Messages through gateway on IP:
- TCP SYN – I‘d like to connect to you.
- Receive TCP SYN+ACK: I heard you. I‘d like to connect back too.
- Send TCP ACK: We‘re now connected

.. Google uses HSTS

- **Cannot use HTTP any more**

- **Let‘s set up the SSL connection**

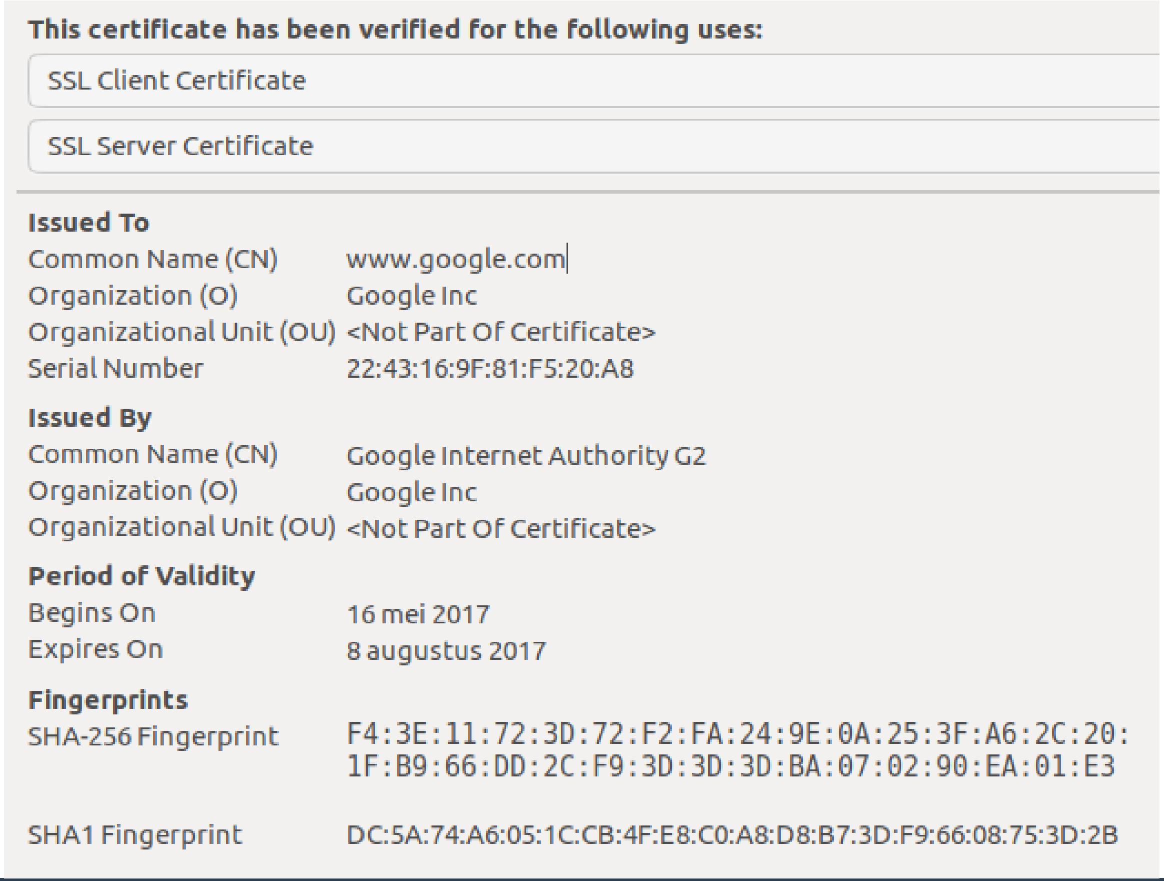
SSL connect

- **Client Hello**
- **Hi. I‘d like to talk to you securely using algorithms X, Y or Z**

SSL connect

- Server Hello
- Let‘s talk in algorithms X, A or B.
- Here‘s my server certificate

SSL connect



This certificate has been verified for the following uses:

SSL Client Certificate

SSL Server Certificate

Issued To

Common Name (CN)	www.google.com
Organization (O)	Google Inc
Organizational Unit (OU)	<Not Part Of Certificate>
Serial Number	22:43:16:9F:81:F5:20:A8

Issued By

Common Name (CN)	Google Internet Authority G2
Organization (O)	Google Inc
Organizational Unit (OU)	<Not Part Of Certificate>

Period of Validity

Begins On	16 mei 2017
Expires On	8 augustus 2017

Fingerprints

SHA-256 Fingerprint	F4:3E:11:72:3D:72:F2:FA:24:9E:0A:25:3F:A6:2C:20: 1F:B9:66:DD:2C:F9:3D:3D:3D:BA:07:02:90:EA:01:E3
SHA1 Fingerprint	DC:5A:74:A6:05:1C:CB:4F:E8:C0:A8:D8:B7:3D:F9:66:08:75:3D:2B

- **So can we trust this server?**

- **Certificate is signed by a trusted CA**
  - Cryptographic check to see it actually checks out
  - We trust him.

- **Does the server actually have the private key?**

- **Let‘s send him a challenge**
  - Encrypted with public key
  - Only way to decrypt is if you have the private key
  - … so if he can successfully do that, he owns the private key

- **Let‘s send him a challenge**
  - Encrypted with public key
  - Only way to decrypt is if you have the private key
  - … so if he can successfully do that, he owns the private key

- **Yay! He got the right answer! Now let‘s exchange keys & secure the channel**

And then finally...

```
GET / HTTP/1.1
Host: google.com
User-Agent: CppNow-LightningTalk v1.0
```

And then finally...