

Laboratorio di Internet

Relazione #4 su applicazione scelta dal gruppo

Lo scopo della tesina di gruppo è osservare il comportamento di applicazioni che usano la rete e capire quale traffico esse scambino a fronte di eventi noti. Usando Wireshark, il debugger di Chrome ed altri software e strumenti, si osserva il traffico generato dal vostro terminale (PC o smartphone) quando viene eseguita l'applicazione per capire come questa usa internet.

Predisporre un terminale sotto il vostro controllo, configurato normalmente per accedere ad internet con una regolare connessione. Dovete essere in grado di catturare il traffico scambiato dal terminale usando Wireshark. Potete quindi eseguire Wireshark sul terminale stesso con sistema operativo Windows, Mac OS, Linux. In caso si voglia catturare il traffico generato da una app in esecuzione su un cellulare, potete configurare un PC in modo che condivida l'accesso ad internet via WiFi, e connettere il cellulare al WiFi del PC. Catturando sulla interfaccia WiFi del PC, sarà possibile osservare tutto il traffico scambiato dal cellulare. Lo scenario risulta il seguente:

Cellulare <== [WiFi] ==> PC <== [ADSL/FTTC/FFTH/Tethering con altro cellulare] ==> Internet

Iniziate a catturare il traffico e quindi eseguite l'applicazione di vostro interesse sul terminale. Fate una serie di operazioni precise, in modo circostanziato e controllato. Per esempio, effettuate il login, mandate un messaggio, scaricate o caricate dei dati, introducete un guasto (scollegando il cavo per esempio), abbandonate l'applicazione, ecc. Eseguite le operazioni in modo indipendente e separatamente.

Per ogni operazione, analizzate la traccia con Wireshark, filtrando ove opportuno il traffico relativo al vostro host, e cercando di capire quale traffico l'applicazione invia a fronte di una determinata azione. Ripetete l'esperimento più volte per verificare che i risultati siano consistenti.

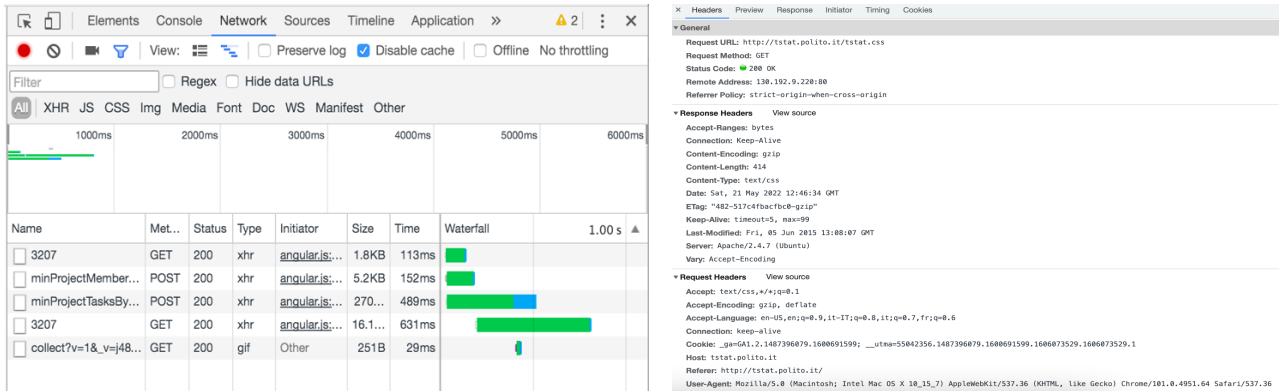
Esempi di informazioni utili da considerare

1. Quali protocolli di livello trasporto e di livello applicazione vengono usati?
2. Quali nomi hanno i server che sono contattati? A quali reti appartengono? Chi è l'amministratore della rete? Dove sono localizzati i server?
3. A fronte di un evento, che traffico genera l'applicazione?
4. È possibile identificare le funzioni svolte dal server? Per esempio, per effettuare l'autenticazione o per scaricare un video, una pubblicità, oggetti di grandi dimensioni?

Alcuni suggerimenti

- Limitare il più possibile traffico di background generato da altre applicazioni/servizi in esecuzione sul terminale stesso, o da altri terminali (se su rete condivisa WiFi). Filtrate il traffico che non interessa. Per esempio, se volete catturare solo il traffico IP generato dal vostro host, potete usare il *filtro di cattura* `ip and host <IP_ADDRESS> and not (broadcast or multicast)`. Questo escluderà il traffico generato da altri host, traffico non IP (ARP, STP, ...), nonché il traffico broadcast e multicast (SSDP, MDNS, ...).
- Configurare Wireshark perché risolva gli indirizzi IP in nomi *View -> Name Resolution -> Resolve Network Addresses*. Spesso il nome di un server ha indicazioni sia sul suo scopo, sia posizione, sia sull'amministratore dello stesso.

- Se la vostra applicazione gira via browser, usate il **debugger di Chrome** per capire quale traffico HTTP(S) genera. Per aprire il debugger, cliccare col tasto destro sulla pagina e selezionare **“Ispeziona”**. Dopodiché, andate sulla **Tab “Network”**. A questo punto ricaricate la pagina. Dovreste ottenere il **“Waterfall”** delle **richieste HTTP** come nella prima immagine sotto. Cliccando su una richiesta HTTP, potete vederne i dettagli, analizzando gli header della richiesta e della risposta (vedi seconda immagine sotto). Utilizzate il debugger per capire quante richieste HTTP ogni pagina fa, a quali server e quali tipi di oggetti scarica (immagini, script, font). Provate differenti funzionalità del sito e vedete quali differenti pattern di richieste HTTP (e verso quali domini) esso genera.



- Potete anche fare un lookup diretto (risolvere un nome in un indirizzo IP) o un reverse-lookup (risolvere un indirizzo IP in un nome). Ricordate che un indirizzo IP può essere associato a tanti nomi. E un nome può essere associato a tanti indirizzi IP. Per fare il lookup da linea di comando usate il comando

host <nome> => si ottiene uno o più indirizzi IP associati a quel nome
 host <ip_address> => si ottiene uno o più nomi associati a quell'indirizzo

Non sempre il reverse-look up ha successo perché l'amministratore può disabilitare questa possibilità. Qui un esempio:

```
(base) mellia@MBP-di-Marco ~ % host www.youtube.com
www.youtube.com is an alias for youtube-ui.l.google.com.
youtube-ui.l.google.com has address 216.58.208.174
youtube-ui.l.google.com has address 216.58.205.78
youtube-ui.l.google.com has address 216.58.198.14
youtube-ui.l.google.com has address 216.58.209.46
youtube-ui.l.google.com has address 216.58.206.46
youtube-ui.l.google.com has address 216.58.198.46
youtube-ui.l.google.com has address 172.217.21.78
youtube-ui.l.google.com has address 216.58.208.142
```

indica che al nome www.youtube.com possono rispondere indifferentemente 7 indirizzi IP. Mentre

```
(base) mellia@MBP-di-Marco ~ % host 216.58.198.46
46.198.58.216.in-addr.arpa domain name pointer mil04s04-in-f46.1e100.net.
46.198.58.216.in-addr.arpa domain name pointer mil04s04-in-f14.1e100.net.
```

indica che l'indirizzo IP 216.58.198.46 ha nome mil04s04-in-f46.1e100.net (nome dei server della CDN di Google).

- Usare le funzioni di **“Statistics -> Endpoints”** per visualizzare informazioni sugli indirizzi IP dei server contattati, e a livello di servizi su UDP e TCP. Osservare il nome dei server con cui si scambiano più dati, le reti a cui appartengono, la loro posizione, ecc.

- In modo analogo, usare "Statistics -> Conversation" per analizzare le singole connessioni TCP, e i flussi UDP generati.
- Usare le funzioni per fare grafici nel tempo "Statistics -> I/O Graphs", o di sequenze di pacchetti "Statistics -> Flow Graph", o di evoluzione dei numeri di sequenze "Statistics -> Flow Graph" se interessanti.
- Usare il servizio "whois" per ottenere delle informazioni in proposito agli indirizzi IP. Per esempio

```
(base) mellia@MBP-di-Marco ~ % whois 216.58.198.46
% IANA WHOIS server
% for more information on IANA, visit http://www.iana.org
% This query returned 1 object

refer:          whois.arin.net

inetnum:        216.0.0.0 - 216.255.255.255
organisation:   ARIN
status:         ALLOCATED

whois:          whois.arin.net

changed:        1998-04
source:         IANA

# whois.arin.net

NetRange:      216.58.192.0 - 216.58.223.255
CIDR:         216.58.192.0/19
NetName:      GOOGLE
NetHandle:    NET-216-58-192-0-1
Parent:       NET216 (NET-216-0-0-0-0)
NetType:      Direct Allocation
OriginAS:     AS15169
Organization: Google LLC (GOGL)
RegDate:      2012-01-27
Updated:      2012-01-27
Ref:          https://rdap.arin.net/registry/ip/216.58.192.0

OrgName:      Google LLC
OrgId:        GOGL
Address:      1600 Amphitheatre Parkway
City:         Mountain View
StateProv:    CA
PostalCode:   94043
Country:      US
RegDate:      2000-03-30
Updated:      2019-10-31
Comment:      Please note that the recommended way to file abuse complaints are located in the
following links.
Comment:
Comment:      To report abuse and illegal activity: https://www.google.com/contact/
Comment:
Comment:      For legal requests: http://support.google.com/legal
Comment:
Comment:      Regards,
Comment:      The Google Team
Ref:          https://rdap.arin.net/registry/entity/GOGL

OrgAbuseHandle: ABUSE5250-ARIN
OrgAbuseName:  Abuse
OrgAbusePhone: +1-650-253-0000
OrgAbuseEmail: network-abuse@google.com
OrgAbuseRef:   https://rdap.arin.net/registry/entity/ABUSE5250-ARIN

OrgTechHandle: ZG39-ARIN
OrgTechName:   Google LLC
OrgTechPhone:  +1-650-253-0000
OrgTechEmail:  arin-contact@google.com
```

OrgTechRef: <https://rdap.arin.net/registry/entity/ZG39-ARIN>


Mostra come questo indirizzi IP e tutta la rete 216.58.192.0/19 siano amministrati da Google.

- Analogamente, usare servizi come <https://www.iplocation.net> per trovare la posizione geografica del server (operazione di geolocalizzazione). Esistono tanti database che offrono il servizio di geolocalizzazione degli indirizzi IP in Internet. Fate attenzione nelle informazioni che questi possono offrire che possono risultare non affidabili. Tra questi, il servizio offerto da MaxMind <https://www.maxmind.com> o <https://db-ip.com> possono essere utilizzati anche via API o direttamente da Wireshark. Per configurare quest'ultimo affinché usi il database GeoIP-lite di MaxMind potete usare le istruzioni in <https://wiki.wireshark.org/HowToUseGeoIP>

Attenzione: in generale è difficile geolocalizzare un indirizzo IP. Non fidatevi ciecamente del risultato ottenuto. Per esempio, un indirizzo IP di un server di Google potrebbe essere localizzato a Mountain View, California. Se fate un ping allo stesso, può il RTT essere di 10ms?

Per esempio


Geolocation data from [IP2Location](#) (Product: DB6, updated on 2020-5-1)

IP Address	Country	Region	City
216.58.198.46	United States of America 	California	Mountain View
ISP	Organization	Latitude	Longitude
Google LLC	Not Available	37.4060	-122.0785

Geolocation data from [ipinfo.io](#) (Product: API, real-time)

IP Address	Country	Region	City
216.58.198.46	United States 	California	Mountain View
ISP	Organization	Latitude	Longitude
Google LLC	Google LLC (google.com)	37.4056	-122.0775

Geolocation data from [DB-IP](#) (Product: Full, 2020-5-1)

IP Address	Country	Region	City
216.58.198.46	Italy 	Lombardy	Milan
ISP	Organization	Latitude	Longitude
Google LLC	Google LLC	45.4642	9.18998

Mostra che questo IP si trova negli US secondo due database (IP2Location e ipinfo.io) e a milano secondo il database DB-IP.

Eseguendo un ping

```
(base) mellia@MBP-di-Marco ~ % ping -c 4 216.58.198.46
PING 216.58.198.46 (216.58.198.46): 56 data bytes
64 bytes from 216.58.198.46: icmp_seq=0 ttl=55 time=11.512 ms
64 bytes from 216.58.198.46: icmp_seq=1 ttl=55 time=11.544 ms
64 bytes from 216.58.198.46: icmp_seq=2 ttl=55 time=11.830 ms
64 bytes from 216.58.198.46: icmp_seq=3 ttl=55 time=11.868 ms

--- 216.58.198.46 ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 11.512/11.689/11.868/0.161 ms
```

Lo stesso e' a circa 11.5ms di distanza. Può essere negli Stati Uniti? E' compatibile con il tempo di propagazione della luce su una distanza di circa 9500km? Tempo di propagazione = $2 \times (9500\text{km} / 2 \times 10^8\text{m/s}) = 95\text{ms}$ minimo.

- Sempre per capire dove si potrebbe trovare un indirizzo IP, potete usare il comando `tracert`, ed analizzare i nomi dei router e i RTT relativi mostrati lungo il percorso – se disponibili
- Per esempio

```
(base) mellia@MBP-di-Marco ~ % traceroute 216.58.198.46
```

```

traceroute to 216.58.198.46 (216.58.198.46), 64 hops max, 52 byte packets
 10 myfastgate (192.168.1.254)  1.582 ms  0.822 ms  0.719 ms
 2 10.1.3.156 (10.1.3.156)  9.233 ms  13.382 ms  6.957 ms
 3 10.103.251.26 (10.103.251.26)  7.511 ms  6.960 ms  6.833 ms
 4 10.254.20.73 (10.254.20.73)  12.073 ms
   10.254.20.77 (10.254.20.77)  22.802 ms
   10.254.20.73 (10.254.20.73)  11.543 ms
 5 93-63-100-113.ip27.fastwebnet.it (93.63.100.113)  12.317 ms
   93-63-100-61.ip27.fastwebnet.it (93.63.100.61)  11.536 ms
   93-63-100-105.ip27.fastwebnet.it (93.63.100.105)  11.463 ms
 6 62-101-124-29.fastres.net (62.101.124.29)  12.119 ms
   62-101-124-25.fastres.net (62.101.124.25)  13.330 ms  13.326 ms
 7 93.62.86.153 (93.62.86.153)  11.412 ms  12.050 ms  11.205 ms
 8 108.170.245.65 (108.170.245.65)  12.601 ms
   108.170.245.81 (108.170.245.81)  13.234 ms
   108.170.245.65 (108.170.245.65)  13.475 ms
 9 216.239.48.229 (216.239.48.229)  11.842 ms
   216.239.48.231 (216.239.48.231)  12.093 ms
   216.239.48.229 (216.239.48.229)  12.029 ms
10 mil04s04-in-f46.1e100.net (216.58.198.46)  11.707 ms  11.704 ms  11.546 ms

```

Non mostra particolari informazioni utili.

Mentre

```

(base) mellia@MBP-di-Marco ~ % traceroute -w 1 www.purdue.edu
traceroute to www.purdue.edu (128.210.7.200), 64 hops max, 52 byte packets
 1 myfastgate (192.168.1.254)  1.656 ms  0.844 ms  0.727 ms
 2 10.1.3.156 (10.1.3.156)  6.703 ms  6.889 ms  6.833 ms
 3 10.103.251.66 (10.103.251.66)  7.789 ms
   10.103.251.26 (10.103.251.26)  6.846 ms  7.538 ms
 4 10.254.20.77 (10.254.20.77)  12.764 ms
   10.254.20.73 (10.254.20.73)  12.679 ms  13.129 ms
 5 93-63-100-105.ip27.fastwebnet.it (93.63.100.105)  12.394 ms  11.977 ms  11.694 ms
 6 62-101-124-29.fastres.net (62.101.124.29)  13.481 ms
   62-101-124-25.fastres.net (62.101.124.25)  12.371 ms
   62-101-124-29.fastres.net (62.101.124.29)  13.142 ms
 7 ipv4.decix-frankfurt.core1.fra1.he.net (80.81.192.172)  35.011 ms  34.571 ms  34.511 ms
 8 100ge6-1.core1.lon2.he.net (184.105.80.37)  34.776 ms  35.541 ms  34.543 ms
 9 100ge13-2.core1.nyc4.he.net (72.52.92.166)  101.620 ms  102.026 ms  102.625 ms
10 e0-36.core2.nyc4.he.net (184.104.192.242)  100.229 ms  101.884 ms  100.457 ms
11 100ge15-1.core1.cmh1.he.net (184.104.193.77)  116.179 ms  114.607 ms  114.084 ms
12 100ge9-2.core1.ind1.he.net (184.104.193.94)  121.630 ms  121.173 ms  121.275 ms
13      indiana-university-co-indiana-gigapop.10gigabitethernet12-5.core1.ind1.he.net
   (184.105.35.194)  125.653 ms  121.805 ms  122.883 ms
14 38.101.160.251 (38.101.160.251)  123.470 ms  124.882 ms  122.369 ms
15 lamb-20-c7710-01-ptp-po103-891.tcom.purdue.edu (192.5.40.185)  125.106 ms  124.506 ms  124.782
   ms
16 * * *
17 * * *
18 128.210.7.200 (128.210.7.200)  124.191 ms  124.088 ms  124.683 ms

```

Mostra il percorso per raggiungere il server dell'università di Purdue in Indiana, via Francoforte, Londra, New York City, Columbus (cmh – codice dell'aeroporto di Columbus), Indianapolis

- Infine, potete usare Google e altri servizi di ricerca per scoprire informazioni aggiuntive. <https://www.google.com/search?q=216.58.198.46> mostra diversi risultati. Il primo è <https://ipinfo.io/216.58.198.46> che mostra

ipinfo.io

Security Privacy BigData OGR QB PIMCity Poll TNG feedly Repubblica LaStampa.it

ipinfo.io IP or AS number search


ABOUT FEATURES USE CASES PRICING DOCUMENTATION


IP ADDRESS DETAILS

216.58.198.46

Mountain View, California, United States

Location



City: Mountain View
Region: California
Postal Code: 94043
Coordinates: 37.4056,-122.0775
Timezone: America/Los_Angeles
Local Time: May 17, 2020 | 03:18 AM
Country:  United States

Connection

Hostname: mil04s04-in-f14.1e100.net
Address type: IPv4
ASN: [AS15169](#) Google LLC
Organization: Google LLC ([google.com](#))
Route: [216.58.198.0/24](#)




Access all of this data with just one line of code using our API.

Hosted Domain Names

There's a single domain name hosted on this IP address.

[zanzare.io](#)

Network Speed

 154.71 ms	Ping
 54.46 Mbps	Download
 56.93 Mbps	Upload

The average network speed for Google LLC in is shown above. See how your own network speed compares at [speedsmart.net](#).

Anche in questo caso, non tutte le informazioni sono corrette.

Relazione

Organizzare il report in sezioni secondo il seguente schema:

- Sez.1 – Breve descrizione dell'applicazione scelta, che cosa permette di fare, che cosa si intende studiare della stessa – max mezza pagina
- Sez.2 – Descrizione del testbed utilizzato: indicare che tipo di terminali, versione del S.O., connettività di rete, indirizzi IP, ecc. dei terminali usati. Elencare i software e i servizi esterni usati per lo studio – max mezza pagina
- Sez.3 – Descrizione esperimenti e risultati. Per ogni esperimento indicare i risultati ottenuti, dividendo la sezione in sottosezioni, o sezioni multiple. Usare tabelle, immagini, e commentare le informazioni ottenute.
- Sez. 4 – Concludere riassumendo i risultati ottenuti - max mezza pagina.
- In appendice potete mettere altre informazioni aggiuntive non strettamente necessarie ed eventuali script che avete fatto per analizzare i risultati.

Limitare la relazione finale a non più di 8 pagine.