

POLITECNICO DI TORINO

A.A. 2019/2020



# Report 4:

Analisi su applicazione scelta: WhatsApp Web

Student:

Simone Galota - 233727 – Gruppo 16

Teacher:

Prof. Marco Mellia

Course:

Laboratorio di Internet

## Sommario:

<i>1</i>	<i>Descrizione applicazione .....</i>	<i>3</i>
1.1	Aspetti considerati .....	3
1.2	Protocolli utilizzati.....	3
1.3	Modello di rete.....	3
<i>2</i>	<i>Descrizione testbed utilizzato.....</i>	<i>3</i>
2.1	Software utilizzati.....	4
<i>3</i>	<i>Descrizione esperimenti svolti sull'app.....</i>	<i>4</i>
3.1	Test 1: Login .....	4
3.1.1	Analisi dei server .....	5
3.2	Test 2: Invio messaggio di testo.....	6
3.3	Test 3: Invio immagine.....	7
3.4	Test 4: Invio video .....	8
3.4.1	Analisi dei server .....	9
3.5	Test 5: Chiusura forzata browser.....	9
<i>4</i>	<i>Conclusioni.....</i>	<i>10</i>

## 1 Descrizione applicazione

WhatsApp è in assoluto l'applicazione di messaggistica istantanea multiplatforma più famosa al mondo. Proprio per questo motivo non ha bisogno di particolari descrizioni. Per usufruirne basta una scheda sim attiva, una connessione internet e uno smartphone. Nata inizialmente per lo scambio di messaggi di testo, ad oggi possiede svariate funzioni disponibili sia tra singoli utenti che tra gruppi: scambio di contenuti multimediali (audio, video, foto), condivisione live della posizione, videochiamate, chiamate vocali. Dopo l'acquisizione da parte di Facebook Inc. per la cifra astronomica di circa \$ 19,3 miliardi, sono state sviluppate anche le versioni desktop e web (fruibile da qualunque browser, che verrà analizzata in questo breve report). Tali versioni sono accessibili sincronizzando nel PC l'account dello smartphone per mezzo di un codice QR.

### 1.1 Aspetti considerati

In particolare, andremo ad analizzare il comportamento dell'applicazione web nei seguenti casi:

- Login;
- Invio messaggio di testo;
- Invio immagine;
- Invio video;
- Chiusura forzata del browser.

### 1.2 Protocolli utilizzati

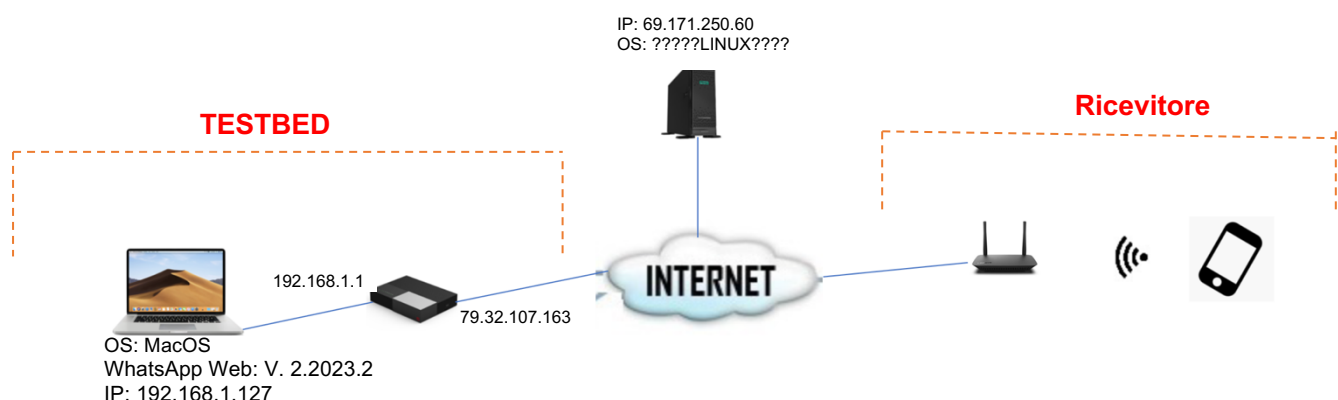
HTTP over TLS versione 1.2 (crittografia end-to-end).

### 1.3 Modello di rete

Per ogni aspetto considerato l'applicazione si comporta seguendo il modello Client-Server (dettagli nella sezione 3). Qualsiasi azione effettuata da un utente, comporta la connessione con un server che inoltra ogni contenuto al destinatario; non c'è mai un contatto diretto tra i dispositivi interessati, anche se collegati alla stessa LAN.

## 2 Descrizione testbed utilizzato

Gli esperimenti sono stati effettuati con la seguente configurazione:



Le prove saranno svolte inviando i contenuti a un ricevitore noto che usa la versione mobile del software. Si ipotizza che ai fini dell'esperimento sia indifferente la configurazione del terminale in ricezione. Per quanto concerne la connessione, il terminale dal quale viene condotta l'analisi è collegato, utilizzando un adattatore USB-C, via Ethernet al modem TIM-Hub che, a sua volta, è connesso ad internet con tecnologia FTTC. Le velocità della connessione, che non sono del tutto rilevanti ai fini del report, sono di 54 Mbps in downlink e di 21 Mbps in uplink.

## 2.1 Software utilizzati

Per l'analisi dell'applicazione è stato utilizzato il software Wireshark per MacOS (Versione 3.2.4) e i tool da sviluppatore presenti in ogni browser (Ispezione elemento). L'accesso al servizio avviene attraverso il browser Google Chrome (Version 83.0.4103.61).

## 3 Descrizione esperimenti svolti sull'app

Per verificare la consistenza dei risultati, l'analisi di ogni caso è stata fatta più volte. Comunque, tutto ciò che è riportato in questa sezione è riferito all'ultimo tentativo, se non diversamente specificato.

### 3.1 Test 1: Login

La prima prova consiste nell'effettuare login inquadrando con la fotocamera dello smartphone il QR-code che compare appena si apre la pagina <https://web.whatsapp.com/>.

Si inizia a catturare non appena viene eseguita la ricerca "WhatsApp web" su Google. Quello che si nota dopo la query DNS (che sfrutta UDP), è l'apertura di 3 connessioni TCP verso la porta 443 (riservata per l'utilizzo sicuro di http – http over TLS) tra l'IP locale 192.168.1.127 e il server di WhatsApp con IP 69.171.250.60. In locale la scelta delle porte su cui ospitare la connessione non è deterministica. Per puro caso sono state utilizzate tre porte consecutive, negli altri tentativi non risulta così.

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
192.168.1.127	50273	69.171.250.60	443	121	80 k	57	5121	64	75 k	4.386702	6.6903	6123	
192.168.1.127	50274	69.171.250.60	443	390	190 k	187	16 k	203	174 k	8.910702	3.6872	35 k	
192.168.1.127	50275	69.171.250.60	443	1,041	935 k	390	29 k	651	905 k	12.521168	0.8472	274 k	

Le connessioni aperte servono, probabilmente, per la scelta degli algoritmi di crittografia, compressione dati e codifica/decodifica da utilizzare. Infatti, non è possibile accedere al contenuto dei pacchetti

3-way handshake

198	4.356891	192.168.1.127	192.168.1.1	DNS	76	Standard query 0xc646 A web.whatsapp.com
215	4.383870	192.168.1.1	192.168.1.127	DNS	129	Standard query response 0xc646 A web.whatsapp.com CNAME mmx-ds.
216	4.386702	192.168.1.127	69.171.250.60	TCP	78	50273 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=67
218	4.418650	69.171.250.60	192.168.1.127	TCP	74	443 → 50273 [SYN, ACK] Seq=0 Ack=1 Win=27760 Len=0 MSS=1400 SAQ
219	4.418742	192.168.1.127	69.171.250.60	TCP	66	50273 → 443 [ACK] Seq=1 Ack=1 Win=131840 Len=0 TSval=674498756
220	4.419164	192.168.1.127	69.171.250.60	TLSv1.3	583	Client Hello
222	4.451104	69.171.250.60	192.168.1.127	TCP	66	443 → 50273 [ACK] Seq=1 Ack=518 Win=28928 Len=0 TSval=126528857
223	4.451105	69.171.250.60	192.168.1.127	TLSv1.3	278	Server Hello, Change Cipher Spec, Application Data
224	4.451203	192.168.1.127	69.171.250.60	TCP	66	50273 → 443 [ACK] Seq=518 Ack=213 Win=131648 Len=0 TSval=674498
225	4.452857	192.168.1.127	69.171.250.60	TLSv1.3	130	Change Cipher Spec, Application Data
228	4.485126	69.171.250.60	192.168.1.127	TLSv1.3	223	Application Data
229	4.485129	69.171.250.60	192.168.1.127	TLSv1.3	140	Application Data
230	4.485425	192.168.1.127	69.171.250.60	TCP	66	50273 → 443 [ACK] Seq=582 Ack=370 Win=131456 Len=0 TSval=674498
231	4.485450	192.168.1.127	69.171.250.60	TCP	66	50273 → 443 [ACK] Seq=582 Ack=444 Win=131392 Len=0 TSval=674498
234	4.502511	192.168.1.127	69.171.250.60	TLSv1.3	158	Application Data
235	4.503026	192.168.1.127	69.171.250.60	TLSv1.3	97	Application Data
236	4.503287	192.168.1.127	69.171.250.60	TLSv1.3	478	Application Data
238	4.534430	69.171.250.60	192.168.1.127	TLSv1.3	97	Application Data

Per verificare quest'ultima affermazione, è stata fatta una cattura del traffico del cellulare al momento dell'inquadratura del QR-code (collegando l'iPhone al wi-fi emesso dal Mac). Sotto è mostrata la conversazione tra telefono e server. Quello che si nota è che tutto il traffico tra i due avviene con TCP, alla porta 5222, senza l'utilizzo di TLS.

Wireshark - Conversations - login\_cell\_stream\_2.pcapng

Ethernet 1 IPv4 1 IPv6 TCP 1 UDP

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
192.168.2.2	60051	69.171.250.61	5222	244	166 k	134	157 k	110	8452	2.303228	1.9293	654 k	

☐ Name resolution ☒ Limit to display filter ☐ Absolute start time

Conversation Types ▼

Help Copy Follow Stream... Graph... Close

Semberebbe essere situato negli USA, ma le varie ricerche effettuate danno almeno tre località troppo distanti tra loro (tutte negli USA). Per cui si procede cercando di fare un minimo di “reverse engineering”.

```
OrgName:      Facebook, Inc.
OrgId:        THEFA-3
Address:      1601 Willow Rd.
City:         Menlo Park
StateProv:    CA
PostalCode:   94025
Country:      US
RegDate:      2004-08-11
Updated:      2012-04-17
Ref:          https://rdap.arin.net/registry/entity/THEFA-3
```



Pingando 69.171.250.60, il RTT è circa 33 ms. Vuole dire che è calcolato su una distanza di circa 4950 km [ $T_p = 2 * (4950\text{km} / 3*10^8\text{m/s}) = 33\text{ ms}$ ].

Dunque, considerando il fatto che tra USA e Sicilia ci sono almeno 8000 km, sembra impossibile che si trovi negli USA. Possiamo comunque ipotizzare che il server si trovi in un'area compresa nel cerchio che ha come centro la Sicilia e un raggio di 4950km (una stima decisamente grossolana, ma è il massimo che si può fare coi dati che si hanno). Considerazioni analoghe possono essere fatte per il server, appartenente alla stessa sottorete, con il quale parla il telefono durante il login (69.171.250.61 - [whatsapp-chatd-edge-shv-01-any2.facebook.com](https://www.whatsapp.com)).

*[L'analisi di altri server continua nella sezione 3.4 – invio video]*

### 3.2 Test 2: Invio messaggio di testo

Questo test è stato svolto in più tentativi, mandando messaggi di testo di dimensioni diverse. Dato che a causa della cifratura la dimensione del pacchetto varia, poiché viene sicuramente aggiunto del padding prima che l'hash del messaggio venga calcolato, non è facile fare un'analisi approfondita guardando la dimensione dei pacchetti e la quantità di bytes scambiati nelle conversazioni. Non è noto di quanto vari la dimensione dopo la cifratura, ma la dimensione della MSS probabilmente è di 1388 bytes.

81	3.868346	192.168.1.127	69.171.250.60	TLSv1.2	375	Application Data
82	3.900444	69.171.250.60	192.168.1.127	TCP	66	443 → 54745 [ACK] Seq=1 Ack=310 Win=388 Len=0 TSv...
85	4.022609	69.171.250.60	192.168.1.127	TLSv1.2	111	Application Data
86	4.022711	192.168.1.127	69.171.250.60	TCP	66	54745 → 443 [ACK] Seq=310 Ack=46 Win=2047 Len=0 T...
95	4.417962	69.171.250.60	192.168.1.127	TLSv1.2	140	Application Data
96	4.418030	192.168.1.127	69.171.250.60	TCP	66	54745 → 443 [ACK] Seq=310 Ack=120 Win=2046 Len=0 ...
168	8.019189	69.171.250.60	192.168.1.127	TLSv1.2	221	Application Data
169	8.019251	192.168.1.127	69.171.250.60	TCP	66	54745 → 443 [ACK] Seq=310 Ack=275 Win=2045 Len=0 ...
170	8.040716	69.171.250.60	192.168.1.127	TLSv1.2	201	Application Data
171	8.040797	192.168.1.127	69.171.250.60	TCP	66	54745 → 443 [ACK] Seq=310 Ack=410 Win=2045 Len=0 ...
264	12.806632	69.171.250.60	192.168.1.127	TLSv1.2	114	Application Data
265	12.806746	192.168.1.127	69.171.250.60	TCP	66	54745 → 443 [ACK] Seq=310 Ack=458 Win=2047 Len=0 ...

Tuttavia, quello che appare dalla cattura è che il messaggio inviato in questo caso (di 120 bytes) sia contenuto nel pacchetto evidenziato in blu (n. 81). È l'unico pacchetto trasmesso dal client al server con TLS, tutti gli altri sono ACK in risposta alle informazioni (PDU n.85, 95, 168, etc.) che il server manda in TLS dopo aver ricevuto il messaggio dal client, ad esempio:

- Ricezione corretta da parte del server (prima spunta nella chat);
- Ricezione avvenuta da parte del destinatario (seconda spunta nella chat);
- Lettura avvenuta da parte del destinatario (spunte blu);
- Presenza online del destinatario o ultimo accesso.

▼ Transport Layer Security
▼ TLSv1.2 Record Layer: Application Data Protocol: http-over-tls
Content Type: Application Data (23)
Version: TLS 1.2 (0x0303)
Length: 304
Encrypted Application Data: 8bacc3742062b2ec41ee4a3cb6ca85d0543d53b5658589c3...

Guardando il pacchetto al livello applicazione, notiamo che il payload cifrato è di 304 bytes così formato: 120 bytes di testo che si vuole mandare + 184 bytes (padding o aggiunti dalla funzione di hash – è poco chiaro). Ciò significa che vi sono 71 bytes di intestazioni (pacchetto n.81 di 375 bytes – 304 di payload al livello 7 = 71 bytes):

- 5 bytes: header TLS (content Type, version, Length);
- 20 bytes: header TCP + 12 bytes di opzioni
- 20 bytes: header IP;

- 14 bytes: header ETH (solo la parte che viene aggiunta prima che il pacchetto sia catturato da wireshark).

### 3.3 Test 3: Invio immagine

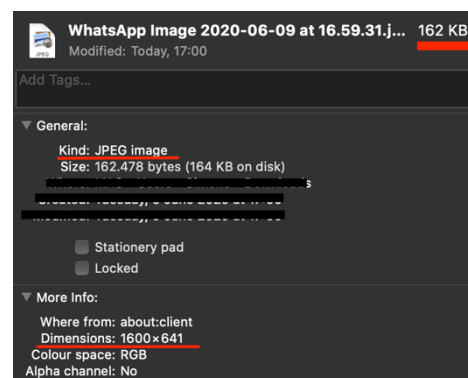
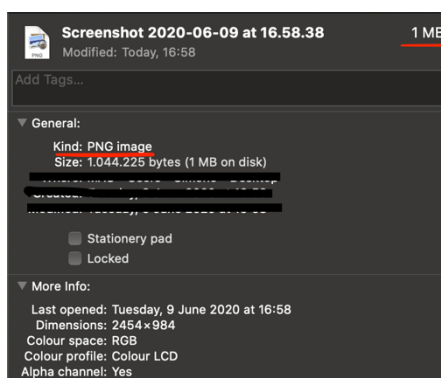
Sono state inviate al destinatario varie immagini in formato PNG di dimensioni comprese tra 20 KB e 1 MB. Le catture di Wireshark fanno riferimento all'invio dell'immagine di 1 MB. Certamente i dati verranno segmentati prima di essere inviati.

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
192.168.1.127	49269	69.171.250.60	443	30	9358	15	3930	15	5428	2.889600	6.4204	4896	4896
192.168.1.127	49327	69.171.250.60	443	201	179 k	131	174 k	70	5496	6.241323	0.6346	2195 k	2195 k

La cattura evidenzia due aspetti principali dell'invio dell'immagine, entrambi svolti dal browser, cioè prima dell'invio:

1. Il file viene compresso prima della trasmissione, infatti lo stream con più dati ha trasmesso solo 179 KB (da A → B 174 KB) a fronte dei 1000 KB inviati. Di conseguenza una foto in alta risoluzione perde qualità se inviata con WhatsApp;
2. Il file, inizialmente PNG, viene codificato in JPEG.

Tali aspetti sono confermati da un'ulteriore prova: dopo la ricezione da parte del server, è stato effettuato, direttamente dalla chat, il download del file inviato.



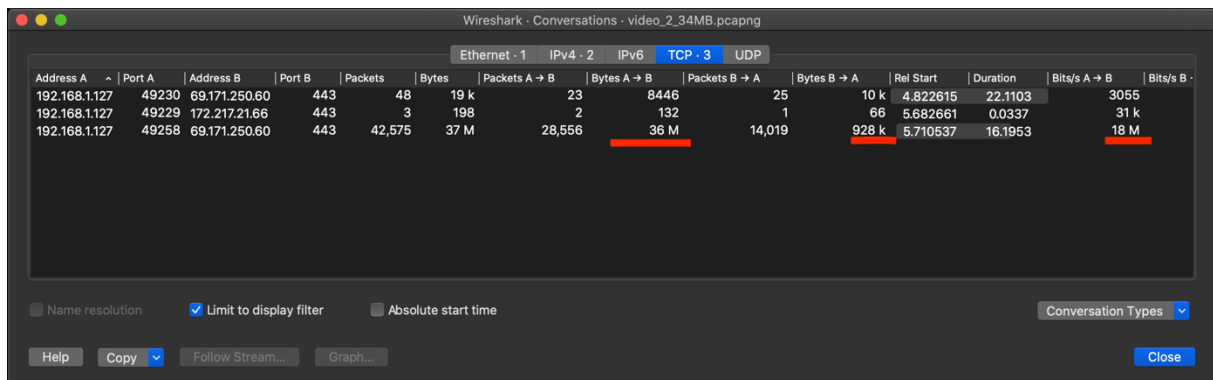
Al solito, per arrivare ai 174 KB indicati dalle statistiche, ai 162 KB di dati effettivi dell'immagine compressa, bisogna aggiungere gli over-head di tutti i segmenti inviati e le varie informazioni di servizio che si scambiano in chat i dispositivi interessati per mezzo del server. Per contro, i dati trasmessi nella direzione opposta comprendono ACK e altre informazioni trasmesse dal server al client, già menzionate nel paragrafo 3.2



### 3.4 Test 4: Invio video

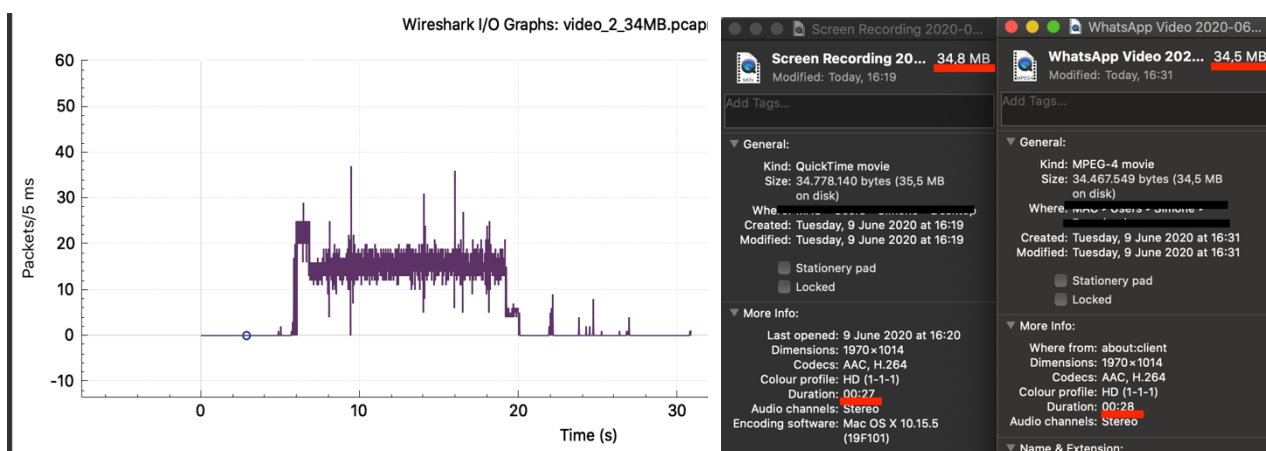
In questo test sono stati inviati video di dimensioni da 1 MB a 34 MB. Da sottolineare che nella versione web la massima dimensione di un allegato è di 64 MB.

Diversamente da quanto succede nel test 3, i video non vengono compressi. Per quanto riguarda la codifica, tutti i video con cui sono stati fatti i test erano del tipo “quick-time movie”, essendo delle registrazioni dello schermo. Per cui, prima dell'invio, il browser effettua una codifica in MP4 (guardare immagine in basso a destra), quindi la dimensione varia di qualche centinaio di KB e, in seguito alla ri-codifica, la durata del video risulta più lunga di un secondo. La quantità di dati in più inviati rispetto alla dimensione del video è dovuta alle intestazioni dei pacchetti.

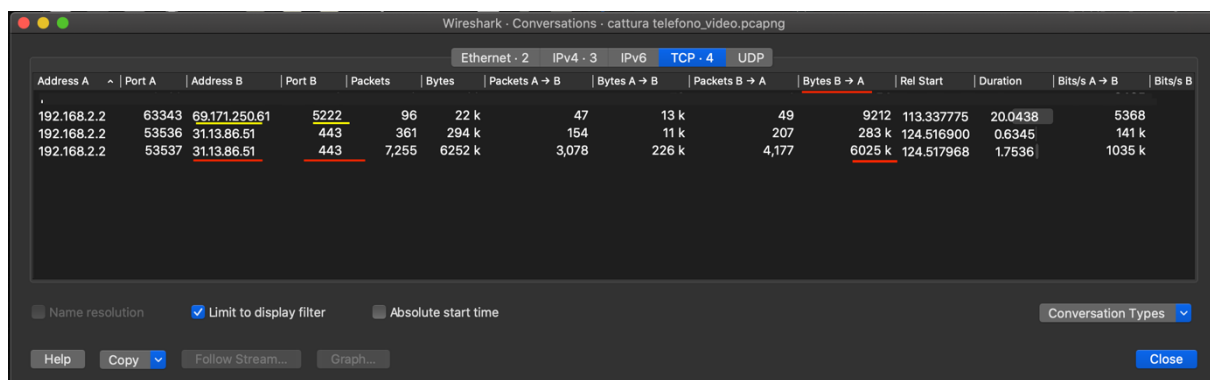


Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
192.168.1.127	49230	69.171.250.60	443	48	19 k	23	8446	25	10 k	4.822615	22.1103	3055	
192.168.1.127	49229	172.217.21.66	443	3	198	2	132	1	66	5.682661	0.0337	31 k	
192.168.1.127	49258	69.171.250.60	443	42,575	37 M	28,556	36 M	14,019	928 k	5.710537	16.1953	18 M	

Il grafico seguente a sinistra mostra il numero di pacchetti inviati nell'unità di tempo (42575 PDU in totale).



È interessante da notare ciò che succede nel telefono collegato all'account che si sta utilizzando dal web. È stata effettuata la cattura del traffico del cellulare (immagine in basso), durante la trasmissione di un video di 6MB dalla versione web.



Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
192.168.2.2	63343	69.171.250.61	5222	96	22 k	47	13 k	49	9212	113.337775	20.0438	5368	
192.168.2.2	53536	31.13.86.51	443	361	294 k	154	11 k	207	283 k	124.516900	0.6345	141 k	
192.168.2.2	53537	31.13.86.51	443	7,255	6252 k	3,078	226 k	4,177	6025 k	124.517968	1.7536	1035 k	



### 3.4.1 Analisi dei server

È evidente che il video inviato compare nella chat del cellulare dopo essere stato trasmesso al cellulare da un server WhatsApp (31.13.86.51 - [media-mxp1-1.cdn.whatsapp.net](https://media-mxp1-1.cdn.whatsapp.net)), diverso da quello del login (69.171.250.61). Il nome del server indicato da wireshark fa pensare che sia un contenitore di file multimediali condivisi su WhatsApp, perché il video non viene inviato al telefono dallo stesso server al quale si è rivolto il browser. Sembra essere situato nella città di Drogheda, nella contea di Louth (Irlanda), ma un altro servizio di informazioni su indirizzi IP lo localizza a Milano (MXP è la sigla dell'aeroporto di Malpensa) dandone anche le coordinate (45.4642, 9.18998); il RTT è mediamente di 34 ms, quindi è probabile che sia comunque in Europa. L'amministratore della sottorete 31.13.64.0 - 31.13.127.255 è, ovviamente, Facebook Inc. Dal lookup col comando "host", il nome del server risulta: [whatsapp-cdn-shv-01-mxp1.fbccdn.net](https://whatsapp-cdn-shv-01-mxp1.fbccdn.net). Oltre a quanto già detto, si è fatto, per errore, un lookup del 31.13.86.52, ed è risultato che fosse parte della "cdn" di Instagram ([instagram-p3-shv-01-mxp1.fbccdn.net](https://instagram-p3-shv-01-mxp1.fbccdn.net)). Quindi, potrei azzardare l'ipotesi che questa sottorete contenga i contenuti multimediali non solo di WhatsApp, ma anche delle altre piattaforme di proprietà Facebook.

### 3.5 Test 5: Chiusura forzata browser

Durante l'utilizzo dell'applicazione web, si è forzata la chiusura del browser con il comando "cmd + Q".

784	13.367985	69.171.250.60	192.168.1.161	TCP	1454	443 → 51019 [ACK] Seq=179351 Ack=399 Win=148 Len=1388	1
785	13.367986	69.171.250.60	192.168.1.161	TCP	1454	443 → 51019 [ACK] Seq=180739 Ack=399 Win=148 Len=1388	1
786	13.367987	69.171.250.60	192.168.1.161	TCP	1454	443 → 51019 [ACK] Seq=182127 Ack=399 Win=148 Len=1388	1
787	13.368107	192.168.1.161	69.171.250.60	TCP	66	51019 → 443 [ACK] Seq=399 Ack=179351 Win=3839 Len=0 TSv	
788	13.368107	192.168.1.161	69.171.250.60	TCP	66	51019 → 443 [ACK] Seq=399 Ack=182127 Win=3796 Len=0 TSv	
789	13.368763	192.168.1.161	69.171.250.60	TCP	66	51019 → 443 [ACK] Seq=399 Ack=183515 Win=3883 Len=0 TSv	
790	13.369439	69.171.250.60	192.168.1.161	TCP	1454	443 → 51019 [ACK] Seq=183515 Ack=399 Win=148 Len=1388	1
791	13.369445	69.171.250.60	192.168.1.161	TCP	1454	443 → 51019 [ACK] Seq=184903 Ack=399 Win=148 Len=1388	1
792	13.369447	69.171.250.60	192.168.1.161	TCP	1454	443 → 51019 [ACK] Seq=186291 Ack=399 Win=148 Len=1388	1
793	13.369448	69.171.250.60	192.168.1.161	TCP	1454	443 → 51019 [ACK] Seq=187679 Ack=399 Win=148 Len=1388	1
794	13.369534	192.168.1.161	69.171.250.60	TCP	66	51019 → 443 [ACK] Seq=399 Ack=186291 Win=3839 Len=0 TSv	
795	13.369534	192.168.1.161	69.171.250.60	TCP	66	51019 → 443 [ACK] Seq=399 Ack=189067 Win=3796 Len=0 TSv	
796	13.369627	192.168.1.161	69.171.250.60	TCP	66	[TCP Window Update] 51019 → 443 [ACK] Seq=399 Ack=189067	
797	13.370736	69.171.250.60	192.168.1.161	TLSv1.2	1454	Application Data [TCP segment of a reassembled PDU]	
798	13.370742	69.171.250.60	192.168.1.161	TLSv1.2	1345	Application Data, Application Data	
799	13.370835	192.168.1.161	69.171.250.60	TCP	66	51019 → 443 [ACK] Seq=399 Ack=191734 Win=3841 Len=0 TSv	
853	22.816853	192.168.1.161	69.171.250.60	TLSv1.3	96	Application Data	
854	22.817813	192.168.1.161	69.171.250.60	TCP	66	51024 → 443 [FIN, ACK] Seq=4304 Ack=199637 Win=131072 Len=0	
855	22.854966	69.171.250.60	192.168.1.161	TCP	66	443 → 51024 [FIN, ACK] Seq=199637 Ack=4305 Win=32768 Len=0	
856	22.855068	192.168.1.161	69.171.250.60	TCP	66	51024 → 443 [ACK] Seq=4305 Ack=199638 Win=131072 Len=0	
857	23.031808	192.168.1.161	69.171.250.60	TCP	66	51019 → 443 [FIN, ACK] Seq=399 Ack=191734 Win=3883 Len=0	
867	23.067076	69.171.250.60	192.168.1.161	TCP	66	443 → 51019 [FIN, ACK] Seq=191734 Ack=400 Win=148 Len=0	
868	23.067397	192.168.1.161	69.171.250.60	TCP	66	51019 → 443 [ACK] Seq=400 Ack=191735 Win=3883 Len=0 TSv	

Dalla traccia di wireshark è possibile vedere come questa chiusura forzata sia interpretata con una richiesta di chiusura di connessione TCP. Quindi viene attuato un doppio 3-way handshake di chiusura connessione (utilizzando il flag FIN anziché SYN), uno per la porta 51024, uno per la porta 51019. Significa che ne erano state aperte due anziché tre.

## 4 Conclusioni

Nonostante molte delle considerazioni fatte in questo report siano supposizioni, e sebbene l'algoritmo di cifratura, come è ovvio che sia, impedisca una completa comprensione dell'applicazione, si è cercato di capire in linee generali il funzionamento delle principali operazioni che offre il servizio. Anche se, per ragioni di brevità, non sono state analizzate tutte le funzioni possibili e tutte le casistiche (condivisione posizione, caricamento documento, videochiamata, chat di gruppo...).

Un dato importante che emerso dai test eseguiti, e che vale la pena di sottolineare, è che la ritrasmissione (l'inoltro) di contenuti già presenti nelle chat non richiede la stessa procedura del primo invio. Dalla cattura è evidente che i bytes trasmessi sono in quantità molto minore rispetto alla dimensione del contenuto (anche se compresso). È ipotizzabile che WhatsApp, per questioni di efficienza (...e non solo...), abbia già quel contenuto cifrato salvato nel proprio server e pronto da inoltrare al destinatario, con le dovute informazioni per decifrarlo. C'è da dire che la crittografia end-to-end permette solo alle parti coinvolte nella conversazione di decifrare i messaggi.