



Università degli Studi di Verona

FACOLTÀ DI SCIENZE E INGEGNERIA
Corso di Laurea in Informatica

Whatsapp Messenger

Candidato:

Alberto Marini

Matricola VR359129

Relatore:

Prof. Damiano Carra

Indice

| | | |
|----------|------------------------------|-----------|
| 1 | Introduzione | 5 |
| 2 | Strumenti utilizzati | 7 |
| 2.1 | Whatsapp Messenger | 7 |
| 2.1.1 | Cos'è | 7 |
| 2.1.2 | Come funziona | 8 |
| 2.2 | Wireshark | 9 |
| 2.2.1 | Cos'è | 9 |
| 2.3 | Whois | 10 |
| 2.4 | Cloud Monitor | 11 |
| 3 | Misurazioni | 13 |
| 4 | Conclusioni | 15 |

Capitolo 1

Introduzione

Capitolo 2

Strumenti utilizzati

2.1 Whatsapp Messenger

2.1.1 Cos'è

Whatsapp Messenger è un'applicazione di messaggistica mobile multi piattaforma che consente di scambiarsi messaggi coi propri contatti senza dover pagare gli SMS. WhatsApp Messenger è disponibile per iPhone, BlackBerry, Android, Windows Phone e Nokia. Tutti questi telefoni possono scambiarsi messaggi gli uni gli altri. Dato che WhatsApp Messenger si serve dello stesso piano dati Internet usato per le e-mail e la navigazione web, non vi sono costi aggiuntivi per mandare messaggi e restare in contatto coi propri amici ed è sicuramente questo uno dei motivi per i quali questa applicazione ha ottenuto un così gran successo in poco tempo.

Oltre alla messaggistica di base gli utenti di WhatsApp possono creare gruppi, scambiarsi messaggi illimitati, video e messaggi audio multimediali.

L'11 aprile 2014 è arrivato il via libera all'acquisizione di WhatsApp da parte di Facebook dalla Federal Trade Commission (Ftc), l'ente governativo americano per la protezione dei consumatori.

2.1.2 Come funziona

Servendosi della rete cellulare, Whatsapp messenger permette di inviare messaggi a qualsiasi altro utente connesso ad una rete. Questa modalità di funzionamento ci permette di affermare che, sicuramente, all'invio di un messaggio viene contattato un server il quale avrà il compito di smistarlo al dispositivo di destinazione.

In particolare, all'invio di un messaggio vengono effettuate le seguenti operazioni:

- Il messaggio arriva ad un server
- Il server comunica al mittente l'avvenuta ricezione del messaggio
- Il server inoltra il messaggio al destinatario
- Il destinatario comunica al server l'avvenuta ricezione del messaggio
- Il server comunica al mittente l'avvenuta ricezione del messaggio da parte del destinatario

Quando il server riceve il messaggio, nel dispositivo mittente compare una spunta; quando il destinatario riceve il messaggio al mittente compare la seconda spunta.

La Figura 2.1 illustra tale funzionamento.

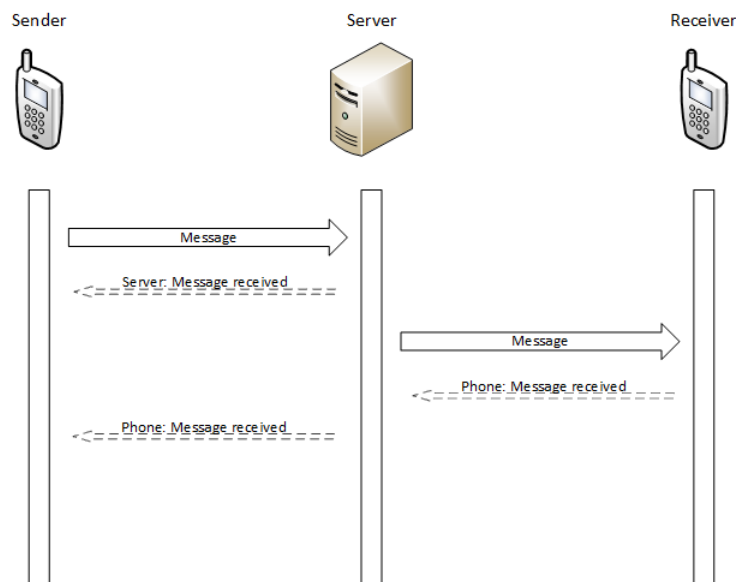


Figura 2.1: Funzionamento Whatsapp

2.2 Wireshark

2.2.1 Cos'è

Wireshark è un analizzatore di rete. Consente di catturare direttamente i dati da una rete attiva oppure di analizzare file contenenti pacchetti precedentemente ottenuti. Inizialmente, il formato dei file catturati da Wireshark era il formato libpcap, che è il formato usato da tcpdump ed altri tools.

I pacchetti catturati sono conformi alla libreria pcap. È possibile applicare filtri ai pacchetti ottenuti, selezionando, per esempio, solo quelli provenienti da un determinato IP sorgente. I filtri applicabili ai pacchetti seguono le regole della libreria pcap.

L'interfaccia grafica di Wireshark (Figura 2.2) mostra il numero di pacchetti catturati, il tempo trascorso tra la cattura dei pacchetti, l'indirizzo sorgente e quello di destinazione, il protocollo usato, la lunghezza del pacchetto ed altre informazioni.

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|----------|-----------------|-----------------|----------|--------|------------------------------------|
| 1 | 0.000000 | 127.0.0.1 | 127.0.0.1 | TCP | 72 | cbt > 47640 [PSH, ACK] Seq=1 Ack=1 |
| 2 | 0.000023 | 127.0.0.1 | 127.0.0.1 | TCP | 68 | 47640 > cbt [ACK] Seq=1 Ack=5 win= |
| 3 | 0.000280 | 127.0.0.1 | 127.0.0.1 | TCP | 84 | cbt > 47640 [PSH, ACK] Seq=5 Ack=1 |
| 4 | 0.000343 | 127.0.0.1 | 127.0.0.1 | TCP | 68 | 47640 > cbt [ACK] Seq=1 Ack=21 win |
| 5 | 0.000631 | 127.0.0.1 | 127.0.0.1 | TCP | 72 | cbt > 47641 [PSH, ACK] Seq=1 Ack=1 |
| 6 | 0.000711 | 127.0.0.1 | 127.0.0.1 | TCP | 68 | 47641 > cbt [ACK] Seq=1 Ack=5 win= |
| 7 | 0.000945 | 127.0.0.1 | 127.0.0.1 | TCP | 84 | cbt > 47641 [PSH, ACK] Seq=5 Ack=1 |
| 8 | 0.000998 | 127.0.0.1 | 127.0.0.1 | TCP | 68 | 47641 > cbt [ACK] Seq=1 Ack=21 win |
| 9 | 0.309231 | 100.77.14.234 | 83.224.70.93 | DNS | 78 | Standard query 0xaf5c A e10.whats |
| 10 | 1.111483 | 100.77.14.234 | 173.194.70.188 | TCP | 76 | 47697 > hpvroom [SYN] Seq=0 win=52 |
| 11 | 2.236683 | 83.224.70.93 | 100.77.14.234 | DNS | 206 | Standard query response 0xaf5c A |
| 12 | 2.283933 | 100.77.14.234 | 184.173.161.163 | TCP | 76 | 41265 > https [SYN] Seq=0 win=5240 |
| 13 | 2.296618 | 173.194.70.188 | 100.77.14.234 | TCP | 76 | hpvroom > 47697 [SYN, ACK] Seq=0 A |
| 14 | 2.296905 | 100.77.14.234 | 173.194.70.188 | TCP | 68 | 47697 > hpvroom [ACK] Seq=1 Ack=1 |
| 15 | 2.477461 | 100.77.14.234 | 173.194.70.188 | TCP | 148 | 47697 > hpvroom [PSH, ACK] Seq=1 A |
| 16 | 2.536633 | 184.173.161.163 | 100.77.14.234 | TCP | 76 | https > 41265 [SYN, ACK] Seq=0 Ack |
| 17 | 2.537055 | 100.77.14.234 | 184.173.161.163 | TCP | 68 | 41265 > https [ACK] Seq=1 Ack=1 wi |
| 18 | 2.591803 | 100.77.14.234 | 184.173.161.163 | SSL | 189 | Continuation Data |
| 19 | 2.689238 | 173.194.70.188 | 100.77.14.234 | TCP | 68 | hpvroom > 47697 [ACK] Seq=1 Ack=81 |
| 20 | 2.757165 | 173.194.70.188 | 100.77.14.234 | TCP | 1366 | hpvroom > 47697 [ACK] Seq=1 Ack=81 |
| 21 | 2.757305 | 100.77.14.234 | 173.194.70.188 | TCP | 68 | 47697 > hpvroom [ACK] Seq=81 Ack=1 |
| 22 | 2.757683 | 173.194.70.188 | 100.77.14.234 | TCP | 1366 | hpvroom > 47697 [ACK] Seq=1299 Ack |
| 23 | 2.758106 | 100.77.14.234 | 173.194.70.188 | TCP | 68 | 47697 > hpvroom [ACK] Seq=81 Ack=2 |

Frame 26: 1366 bytes on wire (10928 bits), 1366 bytes captured (10928 bits)

Linux cooked capture

Internet Protocol Version 4, Src: 173.194.70.188 (173.194.70.188), Dst: 100.77.14.234 (100.77.14.234)

Version: 4

Header length: 20 bytes

Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))

Total Length: 1350

Identification: 0xf405 (62469)

```

0000  00 00 02 00 00 00 00 00 00 00 00 00 08 00  .....
0010  45 00 05 46 f4 05 00 00 28 06 31 f7 ad c2 46 bc  E..F....(.I...F.
0020  64 4d 0e ea 14 6c ba 51 06 e1 59 f2 d2 b7 1a be  dM...l.Q..Y....
0030  80 10 02 99 87 91 00 00 01 01 08 0a ad b0 63 72  ....cr
0040  00 00 97 43 2f 63 72 6c 2e 67 65 6f 74 72 75 73  ...C/cr1.geotrus
0050  74 7a 62 6f 6d 7f 62 77 6e 73 7f 67 74 67 6e 6f  t.cam/cr1.geotrus

```

Figura 2.2: Interfaccia di Wireshark

2.3 Whois

Whois é un servizio utilizzabile da shell di Ubuntu che permette di visualizzare informazioni riguardanti un determinato indirizzo IP. In particolare, applicando whois ad un indirizzo IP, vengono visualizzati il nome della rete, il range di indirizzi ai quali la rete fa riferimento, il luogo in cui si situa l'IP ricercato, l'organizzazione che lo gestisce ed altre informazioni di rete.

```
inetnum:      157.27.0.0 - 157.27.255.255
netname:      IVRUNIV-NET
org:          ORG-UDSD45-RIPE
descr:        Università degli Studi di Verona
country:      IT
admin-c:      GB6434-RIPE
tech-c:       AS9924-RIPE
status:       LEGACY
remarks:       For information on "status:" attribute read https://www.ripe.net/
               tus-values-legacy-resources
remarks:       This prefix is statically assigned
remarks:       To notify abuse mailto: cert@garr.it
remarks:       Centro di Informatica e Calcolo Automatico
remarks:       Università' di Verona
remarks:       GARR - Italian academic and research network
mnt-irt:      IRT-GARR-CERT
mnt-by:       GARR-LIR
source:       RIPE # Filtered

organisation: ORG-UDSD45-RIPE
org-name:     Università' degli Studi di Verona
org-type:     OTHER
address:      Via S.Francesco, 22
address:      I - 37129 Verona (VR)
phone:        +39 045 8028713
fax-no:       +39 045 8028471
mnt-ref:      GARR-LIR
mnt-by:       GARR-LIR
abuse-c:      AG16225-RIPE
source:       RIPE # Filtered
```

Figura 2.3: Esempio Whois

2.4 Cloud Monitor

Cloud Monitor é un'azienda leader nel settore del monitoraggio delle prestazioni di siti ed applicazioni Web. Verifica le prestazioni di siti e server grazie a 95 stazioni di monitoraggio disposte in 48 paesi del mondo. Dato un indirizzo IP o un sito web, effettua, attraverso le 95 stazioni, ping verso quell'indirizzo registrando l'esito dello stesso e, in caso di ping eseguito con successo, RTT minimo, RTT medio ed RTT massimo (RTT - Round Trip Time, tempo impiegato da un pacchetto di dimensione trascurabile per viaggiare da un computer ad un altro e tornare indietro).

| Esegui il ping su: www.google.com | | | | | |
|---|--------------------------|------------|-----------|-------------|--------------------------|
| Punto di controllo | Risultato | RTT minimo | RTT medio | RTT massimo | IP |
| Arabia Saudita - Riyadh (saruh01) | Unknown result from ping | | | | 2a00:1450:4009:808::1011 |
| Argentina - Buenos Aires (arbue01) | Unknown result from ping | | | | 2800:3f0:4002:800::1014 |
| Australia - Brisbane (aubne01) | Okay | 19.3 | 19.4 | 19.6 | 2404:6800:4006:804::1014 |
| Australia - Melbourne (aumel02) | Unknown result from ping | | | | 2404:6800:4006:803::1013 |
| Australia - Perth (auper01) | Unknown result from ping | | | | 2404:6800:4006:806::1012 |
| Australia - Sydney (ausyd02) | Packets lost (100%) | | | | 2404:6800:4006:803::1011 |
| Austria - Vienna (atvie01) | Unknown result from ping | | | | 2a00:1450:4001:80e::1010 |
| Belgio - Anversa (beanr02) | Unknown result from ping | | | | 2a00:1450:4005:809::1012 |
| Brasile - Porto Alegre (brpoa01) | Unknown result from ping | | | | 2607:f8b0:4008:800::1013 |
| Brasile - Rio de Janeiro (brrio01) | Unknown result from ping | | | | 2800:3f0:4004:800::1014 |
| Brasile - San Paolo (brsao03) | Okay | 139.9 | 142.4 | 143.5 | 2607:f8b0:4000:807::1012 |
| Bulgaria - Sofia (bgsof01) | Unknown result from ping | | | | 2a00:1450:4001:c02::67 |
| Canada - Calgary (cacal01) | Unknown result from ping | | | | 2607:f8b0:400a:803::1014 |
| Canada - Montreal (camtr01) | Okay | 26.1 | 27.5 | 30.5 | 2607:f8b0:4009:806::1011 |
| Canada - Toronto (cator01) | Packets lost (100%) | | | | 2607:f8b0:400b:806::1012 |
| Canada - Vancouver (cavan02) | Okay | 25.7 | 25.8 | 25.9 | 2001:4860:400b:c01::68 |
| Cina - Hong Kong (hkhkg01) | Okay | 4.5 | 4.9 | 5.3 | 2404:6800:4005:806::1013 |

Figura 2.4: Esempio Cloud Monitor

Capitolo 3

Misurazioni

Misurazioni + grafici

Capitolo 4

Conclusioni