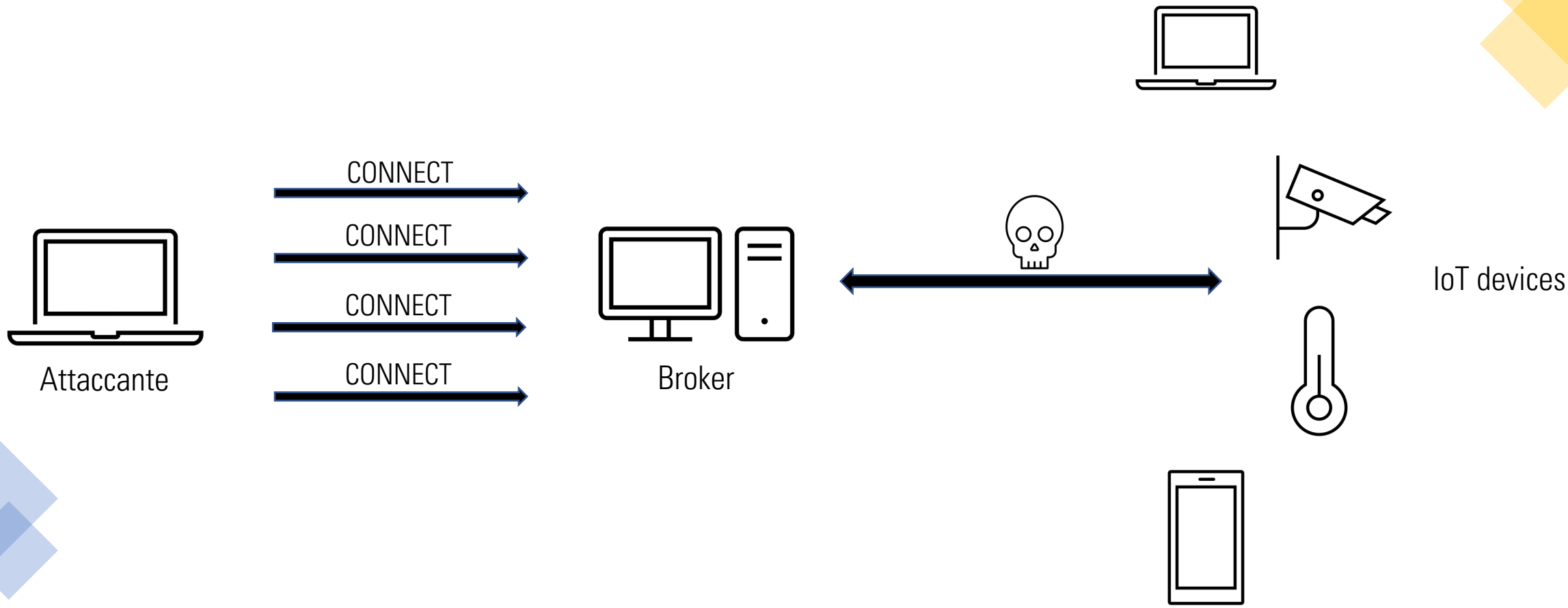
Sfruttamento delle vulnerabilità del protocollo MQTT per un attacco DoS slow rate su dispositivi IoT

Gianluca Boschi

Paola Petri

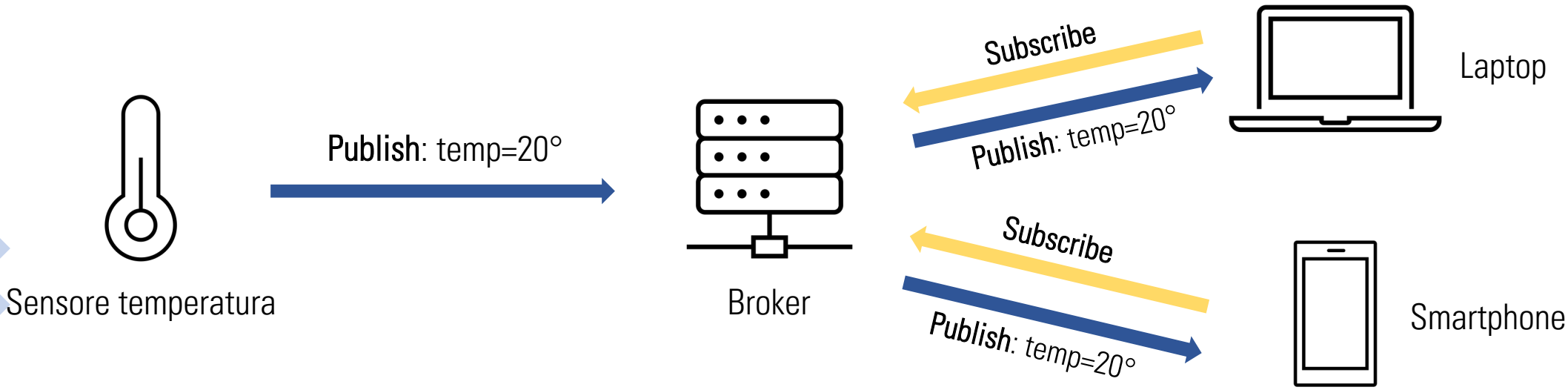
Francesco Carli

Caso d'uso: DoS su protocollo MQTT



MQTT: Message Queuing Telemetry Transport

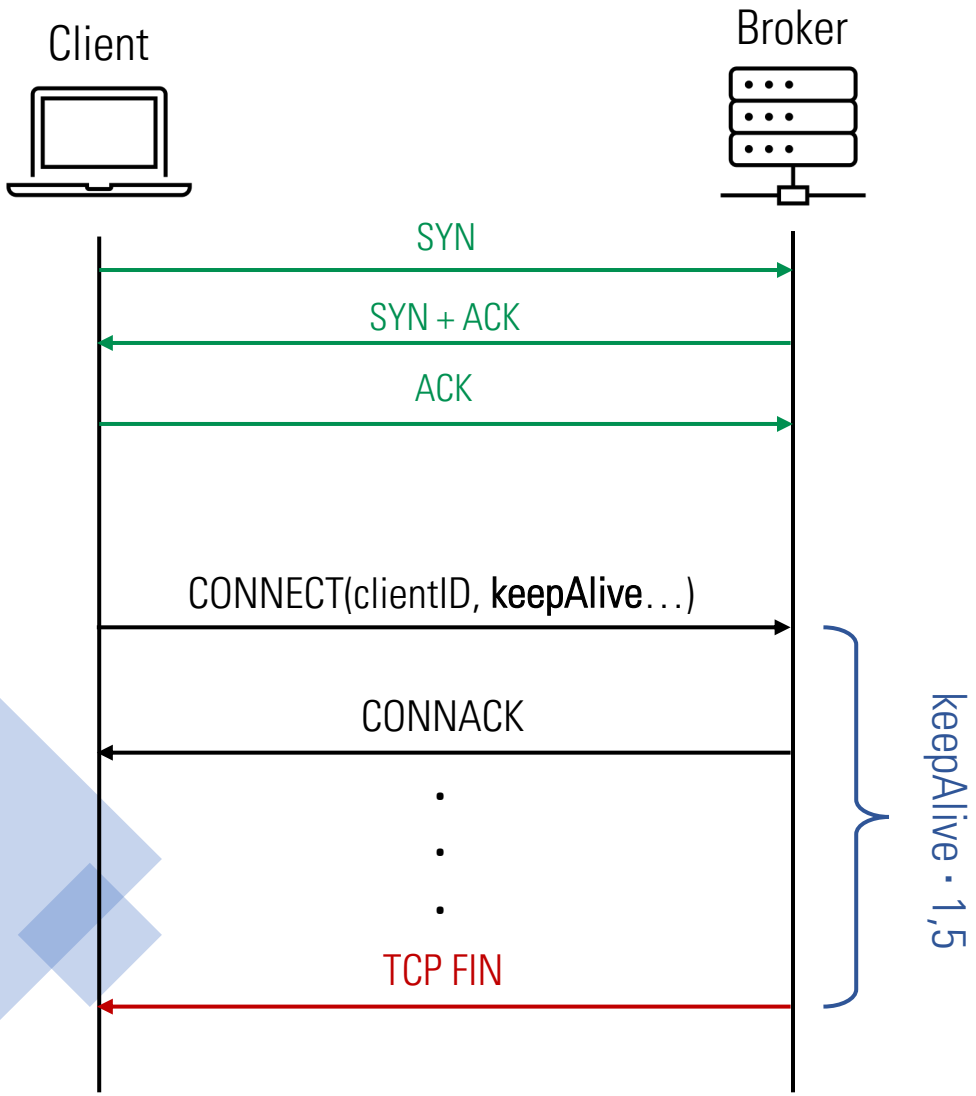
- Protocollo applicativo di tipo **publish-subscribe**, basato su un nodo centrale chiamato **broker**
- Standard per le comunicazioni in ambienti IoT
- Usato all'interno di reti wireless con poca banda disponibile o connessioni non affidabili



SlowITe: Slow DoS against IoT environments

- Attacco slow DoS che ha come target applicazioni basate su protocollo TCP
- Utilizza una **piccola quantità di banda** per mantenere le comunicazioni attive il più a lungo possibile
- Crea un **alto numero di connessioni** con il broker al fine di saturarlo
- Sfruttamento del parametro **keepAlive**

Parametro keepAlive



- Parametro del pacchetto CONNECT
- Campo a 16 bit (valore massimo 65535) che esprime la **durata massima di una connessione**
- Il Broker mantiene la connessione attiva per al massimo **1,5 · KeepAlive**
- Il valore di default è **60 secondi**

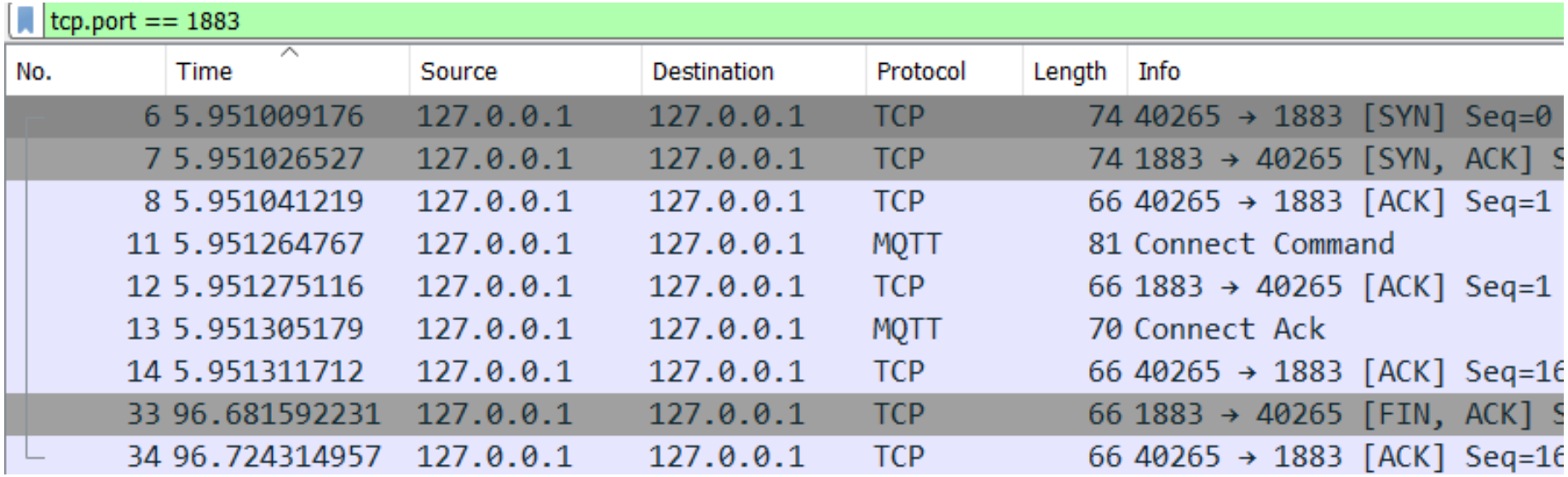
Implementazione dell’attacco

Script utilizzati

- **MQTT_SlowDoS.py** → esegue l'attacco SlowITe contro il broker MQTT Mosquitto
- **MQTT_keepAlive.py** → testa la durata della connessione in base al parametro keepAlive
- **client_pub.py** → publisher che invia messaggi su un topic verso il broker MQTT
- **client_sub.py** → subscriber che crea un topic e ci si sottoscrive per ricevere messaggi

Test durata connessione

1. Esecuzione del broker MQTT Mosquitto
2. Avvio monitoraggio della rete tramite Wireshark
3. Esecuzione dello script MQTT_keepAlive.py
4. Interruzione monitoraggio e analisi del pcap di Wireshark



Durata sessione MQTT

	Tempo
$SYN_{1°} \rightarrow$ CONNECT	0,000255591 sec
CONNECT → CONNACK	0,000040412 sec
CONNACK → FIN	90,73028705 sec
CONNACK → ACK_{FIN}	90,77300970 sec
$SYN_{1°} \rightarrow ACK_{FIN}$	90,77330577 sec

Banda utilizzata da una singola sessione

Dimensione totale comunicazione	629 byte
Durata della comunicazione	90,77330577 sec
Banda utilizzata	55,435 bps

Test attacco SlowITe

1) Esecuzione del broker MQTT Mosquitto

2) Avvio monitoraggio della rete tramite Wireshark

3) Esecuzione dello script MQTT_SlowDoS.py

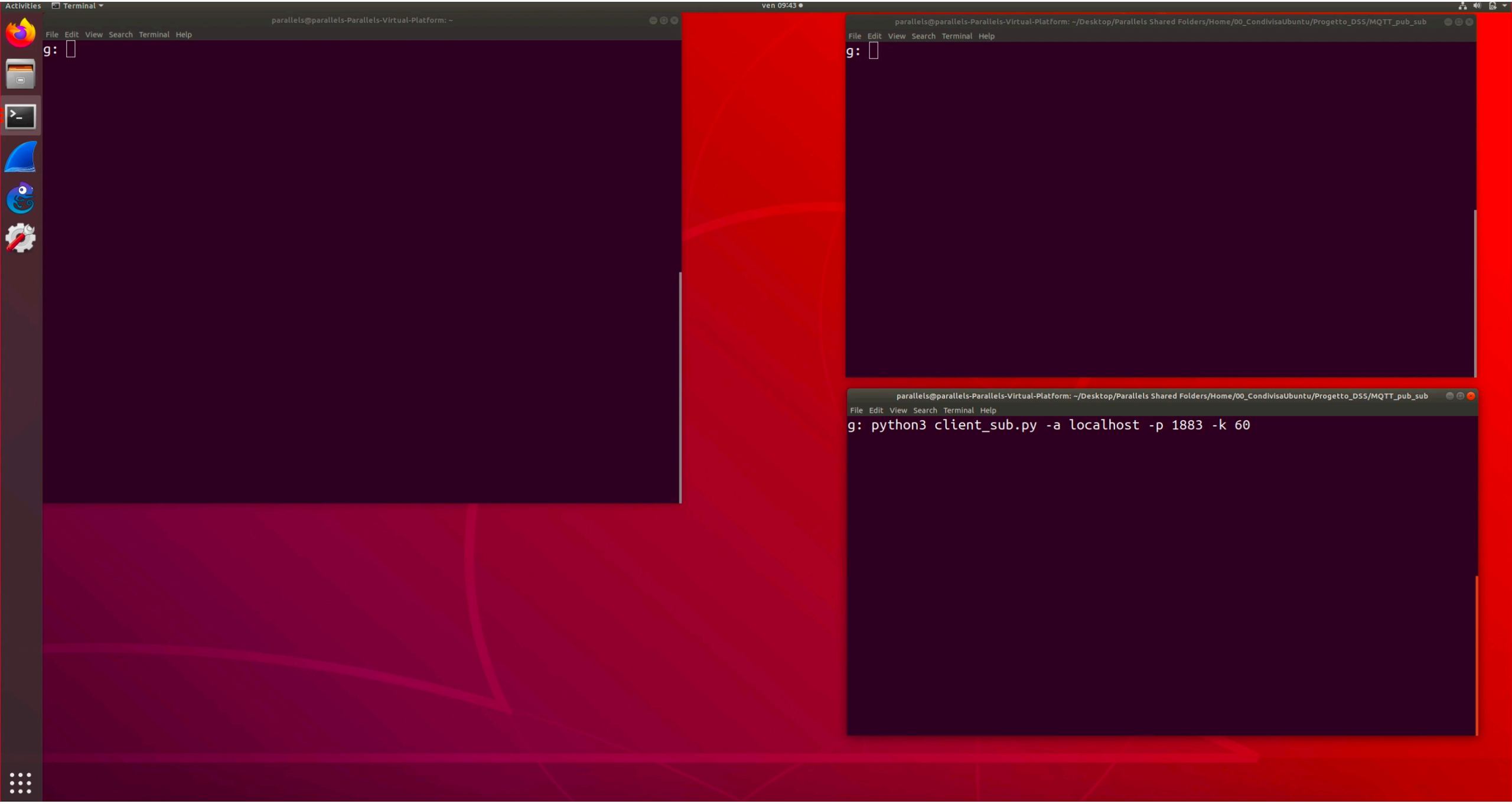
4) Esecuzione di client_pub.py e client_sub.py per testare lo stato di DoS sul broker

5) Interruzione dell'attacco

6) Esecuzione di client_pub.py e client_sub.py per testare la disponibilità del broker

tcp.port==1883 and mqtt

Source	Destination	Protocol	Source Port TCP	Destination Port TCP	Info
127.0.0.1	127.0.0.1	MQTT	1883	56275	Connect Ack
127.0.0.1	127.0.0.1	MQTT	50323	1883	Connect Command
127.0.0.1	127.0.0.1	MQTT	1883	50323	Connect Ack
127.0.0.1	127.0.0.1	MQTT	38169	1883	Connect Command
127.0.0.1	127.0.0.1	MQTT	1883	38169	Connect Ack
127.0.0.1	127.0.0.1	MQTT	59461	1883	Connect Command
127.0.0.1	127.0.0.1	MQTT	1883	59461	Connect Ack
127.0.0.1	127.0.0.1	MQTT	43739	1883	Connect Command
127.0.0.1	127.0.0.1	MQTT	1883	43739	Connect Ack
127.0.0.1	127.0.0.1	MQTT	42221	1883	Connect Command
127.0.0.1	127.0.0.1	MQTT	1883	42221	Connect Ack
127.0.0.1	127.0.0.1	MQTT	52953	1883	Connect Command
127.0.0.1	127.0.0.1	MQTT	1883	52953	Connect Ack
127.0.0.1	127.0.0.1	MQTT	37055	1883	Connect Command
127.0.0.1	127.0.0.1	MQTT	56911	1883	Connect Command
127.0.0.1	127.0.0.1	MQTT	41239	1883	Connect Command
127.0.0.1	127.0.0.1	MQTT	44741	1883	Connect Command
127.0.0.1	127.0.0.1	MQTT	50241	1883	Connect Command
127.0.0.1	127.0.0.1	MQTT	58171	1883	Connect Command
127.0.0.1	127.0.0.1	MQTT	48037	1883	Connect Command
127.0.0.1	127.0.0.1	MQTT	59663	1883	Connect Command


Activities
Terminal
ven 09:43
parallels@parallels-Parallels-Virtual-Platform: ~
File Edit View Search Terminal Help
g:
parallels@parallels-Parallels-Virtual-Platform: ~/Desktop/Parallels Shared Folders/Home/00_CondivisaUbuntu/Progetto_DSS/MQTT_pub_sub
File Edit View Search Terminal Help
g:
parallels@parallels-Parallels-Virtual-Platform: ~/Desktop/Parallels Shared Folders/Home/00_CondivisaUbuntu/Progetto_DSS/MQTT_pub_sub
File Edit View Search Terminal Help
g: python3 client_sub.py -a localhost -p 1883 -k 60

Connessioni gestibili dal broker

CONNECT	CONNACK	Connessioni non accettate
1024	1015	9

Banda utilizzata dall'attacco SlowITe

Tempo di arrivo del $SYN_{1°}$	4,625 sec
Tempo di arrivo del ACK_{FIN}	97,599 sec
Durata totale	92,974 sec
Dimensione totale della comunicazione	649 918 byte
Banda utilizzata	55 927,98 bps

Conclusioni

- Tramite un approccio slow rate, SlowITe sfrutta una specifica vulnerabilità di MQTT che rende l'attaccante in grado di impostare il parametro keepAlive su un valore arbitrario.

- I risultati ottenuti dai test riportati in precedenza sono in linea con i dati illustrati in [1]

- Il codice sorgente ed un breve documento per l'installazione e l'esecuzione dei vari script è disponibile al seguente link GitHub: **github.com/gianluca2414/MQTT_SlowITe**

[1] Vaccari, Ivan, Maurizio Aiello, and Enrico Cambiaso. "SlowITe, a Novel Denial of Service Attack Affecting MQTT." Sensors20.10 (2020): 2932.

GRAZIE DELL' ATTENZIONE