

MQTTSA:

A Tool for Automatically Assisting the Secure Deployments of MQTT brokers

■ ■ ■











**Andrea Palmieri⁺, Paolo Prem⁺, Silvio Ranise^{*},
Umberto Morelli^{*}, and Tahir Ahmad^{*}**

⁺EIT Digital (UniTN and EURECOM)

^{*}Fondazione Bruno Kessler (FBK)

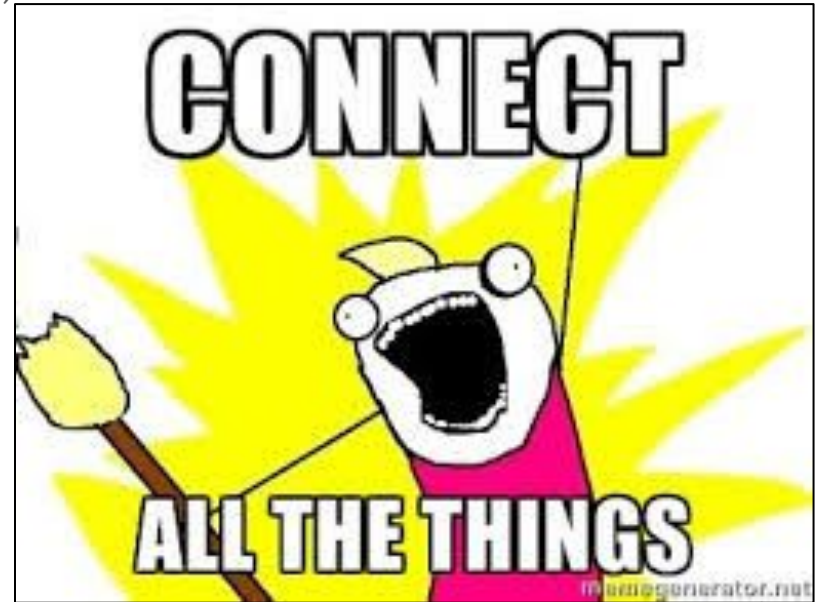


Agenda

-  **MQTT**
 -  Overview
 -  Examples
 -  Security
-  **MQTTSA**
 -  Motivation
 -  Modes
 -  Evaluation
-  **Demo (with Paolo)**
-  **Future work**

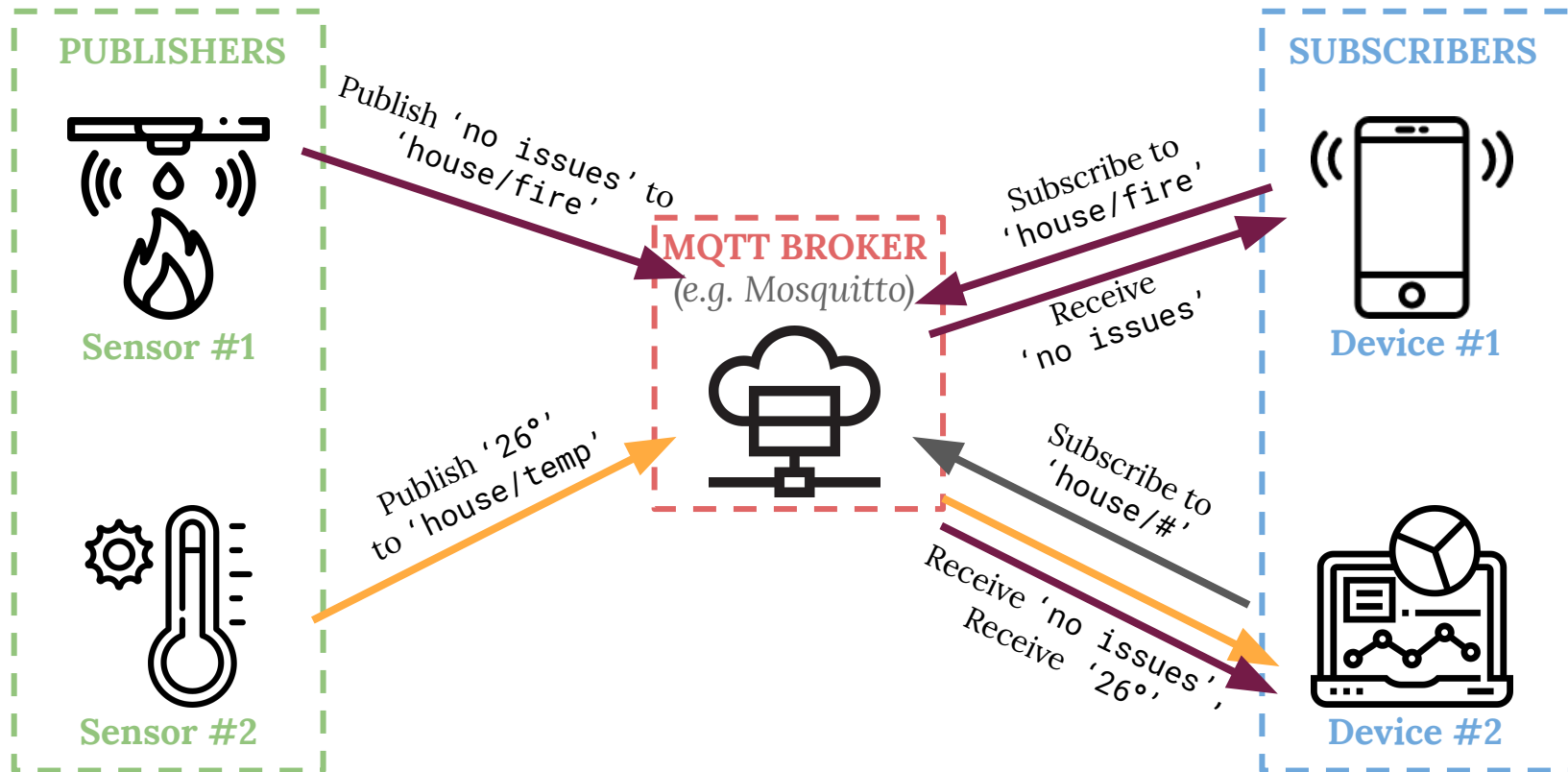
MQTT

- “Message Queuing Telemetry Transport”
- Invented in **1999** by Stanford-Clark and Nipper
- Lightweight messaging protocol (**M2M**)
 - publish/subscribe
 - extremely simple
- Designed for:
 - constrained devices
 - low-bandwidth and high-latency
 - unreliable networks
- v5.0 and v3.1.1 are **OASIS standards**
- *de facto* standard protocol to send messages in **IoT**





MQTT: an example

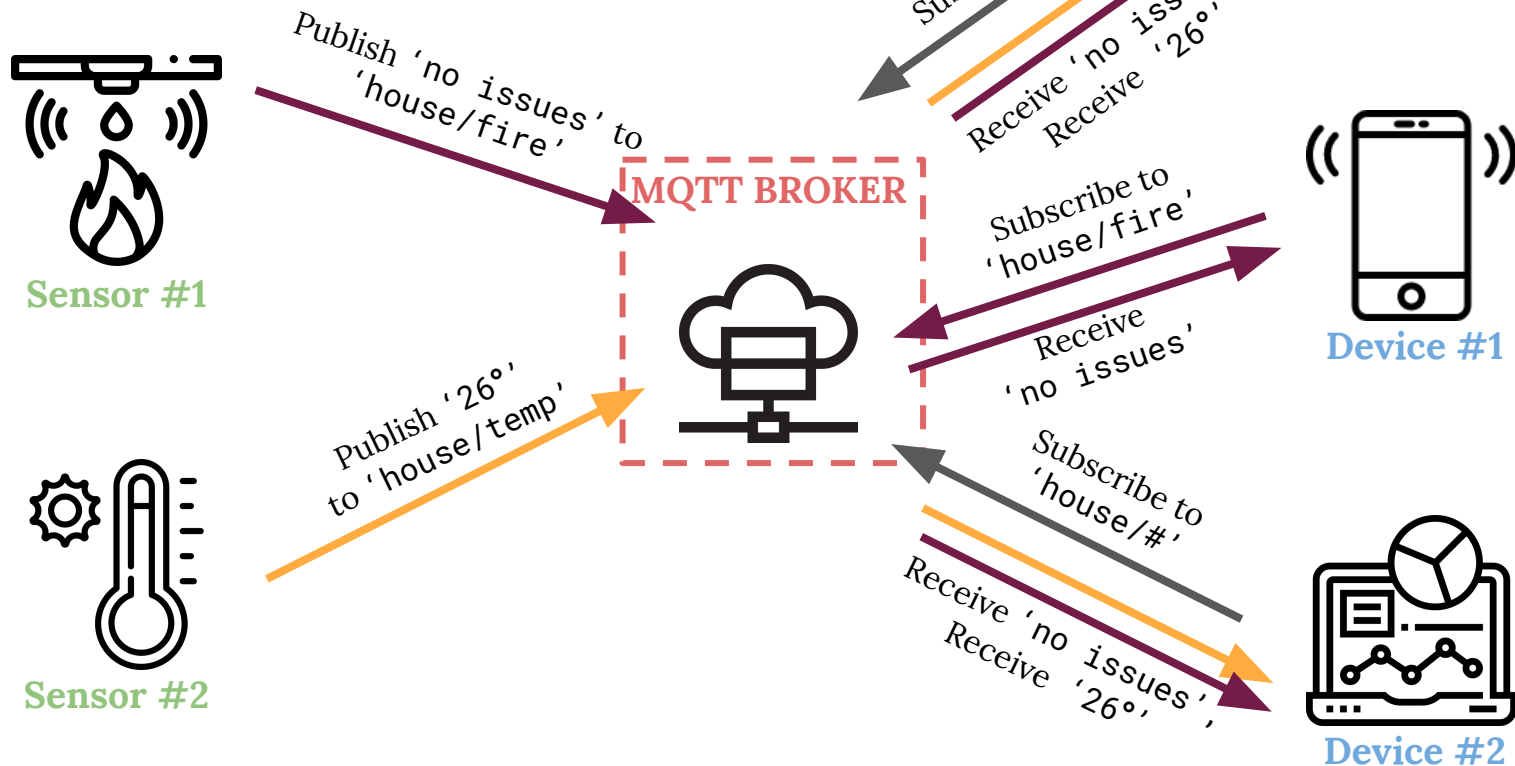


MQTT: what about *security*?

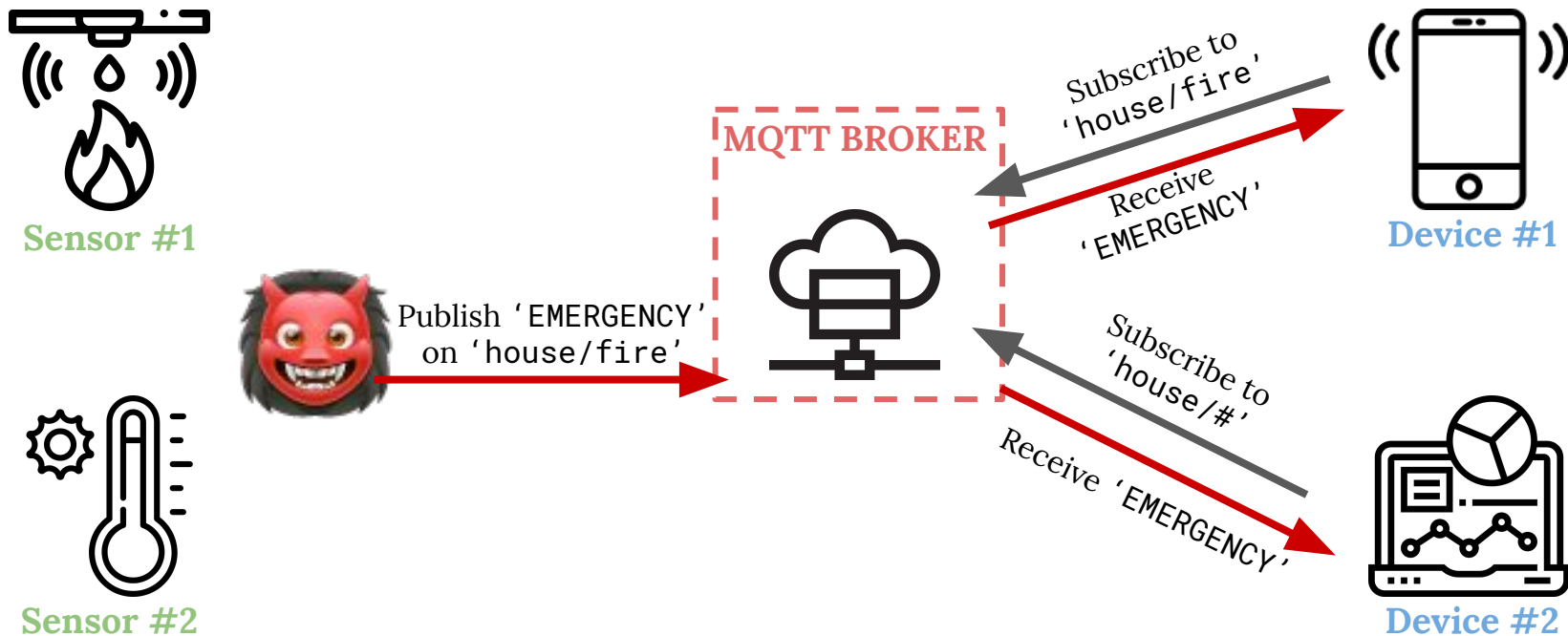
- Publishers have **no guarantees** that messages will be delivered to intended subscribers
- Subscribers have **no guarantees** about the identity of publishers (authentication is handled by brokers)
- Special topics:
 - ' # ': subscribe to **everything**
 - '\$SYS/#': **internal control messages** of the broker (how many connected devices, broker version)
- **TLS, authorization and authentication** not implemented by default



MQTT: confidentiality



MQTT: integrity





MQTT security from the docs

From the official documentation (ISO/IEC 20922:2016)*:

“There are a number of threats that solution providers should consider. For example:

- Devices could be **compromised**
- Data at rest in Clients and Servers might be **accessible**
- Denial of Service and timing **attacks**
- Communications could be **intercepted, altered, re-routed or disclosed** 😱
- **Injection** of spoofed Control Packets”

“MQTT solutions are often deployed in **hostile** environments”

Suggest to implement:

- **Authentication** of devices
- **Access control** to MQTT packets
- **Confidentiality** of MQTT packets
- **Integrity** of MQTT packets

NOT IMPLEMENTED BY DEFAULT!

*docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html



Evaluation: MQTT brokers in the wild

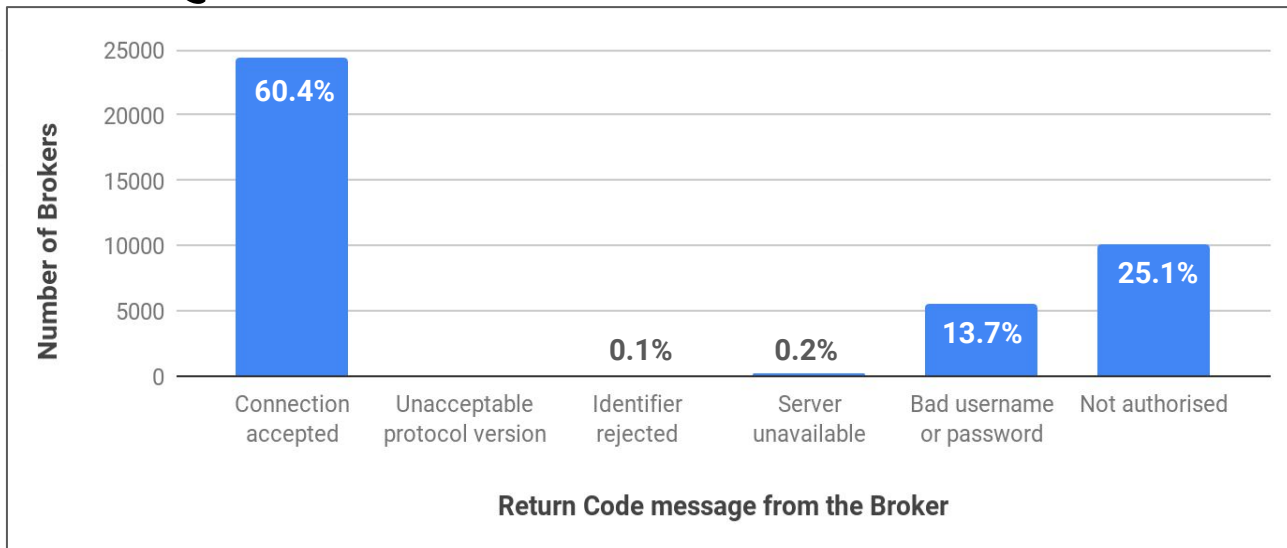
TOTAL RESULTS

79,563

TOP COUNTRIES



China	19,628
United States	13,835
Germany	4,928
Korea, Republic of	3,297
Japan	3,097





Why MQTTSA?

Available tools:

- **Fuzzing:** “MQTT security: A novel fuzzing approach” and F-Secure MQTT simple fuzzer
- **IoTVerif:** Automatic verification of certificates used by specific Android MQTT client applications in case MQTT brokers use TLS protocol
- **MQTT-PWN:** Automated penetration testing tool for MQTT brokers (credential bruteforcing, topic enumeration, identification and extraction of sensitive information)

MQTTSA:

- **Automatically** identify security problems (like the tools above)
- Focus on **security misconfigurations** in MQTT brokers
- Provide **assistance to mitigate** the detected security issues
- Consider also cases in which **TLS** is implemented



MQTTSA: modes

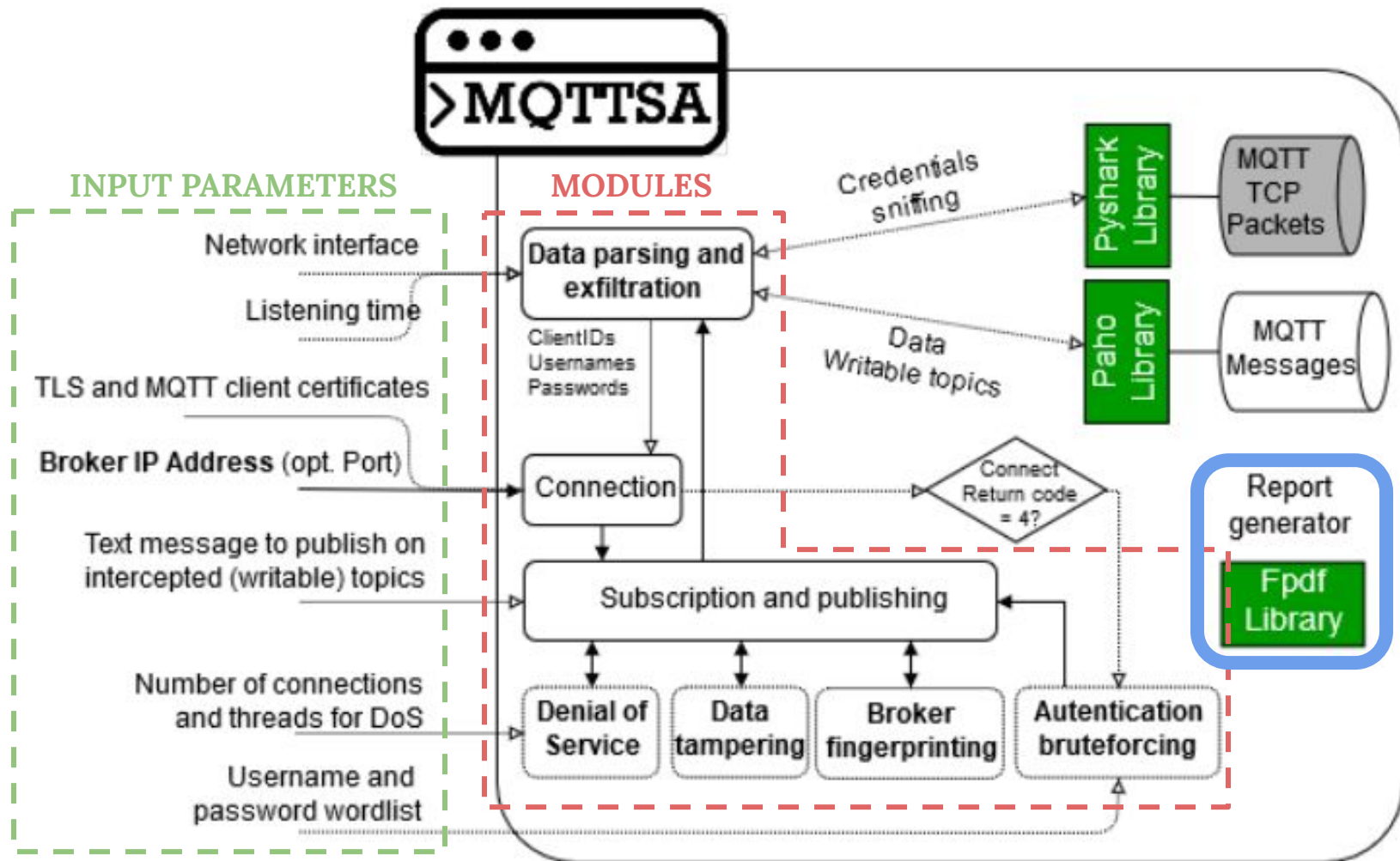
Intrusive mode

- connect
- subscribe
- publish
- broker fingerprinting
- denial of service
- data parsing & exfiltration
- data tampering
- credential bruteforce
- credential sniffing

Non-intrusive mode (`--ni`)

(for *critical* scenarios)

- connect
- subscribe





Evaluation: MQTT brokers in the wild (--ni)

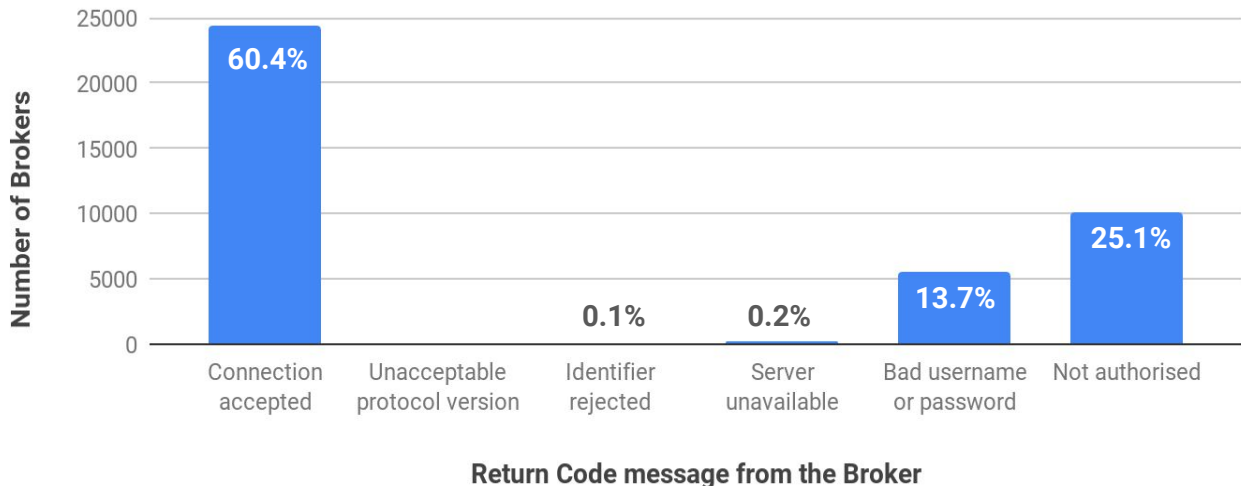
TOTAL RESULTS

79,563

TOP COUNTRIES



China	19,628
United States	13,835
Germany	4,928
Korea, Republic of	3,297
Japan	3,097



	Emails	Password keywords	GPS keywords	Domain names	IPv4 addr.s	MAC addr.s	TOTAL messages	TOTAL SYS messages
TOTAL	15,075	120,213	218,178	384,159	204,143	351,632	2,471,590	3,085,734
UNIQUE	1,796	59,272	182,298	60,666	71,472	57,771		



Evaluation: MQTT brokers in the lab

x = vulnerable | ✓ = secure | P.V. = partially vulnerable

Setup no.	Protocol	Authentic.	Authoriz.	Sniffing	Bruteforce	Subscribe	Publish	DoS
1	TCP	None	None	–	–	x	x	P.V.
2	TCP	usr/psw	None	x	x	x	x	P.V.
3	TLS	None	None	–	–	x	x	P.V.
4	TLS	usr/psw	None	✓	x	x	x	P.V.
5	TLS	certificate	ACL	✓	✓	✓	✓	✓




Future work

- User study: effectiveness of **MQTTSA**
- Expand capabilities of the tool:
 - Incorporate new attacks when CVE is available
 - Assessing security implications of using web-sockets
- Investigate features introduced in the latest version of the protocol (MQTT 5)



Andrea Palmieri

 [@andpalmier](https://twitter.com/andpalmier)

 andrea.palmieri@studenti.unitn.it

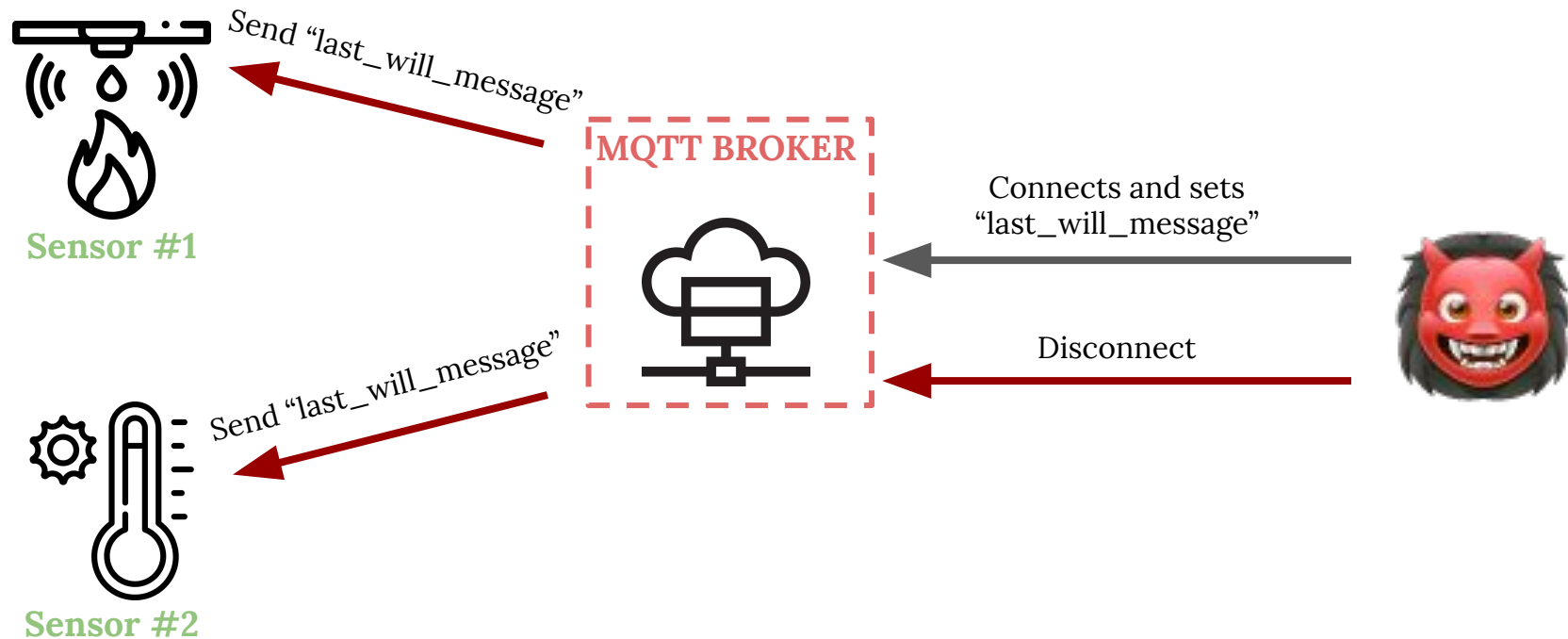


Paolo Prem

[@paolo_prem](https://twitter.com/paolo_prem) 

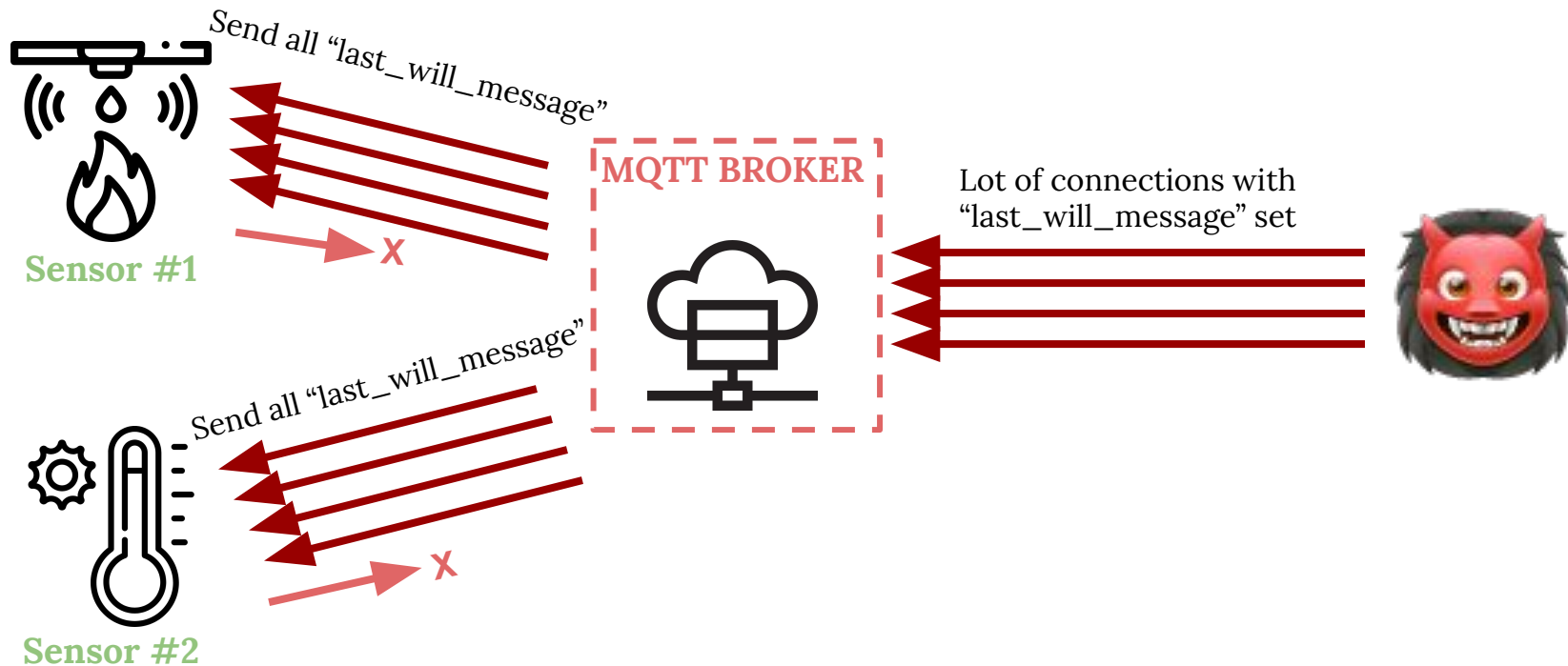
paolo.prem@studenti.unitn.it 

MQTTSA: DoS example





MQTTSA: DoS example





MQTTSA: modules

Connection:

Try to connect to the broker with the specified IP address

Subscribe & publish:

Try to subscribe to '#' and '\$SYS/#' to intercept messages

If some messages are intercepted:

Try to write the specified message in the topics found

Broker fingerprinting:

Try to identify broker info from the messages in the '\$SYS/#' topics

DOS: *(if the number of connections is specified with '-c <#connections>')*

Try to publish a large file performing several concurrent requests from a single process with multiple threads



MQTTSA: modules

Data parsing & exfiltration:

Save the information intercepted (messages and topics)

Detect leakage of sensitive data (with regex)

Data tampering: *(if enabled with the '--md' option)*

Try to crash the broker by triggering missing input validation

Exploiting specific vulns (CVE-2017-76507: ACL bypassed if username is '#' or '+')

Bruteforce: *(if user and wordlist are specified with '-u <username> -w <wordlist_path>')*

Try classic password bruteforce attack with specified username and wordlist



MQTTSA: additional features

Non intrusive analysis: *(if enabled with the '--ni' option)*

To be used in critical cases: MQTTSA will only try to intercept messages (not publishing messages, nor perform data tampering, bruteforce and DoS)

Sniffing credentials: *(if TLS is not used in the implementation, MQTTSA is executed in the same network of the MQTT client target and the interface is specified with '-i <interface>')*

Use the interface to intercept MQTT packets and look for credentials (*client ids* or *usernames*) that could be used



MQTTSA: additional features

Improved modules logic (connection and bruteforce):

If credentials are disclosed with the sniffing module, automatically use them to connect to the broker or to perform the bruteforce

TLS: (if TLS is enabled, the path for a CA certificate can be specified with '`--tls <ca_path>`', if required by the broker, the path of a client certificate and a key can be specified as well with '`--cert <cert_path> --key <key_path>`')
Performs all the specified tests on a MQTT implementation over TLS



Who are we?

- **Andrea Palmieri** and **Paolo Prem** {
Master's double-degree students in Cybersec @
UniTN, Trento & EURECOM, Sophia Antipolis}
- **Silvio Ranise** {
Head of Security & Trust @ Fondazione Bruno Kessler, Trento}
- **Umberto Morelli** {
Collaborator @ Fondazione Bruno Kessler, Trento}
- **Tahir Ahmad** {
PhD @ Fondazione Bruno Kessler, Trento}



MQTTSA: reporting

MQTTSA Report

Details of the assessment

Broker ip: 127.0.0.1
Listening time: 5
Text message: test_test_message
Denial of Service performed: False
Brute force performed: False

Parameters specified by the user

Module and result of the analysis

Authentication

[!] MQTTSA did not detect any authentication mechanism

The tool was able to connect to the broker without specifying any kind of credential information. This may cause remote attackers to successfully connect to the broker. It is strongly advised to support authentication via X.509 client certificates.

Suggested mitigations

Please follow those [guidelines](#) and modify Mosquitto's configuration according to the [official documentation](#). An excerpt of a configuration file is provided below:

```
listener 8883
cafile /etc/mosquitto/certs/ca.crt
certfile /etc/mosquitto/certs/hostname.crt
keyfile /etc/mosquitto/certs/hostname.key
require_certificate true
use_identity_as_username true
crlfile /etc/mosquitto/certs/ca.crl
```

Mitigation and tips based on
the results mentioned above

MQTT
deployment



MQTT
vulnerabilities



MQTTSA
report



MQTT deployment after
applying the tips of the
MQTTSA report

