AWS ALPN RolePlay

⚠ The following roleplay is suited for AWS pub-sub with MQTTS (≠ MQTT over websockets) ⚠

The Role Play doesn't use all the features of ioxy like intercepting message in live and modifying them, but feel free to try them.

Case

Today, I was scanning the home network and found a strange device (evil-device). I also found that the ssh port is open and I can get a shell by entering some generic credentials.

Analysis

I looked in every folder and the only interesting thing I found is this :

```
strangedev/
├── certs
│   ├── AmazonRootCA1.pem
│   ├── aws_evil.crt
│   └── aws_evil.key
├── evilPubSub.sh
└── pubsub.py
```

pubsub[.]py

Simple pub-sub client from AWS You can have a sneak peek over here

evilPubSub[.]sh

Code

```
python3 pubsub.py \
    --endpoint evil-ats.iot.eu-west-3.amazonaws.com \
    --root-ca certs/AmazonRootCA1.pem \
    --cert certs/aws_evil.crt \
    --key certs/aws_evil.key \
    --count "$1" # Number of payload to send
```

Behaviour

```
./evilPubSub 1
Connecting to evil-ats.iot.eu-west-3.amazonaws.com with client ID 'evil-client-id'...
```

```
Connected!
Subscribing to topic 'evil/rock'...
Subscribed with QoS.AT_LEAST_ONCE
Sending 1 message(s)
1 message(s) received.
Disconnecting...
Disconnected!
```

We notice that he is trying to send / receive messages on evil/rock topic

Walkthrough

After some analysis, we know that the evilPubSub send MQTT messages on amazon. It would be very interesting to catch the messages.
Happily,a friend of mine developed an awesome tool called ioxy to intercept MQTT messages stealthily.

Stealing the AWS client's certificate & key

```
# Start a http server on port 1111
root@evil-device:/home/strangedev/certs
➜ python3 -m http.server 1111

# Download certs on the proxy machine
hutchyy@hutchyy-VM:~/aws_evil_analysis
➜ wget -r [evil-device_ip:1111]

hutchyy@hutchyy-VM:~/aws_evil_analysis
➜ tree [evil-device_ip:1111]

[evil-device_ip:1111]
|-- AmazonRootCA1.pem
|-- aws_evil.crt
`-- aws_evil.key
```

Setting up the proxy

```
git clone https://github.com/NVISO-BE/internet-of-
things/tree/Embedded_verification_tool/ioxy
cd ~
cd ioxy/ioxy && go build .
```

Configuring the strange device

```
# Here is the tree
# <- is used by
```

```
hutchyy@hutchyy-VM:~/ioxy/ioxy/
➜ tree certs/

certs
|-- ca
|   |-- rootCA.key
|   `-- rootCA.pem  <- evil-device && ioxy
|-- devices
|   `-- d1
|       |-- d1.csr
|       |-- d1.key  <- evil-device
|       `-- d1.pem  <- evil-device
|-- README.md
`-- verificationCert
    |-- verificationCert.csr
    |-- verificationCert.key  <- ioxy
    `-- verificationCert.pem  <- ioxy
```

```
# Start a http server on port 1111
hutchyy@hutchyy-VM:~/ioxy/ioxy/certs/
➜ python3 -m http.server 1111

# Download client certs on the proxy machine
root@evil-device:/home/strangedev/certs
➜ wget -r [hutchyy_ip:1111]/devices/d1

# Download the CA cert
root@evil-device:/home/strangedev/certs
➜ wget [hutchyy_ip:1111]/ca/rootCA.pem
```

MiTM TIME !

On evil-device

```
# Modifying hosts file to resolve wanted name
echo "[hutchyy_ip]    ioxy.mqtt" >> /etc/hosts

# Ioxy certs tree
root@evil-device:/home/strangedev/
➜ tree [hutchyy_ip:1111]/

[hutchyy_ip:1111]/
├── ca
│   └── rootCA.pem <- evil-client
└── devices
    └── d1
        ├── d1.csr
        ├── d1.key  <- evil-client
        └── d1.pem  <- evil-client

# Modifying evilPubSub.sh to connect on our MQTT proxy (ioxy)
python3 pubsub.py \
    --endpoint  ioxy.mqtt \
    --root-ca   [hutchyy_ip:1111]/ca/rootCA.pem \
    --cert      [hutchyy_ip:1111]/devices/d1/d1.pem \
    --key       [hutchyy_ip:1111]/devices/d1/d1.key \
    --count     "$1" # Number of payload to send
```

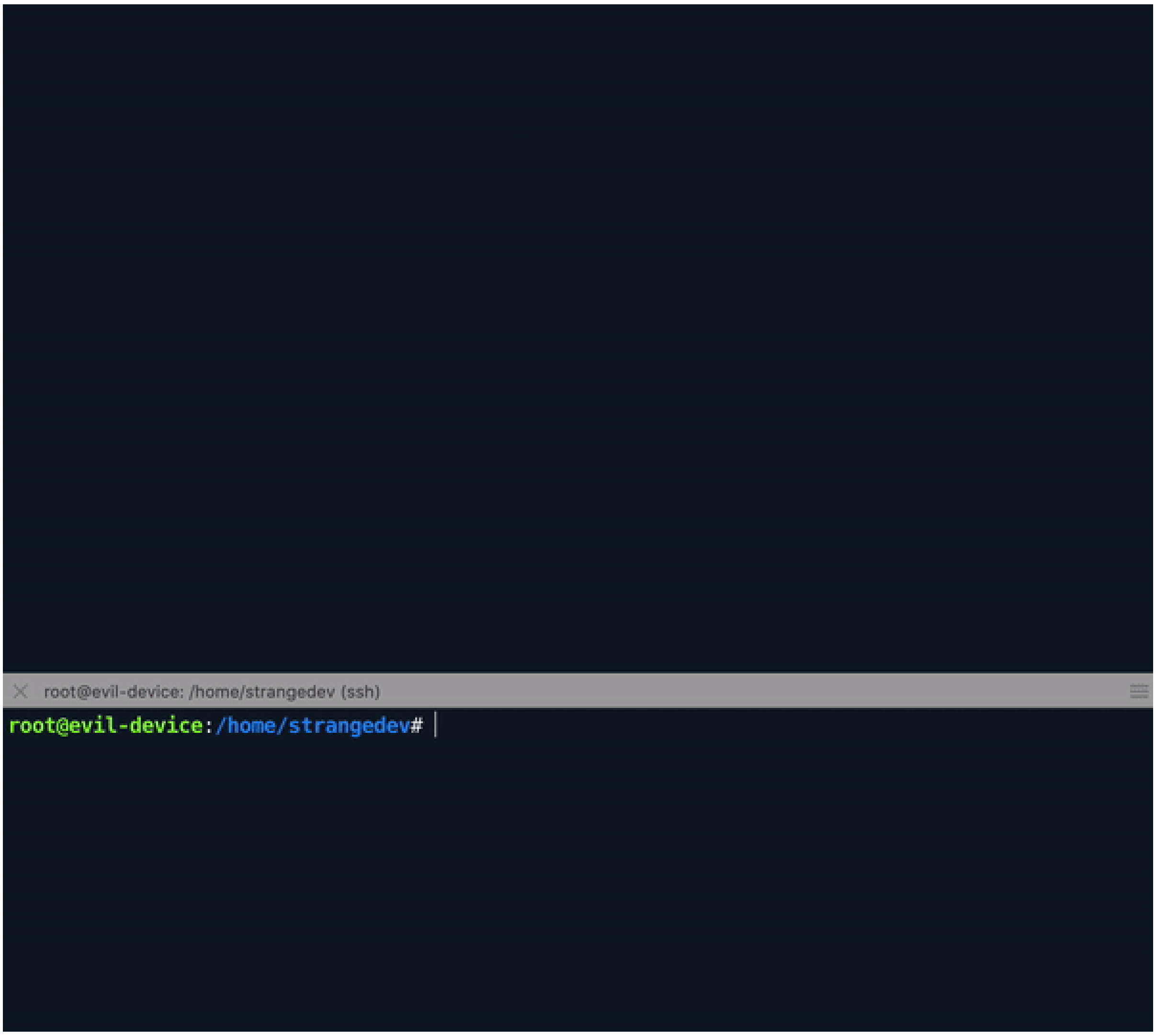
On hutchyy's vm

```
# Set env var to make ioxy command cleaner
AWSCRT = ~/aws_evil_analysis/[evil-device_ip:1111]/aws_evil.crt     && \
AWSKEY = ~/aws_evil_analysis/[evil-device_ip:1111]/aws_evil.key     && \
IOXYCA = ~/ioxy/ioxy/certs/verificationCert/verificationCert.pem    && \
IOXYCRT = ~/ioxy/ioxy/certs/verificationCert/verificationCert.pem   && \
IOXYKEY = ~/ioxy/ioxy/certs/verificationCert/verificationCert.key   && \
AWSEND = evil.iot.eu-west-3.amazonaws.com

# Start ioxy
hutchyy@hutchyy-VM:~/ioxy/ioxy/
→ sudo ./ioxy                       \
    mqtts                           \
    -mqtts-port         443         \
    -mqtts-cert         $IOXYCRT    \
    -mqtts-key          $IOXYKEY    \
    -mqtts-ca           $IOXYCA     \
    broker                          \
    -mqtt-broker-tls                \
    -mqtt-broker-host   $AWSEND     \
    -mqtt-broker-port   443         \
    -mqtt-broker-cert   $AWSCRT     \
    -mqtt-broker-key    $AWSKEY     \
    -x-amzn-mqtt-ca

[2020-03-17 12:19:14.283053739]  INFO Intercept : disabled
[2020-03-17 12:19:14.286094888]  INFO Starting mqtt-proxy @
[2020-03-17 12:19:14.286201998]  INFO Mode : mqtts
[2020-03-17 12:19:14.286232206]  INFO Broker :  evil.iot.eu-west-
3.amazonaws.com 443
[2020-03-17 12:19:14.286247846]  INFO auth : no auth url configured :
bypassing!
[2020-03-17 12:19:14.327314872]  INFO mqtts: listening on
mqtts://0.0.0.0:443
```

On evil-device

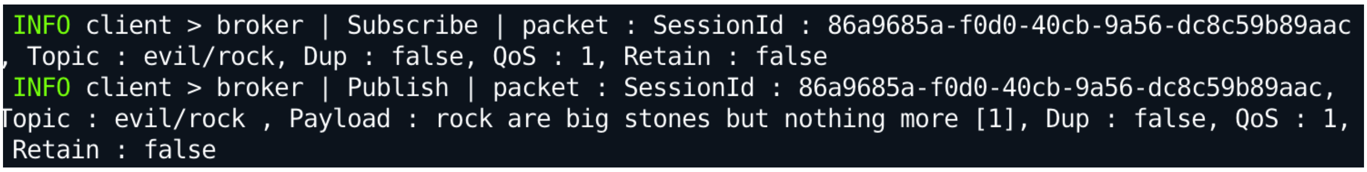
```
# Starting evilPubSub.sh
./evilPubSub.sh 1
```

Results



```
root@evil-device:/home/strangedev#
```

🎉 Yay !! We found it !



```
INFO client > broker | Subscribe | packet : SessionId : 86a9685a-f0d0-40cb-9a56-dc8c59b89aac
, Topic : evil/rock, Dup : false, QoS : 1, Retain : false
INFO client > broker | Publish | packet : SessionId : 86a9685a-f0d0-40cb-9a56-dc8c59b89aac,
Topic : evil/rock , Payload : rock are big stones but nothing more [1], Dup : false, QoS : 1,
Retain : false
```

1. The evil-device subscribed on evil/rock
2. The evil-device sent rock are big stones but nothing more to the broker on the evil/rock topic
3. The evil-device received is own message because he sent in a subscribed topic
4. The client disconnect

Case resolving

Now everything is clear !
This isn't an evil-device it's just my orangePi.

I just remember that a few months ago, I planned to climb the Mont Blanc and I wanted to have my orangePi (evil-device) with me. But why would you ask ! To communicate on AWS so my family could follow me in my journey. This was before I noticed that there is not WiFi Access Point on the Mont Blanc *sad*.

Now that we resolve the case, should I go to the Mont Blanc and add some WiFi AP or should I just stop my project ? *thinking*