

Meross MSS310 Precio: 12€

Documentación API:

<https://albertogeniola.github.io/Merosslot/>

Github:

<https://github.com/albertogeniola/Merosslot>

Peticiones directamente al enchufe

Configuramos un proxy con SQUID

Ejecutamos tcpdump para capturar los paquetes con el siguiente comando:

```
tcpdump -i any -s 65535 -w captura.tcpdump
```

Abrimos el dump con wireshark

Filtramos por petición http y la ip del dispositivo, en mi caso son:

```
192.168.1.53  
192.168.1.79  
192.168.1.123
```

Por ejemplo el filtro:

```
ip.dst == 192.168.1.79 and http
```

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.dst == 192.168.1.79 and http

No.	Time	Source	Destination	Protocol	Length	Info
334	3.798461	10.103.0.4	192.168.1.79	HTTP	353	POST /config HTTP/1.1 (application/json)
336	3.798475	192.168.1.200	192.168.1.79	HTTP	353	POST /config HTTP/1.1 (application/json)
376	3.801746	10.103.0.4	192.168.1.79	HTTP	353	POST /config HTTP/1.1 (application/json)
378	3.801758	192.168.1.200	192.168.1.79	HTTP	353	POST /config HTTP/1.1 (application/json)
388	3.802121	10.103.0.4	192.168.1.79	HTTP	385	POST /config HTTP/1.1 (application/json)
390	3.802134	192.168.1.200	192.168.1.79	HTTP	385	POST /config HTTP/1.1 (application/json)
400	3.806247	10.103.0.4	192.168.1.79	HTTP	359	POST /config HTTP/1.1 (application/json)
402	3.806259	192.168.1.200	192.168.1.79	HTTP	359	POST /config HTTP/1.1 (application/json)

[Calculated window size: 64256]
[Window size scaling factor: 128]
Checksum: 0xc5dc5 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
[SEQ/ACK analysis]
[Timestamps]
TCP payload (329 bytes)
TCP segment data (329 bytes)

0000 00 03 00 01 00 06 02 42 0a 67 00 04 00 00 08 00B.g.....
0010 45 00 01 71 07 56 40 06 40 06 65 cf 0a 67 00 04 E..q.V..@e.g..
0020 c0 a8 01 4f ac 96 00 50 88 57 54 75 2c db 09 2f ...O..P..WTu.../
0030 50 18 01 f6 cd c5 00 00 7b 22 68 65 61 64 65 72 P.....{"header
0040 22 3a 7b 22 66 72 6f 6d 22 3a 22 2f 61 70 70 2f ":{"from":"/app/
0050 37 32 34 33 39 31 2d 63 65 33 30 36 31 63 62 34 724391-ce3061cb4
0060 30 39 66 62 33 32 35 32 37 33 64 36 31 64 62 31 09fb325273d61db1
0070 66 64 63 39 38 38 62 2f 73 75 62 73 63 72 69 62 fdc988b/subscribe
0080 65 22 2c 22 6d 65 73 73 61 67 65 49 64 22 3a 22 e","messageId":
0090 30 64 63 39 38 30 38 31 39 35 62 34 38 61 33 35 0dd9808195b48a35
00a0 38 30 30 33 62 37 64 61 31 36 64 65 36 38 36 32 8003b7da16de6862
00b0 22 2c 22 6d 65 74 68 6f 64 22 3a 22 47 45 54 22 ","method":"GET"
00c0 2c 22 6e 61 6d 65 73 70 61 63 65 22 3a 22 41 70 ","namespace":"Ap
00d0 79 6c 6f 61 6e 63 65 2e 43 6f 6e 74 72 6f 6c 2e pliance.Control.
00e0 45 6c 65 63 74 72 69 63 69 74 79 22 2c 22 70 61 Electricity","pa
00f0 79 6c 6f 61 64 56 65 72 73 69 6f 6e 22 3a 31 2c yloadVersion":1,
0100 22 73 69 6f 6e 22 3a 22 31 37 38 62 65 64 31 35 "sign":"178bed15
0110 32 38 38 38 38 32 36 65 39 32 65 35 39 37 62 33 2888826e92e597b3
0120 62 30 38 65 36 32 66 63 22 2c 22 74 69 6d 65 73 b08e62fc","times
0130 74 61 6d 70 22 3a 31 35 39 37 30 34 37 34 33 37 tamp":1597047437,
0140 2c 22 74 72 69 6f 67 65 72 53 72 63 22 3a 22 41 ","triggerSrc":"A
0150 6e 64 72 6f 69 64 22 7d 2c 22 70 61 79 6c 6f 61 ndroid"},"payload"
0160 64 22 3a 7b 22 65 6c 65 63 74 72 69 63 69 74 79 d":{"electricity
0170 22 3a 7b 22 63 68 61 6e 6e 65 6c 22 3a 39 7d 7d ":{"channel":0}}
0180 7d }

Frame (385 bytes) Reassembled TCP (616 bytes)

A data segment used in reassembly of a lower-level protocol (tcp.segment_data), 329 bytes

Packets: 1564 · Displayed: 8 (0.5%) Profile: Default

Miramos las peticiones hasta que encontremos una que en el payload del post sale electricity:

```
{
  "header": {
    "from": "/app/724391-ce3061cb409fb325273d61db1fdc988b/subscribe",
    "messageId": "0dd9808195b48a358003b7da16de6862",
    "method": "GET",
    "namespace": "Appliance.Control.Electricity",
    "payloadVersion": 1,
    "sign": "178bed152888826e92e597b3b08e62fc",
    "timestamp": 1597047437,
    "triggerSrc": "Android"
  },
  "payload": {
    "electricity": {
      "channel": 0
    }
  }
}
```

Si hacemos un post con esa petición a <ip>/config tenemos los datos de consumo (añadir al final |python -m json.tool para poner formato legible) :

```
curl -d @peticion.json 192.168.1.79/config |python -m json.tool
```

```
{
  "header": {
    "from": "/appliance/20051881797870251h4148e1e91c65ce/publish",
    "messageId": "0dd9808195b48a358003b7da16de6862",
    "method": "GETACK",
    "namespace": "Appliance.Control.Electricity",
    "payloadVersion": 1,
    "sign": "67cd1c537334b5a6d9d8d3784f542135",
    "timestamp": 1597049595,
    "timestampMs": 431
  },
  "payload": {
    "electricity": {
      "channel": 0,
      "config": {
        "electricityRatio": 100,
        "voltageRatio": 188
      },
      "current": 287,
      "power": 41679,
      "voltage": 2293
    }
  }
}
```

Faltaría jugar con los timestamp por si caduca la petición

Para sacar solo el valor de power que es el que interesa, con jq (herramienta para parsear json) lanzamos:

```
curl --silent -d @peticion.json 192.168.1.79/config|jq -r
'.payload.electricity.power'
```

42976

From:

<http://wiki.legido.com/> - **Legido Wiki**

Permanent link:

<http://wiki.legido.com/doku.php?id=energia:monitorizacion:enchufes:meross>

Last update: **2020/08/10 08:53**

