Hacking TP-Link Devices

Fernando Gont

SI6 NETWORKS

NGI @ Troopers 17
Heidelberg, Germany. March 20-24, 2017

About...

- Security Researcher and Consultant at SI6 Networks
- Published:
  - 30 IETF RFCs
  - 10+ active IETF Internet-Drafts
- Author of the SI6 Networks' IPv6 toolkit
  - https://www.si6networks.com/tools/ipv6toolkit
- Admin of a few mailing-lists:
  - {**ipv6hackers**, **iot-hackers**, **sdn-hackers**}**@lists.si6networks.com**
- More information at: https://www.gont.com.ar

Motivation for this work

Motivation

- People are connecting **everything** to the network
  - The so-called “Internet of Things” (also “Internet of S...” ;-))
- Impact of attacks tends to get more “physical”
- Are these “things” prepared for the real world?

Why TP-Link devices?

Why TP-link devices?

- Reasonable price
  - You don't want to spend 50 EUR on a “smart plug”
- They tend to be rather “open”
  - Possible to overwrite their firmware
- Easily available / rather popular
  - I had some available to play with
  - It's also nice to learn about the stuff you're using

TP-Link smart devices

HS110

NC250

Previous work & tools

Previous work & Tools

- Great research on TP-Link Smart plugs by Lubomir Stroetmann (Softcheck):
  - https://www.softscheck.com/en/reverse-engineering-tp-link-hs110/
- Reverse-engineered a protocol employed by TP-Link devices
- Implemented some PoC
- **Very** valuable work!

Our work

Our work

- Further research on the involved protocols
  - Possible attacks on the protocol itself
  - Extended existing analysis by sniffing traffic & implementing tools
- Produce more elaborate tools
- SI6 Networks IoT toolkit v1.0
  - https://www.si6networks.com/tools/iot-toolkit
  - Released during this conference!
- Use this project to trigger other work and brainstorming

TP-Link Smart Plugs

TP-Link Smart Plugs (HS110, HS100)

- Allow remote operation of on/off switch
- Allow timers, event scheduling, etc.
- Some (HS110) are able to measure power consumption
- Can be locally-operated (WiFi)
- Also allow for “cloud” operation

TP-Link Smart Plug Operation

- Main protocol: TP-Link Smart Plug Protocol
  - Available on port 9999 for both TCP and UDP
- Also support TDDP, a debugging protocol
- Some (HS110) are able to measure power consumption
- Can be locally-operated (WiFi)
- Also allow for “cloud” operation

TP-Link Smart Plug Protocol

Introduction

TP-Link Smart Plug Protocol

- Available on port 9999 for both TCP and UDP
- Encrypted
  - “Obfuscated”, you'd say
- JSON-based protocol
- Used for:
  - Device discovery
  - Device configuration
  - Polling and/or modifying device state

Difference between TCP & UDP versions

- UDP-based version:
  - Entire payload devoted to JSON command
  - Commands can be broadcasted
- TCP-based version:
  - Every command is preceded by 4-byte payload length in Network Byte Order
  - Obviously, commands canot be broadasted

TP-Link Smart Plug Protocol

Encryption/Decryption

TP-Link Protocol “Encryption”

- Protocol employs an algorithm to obfuscate the payload
- Encryption:

```
k= 171;
for(i=0; i<LEN; i++){
     t= b[i] xor k;
     k= b[i];
     b[i]= t;
}
```

“XOR each byte with the previous (plaintext) byte. Initial byte is XORed with special value 171”

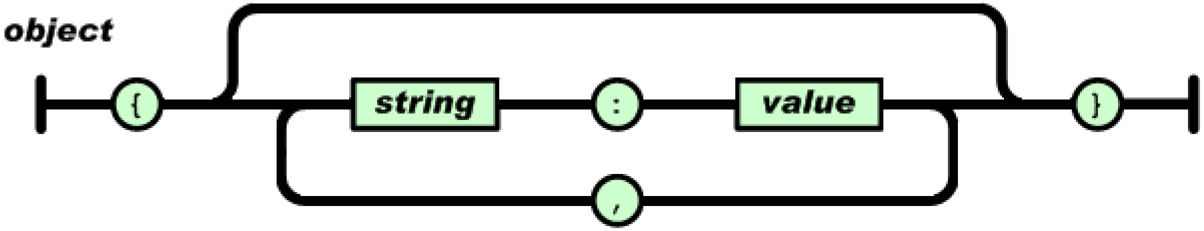
TP-Link Protocol “Decryption”

- Simply invert the algorithm from the previous slide
- Decryption:

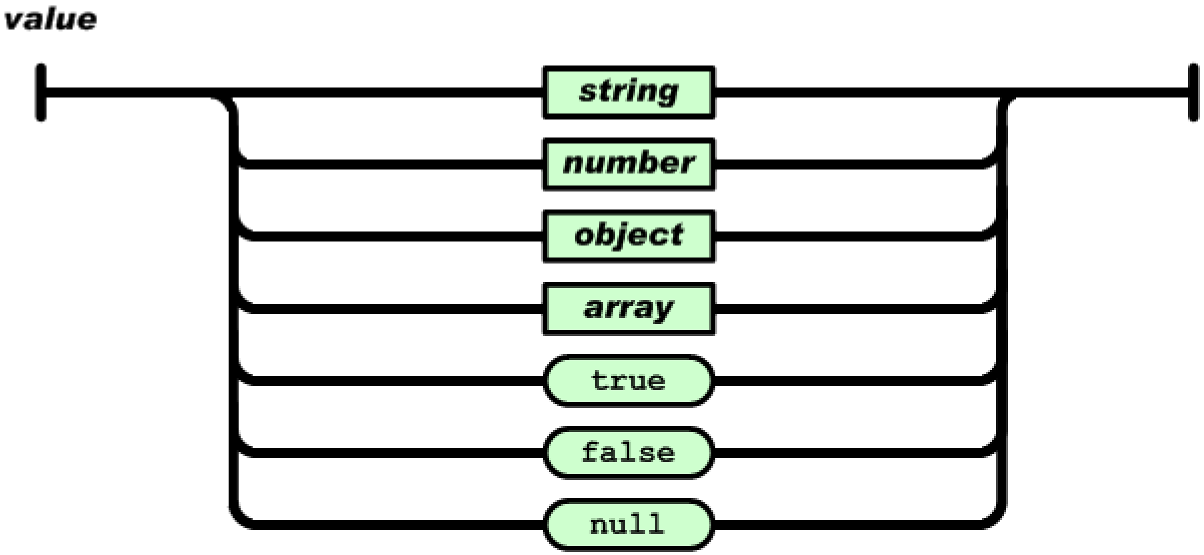
```
k= 171;
for(i=0; i<LEN; i++){
    b[i]= b[i] xor k;
    k= b[i];
}
```

JSON Primer

- JSON is a text-based way to encode data (just as XML is)
- JSON objects take this form:



where:



JSON Primer (II)

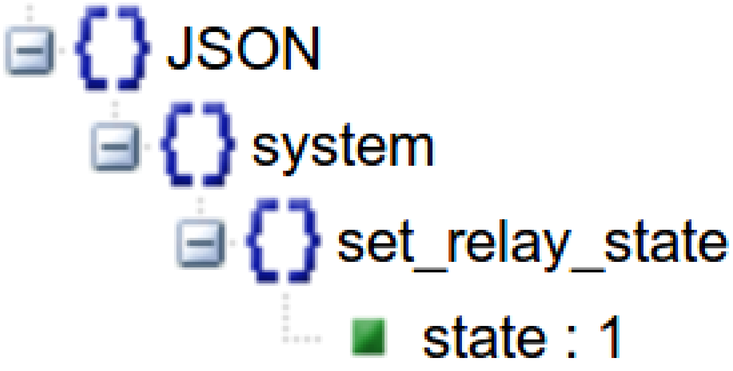
- A sample command, to turn the relay “on”:

```
{"system":{"set_relay_state":{"state":1}}}
```

- Sample response (successfull command):

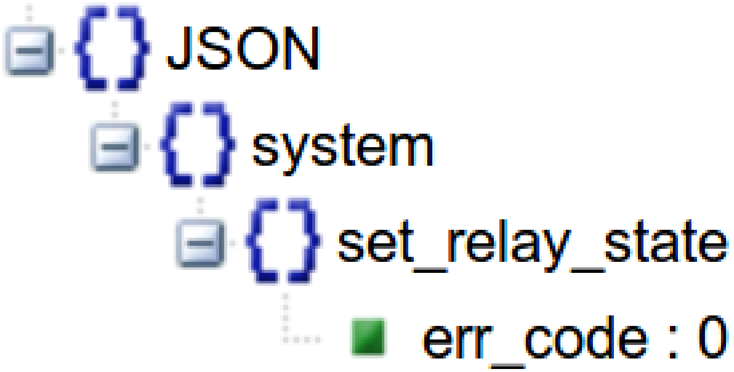
```
{"system":{"set_relay_state":{"err_code":0}}}
```

Command



```
JSON
  system
    set_relay_state
      state : 1
```

Response



```
JSON
  system
    set_relay_state
      err_code : 0
```

TP-Link Smart Plug Protocol

Finding devices on the local network

Finding devices on the local network

- The TP-Link app discovers smartplugs by broadcasting:

```
{"system":{"get_sysinfo":null},"emeter":
{"get_realtime":null}}
```

- These are two queries in the same packet:
  - “system”: Module available on all TP-Link Smart Plugs
  - “emeter”: Energy Monitoring module (available in HS110 model)
- The response will include, among others:
  - Type and model of the device
  - Hardware and software version
  - Device alias
- A single query is enough for exact fingerprinting

Scanning for SmartPlugs with iot-scan

- Sample command:

```
fgont@matrix:~/code/iot-toolkit $ ./iot-scan -i eth0 -L
192.168.3.66 # smartplug: TP-Link HS100(EU): Wi-Fi Smart Plug: "mio"
192.168.3.42 # camera: TP-Link IP camera
192.168.3.43 # camera: TP-Link IP camera
```

Issuing commands with iot-tl-plug

- Sample command:

```
fgont@matrix:~/code/iot-toolkit $ sudo ./iot-tl-plug -L -i eth0 -c
get_info
Got response from: 192.168.3.66, port 9999
{"system":{"get_sysinfo":{"err_code":0,"sw_ver":"1.0.8 Build 151101
Rel.24452","hw_ver":"1.0","type":"smartplug","model":"HS100(EU)","ma
c":"50:C7:BF:00:C4:D0","deviceId":"8006BE9B2C1A6114DBFA0632B02D566D1
70BC38A","hwId":"22603EA5E716DEAEA6642A30BE87AFCA","fwId":"BFF24826F
BC561803E49379DBE74FD71","oemId":"812A90EB2FCF306A993FAD8748024B07",
"alias":"mio","dev_name":"Wi-Fi Smart
Plug","icon_hash":"","relay_state":0,"on_time":0,"active_mode":"sche
dule","feature":"TIM","updating":0,"rssi":-
52,"led_off":0,"latitude":0,"longitude":0}},"emeter":{"err_code":-
1,"err_msg":"module not support"}}
```

TP-Link Smart Plug Protocol

Vulnerabilities & Potential Problems

The obvious

- No encryption or authentication
- UDP-based version of the protocol allows for source address spoofing

Amplification

- One 40-byte query: ({"system":{"get_sysinfo":null}}) will result in a 500-byte response
- A single packet may contain multiple instances of the same query, exacerbating this problem:

```
{"system":{"get_sysinfo":null},"system":
{"get_sysinfo":null},"system":
{"get_sysinfo":null},"system":
{"get_sysinfo":null}}
```

- Nice for amplification
  - but protocol is only local

DoS Attack vector

- Protocol Design 101: “Error messages must not elicit error messages”
- However, a message meant to a non-existing module:

```
{"DoSme":{"err_code":-1,"err_msg":"module not
support"}}
```

will elicit the following response:

```
{"DoSme":{"err_code":-1,"err_msg":"module not
support"}}
```

- One packet will cause a packet war
- This is even worse when original paket is broadcasted

DoS Attack vector: Variant #1

- Packet:
  - Source Address: victim
  - Source Port: 9999
  - Destination Address: victim
  - Destination Port: 9999
  - Payload:

    {"DoSme":{"err_code":-1,"err_msg":"module not support"}}
- This will trigger a packet storm inside the device itself

DoS Attack vector: Variant #2

- Packet:
  - Source Address: victim_1
  - Source Port: 9999
  - Destination Address: victim_2
  - Destination Port: 9999
  - Payload:

    {"DoSme":{"err_code":-1,"err_msg":"module not support"}}
- This will trigger a packet storm between two devices, and possible DoS the network

Fast switching

- Switch on/off very fast:

```
$ iot-tl-plug --toggle TARGET#CYCLE#LENGTH
```

- e.g.

```
$ iot-tl-plug --toggle 255.255.255.255#50#120
```

“*Toggle the relay state of all local smart plugs every 50 ms, for two minutes*”

TP-Link Device Debug Protocol (TDDP)

Introduction

Introduction

- TDDP not used actively for Smart Plugs
- Originally found by reverse engineering
- **Concept** described in a patent

  http://www.google.com/patents/CN102096654A?cl=en
- Protocol employed in other TP-Link devices, **with changes**
- **Not really possible to use TDDP across all TP-Link devices**

TDDP in Smart Plugs

- Simple command-response UDP-based protocol
- Commands must be sent to UDP port 1040
- Responses are received on UDP port 61000
- Employs MD5 as checksum
- Employs DES for encryption

Packet format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Ver      |     Type      |     Code     |   ReplyInfo     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          PktLength                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             PktID             |    SubType   |     Reserve     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        MD5 Digest[0-3]                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        MD5 Digest[4-7]                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        MD5 Digest[8-11]                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        MD5 Digest[12-15]                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

iot-tddp: A TDDP implementation

- You can send arbitrary TDDP messages with iot-tddp.
- Example:

```
$ iot-tddp -d 192.168.3.41 -a 1068
Sending TDDP Packet:
Version: 02   Type: 03   Subtype: 00  Code: 01  ReplyInfo: 00
PktLength: 00000000
PktId: 2000    MD5 Digest: 719085ea0e8c06ab63efca3261461efd
Payload:

Read 28 bytes from 192.168.3.41
Version: 02   Type: 03   Subtype: 17  Code: 02  ReplyInfo: 03
PktLength: 00000000
PktId: 0000   MD5 Digest: 72a9a232add865ae7840ad6208f93416
Payload:
```

TP-Link Cameras

TP-Link IP Cameras (NC220, NC250)

- IP cameras
- Motion detection & notifications
- Support different video resolutions
- Can be locally-operated (WiFi)
- Also allow for “cloud” operation

TP-Link Cameras Operation

- Done via web interface or TDDP
- Video and audio streams, plus camera snapshots available via HTTP
- Examples:

TP-Link Device Deployment Advice

How to use them while reducing trouble

Some deployment guidelines

- Employ a separate network for your IoT devices
  - Anyone with local network access owns you
- Prevent IoT devices from calling (TP-Link) home
  - Overwrite the “cloud” URL
  - Block TP-Link cloud domains & IP addresses
- Replace Tp-Link app with your own
  - Customized web site with firing commands with our toolkit

How will IPv6 affect us?

Futurology

IoT & IPv6: Brief overview

- Most of these IoT devices:
  - Have immature implementations
  - Use insecure protocols
  - Are unlikely to get patched
- IPv6 potentially makes all these devices globally reachable
- **It is extremely likely that that will result in a lot of trouble**

IoT & IPv6: A way forward

- The whole point of IPv6 is its increased address space
  - i.e., be directly connected to the Internet **when and if you need it**
- Having a unique address **need not** imply being reachable
- Connectivity requirements essentially depend on:
  - Push vs pull model
  - Most of these IoT devices employ the pull model!
- At the very least, your IoT devices should be connected with a “diode” firewall
  - This is a side-effect in IPv4 NAT

SI6 Networks' IoT Toolkit

New tools

IoT Toolkit

- Formally released during Troopers
- Repo already available at:

  https://github.com/fgont/iot-toolkit

Questions?

Thanks!

Fernando Gont

fgont@si6networks.com

IoT Hackers mailing-list

http://www.si6networks.com/community/

www.si6networks.com