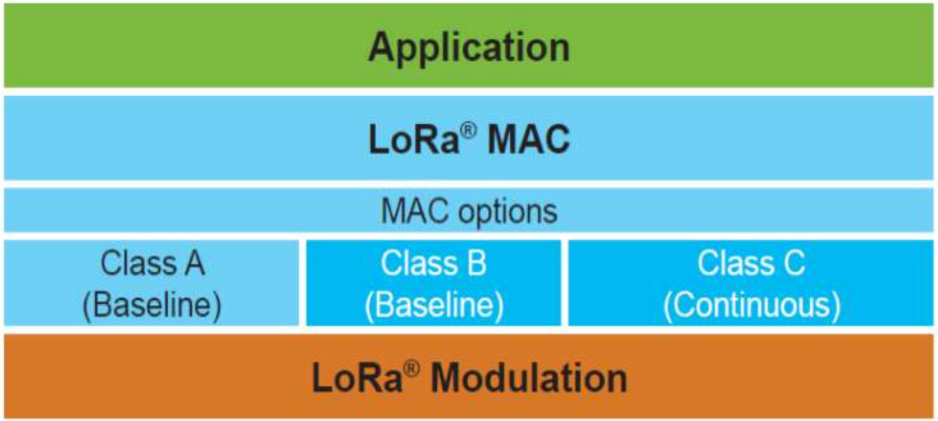
LoRaWAN Auditing Framework (*LAF*)

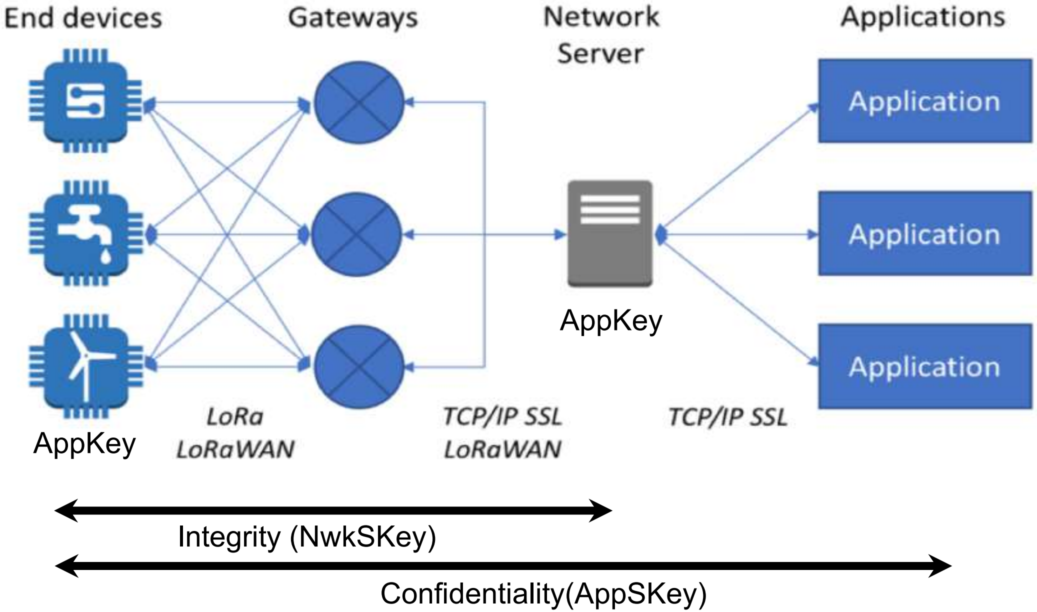
About LoRaWAN

- IoT Protocol
- Built on top of LoRa
  - Low power
  - High range: 1/4km urban y 5/20km rural
  - Unlicensed spectrum (free)
  - Robust
  - From 0.3 up to 50 kbps



LoRaWAN Stack

LoRaWAN Architecture (1.0.* version)



AppKey

AppKey

Integrity (NwkSKey)

Confidentiality(AppSKey)

AES(AppKey, 0x1 / 0x2 + AppNonce + NetID + DevNonce) = AppSKey / NwkSkey

(in)Security in LoRaWAN

- Known vulnerabilities
  - Replay attacks, eaveasdropping, ack spoofing, bitflipping
- Implementation issues
  - Use of well-known or nonrandom keys
  - Devices physically exposed
  - Lack of best practices standard
- Lack of tools to pentest, audit, and protect a LoRaWAN network

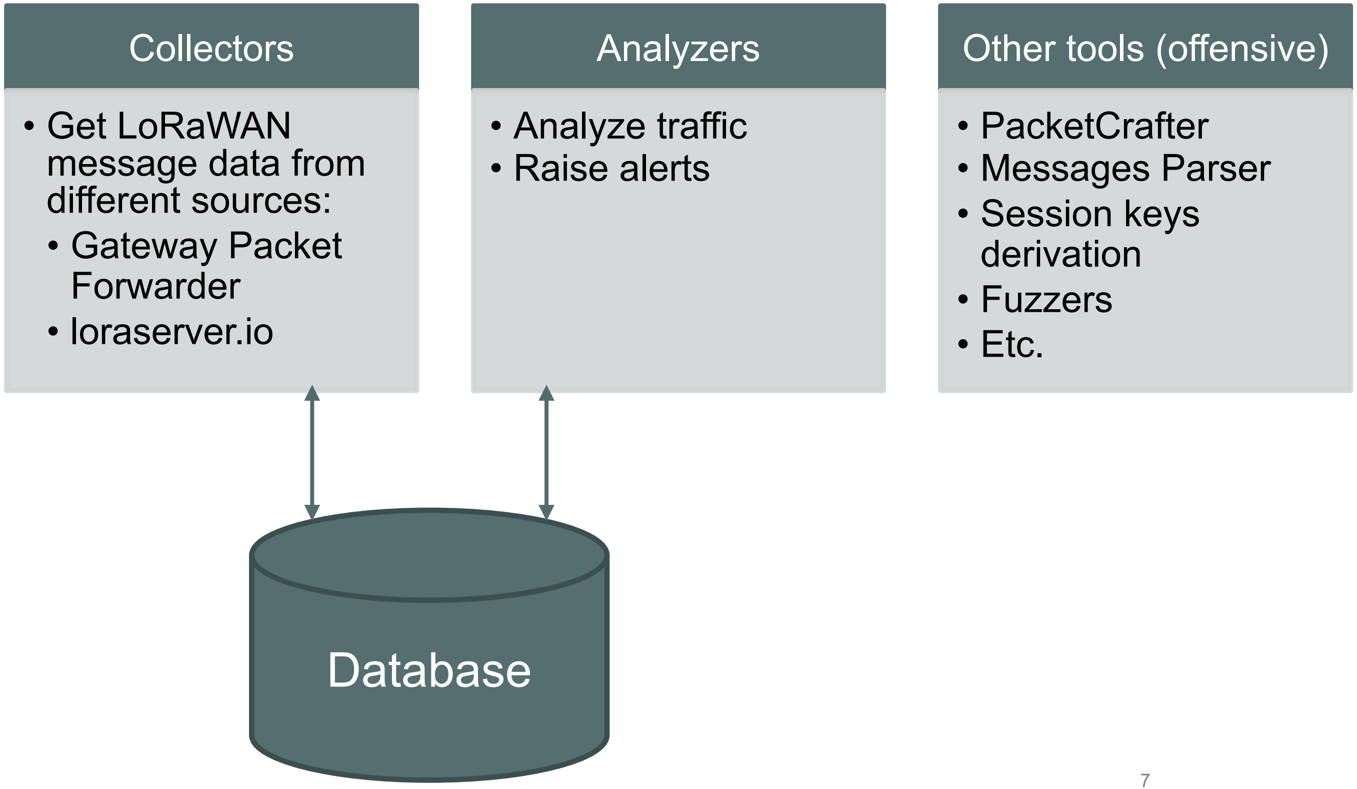
LoRaWAN Auditing Framework (LAF)

- Pentest tools:
  - Traffic sniffing, spoofing, and fuzzing. Keys cracking.
- LoRaWAN messages collectors:
  - loraserver.io / packet_forwarder / write your own collector ☺
- Traffic analyzers to detect :
  - Join replays
  - Possible ABP activated devices
  - Well known or nonrandom keys
  - Duplicated session keys / attacker sending valid messages
  - Devices in the network

LAF Architecture

- Written in python3
- Modular
  - You can contribute developing more messages collectors, analyzers, etc
- SQLite or PostgreSQL / Standalone or Dockerized

LAF Architecture



Collectors

- Get LoRaWAN message data from different sources:
  - Gateway Packet Forwarder
  - loraserver.io

Analyzers

- Analyze traffic
- Raise alerts

Other tools (offensive)

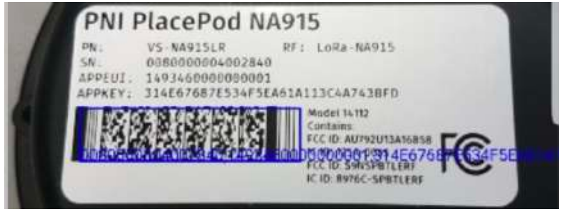
- PacketCrafter
- Messages Parser
- Session keys derivation
- Fuzzers
- Etc.

Database

Demo

Prerequisite: to have stolen / cracked an AppKey:

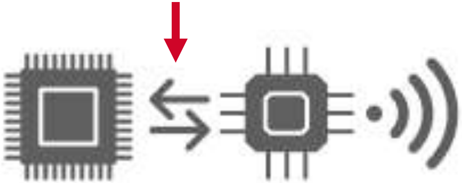
- Device’s reverse engineering
- Tag sticked to a device
- Hardcoded keys in open source code
- Easy to guess or nonrandom keys
- Network servers with default credentials
- Servers with vulnerabilities

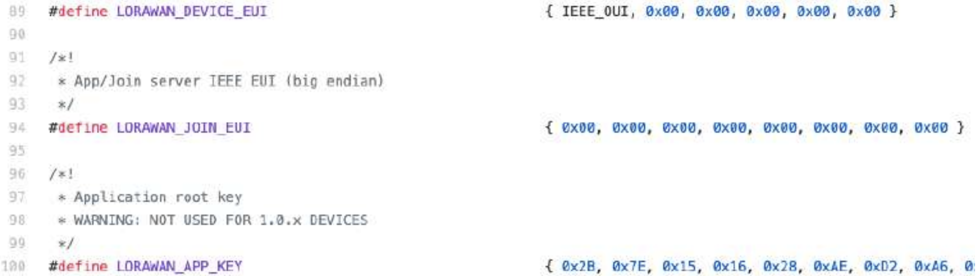


PNI PlacePod NA915

PN: VS-NA915LR RF: LoRa-NA915
SN: 0080000004002840
APPEUI: 1493460000080001
APPKEY: 314E67687E534F5EA61A113C4A743BFD

Model 14112
Contains
FCC ID: AU792U13A16858
FCC ID: S9NSPBTLERF
IC ID: 8976C-SPBTLERF

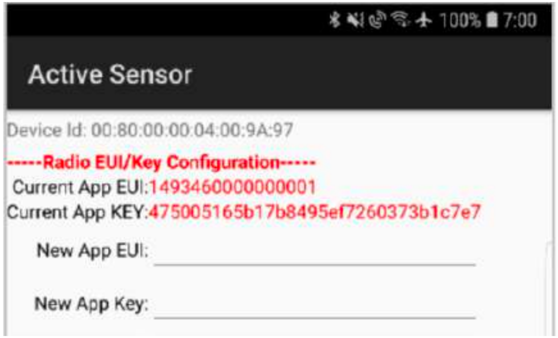




```
#define LORAWAN_DEVICE_EUI                                  { IEEE_OUI, 0x00, 0x00, 0x00, 0x00, 0x00 }

/*!
 * App/Join server IEEE EUI (big endian)
 */
#define LORAWAN_JOIN_EUI                                    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 }

/*!
 * Application root key
 * WARNING: NOT USED FOR 1.0.x DEVICES
 */
#define LORAWAN_APP_KEY                                     { 0x2B, 0x7E, 0x15, 0x16, 0x28, 0xAE, 0xD2, 0xA6, 0
```



Active Sensor

Device Id: 00:80:00:00:04:00:9A:97

-----Radio EUI/Key Configuration-----

Current App EUI:1493460000000001

Current App KEY:475005165b17b8495ef7260373b1c7e7

New App EUI:

New App Key:

Demo

Prerequisite: to have stolen / cracked an AppKey:

STEPS

1. Obtain (sniff) a JoinRequest and uplink data packet
2. Crack the session keys
3. Parse and decrypt a data packet
4. Craft a valid packet with a bigger counter
5. Send the packet though the gatevice using the sender
6. Check network server result
7. Check LAF alert

GET
the
LoRaWAN Auditing Framework

https://github.com/IOActive/laf