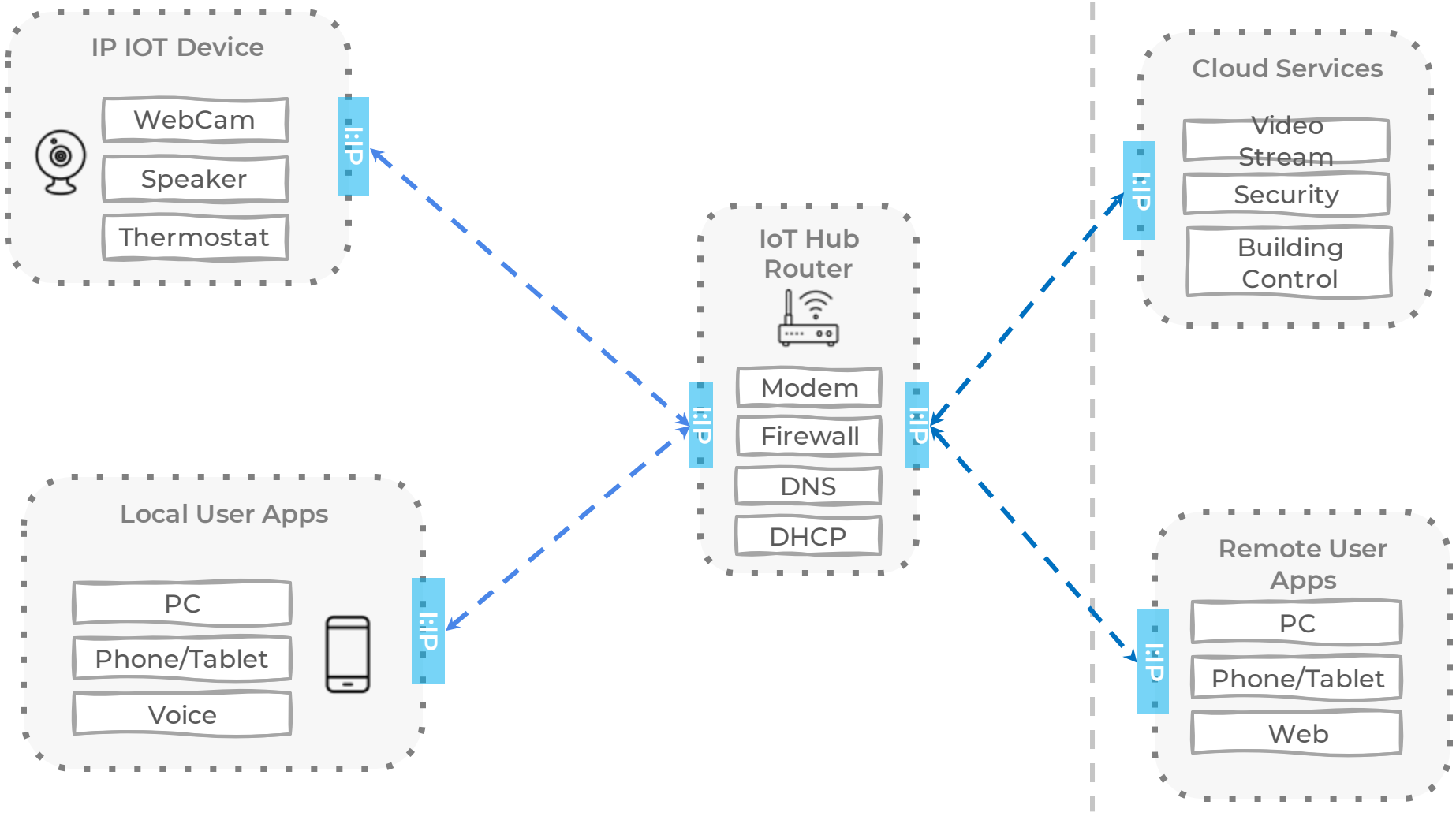
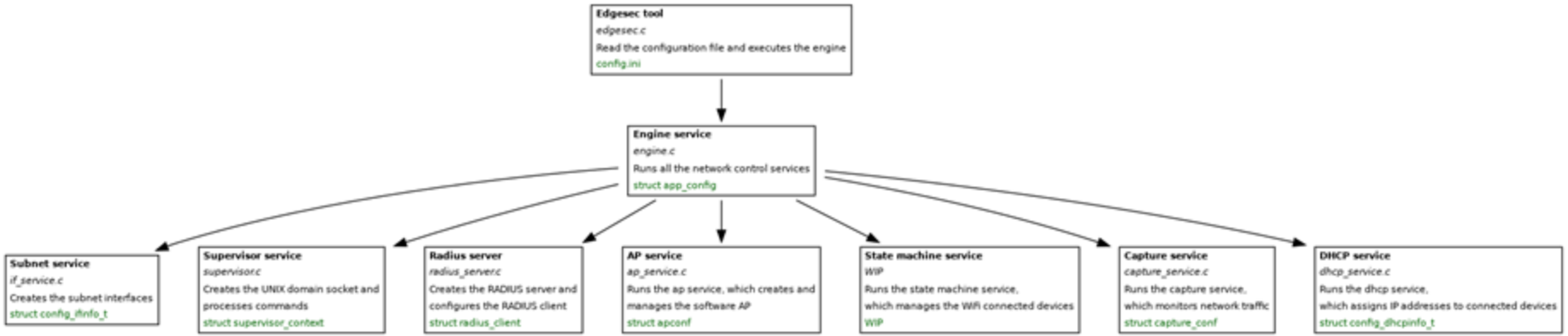
ManySecured Workshop

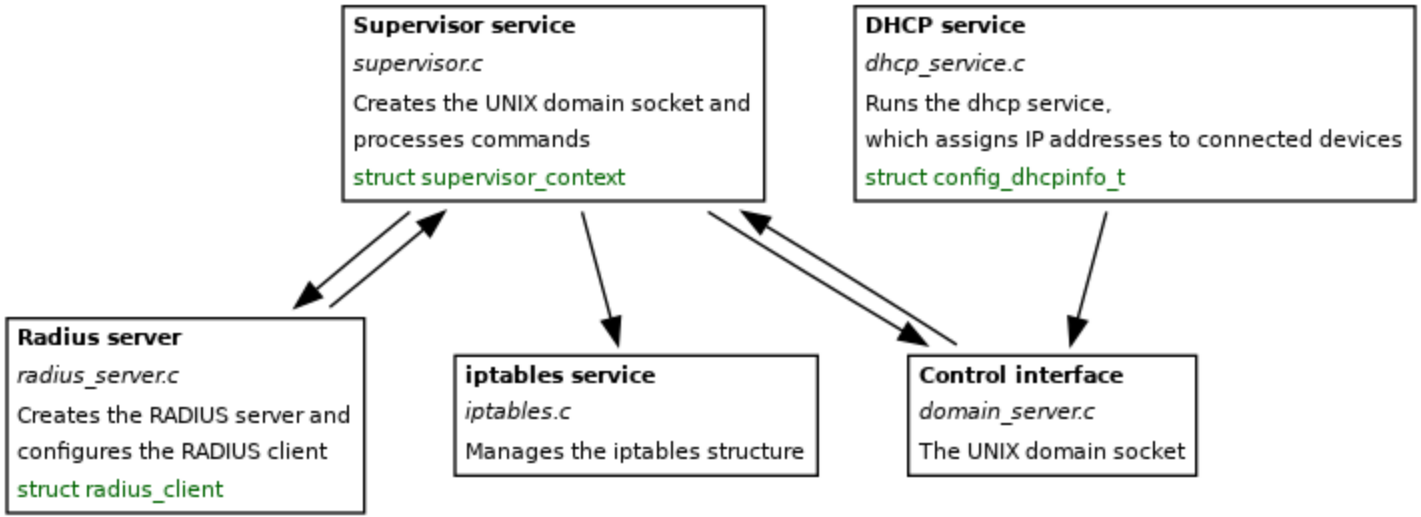
Alexandru Mereacre


IP IOT Device
WebCam
Speaker
Thermostat
IP
Cloud Services
Video Stream
Security
Building Control
IP
IoT Hub Router
Modem
Firewall
DNS
DHCP
IP
IP
Local User Apps
PC
Phone/Tablet
Voice
IP
Remote User Apps
PC
Phone/Tablet
Web
IP

Architecture



Supervisor service



Subnet service

This service creates subnets and maps VLAN IDs to a subnet IP range. It uses the Netlink protocol library suite to access network kernel functionality from the user space.

```
if0 = "0,10.0.0.1,10.0.0.255,255.255.255.0"
if1 = "1,10.0.1.1,10.0.1.255,255.255.255.0"
if2 = "2,10.0.2.1,10.0.2.255,255.255.255.0"
if3 = "3,10.0.3.1,10.0.3.255,255.255.255.0"
if4 = "4,10.0.4.1,10.0.4.255,255.255.255.0"
if5 = "5,10.0.5.1,10.0.5.255,255.255.255.0"
if6 = "6,10.0.6.1,10.0.6.255,255.255.255.0"
if7 = "7,10.0.7.1,10.0.7.255,255.255.255.0"
if8 = "8,10.0.8.1,10.0.8.255,255.255.255.0"
if9 = "9,10.0.9.1,10.0.9.255,255.255.255.0"
if10=
"10,10.0.10.1,10.0.10.255,255.255.255.0"
```

Subnet service

For each subnet the service creates a bridge interface and assigns an IP.

- Create interface
- Set interface IP
- Set interface state

Subnet service

For each subnet the service also resets the iptables firewall rules.

- Reset iptables
- Set default iptables rules for each subnet interface

```
iptables -P INPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -P OUTPUT ACCEPT
iptables -F -t nat
iptables -F -t mangle
iptables -F
iptables -X
```

```
iptables -A FORWARD -t filter -i br0 -j REJECT
iptables -A FORWARD -t filter -i br1 -j REJECT
iptables -A FORWARD -t filter -i br2 -j REJECT
iptables -A FORWARD -t filter -i br3 -j REJECT
iptables -A FORWARD -t filter -i br4 -j REJECT
iptables -A FORWARD -t filter -i br5 -j REJECT
iptables -A FORWARD -t filter -i br6 -j REJECT
iptables -A FORWARD -t filter -i br7 -j REJECT
iptables -A FORWARD -t filter -i br8 -j REJECT
iptables -A FORWARD -t filter -i br9 -j REJECT
iptables -A FORWARD -t filter -i br10 -j REJECT
```

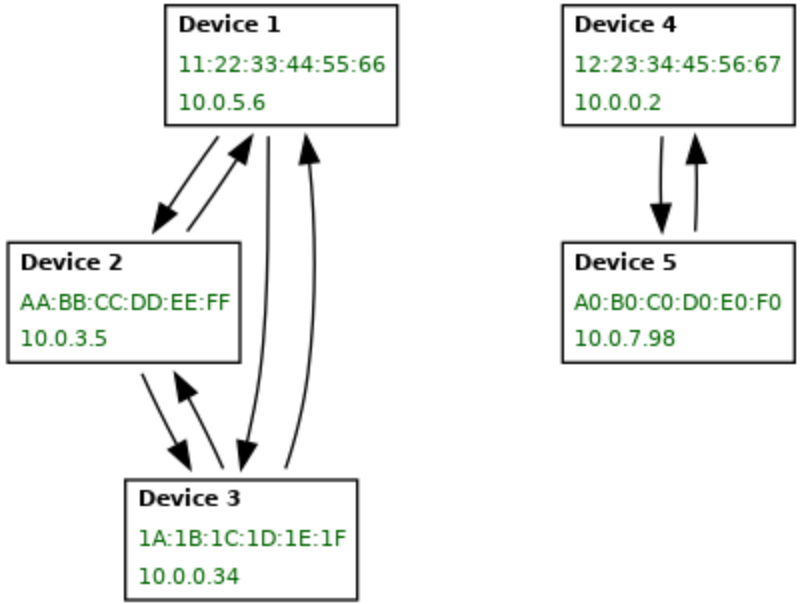
Subnet service

The subnet service creates a secure “enclosure” around each subnet by using the bridge interfaces and iptables firewall rules. It also allows connecting devices into two modes:

- Bridge
- NAT

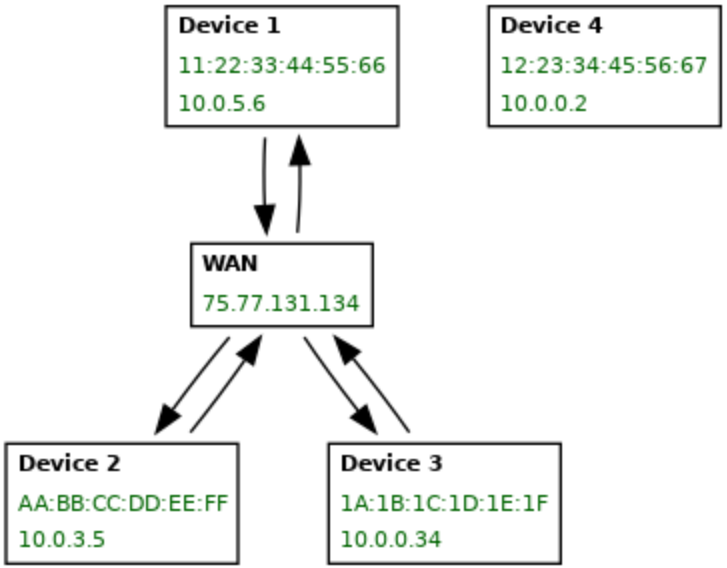
Bridge

The supervisor allows connecting two network devices into a bridge.



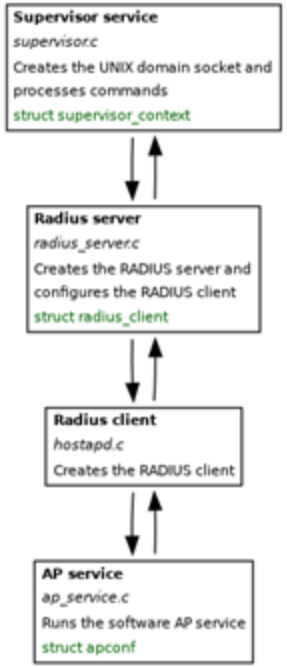
NAT

The supervisor can allow or deny network access to a network device.



Software AP service

The software AP service creates a WiFi access points for network device connection.



The Network Capture architecture

- Packet decoder
- Packet capture
- SQLite header storer
- Raw packet storer

Packet decoder

- The packet decoder extract the metadata from captured packet.
- For each decoded packet the service stores the hash of the header as well as the timestamp.

Packet capture

- The packet capture implements the actual network sniffing process.
- Currently it uses pcap library.
- It also allows interfacing with PF_RING kernel module that implements zero-copy technique.

SQLite storer

The SQLite storer implements the storage process for packet metadata into sqlite databases.

```
CREATE TABLE eth (hash INTEGER NOT NULL, timestamp INTEGER NOT NULL, ethh_hash INTEGER NOT NULL, caplen INTEGER, length INTEGER, ether_dhost TEXT, ether_shost TEXT, ether_type INTEGER,PRIMARY KEY (hash, timestamp, ethh_hash))
```

```
CREATE TABLE ip4 (hash INTEGER NOT NULL, timestamp INTEGER NOT NULL, ethh_hash INTEGER NOT NULL, caplen INTEGER, length INTEGER, ip_hl INTEGER, ip_v INTEGER, ip_tos INTEGER, ip_len INTEGER, ip_id INTEGER, ip_off INTEGER, ip_ttl INTEGER, ip_p INTEGER, ip_sum INTEGER, ip_src TEXT, ip_dst TEXT, PRIMARY KEY (hash, timestamp, ethh_hash))
```

Raw packet storer

- Stores the raw packet into pcap files and the metadata for each file is stored in a SQLite database.
- The file name for each packet is randomly generate and subsequently the name is stored in a SQLite database together with the timestamp and packet length.

Supervisor API

- **PING_SUPERVISOR** - pings the supervisor service
- **ACCEPT_MAC** - add a MAC address to the accept list
- **DENY_MAC** - add a MAC address to the deny list
- **ADD_NAT** - add NAT access to a MAC address
- **REMOVE_NAT** - remove NAT from a MAC address
- **ASSIGN_PSK** - assign a WIFI key to a MAC address
- **GET_MAC** - get the connection info for a MAC address
- **GET_ALL** - get the connection infos for all MAC addresses
- **ADD_BRIDGE** - add a bridge between two MAC addresses
- **REMOVE_BRIDGE** - removes a bridge between two MAC addresses
- **CLEAR_BRIDGE** - removes all bridges for a MAC address
- **GET_BRIDGES** - returns all assigned bridges

Router Hardware Tests

- **The EDGESec toolset was tested on the following devices:**
  - Raspberry Pi 3 B+
  - Raspberry Pi 4 B (Debian, OpenWRT)
  - PCengines apu2 platform
  - NVIDIA Jetson Nano
  - Turris Omnia (OpenWRT)
- **The compatible WiFi modems:**
  - USB Wifi Adapter for the Raspberry Pi
  - Panda Wireless PAU09 N600 Dual Band WiFi adapter
  - Compex WLE200NX 802.11a/b/g/n miniPCI express wireless card
  - Compex WLE600VX 802.11ac miniPCI express wireless card
- **The compatible hardware secure storage modules:**
  - ZYMKEY4i Raspberry Pi and Jetson Nano module