



OCPP – CAN Bus

İletişim Simülasyonu

Elektrikli Araç Şarj İstasyonlarında Güvenlik Analizi

Python

OCPP

CAN-Bus



Proje Özeti

CSMS

Merkez Sistemi

Charge Point

Şarj İstasyonu

vcan0

Sanal CAN Ağı

Proje Akışı

1

OCPP Mesajı

2

Dönüştürme

3

CAN Frame

4

Log & Analiz

⚠ MitM saldırısı ve sahte mesaj senaryoları test edilmiş, TLS ve whitelist savunmaları değerlendirilmiştir.



Kullanılan Teknolojiler

P

Python 3

Ana geliştirme dili

O

ocpp

OCPP mesajlaşma kütüphanesi

W

websockets

Bağlantı ve mesaj alışverişi

P

python-can

CAN frame işlemleri

V

vcan0

Sanal CAN arayüzü

L

Linux

Çalışma ortamı

Ağ katmanı (OCPP) + Cihaz kontrol katmanı (CAN) = Tek simülasyon



SMART Hedefler

S**Specific**

OCPP → CAN komut dönüşümü simülasyonu

**M****Measurable**

En az 2 OCPP mesajının MitM ile değiştirilmesi

**A****Achievable**

Python + vcan0 ile her ortamda çalışır yapı

**R****Relevant**

Şarj istasyonu güvenlik farkındalığı

**T****Time-bound**

Dönem içinde tamamlama





SWOT Analizi

✓ Güçlü Yönler

- İki katmanlı gerçekçi mimari
- Donanım gerektirmeyen yapı

⊗ Zayıf Yönler

- Tüm modüller modellenemiyor
- Basitleştirilmiş saldırı senaryoları

🎵 Fırsatlar

- IoT ve siber güvenlik deneyimi
- Gerçek sistemlere temel oluşturma

⚠️ Tehditler

- TLS yoksa MitM riski
- Hatalı CAN frame'leri



Örnek Kod

charge_point.py

```
import asyncio
from ocpp.v16 import ChargePoint as cp
from ocpp.v16 import call

class ChargePoint(cp):
    async def send_boot_notification(self):
        request = call.BootNotificationPayload(
            charge_point_model="VirtualCP",
            charge_point_vendor="SimVendor"
        )
        response = await self.call(request)
        print(f"Boot: {response.status}")

    async def start_transaction(self, id_tag):
        request = call.StartTransactionPayload(
            connector_id=1,
            id_tag=id_tag,
            meter_start=0,
            timestamp=datetime.utcnow().isoformat()
        )
        return await self.call(request)
```

can_bridge.py

```
import can

def ocpp_to_can(ocpp_msg):
    # OCPP mesajını CAN frame'e dönüştür
    bus = can.interface.Bus(
        channel='vcan0',
        bustype='socketcan'
    )

    if ocpp_msg['action'] == 'StartTransaction':
        arbitration_id = 0x100
        data = [0x01, 0x01, 0x00, 0x00]
    elif ocpp_msg['action'] == 'StopTransaction':
        arbitration_id = 0x101
        data = [0x01, 0x00, 0x00, 0x00]

    msg = can.Message(
        arbitration_id=arbitration_id,
        data=data, is_extended_id=False
    )
    bus.send(msg)
    print(f"CAN Sent: {hex(arbitration_id)}")
```

websockets ile OCPP bağlantısı

python-can ile CAN iletişimi

vcan0 sanal arayüz



Teşekkürler

Sorularınız için hazırım

Büşra Gül
180541037