

CAN ↔ OCPP Anomali Tespit ve Simülasyon Raporu

Hazırlayan: Enes Aydin (sizin için hazırlanmıştır) **Tarih:** 03 Kasım 2025

İçindekiler

1. Yönetici Özeti
2. Proje Kapsamı ve Hedefler
3. Metodoloji ve Test Ortamı
4. Seçilen 10 Anomali (Her biri için detaylı açıklama)
5. SWOT Analizi
6. Öneriler ve Sonraki Adımlar
7. Ek — Hızlı Komutlar & Örnek IDS Pseudo

1. Yönetici Özeti

Bu rapor, OCPP ve CAN entegrasyonuna yönelik eğitim/proof-of-concept (PoC) çalışmaları için **sadece yazılım tabanlı (vcan0 + OCPP/WebSocket)** ortamda kolayca çalıştırılabilen 10 anomali tespit senaryosunu sunar. Her senaryo için: ne olduğu, nasıl simüle edileceği, basit tespit kuralı, beklenen uyarı örneği ve hafifletme önerileri verilmiştir. Rapor ayrıca proje seviyesinde bir SWOT analizi ve uygulamaya yönelik öneriler içerir.

2. Proje Kapsamı ve Hedefler

Kapsam: Eğitim/öğrenme amaçlı, yalnızca yazılım ile tekrarlanabilir senaryolar. Gerçek donanım (USB-CAN, gerçek şarj istasyonu, araç) gerektirmez.

Hedefler: - OCPP ↔ CP (Charge Point) ile ilişkili temel saldırı tiplerini göstermek. - vcan0 üzerinde anomali senaryolarını simüle ederek basit IDS kurallarının etkinliğini test etmek. - Öğrencilerin savunma (HMAC, whitelist, replay cache) ve tespit (frekans, payload validation, correlation) yaklaşımlarını anlamasını sağlamak.

3. Metodoloji ve Test Ortamı

Gereksinimler (minimum): Linux (vcan kernel modülü), `cansend` / `candump` (can-utils), Python (tercihen `python-can`, `websocket`, `ocpp` kütüphaneleri).

vcan0 hazırlığı:

```
sudo modprobe vcan
sudo ip link add dev vcan0 type vcan
sudo ip link set up vcan0
```

Genel Test Yaklaşımı: 1. `candump vcan0` ya da `python-can` ile gerçek zamanlı sniff. 2. Her senaryo için ayrı bir `cansend` script/loop veya OCPP mock ile anomali tetiklemesi. 3. Basit Python IDS script'i ile kural tabanlı uyarı üretimi.

4. Seçilen 10 Anomali (Detaylı)

Bu bölüm her bir anomali için: (A) Tanım, (B) Simülasyon adımları (yazılım), (C) Tespit mantığı/kuralı, (D) Beklenen uyarı örneği, (E) Kısa hafifletme.

1) Beklenmeyen / Rezerv CAN ID (0x9FF) Frekans Spike

- **Tanım:** Normal trafikte görülmeyen `0x9FF` ID'sinin aniden sıklaması.
- **Simülasyon:** `for i in {1..200}; do cansend vcan0 9FF#deadbeef$i; sleep 0.05; done`
- **Tespit kuralı:** `count(0x9FF, last 10s) > 30` → alarm.
- **Örnek uyarı:** `ALERT: Unexpected ID 0x9FF freq spike – 45 frames in last 10s`.
- **Hafifletme:** Allowlist uygulama, kritik trafik izolasyonu.

2) OCPP → CAN Gecikme Artışı (Delay)

- **Tanım:** CSMS'den gelen RemoteStart ile CP'nin CAN'da Start (0x200) publish etmesi arasındaki gecikme beklenenden fazla.
- **Simülasyon:** OCPP gateway mock'ında `sleep(5)` ile gönderim geciktir.
- **Tespit:** `delay = t_CAN - t_OCPP_request ; if delay > 2s -> warn`.
- **Uyarı:** `WARN: RemoteStart->CAN(0x200) delay = 5.3s (>2.0s)`.
- **Hafifletme:** mutual TLS, gateway performans izleme, retry/fallback.

3) Payload Anomalisi — max_current (0x210) Out-of-Range

- **Tanım:** 0x210 payload'ında `max_current` alanı beklenen [0,100] aralığı dışında.
- **Simülasyon:** `cansend vcan0 210#00FF` (255 gönder).
- **Tespit:** `parse edip aralık kontrolü; if not (0 <= val <= 100) -> alert`.
- **Uyarı:** `CRITICAL: 0x210 max_current=255 (out of [0,100])`.
- **Hafifletme:** param validation, HMAC, rate-limit.

4) Frekans Dengesizliği — MeterValues (0x300) Rate Doubling

- **Tanım:** 0x300 frame'i normalde 1 Hz iken aniden 2 Hz veya daha fazlası.
- **Simülasyon:** `while true; do cansend vcan0 300#00112233; sleep 0.5; done`
- **Tespit:** `freq(ID) > expected * 1.5` → alert.

- **Uyarı:** ALERT: 0x300 rate = 2.0Hz (expected 1.0Hz).
 - **Hafifletme:** rate-limit, replay cache.
-

5) OCPP Dışı Kaynaklı CAN Komutu (Start without OCPP)

- **Tanım:** 0x200 Start mesajı OCPP RemoteStart olmadan CAN'da görülmüyor.
 - **Simülasyon:** cansend vcan0 200#01 (manually send)
 - **Tespit:** if CAN(0x200) and not recent(OCPP RemoteStart) -> alert.
 - **Uyarı:** ALERT: 0x200 Start seen but no matching OCPP RemoteStart in last 30s.
 - **Hafifletme:** gateway correlation, HMAC/signature.
-

6) Ardışık Hata Mesajları (0x301 burst)

- **Tanım:** 0x301 Error mesajlarının kısa sürede anormal artışı (DoS/hatalı modül).
 - **Simülasyon:** for i in {1..100}; do cansend vcan0 301#FF; sleep 0.02; done
 - **Tespit:** count(0x301,10s) > threshold.
 - **Uyarı:** WARN: 78 error (0x301) frames in last 10s (threshold 10).
 - **Hafifletme:** circuit breaker, blacklisting.
-

7) Aşırı Yeni WebSocket Bağlantısı (OCPP Flood)

- **Tanım:** Kısa sürede çok sayıda yeni WS bağlantısı açılması.
 - **Simülasyon:** paralel script ile 20 yeni WS bağlantısı aç.
 - **Tespit:** new_conn_rate > 10/min -> alert.
 - **Uyarı:** ALERT: 18 new WS connections in last 60s (threshold 10).
 - **Hafifletme:** conn rate-limit, IP throttling, auth zorunluluğu.
-

8) Hayalet Ölçüm Değişimi — MeterValues (0x300) Büyük Delta

- **Tanım:** MeterValues içinde kısa sürede mantıksız büyük artış/azalış.
 - **Simülasyon:** önce küçük artış sonra anı büyük değer gönder.
 - **Tespit:** if delta(kWh)/dt > implausible_threshold -> alert.
 - **Uyarı:** ALERT: MeterValues jump = +3999 kWh in 1s (implausible).
 - **Hafifletme:** cross-channel validation, HSM imzalama.
-

9) Yazılım / Firmware Kimliği Uyuşmazlığı (BootNotification)

- **Tanım:** BootNotification mesajındaki firmwareVersion beklenen whitelist dışı.
 - **Simülasyon:** OCPP mock client'ında firmwareVersion: "evil-v9" gönder.
 - **Tespit:** if firmware not in ALLOWED_LIST -> alert.
 - **Uyarı:** ALERT: BootNotification firmware "evil-v9" not in whitelist.
 - **Hafifletme:** firmware signing, secure boot.
-

10) CAN Veri Tekrarlama (Replay) — ID+Payload Duplicate

- **Tanım:** Aynı ID ve payload tekrar tekrar gönderiliyor; nonce/seq yok.
 - **Simülasyon:** belirli frame'i loop ile yeniden gönder.
 - **Tespit:** duplicate count(ID,payload, window) > N -> alert.
 - **Uyarı:** ALERT: Replay detected – same 0x200/AABBCC observed 37 times in last 20s.
 - **Hafifletme:** sequence/nonce, HMAC, replay cache.
-

5. SWOT Analizi (Özet)

Güçlü Yönler: - Tamamen yazılım tabanlı, tekrarlanabilir senaryolar. - Hem bağlantı, içerik, zaman ve replay tiplerini kapsayan geniş saldırı seti. - Eğitim amaçlı yüksek fayda.

Zayıf Yönler: - Elektriksel/timing nüansları vcan0 ile birebir eşleşmeyebilir. - Sabit eşeğe dayalı tespitler gerçek trafikte ayarlama gerektirir.

Fırsatlar: - ML tabanlı gelişim ve endüstri iş birlikleri. - Savunma (HMAC, firmware signing) prototipleme.

Tehditler: - Etik/kanuni riskler; izinsiz testler yasak. - Teknoloji hızla değişiyor, düzenli güncelleme lazım.

6. Öneriler ve Sonraki Adımlar

1. **Hızlı Pilot:** Yukarıdaki 10 senaryoyu içeren bir `runbook` (bash + python) oluştur.
 2. **IDS Baseline:** Gerçekçi normal trafik topla (ör. 1-2 saat) ve eşikleri buna göre ayarla.
 3. **ML Araştırması:** Toplanan veri ile IsolationForest ve LSTM deneyleri yap.
 4. **Savunma Prototipleri:** Replay cache, HMAC imzalama, firmware whitelist test et.
 5. **Dokümantasyon & Etik:** Testlerin her zaman izole ortamda yapılacağına dair kural ve izin dokümanları hazırla.
-

7. Ek — Hızlı Komutlar & Örnek IDS PSEUDO

vcan0 kurulum: (tekrar)

```
sudo modprobe vcan
sudo ip link add dev vcan0 type vcan
sudo ip link set up vcan0
```

Örnek cansend snippet (0x9FF spike):

```
for i in {1..200}; do cansend vcan0 9FF#deadbeef$i; sleep 0.05; done
```

Basit Replay tespiti pseudo (uygulama için referans):

```
replay_cache = deque(maxlen=5000) # store last 5000 hashes
for frame in read_vcan():
    h = (frame.id, frame.data)
    if h in replay_cache:
        alert("replay detected", frame.id, frame.data)
    else:
        replay_cache.append(h)
```

Son Not

Bu rapor, laboratuvar/sınıf ortamında hızlıca uygulanabilecek, ölçülebilir ve dokümante edilebilir test senaryoları sunar. Eğer istersen raporu PDF olarak dışarı aktarabilirim veya rapordaki herhangi bir senaryo için adım adım (örnek python IDS script'i + OCPP mock) hazırlayıp ekleyebilirim.