

OCCP ÜZERİNDEN TETİKLENEN TERMAL SALDIRI



1

HAZIRLAYAN : YAVUZ SELİM KARAAĞAÇ



1. GİRİŞ VE AMAÇ

Bu projenin amacı, Elektrikli Araç Sarj Sistemlerinde (EVCS) yaygın olarak kullanılan OCPP (Open Charge Point Protocol) ile araç içi haberleşme protokolü olan CAN-bus arasındaki "köprü" yapısının siber güvenliğini analiz etmektir. Projede, şifresiz veya zayıf korunan bir ağ trafiği üzerinden Ortadaki Adam (MitM) saldırısı gerçekleştirilecek, şarj istasyonuna (CP) gönderilen akım limitlerinin manipüle edilmesi ve bunun sonucunda araç bataryasında fiziksel hasara yol açabilecek bir "Termal Saldırı" senaryosu simüle edilmiştir.



2. SİSTEM MİMARİSİ VE KULLANILAN ARAÇLAR



Simülasyon, Ubuntu işletim sistemi üzerinde sanal bir ortamda gerçekleştirılmıştır . Sistem dört ana bileşenden oluşmaktadır:

1. CSMS (Central System): Şarj işlemini güvenli parametrelerle (80A) başlatmaya çalışan yönetim sunucusu.
2. MitM Proxy (Saldırgan): Ağ trafiğini dinleyen ve veri paketlerini anlık olarak değiştiren yazılım.
3. CP (Charge Point): OCPP komutlarını alıp bunları CAN mesajlarına (Frame) dönüştüren şarj istasyonu simülatörü.
4. BMS Monitor (Vehicle): Sanal CAN hattını (vcan0) dinleyen ve batarya güvenliğini takip eden birim.

3. SALDIRI SENARYOSU

Saldırı, "OCPP Üzerinden Tetiklenen Termal Saldırı" başlığı altında şu adımlarla gerçekleştirilmiştir:



- 1. Başlatma:** CSMS, şarj istasyonuna aracın güvenli bir şekilde şarj olması için SetChargingProfile komutu ile 80 Amper limit gönderir.
- 2. Araya Girme (Interception):** Saldırgan, MitM Proxy kullanarak bu mesajı havada yakalar. İletişim şifresiz (Plain WebSocket) olduğu için mesaj içeriği okunabilir durumdadır.
- 3. Manipülasyon (Tampering):** Saldırgan, JSON formatındaki mesajın içindeki limit: 80.0 değerini, batarya için ölümcül olabilecek limit: 300.0 değeriyile değiştirir ve istasyona iletir.
- 4. Fiziksel Etki (Bridge):** Manipüle edilmiş mesajı alan CP, bunu geçerli bir komut sanar. Kendi içindeki mantıksal dönüştürücü sayesinde bu komutu CAN ID 0x210 (Güç Kontrolü) üzerinden vcan0 hattına basar.
- 5. Sonuç:** Araç tarafından BMS (Battery Management System), fiziksel hattan gelen 300A talebini tespit eder ve sistem "Termal Kaçak" (Thermal Runaway) riskiyle alarma geçer.

4. UYGULAMA BULGULARI VE KANITLAR

Kanıt 1 (CSMS Logları): Merkezi sistemin güvenli bir şekilde limit: 80.0 gönderdiğini gösterir. Sistem saldırısından habersizdir.

```
(venv) ubuntu1@ubuntu1-VirtualBox:~/0CPP_Saldiri_Projesi$ python3 csms.py
[CSMS] Sunucu 9000 portunda hazır (Temiz Mod).
[CSMS] CP_1 baglandı. Güvenli Profil (80A) gönderilir...
[CSMS] Komut başarıyla gönderildi.
```

Kanıt 2 (MitM Proxy Logları): Saldırganın SetChargingProfile paketini yakaladığını ve içeriği 300.0 olarak değiştirdiğini kanıtlar.

```
(venv) ubuntu1@ubuntu1-VirtualBox:~/OCPP_Saldırı_Projesi$ python3 mitm_proxy.py
[SAIDLIGAN] Proxy 9001 portunda hazır (v10.4).
[SAIDLIGAN] Kurban (CP) baglandı: /CP_1

[SAIDLIGAN] SetChargingProfile yakalandı! Icerik degistiriliyor...
[SAHTE] [2,"d6b12dba-9e37-477f-b615-13d6594af092","SetChargingProfile", {"connectorId":1,"csChargingProfiles": {"chargingProfileId":1,"stackLevel":1,"chargingProfilePurpose": "TxProfile", "chargingProfileKind": "Absolute", "chargingSchedule": {"chargingRateUnit": "A", "chargingSchedulePeriod": [{"startPeriod":0,"limit":300.0}]} } } ]
```

Kanıt 3 (CP Logları): Şarj istasyonunun manipüle edilmiş 300.0 A değerini kabul ettiğini ve bunu CAN hattına (ID: 0x210) ilettiğini gösterir.

```
(venv) ubuntu1@ubuntu1-VirtualBox:~/0CPP_Saldiri_Projesi$ python3 cp_simulation.py  
[CP] SALDIRI KOMUTU ALINDI! Akım: 300.0 A  
[CP -> CAN] Fiziksel hatta 300A komutu basıldı (ID: 0x210)
```

Kanıt 4 (BMS Monitor): Aracın CAN hattından gelen veriyi okuduğunu ve "TERMAL SALDIRI TESPIT EDILDI" uyarısı vererek güvenli eşigin (100A) aşılılığını raporladığını gösterir.

```
(venv) ubuntu1@ubuntu1-VirtualBox:~/0CPP_Saldiri_Projesi$ python3 bms_monitor.py  
[BMS MONITOR] CAN hatti dinleniyor (vcan0)... Hazır!
```

```
[CAN ALINDI] ID: 0x210 | Akım Komutu: 300 A
```

```
*****
```

```
!!! UYARI: TERMAL SALDIRI TESPIT EDILDI !!!
```

```
!!! Gelen akım (300A) guvenli limiti (100A) astı !!!
```

```
*****
```

5. SONUÇ VE ÖNERİLER

Bu çalışma, şarj istasyonlarında OCPP ve CAN-bus arasındaki köprüünün (Gateway) ne kadar kritik olduğunu göstermiştir. Siber dünyada yapılan basit bir metin değişikliği (JSON manipülasyonu), fiziksel dünyada batarya yanıklarına yol açabilir.

Alınması Gereken Önlemler:

1. TLS/WSS Kullanımı: OCPP trafiği mutlaka şifrelenmeli, düz WebSocket kullanılmamalıdır.
2. Mesaj İmzalama: Kritik komutlar (şarj profili vb.) dijital olarak imzalanmalıdır.
3. Uçta Doğrulama (Gateway Filtering): Şarj istasyonu, CSMS'ten gelse bile fiziksel donanımın kapasitesini aşan komutları (örn. >100A) reddedecek bir donanım kilidine sahip olmalıdır.