

# TEKNİK RAPOR: SİMÜLASYONDA CSMS ROLÜ VE SteVe KULLANIMI

## 1. SteVe Nedir? (CSMS Simülatörü)

SteVe (Almanca: *Steckdosenverwaltung* – Priz Yönetimi), Açık Şarj Noktası Protokolü (OCPP) için geliştirilmiş, **açık kaynaklı** ve yaygın kullanılan bir sunucu uygulamasıdır. SteVe, simülasyon ve test ortamlarında **Merkezi Şarj İstasyonu Yönetim Sistemi (CSMS)** rolünü üstlenir.

Amacı, OCPP uyumlu şarj istasyonlarının (Charge Point – CS) bağlanıp yönetileceği, uzaktan komutların gönderileceği ve şarj verilerinin toplanacağı bir arka uç sistemi sağlamaktır.

Özellik	Açıklama
Rol	Merkezi Yönetim Sistemi (CSMS)
Teknoloji	Java tabanlı bir web uygulaması (MySQL/MariaDB veritabanı gerektirir).
Kullanım Yeri	Şarj altyapısı geliştirme, test, akademik araştırma ve küçük ölçekli yönetim sistemleri.
Erişim	Bir web tarayıcısı üzerinden grafik kullanıcı arayüzü (Web UI).

## 2. Simülasyon Senaryosundaki Kritik Rolü

SteVe, "**Sahte Firmware Güncellemesi ile Şarj İstasyonunu Ele Geçirme**" anomali senaryonuzun üç temel aşamasında kritik bir rol oynar:

### A. Meşru Hedefin Temsili (Spoofing Hedefi)

- Saldırgan (ATK), kendisini meşru bir **CSMS** olan SteVe gibi göstermeye çalışacaktır.
- Senaryonun dayandığı **UpdateFirmware** komutu normal şartlar altında SteVe'den gelmelidir. SteVe, bu komutu gönderebilme yeteneği ile saldırı vektörünün bağlamını oluşturur.

### B. Oportünite ve Komut Merkezi

- SteVe'nin yönetim arayüzü, meşru bir firmware güncelleme komutunu ("Saldırı Vektörü ve Adımları" raporunuzdaki meşru süreç) tetiklemenizi sağlar.
- Saldırgan (ATK) makinesi, bu meşru trafiği yakalayıp sahte (zararlı) firmware indirme URL'si ile değiştirerek MitM saldırısını gerçekleştirebilir.

### C. Anomali Tespiti ve Hizmet Reddi (DoS) Doğrulaması

- Saldırı başarılı olup Charge Point (CS) fidye yazılımı nedeniyle kilitlendiğinde, SteVe'ye (CSMS) düzenli olarak gönderilmesi gereken **Heartbeat** mesajları kesilir.
- SteVe'nin web arayüzünde, hedef CS'in durumunun aniden **"Offline"** veya **"Error"** olarak değiştiğinin gözlemlenmesi, raporunuzdaki **"Tespit Yöntemleri"** maddesini ve **Hizmet Reddi (DoS)** etkisini merkezi olarak doğrulamanızı sağlar.

---

## 3. Kullanım ve Kurulum Gereksinimleri

SteVe bir web uygulaması olduğu için, simülasyon ortamında bir sunucu görevi görür.

### A. Kullanım Şekli (Ağ İletişimi)

1. **Kurulum Yeri:** Simülasyon topolojisindeki **CSMS** rolünü üstlenen sanal makineye (VM) veya **Docker konteynerine** kurulur.
2. **Erişim:** SteVe çalıştırıldıktan sonra, kurulumun yapıldığı sanal makinenin IP adresi ve portu kullanılarak **herhangi bir web tarayıcısı** üzerinden erişilir (örneğin, [http://<CSMS\\_IP>:8080/](http://<CSMS_IP>:8080/)).
3. **İletişim:** Şarj İstasyonu (CS) makinesi, OCPP protokolü (WebSocket) üzerinden sürekli olarak SteVe sunucusuna bağlanır.

### B. Önerilen Kurulum Metodu

- **Docker:** SteVe ve veritabanı bağımlılıklarını izole edilmiş, hızlı ve temiz bir şekilde kurmak için en çok önerilen yöntem **Docker konteynerizasyonudur**.

# SİMÜLASYON ORTAMI KURULUM KILAVUZU

## A. Simülasyon Ortamı Temel Bileşenleri

Bu ortam, herhangi bir OCPP senaryosunu (firmware güncelleme, uzaktan şarj başlatma/durdurma, veri aktarımı vb.) test etmeye olanak tanır.

Bileşen Adı	Rolü	Teknoloji	Amaç
<b>CSMS (SteVe)</b>	Merkezi Şarj İstasyonu Yönetim Sistemi (Sunucu)	Docker/Java	OCPP komutlarını gönderir ve verileri yönetir.
<b>Charge Point (CS)</b>	Şarj İstasyonu Simülatörü (İstemci)	Docker/Python	CSMS'e bağlanır ve komutları uygular.
<b>Ağ İzleme (İsteğe Bağlı)</b>	Ağ Trafik Analizi	Sanal Makine/Wireshark	İki bileşen arasındaki meşru iletişimi izlemek.

## B. Adım 1: Kurulum Ön Gereksinimleri

Tüm ekip üyelerinin test ortamını çalıştırması için gerekli temel yazılımlar:

- Docker ve Docker Compose:** Sunucu ve istemci bileşenlerini hızlı ve izole bir şekilde kurmak için.
- Temel Linux/Windows Bilgisi:** Komut satırı (CLI) üzerinden Docker komutlarını çalıştırma yeteneği.

## C. Adım 2: CSMS Kurulumu (SteVe)

SteVe'yi, bağımlılıklarından izole edilmiş bir şekilde Docker üzerinde kuralım:

- docker-compose.yml Hazırlama:** SteVe uygulamasını ve verilerini saklayacak bir MySQL/MariaDB veritabanını içeren bir Docker Compose dosyası hazırlayın.

**Kurulumu Başlatma:** Proje dizininde aşağıdaki komutu çalıştırın:

Bash

`docker-compose up -d`

2.

**Erişim Kontrolü:** SteVe'nin çalışıp çalışmadığını kontrol etmek için tarayıcınızdan şu adrese erişin:

`http://localhost:8080/`

3. (Eğer SteVe'yi ayrı bir VM'e kurduysanız, *localhost* yerine o VM'in IP adresi kullanılmalıdır.)

## D. Adım 3: Charge Point (CS) Kurulumu

Bir şarj istasyonu istemcisini SteVe'ye bağlamamız gerekir. Açık kaynaklı bir OCPP Python istemcisi (örneğin *ocpp-simulator*) bu amaç için idealdir.

1. **İstemci İmajını Çekme:** OCPP istemci imajını Docker Hub'dan çekin.

**Bağlantı Konfigürasyonu:** İstemciyi çalıştırırken, CSMS (SteVe) sunucusunun adresini belirtin:

Bash

# Örnek çalıştırma komutu (detaylar kullanılan simülatöre göre değişir)

`docker run -d --name chargepoint1 ocpp-simulator:latest \`

`--csms-url ws://<CSMS_IP>:8887/ChargePoint1`

2.

3. **Bağlantı Kontrolü:** SteVe web arayüzüne girerek, "ChargePoint1" ID'si ile yeni bir şarj istasyonunun **"Connected"** (Bağlı) durumda görüldüğünü doğrulayın.

## E. Adım 4: Genel Test ve Senaryo Uygulama

Artık kurulan bu genel ortam, temel OCPP senaryolarını test etmeye hazırdır:

- **Veri Gönderimi:** CS'in SteVe'ye Heartbeat ve MeterValues (ölçüm değerleri) gönderdiğini doğrulayın.
- **Komut Gönderimi:** SteVe arayüzünden CS'e uzaktan komutlar (örneğin, **RemoteStartTransaction**, **GetConfiguration**) gönderip CS'in doğru yanıt verdiğini test edin.
- **Güvenlik Senaryoları:** Ek araçlar (Wireshark, MitM araçları) kullanarak bu iki bileşen arasındaki trafiği inceleyebilir ve güvenlik senaryolarınızı uygulayabilirsiniz.

Bu genel kılavuz, tüm ekibinizin başlangıç için kullanabileceği standart bir OCPP test ortamı oluşturur.