Simülasyon Ortamı Gereksinimleri Raporu

Senaryo: OCPP Mesaj Yoğunluğu / DoS Hazırlığı

Amaç: CSMS (Central System Management System) hedefli yüksek frekanslı OCPP mesaj saldırılarını gerçekçi şekilde emüle edip tespit, müdahale ve geri kazanım yeteneklerini test etmek. Bu rapor simülasyon ortamında bulunması gereken tüm bileşenleri, her bileşen için önerilen konfigürasyon değişikliklerini, test vakalarını ve kabul kriterlerini içerir.

1. Özet

- Simülasyonun hedefi: Mesaj spiki / DoS tarzı anomali senaryolarının CSMS üzerindeki etkisini ölçmek, tespit/mitigasyon mekanizmalarını değerlendirmek ve operasyonel eylem playbookları geliştirmektir.
- Başarı ölçütleri: Tespit doğruluğu (recall) ≥ %95, müdahale süresi ≤ 30 s, kritik operasyonların erişilebilirliği ≥ %99.

---

2. Ana gereksinimler — bileşen listesi

Aşağıdaki bileşenler simülasyon ortamında mutlaka bulunmalıdır.

2.1. CSMS (Test hedefi)

- Tam OCPP (1.6 ve/veya 2.0) sunucu implementasyonu (WebSocket/JSON).
- Gelen mesajları işleyen, kimlik doğrulama / yetkilendirme kontrolleri bulunan gerçek veya test sürümü.
- Configuration: per-CP rate-limit, connection limits, payload sanity checks, prioritization.

2.2. CP (Charge Point) Emülatörleri

- İki grup: "normal" CP emülatörleri (gerçekçi trafik) ve "atakçı" CP emülatörleri (yüksek mesaj frekansı, tekrarlı mesajlar).
- Skalabilite: 100–200+ eş zamanlı emülatör çalıştırabilmeli.
- Uygulama: Python (asyncio + websockets) veya Node.js tabanlı scriptler.

2.3. Saldırı / Yük Üreteci

- Spike ve uzun süreli yük profilleri oluşturabilen araçlar (locust, custom Python scripts, veya dağıtık docker/k8s agentleri).
- IP/clientId rotasyon yeteneği; farklı payload pattern'leri üretebilme.

2.4. Ağ Emülasyonu

- tc/netem, Mininet veya benzeri ile latency, packet loss ve bandwidth limitasyonu uygulanabilmeli.

2.5. Anomali Tespit Sistemi

- Gerçek zamanlı stream processing (Kafka, Flink, Spark Streaming veya lightweight custom pipeline).
- Hem rule-based (eşikler) hem ML tabanlı (Isolation Forest, EWMA, LSTM gibi) modeller.

2.6. Gerçek Zamanlı Müdahale / Orkestrasyon

- Rate-limiter, connection drop, IP/CP karantina, trafik şekillendirme.
- Otomasyon playbook: detection -> validation -> mitigation (throttle/block/alert).

2.7. Observability & Logging

- Metrikler: Prometheus; dashboard: Grafana.
- Merkezi log: ELK/EFK — raw message saklama (attack esnasında full capture).

2.8. SIEM & Forensics Deposu

- Olay kayıtları, raw mesajlar, IP ve CP id bilgileri ile saklama ve hızlı arama.

2.9. Orkestrasyon & Ölçekleme

- Docker/Docker-Compose veya Kubernetes manifestleri (emülatörlerin kolay ölçeklenmesi için).
- Time sync (NTP) ve CI pipeline (opsiyonel: test senaryolarını otomatik çalıştırmak için).

---

3. Konfigürasyon değişiklikleri (önerilen parametreler)

Not: Tüm eşikler ortamın gerçek baseline ölçümlerine göre ayarlanmalıdır; aşağıdaki değerler başlangıç önerileridir.

3.1. CSMS — Ağ & Protokol

- **Per-CP rate limit:** varsayılan *5 mesaj/s* (1s window) — baseline'a göre düşür veya yükselt.
- **Connection limit (per IP/CP):** 2–4 eşzamanlı bağlantı.
- **Keepalive / Heartbeat toleransı:** normal 60s ise saldırı testlerinde 1s deneyin; CSMS yüksek-frequency heartbeats için throttle uygulamalı.
- **Payload sanity checks:** ardışık aynı payload sayısı eşiği (örn. 10 kez tekrarlayan mesaj -> flag).
- **Prioritization:** authorizations/stop işlemlerine öncelik ver, heartbeat gibi düşük önceliği kuyrukla.

3.2. Rate limiting politikası & yaptırım kademesi

- **Kademeli yaptırım:** 1) throttle (kısa süre), 2) geçici block (5–30dk), 3) uzun dönem block.
- **Burst tolerance:** kısa süreli ani artışlara izin: burst factor = 2x normal.

3.3. Detection pipeline

- **Gözlem pencereleri:** 1s (anlık), 5s (kısa trend), 60s (uzun trend).
- **Eşik formülü (örnek):** alarm if messages/sec > max(absolute_threshold, baseline_mean + 5*baseline_std).
- **Confidence threshold:** ilk alarm için ≥0.8; kritik müdahale için ≥0.95.

3.4. Logging & retention

- **Attack raw message retention:** en az 30 gün.
- **Metric export frekansı:** 1s veya 5s per metric.

3.5. Güvenlik sertifikasyonu

- **mTLS / client cert validation:** CP kimlik doğrulaması zorunlu (test ortamında opsiyonel, ama gerçekçi testler için açık olmalı).

---

4. Örnek test vakaları (priority order)

Aşağıdaki test vakalarını sırayla uygulayın; her test için beklenen davranışı ve kabul kriterlerini kaydedin.

Test 1 — Baseline trafik toplama

- **Konfig:** 100 normal CP, her biri 0.2–1 msg/s; süre: 1 saat.
- **Çıktı:** baseline_mean, baseline_std, normal message patternleri.

Test 2 — Tek CP yüksek frekans spiki

- **Konfig:** 1 ele geçirilmiş CP -> 100–500 msg/s, süre: 60–300s.
- **Beklenen:** detection recall ≥ %95, throttle/block uygulanmalı ≤ 30s.

Test 3 — Dağıtık (botnet benzeri) düşük yoğunluklu DDoS

- **Konfig:** 200 CP; 50 koordineli CP 5–20 msg/s.
- **Beklenen:** aggregate yük altında kritik işlemler çalışmaya devam etmeli; anomali tespit ve karantina.

Test 4 — IP/clientId rotasyonu ile sahte istemci saldırısı

- **Konfig:** saldırgan farklı IP/clientId kullanarak 1000 kaynakla 1–10 msg/s.
- **Beklenen:** credential checks, payload signature ve davranış analizi ile sahteciler tespit edilmeli.

Test 5 — Ağ bozulması altında saldırı

- **Konfig:** Test 2 veya 3 ile birlikte latency 200–500ms ve packet loss %5–10.
- **Beklenen:** retry pattern analizleri, CSMS timeout davranışları, detection doğruluğu.

---

5. Ölçülebilir metrikler ve kabul kriterleri

- **Tespit (recall):** ≥ %95 (saldırı olayları için).
- **Tespit (precision):** tercihen ≥ %90.
- **Müdahale süresi:** detection → uygulanan mitigation ≤ 30 s.
- **Kritik fonksiyon devamlılığı:** Authorize/StopTransaction gibi işlemlerde erişilebilirlik ≥ %99.
- **Sistem kaynak limitleri:** saldırı sırasında mitigasyon uygulandıktan sonra CPU < %80 hedefi.
- **False positive oranı:** normal trafik için alarm oranı < %1.

---

6. Operasyonel tavsiyeler ve playbook

- **Kademeli response:** yanlış pozitifleri azaltmak için önce throttle, sonra temp block, sonra kalıcı block.
- **Otomasyon:** detection pipeline'dan çıkan alarmı manuel onaya gerek kalmadan ilk seviyede otomatik throttle uygulayacak şekilde konfigüre edin; kritik/şüpheli durumlarda insan inisiyatifi ile devam edin.
- **Forensics:** kurulum sırasında raw message capture açık tutulmalı; kritik olaylarda loglar ve mesajlar arşivlenmeli.
- **Test rutini:** haftalık/aylık otomatik tatbikatlar (farklı pattern/rotasyonlarla).

---

7. Hızlı uygulama checklisti

1. Baseline veri toplama (1–7 gün).
2. CSMS üzerinde per-CP rate limit ve connection limit uygula.
3. Detection pipeline kur (1s/5s/60s pencereleri).
4. Real-time mitigation (throttle, block) otomasyonu ekle.
5. Observability: Prometheus + Grafana + ELK.
6. Test senaryolarını sırayla çalıştır, metrikleri belgeleyin.
7. Threshold'ları ayarlayın, false-positive testleri yapın.
8. Sonuç raporu: detection recall/precision, müdahale süreleri ve kaynak kullanımı.