Container Host/Image/Runtime Güvenlik Checklist Şablonu — Yazılım / Sunucu ve Güvenlik (v1.0)

Asset Amaç: Bu asset, container (Docker/Kubernetes) güvenliğini Host OS, Image Build ve Runtime Katmanlarında uçtan uca standartlaştırmak için hazırlanmıştır. Onaylı base image kullanımı, CI/CD zafiyet taramaları (vulnerability scanning), hassas veri (secret) yönetimi ve çalışma zamanı kaynak limit kontrollerini tek bir operasyonel listede toplar. Hem olası güvenlik ihlallerini hem de kaynak tükenmesine bağlı stabilite sorunlarını üretim ortamına geçmeden önce yakalayan sürdürülebilir bir kontrol mekanizması kurar.

Kim Kullanır?: DevOps / SRE Ekipleri, Platform Mühendisliği Birimleri ve Ajans Teknik Ekipleri (Otel / B2B altyapı yöneticileri).

Nasıl Kullanılır?

1. Host, Image ve Runtime kategorilerinde mevcut altyapı durumunuzu kontrol ederek eksik maddeleri işaretleyin.
2. İşaretlenen eksik ve kritik (kırmızı) güvenlik maddelerini 14 günlük uygulama sprint planına aktarın.
3. KPI paneli üzerinden zafiyet tarama kapsamasını, açık kalma sürelerini ve OOM/CPU spike trendlerini anlık olarak takip edin.

CHECKLIST & SPRINT PLAN — Container Host/Image/Runtime Güvenlik Şablonu

A) Ölçüm & Önceliklendirme Kontrol Listesi (Checklist)

- [ ] **Host OS Hardening & Patch Rutini Var:** Container çalıştıran ana sunucunun (Host OS) çekirdek güncellemeleri ve güvenlik yamaları otomatik/periyodik olarak yapılıyor.
- [ ] **Docker Daemon Erişimi Kısıtlı:** Docker socket (/var/run/docker.sock) yetkisiz kullanıcılar ve container'lar için kesinlikle erişime kapalı.
- [ ] **Rootless / Non-Root Default & Privileged İstisnaları Kayıtlı:** Container'lar varsayılan olarak root yetkisi olmadan çalışıyor; privileged yetkili konteynerlar kayıt altında ve kısıtlı.
- [ ] **Approved Base Image Listesi Oluşturuldu:** Yalnızca onaylı, resmi veya minimal (Alpine/Distroless) base image'ların kullanımına izin veriliyor.
- [ ] **CI'da Image Vulnerability Scan Zorunlu:** CI/CD pipeline süreçlerinde zafiyet taraması (Trivy/Grype vb.) zorunlu geçit (gate) olarak tanımlı.
- [ ] **Registry'de Periyodik Tarama Var:** Container Image Registry (ECR/Harbor/Docker Hub) üzerindeki imajlar düzenli olarak taranıyor.
- [ ] **Secrets Image İçine Gömülmüyor (Runtime Secret Store):** Şifreler, API key'ler imaj katmanlarına eklenmiyor; çalışma zamanında (Vault/AWS Secrets Manager) dinamik enjekte ediliyor.
- [ ] **CPU / Memory Limit Tanımlı:** Her bir container için kaynak kullanım üst sınırları (requests/limits) belirlendi.

- [ ] **Network Policy Default-Deny + İzin Listesi Var:** Container'lar arası ağ trafiğinde varsayılan olarak tüm erişimler engelli, sadece gerekli servisler izinli.
- [ ] **OOM / CPU Spike / Anomali Dashboard ve Alert var:** Bellek yetersizliği (Out of Memory), ani CPU artışları ve anomali durumları için anlık izleme ve alarm kuralları aktif.

B) Problem $\rightarrow$ Kök Neden $\rightarrow$ Çözüm Tablosu

Problem	Kök Neden	Çözüm
CVE'li image prod'da çalışıyor	Taramaların yalnızca bildirim amaçlı kalması, otomatik yeniden canlıya alma (redeploy) yapılmaması.	Scan → Rebuild → Rollout otomasyon döngüsünün kurulması.
Host'a sıçrama riski (Container Escape)	Container'ların root kullanıcısı veya privileged modda çalıştırılması.	Rootless kullanım + Linux kernel capability kısıtlamaları.
Sistemde ani kesintiler yaşanıyor	Container düzeyinde CPU/Memory limitlerinin tanımlanmaması.	Metrik bazlı CPU/Memory limitleri + otomatik alarm kurgusu.
Hassas veri sızıntısı (Secret Leak)	Veritabanı ve API şifrelerinin Dockerfile/Image içine gömülmesi.	Dynamic Secret Manager entegrasyonu + build-time log masking.

C) 14 Günlük Container Güvenliği Sprint Planı

Hafta 1: Host, Image Hardening ve CI Gates

- **Gün 1 (Host Baseline Envanteri):** Container host sunucularının listelenmesi, OS yama durumlarının çıkarılması ve hardening planı.
- **Gün 2 (Docker Daemon Güvenliği):** Docker socket erişim haklarının kısıtlanması, TLS sertifikalı uzaktan erişim kurgusu.
- **Gün 3 (Rootless & Privilege Kısıtı):** Tüm Dockerfile'larda USER nonroot standardının getirilmesi ve yetkili istisnaların belgelenmesi.
- **Gün 4 (Approved Base Image Listesi):** Kurum içi onaylı imaj kütüphanesinin (Distroless/Alpine tabanlı) oluşturulması.
- **Gün 5 (CI Scan Gate Ekleme):** GitHub Actions / GitLab CI hatlarına Trivy/Grype adımı eklenerek yüksek riskli imajların derlenmesinin engellenmesi.

- **Gün 6 (Registry Tarama & Alarm):** Image Registry üzerinde günlük zafiyet taramalarının başlatılması ve Slack/Teams entegrasyonu.
- **Gün 7 (Secrets Taşıma):** Dockerfile ve ortam değişkenlerindeki hardcoded şifrelerin temizlenerek Vault / Cloud Secret Manager'a aktarılması.

Hafta 2: Runtime Isolation, Observability ve Rollout

- **Gün 8 (Resource Limit Profilleri):** Servis yüklerine göre RAM/CPU limits & requests değerlerinin belirlenip manifestlere işlenmesi.
- **Gün 9 (Network Policy Default-Deny):** Container ağında tüm açık trafiğin kapatılarak default-deny kuralının uygulanması.
- **Gün 10 (Kritik Servis Policy'leri):** Ödeme geçidi, rezervasyon motoru ve PMS mikroservisleri için mikro-segmentasyon kurallarının yazılması.
- **Gün 11 (Observability Dashboard):** Prometheus / Grafana üzerinden OOM Kills, CPU Throttling ve Container restarts metrik panellerinin kurulması.
- **Gün 12 (Alert Eşikleri ve Runbook):** OOM ve CPU spike durumlarında nöbetçi DevOps mühendisine düşecek alarm ve çözüm adımlarının tanımlanması.
- **Gün 13 (Rebuild / Rollout Takvimi):** Zafiyet tespiti sonrası otomatik imaj güncelleme ve kesintisiz (rolling update) canlıya alma takvimi.
- **Gün 14 (Rapor & İyileştirme Backlog'u):** Güvenlik durum raporunun oluşturulması ve geleceğe dönük teknik borçların backlog'a eklenmesi.

D) Öncesi / Sonrası Durum KPI Tablosu

KPI Metriği	**Yöntem Öncesi (Mevcut Durum)**	**Yöntem Sonrası (Standart)**	**Hedef / Trend**
Scan Kapsaması (%)	Belirsiz / Manuel ($\%15$)	$\%100$ (Tüm Img & Reg)	**Hedef** ↑ (Tam kapsama)
Kritik CVE Açık Süresi	30+ Gün	< 24 Saat	**Hedef** ↓(Anında yama)
OOM / CPU Spike Sayısı	Haftalık 10+ Kesinti	0 (Stabil Kaynak)	**Hedef** ↓ (Sıfır kesinti)
Privileged Container Sayısı	Kısıtlanmamış	0 (Sadece İstisnalar)	**Hedef** ↓ (Tam izolasyon)

Deliverables (Teslim Edilecek Çıktılar)

- **Host Hardening & Container Baseline Rehberi:** İşletim sistemi ve daemon seviyesinde güvenlik sertleştirme standartları.
- **Approved Base Image & Secure Dockerfile Kütüphanesi:** Güvenli ve minimal konteyner imaj şablonları.
- **CI Scan Gate & Redeploy Otomasyon Pipeline'ı:** Zafiyetli imajları engelleyen ve yamalı sürümleri canlıya alan CI/CD yapılandırması.
- **Container Network Policy & Runtime Isolation Seti:** Mikro-segmentasyon ve varsayılan engelleme kuralları.
- **Container Observability KPI Dashboard & Alert Runbook:** Grafana izleme panelleri ve olay müdahale kılavuzu.

[Checklist Card]

> *DevOps mühendisleri, SRE liderleri ve platform mimarları için tasarlanmış, 1200x1200px (1:1) kare formatında yüksek çözünürlüklü bir Container Güvenlik Kontrol Kartı. Koyu karbon gri ve siber mavi/yeşil tonlarındaki teknolojik zemin üzerinde; Docker/Kubernetes koruma kalkanı, imaj tarama büyüteci ve çalışma zamanı kaynak izleme grafikleri yer alıyor. Kartın üzerinde; "Host OS & Daemon Hardening", "CI/CD Image Vulnerability Scanning", "Non-Root & Resource Limits (CPU/RAM)" ve "Default-Deny Network Micro-segmentation" kontrol maddeleri, yanlarında parlak onay işaretleriyle scannable ve kurumsal bir checklist formatında sergileniyor.*