Hafta 8 — Ara Sınav Haftası: Quiz-1

Tarih	31.10–08.11.2026
Öğrenme çıktıları	ÖÇ.1, 2, 3
Süre	3 saat

Bu hafta ne oluyor?

Ara sınav haftasında **Quiz-1** yapılır. Kapsam **1–6. haftaların** konularıdır; Quiz-1 vize notunun **%40'ıdır** (`Not_Vize = 0,6·RAP1 + 0,4·QUIZ1`). Tarih, saat, yer, süre ve soru biçimi ders sınıfında duyurulur. Bu sayfa bir **çalışma rehberidir**: konu haritası, sık karıştırılan kavramlar, bir haftalık çalışma planı ve cevaplı örnek sorular.

En verimli çalışma yolu

Her haftanın sayfasının sonundaki "**Kendini sınama**" sorularını cevaplara bakmadan cevaplayın, sonra kontrol edin. Demoları bir kez daha çalıştırıp her adımda "ne oldu, neden oldu, nasıl düzeltildi?" sorusunu kendinize sorun. Ezber değil, **neden** sorusu ölçülür.

1. Konu haritası: 1–6. haftalar

Hafta	Ana kavramlar	Demolar	Kitap
1 Giriş ve koruma planı	CIA; varlık–tehdit–zafiyet–risk; saldırgan modelleri (beyaz kutu); Saltzer–Schroeder ilkeleri; yedi koruma katmanı; koruma planının 7 adımı; arayüz ve varlık tabloları (C/I/I+); STRIDE ve VAD; saldırı ağacı; risk = olasılık × etki; güvenli başlatma; bellek yönetimi hataları; güvenli silme; bölümleme; sürüm kimliği	PATH ile kandırma · bellekte kalan parola · taşmayla yetki yükseltme · işaretli uzunluk	1.1–1.9, 3.3–3.5, 12.1, 13.2–13.3
2 Zararlı yazılım ve modeller	Virüs türleri, solucan, truva atı; salgın hızı; kural tabanlı tespit; bütünlük izleme; Bell–LaPadula, Biba, Clark–Wilson; Unix ve Windows erişim denetimi; RBAC; CWE, OWASP Top 10, CVE, CVSS; denetim kaydı ve günlük enjeksiyonu	Kural motoru · bütünlük izleme · salgın · günlük enjeksiyonu · kurcalamaya dayanıklı günlük · ACE · umask · RBAC	2.1–2.2, 13.11
3 Veri güvenliği	Verinin üç hali; simetrik/asimetrik/hibrit; AEAD; özet/MAC/imza; CSPRNG ve modulo sapması; nonce, IV, tuz; ECB; PBKDF2/Argon2id; HKDF, ileri gizlilik; anahtar yaşam döngüsü, hiyerarşi, zarflama; TLS 1.3, doğrulama, sabitleme, fail-open; maskeleme, tokenizasyon, takma ad; güvenlik kabukları	AES-GCM · nonce tekrarı · ECB · tur sayısı · HKDF · MITM + sabitleme · SQLite şifreleme · mlock · dört kabuk	4.9–4.13, 9.1–9.3, 10.7–10.9, 11.1–11.11, 13.2
4 C/C++ sağlamlaştırma	SEI CERT; girdi doğrulama ilkeleri; biçim dizisi; UAF; tamsayı taşması ve tanımsız davranış; hata işleme, sinyaller; statik analiz; sanitizer'lar; fuzzing; kanarya, FORTIFY, ASLR, NX, RELRO, CFI; kod gizlemeye giriş; düzleştirme	Biçim dizisi · UAF · UBSan · fuzzing · korumalar · sembol/dize · düzleştirme	3.1–3.5, 12.1, 12.3, 12.8, 12.11, 13.1, 13.4–13.5
5 Java ve yorumlanan diller	Yönetilen dil neyi çözer; SEI CERT Java; enjeksiyonun ortak kökü; SQL, komut, yol geçişi; seri durumdan çıkarma; XXE/XSS; Python/JS, ReDoS; bayt kodu; ProGuard/R8, `-keep`; dize gizleme, yansıma; SBOM, VEX	SQL · komut · yol · bayt kodu · gizleme · SBOM · seri durumdan çıkarma	3.7, 3.10, 3.11, 1.7
6 RASP	Algılama–savunma–caydırma; MATE; RASP mimarisi; bütünlük denetimi; hata ayıklayıcı, ortam, kanca algılama; dinamik bellek koruması; root ve imza doğrulama; kontrol akışı sayacı; tepki politikası, decoy, cihaz bağlama; sınırlar	Bütünlük · anti-debug · VM · LD_PRELOAD · akış sayacı · imza · root · RASP motoru	12.2, 12.12–12.13 (kavram)

Quiz-1 kapsamı (1–6. haftalar)

ezber değil GEREKÇE sorulur: 'neden bu kip?', 'neyi korumaz?'

Hafta 1	CIA, saldırgan modeli, STRIDE, saldırı ağacı
Hafta 2	zararlı yazılım, BLP/Biba, CWE/CVE/CVSS
Hafta 3	verinin üç hâli, AEAD, nonce, KDF, TLS
Hafta 4	C/C++ hataları, sanitizer, derleyici korumaları
Hafta 5	enjeksiyon, deserialization, ProGuard, SBOM
Hafta 6	RASP: algıla / savun / caydır

Her haftanın 'Kendini sınama' bölümünü cevaba bakmadan çözün.

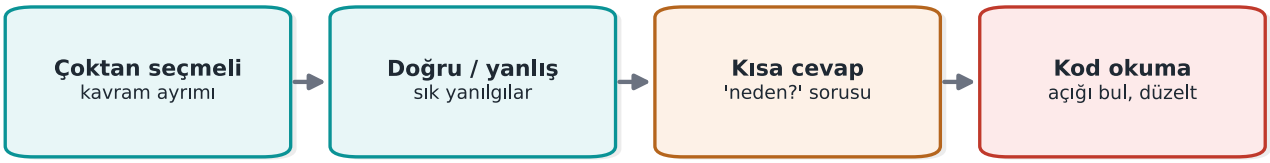
2. Sık karıştırılan kavramlar

Kavram A	Kavram B	Fark
Hata (bug)	Zafiyet	Her zafiyet bir hatadır; zafiyet, bir saldırganın **kullanabildiği** hatadır
Tehdit	Risk	Tehdit olası zarar kaynağıdır; risk, olasılık × etkidir
Kodlama (Base64)	Şifreleme	Kodlama anahtarsız ve geri çevrilebilir, gizlilik sağlamaz
Özet (SHA-256)	MAC (HMAC)	Özet anahtarsızdır, herkes hesaplar; MAC gizli anahtar ister, kimlik doğrular
MAC	Dijital imza	MAC simetrik (iki taraf aynı anahtar), imza asimetrik (inkâr edilemezlik)
Nonce	Tuz	Nonce bir anahtarla **asla tekrar etmemeli**; tuz parola başına benzersiz, gizli değil
IV (CBC)	Nonce (GCM)	CBC IV'ü **tahmin edilemez** olmalı; GCM nonce'u **benzersiz** olmalı
PBKDF2 / Argon2id	HKDF	İlki düşük entropili **parola** için (yavaş); HKDF yüksek entropili **sırdan** türetme (hızlı)
Zincir doğrulama	Ana makine adı denetimi	Biri "güvenilir CA imzalamış mı?", diğeri "bu sertifika **bu sunucu** için mi?"
Sabitleme	TOFU	Sabitlemede beklenen anahtar önceden gömülü; TOFU'da ilk bağlantıda öğrenilir
Fail-open	Fail-closed	Hata durumunda "geçti" vs "reddet"; güvenlik denetimleri fail-closed olur
Kara liste	Beyaz liste	Yasakları saymak her zaman eksik kalır; izin verilenleri tanımlamak doğrudur
Kanarya	ASLR	Kanarya taşmayı fonksiyon dönerken fark eder; ASLR adresleri rastgeleleştirir
ASan	UBSan	ASan bellek erişim hataları; UBSan tanımsız davranış (taşma, kaydırma)
Statik analiz	Dinamik analiz / fuzzing	Statik çalıştırmadan, yanlış alarm verebilir; dinamik yalnız çalışan yolu görür ama kesindir
İşaretsiz sarma	İşaretli taşma	Sarma tanımlıdır (mantık hatası); işaretli taşma **tanımsız davranıştır**
Parametreli sorgu	Kaçış karakteri	Parametre iki kanal, asıl çözüm; kaçış ikinci savunma hattı
ProGuard gizleme	Dize gizleme	ProGuard **adları** değiştirir, dizgeleri gizlemez
Gizleme	Güvenli kodlama	Gizleme geciktirir; hatayı düzeltmez
Algılama	Tepki	Algılama tek başına işe yaramaz; tepki tetikleyiciden uzak ve örtük olmalı

Kavram A	Kavram B	Fark
Maskeleme	Tokenizasyon	Maskeleme gösterimde gizler; tokenizasyon değeri kasadaki gerçek değere bağlı belirteçle değiştirir
Takma adlandırma	Anonimleştirme	Takma adlı veri hâlâ kişisel veridir; anonim veri kişiye bağlanamaz

Quiz-1 soru tipleri

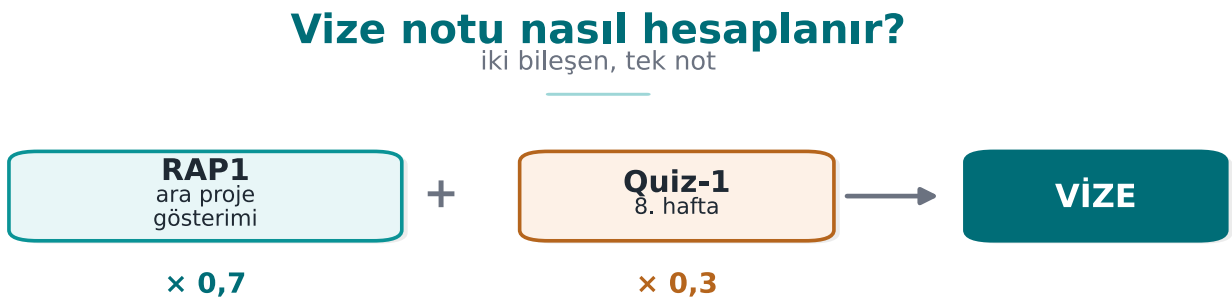
dördü de 'anladın mı' diye sorar, 'ezberledin mi' diye değil



Kod okuma sorusunda önce açığın SINIFINI (CWE) söyleyin, sonra düzeltmeyi.

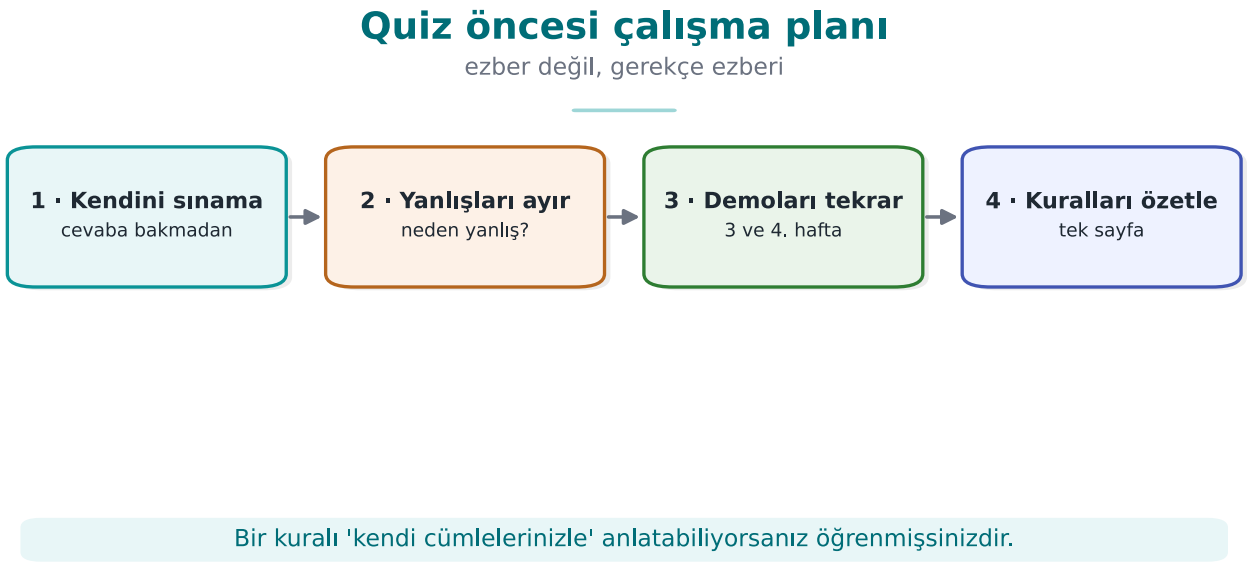
3. Bir haftalık çalışma planı

Gün	Konu	Ne yapılacak?
1	1. hafta	STRIDE, saldırı ağacı, koruma planı; Demo 1–4'ü yeniden çalıştır; kendini sınama 1–25
2	2. hafta	Modeller (BLP, Biba, Clark–Wilson), CWE/CVSS; bir CVSS vektörünü kendin puanla
3	3. hafta (1)	AEAD, nonce, tuz, KDF, rastgele sayılar; Demo 1–5
4	3. hafta (2)	Anahtar yönetimi, TLS, sabitleme, maskeleme, kabuklar; Demo 6–9
5	4. hafta	CERT çiftleri, biçim dizisi, UAF, UB, korumalar tablosu; Demo 1–7
6	5–6. haftalar	Enjeksiyon, seri durumdan çıkarma, ProGuard, SBOM; RASP mimarisi ve tepki
7	Tekrar	Aşağıdaki örnek soruları süre tutarak çöz; yanlışları ilgili bölümden oku



4. Örnek sorular

Aşağıdaki sorular Quiz-1'in biçimini değil, **düşünme türünü** göstermek içindir.



Kavram

1. Bir para transferinde alıcı hesap numarasının yolda değiştirilmesi CIA'nın hangi özelliğini bozar? Hangi iki önlem bunu engeller?

Bütünlük. Mesaj kimlik doğrulama kodu (HMAC) ya da dijital imza; aktarımda TLS (doğrulama ve ana makine adı denetimiyle).

2. STRIDE'da bir veri akışına hangi harfler sorulur? Neden 'S' sorulmaz?

T, I, D. Veri akışının bir kimliği yoktur; kimlik taklidi varlıklara ve süreçlere sorulur.

3. Clark–Wilson modelinde IVP'nin rolü nedir? Hatalı bir TP sertifikalanırsa ne olur?

IVP, verinin tutarlı olduğunu bağımsız olarak doğrular. Hatalı TP sertifikalanırsa zorlama kuralları bunu göremez (işlem yetkili görünür); tutarsızlığı IVP yakalar.

4. GCM'de nonce tekrarı neden felakettir?

Aynı anahtar ve nonce ile aynı anahtar akışı üretilir; iki şifreli metnin XOR'u iki düz metnin XOR'unu verir, ayrıca kimlik doğrulama anahtarı açığa çıkabilir ve sahte etiket üretilebilir.

5. Parolalar için neden SHA-256 değil de Argon2id kullanılır?

SHA-256 hızlıdır; saldırgan saniyede milyarlarca tahmin dener. Argon2id bilerek yavaş ve bellek yoğundur; tuzla birlikte her tahmini pahalı kılar.

6. Yığın kanaryası hangi saldırıyı durdurur, hangisini durdurmaz?

Dönüş adresine ulaşan ardışık yığın taşmasını (programı sonlandırarak) durdurur. Kanaryadan önceki değişkenlerin bozulmasını, öbek taşmasını ve sızdırılmış kanaryayı durdurmaz.

7. ProGuard'ın dört aşaması nedir? Hangisi dizgeleri gizler?

Küçültme, iyileştirme, gizleme (ad değiştirme), ön doğrulama. Hiçbiri dizgeleri gizlemez.

8. RASP'ta 'tepkiyi tetikleyiciden ayırmak' ne demektir? Neden gerekir?

Algılama anında hemen çökmek yerine, tepkiyi zaman ve kod olarak uzak bir yerde (ör. sırrı silip decoy döndürerek) vermek. Hemen çökmek, saldırgana hangi denetimin tetiklendiğini gösterir.

Kod okuma: hatayı bulun

9. Bu kodda hangi hata var, hangi CERT kuralı ihlal ediliyor?

```
void kaydet(const char *ad) {
    char tampon[32];
    sprintf(tampon, "kullanici=%s", ad);
    syslog(LOG_INFO, tampon);
}
```

İki hata: `sprintf` sınırsız yazar (taşma, STR31-C; `snprintf` kullanılmalı) ve `syslog`'a değişken biçim dizgesi verilmiş (biçim dizisi açığı, FIO30-C; `syslog(LOG_INFO, "%s", tampon)`).

10. Bu kodda ne yanlış?

```
char parola[64];
oku(parola, sizeof parola);
dogrula(parola);
memset(parola, 0, sizeof parola);
return;
```

`memset`'ten sonra dizi hiç okunmadığı için derleyici bu satırı kaldırabilir (MSC06-C, CWE-14). `explicit_bzero` / `SecureZeroMemory` kullanılmalı.

11. Bu kod neden güvensiz, nasıl düzeltilir?

```
String sql = "SELECT * FROM notlar WHERE ogrenci = '" + no + "'";
ResultSet rs = st.executeQuery(sql);
```

SQL enjeksiyonu (CWE-89, IDS00-J). `PreparedStatement` ve `?` parametresi kullanılmalı.

12. Bu çözme kodunda hata nerede?

```
EVP_DecryptUpdate(ctx, acik, &n, sifreli, sifreli_n);
kullan(acik, n);
EVP_DecryptFinal_ex(ctx, acik + n, &m);
```

Açık metin, etiket doğrulanmadan (Final'dan önce) kullanılıyor ve Final'ın dönüş değeri denetlenmiyor. Önce etiket ayarlanıp Final başarılıysa açık metin kullanılmalı; başarısızsa silinmeli.

13. Bu yol denetimi neden yetersiz?

```
if (istek.contains("..")) throw new SecurityException();
return Files.readAllBytes(kok.resolve(istek));
```

Ham dizge denetimi atlatılabilir (mutlak yol, sembolik bağlantı, kodlanmış biçimler). Önce kanonikleştirip (`normalize`, `toRealPath`) sonra kök içinde olduğu denetlenmelidir (FIO16-J).

14. Bu sabitleme kodunun sorunu nedir?

```
try { pinDenetle(zincir); }
catch (KeyStoreException e) { e.printStackTrace(); }
```

Fail-open: anahtar deposu hatası yutulur ve bağlantı kabul edilir. Belirsizlik reddetmeye varmalı (istisna `CertificateException` olarak yeniden fırlatılmalı).

Senaryo

15. Bir mobil uygulama, bütün kullanıcılarında aynı gömülü AES anahtarını kullanıyor. Hangi riskler var, ne önerirsiniz?

Tek bir kopyadan çıkarılan anahtar bütün kullanıcıları etkiler ("bir örneğin kırılması diğerlerini etkilememeli" ihlali); bayt kodundan/ikiliden okunabilir. Öneri: kullanıcı/cihaz başına sunucuda üretilen ya da türetilen anahtar, cihaz ve sürüm bağlama, zarflama, gerekirse whitebox; anahtar koda gömülmez.

16. Değerlendirici, uygulamanızın hata ayıklama derlemesinin dağıtıldığını fark etti. Hangi bulgular yazılır?

Gizleme yok, sembol ve günlük dizgeleri açık, `debuggable` bayrağı açık (hata ayıklayıcı bağlanabilir), sanitizer ya da hata ayıklama kodu olabilir. Sürüm derlemesinin korumaları (checksec) ve günlüğün kaldırıldığı doğrulanmalı.

17. Bir ödeme uygulamasında kullanıcı 'bu işlemi ben yapmadım' diyor. Hangi STRIDE harfi? Hangi önlemler kanıt sağlar?

R (inkâr). Kurcalamaya dayanıklı denetim kaydı (HMAC zinciri, anahtar evrimi), imzalı işlem kayıtları, sunucuya anında gönderilen kayıtlar, zaman damgası.

18. Bir arşiv açma fonksiyonu yazıyorsunuz. Hangi hatalara karşı ne yaparsınız?

Zip Slip (dosya adlarında `../`): her girdinin hedef yolunu kanonikleştirip kök içinde olduğunu denetlemek; boyut ve dosya sayısı sınırları (sıkıştırma bombası); sembolik bağlantıları reddetmek; uzunluk alanlarını doğrulamak ve fuzzing.

19. Log4Shell sırasında kurumların en büyük sorunu neydi? Bu hangi uygulamayla önlenirdi?

Hangi sistemde hangi Log4j sürümünün (başka kütüphanelere gömülü kopyalar dahil) bulunduğunu bilmemek. Her sürümde üretilen ve sürekli izlenen SBOM (CycloneDX/SPDX) + SCA aracı.

20. RASP denetimlerinin tek bir `if` ile değerlendirilmesi neden zayıftır? Daha güçlü tasarım nedir?

Tek karşılaştırmayı ters çeviren ya da denetim fonksiyonunu hep "temiz" döndüren saldırgan hepsini atlatır. Denetim sonuçları sonraki hesabın verisine karışmalı (ör. anahtar türetmeye), opak değerlerle taşınmalı, denetimler çok noktalı ve karşılıklı olmalı, tepki gecikmeli verilmeli.