Güvenlik Gereksinimleri

CEN429 Güvenli Programlama — Hafta 13

Dr. Öğr. Üyesi Uğur CORUH · 11.12.2026

Bugün (3 saat)

Zaman	Bölüm	Ne yapılıyor
0:00–0:20	1	Gereksinim nedir, iyi gereksinim nasıl yazılır?
0:20–0:50	2–3	Gereksinimden kanıta izlenebilirlik; gereksinim bloğu kalıbı ve devredilen gereksinimler
0:50–1:00	Ara	
1:00–1:30	4	Ortak Kriterler: TOE, ST, PP, SFR/SAR, EAL
1:30–1:50	5	FIPS 140-3 ve kriptografik modül doğrulaması
1:50–2:00	Ara	
2:00–2:25	6	ETSI, GSMA, EMVCo, PCI MPoC, OWASP MASVS
2:25–2:50	7–8	Gereksinimleri plana ve varlık yönetimine aktarmak; dersin gereksinim aileleri — uyum matrisi etkinliği
2:50–3:00	9+	Proje adımı (S14, S17), kendini sınama

Öğrenme çıktıları (ÖÇ.5, 7): iyi bir güvenlik gereksinimini kötüsünden ayırmak · izlenebilirlik/uyum matrisi kurmak · CC, FIPS, ETSI, EMVCo, PCI, MASVS'in ne istediğini karşılaştırmak

Önceki haftalardan gelenler

- **Varlık, tehdit ve uygulama koruma planı** — bir yazılımın güvenliğini yöneten yazılı plan: kapsam, mimari, varlıklar, tehditler, karşı önlemler, doğrulama, kalan risk; her varlığın nerede durduğu, ne zaman oluşup silindiği ve hangi korumayı (C/I) gerektirdiği bir **varlık tablosunda** listelenir **(Hafta 1)**
- **Bağımsız değerlendirme, bulgu ve kanıt** — bir ürünün geliştiricisi değil bağımsız bir laboratuvarın kanıta dayalı yürüttüğü değerlendirme süreci; değerlendiricinin belgelediği her eksiklik bir **bulgudur (Hafta 12)**

Bu hafta: varlık tablosunu her gereksinime bağlıyoruz, bağımsız değerlendirme disiplinini "gereksinimden kanıta izlenebilirlik zinciri" olarak kullanıyoruz — çıktı, projenizin **uyum matrisi (S17)** ve **devredilenler (S14)**.

Bu haftanın kavramları

Her terim, gövdede ilk geçtiği yerde tanımlanır; burada yalnız **nerede** olduğunu işaretliyoruz.

Kavram	Nerede
Gereksinim	Bölüm 1
Üç tür gereksinim	Bölüm 1
İyi ve kötü gereksinim	Bölüm 1
İzlenebilirlik	Bölüm 2
Uyum matrisi	Bölüm 2
Gereksinim durumları	Bölüm 2
Devredilen gereksinim	Bölüm 3
Ortak Kriterler (CC)	Bölüm 4
TOE, ST, PP	Bölüm 4
SFR, SAR, EAL	Bölüm 4
FIPS 140-3	Bölüm 5
Sektör standartları	Bölüm 6

1. İyi gereksinim nasıl yazılır?

Kısa tarihçe — güvenlik gereksinimleri nasıl standartlaştı?

- **1985** — **TCSEC** gereksinim düzeylerini resmîleştirir
- **1994** — **FIPS 140** kriptografik modül gereksinimleri (bugün **140-3**)
- **1999** — **Ortak Kriterler**: **PP/ST**, **SFR/SAR**, **EAL**
- **2010'lar** — **OWASP MASVS** (mobil), **ETSI EN 303 645** (IoT), **EMVCo/PCI** (ödeme)

> Değişmeyen ilke: gereksinim **ölçülebilir** ve **izlenebilir** olmalı; kanıtsız "karşılandı" geçersizdir.

Gereksinim (requirement) nedir?

- **Gereksinim:** sistemin **karşılaması gereken** bir koşul.
- Güvenlik gereksinimi: bir güvenlik koşulu.
- İyi gereksinim **doğrulanabilir** (test edilebilir).

İyi gereksinim ölçütleri — şema

İyi gereksinimin ölçütleri

bu ölçütleri karşılamayan gereksinim test edilemez

TEKİL	bir cümle, bir gereksinim
DOĞRULANABİLİR	nasıl test edileceği bellidir
İZLENEBİLİR	kaynağı ve kanıtı vardır
ÖLÇÜLEBİLİR	sayı ya da somut teknik terim içerir
'NE' odaklı	'nasıl' yapılacağını dayatmaz

'Uygulama güvenli olmalı' bunların HİÇBİRİNİ karşılamaz.

Üç tür (hatırlatma)

Tür	Soru	Örnek
İşlevsel	Hangi güvenlik işlevi?	Hassas veri AEAD ile korunur
Güvence	Doğru mu yapıldı?	Statik analiz + test raporu
Süreç	Kurum nasıl çalışır?	Her değişiklik incelenir

İyi gereksinim ölçütleri

- **Doğrulanabilir:** test edilebilir mi?
- **Tekil:** tek bir şey mi söylüyor?
- **Ölçülebilir:** somut mu?
- **Gerçekçi:** uygulanabilir mi?

Kötü → iyi (1)

- **Kötü:** "Uygulama güvenli olmalıdır."
- **İyi:** "Sürüm derlemesi yığın koruyucu, PIE ve tam RELRO ile üretilmelidir."

Neden iyi? **Test edilebilir** (checksec).

Kötü → iyi gereksinim — şema

Kötü gereksinim → iyi gereksinim

fark: ölçülebilir dayanak ve tekil ifade

KÖTÜ

’Uygulama güvenli olmalı’
’Anahtarlar iyi yönetilmeli’
’Veriler uygun saklanmalı’

→ doğrulanamaz

İYİ

’C sınıfı her varlık beklemede en az 128 bit AES-GCM ile şifrelenmelidir’

→ test edilebilir

Zayıf gereksinim test edilemez; test edilemeyen gereksinim karşılandığı kanıtlanamaz.

Kötü → iyi (2)

- **Kötü:** "Veriler şifrelenmelidir."
- **İyi:** "C sınıfı her varlık beklemede ≥128 bit düzeyinde AEAD ile şifrelenmelidir."

Neden iyi? Hangi veri, hangi düzey, hangi yöntem belli.

Kötü → iyi (3)

- **Kötü:** "Anahtarlar iyi yönetilmelidir."
- **İyi:** "(1) Her anahtarın amacı, kripto-periyodu ve imhası belgelenir. (2) İş bitince bellekten silinir."

Tekil, doğrulanabilir maddeler.

İşlenmiş örnek · kötü gereksinimi iyileştirmek

Başlangıç cümlesi (gerçek bir taslaktan):

> "Uygulama, kullanıcı verilerini korumalıdır."

Bu cümleyi altı adımda iyi bir gereksinime dönüştüreceğiz; her adımda **önce/sonra** ve **neden değişti**.

Adım 1 — Belirsiz kelimeleri işaretleyin

- **Önce:** "Uygulama, kullanıcı verilerini korumalıdır."
- Belirsiz kelimeler: "kullanıcı verileri" (hangi veri?), "korumalıdır" (neye karşı, hangi düzeyde?).
- **Neden değişti:** değerlendirici bu cümleyi okuyunca test edecek hiçbir şey bulamaz.

Adım 2 — Varlık tablosuna bağlayın

- **Sonra:** "Yerel veritabanındaki **C sınıfı** alanlar korunmalıdır."
- **Neden değişti:** "kullanıcı verileri" tek şey değil; varlık tablosunda (1. hafta) ayrı satırlardır — oturum belirteci, profil, ödeme jetonu… her biri farklı sınıfta.
- Hâlâ "korunmalı" ölçülemez → sıradaki adım.

Adım 3 — Koruma hedefini netleştirin

- **Sonra:** "...C sınıfı alanların **gizliliği ve bütünlüğü** korunmalıdır."
- **Neden değişti:** tehdit modeli hem "dosyayı okuma" (gizlilik) hem "dosyayı değiştirme" (bütünlük) riskini gösteriyor; ikisi de aynı anda gerekli.

Adım 4 — Ölçülebilir teknik kriter ekleyin

- **Sonra:** "...kimlik doğrulamalı şifrelemeyle (**AEAD**) korunmalıdır."
- **Neden değişti:** 9. hafta kuralı — gizlilik ve bütünlük **aynı anda** isteniyorsa doğru araç sınıfı AEAD'dir; hâlâ belirli bir kütüphane adı verilmiyor.

Adım 5 — Durum ve zorunluluk sözcüğünü ekleyin

- **Sonra:** "**Beklemedeki** C sınıfı hassas veri, kimlik doğrulamalı şifrelemeyle (AEAD) korunmalıdır." (**MUST**)
- **Neden değişti:** veri burada beklemede (disk üzerinde); "-malıdır" seçildi çünkü tehdide karşı vazgeçilmez bir koşul — "-abilir" olsaydı isteğe bağlı sayılırdı.

Adım 6 — Kimlik verin ve tehdide bağlayın

- **Sonra (son hâl):** `CEN429-DR-01` — "Beklemedeki C sınıfı hassas veri, kimlik doğrulamalı şifrelemeyle (AEAD) korunmalıdır." *(Tehdit: T-03, "cihazı ele geçiren saldırgan veritabanı dosyasını okur/değiştirir".)*
- **Neden değişti:** kimliksiz bir gereksinim izlenebilirlik zincirine hiç girmez.

Altı adım · hangi belirsizliği giderdi?

Adım	Giderdiği belirsizlik
1	Belirsiz kelimeyi işaretlemek
2	Hangi veri (varlık tablosu)
3	Hangi hedef (gizlilik/bütünlük)
4	Hangi araç sınıfı (AEAD)
5	Hangi durum, hangi zorunluluk
6	Kimlik ve tehdit bağı

Biri eksik kalırsa gereksinim yine tartışmaya açık kalır; değerlendirici geri gönderir.

İkinci ve üçüncü örnekler

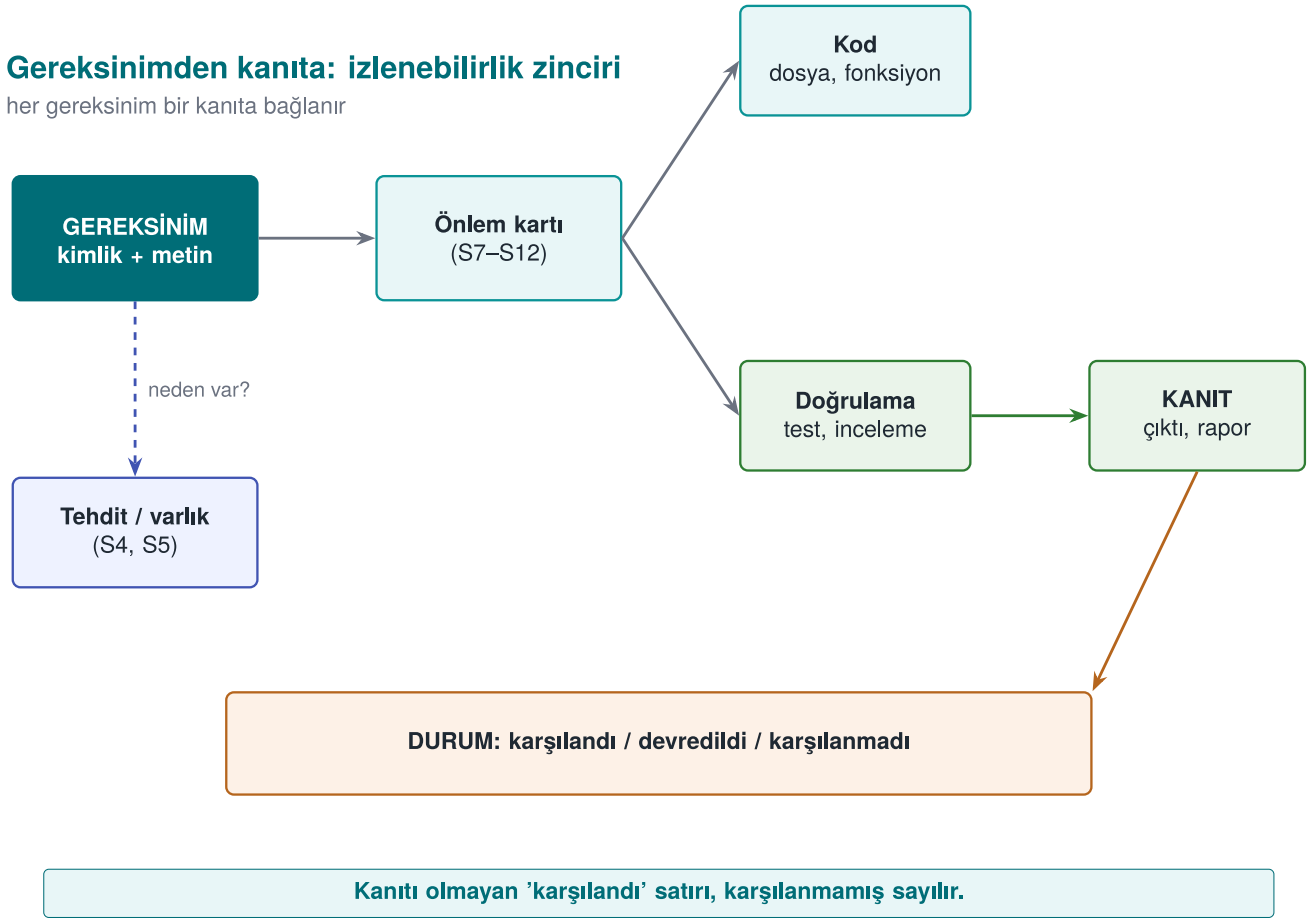
- **İkinci örnek (hız testi):** "Hızlı ve güvenli olmalıdır" → iki ayrı gereksinim tek cümlede. Aynı altı adımdan geçirilince: `CEN429-ID-04` — "Her oturum açma denemesi sunucu tarafında doğrulanmalıdır; istemci tarafı kontrolü tek başına yeterli kabul edilmemelidir."
- **Üçüncü örnek (süreç):** "Kod incelemesi yapılmalıdır." → Hangi değişiklik? Kim inceleyecek? Ne zaman? İyi hâli: "Her değişiklik, ana dala birleştirilmeden önce, yazan geliştirici dışında en az bir kişi tarafından incelenmeli ve onay sürüm kontrol sisteminde kayıt altına alınmalıdır."

Bölüm 1'in kuralı ve zorunluluk sözcükleri

- Gereksinim sütununa **ne** istendiğini yazın; **nasıl** karşıladığınızı ayrı bir önlem cümlesine yazın.
- **Sık hata:** "Gereksinim: AES-256-GCM kullanılmalıdır." → bu bir **önlem**, gereksinim değil; doğrusu "...AEAD ile korunmalıdır" (gereksinim) + "AES-256-GCM, S7.2" (önlem).
- Gereksinim metninde **araç sınıfını** (AEAD, CSPRNG, TLS 1.2+) yazın; belirli kütüphane/sürüm adını önleme koyun.
- **-malıdır (MUST):** zorunlu, karşılanmazsa doğrudan bulgu. **-malı (SHOULD):** güçlü öneri, gerekçe gerekir. **-abilir (MAY):** isteğe bağlı, bulgu değil.

2. Gereksinimden kanıta: izlenebilirlik

Zincir



Her gereksinim bu zincirle bir **kanıta** bağlanmalı.

Durum ve karar kelimeleri — özet

Kelime	Ne zaman kullanılır	Yanında ne yazılmalı
Karşılandı	Ürün gereksinimi kendisi sağlıyor	Önlem + doğrulama + kanıt
Devredildi	Başka bir taraf sağlıyor	Kime + neden + nasıl
Karşılanmadı	Uygulanır ama henüz sağlanmıyor	Kalan risk + planlanan düzeltme
Uygulanmaz	Ürüne hiç uygulanmıyor	Gerekçe
-malıdır (MUST)	Zorunlu	Karşılanmazsa doğrudan bulgu
-malı (SHOULD)	Güçlü öneri	Karşılanmazsa yazılı gerekçe

İzlenebilirlik neden önemli?

- Değerlendirici "bu gereksinim nerede?" diye sorar.
- İzlenebilirlik yoksa: her şeyi baştan aramak.
- Varsa: matristen doğrudan bulunur.

İzlenebilirlik örneği

Gereksinim	Önlem	Doğrulama	Kanıt
Veri AEAD ile	AES-GCM	Birim testi	Test çıktısı
Sürüm korumaları	Bayraklar	checksec	Koruma tablosu

İki yön

- **İleri:** gereksinim → nerede karşılandı?
- **Geri:** bu kod/test → hangi gereksinimi karşılıyor?
- İyi matris **iki yönde** de izlenebilir.

⚠ Kanıtsız "karşılandı"

- Matriste "karşılandı" ama kanıt yoksa...
- Değerlendirici gözünde **karşılanmamış** sayılır.
- İlk bulgulardan biri olur.

İşlenmiş örnek · uçtan uca zincir

Zincirin nasıl kurulduğunu tek bir varlık üzerinden başından sonuna dolduralım: kullanıcının **oturum belirteci**.

Altı halka: varlık → tehdit → gereksinim → önlem → doğrulama → kanıt.

1. Varlık

Alan	Değer
Varlık	Kullanıcının oturum belirteci (session token)
Konum	İstemci belleği; her istekte HTTP başlığında taşınır
Sınıf	**C** (gizlilik)
Yaşam döngüsü	Açılışta üretilir → her istekte kullanılır → çıkışta/zaman aşımında geçersiz kılınır

2. Tehdit

- Ağ üzerinde konumlanan bir saldırgan (ör. aynı halka açık Wi-Fi), şifrelenmemiş bağlantıdan belirteci dinleyip yakalayabilir.
- Yakaladığı belirteçle kendi isteklerinde kullanarak kullanıcı **gibi davranabilir**.
- STRIDE: **Spoofing** + **Information Disclosure** (1. hafta terminolojisi).

3. Gereksinim

Varlık → tehdit → hedef (gizlilik+kimlik) → araç sınıfı → durum → zorunluluk zinciri kısaca tekrar uygulanır:

> `CEN429-DT-01` — "Oturum belirteci yalnız TLS 1.2 ve üzeri, sunucu sertifikası doğrulanmış bir kanaldan taşınmalıdır."

4. Önlem

Kılavuzun **S10.3 "Aktarım güvenliği"** bölümü:

> "İstemci, sunucuya her bağlantıda TLS 1.2+ el sıkışması yapar; sertifika zinciri ve ana bilgisayar adı doğrulanır (bkz. Hafta 10). Kanal kurulmadan hiçbir istek gönderilmez."

5. Doğrulama

Güvenlik testi ekibi bir MITM (ortadaki adam) aracıyla bağlantıya araya girmeye çalışır:

- Geçersiz/kendinden imzalı sertifika sunulunca bağlantının **reddedildiğini** doğrular.
- Düz metin (TLS'siz) bağlantı denemesinin **reddedildiğini** doğrular.

6. Kanıt

- Test aracının günlük çıktısı: `mitm_test_2026-11-03.log`
- CI çalıştırma kaydı: **#617**
- TLS handshake paket dökümü: `handshake.pcapng`
- `evidence/week13/` klasöründe saklanır, matriste bu dosya adlarıyla referans verilir.

Zincirin dolu satırı

Kimlik	Gereksinim	Durum	Önlem	Doğrulama	Kanıt
CEN429-DT-01	Oturum belirteci TLS 1.2+, sertifika doğrulanarak taşınmalı	Karşılandı	S10.3	MITM testi: geçersiz sertifika/düz metin reddedilir	`mitm_test_2026-11-03.log`, CI #617, `handshake.pcapng`

Bir halka eksik olsaydı: varlık yoksa "hangi veri" belirsiz kalır, tehdit yoksa gereksinim keyfi görünür, önlem yoksa iddia desteksiz kalır, doğrulama yoksa "nasıl test edildi" cevapsız kalır, kanıt yoksa satır **ilk bulgu** olur.

"Karşılanmadı" durumunun tam zinciri

Halka	Karşılandı (DT-01)	Karşılanmadı (DT-04)
Gereksinim	Oturum belirteci TLS ile taşınır	Sunucu sertifikası sabitlenmeli
Durum	Karşılandı	**Karşılanmadı**
Önlem	S10.3	— (henüz yok)
Doğrulama	MITM testi geçti	—
Kanıt	Test günlüğü	**Kalan risk:** S16.4'e yazılır

"Karşılanmadı"da önlem/doğrulama boş kalabilir ama **kanıt** (kalan riskin nerede belgelendiği) boş kalamaz.

Bölüm 2'nin kuralı

- Her "karşılandı" satırının kanıt sütununda **bulunabilir, adlandırılmış** bir referans olmalı: dosya adı, CI numarası, test raporu bölümü.
- "Var", "test edildi" kanıt değil, kanıt **vaadidir**.
- Değerlendiricinin en sık yazdığı bulgu: **kanıtsız "karşılandı"**.

Bölüm 1–2 — kısa sınama

1. İyi gereksinimin dört ölçütü?
2. "Uygulama güvenli olmalı" neden kötü?
3. İzlenebilirlik zinciri nedir?

Bölüm 1–2 — cevaplar

1. **Belirli · ölçülebilir/doğrulanabilir · tekil · izlenebilir** (gerçekçi, atomik).
2. **Belirsiz ve ölçülemez**; "güvenli"nin tanımı yok → test edilemez, herkes farklı anlar.
3. **Tehdit/standart → gereksinim → tasarım/kod → test/kanıt.** Her gereksinim bir kaynağa ve bir kanıta **iki yönlü** bağlanır.

3. Gereksinim bloğu ve devredilenler

Devretme üç soru — şema

Devrederken yanıtlanacak üç soru

devretmek serbest, SESSİZ devretmek değil

Kayıt ve onay olmadan atlanan gereksinim = gizli açık, sahipsiz risk.

Gereksinim bloğu kalıbı

Her gereksinim bir **blok** olarak yazılır:

- **Kimlik:** benzersiz numara (ör. CEN429-DU-01)
- **Metin:** ne isteniyor (tekil, ölçülebilir)
- **Durum:** karşılandı / devredildi / karşılanmadı
- **Karşılama:** nasıl karşılandı
- **Doğrulama + kanıt**

Örnek blok

```
CEN429-DU-01
Metin: Yerel DB'deki C sınıfı veri AEAD ile şifrelenir.
Durum: Karşılandı
Karşılama: AES-256-GCM, anahtar TEE'de
Doğrulama: T-05 birim testi
Kanıt: test çıktısı, S16
```

Neden blok?

- Her gereksinim aynı yapıda → **karşılaştırılabilir**.
- Değerlendirici hızlı okur.
- Uyum matrisi bu bloklardan üretilir.

Devredilen gereksinim nedir?

- Bir bileşen bir gereksinimi **kendisi** karşılayamıyor.
- Sorumluluğu üst uygulamaya/OS'a/donanıma **devreder**.
- Bu bir kaçış değil, **açık bir sözleşmedir**.

Devrederken üç soru

Devredilen her gereksinimde yaz:

- **Kime?** (üst uygulama / OS / donanım)
- **Neden?** (neden bu bileşen karşılayamıyor)
- **Nasıl?** (karşı taraf nasıl karşılayacak)

Devredilen · örnek

```
CEN429-AP-03: Güvenli kurulum ve güncelleme
Durum: Devredildi
Kime: Üst mobil uygulama (MPA)
Neden: SDK dağıtım kanalına sahip değil
Nasıl: MPA imzalı güncelleme + sürüm denetimi sağlar
```

⚠️ Sessiz devretme yok

- Bir gereksinimi karşılamıyor ve **yazmıyorsanız** → eksik.
- Devretme **belgelenmeli**; aksi halde "karşılanmadı" sayılır.
- S14 bölümü tam bunun için.

Karşılandı mı, devredildi mi?

Durum	Anlamı
Karşılandı	Ürün sağlar (kanıtla)
Devredildi	Başka taraf sağlar (kime/neden/nasıl)
Karşılanmadı	Henüz yok (kalan risk)

İşlenmiş örnek · bloğu doldurmak (Karşılandı)

Bu, 2. bölümdeki `CEN429-DR-01` zincirinin **kılavuza yazılmış** hâlidir:

> [CEN429-DR] CEN429-DR-01 — KARŞILANDI
> Gereksinim: Beklemedeki C sınıfı hassas veri, kimlik doğrulamalı şifrelemeyle (AEAD) korunmalıdır.
> Karşılama: Yerel kasa dosyası AES-256-GCM ile şifrelenir (S7.2); anahtar cihazın güvenli depolama biriminden türetilir. Doğrulama: birim testi, tek bayt değişince çözmenin reddedildiğini kontrol eder. Kanıt: CI #482, `evidence/week13/test_butunluk_ci482.log` .

Aynı kalıp · Devredildi

> [CEN429-AP] CEN429-AP-07 — DEVREDİLDİ (üst uygulamaya)
> Gereksinim: Uygulama güvenli biçimde kurulmalı ve güncellenmelidir.
> Karşılama: Kütüphanenin kendi dağıtım/güncelleme mekanizması yok. Kime: üst uygulama geliştiricisi. Neden: kütüphane ağ/dosya sistemi seviyesinde kuruluma erişmez. Nasıl: mağaza imza doğrulaması ya da Hafta 6'daki imzalama şemasıyla imzalı güncelleme paketleri.

Devredilen blok da **Karşılama** alanına sahiptir — ama "karşı taraf nasıl yapmalı, biz neden yapmıyoruz" sorusunu cevaplar.

Aynı kalıp · Karşılanmadı

> [CEN429-DT] CEN429-DT-04 — KARŞILANMADI
> Gereksinim: Sunucu sertifikası sabitlenmelidir (certificate pinning).
> Durum: Şu an yalnız standart TLS zinciri doğrulaması var, sabitleme yok. Kalan risk: güvenilir bir kökten sahte sertifika alınırsa MITM mümkün olabilir (S16.4). Planlanan düzeltme: v1.1 (görev #217).

"Karşılanmadı" bloğu **Karşılama** yerine **Durum** ve **Kalan risk** alanlarını doldurur.

Blok alanlarının görevi

- **Başlık satırı** ([Aile] Kimlik — Durum): değerlendiricinin gözünü hızlıca nereye götüreceğini söyler.
- **Gereksinim** satırı: standarttan/tehditten gelen "ne"yi tekrarlar, yeniden yorumlanmaz.
- **Karşılama:** "nasıl"ı somut dosya/bölüm adlarıyla anlatır.
- **Doğrulama + Kanıt:** izlenebilirlik zincirinin son iki halkası.

Donanım devretmesi ve Bölüm 3'ün kuralı

- Güvenli eleman/TEE **varsa**: gereksinim donanıma devredilebilir.
- Güvenli eleman **yoksa**: devredilemez — devredilecek taraf yok; ürün yazılım tabanlı önlem almalı ya da "karşılanmadı" + kalan risk yazmalı. "Donanıma devredildi" yazmak burada **geçersiz devretmedir**.
- **Kural:** karşılama metni her zaman somut bir referansla bitmeli; devretmeden önce karşı tarafın **gerçekten** karşılayabileceğini doğrulayın.

Bölüm 3 — kısa sınama

1. Gereksinim bloğunun alanları?
2. Devrederken hangi üç soru yanıtlanır?
3. Sessiz devretme neden sorun?

Bölüm 3 — cevaplar

1. **ID · ölçülebilir ifade · kaynak/gerekçe · ilgili varlık/tehdit · doğrulama yöntemi (kanıt) · durum/öncelik.**
2. (1) **Neden** şimdi karşılanmıyor? (2) **Riski ne, kim kabul ediyor** (onay)? (3) **Ne zaman/nasıl** karşılanacak (plan/telafi)?
3. Kayıt/onay olmadan gereksinim atlanır → **gizli açık**, sorumlusu yok, denetimde patlar. Açık, gerekçeli, onaylı devret.

4. Ortak Kriterler (ISO/IEC 15408)

Ortak Kriterler — şema

Ortak Kriterler kavramları

EAL4+ her zaman ’daha güvenli’ demek DEĞİLDİR

PP (Protection Profile)	ürün SINIFI için standart gereksinim kümesi
ST (Security Target)	BELİRLİ ürünün güvenlik hedefi
SFR	ürün NE yapacak (fonksiyonel)
SAR	buna NE KADAR güvenilecek (güvence)
EAL	güvence DERİNLİĞİ – güvenlik miktarı değil

Kapsam (ST/TOE) darsa yüksek EAL az şey kapsar. Kapsamı bilmeden EAL karşılaştırılmaz.

CC nedir?

- **Ortak Kriterler:** ürün güvenliğini değerlendirmenin uluslararası standardı.
- Ortak bir **dil** ve **yöntem** sağlar.
- Ülkeler birbirinin değerlendirmesini tanır.

TOE · değerlendirme hedefi

- **TOE (Target of Evaluation):** değerlendirilen tam ürün.
- Benzersiz tanımlanır (12. hafta): sürüm + ikili + özet.
- Kapsam dışı bileşenler yazılır.

ST · güvenlik hedefi

- **ST (Security Target): bu** ürünün güvenlik hedefi belgesi.
- İçinde: tehditler, amaçlar, gereksinimler (SFR/SAR).
- Ürüne özgüdür.

PP · koruma profili

- **PP (Protection Profile):** bir ürün **sınıfı** için ortak gereksinim seti.
- Örnek: "mobil cihaz PP", "akıllı kart PP".
- ST'ler genelde bir PP'yi temel alır.

ST vs PP

	ST	PP
Kapsam	Tek ürün	Ürün sınıfı
Kim yazar	Geliştirici	Topluluk/otorite
Rol	Bu ürünün hedefi	Ortak taban

SFR · işlevsel gereksinimler

- **SFR (Security Functional Requirements):** ürünün **ne yapacağı**.
- Örnek: şifreleme, erişim denetimi, günlükleme.
- Standart bir katalogdan seçilir.

SAR · güvence gereksinimleri

- **SAR (Security Assurance Requirements):** doğru yapıldığına **nasıl güveneceğiz**.
- Örnek: kaynak inceleme, test derinliği, zafiyet analizi.
- EAL bunları belirler.

EAL · değerlendirme derinliği

- **EAL (Evaluation Assurance Level):** 1'den 7'ye derinlik.
- Yüksek EAL = daha derin inceleme.
- "+" ek güvence bileşenlerini gösterir (EAL4+).

⚠️ EAL yanlış anlaşılması

- EAL güvenliğin **derinliği**, güvenliğin **miktarı** değil.
- "EAL4+ > EAL2 daha güvenli" demek **yanlış**.
- Güvenlik, **ST'deki tehdit ve amaçlara** bağlıdır.

İşlenmiş örnek · ST iskeleti — TOE

Bir mobil ödeme bileşeni için üç adımda dolduralım.

Adım 1 — TOE'yi tanımlayın:

> TOE: "CEN429-Pay" kütüphanesi, sürüm 1.0, yalnız Android ARM64 derlemesi.
> **Kapsam dışı:** üst uygulamanın arayüzü, sunucu tarafı bileşenler, işletim sistemi çekirdeği.

Kapsam dışını yazmak, kapsamı yazmak kadar önemlidir.

Tehdit → amaç → SFR eşlemesi

Tehdit	Güvenlik amacı	SFR ailesi	Dersteki gereksinim
T.EAVESDROP	Aktarılan veri gizli/bütünlüklü kalmalı	FCS, FTP	CEN429-DT-01
T.TAMPER	Uygulama bütünlüğünü denetlemeli	FPT	CEN429-AP-04
—	Kimlik doğrulanmalı	FIA	CEN429-ID-02

Bu tablo ST'nin kalbidir: her satır "neden bu gereksinim var" ile "hangi kanıt istenir" arasındaki köprü.

Bölüm 4'ün kuralı

- TOE tanımı her zaman **hem içerdiklerini hem dışarıda bıraktıklarını** listeler.
- SFR'ler katalogdan seçilir, **uydurulmaz**; karşılığı yoksa "genişletilmiş bileşen" olarak gerekçelendirilir.
- Değerlendirici her tehdidin bir amaca, her amacın bir SFR'ye bağlandığını kontrol eder.

5. FIPS 140-3

FIPS 140-3 düzeyleri — şema

FIPS 140-3 güvence düzeyleri

yukarı çıktıkça fiziksel koruma beklentisi artar

Düzey 4	aktif savunma; çevresel saldırılara direnç
Düzey 3	kurcalamaya direnç + yanıt; kimlik tabanlı doğrulama
Düzey 2	kurcalama İZİ + rol tabanlı kimlik doğrulama
Düzey 1	temel: onaylı algoritmalar

'FIPS doğrulanmış kütüphane kullanıyoruz' demek yetmez: doğru kip ve yapılandırma şart.

FIPS 140-3 nedir?

- **Kriptografik modüllerin** doğrulanması standardı.
- Modülün algoritmaları, kendini testi, anahtar yönetimi.
- Yalnız **modülü** kapsar.

Güvenlik düzeyleri (1–4)

Düzey	Kabaca
1	Temel; onaylı algoritmalar
2	Kurcalama **izi**
3	Kurcalamaya **direnç** ve yanıt
4	En yüksek fiziksel koruma

⚠️ FIPS uyumluluğu tuzağı

- FIPS **doğrulanmış** kütüphane kullanmak, uygulamayı otomatik FIPS uyumlu **yapmaz**.
- Uygulama modülü **onaylı kipte** ve **doğru** kullanmalı.
- Anahtarları da doğru yönetmeli.

CC ve FIPS birlikte

- **CC:** ürünün bütününü değerlendirir.
- **FIPS:** kripto modülünü doğrular.
- Bir ürün CC değerlendirmesinde FIPS'li modül kullanabilir.

İşlenmiş örnek · FIPS düzey 1 denetimi

Proje kripto katmanı: OpenSSL `EVP` arayüzü, AES-256-GCM, anahtarlar `getrandom()`'dan üretiliyor, **hata durumunda sessizce varsayılan bir anahtara düşüyor**.

Düzey 1 beklentilerini tek tek işaretleyelim.

Düzey 1 · durum tablosu

Düzey 1 beklentisi	Durum	Neden
Onaylı algoritma, onaylı kipte	Kısmen	CAVP sertifikası yoksa iddia edilemez
Açılış/koşullu öz testler	Karşılanmadı	Proje kendi öz testini çalıştırmıyor
Roller ve hizmetler tanımlı	Karşılanmadı	Kod rol ayrımı yapmıyor
Parametrelerin sıfırlanması	Karşılanmadı	Hata durumunda varsayılan anahtara düşülüyor
Güvenlik politikası belgesi	Karşılanmadı	Böyle bir belge yok

En kritik açık ve Bölüm 5'in kuralı

- Hata durumunda "varsayılan anahtara düşmek", **sıfırlama (zeroization) ilkesinin tam tersidir**.
- iii. haftadaki kural: "rastgele üreteç başarısız olursa **dur**, devam etme" — burada tersi yapılıyor.
- **Kural:** bir FIPS iddiasının arkasında her zaman bir **CMVP sertifika numarası ve kapsamı** olmalı; yoksa "FIPS onaylı algoritmalar kullanıyoruz (doğrulanmış modül değil)" diye dürüstçe daraltın.

Bölüm 4–5 — kısa sınama

1. ST ile PP farkı?
2. SFR ve SAR nedir?
3. "EAL4+ her zaman daha güvenli" neden yanlış?
4. FIPS'li kütüphane uygulamayı FIPS uyumlu yapar mı?

Bölüm 4–5 — cevaplar

1. **PP** bir ürün **sınıfı** için standart gereksinim kümesi; **ST** belirli bir **ürünün (TOE)** neyi karşıladığını söyleyen belge.
2. **SFR** ürünün **ne yapacağı** (fonksiyonel); **SAR** buna **ne kadar güvenileceği** (güvence; EAL buradan gelir).
3. EAL **güvence derinliğini** ölçer, güvenlik miktarını değil; **kapsam (ST/TOE)** darsa yüksek EAL az şey kapsar.
4. **Hayır.** Modülü **doğru kip/yapılandırmada, onaylı algoritmalarla** kullanmak gerekir. Modül sertifikası ≠ uygulama uyumu.

6. Sektöre özgü gereksinim setleri

Sektör standartları — şema

Sektöre özgü gereksinim setleri

hepsi aynı soruyu sorar: hangi gereksinim, hangi kanıtla karşılandı?

OWASP MASVS / MASTG	mobil uygulama – bu dersin projesine en yakın
ETSI EN 303 645	tüketici IoT
EMVCo / PCI	ödeme kartı ekosistemi
GSMA	mobil operatör / SIM

Projenizde MASVS seçin: somut, kontrol edilebilir maddeler + MASTG test rehberi.

Neden sektör standartları?

- CC/FIPS geneldir; sektörler kendi risklerine göre **ek** gereksinim ister.
- IoT, mobil, ödeme farklı tehdit ortamları.
- Projeniz hangisine yakınsa onu izleyin.

ETSI EN 303 645 (IoT)

- Tüketici IoT için **temel** güvenlik gereksinimleri.
- Örnek: varsayılan parola yok, güvenli güncelleme, hassas parametreleri güvenle sakla.
- Geniş, uygulanabilir taban.

ETSI · örnek maddeler

- Benzersiz parolalar (ortak varsayılan yok).
- Güvenlik güncellemelerini yönet.
- **Hassas güvenlik parametrelerini güvenle sakla** (→ projede S5/S7/S8).

GSMA

- Mobil operatör ekosistemi için güvenlik yönergeleri.
- Cihaz, ağ, servis güvenliği.
- SIM/eSIM, kimlik.

EMVCo

- Kart ödeme ekosistemi standartları (kartlar, terminaller, mobil ödeme).
- İşlevsel uygunluk + güvenlik değerlendirmesi.
- Saldırı potansiyeli puanlaması (12. hafta).

PCI

- **PCI DSS:** kart verisi işleyen sistemler için güvenlik.
- **PCI MPoC/CPoC:** yazılım tabanlı ödeme kabulü.
- Sıkı; laboratuvar değerlendirmesi.

OWASP MASVS

- **Mobile Application Security Verification Standard.**
- Mobil uygulama güvenlik **gereksinimleri**.
- Düzeyler: temel (L1), derinlemesine (L2), direnç (R).

MASVS · projeniz için

- **En uygulanabilir** gereksinim seti.
- MASTG ile test edilir (12. hafta).
- Kripto, depolama, iletişim, kod kalitesi, direnç.

Standartlar · karşılaştırma

Standart	Alan	Odak
CC	Genel ürün	Değerlendirme derinliği
FIPS 140-3	Kripto modülü	Modül doğrulama
ETSI 303 645	IoT	Temel gereksinim
EMVCo/PCI	Ödeme	İşlevsel + güvenlik
MASVS	Mobil	Uygulama gereksinimi

Karşılaştırma · ders

- Hepsi aynı kökü paylaşır: **yazılı, doğrulanabilir** gereksinim.
- Fark: kapsam ve sıkılık.
- Projeniz için **MASVS** en pratik başlangıç.

Aynı gereksinim, dört standart (1) — beklemede veri

Standart	Karşılığı
Ortak Kriterler	FCS + FDP bileşenleri
FIPS 140-3	Düzey 1: onaylı algoritma, öz test, sıfırlama
ETSI EN 303 645	Madde 5.4 "Hassas parametreleri güvenle sakla"
OWASP MASVS	MASVS-STORAGE
Dersin ailesi	`CEN429-DR-01`

Aynı gereksinim, dört standart (2) — zayıf kimlik doğrulama

Standart	Karşılığı
Ortak Kriterler	FIA bileşenleri
FIPS 140-3	Düzey 2: rol tabanlı kimlik doğrulama
ETSI EN 303 645	Madde 5.1 "Evrensel varsayılan parola yok"
OWASP MASVS	MASVS-AUTH
Dersin ailesi	`CEN429-ID-01` / `CEN429-ID-02`

Örtüşme ve Bölüm 6'nın kuralı

- Bir standarda uyum kanıtı, başka bir standartta da **kısmen yeniden kullanılabilir** — ama "otomatik karşılanır" demek değildir; her standardın **kendi ek ölçütü** vardır.
- Aynı gereksinimi birden çok standarda eşlemek (bir standart sütunu ekleyerek) işi hızlandırır; sertifikasyon değişse bile iş tekrarlanmaz.
- "FIPS'li modül kullanıyoruz, o yüzden MASVS-CRYPTO'yu da karşılıyoruz" cümlesi **tehlikelidir**.
- **Kural:** örtüşmeyi kanıt toplamayı hızlandırmak için kullanın, ama her standardı **ayrı ayrı** işaretleyin.

Bölüm 6 — kısa sınama

1. ETSI EN 303 645 kimin için?
2. MASVS neden projeniz için en uygun?
3. CC ile MASVS farkı?

Bölüm 6 — cevaplar

1. Tüketici **IoT** cihazları için temel güvenlik (varsayılan parola yok, güncelleme, güvenli iletişim…).
2. Proje uygulama odaklı; **MASVS** somut, kontrol edilebilir maddeler + **MASTG** test rehberi → doğrudan uygulanır.
3. **CC** ağır, akredite laboratuvar/sertifika (EAL); **MASVS** hafif, açık, geliştirici-dostu **öz-değerlendirme** standardı.

7. Gereksinimleri yazılım planına ve varlık yönetimine aktarmak

Uyum matrisinin yapısı — şema

Uyum matrisi: her satır bir kanıt ister

S17'nin iskeleti

Kimlik	Gereksinim	Durum	Önlem	Kanıt
DR-01	Beklemede AEAD	Karşılandı	S7.2	test çıktısı
AP-07	Güvenli güncelleme	Devredildi	S14.2	gerekçe
DT-04	Sertifika sabitleme	Karşılanmadı	–	kalan risk

Kanıt sütunu boş olan "Karşılandı" satırı, KARŞILANMAMIŞ sayılır.

Gereksinim aktarma kararı — şema

Gereksinim → varlık → önlem

Gereksinim bloğunun alanları

her gereksinim bu altı alanla yazılır

Kimlik (ID)	`CEN429-DR-01` gibi benzersiz
Ölçülebilir ifade	ne yapılacak
Kaynak / gerekçe	hangi tehdit ya da standart
İlgili varlık	neyi koruyor (S5)
Doğrulama yöntemi	hangi test kanıtlayacak (S16)
Durum	karşılandı / devredildi / karşılanmadı

Alanlardan biri boşsa gereksinim izlenebilir değildir.

Her gereksinim bir **varlığa** ve bir **önleme** bağlanır.

Varlık yönetimiyle bağ

- Gereksinimler S5 **varlık listesine** bağlanır.
- Her varlık: C/I/I+ etiketi, koruma önlemi.
- Gereksinim, o korumayı **zorunlu** kılar.

Plana aktarmak

- İşlevsel gereksinim → tasarım/kod (S6–S11).
- Güvence gereksinimi → test (S16).
- Süreç gereksinimi → geliştirme süreci (S13).

İşlenmiş örnek · CR-03'e altı adımı uygulamak

`CEN429-CR-03` ("anahtarlar iş bitince silinmelidir"):

- **1. Uygulanabilirlik:** oturum ve kasa dosyası anahtarı var → uygulanır.
- **2. Sorumluluk:** kod projenin kendisi → biz karşılarız.
- **3. Varlıklara bağlama:** kasa dosyası anahtarının "silinme" sütunu boş — **eksiklik burada fark edilir**.

Aynı örnek · tehditten sürüm planına

- **4. Tehditlere bağlama:** "fiziksel erişimi olan saldırgan bellek dökümü alır" tehdidine bağlanır.
- **5. Önlem/doğrulama:** `sifreleme_bellek_sil()` çağrısı; doğrulama: bellek dökümünde anahtar baytlarının **bulunmadığını** göstermek.
- **6. Sürüm planı:** oturum anahtarı v0.9'da karşılandı; kasa dosyası anahtarı v1.0'a planlandı, o güne kadar **karşılanmadı** + kalan risk yazılır.

Bölüm 7'nin kuralı

- Her "uygulanmaz" kararı, gerekçesini destekleyen bir **kanıtla** (kod taraması, mimari diyagram) birlikte yazılmalı.
- Gerekçesiz "uygulanmaz", gerekçesiz "karşılandı" kadar güvenilmezdir.
- Altı adımdan 3. ve 4.'ü (varlık/tehdit bağlama) atlamak en sık yapılan hatadır.

8. Dersin gereksinim aileleri

Gereksinim aileleri — şema

Dersin kendi kimlikleri

Bu ders, açık standartlardan uyarlanmış kendi gereksinim ailelerini kullanır:

CEN429-**AP** (uygulama koruma), **ID** (kimlik), **AS** (varlık), **DR/DU/DT** (veri: beklemede/kullanımda/aktarımda)…

Aileler (1)

Aile	Konu	Örnek
AP	Uygulama koruma (gizleme, RASP)	Hassas fonksiyonlar gizlenir ve bütünlük denetimiyle korunur
ID	Kimlik ve bağlama	Cihaz ve sürüm, anahtar kullanımına bağlanır
AS	Varlık yönetimi	Her varlık C/I/I+ ile etiketlenir ve yaşam döngüsü belgelenir
DR	Beklemede veri	Yerel veri AEAD ile şifreli

Aileler (2)

Aile	Konu	Örnek
DU	Kullanımdaki veri	Anahtar kullanımdan sonra silinir
DT	Aktarımdaki veri	TLS 1.3 + zincir doğrulama
RP	Raporlama/günlük	Sürümde hassas veri düz loglanmaz
CR	Kriptografi	Onaylı algoritma, kip, anahtar uzunluğu; anahtar hiyerarşisi belgeli
DV	Geliştirme/süreç	Her değişiklik birleştirilmeden önce incelenir; SBOM güncellenir

Her aile → haftalara bağ

- AP → 4, 5, 6, 9, 11, 14
- CR/DR/DU/DT → 3, 10
- AS → 1, 3
- DV/RP → 12, 13

Projeniz bu ailelerden bir **alt küme** seçer.

Uyum matrisi etkinliği · Adım 1

- Kendi projenizden **beş gereksinim** seçin (farklı ailelerden).
- Her biri için bir **gereksinim bloğu** yazın.

Etkinlik · adım 2

Her gereksinim için matris satırı doldurun:

Gereksinim	Durum	Bölüm	Doğrulama	Kanıt
…	…	…	…	…

Etkinlik · adım 3

- En az birini **devredilen** olarak yazın (kime/neden/nasıl).
- En az birinin **kanıtını** somut gösterin.
- Bir "karşılanmadı" varsa **kalan riske** yazın.

Etkinlik · değerlendirme

Değerlendirici gözüyle kontrol:

- Her satırın kanıtı var mı?
- Devredilenler açık mı?
- Gereksinimler **doğrulanabilir** mi?

İşlenmiş örnek · ilk taslak (yetersiz)

Bir takımın `CEN429-AP-04` için matris satırı, **ilk taslakta**:

Kimlik	Gereksinim	Durum	Önlem	Doğrulama	Kanıt
CEN429-AP-04	Uygulama çalışma anında bütünlüğünü denetlemeli	Karşılandı	Bütünlük kontrolü var	Test edildi	—

Üç sorun: (1) önlem, gereksinimin **tekrarı**. (2) "Test edildi" doğrulama **yöntemi değil**. (3) Kanıt **boş**.

Taslağı düzeltme · adım adım

1. Önlemi somutlaştır: açılışta kod bölümünün özeti (hash) hesaplanır, derleme zamanında gömülü beklenen özetle karşılaştırılır; uyuşmazsa çalışma durur (S12.3).
2. Doğrulama yöntemini netleştir: ikili dosyanın bir baytı değiştirilir, uygulamanın **başlamayı reddettiği** doğrulanır.
3. Kanıt ekle: CI işi `ci-integrity-check`, çalıştırma **#391**, `integrity_test.log`.

Düzeltilmiş satır

Kimlik	Durum	Önlem	Doğrulama	Kanıt
CEN429-AP-04	Karşılandı	S12.3: açılış özeti hesaplanır, gömülü özetle karşılaştırılır	Bayt değişince başlamanın reddedildiği test edilir	CI `ci-integrity-check` #391, `integrity_test.log`

Fark neyde?

- Fark satırın **uzunluğunda** değil, her hücrenin **somutluğunda**.
- İlk taslak da "dolu" görünüyordu ama hiçbir hücresi **bağımsız olarak doğrulanamazdı**.
- Değerlendirici için önemli olan: her hücreyi okuyunca "bunu nereden biliyoruz?" sorusuna cevap bulabilmek.

Bölüm 8'in kuralı

- Her satırın kanıtı **o satıra özgü** olmalı: belirli test adı, belirli CI numarası, belirli günlük dosyası.
- Tek bir genel raporu bütün satırlara kopyalamak, matrisin **tamamını** şüpheli hâle getirir.
- Genel bir rapora referans veriliyorsa, raporun **hangi bölümü/sayfası** olduğu da belirtilmeli.

Bölüm 7–8 — kısa sınama

1. Gereksinim hangi iki şeye bağlanır?
2. Dersin gereksinim ailelerinden üçünü sayın.
3. Uyum matrisinde her satırda ne olmalı?

Bölüm 7–8 — cevaplar

1. Yukarıda **kaynağa** (tehdit/standart/varlık) ve aşağıda **kanıta** (test/kod) — iki yön.
2. (Dokuz aileden herhangi üçü) ör. **veri güvenliği · kod sağlamlaştırma (hardening) · RASP · kriptografi/sertifika · bellek koruması · arayüz koruması**.
3. Gereksinim **ID+ifade · durum** (karşılandı/devredildi) **· kanıt** (test/dosya/S-bölümü). **Kanıtsız satır karşılanmamış** sayılır.

Uçtan uca: NotKasa'nın uyum matrisi (1/2)

Sentetik uygulama **NotKasa** (yerel şifreli not tutar, sunucuyla konuşur, güncellenir) için gereksinim tehditten türer:

```
Tehdit: cihaz çalınırsa yerel notlar okunur
   ↓
Amaç: beklemede gizlilik
   ↓
```

Bu zincirden iki "karşılandı" bloğu:

DR-01 — beklemede veri:

```
CEN429-DR-01
Metin: Yerel DB'deki C sınıfı veri AES-256-GCM ile şifrelenir.
Durum: Karşılandı
Karşılama: AES-256-GCM, anahtar TEE'de
Doğrulama: T-05
```

CR-02 — kripto:

```
CEN429-CR-02
Metin: Anahtarların amacı, kripto-periyodu, imhası belgelenir.
Durum: Karşılandı
Karşılama: S8 anahtar tablosu
Doğrulama: belge incelemesi
```

Uçtan uca: NotKasa'nın uyum matrisi (2/2)

DT-03 — aktarımda veri:

```
CEN429-DT-03
Metin: Sunucu iletişimi TLS 1.3 ve sertifika zinciri doğrulaması kullanır.
Durum: Karşılandı
Karşılama: TLS 1.3 + SAN denetimi + SPKI pin
Doğrulama: T-11
Kanıt: test + S11
```

AP-04 — uygulama koruma (devredilen):

```
CEN429-AP-04
Metin: Güvenli kurulum ve güncelleme sağlanır.
Durum: Devredildi
Kime: Üst uygulama (MPA)
Neden: SDK dağıtım kanalına sahip değil
Nasıl: MPA imzalı güncelleme + sürüm denetimi
```

AS-05 — karşılanmadı:

```
CEN429-AS-05
Metin: Tüm hassas varlıklar için bellek izleme tespiti.
Durum: Karşılanmadı
Kalan risk: köklü cihazda canlı bellek analizi
Azaltma: kısa ömürlü anahtar + sunucu denetimi
```

Matris · toplu görünüm

ID	Durum	Bölüm	Kanıt
DR-01	Karşılandı	S8	T-05
CR-02	Karşılandı	S8	S8
DT-03	Karşılandı	S11	T-11
AP-04	Devredildi	S14	—
AS-05	Karşılanmadı	S12	kalan risk

Matristen okunanlar

- Üç karşılandı (kanıtlı), bir devredildi, bir kalan risk.
- Her satır izlenebilir.
- Değerlendirici bu tablodan başlar.

Klasik hatalar — özet

Hata	Bölüm
Doğrulanamaz gereksinim ("güvenli olmalı" gibi)	Bölüm 1
Kanıtsız "karşılandı"	Bölüm 2
Sessiz devretme	Bölüm 3
EAL'i güvenlik miktarı sanmak	Bölüm 4
FIPS'li kütüphane = FIPS uyumlu sanmak	Bölüm 5
Kalan riski boş bırakmak	Bölüm 2

Her satırın ayrıntısı, ilgili bölümdeki ⚠️ kutusunda ve kuralda işlendi; burada tek bakışta toparlıyoruz.

9. Dönem projesi: bu hafta

Proje · S14 ve S17

S17 ve S14, değerlendiricinin **ilk açtığı** belgelerdir.

- [ ] **S17 uyum matrisi:** uygulanan her gereksinim için durum, bölüm, doğrulama, kanıt.
- [ ] **S14 devredilenler:** kime, neden, nasıl.
- [ ] En az bir "karşılanmadı" varsa **kalan riske** yaz.

Değerlendirici gözüyle

- Kanıtsız "karşılandı" = karşılanmamış.
- Devredilenler açık ve gerekçeli mi?
- Gereksinimler doğrulanabilir mi?

Kontrol listesi

- [ ] Her gereksinim doğrulanabilir + tekil
- [ ] Her satırda durum/bölüm/doğrulama/kanıt
- [ ] Devredilenler: kime/neden/nasıl
- [ ] Karşılanmayanlar → kalan risk
- [ ] Standart eşlemesi (ETSI/MASVS/CC)
- [ ] En az bir aile başına gereksinim

45 dakikada nasıl hazırlanır + kanıt klasörü

1. Aile tablosundan uygulanan gereksinimleri işaretleyin; uygulanmayanları "uygulanmaz, gerekçe: ..." bırakın.
2. Altı sütunu (kimlik, gereksinim, durum, önlem, doğrulama, kanıt) doldurun; kanıtsız "karşılandı" yazmayın.
3. "Devredildi" satırlarını S14'e, kime/neden/nasıl ile taşıyın.
4. Varlık tablonuzu (S5) her gereksinime bağlayın; S1'e dayandığınız standartları yazın.

```
evidence/week13/  <- test/log/pcap dosyaları
  README.md       <- her dosya hangi gereksinimin kanıtı, bir satırla
```

Zaman kısıtlıysa · öncelik sırası

1. Önce **en az 15 gereksinimlik** bir S17 iskeleti (kimlik+gereksinim+durum) — boş S17'den çok daha iyi.
2. Elde **gerçek kanıtı olan** "karşılandı" satırlarını tamamlayın; kanıtı olmayanı dürüstçe "karşılanmadı" yapın.
3. Devredilen satırları S14'e taşıyın.
4. Son olarak kalan önlem/doğrulama sütunlarını doldurun.

Kural: "az ama dürüst" bir matris, "çok ama kanıtsız" olandan her zaman daha iyi puan alır.

Çözümlü kendini sınama

Soru 1

İyi bir güvenlik gereksiniminin dört ölçütü?

Cevap: Doğrulanabilir, tekil, ölçülebilir, gerçekçi.

Soru 2

"Uygulama güvenli olmalıdır" neden kötü? Nasıl düzeltilir?

Cevap: Doğrulanamaz. Ölçülebilir yazılır: ör. "sürüm derlemesi yığın koruyucu, PIE ve tam RELRO ile üretilir".

Soru 3

İzlenebilirlik zinciri nedir?

Cevap: Gereksinim → önlem → doğrulama → kanıt. Her gereksinim bir kanıta bağlanır.

Soru 4

Uyum matrisinde kanıtsız "karşılandı" ne demek?

Cevap: Değerlendirici için karşılanmamış sayılır; ilk bulgulardan biridir.

Soru 5

Bir gereksinimi devrederken hangi üç soru yanıtlanır?

Cevap: Kime, neden, nasıl. Sessiz devretme "karşılanmadı" sayılır.

Soru 6

ST ile PP farkı?

Cevap: ST tek ürünün güvenlik hedefi; PP bir ürün sınıfı için ortak gereksinim seti. ST genelde bir PP'yi temel alır.

Soru 7

"EAL4+ her zaman EAL2'den güvenli" doğru mu?

Cevap: Hayır. EAL değerlendirmenin derinliğidir; güvenlik ST'deki tehdit ve amaçlara bağlıdır. "+" ek güvence bileşenidir.

Soru 8

FIPS doğrulanmış kütüphane kullanan uygulama FIPS uyumlu mudur?

Cevap: Tek başına değil. Doğrulama modülü kapsar; uygulama onaylı kipte, doğru kullanmalı, anahtarları doğru yönetmeli.

Soru 9

ETSI EN 303 645'in "hassas parametreleri güvenle sakla" maddesi projede nereye karşılık gelir?

Cevap: S5 varlık listesi, S7 veri güvenliği/kabuk matrisi, S8 anahtar yaşam döngüsü.

Soru 10

Bir kütüphane "güvenli güncelleme" gereksinimini karşılayamıyorsa?

Cevap: Üst uygulamaya devreder ve kılavuzda kime/neden/nasıl karşılanacağını yazar.

Sözlük

Terim	Anlam
İşlevsel/güvence/süreç	Üç gereksinim türü
İzlenebilirlik	Gereksinim→kanıt zinciri
Devredilen	Başka tarafa aktarılan gereksinim
TOE/ST/PP	CC temel kavramları
SFR/SAR/EAL	İşlevsel/güvence/derinlik

Bir sonraki hafta

14. hafta — Tigress ve çeşitlendirme

Bu hafta tanımladığımız dersin gereksinim ailelerinden **CEN429-AP** (uygulama koruması) grubundaki "hassas kod bölümleri gizlenmelidir" gibi gereksinimlerin somut karşılığını, yani **önlem** tarafını, 14. haftada göreceğiz: Tigress aracıyla kod gizleme ve çeşitlendirme dönüşümlerini uygulayıp bunları uyum matrisinizin (S17) ilgili satırlarına bağlayacaksınız. Gizleme kurallarının otomatik, çeşitlendirilmiş uygulaması; ölçme ve derleme hattı.

> Bir gereksinim, tehditten doğar; bir önleme, bir teste ve bir **kanıta** bağlanır; karşılanamıyorsa açıkça
> **devredilir**. Uyum matrisi bu zincirin haritasıdır.