Uygulama Güvenliği Doğrulama Standardı 4.0

Final

Türkçe çeviri tarihi; Temmuz 2020

İçindekiler

Bilgilendirme

Standart Hakkında

Uygulama güvenliği doğrulama standardı; yazılım tasarımcılar, yazılım geliştiriciler, test yapanlar, güvenlik uzmanları ve hatta müşteriler için uygulama güvenliği gereksinimlerinden oluşan bir listedir.

Copyright and License

Version 4.0.1, March 2019

Proje Liderleri

- Andrew van der Stock
- Daniel Cuthbert
- Jim Manico
- Josh C Grossman
- Mark Burnett

Değerlendirenler ve Katkıda Bulunanlar

- Osama Elnaggar
- Erlend Oftedal
- Serg Belkommen
- David Johansson
- Tonimir Kisasondi
- Ron Perris
- Jason Axley
- Abhay Bhargav
- Benedikt Bauer
- Elar Lang
- ScriptingXSS
- Philippe De Ryck
- Grog's Axle
- Marco Schnüriger
- Jacob Salassi
- Glenn ten Cate
- Anthony Weems
- bschach
- javixeneize
- Dan Cornell
- hello7s
- Lewis Ardern
- Jim Newman
- Stuart Gunter
- Geoff Baskwill
- Talargoni
- Ståle Pettersen
- Kelby Ludwig
- Jason Morrow
- Rogan Dawes

Türkçe Çeviri Ekibi

- Bekir Akgül
- Bünyamin Demir
- Eyüp Sercan Akgül
- Fatih Ersınadım
- Onur Karasalihoğlu

Önsöz

Uygulama Güvenliği Doğrulama Standardı (ASVS) sürüm 4.0'a hoş geldiniz. Bir topluluk tarafından oluşturulan ASVS; modern web uygulamalarını tasarlarken, geliştirirken ve test ederken gerekli olan fonksiyonel ve fonksiyonel olmayan güvenlik kontrollerini içeren bir temel yapı oluşturmayı amaçlamaktadır.

ASVS v4.0, topluluk çabası ve endüstri geribildirimlerinin bir sonucu olarak oluşturulmuştur. Bu sürümde, gerçek dünya deneyimlerinin de ASVS'ye katkıda bulunması gerektiğini düşündük. Bu sayede ASVS uygulayacak firmalar, mevcut firmalara katkıda bulunurken, bir yandan da diğer firmaların deneyimlerinden faydalanabileceklerdir.

Risk analizi öznel bir konu olduğu için tüm standartlara uygun bir genelleme yapmak zordur. Bu yüzden standart üzerinde %100 anlaşma olmasını beklemiyoruz. Ancak, bu sürümde yapılan son güncellemelerin doğru yönde atılmış bir adım olduğunu umuyoruz ve endüstri standardındaki önemli konseptleri özenle geliştirmeye çalıştık.

ASVS 4.0'daki Yenilikler

Bu versiyondaki en önemli değişiklik; modern, kanıta dayalı ve gelişmiş kimlik doğrulama kontrollerini getiren NIST 800-63-3 Dijital Kimlik Kılavuzlarının benimsenmesidir. Gelişmiş bir kimlik doğrulama standardı ile uyum konusunda bir miktar olumsuz geri dönüş beklememize rağmen, standartların saygın bir başka uygulama güvenlik standardı ile uyumlu hale getirilmesinin gerekli olduğunu düşünüyoruz.

Bilgi güvenliği standartları konusunda gerekliliklerin sayısı en aza indirmeye çalışmalıdır, böylece kuruluşlar tutarsız olabilen veya kendi yapılarına uygun olmayan kontrollere karar vermek zorunda kalmazlar. OWASP Top 10 2017 ve şimdi OWASP Uygulama Güvenliği Doğrulama Standardı (ASVS) kimlik doğrulama ve oturum yönetimi için NIST 800-63 ile uyumlu hale geldi. Güvenliği en üst düzeye çıkarmak ve uyumluluk maliyetlerini en aza indirmek için diğer standart belirleyen kurumların bizimle, NIST ve diğerleriyle birlikte çalışmasını teşvik ediyoruz.

ASVS 4.0 baştan sona yeniden numaralandırıldı. Yeni numaralandırma şeması; uzun süredir yok olan bölümlerdeki boşlukları kapatmamıza ve bir geliştiricinin/ekibin uyması gereken kontrol sayısını en aza indirgemek için uzun bölümleri bölerek kısaltmamıza izin verdi. Örneğin, bir uygulama JWT kullanmıyorsa, oturum yönetimindeki JWT ile ilgili bölümün tamamen atlanabilir hale geldi.

4.0 sürümündeki yenilikler ile son on yılda en çok istenen özelliklerden biri olan Ortak Zayıflık Numaralandırması (Common Weakness Enumeration - CWE) ile ASVS maddelerinin eşlenmesi sağlandı. CWE eşleme, araç üreticilerinin ve güvenlik açığı yönetim yazılımı kullananların diğer araçlardan ve önceki ASVS sürümlerinden gelen sonuçları 4.0 ve sonraki sürümlerle eşleştirmelerine olanak tanıyor. CWE numaralarına yer açmak için; tamamen yeniden numaralandırdığımız ASVS maddelerinde yer alan "Since" ('den beri) sütununu emekliye ayırdık. Bu sütun zaten bütün maddelerde yeniden numaralandırma yaptığımız için çok anlamlı değildi.

ASVS'deki her öğenin ilişkili bir CWE'si yoktur ve CWE'nin çok fazla tekrarı olduğundan, en yakın eşleşme yerine en yaygın kullanılan CWE'leri kullanmaya çalıştık. Güvenlik kontrolleri her zaman zayıflıklarla eşleştirilemez bu nedenle CWE topluluğu ile devam eden tartışmaya ve daha genel olarak bu boşluğu kapatmaya, bilgi güvenliği alanında çalışanları davet ediyoruz.

OWASP Top 10 2017 ve OWASP Proaktif Kontroller 2018 gereksinimlerini kapsayabilmek ve üzerine katabilmek için çalıştık.

OWASP Top 10 2017 listesi olası güvenlik ihlalinden kaçınmak için gereken minimum gereksinimlerden oluşur. Bu nedenle Level 1'de en iyi 10 gereksinimlerini zorunlu hale getirdik. Böylece OWASP Top 10 uygulayıcılarının gerçek bir güvenlik standardına yükselmesini sağladık.

ASVS 4.0 Seviye 1'in; uygulama tasarımı, kodlama, test etme, güvenli kod incelemeleri ve sızma testleri için PCI DSS 3.2.1 Bölüm 6.5'in kapsamlı bir üst kümesi olmasını sağlamak için yola çıktık. Bu, V5'teki arabellek taşması ve güvensiz bellek işlemlerini ve V14'teki güvensiz bellekle ilgili derleme bayraklarının yanı sıra mevcut endüstri lideri uygulama ve web hizmeti doğrulama gereksinimlerini de kapsıyor.

İşlevsel programlama, sunucusuz API, mobil, bulut, kapsayıcılar (containers), CI/CD ve DevSecOps, federasyon ve daha fazlası, modern uygulama mimarisini görmezden gelemeyeceğimiz için; modern uygulamalar, orijinal ASVS'nin 2009 yılında piyasaya sürülmesinden sonra yapılan uygulamalardan çok farklı bir şekilde tasarlandığı için; ASVS'nin yalnızca monolitik sunucu tarafı denetimlerinden, tüm modern uygulamalar ve API'ler için güvenlik denetimleri sağlamaya geçişini tamamladık. ASVS, her zaman geleceğe bakmalıdır, böylece birincil kitlemiz olan geliştiricilerimize sağlam önerilerde bulunabiliriz. Uygulamaların, sadece tek bir kuruluşun sahip olduğu sistemlerde çalıştığını varsayan gereksinimleri açıklığa kavuşturduk veya sildik.

ASVS 4.0'ın boyutu ve diğer tüm ASVS düzenlemeleri için temel ASVS olma arzumuz nedeniyle, Mobil Uygulama Güvenliği Doğrulama Standardı (MASVS) için mobil uygulama bölümümüzü kaldırdık. Nesnelerin İnterneti Eki gelecekte OWASP Nesnelerin İnterneti Projesi'nin IoT ASVS içerisinde kullanılacaktır. Ancak Ek C'ye IoT ASVS'nin erken bir ön izlemesini ekledik. Hem OWASP Mobil Ekibi'ne hem de OWASP IoT Proje Ekibi'ne ASVS'yi desteklediklerini için teşekkür ediyor ve gelecekte tamamlayıcı standartlar sağlamak için onlarla çalışmayı dört gözle bekliyoruz.

Son olarak, daha az etkili kontrolleri listeden kaldırdık. Zamanla, ASVS kapsamlı bir kontrol seti olmaya başladı, ancak tüm kontroller güvenli yazılım üretmede eşit katkı sağlamaz. ASVS'nin gelecekteki bir sayısında, Ortak Zayıflık Puanlama Sistemi (CWSS), gerçekten önemli olan ve daha fazla önceliklendirilmesi gereken maddelerin ayrıştırılmasında yardımcı olacaktır.

Sürüm 4.0 itibariyle, ASVS yalnızca geleneksel ve modern uygulama mimarisini, çevik güvenlik uygulamalarını ve DevSecOps kültürünü kapsayan, lider web uygulamaları ve hizmet standardı olmaya odaklanacaktır.

ASVS Kullanımı

ASVS'nin iki temel amacı vardır:

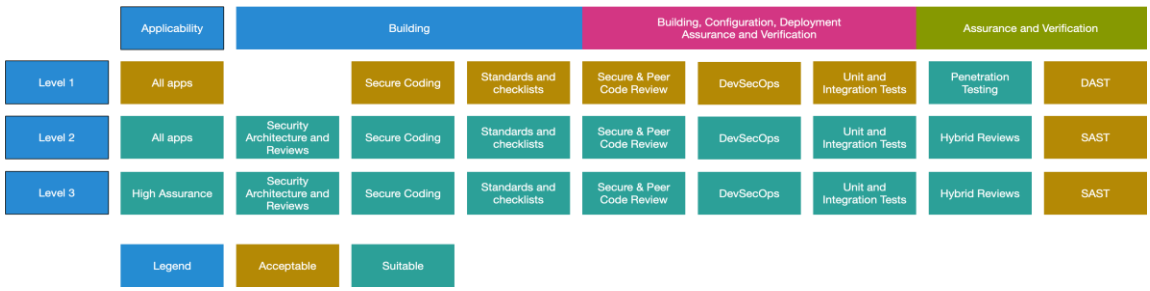
- Organizasyonların güvenli uygulamalar geliştirerek bunu devam ettirmesi
- Müşteriler ile güvenlik araç ya da servis sağlayıcıları arasında gereksinimlerin ve önerilerin ortak bir payda çerçevesinde belirlenmesi

Uygulama Güvenliği Doğrulama Seviyeleri

ASVS, üç çeşit doğrulama seviyesi belirlemektedir. Doğrulamanın derinliği, seviye arttıkça artmaktadır.

- ASVS Seviye 1, tüm yazılımlar içindir.
- ASVS Seviye 2, koruma gerektiren ve aynı zamanda hassas veri içeren uygulamalar içindir.
- ASVS Seviye 3, en kritik yazılımlar içindir. Bu yazılımlar yüksek miktarlarda (değeri yüksek) ödemeler gerçekleştiren, hassas tıbbi veriler içeren veya yüksek düzeyde güvenlikli olması gereken yazılımlardır.

Her bir ASVS seviyesinde güvenlik gereksinimlerinin bir listesi bulunur. Bu gereksinimlerden her biri, güvenlik için gerekli özelliklerle ve yazılım geliştiricileri tarafından sistem üzerinde yapılması gereken değişikliklerle eşleştirilebilir.



Figür 1 - OWASP Application Security Verification Standard 4.0 Seviyeleri

Seviye 1, insanlar tarafından tamamen sızma (sızma) testi içerisinde uygulanabilir olan tek seviyedir. Diğer seviyelerin tümü uygulamaya ait dokümanlara, uygulama kaynak koduna, yapılandırmaya ve geliştirme sürecine dahil olan kişilere erişim gerektirir. Ancak, L1 "kara kutu" (dokümantasyon ve kaynak kod olmadan) sızma testi yapılmasına izin verse bile, bu sızma testi sonucu etkili bir güvence değildir. Kötü niyetli saldırganların çok zamanları vardır ancak sızma testlerinin çoğu birkaç hafta içinde sona erer. Savunucuların; güvenlik kontrolleri oluşturması, tüm zayıflıkları koruması, bulması/çözmesi, kötü niyetli aktörleri makul bir sürede tespit etmesi ve bunlara yanıt vermesi gerekir. Kötü niyetli aktörler esasen sonsuz zamana sahiptir ve başarılı olmak için sadece tek bir zayıf savunma, tek bir zayıflık veya bir hatalı güvenlik anlayışını tespit etmesi yeterlidir. Genellikle geliştirme sonunda hızlı bir şekilde yapılan veya hiç yapılmayan kara kutu testleri bu asimetri ile baş edemez.

Son 30 yıldan fazla bir süredir, kara kutu testinin, giderek daha büyük ihlallere yol açan kritik güvenlik sorunlarını gözden kaçırdığı defalarca kanıtlanmıştır. Sızma testi uzmanlarına, geliştirme aşaması boyunca geliştiricilere ve belgelere tam erişim sağlanmasına ek olarak Seviye 1'deki kaynak kodu yönlendirmeli (hibrit) sızma testlerinin gerçekleştirilmesini şiddetle tavsiye ediyoruz. Mali düzenleyiciler; kitaplara, örnek işlemlere veya denetimleri yapan kişilere erişimi olmayan dış denetimlere tahammül etmez. Endüstri ve hükümetler de, yazılım mühendisliği alanında aynı şeffaflık standardını talep etmelidir.

Geliştirme süreci ve “build pipeline” içinde, DAST ve SAST araçları gibi güvenlik araçlarının sürekli kullanımını güvenlik sorunlarını bulmak için şiddetle tavsiye ediyoruz.

Otomatik araçlar ve çevrimiçi taramalar ASVS'nin yarısından fazlasını insan yardımı olmadan tamamlayamaz. Her derleme için kapsamlı test otomasyonu gerekiyorsa, derlemenin başlattığı çevrimiçi taramaların yanı sıra özel birim (unit) ve entegrasyon testlerinin bir kombinasyonu kullanılmalıdır. İş mantığı kusurları ve erişim kontrolü testi yalnızca insan yardımı ile mümkündür. Bunlar birim ve entegrasyon testlerine dönüştürülmelidir.

Standart Nasıl Kullanılmalı?

ASVS’yi kullanmak için en iyi sebeplerden birisi; uygulama, platform ya da kuruluşa özgü Güvenli Kodlama Kontrol Listesi (Secure Coding Checklist) oluşturmaktır. ASVS’yi kendi kullanım durumunuza (use-case) göre ayarlamak; projenizin ve ortamınızın (environment) güvenlik gereksinimlerine olan hassaslığını artıracaktır ve güvenlik gereksinimlerini daha görünür hale getirecektir.

Seviye 1: Fırsatçı (Opportunistic)

Bir uygulamanın, tespit edilmesi kolay olan zafiyetlere karşı (OWASP Top 10 vb.) aldığı güvenlik önlemleri yeterli ise seviyesi ASVS Seviye 1’dir.

Seviye 1 tipik olarak güvenlik kontrollerinin yerinde kullanıldığı konusunda bir güven oluşturulmasına ihtiyaç duyulan uygulamalar için uygundur. Bunun yanı sıra kurumsal uygulamaların tamamına genel bir bakış açısıyla, daha ilerideki denetimler için bir yol haritası çıkarmak amacıyla Seviye 1 kullanılabilir. Seviye 1 uygulamalarının kontrolü, otomatize araçlar yardımı ile ve kaynak kod olmadan manuel olarak gerçekleştirilebilir. Tüm uygulamalar için Seviye 1 dikkate alınmalıdır.

Uygulamalara doğrultulan tehditler genel olarak; tespit edilmesi ve istismar edilmesi kolay zafiyetleri belirlemek için basit teknikler kullanan saldırganlar tarafından gerçekleştirilmektedir. Bu saldırganlar doğrudan uygulamayı

hedef alıp üzerine yoğunlaşmazlar. Eğer uygulama tarafından işlenen veriler kritik değere sahipse nadiren de olsa Seviye 1 sürecinde durmak isteyeceksinizdir.

Seviye 2: Standart

Seviye 2'deki bir uygulama, ortalama ve ciddi seviyede risklere yol açan zafiyetlere karşı yeterli seviyede korumalıdır.

Seviye 2, uygulanan güvenlik kontrollerinin yerinde ve etkili kullanıldığını ve bu sayede uygulamaya özel politikaların zorlandığını garanti eder. Bu seviye tipik olarak, işten-işe hassas aktarım (transaction) yapan sağlık bilgisi, hassas fonksiyonlar ve diğer hassas bilgileri içeren uygulamalar için uygundur.

Bu seviyedeki tehditler genellikle yetenekli ve motive olmuş saldırganlar tarafından oluşturulan, elle yapılan test tekniklerini içeren tehditlerdir. Uygulamalar içerisindeki zafiyetleri keşfetmek ve istismar etmek için oldukça etkili araçlar ve teknikler kullanabilmektedirler.

Seviye 3: Gelişmiş (Advanced)

Seviye 3, ASVS seviyeleri içerisinde en yüksek seviyede olandır. Tipik olarak; yaşam ve güvenliği koruyan kritik uygulamalar, hassas bilgileri işleyen uygulamalar, kritik altyapıya sahip ya da savunma mekanizmaları ve istismar edilmesi durumunda organizasyona büyük derecede zarar verecek uygulamalar için Seviye 3 uygundur.

Bu seviyedeki tehditler, saldırı amaçlı kullanılan özel tarama araçları ile uygulamayı istismar etmeye çalışan, odaklanmış ve uzman saldırganlar tarafından oluşturulur. 3. seviyedeki bir uygulama, tüm gelişmiş uygulama güvenliği zafiyetlerine karşı korunmalı ve ayrıca iyi bir güvenlik tasarımına sahip olmalıdır.

Seviye 3 bir uygulama derinlemesine analize, mimariye, kodlamaya ve diğer tüm adımların testine ihtiyaç duyar. Güvenli bir uygulama doğru bir şekilde birimlere ayrılmalıdır. Kolaylaştırma, esneklik, ölçeklenebilirlik ve diğer tüm güvenlik katmanlarına ve her modül (ağ bağlantısı ve / veya fiziksel örneği ile ayrılmalı) güvenlik sorumluluklarını önemsemek için iyi bir belgelendirmeye ihtiyaç duyar. Bu görevler gizlilik (örneğin şifreleme), bütünlük (örneğin işlemler, girdi doğrulama), erişilebilirlik, kimlik doğrulama (sistemler arası), tanınmayan yetkilendirme ve günlük denetim (Loglama) özelliklerini içermektedir.

ASVS'yi Pratikte Uygulamak

Farklı tehditler farklı motivasyonlara sahiptir. Bazı endüstriler, benzersiz teknoloji varlıklarına, benzersiz bilgilere ve benzersiz uyumluluk denetimlerine sahiptir.

Yazılımın Değerlendirilmesi ve Bir Doğrulama Seviyesine Ulaşması

OWASP'nin ASVS Yetkilendirmeleri ve Güven Duyulan Marka Duruşu

OWASP, sağlayıcı ayrımı gözetmeyen ve kâr amacı gütmeyen bir organizasyon olarak, herhangi bir sağlayıcıyı, doğrulayıcıyı veya yazılımı onaylamaz.

Tüm bu güvence beyanları, güven duyulan markalar ya da sertifikalar resmi olarak OWASP tarafından incelenmediğinden; ASVS sertifikası sağlayacağını iddia eden üçüncü parti bir kuruluşa ya da şahsa güven konusunda temkinli olunmalıdır.

Bu, resmi OWASP sertifikasını talep etmedikleri sürece kuruluşların bu tür güvence hizmetleri sunmasını engellememelidir.

Organizasyonları Yetkilendirmek İçin Rehberlik

Uygulama Güvenliği Doğrulama Standardı, uygulamanın açık kitap doğrulaması olarak kullanılabilir. Kaynaklara açık ve serbest olarak erişimi olan yazılım mimarları ve geliştiriciler gibi, proje dokümantasyonu, kaynak kodu, test sistemlerine kimliği doğrulanmış bir şekilde erişim (her bir rol için en az bir erişim hesabı dahil) ve özellikle L2 ve L3 doğrulama seviyeleri için kullanılabilir.

Tarihsel olarak, sızma testi ve kaynak kod analizinin sorunları içerisinde yalnızca istisnai durumlarda başarısız konular final raporda görünmektedir. Bir yetkilendirme organizasyonu herhangi bir raporda doğrulama kapsamını belirtmelidir. Özellikle SSO yetkilendirmesi gibi önemli bir bileşen kapsam dışındaysa; başarılı ve başarısız testleri de içeren ve bununla birlikte başarısız testlerin nasıl çözüleceğine ilişkin yöntemler de açıkça belirtilmelidir.

Detaylı dokümantasyonu, ekran ve video görüntülerini, zafiyetleri istismar edebilen programları (exploit), araya girmede kullanılan vekil (proxy) araç kayıtları gibi elektronik kayıtları saklamak kabul görmüş bir standarttır. Bu veriler, zafiyetlere itiraz eden geliştiricileri ikna etmek için bulguların kanıtını göstermek adına faydalı olabilir. Bir aracı çalıştırmak ve bulguları raporlamak yeterli değildir. Bu, belirli doğrulama seviyelerindeki gereksinimlerin sınandığı ve yeterli düzeyde test edildiğine dair yeterli kanıt sağlamamaktadır. Anlaşmazlık (itiraz) olduğu durumlarda, her doğrulama gereksiniminin gerçekten test edildiğine dair yeterince kanıt bulunmalıdır.

Test Metodolojisi

Sertifika veren kuruluşlar uygun test yöntemlerini seçmekte serbesttir ancak bu yöntemler rapor içerisinde belirtilmelidir.

Test edilen uygulamaya veya kontrol gerekliliklerine bağlı olarak farklı test metotları kullanılabilir. Örneğin bir uygulamanın girdi doğrulama mekanizmasının etkinliğinin doğrulanması için, manuel bir sızma testi veya kaynak kod analizi yapılabilir.

Otomatik Sızma Testi Araçlarının Rolü

Otomatik sızma testi araçlarının olabildiğince geniş kapsam sağlamaları ve mümkün olduğunca farklı zararlı girdi biçimleriyle çok sayıda farklı form ve parametreyi denetlemeleri desteklenmelidir.

Otomatik sızma testi araçlarını tek başına kullanarak ASVS doğrulamasını tam anlamıyla yapmak mümkün değildir. L1 (seviye 1)'deki gereksinimlerin büyük çoğunluğu otomatik test araçları ile yapılabilirken kalan çoğu gereksinimler otomatikleştirilmiş sızma testine dahil değildir.

Uygulama güvenliği endüstrisinin gelişmesiyle birlikte; otomatize ve elle (manuel) yapılan testlerin arasındaki çizginin inceldiğini lütfen unutmayın. Otomatik araçlar genelde uzmanlar tarafından el ile (manuel) ayarlanır ve el ile (manuel) test yapan uzmanlar da genellikle otomatik araçlardan yararlanırlar.

Sızma Testinin Rolü

Sürüm 4.0'da; uygulama kaynak koduna, uygulamaya ait belgelere veya uygulama geliştiricilerine erişim olmadan L1'i tamamen sadece sızma testi ile kontrol edilebilir hale getirmeye karar verdik. Ancak OWASP Top 10 2017 A10'a uymak için gereken iki öğe, tıpkı OWASP Top 10 2017'de olduğu gibi uygulama geliştiricisiyle yapılan karşılık görüşmeler, ekran görüntüleri veya diğer kanıtların toplanmasını gerektirecektir. Kaynak kod analizleri, tehditlerin ve eksik kontrollerin belirlenmesi ve daha kısa bir zaman diliminde çok daha kapsamlı bir test yapma olasılığını kaçırdığı için gerekli bilgilere erişim olmadan sadece sızma testi yapmak ideal bir yöntem değildir.

L2 veya L3 Değerlendirmesi yapılırken, mümkün ise; uygulama geliştiricilerine, uygulama ile ilgili dokümantasyonlara, uygulama kaynak koduna ve üretim dışı veriler içeren bir test uygulamasına erişim gerekir. "hibrit incelemeler" veya "hibrit sızma testleri" bu erişim düzeyini gerektirir.

ASVS Standardının Diğer Kullanım Alanları

Bir uygulamanın güvenliğini değerlendirmek için kullanılmasının yanı sıra, ASVS için bir dizi başka potansiyel kullanım belirledik.

Detaylı Güvenlik Mimarisi Rehberi Olarak

Uygulama Güvenliği Doğrulama Standardı, güvenlik mimarlarına (security architects) tarafından yaygın olarak kullanılmaktadır. Önemli güvenlik mimari çatılarından (framework) olan SABSA; uygulama güvenliği mimarisinin incelenmesinin tamamlanabilmesi için çokça eksik bilgiye sahiptir. ASVS, güvenlik mimarlarının veri koruma yöntemleri ve girdi doğrulama stratejileri gibi ortak sorunları için daha iyi denetimler seçmesine izin vererek bu eksiklikleri gidermektedir.

Hazır Güvenli Kodlama Denetim Listeleri Yerine

Çoğu kuruluş üç seviyeden birini seçerek veya her uygulamanın gereksinimine göre risk seviyelerini kendine özgü bir şekilde değiştirerek ASVS'den faydalanabilir. İzlenebilirlik sağlanabildiği sürece bu tür değişiklikleri destekliyoruz. Şayet bir uygulama 4.1 gereksinimini sağlıyorsa; bu hem orijinal hem de değiştirilmiş ASVS için aynı anlama gelmektedir.

Otomatik Birim ve Entegrasyon Testleri Rehberi Olarak

ASVS, mimari ve kötü amaçlı kod gereksinimleri dışında son derece test edilebilir şekilde tasarlanmıştır. İstismar vakalarının simüle edilebileceği özelleştirilmiş birim ve entegrasyon testleri ile uygulama her geliştirmede neredeyse kendi kendini doğrulayabilecek hale gelebilir. Örneğin, bir oturum açma denetleyicisi için ek testler hazırlanabilir. Bunlar, sık kullanılan kullanıcı adları, hesap numaralandırma (enumeration), kaba kuvvet, LDAP ve SQL enjeksiyonu, XSS gibi ek sınamalar işlenebilir. Benzer şekilde parola parametresi için sık kullanılan parolalar, parola uzunluğu, boş (null) byte enjeksiyonu, parametre silme, XSS, hesap numaralandırma ve benzeri zafiyetlere karşı sınanmalıdır.

Güvenli Yazılım Geliştirme Eğitim Kaynağı Olarak

ASVS, güvenli yazılımın karakteristiğini tanımlamakta da kullanılabilir. Çoğu "güvenli kodlama" eğitimleri, saldırı temelli eğitimlerdir ve kodlama ipuçlarına fazla yer verilmez. Bunun geliştiricilere bir yardımı dokunmaz. Bunun yerine güvenli geliştirme kursları, yapılmaması gereken 10 olumsuz şey yerine, ASVS'de bulunan koruyucu kontrollere yoğun bir şekilde odaklanmalıdır.

Agile (Çevik) Uygulama Güvenliği Öncüsü Olarak

ASVS, güvenli bir ürüne sahip olmak için ekip tarafından uygulanması gereken görevleri tanımlamak için çevik geliştirme sürecinde bir çerçeve olarak kullanılabilir.

Örnek olarak: Seviye 1'den başlayarak; uygulama veya sistemi ilgili seviye için ASVS gereksinimlerine göre doğrulayın, hangi kontrollerin eksik olduğunu bulun ve bu eksikleri kontrol listesine ekleyin. Bu, belirli görevlerin önceliklendirilmesine yardımcı olur ve çevik süreçte güvenliği görünür hale getirecektir. Ayrıca belirli bir ASVS gereksiniminin, sorumlu ekip üyesi için bir sürücü olabileceği unutulmamalıdır. İlgili kontrol maddesinin kontrol listesinde "yapılacak" olarak görülmesi ilgili işe öncelik verilmesi sağlanır.

Güvenli Yazılım Tedarikine Yönelik Bir Çerçeve Olarak

ASVS, güvenli yazılım tedarikine veya özel geliştirme hizmetlerinin tedarikine yardımcı olmak için harika bir çerçevedir. Alıcı, tedarik etmek istedikleri yazılımın uygun gördükleri ASVS seviyesine uygun geliştirilmesi gerektiğine dair bir gereksinim belirleyebilir ve satıcının yazılımın bu ASVS seviyesini karşıladığını kanıtlamasını isteyebilir. Bu yöntem OWASP Güvenli Yazılım Sözleşmesi Eki ile kullanılabilir.

V1: Mimari, Tasarım ve Tehdit Modelleme

Kontrolün Amacı

Güvenlik mimarisi birçok organizasyonda neredeyse kaybolmuş bir sanat haline gelmiş ve kurumsal mimarinin yerini DevSecOps almıştır. Uygulama güvenliği alanı, yazılım geliştiricilerine önde gelen güvenlik mimarisi ilkelerini tanıtırken, çevik güvenlik ilkelerini de yakalamalı ve benimsemelidir. Uygulama mimarisi sadece yerine getirilecek bir uygulama olarak değil, potansiyel birçok farklı yanıtı olan ve tek bir "doğru" yanıtı olmayan bir problem üzerinde düşünmenin bir yoludur. Güvenlik çoğu zaman, geliştiricilerin sorunu çözmenin yazılımsal olarak çok daha iyi bir yolunu bilmelerine rağmen; kodu farklı bir şekilde düzeltmelerini gerektiren, esnek olmayan ve talepkar bir yaptırım olarak görülür. Mimari için tek ve basit bir çözüm yoktur zaten aksi durum yazılım mühendisliği alanını olumsuz etkileyecektir.

Bir web uygulamasının kullanım ömrü boyunca sürekli olarak gözden geçirilmesi muhtemeldir, ancak genel mimari nadiren değişecek ve yavaş gelişecektir. Güvenlik mimarisi de aynıdır. Bugün kimlik doğrulamaya ihtiyacımız var, yarın kimlik doğrulamaya ihtiyacımız olacak ve bundan beş yıl sonra da buna ihtiyacımız olacak. Bugün sağlam kararlar alırsak, mimari olarak uyumlu çözümleri seçip kullanırsak; çok fazla çaba, zaman ve para tasarrufu sağlayabiliriz. Örneğin, bundan on yıl önce çok faktörlü kimlik doğrulaması nadiren uygulanırdı.

Geliştiriciler SAML gibi tek ve güvenli bir kimlik sağlayıcı modeline yatırım yapmışlarsa, orijinal uygulamanın arabirimlerini değiştirmeksizin, kimlik sağlayıcı NIST 800-63 uyumluluğu gibi yeni gereksinimleri içerecek şekilde güncellenebilir. Birçok uygulama aynı güvenlik mimarisi altyapısını ve dolayısıyla aynı bileşeni paylaşıyorsa, hepsi bu yükseltmeden hemen yararlanmış olacaktır. Bununla birlikte, SAML her zaman en iyi veya en uygun kimlik doğrulama çözümü olarak kalmayacaktır, gereksinimler değiştikçe diğer çözümler ile değiştirilmesi gerekebilir. Bunun gibi değişiklikler karmaşıktır, tüm kodun yeniden yazılmasını gerektirecek kadar maliyetli olabilir ve bu durum bir güvenlik mimarisi olmadan açıkça imkansızdır.

Bu bölümde ASVS, sağlam güvenlik mimarisinin temel unsurlarını ele alır; kullanılabilirlik, gizlilik, işlem bütünlüğü, reddedilememe ve gizlilik bu bölümde ele alınmıştır. Bu güvenlik ilkelerinin her biri yerleşik olmalı ve tüm uygulamalar için temelde uygulanmış olmalıdır. Erken yazılım geliştirme döngüsünde öncelikle; güvenli kodlama kontrol listelerinin oluşturulması, güvenli uygulama rehberliği ve eğitiminin alınması gerekmektedir. Sonrasında kodlama, testler, derleme, dağıtım, yapılandırma gerçekleştirilmeli ve son olarak tüm güvenlik kontrollerinin yapıldığına ve işlevsel olduğuna emin olmak için bağımsız güvenlik testlerinin gerçekleştirilmesi gerekmektedir.

Son adım endüstri olarak uyguladığımız tek şeydi, ancak geliştiriciler kodu günde onlarca veya yüzlerce kez ürettiğinde artık yeterli değil. Bu yüzden uygulama güvenliği uzmanları çevik tekniklere ayak uydurmalıdır. Çevik geliştirme tekniğine ayak uydurmak, uygulama tamamlandığında veya tamamlanmaya yaklaştığında projeyi eleştirmek yerine geliştirici araçlarını benimsemek, kodlamayı öğrenmek ve geliştiricilerle çalışmak anlamına gelir.

V1.1 Güvenli Yazılım Geliştirme Yaşam Döngüsü Gereksinimleri

#	Açıklama	L1	L2	L3	CWE
1.1.1	Güvenliği tüm aşamalarında adresleyen bir güvenli yazılım geliştirme yaşam döngüsünün kullanıldığı doğrulanmalıdır.		✓	✓	
1.1.2	Güvenlik tehditlerini tanımlamak, karşı önlemleri planlamak, risklere yönelik yapılabilecek işlemleri belirlemek ve güvenlik testlerini yönlendirmek için her tasarım değişikliği veya sprint planlaması için tehdit modellemesinin kullanımı doğrulanmalıdır.		✓	✓	1053
1.1.3	Tüm kullanıcı öykülerinin ve özelliklerinin, "Bir kullanıcı olarak profilimi görüntüleyebilmeli ve düzenleyebilmeliyim. Başka birinin profilini görüntüleyemem ya da düzenleyemem" gibi işlevsel güvenlik kısıtlamaları içerdiği doğrulanmalıdır.		✓	✓	1110

#	Açıklama	L1	L2	L3	CWE
1.1.4	Uygulamaya ait tüm güven sınırları, bileşenleri ve önemli veri akışlarının dokümantasyonu ve gerekçelendirilmesi doğrulanmalıdır.		✓	✓	1059
1.1.5	Uygulamanın üst düzey mimarisinin ve bağlı tüm uzak servislerin tanımının ve güvenlik analizinin bulunması gerekmektedir.		✓	✓	1059
1.1.6	Yinelenen, eksik, etkisiz ve/veya güvensiz kontrolleri önlemek için merkezi, basit, onaylanmış, güvenli ve yeniden kullanılabilir güvenlik kontrollerinin uygulandığı doğrulanmalıdır.		✓	✓	637
1.1.7	Tüm yazılım geliştiriciler ve test uzmanlarının güvenli kodlama kontrol listesi, güvenlik gereksinimleri, kılavuz veya politikasına erişiminden emin olunmalıdır.		✓	✓	637

V1.2 Kimlik Doğrulaması Mimari Gereksinimleri

Kimlik doğrulama mimarisi tasarlarken örneğin donanım etkinleştirilmiş çok faktörlü kimlik doğrulamanızın olması, bir saldırgan çağrı merkezini arayarak ve yaygın olarak bilinen soruları yanıtlayıp hedef aldığı hesabı sıfırlayabiliyorsa önemsiz hale gelmektedir. Kimlik ispatı yapılırken, tüm kimlik doğrulama yolları aynı güce sahip olmalıdır.

#	Açıklama	L1	L2	L3	CWE
1.2.1	Her bir uygulama bileşeni, servis ve sunucuda kullanılacak işletim sistemi hesaplarının kullanıma özel veya en az yetki prensibine uygun olarak yapılandırılması gerekmektedir.		✓	✓	250
1.2.2	API'ler, ara katman yazılımı ve veri katmanları da dahil olmak üzere uygulama bileşenleri arasındaki iletişimin yetki/erişim/kimlik doğrulaması ile yapıldığı doğrulanmalıdır. Bileşenler sadece gereken işlemi yapabilecek kadar yetkiye sahip olmalıdır.		✓	✓	306
1.2.3	Uygulamanın güvenli olduğu bilinen tek bir denetimli kimlik doğrulama mekanizması kullandığı, güçlü kimlik doğrulaması içerecek şekilde genişletilebildiği ve herhangi bir hesabın kötüye kullanılması veya ihlal durumunu tespit etmek için yeterli iz kaydı ve izlemeye sahip olduğu doğrulanmalıdır.		✓	✓	306
1.2.4	Tüm kimlik doğrulama metotları doğrulanmalı, kimlik yönetimi API'lerinin zayıf güvenlik kontrol alternatiflerini seçip uygulama için risk oluşturmadığına emin olunmalıdır.		✓	✓	306

V1.3 Oturum Yönetimi Mimari Gereksinimleri

Bu bölüm, gelecek mimari gereksinimler için ayrılmıştır.

V1.4 Erişim Kontrolü Mimari Gereksinimler

#	Açıklama	L1	L2	L3	CWE
1.4.1	Erişim kontrol işlemlerinin ağ geçitleri (gateways), sunucular veya sunucusuz fonksiyonlar gibi güvenilir uygulama noktalarında olmasını sağlayın. Asla erişim denetimlerini istemci tarafında uygulamayın.		✓	✓	602
1.4.2	Seçilen erişim kontrol çözümünün uygulamanın ihtiyaçlarını karşılayabilecek esneklikte olmasına dikkat edin.		✓	✓	284

#	Açıklama	L1	L2	L3	CWE
1.4.3	Fonksiyonlar, veri dosyaları, URL'ler, servisler ve diğer kaynaklar için minimum yeterli yetki verilmesine dikkat edin ve ihtiyaç dışında bir yetki verilmediğini doğrulayın. Bu sayede kimlik sahtekarlığı ve hak yükseltme gibi konulara karşı koruma sağlanabilir.		✓	✓	272
1.4.4	Uygulamanın, korunan veri ve kaynaklara erişiminde tek ve güvenilir bir erişim kontrol mekanizması kullanması gerekmektedir. Tüm istekler bu erişim kontrol mekanizmasından geçirilerek kopyala-yapıştır istekler ve güvensiz alternatif erişim yöntemleri engellenmelidir.		✓	✓	284
1.4.5	Bir kullanıcının herhangi bir veriye erişiminde sadece kullanıcı rolüne bakmak yerine öznitelik ve özellik tabanlı olarak ilgili kullanıcının ilgili veri bileşeni/özelliğe yetkisinin kontrolü sağlanmalıdır. İzinler yine de kullanıcı rolleri kullanılarak tahsis edilmelidir.		✓	✓	275

V1.5 Girdi ve Çıktı Mimari Gereksinimleri

Güven sınırı konusu hala endişe vericidir, tarayıcılarda veya istemci cihazlarda uygulama akışına karar vermek atlatılabilir bir çözümdür. Bununla birlikte, günümüzün ana akım mimari uygulamalarında, güven uygulama noktası önemli ölçüde değişti. Bu nedenle, ASVS'de "güvenilir hizmet katmanı" terimi kullanıldığında; mikro servis, sunucusuz API, sunucu tarafı, güvenli önyüklemeli bir istemci cihazda güvenilir bir API, iş ortağı veya harici API'ler gibi konumdan bağımsız olarak herhangi bir güvenilir uygulama noktasını kastediyoruz.

#	Açıklama	L1	L2	L3	CWE
1.5.1	Girdi ve çıktı ihtiyaçlarının, verilerin tür, içerik, geçerli yasalar ve kanunlara göre nasıl işlenip kullanılacağının açık bir şekilde belirlendiği doğrulanmalıdır.		✓	✓	1029
1.5.2	Güvenilmeyen istemcilerle iletişimde serileştirme kullanılmamalıdır. Eğer bu mümkün değilse, yeterli bütünlük kontrolleri (ve eğer hassas veri var ise bununla birlikte şifreleme de) yapılması zorunlu tutularak nesne enjeksiyonu gibi serileştirme saldırıları önlenmelidir.		✓	✓	502
1.5.3	Girdi kontrollünün güvenilen bir servis katmanında yapılması gerekmektedir.		✓	✓	602
1.5.4	Çıktı karakter kodlamasının (encoding) yorumlayıcının kendisi tarafından veya yakın bir katmanda yapılması sağlanmalıdır.		✓	✓	116

V1.6 Kriptografik Mimari Gereksinimler

Uygulama verilerinin hassasiyet sınıflandırmalarına göre korunması için güçlü bir şifreleme mimarisiyle tasarlanması gerekir. Her şeyi şifrelemek israf olacaktır ancak hiçbir şeyi şifrelememek ise yasal olarak ihmaldir. Genellikle; üst düzey mimari tasarım, tasarım sprintleri veya mimari sivri uçlarda bir denge kurulmalıdır. Kriptografik mimariyi uygulama geliştirildikten sonra tasarlamak veya güçlendirmek, başlangıçtan itibaren oluşturmaktan daha pahalıya mal olacaktır.

Mimari gereksinimleri kod tabanının tamamına özgüdür ve bu nedenle testlerini birleştirmek veya entegre etmek zordur. Mimari gereksinimler, kodlama aşaması boyunca kodlama standartlarında dikkate alınmalıdır. Ek olarak güvenlik mimarisi, ikili kod incelemeleri veya geriye dönük olarak periyodik olarak gözden geçirilmelidir.

#	Açıklama	L1	L2	L3	CWE
1.6.1	Şifreleme anahtarlarının nasıl yönetileceğine dair açık / belirgin bir politika olduğu doğrulanmalıdır ve bu politikanın da NIST SP 800-57 gibi bir anahtar yaşam döngüsü yönetim standardı tabanlı olması sağlanmalıdır.		✓	✓	320
1.6.2	Tüm kriptografik servis kullanıcılarının, anahtar materyallerini ve diğer gizli bilgileri / materyalleri anahtar kasalarında veya API tabanlı alternatiflerde sakladıkları doğrulanmalıdır.		✓	✓	320
1.6.3	Tüm şifreleme anahtarların ve parolaların değiştirilebilir olduğu ve hassas verileri yeniden şifrelemek için iyi tanımlanmış bir sürecin parçası olduğu doğrulanmalıdır.		✓	✓	320
1.6.4	İstemciler tarafından oluşturulan veya onlarla paylaşılan simetrik şifreleme anahtarları, parolalar ve API sırlarının yalnızca yerel depolama veya parametre karıştırma benzeri geçici verilerin şifrelenmesi gibi düşük riskli işlerde kullanılması doğrulanmalıdır. Müşterilerle sırların açık metin paylaşımı da mimari olarak yukarıdaki gibi olmalıdır.		✓	✓	320

V1.7 Hata, Loglama ve Denetim Mimari Gereksinimleri

#	Açıklama	L1	L2	L3	CWE
1.7.1	Sistem genelinde ortak bir loglama / kayıt biçimi ve yaklaşımının kullanıldığı doğrulanmalıdır.		✓	✓	1009
1.7.2	Logların, tercihen uzaktaki bir sisteme güvenli bir şekilde analiz, algılama, uyarı ve yönlendirme işlemleri için iletilmesi doğrulanmalıdır.		✓	✓	

V1.8 Veri Koruma ve Gizlilik Mimari Gereksinimleri

#	Açıklama	L1	L2	L3	CWE
1.8.1	Tüm hassas verilerin tanımlanması sağlanmalı ve koruma düzeylerinde sınıflandırıldığı doğrulanmalıdır.		✓	✓	
1.8.2	Tüm veri koruma seviyelerinin, şifreleme gereksinimleri, bütünlük gereksinimleri, saklama, veri gizliliği ve diğer gizlilik gereksinimlerine göre ilişkili bir kümede olması ve mimaride bu durumun uygulanması gerekmektedir.		✓	✓	

V1.9 İletişim Mimari Gereksinimleri

#	Açıklama	L1	L2	L3	CWE
1.9.1	Uygulamanın, özellikle bileşenler farklı konteynerlerde, sistemlerde, sitelerde veya bulut sağlayıcılarında olduğunda bileşenler arasındaki iletişimi şifrelemesi doğrulanmalıdır.		✓	✓	319
1.9.2	Uygulama bileşenlerinin iletişiminde ortadaki adam saldırılarını önlemek adına kimlik doğrulaması yapılmalıdır. Örneğin, uygulama bileşenleri TLS sertifikaları ve zincirlerini doğrular.		✓	✓	295

V1.10 Kötücül Yazılım Mimari Gereksinimleri

#	Açıklama	L1	L2	L3	CWE
1.10.1	Kaynak kod kontrol sistemlerinin, tüm aktarma (check-in) işlemlerini belirli bir sorun kaydı veya değişiklik talebine ilişkilendirmesi gerekmektedir. Kaynak kod		✓	✓	284

kontrol sistemi, herhangi bir değişikliğin izlenebilirliğini sağlamak için erişim kontrolüne ve tanımlanabilir kullanıcılara sahip olmalıdır.

V1.11 İş Mantığı Mimari Gereksinimleri

#	Açıklama	L1	L2	L3	CWE
1.11.1	Uygulamaya ait tüm bileşenlerin sağladıkları iş veya güvenlik fonksiyonlarına dair tanımlarının yapılması ve belgelendirmenin sağlanması beklenmektedir.		✓	✓	1059
1.11.2	Kimlik doğrulama, oturum yönetimi ve erişim kontrolü gibi tüm yüksek önemdeki iş mantığı akışlarının senkronize olmayan bir yapıda olmadığı doğrulanmalıdır.		✓	✓	362
1.11.3	Kimlik doğrulama, oturum yönetimi ve erişim kontrolü gibi tüm yüksek önemdeki iş mantığı akışlarının thread-safe yapıda olması ve TOC-TOU ataklarına karşı dirençli olması gerekmektedir.			✓	367

V1.12 Güvenli Dosya Yükleme Mimari Gereksinimleri

#	Açıklama	L1	L2	L3	CWE
1.12.1	Kullanıcıların sisteme yüklediği dosyaların web root klasörü dışında bir yerde tutulduğu doğrulanmalıdır.		✓	✓	552
1.12.2	Gerekli durumlarda kullanıcının yüklediği dosyaların, uygulamada görüntülenmesi veya indirilmesi gerekiyorsa (oktet akışı indirmeleri veya bir bulut depolama alanı gibi ilgisiz bir etki alanı olabilir) XSS veya yüklenen dosyadaki diğer güvenlik risklerini önleyebilmek için uygun bir içerik güvenlik politikası (Content Security Policy) uygulanmalıdır.		✓	✓	646

V1.13 API Mimari Gereksinimleri

Bu bölüm, gelecek mimari gereksinimler için ayrılmıştır.

V1.14 Konfigürasyon Mimari Gereksinimleri

#	Açıklama	L1	L2	L3	CWE
1.14.1	Farklı güven seviyelerinde bulunan bileşenlerin güvenlik duvarı kuralları, ağ bölümlendirme, ters vekil sunucusu (reverse proxy) veya bulut tabanlı güvenlik grupları yoluyla birbirinden ayrı tanımladığından emin olunmalıdır.		✓	✓	923
1.14.2	Eğer derlenmiş (binary) dosyalar, güvenilmeyen cihazlara kuruluyorsa, binary imzalama, güvenilir bağlantılar ve doğrulanmış uçların (endpoint) varlığı doğrulanmalıdır.		✓	✓	494
1.14.3	Derleme akışının (pipeline) eski ve güvensiz bileşenler için uyarı ürettiğini ve gerekli aksiyonları aldığından emin olunmalıdır.		✓	✓	1104
1.14.4	Derleme akışının otomatik derleme adımı içermesi ve özellikle uygulama için uygulamanın güvenli bir şekilde dağıtımının (üretim ortamına alındığının) sağlandığı doğrulanmalıdır.		✓	✓	
1.14.5	Uygulama dağıtımlarının/sürüm geçişlerinin, özellikle de serileştirme gibi hassas veya tehlikeli eylemler gerçekleştirilmesi ile saldırganların diğer uygulamalara saldırmasını geciktirmek ve caydırmak için yeterince kum havuzuna alımı (sandbox), konteyner ile kullanım ve / veya ağ katmanı düzeyinde yalıtımlı olduğu doğrulanmalıdır.		✓	✓	265

1.14.6	Uygulamanın NSAPI eklentileri, Flash, Shockwave, ActiveX, Silverlight, NACL veya istemci tarafı Java appletleri gibi desteklenmeyen, güvensiz veya kullanımdan kaldırılmış istemci tarafı teknolojileri kullanmadığı doğrulanmalıdır.	✓	✓	477

Referanslar

Detaylı bilgi için aşağıdaki adresleri ziyaret edebilirsiniz:

- OWASP Threat Modeling Cheat Sheet
- OWASP Attack Surface Analysis Cheat Sheet
- OWASP Threat modeling
- OWASP Secure SDLC Cheat Sheet
- Microsoft SDL
- NIST SP 800-57

V2: Kimlik Doğrulama Gereksinimleri

Kontrol Amacı

Kimlik doğrulama, bir kişinin (veya bir şeyin) özgün olarak oluşturulması (tanımlanması) ve onaylanması işlemidir. Kimlik doğrulama bir kişi veya bir cihaz tarafından yapılan hak taleplerinin doğruluğunu kontrol eder. Kimlik doğrulama kimliğe bürünmeye karşı dirençlidir ve şifrelerin güvensiz hale geri dönüştürülmesini veya ele geçirilmesini engeller.

ASVS ilk kez piyasaya sürüldüğünde, kullanıcı adı ve şifre doğrulaması yüksek güvenlik sistemleri dışındaki en yaygın kimlik doğrulama biçimiydi. Çok faktörlü kimlik doğrulama (MFA) güvenlik çevrelerinde yaygın olarak kabul edilmesine rağmen nadiren kullanılırdı. Parola ihlallerinin sayısı arttıkça, kullanıcı adlarının gizli olduğu ve parolaların bilinmediği fikri savunulamaz hale geldi. Örneğin; NIST 800-63, kullanıcı adlarını ve bilgi tabanlı kimlik doğrulama (KBA) verilerini herkese açık bilgi olarak kabul eder. SMS ve e-posta bildirimlerini ise "kısıtlı" kimlik doğrulayıcı türleri olarak görür. Şifreleri ise önceden ihlal edilmiş (sızdırılmış) olarak kabul eder. Bu durum; bilgiye dayalı kimlik doğrulayıcıları, SMS ve e-posta kurtarma, şifre geçmişi, parola karmaşıklığı ve parola değiştirme kontrollerini işe yaramaz hale getirir. Bu kontroller her zaman faydasız olmuştur hatta çoğu zaman kullanıcıları birkaç ayda bir zayıf şifreler bulmaya zorlar. Ancak 5 milyardan fazla kullanıcı adı ve şifrenin internete sızdırılması ile aksiyon alma zamanı gelmiştir.

Kimlik doğrulama ve oturum yönetimi bölümleri ASVS'de yer alan tüm bölümlerden daha çok değişti. Etkili, kanıta dayalı lider uygulamaların benimsenmesi birçok kişi için zor olacaktır ancak bu durum kabul edilebilirdir. Şimdi şifre sonrası bir geleceğe geçişe başlamalıyız.

NIST 800-63 - Modern, Kanıta Dayalı Kimlik Doğrulama Standardı

NIST 800-63b; modern, kanıta dayalı bir standarttır ve uygulanabilirliğine bakılmaksızın mevcut en iyi tavsiyeyi temsil eder. Standart, tüm dünyadaki tüm kuruluşlar için yararlıdır, ancak özellikle ABD ajansları ve ABD ajanslarıyla ilgilenenler için geçerlidir.

NIST 800-63 terminolojisi ilk başta biraz kafa karıştırıcı olabilir, özellikle de sadece kullanıcı adı ve şifre kimlik doğrulaması kullanımına alışkınsanız. Modern kimlik doğrulamada ilerlemeler gereklidir, bu nedenle gelecekte yaygınlaşacak terminolojiyi tanıtmak zorundayız, ancak bu yeni koşullara alışana kadar endüstrinin yaşayacağı zorluğunu anlıyoruz. Bu bölümün sonunda size yardımcı olması için bir sözlük hazırladık. İhtiyaç şartlarını değil, şartın amacını karşılamak için birçok gereksinimi yeniden ifade ettik. Örneğin, NIST bu standartta "ezberlenmiş sır" kullandığında ASVS "şifre" terimini kullanır.

ASVS V2 Kimlik Doğrulaması, V3 Oturum Yönetimi ve daha az ölçüde, V4 Erişim Kontrolleri, yaygın tehditlere ve yaygın olarak sömürülen kimlik doğrulama zayıflıklarına odaklanan seçilen NIST 800-63b kontrollerinin uyumlu bir alt kümesi olarak uyarlanmıştır. Tam NIST 800-63 uyumluluğunun gerekli olduğu durumlarda, lütfen NIST 800-63'e başvurun.

NIST AAL Seviyesi Seçimi

Uygulama Güvenliği Doğrulama Standardı, ASVS L1'i NIST AAL1 gereksinimlerine, L2'yi AAL2'ye ve L3'ü AAL3'e eşlemeye çalıştı. Bununla birlikte, ASVS Seviye 1'in "temel" (essential) kontrolleri, bir uygulamayı veya API'yi doğrulamak için doğru AAL seviyesi olmayabilir. Örneğin, uygulama bir Seviye 3 uygulamasıysa veya AAL3 için düzenleyici gerekliliklere sahipse, V2 ve V3 Oturum Yönetimi'nde Seviye 3 Bölüm seçilmelidir. NIST uyumlu kimlik doğrulama onay düzeyi (AAL) seçimi, NIST 800-63b Bölüm 6.2'de AAL Seçimi bölümünde belirtilen NIST 800-63b yönergelerine göre gerçekleştirilmelidir.

Semboller

Uygulamalar, özellikle modern kimlik doğrulaması bir uygulamanın yol haritasında olduğunda, ilgili düzeyin gereksinimlerini aşabilir. Daha önce ASVS zorunlu MFA gerektiriyordu ancak NIST zorunlu MFA gerektirmez. Bu

nedenle, ASVS'nin hangi maddeleri tavsiye ettiğini, ancak kontrol gerektirmediğini belirtmek için bu bölümde isteğe bağlı bir adlandırma kullandık. Bu standart boyunca aşağıdaki anahtarlar kullanıldı:

Sembol	Açıklama
	Zorunlu değil
o	Önerilmektedir, fakat zorunlu değildir
✓	Zorunlu

V2.1 Parola Güvenliği Gereksinimleri

NIST 800-63 tarafından "Ezberlenen Sırlar" olarak adlandırılan parolalar arasında; parolalar, pinler, kilit açma düzenleri, doğru kedi yavrusu veya başka bir görüntü öğesi seçimi ve birçok kelime ile oluşturulan parolalar bulunur. Genellikle bir güvenlik terimi olan "bildiğiniz bir şey" olarak kabul edilir ve genellikle tek faktörlü kimlik doğrulayıcılar olarak kullanılırlar. İnternet'te sızdırılan milyarlarca geçerli kullanıcı adı ve parola, varsayılan veya zayıf parolalar, gökkuşağı tabloları ve en yaygın parolaların sıralı sözlükleri de dahil olmak üzere tek faktörlü kimlik doğrulamanın sürekli kullanımına ilişkin önemli zorluklar vardır.

Uygulamalar, kullanıcıları çok faktörlü kimlik doğrulamasına kaydolmaya teşvik etmeli ve kullanıcıların FIDO veya U2F belirteçleri gibi zaten sahip oldukları belirteçleri yeniden kullanmalarına veya çok faktörlü kimlik doğrulama sağlayan bir kimlik bilgisi hizmet sağlayıcısına bağlanmalarına izin vermelidir.

“Credential Service Providers” olarak bilinen kimlik bilgileri hizmet sağlayıcıları (CSP'ler) kullanıcılar için çoklu ortamlarda kullanabilecekleri bir kimlik sağlar. Kullanıcılar genellikle birden çok CSP'ye sahip birden fazla kimliğe sahip olurlar. Örneğin Azure AD, Okta, Ping Identity veya Google ve benzerleri işletme kimlik örnekleridir. Facebook, Twitter, Google veya WeChat gibi kimlikler ise tüketici kimliklerine örnektir. Bu liste, bu şirketlerin veya hizmetlerin onaylanması değil, geliştiricilerin birçok kullanıcının birçok yerleşik kimliğe sahip olduğu gerçeğini dikkate almaları için bir teşviktir. Kuruluşlar, CSP'nin kimlik doğrulama gücünün risk profiline göre mevcut kullanıcı kimlikleriyle bütünleşmeyi düşünmelidir.

Örneğin, bir hükümet kuruluşunun bir sosyal medya kimliğini hassas sistemler için bir giriş olarak kabul etmesi olası değildir, çünkü sosyal medya kullanılarak sahte bir kimlik oluşturmak kolaydır. Oysaki bir mobil oyun şirketinin aktif oyuncu tabanlarını büyütmek için büyük sosyal medya platformlarıyla entegre olması gerekebilir.

#	Açıklama	L1	L2	L3	CWE	NIST §
2.1.1	Kullanıcıların tanımladığı parola değerleri en az 12 karakter uzunluğunda olmalıdır.	✓	✓	✓	521	5.1.1.2
2.1.2	Parola değerleri 64 karakter veya daha uzun olacak şekilde ayarlanabilmelidir.	✓	✓	✓	521	5.1.1.2
2.1.3	Parolaların boşluk veya kısaltma içermediği doğrulanmalıdır. İsteğe bağlı olarak ardışık boşluklar birleştirilebilir	✓	✓	✓	521	5.1.1.2
2.1.4	Parolalarda unicode karakterlere izin verildiği doğrulanmalıdır. Bir tek unicode kod bileşeni bir karakter olarak kabul edilir, bu nedenle 12 emoji veya 64 kanji karakterler geçerli olur ve izin verilmelidir.	✓	✓	✓	521	5.1.1.2
2.1.5	Kullanıcıların parolalarını değiştirebildikleri doğrulanmalıdır.	✓	✓	✓	620	5.1.1.2
2.1.6	Kullanıcı parola değiştirme işlemlerinde hem eski hem de yeni belirleyeceği parolanın bulunduğu doğrulanmalıdır.	✓	✓	✓	620	5.1.1.2
2.1.7	Hesap oluşturma, oturum açma ve şifre değişikliği sırasında gönderilen şifrelerin, yerel olarak veya harici bir API kullanarak bir dizi ifşa edilmiş	✓	✓	✓	521	5.1.1.2

#	Açıklama	L1	L2	L3	CWE	NIST §
	şifreye göre kontrol edildiği doğrulanmalıdır. Bir API kullanılıyorsa, kontrol için gönderilen şifrelerin açık metin gönderilmediğinden emin olunmalıdır. Eğer parola kontrol sonucu ifşa edilen parola listesinde tespit edilirse kullanıcının uygulama üzerinden yeni ve ifşa edilmemiş bir parola seçmesi zorlanmalıdır.					
2.1.8	Kullanıcılar şifre belirlerken anlık olarak parola gücünü gösteren bir yapı olması doğrulanmalıdır.	✓	✓	✓	521	5.1.1.2
2.1.9	Karakter kısıtlaması getirecek bir parola belirleme yapısının olmaması gereklidir. Büyük harf, Küçük harf, sayılar ve özel karakterler için bir parola kompozisyonu uygulanmamalıdır.	✓	✓	✓	521	5.1.1.2
2.1.10	Periyodik kimlik bilgisi/parola değişikliği veya parola geçmişi gereksinimleri olmamalıdır.	✓	✓	✓	263	5.1.1.2
2.1.11	Parola kopyalama özelliği, internet gezgini yardımcıları ve harici parola yönetim araçlarına izin verildiği doğrulanmalıdır.	✓	✓	✓	521	5.1.1.2
2.1.12	Kullanıcının parola girişi yaparken parolanın bir kısmını maskeli gösteren veya son girilen karakteri belli bir süre gösteren yapının olmadığı platformlarda bu özelliğin olduğu (seçilebilir olabilir) doğrulanmalıdır.	✓	✓	✓	521	5.1.1.2

Not: Kullanıcının parolasını görüntülemesine veya son karakteri geçici olarak görmesine izin vermenin amacı, özellikle daha uzun parolaların ve parola yöneticilerinin kullanımında kimlik bilgisi girişinin kullanılabilirliğini geliştirmektir. Gereksinimi dahil etmenin bir başka nedeni, kuruluşların bu modern kullanıcı dostu güvenlik deneyimini kaldırmak için yerel platform şifre alanı davranışını geçersiz kılmasını gerektiren test raporlarını gereksiz yere caydırmak veya önlemektir.

V2.2 Genel Doğrulayıcı (Authenticator) Gereksinimleri

Kimlik doğrulayıcı çevikliği geleceğe yönelik uygulamalar için çok önemlidir. Uygulama kimlik doğrulayıcıları üzerinde yeniden düzenleme yapan uygulama doğrulayıcıları, kullanıcı tercihlerine göre ek kimlik doğrulayıcılara izin vermenin yanı sıra kullanımdan kaldırılmış veya güvenli olmayan kimlik doğrulayıcıların düzenli bir şekilde kullanımdan kaldırılmasına izin verebilmelidir.

NIST, e-posta ve SMS'i "kısıtlı" kimlik doğrulayıcı türleri olarak görmektedir ve gelecekteki bir noktada NIST 800-63'ten ve dolayısıyla ASVS'den kaldırılması muhtemeldir. Uygulamalar, e-posta veya SMS kullanımını gerektirmeyen bir doğrulama yol haritası planlamalıdır.

#	Açıklama	L1	L2	L3	CWE	NIST §
2.2.1	İfşa edilmiş kimlik bilgileri testleri, kaba kuvvet saldırıları ve hesap kilitleme saldırılarına karşı etkili otomasyon önleyici kontrollerinin bulunduğu doğrulanmalıdır. Bu kontroller, en çok ifşa edilen parola bilgilerini bloklamalı, yumuşak kilitleme yapabilmeli, işlem kısıtı getirebilmeli, CAPTCHA eklenebilmeli, saldırı denemeleri arasında artan gecikmeler yapabilmeli, IP adresi kısıtlaması yapabilmeli, veya konum, ilk oturum açılan aygıt, son denemeler gibi risk bazlı kısıtları içermelidir. Tek bir hesap için saat başı 100 adet başarısız deneme sonrası işlemlere izin verilmemelidir.	✓	✓	✓	307	5.2.2 / 5.1.1.2 / 5.1.4.2 / 5.1.5.2
2.2.2	Zayıf kimlik doğrulama mekanizmalarının (SMS ve e-posta gibi) ikincil doğrulama mekanizmaları ve işlem onaylamaları ile kullanımı	✓	✓	✓	304	5.2.10

#	Açıklama	L1	L2	L3	CWE	NIST §
	sağlanmalı ve daha güvenli kimlik doğrulama mekanizmaları yerine kullanılmadığı doğrulanmalıdır. Yeterince güçlü metotların zayıf metotlardan önce kullanımının önerilmesi, kullanıcılarının riskten haberdar edilmesi veya gerekli önlemlerin alınarak riski azaltılması hesap ele geçirilmelerini önleyecektir.					
2.2.3	Oturum açma bilgi detayları değişikliği, e-posta ve adres değişiklikleri, bilinmeyen veya riskli bir konumdan oturum açma bilgileri güvenli bilgilendirme kanallarından kullanıcıya iletilmelidir. SMS veya e-posta yerine push bilgilendirme tercih edilmelidir, ancak bu yapı olmadığı zaman herhangi bir hassas veri ifşasına yol açmadan SMS ve e-posta gönderimi yapılabilir.	✓	✓	✓	620	
2.2.4	Çok faktörlü kimlik doğrulama, kriptografik aygıtların kullanımı veya yüksek seviyeli AAL ve istemci taraflı sertifika kullanımları ile başkasının kimliğine bürünek (impersonation) oltalama (phishing) saldırılarına karşı önlem alındığı doğrulanmalıdır.			✓	308	5.2.5
2.2.5	Kimlik bilgileri servis sağlayıcısı ve uygulama kimlik doğrulamasının ayrıldığından emin olunmalıdır, karşılıklı olarak iki uç nokta arasında TLS bağlantısının olduğu doğrulanmalıdır.			✓	319	5.2.6
2.2.6	Tekrarlama saldırısına direnç olması için OTP aygıtları, kriptografik kimlik doğrulayıcılar veya lookup kodlar kullanılması zorunlu tutulmalıdır.			✓	308	5.2.8
2.2.7	OTP anahtarı (token) girerek veya FIDO donanım anahtarında bir tuşa basarak kullanıcının başlattığı bir işlem ile kimlik doğrulama başlatılması doğrulanmalıdır.			✓	308	5.2.9

V2.3 Doğrulayıcı (Authenticator) Yaşam Döngüsü Gereksinimleri

Doğrulayıcılar; parolalar, yazılımsal jetonlar, donanımsal jetonlar ve biyometrik cihazlardır. Kimlik doğrulayıcıların yaşam döngüsü, bir uygulamanın güvenliği için kritik öneme sahiptir. Örneğin herhangi biri kimlik kanıtı olmayan bir hesabı kendi kendine kaydedebiliyorsa, bu uygulamanın kimlik beyanına çok az güven duyulabilir. Reddit gibi sosyal medya siteleri için bu durum bir risk oluşturmamaktadır ancak bankacılık sistemleri için, kimlik bilgilerinin ve cihazların kaydedilmesine daha fazla odaklanılması uygulamanın güvenliği için kritik öneme sahiptir.

Not: Parolaların bir maksimum ömrü yoktur ve periyodik olarak değiştirilmemelidir. Bunun yerine parolaların sızıp sızmadığı kontrol edilmelidir.

#	Açıklama	L1	L2	L3	CWE	NIST §
2.3.1	Sistem tarafından oluşturulan ilk şifreler veya etkinleştirme kodları güvenli bir şekilde rastgele oluşturulmalıdır, en az 6 karakter uzunluğunda olmalıdır ve harf ve rakam içerebilir şekilde kısa bir süre sonra kullanılamaz hale gelebilmelidir. Otomatik ilk verilen şifreler kalıcı olarak şifre yerine geçmemelidir.	✓	✓	✓	330	5.1.1.2 / A.3
2.3.2	U2F veya FIDO anahtarları gibi kayıt ve abone/kullanıcı kimlik doğrulama aygıtlarının desteklendiği doğrulanmalıdır.		✓	✓	308	6.1.3

#	Açıklama	L1	L2	L3	CWE	NIST §
2.3.3	Zamana bağlı kimlik doğrulayıcıların yenilenme talimatlarının yenilenme için yeterli süre kalacak şekilde zaman verilerek gönderildiği doğrulanmalıdır.		✓	✓	287	6.1.4

V2.4 Kimlik Doğrulama Bilgilerinin Saklanma Gereksinimleri

Mimarlar ve geliştiriciler kod oluştururken veya kodları yeniden düzenlerken bu bölümde belirtilen kurallara uymalıdır. Bu bölümün tam olarak doğrulanabilmesi için kaynak kodu incelemesi veya güvenli birim/entegrasyon testleri yapılmalıdır. Sadece sızma testi gerçekleştirilmesi bu sorunların hiçbirini tanımlayamaz.

Onaylanmış tek yönlü anahtar türetme işlevlerinin listesi NIST 800-63 B bölüm 5.1.1.2 ve BSI Kryptographische Verfahren: Empfehlungen und Schlussellängen (2018) 'de ayrıntılı olarak açıklanmaktadır. Bu seçenekler yerine en son ulusal/bölgesel algoritma ve anahtar uzunluk standartları kullanılabilir.

Bu bölümün sızma testi ile tamamlanması mümkün olmadığı için kontroller L1 olarak işaretlenmemiştir. Bu bölüm kimlik bilgilerinin güvenliği için hayati önem taşımaktadır. Bu nedenle ASVS'yi bir mimari ve kodlama kılavuzu veya kaynak kodu inceleme kontrol listesi için talep ediyorsanız bu kontrollerin özel sürümünüzde L1'e yerleştirilmesi gerekmektedir.

#	Açıklama	L1	L2	L3	CWE	NIST §
2.4.1	Şifrelerin çevrim dışı ataklara karşı dayanıklı bir şekilde tutulduğu doğrulanmalıdır. Şifreler tuzlanmalı ve güvenilir olduğu bilenen bir metot ile tek yönlü özetleri (hash) alınarak saklanmalıdır.		✓	✓	916	5.1.1.2
2.4.2	Kullanılan tuzlama değişkeninin minimum 32 bit uzunluğunda olması ve rastgele aynı tuzlama değerini vermemesi beklenmektedir. Her bir şifre için tekil ve özel bir tuzlama değeri ve sonuçta oluşan özet kaydedilmelidir.		✓	✓	916	5.1.1.2
2.4.3	PBKDF2 kullanımı doğrulanmalı, iterasyon sayısı genel olarak doğrulama sunucusunun performansına da bağlı olarak en az 100.000 iterasyon olacak şekilde verilmelidir.		✓	✓	916	5.1.1.2
2.4.4	Bcrypt kullanılıyorsa, iş faktörü, doğrulama sunucusunun performansına izin verecek kadar büyük olmalıdır (genellikle en az 13).		✓	✓	916	5.1.1.2
2.4.5	Anahtar türetme için ekstra bir iterasyon yapılmalı ve tuzlama değeri gizli tutularak sadece doğrulama yapacak taraf tarafından bilinmelidir. Tuzlama değeri güvenilirliği kanıtlanmış bir rastgele bit üretici tarafından yapılmalı ve SP 800-131A dokümanında bulunan minimum Güvenlik gereksinimlerine uygun olmalıdır. Gizli tuzlama değeri özeti alınmış şifre değerlerinden ayrılmış olarak özel bir aygıt veya HSM'de tutulmalıdır.		✓	✓	916	5.1.1.2

ABD standartlarından bahsedildiği noktalarda, ABD standartlarının yerine veya bunlara ek olarak bölgesel standartlar kullanılabilir.

V2.5 Kimlik Doğrulama Bilgilerinin Kurtarma Gereksinimleri

#	Açıklama	L1	L2	L3	CWE	NIST §
2.5.1	Sistem tarafından üretilen bir aktivasyon veya kurtarma sırrının açık metin olarak kullanıcıya gönderilmediği doğrulanmalıdır.	✓	✓	✓	640	5.1.1.2

#	Açıklama	L1	L2	L3	CWE	NIST §
2.5.2	Parola ipuçları veya bilgiye dayı kimlik doğrulamanın var olmadığı / kullanılmadığı doğrulanmalıdır.	✓	✓	✓	640	5.1.1.2
2.5.3	Kimlik doğrulama bilgileri kurtarma işlemi sırasında mevcuttaki parola bilgisinin ifşa edilmediği doğrulanmalıdır.	✓	✓	✓	640	5.1.1.2
2.5.4	Paylaşılan veya önceden belirlenmiş (default) hesapların kullanılmadığı doğrulanmalıdır. ((Örneğin "root", "admin", veya "sa")	✓	✓	✓	16	5.1.1.2 / A.3
2.5.5	Herhangi bir kimlik doğrulama faktörü değişirse bu olaydan kullanıcının haberdar edilmesi gerekmektedir.	✓	✓	✓	304	6.1.2.3
2.5.6	Unutulan şifreler ve diğer kurtarma işlemlerinin güvenli kurtarma mekanizmalarını kullandığı doğrulanmalıdır. (Örneğin TOTP veya diğer soft anahtarlar, vb.)	✓	✓	✓	640	5.1.1.2
2.5.7	OTP veya çok faktörlü kimlik doğrulama faktörleri kaybolursa kimlik ispatlama kanıtı, kayıt anında aynı seviyede uygulanmalıdır.		✓	✓	308	6.1.2.3

V2.6 Ön Tanımlı Sırların (Look-up Secret) Doğrulanma Gereksinimleri

“Look-up Secret” değerleri; Transaction Authorization Numbers (TAN), sosyal medya kurtarma kodları veya bir dizi rastgele değer içeren önceden oluşturuluş gizli kod listeleridir. Bu kodlar kullanıcılara güvenli bir yoldan dağıtılır. Bu kodlar bir kez kullanılır ve kullanıldığında “Loop-up Secret” listesinden kaldırılır. Bu tarz bir doğrulayıcı “sahip olduğunuz bir şey” ile kimlik doğrulama yöntemi olarak kabul edilir.

#	Açıklama	L1	L2	L3	CWE	NIST §
2.6.1	Ön tanımlı listede gelen sırların yalnızca bir kez kullanılıp daha sonar kullanılamadığı doğrulanmalıdır.		✓	✓	308	5.1.2.2
2.6.2	Ön tanımlı sırların yeterli rastgele entropisiyle (112 bit entropi) üretildiği, bu değerin altında ise tekil özel bir tuzlama ile 32 bit tuzlanması ve tek yönlü geri çevrilemez bir özetleme ile özetinin alındığı doğrulanmalıdır.		✓	✓	330	5.1.2.2
2.6.3	Ön tanımlı sırların tahmin edilebilir değerleri gibi çevrim dışı saldırılara karşı dayanıklı olması doğrulanmalıdır.		✓	✓	310	5.1.2.2

V2.7 Alternatif Kanal (Out of Band) Doğrulama Gereksinimleri

Geçmişte, bant dışı doğrulayıcıları çoğu kullanıcılara şifre sıfırlama bağlantısı içeren bir e-posta veya SMS gönderirlerdi. Saldırganlar bu zayıf mekanizmayı istismar etmek için bir kişinin e-posta hesabını geçirirlerdi ve keşfedilen sıfırlama bağlantılarını yeniden kullanarak hedef aldıkları kullanıcılara ait hesap parolalarını sıfırlarlardı. Bu nedenle bant dışı doğrulamaları yapmanın yeni yolları geliştirilmiştir.

Güvenli bant dışı kimlik doğrulayıcılar, doğrulayıcı ile güvenli bir ikincil kanal üzerinden iletişim kurabilen fiziksel cihazlardır. Mobil cihaz anında iletim bildirimleri (push notification) güvenli bant dışı kimlik doğrulama için güzel bir örnektir. Bu tür bir doğrulayıcı “sahip olduğunuz bir şey” doğrulama yöntemi olarak değerlendirilebilir. Bir kullanıcı kimlik doğrulaması yapmak istediğinde, doğrulama yapmak isteyen uygulama, üçüncü taraf bir hizmet sağlayıcı aracılığıyla veya doğrudan bağlantı yoluyla bant dışı kimlik doğrulayıcıya bir mesaj gönderir. Bu mesaj bir kimlik doğrulama kodu (genellikle rastgele 6 haneli sayı veya bir onaylama diyalogu) içerir. Doğrulama uygulaması, kimlik doğrulama kodunu birinci kanaldan almayı bekler ve alınan değerin özetini (hash) orijinal doğrulama kodunun özeti ile karşılaştırır. Eğer bu iki değer eşleşirse, bant dışı doğrulayıcı kullanıcı kimliğinin doğrulandığını varsayabilecektir.

ASVS yalnızca birkaç geliştiricinin push bildirimleri gibi yeni bant dışı kimlik doğrulayıcıları geliştireceğini varsayar. Bu nedenle kimlik doğrulama API'si, uygulamalar ve tek oturum açma uygulamaları (single sign-on) gibi doğrulayıcılar için aşağıdaki ASVS denetimleri geçerlidir. Eğer yeni bir bant dışı kimlik doğrulayıcısı geliştiriyorsanız NIST 800-63B § 5.1.3.1'e bakın.

E-posta ve VOIP gibi güvenli olmayan bant dışı kimlik doğrulayıcıların kullanımına izin verilmez. OSTN ve SMS gibi kimlik doğrulama yöntemleri şu anda NIST tarafından kısıtlanmıştır ve bu doğrulamaların yerine anında iletim bildirimleri (push notification) benzeri yapılar kullanılmalıdır. Bant kimlik doğrulamasında telefon veya SMS kullanmanız gerekiyorsa, lütfen § 5.1.3.3'e bakın.

#	Açıklama	L1	L2	L3	CWE	NIST §
2.7.1	Push bildirimleri gibi güvenli kimlik doğrulayıcılar ilk kullanımda öncelikli önerilmeli, SMS veya PSTN gibi açık metin ilk kullanım doğrulayıcıları önerilmemelidir.	✓	✓	✓	287	5.1.3.2
2.7.2	İlk kullanım isteği yapıldıktan 10 dakika sonra ilk kullanım anahtarları veya kodları kullanımı süresi dolmalıdır.	✓	✓	✓	287	5.1.3.2
2.7.3	İlk kullanım kimlik doğrulama istekleri, kodları veya anahtarlarının sadece bir sefere mahsus olarak kullanılması ve yalnızca orijinal oturum açma isteği için olduğu doğrulanmalıdır.	✓	✓	✓	287	5.1.3.2
2.7.4	İlk kullanım oturum açma ve kimlik doğrulama mekanizmalarının bağımsız ve güvenli bir kanal üzerinden iletişime geçtiği doğrulanmalıdır.	✓	✓	✓	523	5.1.3.2
2.7.5	İlk kullanım doğrulayıcısının yalnızca kimlik doğrulama kodunun özet bilgisine sahip olduğu doğrulanmalıdır.		✓	✓	256	5.1.3.2
2.7.6	Başlangıç kimlik doğrulama kodunun güvenli rastgele bir sayı üreticisinden en az 20 bit entropi ile üretildiği doğrulanmalıdır.		✓	✓	310	5.1.3.2

V2.8 Tekli ya da Çoklu Kimlik Doğrulama Gereksinimleri

Tek faktörlü bir defalık şifreler (OTP – One Time Password) her üretimde rastgele farklı bir değer üretip gösteren fiziksel veya yazılımsal jetonlardır. Bu cihazlar "phishing" (kimliğe bürünme) saldırılarını imkânsız hale getirmeseler de zorlaştırırlar. Bu tür bir doğrulayıcı, kimlik doğrulama yöntemlerinden "sahip olduğunuz bir şey" isimli doğrulama yöntemini gerçekleştirmektedir ve kimlik doğrulama yapmak için kişinin sahip olduğu bir şeye gereksinim duyar.

Çok faktörlü jetonlar ise (multi-factor OTP) tek faktörlü jetonlardan farklı olarak, OTP'yi üretmek için; geçerli bir PIN kodu, biyometrik kilit açma, USB cihaz yerleştirme, NFC cihaz eşleştirmesi veya işlem imzalama hesaplayıcıları gibi başka ek değerlerin girilmesini gerektirirler.

#	Açıklama	L1	L2	L3	CWE	NIST §
2.8.1	Zaman tabanlı OTP'lerin tanımlı bir yaşam süresi ile kullanımları sonlandırılmalıdır.	✓	✓	✓	613	5.1.4.2 / 5.1.5.2
2.8.2	HSM veya güvenli işletim sistemi anahtar depolama özelliği kullanılarak gönderilen OTP'lerin doğrulanmasında görev alan simetrik anahtarların yüksek derecede korunması doğrulanmalıdır.		✓	✓	320	5.1.4.2 / 5.1.5.2
2.8.3	Oluşturma, yerleştirme ve doğrulama işlemleri için güvenilir kripto algoritmaların kullanıldığı doğrulanmalıdır.		✓	✓	326	5.1.4.2 / 5.1.5.2

2.8.4	Geçerlilik süresi boyunca OTP'nin yalnızca bir kere kullanılabilir olduğu doğrulanmalıdır.	✓	✓	287	5.1.4.2 / 5.1.5.2
2.8.5	Zaman tabanlı çok faktörlü OTP geçerlilik süresi içinde tekrar kullanılırsa bu durum kayıt altına alınarak güvenli bilgilendirme yoluyla aygıt sahibine bilgilendirme yapılmalıdır.	✓	✓	287	5.1.5.2
2.8.6	Fiziksel tek faktör OTP oluşturucusu hırsızlık veya diğer kayıplarda iptal edilmelidir. İptal etme işleminin anında ve etkili bir şekilde olduğu ve konuma bakılmaksızın tüm açık oturumlarda yapılması sağlanmalıdır.	✓	✓	613	5.2.1
2.8.7	Biyometrik kimlik doğrulayıcıların yalnızca sahip olunan bir şeyle veya bilinen bir şeyle bağlantılı ikincil faktörler şekilde kullanımla sınırlı olduğu doğrulanmalıdır.	o	✓	308	5.2.3

V2.9 Kriptografik Yazılım ve Donanım Doğrulama Gereksinimleri

Kriptografik güvenlik anahtarları, kullanıcıların kimlik doğrulamayı tamamlamak için kriptografik aygıtı bilgisayara takması veya eşleştirmesi gereken akıllı kartlar veya FIDO anahtarlarıdır. Doğrulayıcı yazılımlar bu kriptografik cihazlara veya yazılıma bir kimlik sorma değeri (challenge) gönderir ve kriptografik cihaz veya yazılım güvenli bir şekilde sakladığı kriptografik anahtar ile bu gönderilen değere ait bir yanıt hesaplar (response).

Tek faktörlü kriptografik cihazlar ve yazılım, çok faktörlü kriptografik cihazlar ve yazılımlar için gereklilikler aynıdır çünkü kriptografik kimlik doğrulayıcının doğrulanması kimlik doğrulama faktörüne sahip olduğunu kanıtlar.

#	Açıklama	L1	L2	L3	CWE	NIST §
2.9.1	Doğrulamada kullanılan kriptografik anahtarların ifşa edilmesini engellemek adına TPM, HSM veya işletim sisteminde güvenli depolamaya izin veren servisler aracılığıyla güvenli bir şekilde depolanması ve korunması gerekmektedir.		✓	✓	320	5.1.7.2
2.9.2	Meydan okuma nota (challenge nonce) değerinin en az 64 bit olması ve istatistiksel olarak tekil veya kriptografik cihazın yaşam süresince tekil olması doğrulanmalıdır.		✓	✓	330	5.1.7.2
2.9.3	Oluşturma, yerleştirme ve doğrulama işlemleri için güvenilir kriptografik algoritmaların kullanıldığı doğrulanmalıdır.		✓	✓	327	5.1.7.2

V2.10 Servis Kimlik Doğrulama Gereksinimleri

Bu bölümde yer alan maddeler sızma testleri ile tespit edilemeyecek maddelerdir ve bu nedenle herhangi bir L1 gereksinimi bulunmamaktadır. Ancak bu maddeler bir mimaride, kodlamada veya kod güvenliği incelemesinde kullanılıyor ise, yazılımın (Java Key Store gibi) L1 için minimum gereksinim olduğu varsayılmalıdır. Gizli bilgilerin açık metin olarak saklanmasının hiçbir koşulda kabul edilmediği unutulmamalıdır.

#	Açıklama	L1	L2	L3	CWE	NIST §
2.10.1	Bütünlüğü sağlayan sırların, API anahtarları veya ortak yetkili hesaplarınki gibi değişmeyen parolalardan olmadığı doğrulanmalıdır.		OS assisted	HSM	287	5.1.1.1
2.10.2	Parola gerekli ise ön tanımlı parolaların kullanılmadığı doğrulanmalıdır.		OS assisted	HSM	255	5.1.1.1

2.10.3	Parolaların, yerel sistem erişimi ve çevrimdışı Kurtarma saldırıları gibi durumlara karşı yeterince korunarak saklandığı doğrulanmalıdır.	OS assisted	HSM	522	5.1.1.1
2.10.4	Parolaların, veri tabanları ve üçüncü taraflar ile entegrasyonların, iç sırların ve API anahtarlarının güvenli bir şekilde ve kaynak kodda veya kod depolarında tutulmaması gerekir. Bu depolama çevrimdışı saldırılara karşı korunmalıdır. Güvenli yazılım anahtarı saklama aygıtları parola koruması için önerilmektedir. (TPM,HSM gibi)	OS assisted	HSM	798	

Ek ABD Kurumu Gereksinimleri

ABD Ajanslarının NIST 800-63 ile ilgili zorunlu gereksinimleri vardır. Uygulama Güvenliği Doğrulama Standardı, uygulamaların yaklaşık % 100'ü için geçerli olan kontrollerin yaklaşık %80'i olmuştur ve gelişmiş kontrollerin son % 20'si veya sınırlı uygulanabilirliği olanların %80'i olmuştur. Bu nedenle, ASVS, özellikle IAL1/2 ve AAL1/2 sınıflamaları için NIST 800-63'ün katı bir alt kümesidir ancak özellikle IAL3/AAL3 sınıflamaları ile ilgili olarak yeterince kapsamlı değildir.

ABD hükümet kurumlarını NIST 800-63'ü bütünüyle gözden geçirmesi ve uygulaması gerekmektedir.

Terimler Sözlüğü

Terim	Anlam
CSP	Kimlik Bilgisi Hizmeti Sağlayıcısı, Kimlik Sağlayıcısı olarak da bilinir.
Authenticator (Kimlik Denetleyicisi)	Parola, Jeton, MFA, birleşik onaylama (federated assertion) vb. kimlik doğrulaması yapan kod.
Verifier (Doğrulayıcı)	“Bir kimlik doğrulama protokolü kullanarak kimlik doğrulaması yapmak isteyenin sahipliğini kontrol eder. Buna ek olarak bir veya iki kimlik doğrulayıcısını doğrulama protokolü kullanarak kontrol eder. Bunu yapmak için, doğrulayıcının kimlik doğrulayıcıyı abone tanımlayıcısına bağlayan kimlik bilgilerini doğrulaması ve durumunu kontrol etmesi gerekebilir.”
OTP	Tek seferlik şifre
SFA	Bildiğiniz bir şey (ezberlenmiş sırlar, şifreler, parolalar, pinler), olduğunuz bir şey (biyometri, parmak izi, yüz taramaları) veya sahip olduğunuz bir şey (OTP belirteçleri, akıllı kart gibi bir şifreleme cihazı) gibi tek faktörlü kimlik doğrulayıcılar
MFA	Çok Faktörlü Kimlik Doğrulayıcı (Multi factor authenticator), iki veya daha fazla faktör içeren kimlik doğrulayıcı

Referanslar

Detaylı bilgi için aşağıdaki bağlantıları ziyaret edebilirsiniz:

- NIST 800-63 - Digital Identity Guidelines
- NIST 800-63 A - Enrollment and Identity Proofing
- NIST 800-63 B - Authentication and Lifecycle Management
- NIST 800-63 C - Federation and Assertions

- NIST 800-63 FAQ
- OWASP Testing Guide 4.0: Testing for Authentication
- OWASP Cheat Sheet - Password storage
- OWASP Cheat Sheet - Forgot password
- OWASP Cheat Sheet - Choosing and using security questions

V3: Oturum Yönetimi Doğrulama Gereksinimleri

Kontrol Amacı

Web tabanlı bir uygulamanın temel bileşenlerinden biri, uygulamayla etkileşim kuran bir kullanıcının durumunu kontrol ettiği ve koruduğu mekanizmadır. Buna "Oturum Yönetimi (Session Management)" adı verilmektedir. Bir uygulama ile bir kullanıcının arasındaki tüm etkileşim durumlarını yöneten denetimler kümesi olarak tanımlanabilmektedir.

Doğrulanmış bir uygulamanın aşağıdaki üst düzey oturum yönetimi gereksinimlerini karşıladığından emin olun:

- Oturumlar, her bir kullanıcı için benzersizdir. Paylaşılamaz ve tahmin edilemezdir.
- Oturumlar, artık gerekmedikçe ve uzun süre işlem yapılmadığında zaman aşımına uğradığından geçersiz kılınır.

Daha önce belirtildiği gibi; bu gereksinimler, yaygın tehditlere ve yaygın olarak sömürülen kimlik doğrulama zayıflıklarına odaklanmış ve seçilen NIST 800-63b kontrollerinin uyumlu bir alt kümesi olarak uyarlanmıştır. Önceki doğrulama gerekliliklerinde bire bir aynı olanlar kaldırılmıştır ve çoğu madde zorunlu NIST 800-63b gereklilikleri ile güçlü bir şekilde uyumlu olacak şekilde uyarlanmıştır.

Güvenlik Doğrulama Gereksinimleri

V3.1 Temel Oturum Yönetimi Gereksinimleri

#	Açıklama	L1	L2	L3	CWE	NIST §
3.1.1	Uygulamanın asla oturum anahtarları URL parametrelerinde veya hata mesajlarında göstermediği doğrulanmalıdır.	✓	✓	✓	598	

V3.2 Oturum Bağlama (Session Binding) Gereksinimleri

#	Açıklama	L1	L2	L3	CWE	NIST §
3.2.1	Kullanıcı oturumu anahtarının Kullanıcı bilgilerinin her girdiğinde yenilenmesi doğrulanmalıdır.	✓	✓	✓	384	7.1
3.2.2	Oturum anahtarlarının en az 64 entropi ile üretildiği doğrulanmalıdır.	✓	✓	✓	331	7.1
3.2.3	Uygulamanın sadece HTML 5 depolama ve uygun güvenli çerezler vasıtasıyla tarayıcıda oturum anahtarı tutabileceği doğrulanmalıdır.	✓	✓	✓	539	7.1
3.2.4	Oturum anahtarlarının güvenilir kriptografik algoritmalar ile üretildiği doğrulanmalıdır.		✓	✓	331	7.1

TLS veya başka bir güvenli ulaşım kanalı oturum yönetimi için zorunludur. Bu, başlık İletişim Güvenliği Doğrulama Gereksinimleri bölümünde ele alınmıştır.

V3.3 Oturum Sonlanma ve Zaman Aşımı Gereksinimleri

Oturum zaman aşımları, güvenlik standartlarının geleneksel olarak izin verdiğinden çok daha uzun oturum zaman aşımlarına izin veren NIST 800-63 ile uyumludur. Kuruluşlar aşağıdaki tabloyu gözden geçirmeli ve eğer uygulamanın riskine göre daha uzun bir oturum süresi isteniyorsa, aktivitesiz oturum zaman aşım değeri üst sınırı NIST tablosunda yer alan süre olacak şekilde ayarlanmalıdır.

Bu bağlamda L1 - IAL1/AAL1, L2 - IAL2/AAL3, L3 - IAL3/AAL3'tür. IAL2/AAL2 ve IAL3/AAL3 için, daha kısa boşta kalma zaman aşımı süresi belirlemek, oturumu sürdürmek, oturumu kapatmak veya oturumu yeniden doğrulamak üzere boşta kalma sürelerinin alt sınırıdır.

#	Açıklama	L1	L2	L3	CWE	NIST §
3.3.1	Oturum sonlandırma ve zaman aşımının oturum anahtarını geçersiz kılması gerekmekte olup, güvenilen taraflar arasında bile geçersiz kılma işlemi sonrası geri tuşuyla veya diğer yollarla oturum açılmış gibi işlem yapılamadığı doğrulanmalıdır.	✓	✓	✓	613	7.1
3.3.2	Eğer kimlik doğrulayıcılar kullanıcılara oturumlarını açık bırakma hakkı tanıyorsa, belirli bir periyodik süre sonrası aktif veya pasif durum olmasına bakılmaksızın yeniden oturum açılması sağlanmalıdır.	30 gün	12 saat ya da 30 dakika aktivitesizlik, 2FA opsiyonel	12 saat ya da 15 dakika aktivitesizlik, 2FA opsiyonel	613	7.2
3.3.3	Uygulama başarılı bir parola değişikliği sonrası bütün açık oturumlardan Uygulama genelinde ve buna Bağlı üçüncü taraflarda oturum sonlandırmaya gitmelidir.		✓	✓	613	
3.3.4	Kullanıcılar tüm mevcut aktif oturumlarını ve oturum açan cihazları görebiliyor ve oturum sonlandırabiliyor olmalıdır.		✓	✓	613	7.1

V3.4 Çerez Tabanlı Oturum Yönetimi

#	Açıklama	L1	L2	L3	CWE	NIST §
3.4.1	Çerez tabanlı oturum anahtarlarının 'Secure' özelliğine sahip olduğundan emin olunmalıdır.	✓	✓	✓	614	7.1.1
3.4.2	Çerez tabanlı oturum anahtarlarının 'HttpOnly' özelliğine sahip olduğundan emin olunmalıdır.	✓	✓	✓	1004	7.1.1
3.4.3	Çerez tabanlı oturum anahtarlarının 'SameSite' özelliğini kullanarak Siteler Arası İstek Sahteciliği'ne (CSRF) karşı maruziyeti azalttığından emin olunmalıdır.	✓	✓	✓	16	7.1.1
3.4.4	Çerez tabanlı oturum anahtarlarının "__Host-" ön ekini kullanarak çerez gizliliği sağladığı doğrulanmalıdır.	✓	✓	✓	16	7.1.1
3.4.5	Eğer uygulama diğer uygulamalar ile aynı alan adı altında yayınlanıyorsa ve oturum çerezlerinin birbirlerince atanması ve dolayısıyla açık olması halinde çerezlerde en doğru metot seçilerek 'path' özelliğinin kullanıldığı doğrulanmalıdır.	✓	✓	✓	16	7.1.1

V3.5 Jeton (Token) Bazlı Oturum Yönetimi

Jeton tabanlı oturum yönetimi JWT, OAuth, SAML ve API anahtarları kullanımını içerir. Bu yöntemler içerisinden API anahtarı kullanımının zayıf olduğu bilinmektedir ve bu altyapı yeni geliştirilen kodlarda kullanılmamalıdır.

#	Açıklama	L1	L2	L3	CWE	NIST §
3.5.1	Uygulamaların OAuth dışına çıkarak anahtarlarını kendilerinin yenilemesine izin verilmemeli ve kullanıcılara, bağlantılı uygulamalarla güven ilişkisini sonlandırma yetkisiz verilmelidir.		✓	✓	290	7.1.2
3.5.2	Uygulamanın statik API sırrı / anahtarı kullanması yerine oturum anahtarlarının kullanımı doğrulanmalıdır. (Eski Uygulamalar hariç olabilir).		✓	✓	798	
3.5.3	Kötü amaçlı değişiklik, tekrarlama, boş cipher ve anahtar değiştirme saldırılarına karşı önlem alınarak oturum çerezinin bulunmadığı oturumlarda dijial imzalama, şifreleme ve diğer gerekli önlemlerin alınması gerekmektedir.		✓	✓	345	

V3.6 Federasyon veya Onaylama ile Yeniden Kimlik Doğrulama

Bu bölüm “relying party (RP)” veya “credential service provider (CSP)” yapılarını geliştirenler içindir. Bu geliştirmeler için aşağıdaki maddelerin doğru şekilde ele alındığına emin olunmalıdır.

#	Açıklama	L1	L2	L3	CWE	NIST §
3.6.1	Kimlik doğrulama servis sağlayıcılarına maksimum bir oturum açma süresi tanımlanarak bu süre zarfında daha önce bir istekte bulunmamış bir başvuru sahibinin sisteme yeniden oturum açma isteği bulunursa kimlik doğrulama servis sağlayıcısı tarafından bu isteğin karşılanması beklenmektedir.			✓	613	7.2.1
3.6.2	Kimlik doğrulama servis sağlayıcısı kendisine güvenen tarafa son oturum açma olayını bildirerek servise gelen tarafın yeniden oturum açma ihtiyacı olup olmadığını anlamasına olanak sağlar.			✓	613	7.2.1

V3.7 Oturum Yönetimini Hedef Alan Saldırılara Karşı Savunma

Az sayıda bilinen oturum yönetimi saldırısı bulunmaktadır ve bu saldırıların bir kısmı oturumların kullanıcı deneyimi (UX) ile ilgilidir. Geçmiş ISO 27002 gereksinimine dayanarak, ASVS içerisinde eşzamanlı çoklu oturumların engellenmesi gerekliliği yer alıyordu. Ancak günümüzde eşzamanlı çoklu oturumun kullanımının engellemesi uygun değildir; sadece modern kullanıcıların birden fazla cihaza sahip olması veya uygulamanın tarayıcı oturumu gerektirmeyen bir API olmasından dolayı değil, genellikle bu tarz uygulamalarda son kimlik doğrulamadan geçen kullanıcının uygulamaya erişim sağlaması ve genelde bu kullanıcının bir saldırgan olmasıdır.

Bu bölüm, uygulama kodları ile oturum yönetimi saldırılarını caydırmak, geciktirmek ve tespit için bir rehber sağlar.

“Half-open Attack” Açıklaması

2018 yılının başlarında, bazı finansal kurumlar saldırganların “half-open attacks” ismini verdiği saldırılardan zarar gördü ve bu saldırı terimi endüstriye yerleşti. Saldırganlar, farklı kod tabanlarına sahip birden fazla kuruma saldırsa da aslında hedef alınan kodlar aynı kurumlar içinde farklı kod tabanları gibi görünüyor. “half-open attacks” ismi ile bilinen saldırı birçok mevcut kimlik doğrulama, oturum yönetimi ve erişim kontrol sisteminde yaygın olarak bulunan bir tasarım modelinden yararlanıyor.

Bu saldırı modelini uygulamak için saldırganlar; kimlik bilgilerini kilitlemeye, sıfırlamaya veya kurtarmaya çalışarak yarı açık bir saldırı başlatır. Oturum yönetimi modellerinde sıklıkla; kimliği doğrulanmamış, yarı kimliği doğrulanmış (parola sıfırlamaları, kullanıcı adını unuttum ekranları vb.) ve tam olarak doğrulanmış kullanıcı durumları arasında oturum nesneleri (session objects) yeniden kullanılır.

Bu tasarım deseni, parola özetleri (hash) ve rolleri dahil olmak üzere mağdurun profilini içeren geçerli bir oturum nesnesini veya jetonunu doldurur. Denetleyicilerdeki (controllers) veya yönlendiricilerdeki (routers) erişim denetimi kontrolleri kullanıcının tam olarak oturum açtığını doğru şekilde doğrulamazsa, saldırgan hedef aldığı kullanıcı olarak hareket edebilir. Saldırılar, kullanıcının şifresini bilinen bir değere değiştirmeyi, geçerli bir şifre sıfırlaması gerçekleştirmek için e-posta adresini güncellemeyi, çok faktörlü kimlik doğrulamayı devre dışı bırakmayı, yeni bir MFA cihazını kaydetmeyi veya API anahtarlarını ortaya çıkarmayı/değiştirmeyi vb. içerebilir.

#	Açıklama	L1	L2	L3	CWE	NIST §
3.7.1	Uygulamanın herhangi bir hassas işlem veya hesap değişikliği öncesinde geçerli oturumun açık olduğundan emin olması veya yeniden oturum açmaya yönlendirmesi veya ikincil bir doğrulama yapması gerekmektedir.	✓	✓	✓	778	

Referanslar

Detaylı bilgi için aşağıdaki bağlantıları ziyaret edebilirsiniz:

- OWASP Testing Guide 4.0: Session Management Testing
- OWASP Session Management Cheat Sheet
- Set-Cookie Host- prefix details

V4: Erişim Kontrolleri Doğrulama Gereksinimleri

Kontrol Amacı

Yetkilendirme, kaynakların kullanımına yalnızca izin verilen kişilerin erişimine izin verme konseptidir. Doğrulanmış bir uygulamanın aşağıdaki yüksek seviye gereksinimleri karşıladığından emin olun:

- Kaynaklara erişen kişiler için geçerli kimlik bilgilerini bulundurur.
- Kullanıcılar iyi tanımlanmış roller ve yetkiler ile ilişkilendirilmiştir.
- Rol ve izin verisi, tekrar düzenlenmeye ve kurcalanmaya karşı korunmuştur.

Güvenlik Doğrulama Gereksinimleri

V4.1 Genel Erişim Kontrol Tasarımı

#	Açıklama	L1	L2	L3	CWE
4.1.1	Uygulamanın hem genel hem de özellikle istemci tarafında erişim kontrolü mevcut ve aşılabilir olduğu durumlarda erişim kontrolünü güvenilir bir servis katmanında yapmaya zorladığı doğrulanmalıdır.	✓	✓	✓	602
4.1.2	Özel olarak yetkilendirilmediği sürece, erişim kontrolleri tarafından kullanılan tüm kullanıcı verileri ve politika bilgilerinin son kullanıcı tarafından değiştirilemediği doğrulanmalıdır.	✓	✓	✓	639
4.1.3	En az yetki ilkesinin uygulandığı doğrulanmalıdır. Kullanıcılar sadece erişim yetkileri olan fonksiyonlara, dosyalara, URL'lere, denetleyicilere (controller), servislere veya diğer kaynaklara erişebilir olmalıdır. Bu sayede yetki yükseltme ve "spoofing" saldırılarının önüne geçilecektir	✓	✓	✓	285
4.1.4	Yeni Kullanıcı / rol tanımlarında minimum veya hiç yetki tanımlamama ilkesi geçerli olmalı ve kullanıcılar / roller yeni bir özelliğe ayrıca bir atama yoksa doğrudan erişememelidir.	✓	✓	✓	276
4.1.5	Erişim kontrolleri hatalarının güvenli bir şekilde ele alındığı doğrulanmalıdır.	✓	✓	✓	285

V4.2 İşlem Seviyesinde Erişim Kontrolü

#	Açıklama	L1	L2	L3	CWE
4.2.1	Hassas veri ve API'lerin doğrudan nesne erişim saldırılarına karşı korunup farklı kullanıcıların kayıtlarına erişim, değişiklik, silme vb. durumların olmaması sağlanmalıdır.	✓	✓	✓	639
4.2.2	Uygulamanın ve uygulama çerçevesinin (framework), yeterince karmaşıklığa sahip ve tahmin edilemez bir anti-CSRF (siteler arası istek sahteciliği) anahtarı ürettiğinden veya benzeri koruma mekanizmasına sahip olduğundan emin olunmalıdır.	✓	✓	✓	352

V4.3 Dikkat Edilmesi Gereken Diğer Erişim Kontrol Maddeleri

#	Açıklama	L1	L2	L3	CWE
4.3.1	Yönetim amaçlı ara yüzlerin yetkisiz erişimlere karşı çok faktörlü kimlik doğrulaması ile korunması gerekmektedir.	✓	✓	✓	419

#	Açıklama	L1	L2	L3	CWE
4.3.2	Özellikle istenmediği sürece dizin geziniminin (directory browsing) kapalı olduğu doğrulanmalıdır. Ayrıca uygulamalar, bir dosyanın ifşasına veya dizin gezinimine izin vermemelidir. Örneğin; Thumbs.db, .DS_Store, .git veya .svn gibi dizinler.	✓	✓	✓	548
4.3.3	Uygulamanın düşük değerli sistemler için ek yetkilendirmeye (yükseltme veya uyumlu bir kimlik doğrulama) sahip olduğu doğrulanmalıdır. Yüksek değere sahip uygulamalarda, dolandırıcılık faaliyetlerini önlemek ve daha önceki vakalara karşı önlem almak için görev ayrımı yapılmalıdır		✓	✓	732

Referanslar

Detaylı bilgi için aşağıdaki bağlantıları ziyaret edebilirsiniz:

- OWASP Testing Guide 4.0: Authorization
- OWASP Cheat Sheet: Access Control
- OWASP CSRF Cheat Sheet
- OWASP REST Cheat Sheet

V5: Kötü Amaçlı Girdi Verilerinin Doğrulama Gereksinimleri

Kontrol Amacı

En yaygın web uygulaması zafiyetleri, istemciden veya ortamdan gelen girdiyi kullanmadan önce uygun bir şekilde doğrulamamaktan ortaya çıkmaktadır. Bu doğrulamanın yapılmaması, XSS, SQL enjeksiyonu, dosya sistemi saldırıları, yorumlayıcı enjeksiyonları, Unicode saldırıları ve bellek taşmaları gibi en bilinen zafiyetlerin ortaya çıkmasına neden olmaktadır.

Doğrulanmış bir uygulamanın aşağıdaki yüksek seviye gereksinimleri karşıladığından emin olun:

- Girdi doğrulama ve çıktı kodlama altyapıları birbiri ile ilişkili tasarlanarak enjeksiyon saldırıları engellenmelidir.
- Kullanıcı girdilerine ait veri türleri ve bu veri türlerinin beklenen formatta olduğu doğrulanmalı, alabileceği uzunluk/aralık değerlerine uyup uymadığı kontrol edilmeli veya en kötü ihtimalle bu kullanıcı verileri filtrelenmelidir.
- Veri çıktıları yorumlayıcıya olabildiğince yakın bir yerde kodlanmalı (encode) veya kaçış karakterleri engellenmelidir.

Modern web uygulama mimarisi ile bazı durumlarda sağlam bir girdi doğrulaması yapmak zorlaşmış ve çıktı kodlama her zamankinden daha önemli bir hal almıştır. Bu nedenle parametrelenmiş sorgular (parameterized queries) kullanılması, otomatik kaçış karakterlerini (auto-escape) engelleyen şablon çerçevelerinin (framework) kullanımı ve çıktı kodlama API'larının dikkatli şekilde kullanımı güvenlik açısından kritik önem taşımaktadır.

V5.1 Girdi Doğrulama Gereksinimleri

Kabul edilen verilere ait bir beyaz liste ile ve güçlü veri türü kontrolleri ile yapılmış girdi doğrulama kontrolleri, tüm enjeksiyon saldırılarının %90'dan fazlasını ortadan kaldırabilir. Buna ek olarak girdiler üzerinde yapılacak uzunluk ve izin verilen aralık kontrolleri bu riski daha da azaltacaktır. Güvenli girdi doğrulama altyapısı; uygulama mimarisi tasarlanırken, müşteri ile tasarım yapılırken, kodlama anında, birim ve entegrasyon testleri yapılırken oluşturulmalıdır. Bu öğelerin birçoğu sızma testleri esnasında tespit edilemese de girdi doğrulama kontrollerinin yapılmamış olması genellikle V5.3 – Çıktı kodlama ve Enjeksiyon Önleme Gereksinimleri içerisinde görülebilir. Uygulama geliştiricilerinin ve kod güvenliği analizi yapanların, enjeksiyon saldırılarını önlemek için, bu maddeye ait tüm öğeleri L1 gereksinimi gibi değerlendirmesi önerilmektedir.

#	Açıklama	L1	L2	L3	CWE
5.1.1	Özellikle uygulama çatısı sorgu parametrelerinin (GET, POST, çerezler, başlıklar) kaynağına ilişkin bir ayrım yok ise uygulamanın HTTP Parametre Kirliliği (HTTP Parameter Pollution) saldırılarına karşı koruma sağladığı doğrulanmalıdır.	✓	✓	✓	235
5.1.2	Eğer uygulama otomatik parametre atamasına izin veriyorsa (otomatik değişken bağlama -automatic variable binding-) hassas alanların (rol, parola, hesap özeti gibi) zararlı parametrelerden korunduğu doğrulanmalıdır.	✓	✓	✓	915
5.1.3	Tüm girdilerin beyaz liste yöntemiyle girdi kontrolüne tabi tutulması gerekmektedir. (HTML form alanları, REST istekleri, URL parametreleri, HTTP başlıkları, çerezler, toplu iş dosyaları, RSS, etc)	✓	✓	✓	20
5.1.4	Kredi kartı veya telefon numarası gibi belirli bir formatı olan verilerin, uzunluk, karakter ve desen özelliklerinin kontrol edildiğine emin olunmalıdır. Belirli bir şema içerisinde bu doğrulamaların güçlü bir şekilde yapıldığından emin olunmalıdır.	✓	✓	✓	20

5.1.5	URL yönlendirmelerinde yalnızca beyaz liste kontrolü ile belirli bir listedeki hedeflere yönlendirme yapıldığı doğrulanmalıdır. Eğer güvenilir olmayan bir içeriğe yönlendirme yapılıyorsa bir uyarı ile bilgilendirme yapılmalıdır.	✓	✓	✓	601

V5.2 Zararlı Girdinin Temizlenme ve İzolasyon Gereksinimleri

#	Açıklama	L1	L2	L3	CWE
5.2.1	WYSIWYG veya benzeri editörlerden gelen güvensiz HTML kodlarının bir HTML temizleyicisi ile uygun şekilde filtrelenmesinden (sanitize) ve doğru bir şekilde "encode" edilmesinden emin olunmalıdır.	✓	✓	✓	116
5.2.2	Herhangi bir yapısal formatı olmayan verilerin de izin verilen karakterler ve içerik uzunluğu kontrolü ile girdi temizlemesine alındığı doğrulanmalıdır.	✓	✓	✓	138
5.2.3	SMTP veya IMAP enjeksiyonlarına karşı koruma sağlamak için uygulamanın kullanıcı girdisini temizledikten sonra e-posta sistemlerine aktarması doğrulanmalıdır.	✓	✓	✓	147
5.2.4	Uygulamanın eval() veya diğer dinamik kod yürütme özelliklerini kullanılmasını önlediği doğrulanmalıdır. Eğer böyle dinamik bir ihtiyaç için alternatifi bulunmuyor ise tüm kullanıcı girdisi çalıştırılmadan önce zararlı karakterler girişlerine karşı temizlenmeli veya kum havuzunda (sandbox) çalıştırılmalıdır.	✓	✓	✓	95
5.2.5	Uygulamanın şablon enjeksiyona saldırılarına karşı her türlü kullanıcı girdisini temizlediği veya kum havuzunda çalıştırdığı doğrulanmalıdır.	✓	✓	✓	94
5.2.6	Uygulamanın SSRF saldırılarına karşı korunma amacıyla güvenilir bir kaynaktan gelmeyen veriler için girdi kontrolü veya temizleme yaptığı, beyaz liste metoduyla HTTP dosya meta verisi, dosya isimleri, URL alanları, protokol, etki alanı, yol ve port değişkenlerini kontrol ettiği doğrulanmalıdır.	✓	✓	✓	918
5.2.7	Uygulamanın kullanıcının sağladığı SVG betik içeriğini özellikle XSS'e yol açabilme ihtimali olması dolayısıyla temizlemesi, yasaklaması veya kum havuzunda çalıştırması doğrulanmalıdır.	✓	✓	✓	159
5.2.8	Uygulamanın kullanıcının sağladığı betikler veya Markdown, CSS, XSL stilleri, BBCode veya benzeri içerikleri temizlemesi, yasaklaması veya kum havuzunda çalıştırması doğrulanmalıdır.	✓	✓	✓	94

V5.3 Çıktı Kodlama ve Enjeksiyon Engelleme Gereksinimleri

Kullanılan yorumlayıcıya (interpreter) olabildiğince yakın bir yerde çıktı kodlama yapılması uygulamanın güvenliği için kritik rol oynamaktadır. Çıktı kodlama kalıcı bir çözüm sunmamaktadır ancak anlık kullanım için, kullanıcıdan alınan verinin, doğru çıkış bağlamında (output context) güvenli şekilde işlenmesini (render) sağlamak için kullanılır. Çıktı kodlama işleminin doğru şekilde yapılmaması güvenli olmayan ve zararlı kod enjekte edilebilir bir uygulama ile sonuçlanacaktır.

#	Açıklama	L1	L2	L3	CWE
5.3.1	Yapılan çıktı kodlamasının yorumlayıcı ve içeriğe uygun olduğu kontrol edilmelidir. Belirli bir formata uymayan verilerin, izin verilen karakterler, veri uzunluğu ve potansiyel zararlı karakterler gibi (örneğin, ねこ veya O'Hara gibi	✓	✓	✓	116

#	Açıklama	L1	L2	L3	CWE
	isimlerde bulunan karakterler gibi) her veri için uygulanabilen güvenlik önlemlerinden geçtiği doğrulanmalıdır.				
5.3.2	Yapılan çıktı kodlamasının kullanıcının seçtiği karakter setine uygun ve güvenli yapılması ve sonuçların karakter setini bozmadan uygulandığı doğrulanmalıdır.	✓	✓	✓	176
5.3.3	Yansıtılmış (reflected), yerleşik (stored) ve DOM tabanlı siteler arası betik çalıştırma (XSS) zafiyetlerini önlemek için içeriğe uygun, tercihen otomatik, en kötü durumda elle çıktı kodlamasının doğru şekilde yapılması gerekmektedir.	✓	✓	✓	79
5.3.4	Bütün SQL sorgularının, HQL, OSQL, NOSQL ve stored procedure'lerin “prepared statement” veya “query parameterization” ile yapıldığı doğrulanmalıdır. Bu sayede SQL injection saldırısının önüne geçilecektir.	✓	✓	✓	89
5.3.5	Parametrik veya güvenli bir mekanizmanın olmadığı durumlarda içeriğe özel SQL Escape işlemi ile çıktı kodlaması yapılarak SQL enjeksiyonu saldırılarına karşı koruma sağlanmalıdır.	✓	✓	✓	89
5.3.6	Uygulamanın uzaktan JavaScript çalıştırma, CSP atlatma, DOM XSS gibi eval, JavaScript veya JSON saldırılarına karşı koruma sağladığı doğrulanmalıdır.	✓	✓	✓	830
5.3.7	Uygulamanın LDAP Enjeksiyonu (LDAP Injection)’na olanak tanımadığı veya güvenlik önlemlerinin LDAP Enjeksiyonu’na karşı koruma sağladığı doğrulanmalıdır..	✓	✓	✓	943
5.3.8	Uygulamanın İşletim Sistemi Komut Enjeksiyonu (OS Command Injection)’na olanak tanımadığı veya işletim sistemi çağrılarında parametrik sorgularla ve içeriğe bağlı komut satırı kodlamasıyla güvenlik önlemleri alınarak İşletim Sistemi Komut Enjeksiyonu’na karşı koruma sağladığı doğrulanmalıdır.	✓	✓	✓	78
5.3.9	Uygulamanın RFI veya LFI saldırılarına olanak tanımadığı veya güvenlik önlemlerinin RFI veya LFI saldırılarına karşı koruma sağladığı doğrulanmalıdır.	✓	✓	✓	829
5.3.10	Uygulamanın XPath sorgu değiştirme, XXE (XML External Entity) ve XML Injection saldırıları gibi bilinen XML saldırılarına karşı önlem aldığı doğrulanmalıdır.	✓	✓	✓	643

Not: Parametreli sorgular kullanmak (parameterized query) veya kaçış karakterlerinin kısıtlanması; ORDER BY cümleciği, tablo ve sütun adları gibi alanlarda yeterli bir çözüm olamamaktadır. Kullanıcıdan alınan verilerin bu gibi alanlarda kullanılması sonucunda SQL sorgularının bozulması ya da ilgili kodların SQL enjeksiyonu zafiyetinden etkilenir hale gelmesi olasıdır.

Not: SVG dosya formatı hemen hemen tüm bağlamlarda (context) ECMA Script kodlarının çalıştırılmasına izin vermektedir. Bu nedenle SVG ile yapılan XSS saldırı vektörlerinin tamamen engellenmesi mümkün olmayabilir. Eğer SVG dosyalarının yüklenmesi gerekiyor ise, yüklenen bu dosyaların kullanıcılara düz metin olarak sunulması veya XSS saldırısının uygulama üzerindeki etkisini azaltmak adına bu dosyaların ayrı bir alan adında saklanması gerekmektedir.

V5.4 Bellek, Katar (String) ve Yönetilemeyen Kod Gereksinimleri

Aşağıdaki gereksinimler, geliştirilen uygulama bir sistem dili veya doğrudan işletim sistemi üzerinde çalıştırılacak bir kod (unmanaged code) kullandığında geçerlidir.

#	Açıklama	L1	L2	L3	CWE
5.4.1	Uygulamanın yığın, tampon veya bellek taşmalarını tespit ve önleme amacıyla güvenlir metin değişkenleri,güvenilir bellek kopyalama ve işaretçi artimetiği kullandığı doğrulanmalıdır.		✓	✓	120
5.4.2	Metin biçimleyicinin (format string) potansiyel tehklikeli girdi ve sabit almadığından emin olunmalıdır.		✓	✓	134
5.4.3	Tam sayı taşmalarını engellemek için işaret, aralık ve girdi kontrol tekniklerinin kullanıldığı doğrulanmalıdır.		✓	✓	190

V5.5 Deserialization Önleme Gereksinimleri

#	Açıklama	L1	L2	L3	CWE
5.5.1	Serileştirilen nesnelerin bütünlük kontrolü yaptığı veya şifrelenmiş olduğu doğrulanarak kötücül nesne oluşturulması veya verinin izinsiz değiştirilmesi engellenmelidir.	✓	✓	✓	502
5.5.2	Uygulamanın XXE'yi önleyebilmek için doğru bir şekilde XML ayrıştırılmasını (parsing) yapacak kısıtlı ayarları uygulayıp güvenilir olmayan dış varlık çözümlemesi özelliğini kullanım dışı bıraktığından emin olunmalıdır.	✓	✓	✓	611
5.5.3	Hem özelleştirilmiş kodda hem de üçüncü taraf kütüphanelerde (JSON, XML veya YAML gibi) güvenilir verilerin tersine serileştirilmesinden kaçınılmalı veya korumalı bir şekilde işlem gerçekleştirilmelidir.	✓	✓	✓	502
5.5.4	İnternet tarayıcısında veya JavaScript arka ucunda JSON ayrıştırılırken (parsing) JSON.parse kullanıldığı doğrulanmalıdır. JSON ayrıştırılırken (parsing) eval() kullanılmamalıdır.	✓	✓	✓	95

Referanslar

Detaylı bilgi için aşağıdaki bağlantıları ziyaret edebilirsiniz:

- OWASP Testing Guide 4.0: Input Validation Testing
- OWASP Cheat Sheet: Input Validation
- OWASP Testing Guide 4.0: Testing for HTTP Parameter Pollution
- OWASP LDAP Injection Cheat Sheet
- OWASP Testing Guide 4.0: Client Side Testing
- OWASP Cross Site Scripting Prevention Cheat Sheet
- OWASP DOM Based Cross Site Scripting Prevention Cheat Sheet
- OWASP Java Encoding Project
- OWASP Mass Assignment Prevention Cheat Sheet
- DOMPurify - Client-side HTML Sanitization Library
- XML External Entity (XXE) Prevention Cheat Sheet)

“auto-escaping” hakkında daha detaylı bilgi için aşağıdaki bağlantıları ziyaret edebilirsiniz:

- Reducing XSS by way of Automatic Context-Aware Escaping in Template Systems
- AngularJS Strict Contextual Escaping
- AngularJS ngBind
- Angular Sanitization
- Angular Template Security
- ReactJS Escaping
- Improperly Controlled Modification of Dynamically-Determined Object Attributes

Deserialization hakkında daha detaylı bilgi için aşağıdaki bağlantıları ziyaret edebilirsiniz:

- OWASP Deserialization Cheat Sheet
- OWASP Deserialization of Untrusted Data Guide

V6 Kriptografi İşlemleri Doğrulama Gereksinimleri

Kontrol Amacı

İlgili maddeleri uygulamış bir uygulamanın aşağıdaki yüksek seviye gereksinimleri karşıladığından emin olunmalıdır:

- Tüm şifreleme modülleri güvenli bir şekilde hata vermeli ve bu hatalar doğru şekilde işlenmelidir.
- Rastgeleliğe (randomness) ihtiyaç duyulduğunda, uygun bir rastgele sayı üreticisi kullanılmalıdır.
- Anahtarlara erişim güvenli bir şekilde yönetilmelidir.

V6.1 Veri Sınıflandırma

Bir uygulamaya ait en önemli varlık; uygulama tarafından işlenen, depolanan veya iletilen verilerdir. Bu veriler üzerinde uygulanacak doğru veri koruma ihtiyaçlarını tespit etmek adına, ilgili verilere ait bir gizlilik etkisi değerlendirmesi yapılmalıdır.

#	Açıklama	L1	L2	L3	CWE
6.1.1	Kişisel tanımlayıcı bilgileri (PII), hassas kişisel bilgileri veya AB'nin GDPR regülasyonu gibi regülasyonlara konu özel verilerin saklanma aşamasında şifrelenerek tutulması gerekmektedir.		✓	✓	311
6.1.2	Sağlık kayıtları, sağlık cihazı detayları veya özel araştırma kayıtları gibi regülasyona tabi sağlık verileri saklanma aşamasında şifreli şekilde tutulmalıdır.		✓	✓	311
6.1.3	Finansal hesap bilgileri, Kredi geçmişi, vergi kayıtları, Ödeme geçmişi, bonuslar veya özel market ve araştırma kayıtları gibi Finansal regülasyonlara tabi veriler saklanma aşamasında şifreli şekilde tutulmalıdır.		✓	✓	311

V6.2 Algoritmalar

Kriptografi alanında yapılan son gelişmeler sonrasında, güvenli olarak bilinen bazı algoritma ve anahtar uzunluklarının, güvenli veri saklamada artık yeterli güvenlikte olmadığı görülmüştür. Bu nedenle bu tip uygulamalar tasarlanırken algoritma ve anahtarlar değiştirilebilir şekilde kullanılmalıdır.

Bu bölüm sızma testi ile kolayca doğrulanamasa da geliştiricilerin (L1 çoğu öğede eksik olmasına rağmen) bu bölümü gereklilik olarak kabul etmesi gerekmektedir.

#	Açıklama	L1	L2	L3	CWE
6.2.1	Tüm şifreleme modüllerinin güvenli bir şekilde hata durumuna geçtiği doğrulanmalıdır. Hatalar 'oracle padding'i etkinleştirmeyecek şekilde işlenmelidir.	✓	✓	✓	310
6.2.2	Yazılımcılar tarafından oluşturulan kriptografik algoritmalar yerine endüstride Kabul görmüş veya devlet standartlarında onaylanmış kriptografik algoritmalar, modlar ve kütüphaneler kullanılmalıdır.		✓	✓	327
6.2.3	Şifreleme başlangıç vektörleri, cipher ayarları ve blok modlarının en son güvenli yapılandırma tavsiyeleri üzerinden konfigüre edildiği doğrulanmalıdır.		✓	✓	326
6.2.4	Kriptografik çözülme saldırılarına karşı rastgele numara, şifreleme ve özet alma algoritmaları, anahtar uzunlukları, cipher ve modların herhangi bir zamanda tekrar konfigüre edilebilmesi, güncellenebilmesi veya değiştirilebilesi gerekmektedir.		✓	✓	326

#	Açıklama	L1	L2	L3	CWE
6.2.5	Geçmiş sürümlerle uyumluluk söz konusu değil ise güvensiz olarak bilinen blok modlarının (ECB gibi), padding modlarının (PKCS#1 v1.5 gibi), küçük blok uzunluklu cipher'ların (Triple-DES, Blowfish gibi) ve zayıf özetleme algoritmalarının (MD5, SHA1 gibi) kullanılmaması gerekmektedir.		✓	✓	326
6.2.6	Notalar, başlangıç vektörleri, ve diğer tek kullanımlık numaraların verilen şifreleme anahtarı ile birden fazla kullanılmadığı doğrulanmalıdır. Oluşturma metodu seçilen algoritma ile uygun olmalıdır.		✓	✓	326
6.2.7	Şifrelenmiş verinin yetkisiz biri tarafından değiştirilmediğinden emin olmak için bu verinin dijital imza, kimlik doğrulamasından geçmiş cipher mod veya HMAC ile doğrulanması gerekmektedir.			✓	326
6.2.8	Eşleştirme, hesaplama veya dönüş işlemlerini de içerek şekilde tüm kriptografik operasyonların bilgi sızıntısını engellemek adına sabit bir sürede tamamlanması gerekmektedir.			✓	385

V6.3 Rastgele Değerler

Yüksek entropi (rastgelelik) oranında sözde-sayı (pseudo-random) üretimi yapılması oldukça zordur. Yüksek entropi (rastgelelilik) oranına sahip kaynakların aşırı kullanımı sonrasında da bu kaynaklar hızlıca tükenecektir. Buna ek olarak daha az entropi (rastgelelilik) oranına sahip kaynaklar da tahmin edilebilir anahtarların üretilmesine sebebiyet verebilir.

#	Açıklama	L1	L2	L3	CWE
6.3.1	Tüm rastgele üretilen sayılar, dosya isimleri, global eşsiz kimlikleyiciler (GUID) ve karakter dizilerinin saldırgan için tahmin edilemez olmasını sağlama açısından şifreleme modülünün onaylanmış rasgele sayı üretecini kullanarak üretildiği doğrulanmalıdır.		✓	✓	338
6.3.2	Rastgele GUID'lerin GUID v4 algoritması ve kripto güvenli pseudo rastgele numara oluşturucu (CSPRNG) oluşturulması sağlanmalıdır. Bu yöntemler dışında oluşturulan rastgele GUID'ler tahmin edilebilmektedir.		✓	✓	338
6.3.3	Uygulamanın ağır yük altında bile yeterli rastgelelikte sayılar üretebildiğinden veya bu gibi durumlarda uygulamanın zarifçe isteği reddettiğinden emin olunmalıdır.			✓	338

V6.4 Hassas Veri Yönetimi

Bu bölüm sızma testi ile kolayca doğrulanamasa da geliştiricilerin (L1 çoğu öğede eksik olmasına rağmen) bu bölümü gereklilik olarak kabul etmesi gerekmektedir.

#	Açıklama	L1	L2	L3	CWE
6.4.1	Güvenli bir şekilde oluşturma, saklama, erişimi kontrol etme ve imha yapabilmek için anahtar kasası çözümü ile sır yönetimi sağlandığı doğrulanmalıdır.		✓	✓	798
6.4.2	Anahtar bilgilerinin uygulamaya doğrudan ifşa edilmeyip bunun yerine izole bir güvenli kripto modülünden alınarak kullanılması doğrulanmalıdır.		✓	✓	320

Referanslar

Detaylı bilgi için aşağıdaki bağlantıları ziyaret edebilirsiniz:

- OWASP Testing Guide 4.0: Testing for weak Cryptography

- OWASP Cheat Sheet: Cryptographic Storage
- FIPS 140-2

V7: Hata Ayıklama ve Kayıt Doğrulama Gereksinimleri

Kontrol Amacı

Hata işleme ve kayıt altına almanın temel amacı; kullanıcılar, yöneticiler ve olayla mücadele ekiplerinden (incident response teams) yararlı aksiyonlar alabilmektir. Amaç çok kayıt tutmak değil, kaliteli ve gürültüden arındırılmış kayıtlar tutmaktır.

Yüksek kalitedeki kayıtlar (loglar) genellikle hassas veriler içerecektir ve kişisel verilerin korunması ile alakalı yasalara uygun bir şekilde korunmalıdır. Bu durum ile alakalı aşağıdaki öneriler incelenebilir;

- Özel olarak istenmiyorsa hassas bilgi toplanmamalı ve kaydedilmemelidir.
- Kayıtlı bilgilerin güvenliği sağlanmalı ve veriler sınıflandırılarak korunmalıdır.
- Kayıtlar süresiz olarak saklanmamalı, saklanma süresi mümkün olduğunca kısa olmalıdır.

Kayıtların içeriği ülkeden ülkeye değişmekle beraber hassas bilgiler içerebilir. Bu gibi durumlar, saldırganlar için çok cazip olmaktadır.

V7.1 Hata Kaydı İçerik Gereksinimleri

Hassas bilgilerin kayıt günlüklerine eklenmesi tehlikeli olabilecek bir durumdur. Kayıtlar; şifrelenmiş halde, veri saklama politikalarına uygun olarak ve güvenlik denetimleri için erişilebilir şekilde tutulmalıdır. Yalnızca gerekli olan bilgilerin kayıt altına alındığına emin olunmalıdır. Bu kayıtlar içerisinde kesinlikle; ödeme bilgisi, kimlik bilgileri (oturum anahtarları da kaydedilmemelidir), hassas veya kişisel bilgilerin sızmasına sebep olabilecek bilgilerin bulunmadığına emin olunmalıdır.

V7.1, OWASP Top 10 2017 - A10 maddesini ele almaktadır. Bu bölüm sızma testi ile doğrulanamadığı için aşağıdaki maddelerin uygulanması önemlidir:

- Bütün öğeler L1 olarak işaretlenmişçesine, uygulama geliştiricilerinin, bu bölüm ile tam uyumluluk sağlaması gerekmektedir.
- Sızma testi gerçekleştiren kişilerin; geliştiriciler ile görüşerek, ekran görüntüleri ile veya ispatlama yolu ile bütün öğelerin V7.2 ile uyumluluk sağladığını doğrulaması gerekmektedir.

#	Açıklama	L1	L2	L3	CWE
7.1.1	Uygulamanın kimlik bilgileri / parolaları veya Ödeme detaylarını kayıt altına almadığı doğrulanmalıdır. Oturum anahtarları sadece geri döndürülemez haliyle özeti alınarak saklanmalıdır.	✓	✓	✓	532
7.1.2	Uygulamanın yerel gizlilik kanunları ve Güvenlik politikalarında tanımlanan hassas verileri sistem kayıtlarında (log) tutmadığı doğrulanmalıdır.	✓	✓	✓	532
7.1.3	Güvenlik kayıt kontrol mekanizmalarının, güvenlik ile ilgili başarılı ve başarılı olmayan olayları, erişim kontrol hataları, tersine serileştirme hataları ve girdi doğrulama hatalarını içeren kayıt kabiliyeti sunduğu doğrulanmalıdır.		✓	✓	778
7.1.4	Kayıt günlüklerinin (loglar) olayın gerçekleştiği anda, bilgileri ve zaman çizelgesini ayrıntılı tuttuğunun doğrulaması yapılmalıdır. Bu günlükler, ayrıntılı incelemelerde yeterince bilgi sağlamalıdır.		✓	✓	778

V7.2 Hata Kaydı İşleme Gereksinimleri

Zamanlı kayıtların oluşturulması olay denetimleri, kayıtların önceliklerinin belirlenmesi ve bir üst yetkiliye iletilebilmesi için önemlidir. Bunu sağlayabilmek için olay kayıtlarının temiz olduğuna, rahatlıkla izlenebildiğine, yerel olarak veya uzak sistemlere alınarak analiz edilebildiğine emin olunmalıdır.

V7.2, OWASP Top 10 2017 - A10 maddesini ele almaktadır. Bu bölüm sızma testi ile doğrulanamadığı için aşağıdaki maddelerin uygulanması önemlidir:

- Bütün öğeler L1 olarak işaretlenmişçesine, uygulama geliştiricilerinin, bu bölüm ile tam uyumluluk sağlaması gerekmektedir.
- Sızma testi gerçekleştiren kişilerin; geliştiriciler ile görüşerek, ekran görüntüleri ile veya ispatlama yolu ile bütün öğelerin V7.2 ile uyumluluk sağladığını doğrulaması gerekmektedir.

#	Açıklama	L1	L2	L3	CWE
7.2.1	Herhangi bir hassas veri, oturum bilgisi veya parolasını içermeyecek şekilde oturum açma istekleri kaydedilmelidir. Güvenlik incelemesinde lazım olacak tüm meta bilgilerin istekten alınması gerekmektedir.		✓	✓	778
7.2.2	Tüm erişim kontrol kararlarının ve başarısız olan istekler kaydedilmelidir. Güvenlik incelemesinde lazım olacak tüm meta bilgilerin istekten alınması gerekmektedir.		✓	✓	285

V7.3 Hata Kaydı Koruma Gereksinimleri

Kolaylıkla değiştirilebilen veya silinebilen kayıtlar adli soruşturma ve davalarda kullanılamamaktadır. Buna ek olarak, kayıtların ifşa olması durumunda uygulamaya ait iç işleyişin veya kullanıcı verilerinin açığa çıkması mümkündür. Bu nedenle kayıtların ifşa olmasının, değiştirilmesinin ve silinmesinin önüne geçilmelidir.

#	Açıklama	L1	L2	L3	CWE
7.3.1	Log enjeksiyonunu engellemek adına kullanıcıların uygulamaya sağladığı verinin doğru şekilde encode edildiği doğrulanmalıdır.		✓	✓	117
7.3.2	Kayıt görüntüleme yazılımında görüntüleme esnasında kayıtların enjeksiyondan korunduğun doğrulanmalıdır.		✓	✓	117
7.3.3	Güvenlik kayıtlarının izinsiz erişimlere ve değişikliklere karşı korunduğu doğrulanmalıdır.		✓	✓	200
7.3.4	Günlüklerin, doğru bir zamana ve yerel saat aralığına sahip olduğunun doğrulanması için zaman kaynaklarıyla senkronize edilmelidir. Eğer sistemler global ise olay sonrası incelemelerde kolaylık sağlaması adına kayıt işlemleri için UTC formatının kullanılması önemle tavsiye edilmektedir.		✓	✓	

Not: 7.3.1 maddesinde değinilen log (kayıt) enjeksiyonunu engellemek için yapılması gereken log kodlamasının (encoding) test edilmesi, dinamik analiz araçları ile analiz edilmesi veya sızma testleri ile bulunması oldukça zordur. Uygulama mimarları, kod geliştiricileri ve kaynak kodu analiz edenlerin log kodlamasını bir L1 gereksinimi olarak düşünmesi ve analiz etmesi gerekmektedir.

V7.4 Hata Ayıklama Yönetimi

Hata yönetiminin amacı günlük kayıt oluşturmaktan çok güvenlik ile ilgili olayları izlemek, önceliklerini belirlemek ve eğer gerekiyorsa bu kayıtları bir üst yetkiliye çıkartmaktır. Güvenlik ile ilgili olaylar kayıt altına alınırken, alınan her kayıtın bir amacı olduğuna ve SIEM ya da analiz yazılımları tarafından ayrıştırılabildiğine emin olunmalıdır.

#	Açıklama	L1	L2	L3	CWE
7.4.1	Beklenmedik veya güvelik hassasiyeti olan bir hata meydana geldiğinde, genel bir hata mesajı gösterilmeli, sonrasında destek personelinin incelemesini sağlamak adına özel bir ID verilmelidir.	✓	✓	✓	210
7.4.2	Tüm kodda beklenen ve beklenmedik hata durumlarını karşılamak için kullanılan bir hata yakalama işleyişi bulunduğu doğrulanmalıdır.		✓	✓	544
7.4.3	Tüm değerlendirmeye girmemiş hataların yakalanıp gerekli hata yakalama işlemini yapacak bir son nokta hata ayıklayıcı bulunduğu doğrulanmalıdır.		✓	✓	460

Not: Swift/Go gibi bazı geliştirme dilleri ve bazı fonksiyonel diller, istisna yakalayıcıları (try-catch bloğu gibi) ve son çare işleyicileri desteklemezler. Bu durumda uygulama mimarları ve geliştiricilerinin güvenlik ile ilgili istisnai ve beklenmedik durumları güvenli bir şekilde gerçekleştirebilmek için ortak güvenli bir kod dizini (pattern) veya kod çerçevesi (framework) kullanmaları gerekecektir.

Referanslar

Detaylı bilgi için aşağıdaki bağlantıları ziyaret edebilirsiniz:

- OWASP Testing Guide 4.0 content: Testing for Error Handling

V8: Veri Koruma Doğrulama Gereksinimleri

Kontrol Amacı

Veri koruması için üç önemli unsur vardır: Gizlilik, Bütünlük ve Erişilebilirlik (İngilizce kısaltması CIA olarak bilinir). Bu standart, sıkılaştırılmış ve gerekli güvenlik önlemlerinin alınmış olduğu güvenli bir sistemde veri güvenliğinin sağlandığını varsayar.

Uygulamalar, kullanıcılarının cihazlarının bir şekilde tehlikede olabileceğini varsaymak zorundadır. Bir uygulama, bilgisayar, telefon veya tablet gibi güvensiz cihazlar arasında veri alışverişi yapılması ve saklanması durumunda; verilerin şifrelenmesinden, kolayca elde edilememesinden ve değiştirilememesinden sorumludur.

Doğrulanmış bir uygulamanın aşağıdaki üst düzey veri koruma gereksinimlerini karşıladığından emin olun:

- **Gizlilik**: Veriler iletimdeyken ya da depolandığında yetkisiz olarak erişilemez ve ifşa edilemez olmalıdır.
- **Bütünlük**: Veriler kötü amaçlı olarak oluşturulamaz, değiştirilemez veya yetkisiz olarak silinemez olmalıdır.
- **Erişilebilirlik**: Veriler yetkili kullanıcılar tarafından her an kullanılabilmelidir.

V8.1 Genel Veri Koruma

#	Açıklama	L1	L2	L3	CWE
8.1.1	Uygulamanın yük dengeleyici veya önbellek gibi sunucu bileşenlerinde hassas verilerin önbelleğe alınmasına karşı koruma sağladığı doğrulanmalıdır.		✓	✓	524
8.1.2	Sunucu üzerinde bulunan önbellekteki verilerin veya geçici tüm hassas verilerin yetkisiz erişimlere karşı korumalı olduğu ya da yetkili kullanıcı tarafından kullanıldıktan sonra temizlendiği/geçersiz kılındığı doğrulanmalıdır.		✓	✓	524
8.1.3	Uygulamanın, gizli alanlar, Ajax değişkenleri, çerezler ve başlık değerleri gibi güvenilmeyen sistemlere gönderilecek olan değişken sayısını minimum seviyede tuttuğu doğrulanmalıdır.		✓	✓	233
8.1.4	Uygulamanın, olağandışı istek sayısı veya ekran dönüşümü (screen scraping), gibi durumları tespit edebildiği ve gerekli yerlere alarm mesajları gönderebildiği doğrulanmalıdır.		✓	✓	770
8.1.5	Önemli veriler için düzenli Yedekleme alındığı doğrulanmalı ve yedeklemeden geriye dönülebildiği test edilmelidir.			✓	19
8.1.6	Yedeklemelerin veri hırsızlığı veya bozulmaya karşı güvenli bir şekilde tutulduğu doğrulanmalıdır.			✓	19

V8.2 İstemci Tarafında Veri Koruma

#	Açıklama	L1	L2	L3	CWE
8.2.1	Uygulamanın, uygun önbellek önleme (anti-caching) başlıklarını yapılandırarak modern tarayıcılarda hassas bilgilerin önbelleğe alınmadığı doğrulanmalıdır.	✓	✓	✓	525
8.2.2	Kullanıcı tarafında HTML5 yerel depolama, oturum depolama, indexedDB, düzenli çerezler veya Flash çerezler gibi hassas veri içeren depolamaların yapılmadığından emin olunmalıdır.	✓	✓	✓	922

#	Açıklama	L1	L2	L3	CWE
8.2.3	Tarayıcı DOM'u gibi istemci tarafında bulunan kimlik doğrulaması sonrası erişilebilir verilerin istemci çıktığında ya da oturumu kapattığında temizlendiği doğrulanmalıdır.	✓	✓	✓	922

V8.3 Hassas Kişisel Veri

Bu bölüm, hassas verilerin yetkisiz olarak ve genellikle toplu bir biçimde oluşturulmasına, okunmasına, güncellenmesine veya silinmesine karşın koruma sağlar.

Bu bölüme uyulması, V4 Erişim Kontrolü ve özellikle V4.2 ile uyumluluğu gerektirir. Örneğin, yetkisiz güncellemelere veya hassas kişisel bilgilerin açıklanmasına karşı korunmak için V4.2.1'e uyulması gerekir. Tam kapsama için lütfen bu bölüme ve V4'e uyun.

Not: BDDK, PCI veya GDPR benzeri doğrudan uygulamaların hassas kişisel bilgileri nasıl sakladığını ve ağ üzerinden ilettiğini etkileyen yasa ve regülasyonlar mevcuttur. Bu yasa ve regülasyonların ağır cezalardan basit tavsiyelere kadar değişebilen yaptırımları bulunmaktadır. Bu yasa ve regülasyonlar uygulanırken nitelikli bir gizlilik uzmanına veya avukata danışılmalıdır.

#	Açıklama	L1	L2	L3	CWE
8.3.1	Hassas verilerin sunucuya HTTP mesaj gövdesinde veya HTTP başlığı ile gönderildiği doğrulanmalıdır. Hiçbir hassas veri URL parametresi olarak gönderilmemelidir.	✓	✓	✓	319
8.3.2	Kullanıcıların istenmesi halinde verilerini slime veya dışa aktarma seçeneği olduğu doğrulanmalıdır.	✓	✓	✓	212
8.3.3	Toplanan ve kullanılacak olan kişisel bilgilerin temiz bir dille kullanıcıya bildirilmesi ve hangi amaçla kullanılacak olursa olsun Kullanıcıdan rıza alınarak ilerlenmesi gerekmektedir.	✓	✓	✓	285
8.3.4	Oluşturulan tüm hassas verilerin yalnızca belirtilen uygulama tarafından kullanıldığı doğrulanmalı ve bir politika ile hassas verilerin nasıl kullanılacağı düzenlenmelidir.	✓	✓	✓	200
8.3.5	Eğer toplanan veri herhangi bir veri koruma düzenlemesine tabi veya veriye erişimin kayıt altına alınması zorunlu ise hassas veriye erişim kayıt altına alınarak denetlenmelidir. (Ancak kayıtları hassas verinin kendisi yazılmamalıdır.)		✓	✓	532
8.3.6	Hassas verilere ihtiyaç duyulmadığında hafızadan hızlıca silinmesi gerekmektedir. Bu bellek okuma (dumping) saldırılarının önüne geçmektedir.		✓	✓	226
8.3.7	Gizlilik ve bütünlük kavramlarına uygun olarak şifrelenmesi gereken hassas veya özel bilgilerin güvenilir algoritmalarla şifrelendiği doğrulanmalıdır.		✓	✓	327
8.3.8	Hassas kişisel bilgilerden silinecek kapsamına alınan eski veya geçerliliği kalmamış kategorisindeki kayıtların otomatik olarak veya duruma göre silinmesi doğrulanmalıdır.		✓	✓	285

Veri koruması düşünülürken; birincil husus, toplu çıkarma (bulk extraction), değiştirme (modification) veya aşırı kullanımla (excessive usage) ilgili olmalıdır. Örneğin, birçok sosyal medya sistemi, kullanıcıların yalnızca günde 100 yeni arkadaş eklemesine izin verir, ancak bu isteklerin hangi sistemden geldiği önemli değildir. Bir bankacılık platformu, dış kuruluşlara 1000 Euro'dan fazla fon transfer ederek saatte 5'ten fazla işlemi engellemek isteyebilir.

Her sistemin gereksinimleri farklı olabilir, bu nedenle "anormal" bir durum üzerinde bir karar verilirken tehdit modeli ve iş riski göz önünde bulundurmalıdır. Bu tür anormal ve çok kez tekrarlanan eylemleri tespit etme, caydırma veya tercihen bloke etme yetenekleri gereksinim tespiti esnasında ön planda tutulmalıdır.

Referanslar

Detaylı bilgi için aşağıdaki bağlantıları ziyaret edebilirsiniz:

- Consider using Security Headers website to check security and anti-caching headers
- OWASP Secure Headers project
- OWASP Privacy Risks Project
- OWASP User Privacy Protection Cheat Sheet
- European Union General Data Protection Regulation (GDPR) overview
- European Union Data Protection Supervisor - Internet Privacy Engineering Network

V9: İletişim Güvenliği Doğrulama Gereksinimleri

Kontrol Amacı

Doğrulanmış bir uygulamanın aşağıdaki yüksek seviye gereksinimleri karşıladığından emin olun:

- Hassas verilerin iletildiği yerlerde TLS ya da daha güvenli şifreleme yöntemleri kullanılmalıdır.
- Her zaman önerilen en güncel algoritmalar ve şifreleme teknikleri tercih edilmelidir.
- Zayıf ya da yakın zamanda desteği sonlanacak algoritmalar ve şifreleme teknikleri kullanılmamalıdır.

V9.1 İletişim Güvenliği Gereksinimleri

Tüm istemci iletişimleri yalnızca şifrelenmiş ve güvenli kabul edilen iletişim yolları üzerinden gerçekleştirilmelidir. Yapılandırma, otomatize araçlar kullanılarak düzenli olarak gözden geçirilmelidir.

#	Açıklama	L1	L2	L3	CWE
9.1.1	Taşıma Katmanı Güvenliği (TLS)'in kullanıldığı doğrulanmalı ve Taşıma Katmanı Güvenliği (TLS) bağlantılarının, başarısız oldukları anda güvensiz veya açık metin bir HTTP bağlantısına dönüşmedikleri doğrulanmalıdır.	✓	✓	✓	319
9.1.2	Çevrimiçi veya güncel TLS test araçları kullanarak güçlü algoritmalar, cipherlar ve protokollerin seçilmesi doğrulanmalı ve sağlanmalıdır.	✓	✓	✓	326
9.1.3	SSLv2, SSLv3, veya TLS 1.0 ve TLS 1.1 gibi TLS ve SSL protokollerinin eski versiyonları, algoritmalar, cipher ve ayarların kullanılmadığı doğrulanmalıdır. En son sürüm TLS kullanımı tercih edilmelidir.	✓	✓	✓	326

V9.2 Sunucu İletişimi Güvenlik Gereksinimleri

Sunucu iletişimleri HTTP'den daha fazlasıdır. İzleme sistemleri, yönetim araçları, uzaktan erişim ve ssh, ara katman yazılımı, veri tabanı, ana bilgisayarlar, ortak veya harici kaynak sistemleri gibi diğer sistemlere ve bu sistemlerden güvenli bağlantılar kurulmalıdır.

#	Açıklama	L1	L2	L3	CWE
9.2.1	Sunucudan ve sunucuya yapılan bağlantıların güvenilir TLS sertifikası kullandığı doğrulanmalıdır. İç ağ ortamında üretilen veya kendinden imzalı (self-signed) sertifikaların kullanıldığı durumlarda yalnızca belirli iç sertifika otoritesine (CA) ve belirli sertifikalara güvenilmelidir. Diğer sertifikalar kabul edilmemelidir.		✓	✓	295
9.2.2	Yönetim portları, izleme, kimlik doğrulama, API, veya web servis çağrımları, veri tabanları, bulut, sunucusuz yapılar, ana bilgisayar, dış ve partner bağlantılarında hem iç hem de dış bağlantı işlemlerinde iletişimi şifrelemek için TLS kullanıldığı doğrulanmalıdır. Sunucu bağlantılarının, başarısız oldukları anda güvensiz veya açık metin bir HTTP bağlantısına dönüşmemesi gerekmektedir.		✓	✓	319
9.2.3	Dış sistemlere yapılan ve hassas veri/bilgiler ya da işlevler içeren her bağlantının yetkilendirildiği doğrulanmalıdır.		✓	✓	287
9.2.4	Online Certificate Status Protocol (OSCP) Stapling gibi bir yöntem ile sertifika iptali yapılabildiği doğrulanmalıdır.		✓	✓	299
9.2.5	Başarısız Taşıma Katmanı Güvenliği (TLS) bağlantılarının kayıt altına alındığı doğrulanmalıdır.			✓	544

Referanslar

Detaylı bilgi için aşağıdaki bağlantıları ziyaret edebilirsiniz:

- OWASP – TLS Cheat Sheet
- **"TLS için kabul edilebilir mod'lar" ile ilgili not**. Daha önceki ASVS sürümlerinde ABD standardı olan FIPS 140-2 önerilmekteydi. Fakat global olarak düşünüldüğünde bu standardı kabul etmek zor, çelişkili ve kafa karıştırıcı olabilir. Bunun yerine standartlara uyum sağlamanın daha iyi bir yöntemi, (https://wiki.mozilla.org/Security/Server_Side_TLS) gibi kılavuzları incelemek, bilinen en iyi yapılandırmaları uygulamak (https://mozilla.github.io/server-side- tls / ssl-config-generator /) ve istediğiniz güvenlik seviyesini elde etmek için sslyze gibi çeşitli güvenlik açığı tarayıcıları veya güvenilir TLS çevrimiçi değerlendirme hizmetleri gibi bilinen TLS değerlendirme araçlarının kullanımı önerilmektedir.

V10: Zararlı Kod Doğrulama Gereksinimleri

Kontrol Amacı

Doğrulanmış bir uygulamanın aşağıdaki yüksek seviye gereksinimleri karşıladığından emin olun:

- Kötü amaçlı etkinlikler, uygulamanın geri kalanını etkilememek için güvenli ve düzgün bir şekilde yönetilmelidir.
- Uygulamanın zaman bombaları (time bombs) ya da diğer zaman bazlı saldırıları içermiyor olduğuna emin olunmalıdır.
- Zararlı ve bilinmeyen hedeflerle iletişim kurulmamalıdır.
- Uygulamalarda arka kapılar, gizli özellikler (Easter Eggs), tekrar edebilecek küçük saldırılar (Salami Attacks) veya saldırgan tarafından kontrol edilebilen mantıksal hatalar bulunmamalıdır.

V10.1 Kod Bütünlüğü Denetimi

Kötü amaçlı koda karşı en iyi savunma "güven, ancak doğrula" yaklaşımıdır. Yetkisiz veya zararlı kod parçasının uygulama kaynak koduna eklenmesi çoğu yargı alanında genellikle ceza gerektiren bir suçtur. Politikalar ve prosedürler, kötü amaçlı koda ilişkin yaptırımları netleştirmelidir.

Yazılım geliştirme takım liderleri, düzenli olarak kod gözden geçirmesi yaparak zararlı kod parçalarının kaynak koda eklenmediğine emin olmalıdırlar.

#	Açıklama	L1	L2	L3	CWE
10.1.1	Zaman fonksiyonları, güvensiz dosya operasyonları ve ağ bağlantıları gibi potansiyel zararlı kod bulundurabilecek İşlemler için kod Analizi aracı kullanıldığı doğrulanmalıdır.			✓	749

V10.2 Zararlı Kod Arama

Kötü amaçlı kodlar çok seyrektir ve tespit edilmesi zordur. Satır satır bir kodu incelemek mantıksal hatalar bulunmasına yardımcı olabilir fakat bazı durumlarda en deneyimli kod analizcileri bile bu hataları fark edemez.

Bu bölümün kaynak kod ve kullanılan üçüncü parti kütüphanelere erişim olmadan tamamlanması imkansızdır

#	Açıklama	L1	L2	L3	CWE
10.2.1	Uygulama kaynak kodu ve üçüncü taraf kütüphanelerin yetkisiz bir erişim veya veri toplama özelliğinin olmadığı doğrulanmalıdır. Eğer benzer işlevsellikler var ise veri toplamadan önce kullanıcının rızasının alınmış olması gerekmektedir.		✓	✓	359
10.2.2	Uygulamanın kişi listesi, kamera, mikrofon veya konum gibi gizlilik arz eden bilgileri ilgilendiren gereksiz veya fazla izinleri talep etmediği doğrulanmalıdır.		✓	✓	272
10.2.3	Uygulamanın ve üçüncü taraf kütüphanelerin tespit edildiğinde kötü amaçla kullanılabilecek açık metin hesap veya parola bilgisi, kod karmaşıklaştırma, belgelenmemiş derlenmiş blok, rootkit veya anti ayıklama, güvensiz ayıklama özellikleri veya güvensiz, gizli, geçerliliği dolmuş fonksiyon içermediği doğrulanmalıdır.			✓	507

#	Açıklama	L1	L2	L3	CWE
10.2.4	Uygulama ve üçüncü taraf kütüphanelerinin, tarih ve zamana bağlı fonksiyonları ile ilişkili zaman bombası içermediği doğrulanmalıdır.			✓	511
10.2.5	Uygulama ve üçüncü taraf kütüphanelerinin, salami saldırıları, mantıksal atlatmalar veya mantık bombaları gibi kötücül kod içermediği doğrulanmalıdır.			✓	511
10.2.6	Uygulama ve üçüncü taraf kütüphanelerinin, potansiye olarak zarar verebilecek, istenmeyen ve gizli bir fonksiyon (Easter Egss, vb.) içermediği doğrulanmalıdır.			✓	507

V10.3 Dağıtılmış (Deployed) Uygulamanın Bütünlük Denetimi

Bir uygulama üretim ortamına alındıktan sonra zararlı kod yine de eklenebilir. Uygulamaların kendilerini alt alan adı devralmaları (sub-domian takeover) ve güvenilmeyen kaynaklardan imzasız kod yürütme gibi yaygın saldırılara karşı korunması gerekir.

#	Açıklama	L1	L2	L3	CWE
10.3.1	Eğer uygulama istemci veya sunucu tabanlı otomatik güncelleme özelliğine sahip ise güncellemelerin güvenli kanallar üzerinden alındığı ve dijital olarak imzalandığı doğrulanmalıdır. Güncelleme kodu gelen güncelleme dosyası yüklenmeden ve çalıştırılmadan önce dijital imzanın geçerliliğini doğrulamalıdır.	✓	✓	✓	16
10.3.2	Uygulamanın kod imzalama veya kaynak bütünlüğü kontrolü ile bütünlük koruması sağladığı doğrulanmalıdır. Uygulamanın internetten veya güvensiz kaynaklardan modül, eklenti veya kütüphane yüklemesi yapmaması veya çalıştırmaması gerekmektedir.	✓	✓	✓	353
10.3.3	Uygulamanın, DNS kayıtları veya DNS alt alanlarına güvendiği durumlarda karşılaşılabilen süresi dolmuş alan adları, geçerliliği bitmiş DNS veya CNAME değerleri, süresi dolmuş projeler için internete açık kaynak kod repoları veya bulut API'ler, sunucusuz fonksiyonlar vb. gibi alt alan adı devralmaya karşı koruma sağladığı doğrulanmalıdır. Korumanın uygulama tarafından kullanılan DNS isimlerini sürekli olarak geçerlilik ve değişimlere karşı sorgulamayı içermesi gerekmektedir.	✓	✓	✓	350

Referanslar

- Hostile Sub-Domain Takeover, Detectify Labs
- Hijacking of abandoned subdomains part 2, Detectify Labs

V11: İş Mantığı Doğrulama Gereksinimleri

Kontrol Amacı

Doğrulanmış bir uygulamanın aşağıdaki yüksek seviye gereksinimleri karşıladığından emin olun:

- İş mantığı akışı ardışıktır ve sırayla ilerler.
- İş mantığı, sürekli küçük para transferleri veya bir seferde bir milyon arkadaş eklenmesi gibi otomatik saldırıları tespit etmek ve önlemek için sınırlar içerir.
- Yüksek değerli iş mantığı akışları kötüye kullanım vakalarını ve kötü niyetli aktörleri değerlendirmiştir. Sahtecilik(Spoofing), manipüle etme(tampering), geri çevirme(repudiation), bilginin açığa çıkması ve yetki yükseltme saldırılarına karşı koruma sağlamıştır.

V11.1 İş Mantığı Güvenlik Gereksinimleri

İş mantığı problemleri her uygulama için, o uygulamaya özel maddeler içermesi gerekeceğinden genel bir kontrol listesi kullanılması çok zordur. İş mantığı problemleri WAF kullanmak ya da güvenli iletişim kanallarının kullanılmasını sağlamak gibi yöntemler ile çözülememektedir. Tasarım sprintleri sırasında, örneğin OWASP Cornucopia veya benzeri araçların kullanılmasıyla tehdit modellemesi yapılması ve olası risklerin erken safhalarda tespit edilerek çözülmesi önerilmektedir.

#	Açıklama	L1	L2	L3	CWE
11.1.1	Uygulamanın iş mantığı akış adımlarını sıralı ve başka bir kullanıcıya ait adımları işlemeyerek ve adım sırasını bozmadan devam ettiği doğrulanmalıdır.	✓	✓	✓	841
11.1.2	Uygulamanın tüm iş mantığı akış adımlarını gerçekçi insan zamanlamasına göre yaptığı ve çok hızlı gönderilen işlemleri işlemediği doğrulanmalıdır.	✓	✓	✓	779
11.1.3	Uygulamanın her bir kullanıcı nezdinde uygun ve belirli bir iş aksiyon veya işlem limitini doğru bir şekilde tanımladığı doğrulanmalıdır.	✓	✓	✓	770
11.1.4	Uygulamanın veri kaybı, iş mantığını zorlayan sayıda istek gönderimi, çok sayıda dosya yükleme veya hizmet engelleme (DoS) saldırılarını tespit ve koruma amaçlı yeterli kabiliyette anti-otomasyon kontollerine sahip olduğu doğrulanmalıdır.	✓	✓	✓	770
11.1.5	Uygulamanın tehdit modelleme veya benzer metodolojilerle ortaya koyduğu iş riskleri veya tehditlere karşı gerekli iş mantığı limit veya doğrulamalarına sahip olduğu doğrulanmalıdır.	✓	✓	✓	841
11.1.6	Uygulamanın hassas operasyonlarda TOCTOU "time of check to time of use" veya benzer bir “race condition” dan etkilenmediği doğrulanmalıdır.		✓	✓	367
11.1.7	Uygulamanın iş mantığı bakış açısıyla alışılmadık olay ve aktiviteleri gözlemlediği doğrulanmalıdır. Örneğin, normal bir kullanıcının asla denemeyeceği anormal veya kullanım dışı işlemler gibi.		✓	✓	754
11.1.8	Uygulamanın otomatize saldırıları veya alışılmadık aktiviteleri yakalayacak ve bildirecek bir ayarlamaya sahip olduğu doğrulanmalıdır.		✓	✓	390

Referanslar

Detaylı bilgi için aşağıdaki bağlantıları ziyaret edebilirsiniz:

- OWASP Testing Guide 4.0: Business Logic Testing
- OWASP Cheat Sheet
- OWASP Automated Threats to Web Applications
- OWASP AppSensor
- OWASP Cornucopia

V12: Dosya ve Kaynakların Doğrulama Gereksinimleri

Kontrol Amacı

Doğrulanmış bir uygulamanın aşağıdaki yüksek seviye gereksinimleri karşıladığından emin olun:

- Güvensiz dosya verileri güvenli bir şekilde işlenmelidir.
- Güvenilmeyen kaynaklardan elde edilen dosya veya kaynaklar kök dizini (webroot) dışında ve sınırlı izinlerde saklanır.

V12.1 Dosya Yükleme Gereksinimleri

Her ne kadar "zip bomb" gibi zafiyetler sızma testi esnasında kolayca test edilebilir olsa da L2 ve daha üst seviyedeki maddeler için daha detaylı manuel testler yapma ihtiyacı doğabilir.

#	Açıklama	L1	L2	L3	CWE
12.1.1	Uygulamanın saklama alanını gereksiz dolduracak veya Hizmet kesintisine yol açabilecek türden büyük boyutlu dosyaları Kabul etmediği doğrulanmalıdır.	✓	✓	✓	400
12.1.2	Sıkıştırılmış dosyaların, açma işlemi ile dosya saklama limitlerini aşabilecek çok büyük boyutlu dosyalara ulaşmasını engellemek adına küçük boyutlu sıkıştırılmış dosyaların "zip" bombasına karşı kontrol edildiği doğrulanmalıdır.		✓	✓	409
12.1.3	Tek bir kullanıcının saklama alanını bir çok dosyalar veya çok büyük boyutlu dosya ile doldurmasını önlemek adına her bir kullanıcı özelinde bir dosya boyutu kotası veya dosya sayısı kısıtlaması uygulandığı doğrulanmalıdır.		✓	✓	770

V12.2 Dosya Bütünlüğü Gereksinimleri

#	Açıklama	L1	L2	L3	CWE
12.2.1	Güvenilmeyen kaynaklardan elde edilen dosyaların içeriklerine bağlı olarak beklenilen tipte olup olmadıkları doğrulanmalıdır.		✓	✓	434

V12.3 Dosya Çalıştırma Gereksinimleri

#	Açıklama	L1	L2	L3	CWE
12.3.1	Kullanıcıların gönderdiği dosya ismi verisinin sistem veya çatı (framework) tarafından ve URL API'si olarak doğrudan kullanılmayarak dizin gezinme/atlatmaya karşı koruma sağlandığı doğrulanmalıdır.	✓	✓	✓	22
12.3.2	Kullanıcıların gönderdiği dosya ismi verisinin doğrulanması veya hiç kullanılmaması sağlanarak yerel dosyaların ifşası, yeni dosya üretme, güncelleme ve silme işlemlerine engel olunduğu doğrulanmalıdır.	✓	✓	✓	73
12.3.3	Kullanıcıların gönderdiği dosya ismi verisinin doğrulanması veya hiç kullanılmaması sağlanarak veri ifşası ve SSRF zafiyetine yol açabilmesi muhtemel uzaktan dosya çalıştırma (RFI) durumlarının engellendiği doğrulanmalıdır.	✓	✓	✓	98
12.3.4	Uygulamanın kullanıcıların gönderdiği JSON, JSONP veya URL parametrelerini dosya ismi doğrulaması veya hiç kullanılmaması ile yansımalı dosya indirmeye (RFD) karşı önlem alındığı doğrulanmalıdır. Cevapta "Content-Type" başığı text/plain olarak verilmeli ve "Content-Disposition" başlığı ise sabit bir dosya ismi olmalıdır.	✓	✓	✓	641

#	Açıklama	L1	L2	L3	CWE
12.3.5	OS komut enjeksiyonuna karşı koruma sağlamak için güvenilmeyen dosya meta verilerinin doğrudan sistem API'si veya kütüphaneleri ile kullanılmadığı doğrulanmalıdır.	✓	✓	✓	78
12.3.6	Uygulamanın doğrulanmamış içerik dağıtıcı ağlar, JavaScript kütüphaneleri, node npm kütüphaneleri veya sunucu taraflı DDL'leri gibi güvenilmeyen kaynaklardan herhangi bir kaynağı dahil etmemesi veya bir fonksiyonunu çalıştırmaması doğrulanmalıdır.		✓	✓	829

V12.4 Dosya Saklama Alanı Gereksinimleri

#	Açıklama	L1	L2	L3	CWE
12.4.1	Güvenilmeyen kaynaklardan alınan dosyaların web root klasörü dışında, kısıtlı izinlerle, tercihen güçlü bir doğrulama sonrasında saklanması gerekmektedir.	✓	✓	✓	922
12.4.2	Güvenilmeyen kaynaklardan elde edilen dosyaların, zararlı içerik yüklenmesini engellemek amacıyla anti virüs sistemleri tarafından taramadan geçirildiği doğrulanmalıdır.	✓	✓	✓	509

V12.5 Dosya İndirme Gereksinimleri

#	Açıklama	L1	L2	L3	CWE
12.5.1	İstenmeyen bilgi kaybı ve kaynak kod sızıntılarına önlem olması açısından web katmanının sadece belirli dosya uzantıları için çalışması sağlanmalıdır. Örneğin, yedekleme dosyaları (.bak), geçici çalışma dosyaları (.swp), sıkıştırılmış dosyalar (.zip,.tar) ve editörler tarafından ortak kullanılan uzantılar gerekmedikçe bloklanmalıdır.	✓	✓	✓	552
12.5.2	Yüklenen dosyalara doğrudan yapılan istekler asla HTML/JavaScript içerik olarak çalıştırılmamalıdır.	✓	✓	✓	434

V12.6 SSRF (Sunucu Taraflı İstek Sahteciliği) Koruma Gereksinimleri

#	Açıklama	L1	L2	L3	CWE
12.6.1	Web veya Uygulama sunucusunun, istek yollayabileceği veya veri/dosya yükleyebileceği kaynak ve sistemlerin belirli bir beyaz liste ile ayarlanmış olduğu doğrulanmalıdır.	✓	✓	✓	918

Referanslar

Detaylı bilgi için aşağıdaki bağlantıları ziyaret edebilirsiniz:

- File Extension Handling for Sensitive Information
- Reflective file download by Oren Hafif
- OWASP Third Party JavaScript Management Cheat Sheet

V13: API ve Web Servisleri Doğrulama Gereksinimleri

Kontrol Amacı

Güvenilir hizmet katmanı API'larını (genellikle JSON, XML ya da GraphQL) kullanan uygulamaların aşağıdaki özelliklere sahip olduğu doğrulanmalıdır:

- Tüm web servislerinin yeterli kimlik doğrulaması, oturum yönetimi ve yetkilendirilmesi.
- Düşük seviyeden yüksek güven seviyesine geçen tüm parametrelerin girdi doğrulaması.
- Bulut (Cloud) ve Sunucusuz (Serverless) API dahil tüm API türleri için etkili güvenlik kontrolleri

Bu bölüm içerisinde, mükerrer maddeler oluşmaması adına yetkilendirme gibi konular tekrar paylaşılmamıştır. Lütfen, bu bölümü okurken diğer bölümlerdeki ilgili konu başlıklarını da inceleyin.

V13.1 Genel Web Servis Güvenlik Doğrulama Gereksinimleri

#	Açıklama	L1	L2	L3	CWE
13.1.1	Tüm uygulama bileşenleri arasında aynı kodlama(encoding) ve bölütleme (parser) metodunun kullanılarak farklı URI veya dosya bölütleme ile SSRF ve RFI saldırılarının engellendiği doğrulanmalıdır.	✓	✓	✓	116
13.1.2	Web servis uygulaması içindeki yönetim alanının ve yönetim fonksiyonlarının yalnızca servis yöneticileri tarafından erişilebildiğinin doğrulaması yapılmalıdır.	✓	✓	✓	419
13.1.3	API URL'lerinin API veya oturum anahtarları gibi hassas bilgileri ifşa etmedikleri doğrulanmalıdır.	✓	✓	✓	598
13.1.4	Yetkilendirme kararları hem URI'da yazılımsal olarak veya denetleyici ve yönlendiricide bildirimsel olarak, hem de kaynaklar seviyesinde model bazlı yetkilendirmelerle verildiği doğrulanmalıdır.		✓	✓	285
13.1.5	Beklenmeyen veya eksik içerik tipleriyle gelen isteklerin uygun HTTP başlıkları ile reddedilmesi gerekmektedir. (HTTP cevap statüsü 406 Unacceptable veya 415 Unsupported Media Type)		✓	✓	434

V13.2 RESTful Web Servis Doğrulama Gereksinimleri

JSON şema doğrulaması henüz standardizasyonun taslak aşmasındadır. (bknz Referanslar). SOAP web servisleri için en iyi pratiklerden olan JSON şema doğrulaması kullanılırken, aşağıdaki veri doğrulama stratejileri ile kullanılması önerilmektedir:

- JSON objeleri ayrıştırılırken (parsing), eksik ya da fazla eleman içerip içermediğinin kontrolü.
- Veri türü, veri biçimi, uzunluk vb. standart girdi doğrulama yöntemleri kullanılarak JSON obje elemanlarının doğrulanması.
- Ve resmi JSON şema doğrulaması.

JSON şema doğrulama standardı resmileştirildikten sonra, ASVS bu alandaki tavsiyesini güncelleyecektir. Kullanımdaki tüm JSON şema doğrulama kütüphanelerini dikkatlice takip edin, çünkü standart resmileştirilinceye ve hatalar referans uygulamalardan temizlenene kadar düzenli olarak güncellenmeleri gerekecektir.

#	Açıklama	L1	L2	L3	CWE
13.2.1	Korunan API ve kaynaklar üzerinde normal kullanıcıların DELETE veya PUT gibi aksiyonları kullanmasının engellenmesi ile geçerli RESTful HTTP metotlarının etkin kılındığı doğrulanmalıdır.	✓	✓	✓	650

13.2.2	JSON şemalarına girdinin alınmadan önce doğrulandığından emin olunmalıdır.	✓	✓	✓	20
13.2.3	Çerez kullanan RESTful servislerinin CSRF saldırılarından korunduğundan emin olunmalıdır. Önlem olarak ORIGIN kontrolleri, çerez desenini çift gönderme, CSRF notaları ve yönlendirme kontrollerinden en az bir veya ikisi kullanılmalıdır.	✓	✓	✓	352
13.2.4	Özellikle kimlik doğrulaması olmadan çalışan API'ler başta olmak üzere REST servislerinin anti otomatizasyon kontrolleri sağlayarak çok fazla sayıda çağrıma karşı korunduğun doğrulanmalıdır.		✓	✓	779
13.2.5	REST servis için beklenen data'nın application/xml veya application/JSON gibi açıkça tanımlı olduğu kontrol edilmelidir.		✓	✓	436
13.2.6	Mesaj başlıkları ve içeriğin (payload) iletimde değiştirilmemiş olduğundan emin olunmalıdır. Bir çok durumda gizlilik ve bütünlük ile güçlü bir taşıma anı şifrelemesi ile koruma sağlayan TLS yeterli olacaktır. Her mesaj için dijital imza kullanımı ekstra güvence sağlayacaktır ancak karmaşıklık ve bazı risklerle beraber durum dengelenmektedir.		✓	✓	345

V13.3 SOAP Web Servis Doğrulama Gereksinimleri

#	Açıklama	L1	L2	L3	CWE
13.3.1	XSD şema doğrulaması yapılarak dokümanın doğru XML formunda olduğundan emin olunmalı, takiben veri geçen tüm girdi alanları işlenmeden önce doğrulanmalıdır.	✓	✓	✓	20
13.3.2	Mesaj içeriği, istemci ve sunucu arasında güvenilir iletimi sağlamak adına WS-Security ile imzalanmalıdır.		✓	✓	345

Not: DTD'lere yönelik XXE saldırılarıyla ilgili sorunlar nedeniyle, DTD doğrulaması kullanılmamalıdır ve çerçeve DTD değerlendirmesi V14 yapılandırmasında belirtilen gereksinimlere göre devre dışı bırakılmalıdır.

V13.4 GraphQL ve Diğer Web Servisler Veri Katmanı Doğrulama Gereksinimleri

#	Açıklama	L1	L2	L3	CWE
13.4.1	GraphQL veya karmaşık ve iç içe sorguların veri katmanı ifadeleri hizmet kesintisi oluşturmasını engellemek adına sorgular için belirli bir liste kullanımı (beyaz liste metodu) veya derinlik ve tutar limiti kullanılmalıdır. Daha karmaşık senaryolarda, sorgu maliyet analizi kullanılmalıdır.		✓	✓	770
13.4.2	GraphQL veya diğer veri katmanı yetkilendirme mantığının, GraphQL katmanı yerine iş mantığı katmanında olduğu doğrulanmalıdır.		✓	✓	285

Referanslar

Detaylı bilgi için aşağıdaki bağlantıları ziyaret edebilirsiniz:

- OWASP Serverless Top 10
- OWASP Serverless Project
- OWASP Testing Guide 4.0: Configuration and Deployment Management Testing
- OWASP Cross-Site Request Forgery cheat sheet
- OWASP XML External Entity Prevention Cheat Sheet - General Guidance* JSON Web Tokens (and Signing)
- REST Security Cheat Sheet
- JSON Schema
- XML DTD Entity Attacks
- Orange Tsai - A new era of SSRF Exploiting URL Parser In Trending Programming Languages

V14: Konfigürasyon Doğrulama Gereksinimleri

Kontrol Amacı

Doğrulanmış bir uygulamanın aşağıdaki özelliklere sahip olması gerekmektedir:

- Güvenli, tekrar ve otomatize edilebilen bir build yönetim süreci.
- Üçüncü parti kütüphaneler için bağımlılık ve konfigürasyon yönetimi sıkılaştırılmış bir şekilde uygulanıyor olmalıdır. Örneğin güncel ya da güvenli olmayan kütüphanelerin kullanımına izin verilmiyor olmalıdır.
- Kendiliğinden güvenli bir konfigürasyon sistemi kurulmalıdır.

V14.1 Build

Build pipeline'ları tekrarlanabilir güvenliğin temelini oluşturmaktadır, bir güvenlik zafiyeti tespit edildiğinde ve kaynak kodda çözüldüğünde otomatik olarak test edilebilmesine imkan sağlar. Kod üretim ortamına taşınmadan önce, build pipeline'ları kullanılarak önemli bir güvenlik zafiyeti tespit edildiğinde uyarı veren ya da build'in kırılmasını sağlayan yapıların kurulması tavsiye edilmektedir. Düzensiz olarak yapılan analizler bazı zafiyetlerin gözden kaçırılmasına sebep olabilir.

Günümüzde, kurumlar DevSecOps modelini uygulamaya başladıkça, herhangi bir hacklenme vakası olsa dahi bilinen en iyi versiyona çok kısa bir sürede dönülebilmektedir. Eğer, geleneksel modeller kullanılmaya devam ediliyor ise, herhangi olumsuz bir durum oluşması durumunda bilinen en iyi duruma dönebilmek için standartların hazırlanmış olması ve düzenli periyotta backup alınıyor olması önerilmektedir.

Bu bölümdeki maddelerin sağlanabilmesi için, kurumda otomatik derleme sistemleri (automated build system) kullanılıyor olmalı ve kullanılan build ve deployment betiklerine erişilebiliyor olmalıdır.

#	Açıklama	L1	L2	L3	CWE
14.1.1	Uygulamanın derleme ve yayınlama işlemlerinin güvenli ve tekrarlanabilir bir CI/CD otomasyonuyla, otomatikleştirilmiş ayarlarla ve otomatik betiklerle yapıldığı doğrulanmalıdır.		✓	✓	
14.1.2	Tüm derleme bayraklarının yığın rastgeleleştirme, ve veri çalıştırma engellemeyi (DEP) de içerecek şekilde bellek taşmalarına karşı koruma sağlaması, eğer herhangi bir güvensiz operasyon (işaretçi,tamsayı, metin,metin formatı,vb.) tespit edilirse derlemeyi durdurması gerekmektedir.		✓	✓	120
14.1.3	Kullanılan uygulama sunucusu ve yazılım çatısına bağlı olarak sunucu ayarlarının tavsiye edilen şekilde sıkılaştırılması gerekmektedir.		✓	✓	16
14.1.4	Otomatize yayınlama betikleri kullanılarak dokümante ve test edilmiş bir metotla uygulama, konfigürasyon ve tüm bağımlılıkların makul bir sürede tekrar yayınlanabilmesi sağlanabilmeli veya zamanlıca bir yedekten geri dönülebilmelidir.		✓	✓	
14.1.5	Yetkili yöneticilerinin, güvenlik konfigürasyonlarının değiştirilmesi/manipule edilmesi (tampering) gibi durumları denetleyebildiklerine emin olunmalıdır.			✓	

V14.2 Bağımlılıklar

Bağımlılık yönetimi, uygulamaların güvenli olarak çalışabilmeleri için kritik öneme sahiptir. Bugüne kadar gerçekleşen maddi hasarı en büyük çoğu zafiyetin ana sebebini zafiyetli ya da güncel olmayan kütüphaneler oluşturmaktadır.

#	Açıklama	L1	L2	L3	CWE
14.2.1	Tüm bileşenlerin güncel olduğu tercihen bir bağımlılık kontrol aracı kullanılarak derleme zamanında doğrulanmalıdır.	✓	✓	✓	1026
14.2.2	Örnek uygulamalar, platform dokümanları veya varsayılan ve örnek/test amaçlı kullanıcılar gibi tüm gereksiz özelliklerin, belgelerin, örneklerin ve ayarların kaldırılması gerekmektedir.	✓	✓	✓	1002
14.2.3	Tüm uygulama kaynaklarının yine uygulama tarafından barındırıldığından emin olunmalıdır. Örneğin; JavaScript kütüphaneleri, CSS stylesheets ve web yazı karakterlerini CDN veya dış kaynaktan sağlamak yerine uygulamanın kendi içerisinde barındırılmalıdır.	✓	✓	✓	714
14.2.4	Üçüncü taraf bileşenlerin önceden tanımlı, güvenilir ve sürekli güncellenen depolardan geldiği doğrulanmalıdır.		✓	✓	829
14.2.5	Tüm üçüncü parti kütüphaneler için envanter kataloğu tutulduğu doğrulanmalıdır.		✓	✓	
14.2.6	Kum havuzu veya enkapsülasyon yapılarak üçüncü parti kütüphanelerin Uygulama tarafından ihtiyaç duyulan davranışı sergilediği doğrulanarak saldırı yüzeyi daraltılmalıdır.		✓	✓	265

V14.3 İstenmeyen Bilgi İfşası Önleme Gereksinimleri

Üretim ortamındaki uygulamalar; detaylı hata mesajı ifşası, XSS ve RFI gibi bilinen zafiyetlere engel olabilecek şekilde ayarlanmış olmalıdır. Bilgi ifşasına sebep olan zafiyetlerin birçoğu tek başına düşük riskli olsa da farklı zafiyetler ile kullanıldığında etkileri artabilmektedir.

#	Açıklama	L1	L2	L3	CWE
14.3.1	Herhangi bir planlanmamış güvenlik açığına sebebiyet vermemek adına web, uygulama sunucusu veya uygulama çatısı hata mesajlarının Kullanıcı aksiyonuna göre ayarlandığı ve bilgi ifşası yapmayacak şekilde düzenlendiği doğrulanmalıdır.	✓	✓	✓	209
14.3.2	Herhangi bir planlanmamış güvenlik açığına sebebiyet vermemek adına web, uygulama sunucusu veya uygulama çatısının üretim ortamında hata ayıklama modunda çalışması engellenmelidir.	✓	✓	✓	497
14.3.3	HTTP başlıkları veya cevaplarının herhangi bir şekilde sistem bileşen versiyon detaylarını ifşa etmemesi gerekmektedir.	✓	✓	✓	200

V14.4 HTTP Güvenlik Başlıkları (Security Headers) Gereksinimleri

#	Açıklama	L1	L2	L3	CWE
14.4.1	Her HTTP cevabının güvenli karakter setini tanımladığı bir içerik tipi başlığı olduğu doğrulanmalıdır. (UTF-8, ISO 8859-1 gibi)	✓	✓	✓	173
14.4.2	Tüm API cevaplarının Content-Disposition: attachment; filename="api.json" değerlerini içerdiği doğrulanmalıdır.	✓	✓	✓	116
14.4.3	İçerik Güvenliği politikasının (CSPv2) HTML, DOM, JSON ve JavaScript enjeksiyonu ile XSS saldırılarına sebebiyet vermeyi önleme adına kullanımda olduğu doğrulanmalıdır.	✓	✓	✓	1021

14.4.4	Tüm cevapların X-Content-Type-Options: nosniff özelliğini içerdiği doğrulanmalıdır	✓	✓	✓	116
14.4.5	HTTP Strict Transport Security başlığının 'Strict-Transport-Security: max-age=15724800; includeSubdomains ' şeklinde bütün isteklerde ve bütün 'subdomain'lerde yer aldığı doğrulanmalıdır.	✓	✓	✓	523
14.4.6	"no-referrer" veya "same-origin" gibi uygun “Referrer-Policy” başlığının eklenmiş olduğu doğrulanmalıdır.	✓	✓	✓	116
14.4.7	Üçüncü parti X-Frame tarafından görüntülenmesi gerekmeyen içerikler için uygun bir X-FRAME-OPTIONS başlığının bulunduğundan emin olunmalıdır.	✓	✓	✓	346

V14.5 HTTP İstek Başlık Gereksinimlerinin Doğrulanması

#	Açıklama	L1	L2	L3	CWE
14.5.1	Uygulama sunucusunun yalnızca uygulama veya API tarafından kullanılan HTTP metotlarını kabul ettiği doğrulanmalıdır.	✓	✓	✓	749
14.5.2	Kaynak başlığı (origin header) saldırgan tarafından kolayca değiştirilebildiği için hiçbir surette kimlik doğrulama veya erişim kontrolü kararlarında belirleyici olarak kullanılmamalıdır.	✓	✓	✓	346
14.5.3	Cross-domain resource sharing (CORS) “Access-Control-AllowOrigin” başlığını kullanarak katı bir beyaz liste ile güvenilir alan adı kontrolü yapıp, boş kaynak (null origin) desteği sağlamadığı doğrulanmalıdır.	✓	✓	✓	346
14.5.4	Bearer anahtarı gibi güvenilir bir vekil sunucu veya SSO cihazı tarafından eklenen HTTP başlıklarının Uygulama tarafından kimlik doğrulamasına tutulduğu doğrulanmalıdır.		✓	✓	306

Referanslar

Detaylı bilgi için aşağıdaki bağlantıları ziyaret edebilirsiniz:

- OWASP Testing Guide 4.0: Testing for HTTP Verb Tampering
- Adding Content-Disposition to API responses helps prevent many attacks based on misunderstanding on the MIME type between client and server, and the "filename" option specifically helps prevent Reflected File Download attacks.
- Content Security Policy Cheat Sheet
- Exploiting CORS misconfiguration for BitCoins and Bounties
- OWASP Testing Guide 4.0: Configuration and Deployment Management Testing
- Sandboxing third party components

Ek A: Sözlük

- **2FA** – Two-factor authentication(2FA) uygulamaya giriş yapılırken ikinci bir erişim kontrolü eklenmesini sağlar.
- **Adres Alanı Düzeni Rastgeleleştirme (ASLR)** – Ara bellek taşması saldırılarına karşı koruma sağlayan bir teknik.
- **Application Security** – Uygulama seviyesindeki güvenlik zafiyetlerine odaklanmaktadır.
- **Application Security Verification** – OWASP ASVS kontrol listesi kullanılarak yapılan teknik denetimi ifade eder.
- **Application Security Verification Report** – Denetçinin ASVS denetim sonuçları ve analizlerini paylaşmış olduğu rapor.
- **Authentication** – Sunulan kullanıcı kimliğinin doğrulanmasıdır.
- **Automated Verification** – Zafiyet imzalarını kullanarak sorunları tespit eden otomatize araçların kullanılmasıdır. (dinamik, statik ya da ikisinin birlikte kullanıldığı araçlar)
- **Black box testing** – Uygulamanın kaynak kodu ve nasıl çalıştığı ön bilgileri olmadan, fonksiyonların nasıl çalıştığı tahmin edilerek yapılan güvenlik testidir.
- **Component** – diğer bileşenlerle iletişim kuran ilişkili disk ve ağ arabirimlerine sahip bağımsız bir kod birimi.
- **Cross-Site Scripting** (XSS) – Tipik olarak web uygulamalarında görülen ve istemci tarafındaki içeriğe müdahale edilmesine izin veren bir zafiyet.
- **Cryptographic module** – Şifreleme algoritmaları ya da şifreleme anahtarları üreten donanım veya yazılım.
- **CWE** - Common Weakness Enumeration (CWE), topluluk tarafından geliştirilen ve uygulamaların yayınlanmış güvenlik zafiyetlerinin listelendiği bir platformdur.
- **DAST** – Dinamik uygulama güvenliği testi (DAST), çalışan uygulamada otomatize zafiyet taramasının yapılması işlemidir.
- **Design Verification** – Bir uygulamanın güvenlik mimarisinin teknik olarak değerlendirilmesidir.
- **Dynamic Verification** – Zafiyet imzalarını kullanarak zafiyetlerini tespit eden otomatik araçlar yoluyla yapılan doğrulama.
- **Globally Unique Identifier** (GUID) – Bir yazılımda tanımlayıcı olarak kullanılan benzersiz referans numarası.
- **Hyper Text Transfer Protocol** (HTTP/S) – Dağınık, işbirlikçi hiper-medya bilgi sistemleri için tasarlanmış bir uygulama protokolü. An application protocol for distributed, collaborative, hypermedia information systems. World Wide Web veri iletişimi temelidir.
- **Hardcoded keys** – Doğrudan kaynak kod, yorum satırları ya da dosyaların içerisinde gömülü olarak bulunan ve şifreleme anahtarlarıdır.
- **Input Validation** – Güvenilmeyen kullanıcı girdilerinin doğrulanması ve standartlaştırılmasıdır (canonicalization).
- **Malicious Code** – Bir kodun geliştirilmesi esnasında uygulama sahibinden gizli olarak yerleştirilen ve uygulama güvenliği politikasını bozan kod parçacığı. Zararlı yazılım(malware), virüsler ya da solucanlar (worm)'dan farklıdır.
- **Malware** – Uygulama kullanıcısı ya da yöneticisinin haberi olmadan, uygulama koşarken içine yerleştirilen çalışabilir zararlı kod parçası.

- **Open Web Application Security Project** (OWASP) – Kurumların güvenli uygulamalar geliştirmeleri, güvenli uygulamalar satın almaları ve uygulamaları güvenli bir şekilde sürdürmelerine yardımcı olmak amaçlarını benimsemiş açık bir topluluktur. Bkz: http://www.owasp.org/
- **Personally Identifiable Information** (PII) - Tek bir kişiyi tanımlamak, bağlantı kurmak veya yerini belirlemek için kullanılabilen bilgilerdir.
- **PIE** – Konumdan bağımsız çalıştırılabilir kod parçacığı (PIE), birincil bellekte bir yere yerleştirilen mutlak adresinden bağımsız olarak düzgün çalışan bir makine kodu gövdesidir.
- **PKI** – Ortak Anahtar Altyapısı (PKI), ortak anahtarları ilgili varlık kimliklerine bağlayan bir düzenlemedir. Bağlama, bir sertifika otoritesi (CA) tarafından ve bir sertifika otoritesi (CA) tarafından sertifikaların tescili ve verilmesi işlemiyle kurulur.
- **SAST** – Uygulama kaynak kodundaki sorunları zafiyetlerin imzalarını kullanarak tespit etmeye çalışan otomatik araçların kullanılması.
- **SDLC** – Yazılım geliştirme yaşam döngüsü.
- **Security Architecture** – Uygulama tasarımında güvenlik kontrollerinin nerde ve nasıl kullanıldığını tanımlayan bir soyutlamadır.
- **Security Configuration** – Güvenlik kontrollerinin nasıl kullanılacağını etkileyen işleyiş zamanı uygulama yapılandırmasıdır.
- **Security Control** – Güvenlik kontrolü yapan bir fonksiyon ya da bileşen. (Ör: Erişim yetkisi kontrolü)
- **SQL Injection (SQLi)** – Veri tabanlı uygulamalarda, bir girdi noktasına zararlı SQL cümleciklerinin yerleştirilmesi ile gerçekleştirilen bir kod yerleştirme saldırısıdır.
- **SSO Authentication** – Single Sign On (SSO), bir kullanıcı bir uygulamada oturum açtığında, daha sonra yeniden kimlik doğrulaması yapmak zorunda kalmadan otomatik olarak diğer uygulamalarda oturum açabilmesini sağlar. Örneğin, Google'a giriş yaptığınızda, Youtube, Google Dokümanlar ve Gmail gibi diğer Google hizmetlerine eriştiğinizde otomatik olarak giriş yapabilirsiniz.
- **Threat Modeling** - Önemli teknik iş varlıklarını, güvenlik alanlarını ve tehdit ajanlarını belirlemek için iyileştirilmiş güvenlik mimarilerinin sürekli geliştirilmesidir.
- **Transport Layer Security** – İnternet üzerindeki iletişimde güvenliğin sağlanması için uygulanan şifreleme protokolleridir.
- **URI/URL/URL fragments** – Tekdüze Kaynak Tanımlayıcı(URI), bir ismi ya da web kaynağını tanımlamak için kullanılan karakter dizisidir. Tekdüze Kaynak Yer Belirleyici(URL) ise genellikle bir kaynağa referans olarak kullanılır.
- **Verifier** – Uygulamayı OWASP ASVS gereksinimlerine göre gözden geçiren takım ya da kişidir.
- **Whitelist** – İzin verilen veri ya da operasyonların listesidir. Ör: girdi doğrulamada kabul edilen karakterler listesi.
- **X.509 Certificate** – X.509 sertifikası, genel anahtarın sertifikada yer alan kullanıcı, bilgisayar veya hizmet kimliğine ait olduğunu doğrulamak için yaygın olarak kabul edilen uluslararası X.509 ortak anahtar altyapısı (PKI) standardını kullanan dijital bir sertifikadır.

Ek B: Referanslar

OWASP Core Projects

1. OWASP Top 10 Project: https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project
2. OWASP Testing Guide: https://www.owasp.org/index.php/OWASP_Testing_Project
3. OWASP Proactive Controls: https://www.owasp.org/index.php/OWASP_Proactive_Controls
4. OWASP Security Knowledge Framework: https://www.owasp.org/index.php/OWASP_Security_Knowledge_Framework
5. OWASP Software Assurance Maturity Model (SAMM): https://www.owasp.org/index.php/OWASP_SAMM_Project

Mobile Security Related Projects

1. OWASP Mobile Security Project: https://www.owasp.org/index.php/OWASP_Mobile_Security_Project
2. OWASP Mobile Top 10 Risks: https://www.owasp.org/index.php/Projects/OWASP_Mobile_Security_Project_-_Top_Ten_Mobile_Risks
3. OWASP Mobile Security Testing Guide: https://www.owasp.org/index.php/OWASP_Mobile_Security_Testing_Guide

OWASP Internet of Things related projects

1. OWASP Internet of Things Project: https://www.owasp.org/index.php/OWASP_Internet_of_Things_Project

OWASP Serverless projects

1. OWASP Serverless Project: https://www.owasp.org/index.php/OWASP_Serverless_Top_10_Project

Diğer

1. SecLists Github: https://github.com/danielmiessler/SecLists
2. MITRE Common Weakness Enumeration: https://cwe.mitre.org/
3. PCI Security Standards Council: https://www.pcisecuritystandards.org
4. PCI Data Security Standard (DSS) v3.2.1 Requirements and Security Assessment Procedures: https://www.pcisecuritystandards.org/documents/PCI_DSS_v3-2-1.pdf
5. PCI Software Security Framework - Secure Software Requirements and Assessment Procedures: https://www.pcisecuritystandards.org/documents/PCI-Secure-Software-Standard-v1_0.pdf
6. PCI Secure Software Lifecycle (Secure SLC) Requirements and Assessment Procedures: https://www.pcisecuritystandards.org/documents/PCI-Secure-SLC-Standard-v1_0.pdf

Ek C: Internet of Things Doğrulama Gereksinimleri

Bu bölüm, aslında ana maddelerden birisi olması için hazırlanmıştı fakat OWASP IoT ekibinin aynı başlık için çalışmalarını tamamlaması sonrasında tek başlık için iki farklı standart idame etmenin mantılı olmadığı düşünülerek, ana maddeler içerisinden çıkartılmıştır. Bu başlık ile alakalı daha detaylı bilgi edinmek isteyen kişilerin OWASP IoT project sayfasını ziyaret etmelerini tavsiye ediyoruz.

Kontrol Amacı

Gömülü/IoT cihazlar aşağıdaki şartları sağlamalıdır:

- Barındırdığı sunucu ile aynı seviyede güvenlik gereksinimlerini sağlıyor olmalıdır.
- Cihazda saklanması gereken hassas veriler, donanım destekli depolama birimleri gibi güvenlik elemanları aracılığı ile saklanmalıdır.
- Cihaz üzerinde taşınan tüm hassas veriler, güvenli iletişim kanallarını kullanmalıdır.

Güvenlik Doğrulama Gereksinimleri

#	Açıklama	L1	L2	L3	Since
C.1	USB, UART vs. uygulama katmanı hata ayıklama arabirimlerinin devre dışı bırakıldığı ya da karmaşık bir parola ile korunuyor olduğu doğrulanmalıdır.	✓	✓	✓	4.0
C.2	Kriptografik anahtarlar ve sertifikaların her bir cihaz için eşsiz olduğu doğrulanmalıdır.	✓	✓	✓	4.0
C.3	Eğer destekleniyor ise, gömülü/IoT işletim sisteminde, ASLR ve DEP gibi hafıza koruma kontrollerinin aktif olduğu doğrulanmalıdır.	✓	✓	✓	4.0
C.4	JTAG veya SWD gibi çip üzerinde hata ayıklama arabirimlerinin devre dışı bırakıldığı veya kullanılabilir koruma mekanizmasının etkin ve uygun şekilde yapılandırıldığı doğrulanmalıdır.	✓	✓	✓	4.0
C.5	Eğer destekleniyor ise, cihaz SoC veya CPU'sunda güvenilir yürütmenin uygulandığı ve etkinleştirildiği doğrulanmalıdır.	✓	✓	✓	4.0
C.6	Hassas verilerin, özel anahtarların ve sertifikaların Güvenli Öğe (Secure Element), TPM, TEE (Güvenilir Yürütme Ortamı) içinde güvenli bir şekilde saklandığı veya güçlü şifreleme kullanılarak korunduğu doğrulanmalıdır.	✓	✓	✓	4.0
C.7	Donanım yazılımlarının taşınan verilerin güvenliğini TLS kullanılarak sağladığı doğrulanmalıdır.	✓	✓	✓	4.0
C.8	Donanım yazılımlarının sunucu bağlantılarının dijital imzasını kontrol ediyor olduğu doğrulanmalıdır.	✓	✓	✓	4.0
C.9	Kablosuz iletişimin karşılıklı yetkilendirme ile sağlandığı doğrulanmalıdır.	✓	✓	✓	4.0
C.10	Kablosuz iletişimin şifreli kanallar üzerinden sağlandığı doğrulanmalıdır.	✓	✓	✓	4.0
C.11	Herhangi yasaklı bir C fonksiyonu kullanıldığında, uygun güvenli eşdeğer fonksiyonla değiştirildiği doğrulanmalıdır.	✓	✓	✓	4.0

C.12	Her donanım yazılım bileşeninin, üçüncü taraf bileşenleri, sürümleri ve yayımlanmış güvenlik açıklarını kataloglayan bir yazılım listesi tutulduğu doğrulanmalıdır.	✓	✓	✓	4.0
C.13	Üçüncü parti ikili dosya (binary), kütüphane ve uygulama çatıları da dahil olmak üzere tüm kod bileşenleri için, içerisinde gömülü kimlik bilgilerinin bulunup bulunmadığı ile alakalı bir gözden geçirme işleminin yapılıyor olduğu doğrulanmalıdır.	✓	✓	✓	4.0
C.14	Cihaz üzerinde çalışan donanım yazılımları ve diğer yazılımların işletim sistemi üzerinde yetkisiz komut çalıştırılabilmesine sebep olan “Komut Enjeksiyonu” zafiyetinden etkilenmiyor olduğu doğrulanmalıdır.	✓	✓	✓	4.0
C.15	Donanım yazılımlarının güvenilir sunucuların dijital imzalarını barındırıyor olduğu doğrulanmalıdır.		✓	✓	4.0
C.16	Dış müdahale (tamper) direnci ve/veya algılama özelliğinin varlığı doğrulanmalıdır.		✓	✓	4.0
C.17	Çip üreticisi tarafından sağlanan mevcut Fikri Mülkiyet koruma teknolojilerinin etkinleştirildiği doğrulanmalıdır.		✓	✓	4.0
C.18	Tersine mühendislik yöntemlerine karşın güvenlik önlemlerinin alınmış olduğu doğrulanmalıdır.		✓	✓	4.0
C.19	Cihazın ilk açılışta işletim sistemi imajını yüklemeden önce imza kontrolü yapıyor olduğu doğrulanmıştır.		✓	✓	4.0
C.20	Donanım yazılımı güncelleme işleminin, kontrol zamanı ve kullanım süresi (time-of-check, time-of-use) saldırılarına karşı savunmasız olmadığı doğrulanmalıdır.		✓	✓	4.0
C.21	Donanım yazılım güncellemeleri yapılmadan önce, dijital kod imzalama ve imza kontrolü işlemlerinin yapılıyor olduğu doğrulanmalıdır.		✓	✓	4.0
C.22	Cihazın kullanmakta olduğu donanım yazılım sürümünün, daha eski bir versiyona düşürülemiyor olduğu doğrulanmalıdır.		✓	✓	4.0
C.23	Cihazda rastgele değerler üretmek için kullanılan kütüphanelerin güvenli olduğu, gerçekten rastgele değerler üretebildiği doğrulanmalıdır.		✓	✓	4.0
C.24	Donanım yazılımının belirlenen periyotta otomatik olarak güncellenebilmesine imkân sağlanabildiği doğrulanmalıdır.		✓	✓	4.0
C.25	Kullandığı tespit edildikten ya da geçersiz bir mesaj alındıktan sonra donanım yazılımı ve hassas verilerin yok edildiği doğrulanmalıdır.			✓	4.0
C.26	Yalnızca hata ayıklama arabirimlerini devre dışı bırakmayı destekleyen mikro denetleyicilerin (örn. JTAG, SWD) kullanıldığı doğrulanmalıdır.			✓	4.0
C.27	Yalnızca kapatma ve yan kanal saldırılarına karşı önemli koruma sağlayan mikro denetleyicilerin kullanıldığı doğrulanmalıdır.			✓	4.0
C.28	Devre kartının dış kısımlarında hassas verilerin bulunmadığı doğrulanmalıdır.			✓	4.0
C.29	Yongalar arası iletişimin şifreli olduğunu doğrulanmalıdır.			✓	4.0

C.30	Cihazın kod imzalama kullandığı ve çalıştırılmadan önce kodu doğruladığı doğrulanmalıdır.	✓	4.0
C.31	Tekrar kullanılma ihtiyacı bulunmayan hassas verilerin hafızada sıfır karakteri ile üzerinin yazılarak temizlendiği doğrulanmalıdır.	✓	4.0
C.32	Donanım yazılımlarının, uygulamalar arası izolasyonun sağlanması için kernel container'ları kullanıyor olduğu doğrulanmalıdır.	✓	4.0
C.33	Donanım yazılımlarının -fPIE, -fstack-protector-all, -Wl, -z, noexecstack, -Wl, -z, noexecheap gibi güvenli derleyiciler ile derlenecek şekilde konfigure edildiği doğrulanmalıdır.	✓	4.0
C.34	Eğer mümkün ise, mikro denetçilerin kod koruması yapılacak şekilde konfigure edildiği doğrulanmalıdır.	✓	4.0

Referanslar

Detaylı bilgi için aşağıdaki bağlantıları ziyaret edebilirsiniz:

- OWASP Internet of Things Top 10
- OWASP Embedded Application Security Project
- OWASP Internet of Things Project
- Trudy TCP Proxy Tool