OWASP

Open Web Application Security Project

Uygulama Güvenliği Doğrulama Standardı 3.0.1

Temmuz 2016

Türkçe Çeviri: Mayıs 2018

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.

Telif Hakkı ve Lisans

Versiyon 3.0, 2015

Proje Liderleri	Başyazarlar	Değerlendirenler ve Katkıda Bulunanlar
Andrew van der Stock Daniel Cuthbert	Jim Manico	Abhinav Sejpal Ari Kesäniemi Boy Baukema Colin Watson Cristinel Dumitru David Ryan François-Eric Guyomarc'h Gary Robinson Glenn Ten Cate James Holland Martin Knobloch Raoul Endres Ravishankar S Riccardo Ten Cate Roberto Martelloni Ryan Dewhurst Stephen de Vries Steven van der Baan

Versiyon 2.0, 2014

Proje Liderleri	Başyazarlar	Değerlendirenler ve Katkıda Bulunanlar
Daniel Cuthbert Sahba Kazerooni	Andrew van der Stock Krishna Raja	Antonio Fontes Archangel Cuison Ari Kesäniemi Boy Baukema Colin Watson Dr Emin Tatli Etienne Stalmans

Proje Liderleri	Başyazarlar	Değerlendirenler ve Katkıda Bulunanlar
		Evan Gaustad Jeff Sergeant Jerome Athias Jim Manico Mait Peekma Pekka Sillanpää Safuat Hamdy Scott Luc Sebastien Deleersnyder

Versiyon 1.0, 2009

Proje Liderleri	Başyazarlar	Değerlendirenler ve Katkıda Bulunanlar
Mike Boberski Jeff Williams Dave Wichers	Jim Manico	Andrew van der Stock Barry Boyd Bedirhan Urgun Colin Watson Dan Cornell Dave Hausladen Dave van Stein Dr. Sarbari Gupta Dr. Thomas Braun Eoin Keary Gaurang Shah George Lawless Jeff LoSapio Jeremiah Grossman John Martin John Steven Ken Huang Ketan Dilipkumar Vyas Liz Fong Shouvik Bardhan Mandeep Khera Matt Presson Nam Nguyen Paul Douthit Pierre Parrend Richard Campbell Scott Matsumoto Stan Wisseman Stephen de Vries Steve Coyle Terrie Diaz Theodore Winograd

Önsöz

Uygulama Güvenliği Doğrulama Standardı (ASVS) sürüm 3.0'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 v3.0, 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.

3.0'daki Yenilikler

Sürüm 3.0'da, modern uygulamalara daha iyi uyum sağlayabilmek için; yapılandırma, web servisleri, modern (client) tabanlı uygulamalar, HTML 5 ile birlikte uyumlu ara yüzleri olan 'responsive' uygulamalar ve SAML kimlik doğrulama (authentication) kullanan, RESTful web servisleri kullanan mobil istemciler ile ilgili bölümler ekledik.

Standart üzerinden tekrarlamaları kaldırdık. Örneğin; mobil yazılımcıların aynı maddeleri yinelemesinin önüne geçtik.

CWE (Common Weakness Enumeration) sözlüğü ile eşleşme sağladık. CWE eşleşmesi, zafiyetlerin istismar edilebilme olasılığı ve başarılı istismar sonrası sistem üzerindeki etkisi gibi gerekli güvenlik önlemleri alınmadığı durumlarda; sistem üzerinde oluşabilecek olumsuz etkiyi ve bu olumsuz etkinin nasıl giderilebileceği hakkında bilgi sağlamaktadır.

Son olarak topluluğa erişebilmek için AppSec EU 2015 konferansı sonrasında hakem denetimi (peer review) oturumları düzenledik ve AppSec USA 2015'de yaptığımız final çalışmasında önemli ölçüde geri dönüş aldık. Hakem denetimleri sırasında, kontrollerin anlamındaki değişiklikler önemli ölçüdeyse yeni bir kontrol hazırladık ve eski kontrolü kullanımdan kaldırdık. Kaldırılan kontrollerin yeniden kullanılmaması ve karışıklık olmaması için özen gösterdik, fakat bir karışıklık olması durumunda değişikliklerin detaylarına Ek A'dan ulaşabilirsiniz.

Son olarak v3.0, ASVS tarihinin en büyük değişime uğramış standardıdır. Standartta yapılan güncellemeleri faydalı bulmanızı umuyoruz.

Uygulama Güvenliği Doğrulama Standardının 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.

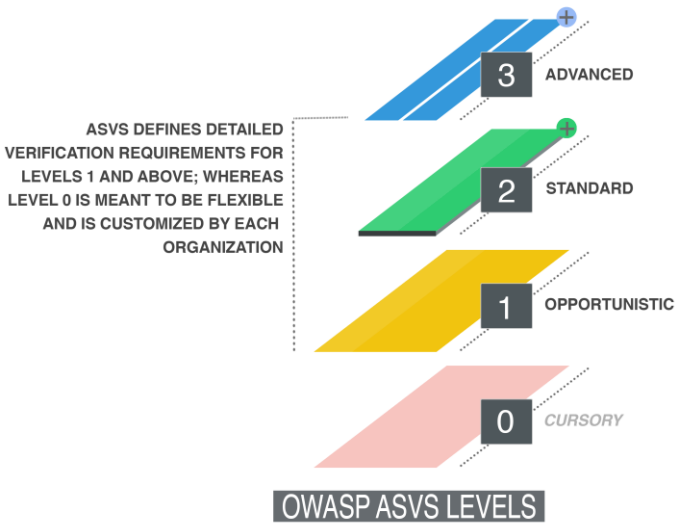

Figure 1 - OWASP Uygulama Güvenliği Doğrulama Standardı 3.0 Seviyeleri

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.

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 veya 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. Seviye 3 seviyesindeki 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 dokümantasyona 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.

Aşağıda, ilgili ASVS seviyeleri için endüstriye özel kılavuzlar sunuyoruz. Her ne kadar her endüstrinin kendine özel kriterleri ve farklılıkları olsa da tüm endüstriler için geçerli olan ortak görüş; internete açık olan uygulamaların fırsatçı saldırganlar tarafından istismar edilebilmesinin her zaman mümkün olduğudur. Bu sebeple ASVS seviye 1, endüstri bağımsız olarak tüm internete açık uygulamalar için tavsiye edilmektedir. Az sayıda risk faktörünü kapsadığı için bu bir başlangıç noktası olarak görülebilir. Organizasyonlar bu seviyeden sonra mutlaka kendi iş alanlarını ilgilendiren risk faktörlerine derinlemesine eğilmelidirler. En üst seviye olan ASVS Seviye 3 ise insan güvenliğini ilgilendiren veya organizasyonun tamamı açısından kritik öneme sahip uygulamalar gibi yüksek risk taşıyan uygulamalar için tavsiye edilmektedir.

Endüstri	Tehdit Profili	S1 Önerileri	S2 Önerileri	S3 Önerileri
Finans ve Sigorta	Her ne kadar bu iş alanındaki uygulamalar ilk aşamada fırsatçı saldırganların ilgisini çekse de özellikle finansal alana motive olmuş tecrübeli saldırganlar da bu iş alanları için büyük bir tehdittir. Saldırganlar genel olarak kullanıcılara ait özel bilgileri ve hesap bilgilerini ele geçirmeye çalışmaktadır. Bunun yanında uygulamalar aracılığı ile yapılan para hareketlerinde sahtekarlık yapmaya çalışmaktadırlar. Teknik olarak kullanıcı giriş bilgilerinin çalınması, uygulama seviyesindeki saldırılar ve sosyal mühendislik teknikleri gibi saldırılar kullanılmaktadır. Bazı büyük uyumluluk hususları (compliance considerations), Payment Card Industry Data Security Standard (PCI DSS), Gramm-Leech Bliley act ve Sarbanes Oxley (SOX)'i kapsamaktadır.	İnternete açık olan tüm uygulamalar.	Kısıtlı yollardan sınırlı miktarlarda paraların taşınmasına sebep olabilecek kredi kartı numarası ve kişisel bilgiler gibi hassas veriler içeren uygulamalar. Örneğin: (i) Aynı kuruma ait hesaplar arasında para transferi, (ii) Transfer limitleriyle daha düşük para hareketi (örn. ACH), (iii) Belirli zaman aralıklarında yüksek transfer limitleri ile havale yapan uygulamalar.	Büyük miktarlarda hassas bilgi içeren ya da hızlı bir şekilde büyük miktarda para transferi (Örn. Havale) yapan, bireysel işlemler şeklinde yapılan küçük para transferleriyle büyük miktarda para transferine sebep olan uygulamalar.
Üretim, Profesyonel Ulaşım, Teknoloji, Kamu Kuruluşları, Altyapı ve Savunma	Bu endüstriler ilk bakışta pek fazla ortak nokta barındırmıyor gibi gözükebilir. Ancak bu bölümdeki kuruluşlara saldırmak isteyen saldırganların; daha fazla zaman, beceri ve kaynaklarla daha kararlı saldırılar gerçekleştirme olanakları yüksektir. Bu alanlarda, hassas bilgilerin ve sistemlerin bulunması kolay değildir ve genellikle bunların tespiti için içeriden yardım almak ya da sosyal mühendislik gibi teknikler uygulamak gerekir. Saldırganlar içerden, dışardan veya her iki şekilde saldırabilir. Saldırıların amacı, akıllı ve değerli sistemlere erişim kazanmak ya da teknik olarak avantaj sağlamak için olabilir. Ayrıca, uygulamanın işleyişini kötüye kullanmaya çalışan saldırganlar göz ardı edilmemelidir. Hassas sistemlerin davranışını etkileyebilirler veya aksatabilirler. Birçok saldırgan, kendine fayda sağlayacak ve doğrudan ya da dolaylı olarak kullanabileceği kişisel bilgiler ve ödeme bilgileri bulmaya çalışmaktadır. Ele geçirilen bilgiler sıklıkla kimlik hırsızlığı, sahte ödemeler veya çeşitli sahtekarlıklar için kullanılmaktadır.	İnternete açık olan tüm uygulamalar.	Sosyal mühendislik saldırılarında kullanılabilecek personel bilgilerini içeren ve değerli fikir ve ticaret sırlarını saklayan uygulamalar.	Organizasyonun devamı ve başarısı için gerekli olan gizli bilgileri, sırları, değerli mal mülkleri, devlet sırlarını saklayan ve İnsan hayatını tehlikeye atabilecek (ulaşım sistemleri parçaları üretimi, kontrol sistemleri gibi) kritik sistemlere sahip uygulamalar.

Endüstri	Tehdit Profili	S1 Önerileri	S2 Önerileri	S3 Önerileri
Sağlık Hizmeti	Birçok saldırgan, kimlik doğrulama ile yapılan para transferi işlemlerini istismar etmek için kimlik bilgisi gibi hassas bilgileri ele geçirmeye çalışmaktadır. Elde edilen bu bilgiler, kimlik hırsızlığı, hileli ödemeler ve birçok dolandırıcılık işlemlerinde kullanılmaktadır.	İnternete açık olan tüm uygulamalar.	Orta düzeyde hassas sağlık bilgisi taşıyan uygulamalar. (Sağlık ödemeleri, ilaç ödemeleri, kişiye özgü bilgiler)	Medikal cihazları, sistemleri ve kayıtları kontrol eden ve insan hayatını etkileyebilecek uygulamalar. Dolandırıcılık için kullanılabilecek birçok hassas bilgi barındıran POS sistemleri. Ayrıca bu uygulamaların yönetim ekranları da dahildir.
Perakende, Gıda, Konaklama	Bu alandaki saldırganlar genel olarak "kap kaç" taktikleri uygulamaktadır. Bunun yanında ödeme sistemleri ve kimlik bilgileri gibi hassas sistemler için de her zaman genel tehditler mevcuttur. Yukarıda bahsedilen tehditlere göre gerçekleşme olasılığı daha az olmasına rağmen organizasyonun fikri mülkiyetlerini çalmak, rekabet etmek için istihbarat elde etmek ya da görüşmelerde iş ortaklarına avantaj sağlamak için gizli ve hassas bilgilere erişmeye çalışan daha kapsamlı saldırılar da görülmektedir.	İnternete açık olan tüm uygulamalar.	Ticari uygulamalar, ürün kataloğu bilgileri, şirket içi kurumsal bilgiler ve sınırlı sayıda kullanıcı bilgisi olan uygulamalar (örneğin, iletişim bilgileri) için uygundur. Küçük ya da orta düzeyde ödeme verileri ya da ödeme işlevleri bulunan uygulamalar.	Dolandırıcılık yapmak için kullanılabilecek büyük miktarda işlem verisi içeren POS sistemler ve bu uygulamaların yönetim ara yüzleri. Kredi kartı bilgisi, anne kızlık soyadı, T.C. kimlik numarası gibi birçok hassas bilgi içeren büyük uygulamalar

Örnek Olaylar

Örnek Olay 1: Bir Güvenlik Test Rehberi Olarak

UTAH Amerika'da özel bir üniversitede kampüs Kırmızı Takımı, uygulama sızma testlerinde OWASP Uygulama Güvenliği Doğrulama Standardı (ASVS) kullanmaktadır. İlk planlamadan, test faaliyetleri için kapsam belirleme toplantılarına, nihai raporun müşterilere iletilmesinden, sızma testi süresi sonuna kadar ASVS kullanılıyor. Ayrıca Kırmızı Takım, ekiplerin ASVS kullanmaları için eğitimler düzenliyor.

Kampüs Kırmızı Takım'ı, üniversitenin bilgi güvenliği stratejilerine göre çeşitli bölümler için ağ ve uygulama güvenliği sızma testleri gerçekleştiriyor. Müşteriler genellikle ilk planlama toplantılarında, test edilecek uygulamanın dışarıya bilgi sızdırmayacak takımlar tarafından test edilmesine izin veriyor. ASVS'ye girişte, test boyunca ASVS standartlarının izleneceği ve final raporunda uygulamaların nasıl ASVS standartlarına göre şekillendirildiği ve birçok bulgunun nasıl giderildiği ilgili taraflara açıklanmaktadır. ASVS, kapsam süresince test için ne kadar zaman ve efor harcanacağının belirlenmesine yardımcı olmaktadır. Kırmızı Takım'ın risk bazlı açıkladığı ASVS'nin ön tanımlı doğrulama seviyelerini kullanmak, müşterilere ve ilgili kişilere (yatırımcılar vs.) yardımcı olmakta ve ekip ilgili uygulama için uygun bir kapsam üzerinde anlaşmaya varmaktadır.

Kırmızı takım, test başlamadan faaliyetleri organize etmek ve iş yükünü bölmek için testlerde ASVS kullanmaktadır. Hangi doğrulama gereksinimlerinin test edildiğini ve hangilerinin hala beklemede olduğunu izleyerek, proje yöneticileri testin nasıl ilerlediğini kolayca görebilirler. Bu, müşterilerle daha iyi iletişim kurulmasına ve proje yöneticilerinin kaynakları daha iyi yönetebilmesini sağlar. Kırmızı Takım, çoğunlukla öğrencilerden oluştuğu için birçok takım üyesinin farklı kurslardan talepleri vardır. Bireysel doğrulama gereksinimlerine veya tüm kategorilere dayalı olarak iyi tanımlanmış görevler yardımıyla ekip üyeleri mevcut testlerin tahmini ne kadar sürede biteceğini tam olarak bilmektedir. Takım üyeleri, sızma testi ile birlikte eş zamanlı olarak sızma testi raporunu da yazabilir.

Kırmızı Takım, her doğrulama gereksiniminin durumunu bildirir, gerekli yerlerde daha ayrıntılı bilgi sağlar ve ASVS'ye uygun olarak nihai raporu düzenler. Bu rapor müşterilere ve paydaşlara verilerek uygulamanın süreç içerisinde güvenliğinin nasıl gerileyip/ilerlediği gösterilmektedir. Ayrıca uygulamanın rapor formatı, ASVS ile hemen hemen aynı olduğundan belirli bir uygulama veya uygulamalar hakkında kolayca bilgi edinilebilir. ASVS, önceki rapor formatına uygun raporlar yazmak için yeni takım üyelerinin eğitimini daha kolay hale getirmektedir.

Son olarak, Kırmızı Takım'ın yetkinliğinin ASVS adaptasyonu sonrasında iyileştiği görülmüştür. Daha öncesinde, proje veya takım lideri tarafından seçilen bir konu üzerinde durulmaktaydı. Bu konular takım üyeleri ve ihtiyaçları doğrultusunda belirlenmekteydi. Bu kriterlere göre eğitim, takım üyelerinin yeteneklerini geliştirme potansiyeline sahipti ama doğrudan çekirdek faaliyetler ile ilgili değildi. Diğer bir deyişle takım, sızma testlerinden önemli ölçüde verim alamamaktaydı. ASVS adaptasyonu sonrasında takım, artık bireysel

doğrulama gereksinimlerinin nasıl test edileceği üzerinde duruyor. Bu durum ekip üyelerinin becerilerinde ve rapor kalitesinde iyileştirme sağlamıştır.

Örnek Olay 2: Güvenli Bir SDLC (Yazılım Geliştirme Hayat Döngüsü) Olarak

Finansal kurumlara büyük veri (big data) analizi sağlamak isteyen bir kuruluşun, gereksinim listesinin en üstünde güvenliğin olması gerekmektedir. Çünkü kuruluş, finansal verilere erişebilir ve bunları işleyebilir pozisyondadır. Bu nedenle kuruluş, çevik ve güvenli geliştirme yaşam döngüsünün (agile secure development lifecycle) temelinde ASVS kullanmayı seçti.

Kuruluş güvenlik gereksinimlerinin en iyi şekilde nasıl uygulanacağı gibi durumlarda kullanım örneği oluşturmak için ASVS'yi kullanmaktadır. Örneğin; oturum açma fonksiyonunun en güvenli şekilde nasıl uygulanacağı konusunda ASVS'den yararlanmaktadır. Kuruluş, ASVS'yi görüldüğünden çok farklı bir şekilde kullanıyor. Geliştirme aşamasında mevcut işe uyan gereksinimleri belirler ve fonksiyonel bir gereklilik ise onları doğrudan ekler ya da mevcut kullanım örneklerine bir sınırlama olarak uygular. Örneğin, TOTP iki faktörlü kimlik doğrulaması ekleniyor ve bununla birlikte parola politikaları ve kaba kuvvet algılama-önleme mekanizması eklenerek güvenliği iki katına çıkarılmış bir web servis düzenleyicisi inşa ediliyor. Gelecek geliştirmelerde ek gereksinimler tecrübe esasına dayalı olarak seçilecektir

Geliştiriciler ASVS'yi bir gözden geçirme listesi olarak kullanırlar. Güvensiz kodların kontrol edilmesini ve geriye dönük olarak; kodun kontrol edildiğini düşünen geliştiricilerle tartışma imkânı sağlanır. Böylece yeni eklenen özelliklerin gerçekten kontrol edildiği garanti altına alınmış olur. ASVS gereksinimleri gelecekte artırılabilir veya azaltılabilir.

Son olarak, geliştiriciler ASVS'yi otomatik güvenli doğrulama biriminin bir parçası olarak kullanıyorlar ve test ortamlarını istismar etme ve "fuzzing" işlemleri için entegre ediyorlar. Amaçları, şelale (waterfall) metodolojisine göre sızma testi sonda yapıldığı için üretime geçmenin riskli yanlarını azaltmaktır. Uygulamayı her defasında "refactor" etmek maliyetli bir iştir. Her geliştirmeden sonra yeni kod değişiklikleri olabileceğinden, tek bir sızma testine güvenmek yeterli değildir. Bundan dolayı sızma testlerinin periyodik hale gelmesi gerekmektedir. Bu düzen sağlanırsa yetenekli bir sızma testi uzmanının uygulamayı test edebilmek için haftalarca zamanı olacaktır.

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.

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

Kaynak koda erişilmeden kapalı kutu (blackbox) şeklinde gerçekleştirilen bir sızma testi L1 (Seviye 1) gereksinimlerini karşılamaktadır. L2 (Seviye 2) aşamasında ise kaynak koda kısmen erişim sağlanabilmeli ve sistem hakkında bilgi sahibi olunması gerekmektedir. L3 (Seviye 3) için ise kaynak kod analizi, tehdit modelleme ve ekstra çalışmalar yapılması gerektiğinden kapsamın tam olarak belirlenebilmesi mümkün değildir.

Detaylı Güvenlik Mimarisi Rehberi Olarak

Uygulama Güvenliği Doğrulama Standardının yaygın kullanımlarından biri, güvenlik mimarlarına (security architects) kaynak olmaktır. İki önemli güvenlik mimari çatısı (framework) olan SABSA ve TOGAF; uygulama güvenliği mimarisinin incelenmesinin tamamlanabilmesi için çokça bilgi eksikliği barındırmaktadır. ASVS, güvenlik mimarlarının veri koruma desenleri 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 Geliştirme Eğitimi 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.

ASVS'yi Kullanan OWASP Projeleri

Security Knowledge Framework

https://www.owasp.org/index.php/OWASP_Security_Knowledge_Framework

"Geliştiricilere güvenli kod yazma konusunda eğitim verin" – SKF, OWASP ASVS kullanan sizi ve takımınızı güvenli kod yazma konusunda geliştirmek için kullanılabilecek tamamen açık kaynak kodlu Python-Flask ile yazılmış bir web uygulamasıdır.

OWASP Zed Attack Proxy

https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project

OWASP Zed Attack Proxy (ZAP), web uygulamalarında zafiyet bulmak için kullanılan, entegrasyonu ve kullanımı oldukça kolay olan bir araçtır. Geniş bir güvenlik deneyimi olan insanlar tarafından tasarlanmıştır. Sızma testinde yeni olan veya uzman olan kişiler için idealdir. ZAP, mevcut otomatik araçlarının yanı sıra güvenlik açıklarını elle (manuel) test edebileceğiniz bir dizi araçta sunmaktadır.

OWASP Cornucopia

https://www.owasp.org/index.php/OWASP_Cornucopia

OWASP Cornucopia, yazılım geliştirme ekiplerinin çevik (agile), konvansiyonel ve resmi geliştirme süreçlerinde güvenlik gereksinimlerini belirlemelerine yardımcı olacak kart oyunu şeklinde bir mekanizmadır. Dil, platform ve teknoloji bağımsızdır. OWASP Secure Coding Practices – Quick Reference Guide (SCP), OWASP ASVS, OWASP Testing Guide and Davi Rook's Principles of Secure Development uygulamalarına göre şekillendirilmiştir.

Detaylı Doğrulama Gereksinimleri

V1. Mimari, Tasarım ve Tehdit Modelleme

V2. Kimlik Doğrulama Gereksinimleri

V3. Oturum Yönetimi Doğrulama Gereksinimleri

V4. Erişim Kontrolleri Doğrulama Gereksinimleri

V5. Kötü Amaçlı Girdi Verilerinin Doğrulama Gereksinimleri

V7. Kriptografi İşlemleri Doğrulama Gereksinimleri

V8. Hata Ayıklama ve Kayıt Doğrulama Gereksinimleri

V9. Veri Koruma Doğrulama Gereksinimleri

V10. İletişim Güvenliği Doğrulama Gereksinimleri

V11. HTTP Güvenliği Doğrulama Gereksinimleri

V13. Zararlı Kontrolleri Doğrulama Gereksinimleri

V15. İş Mantığı Doğrulama Gereksinimleri

V16. Dosya ve Kaynakların Doğrulama Gereksinimleri

V17. Mobil Doğrulama Gereksinimleri

V18. Web Servisleri Doğrulama Gereksinimleri (3.0'da YENİ)

V19. Konfigürasyon Doğrulama Gereksinimleri (3.0'da YENİ)

V1: Mimari, Tasarım ve Tehdit Modelleme

Kontrolün Amacı

Doğrulanmış bir uygulamanın aşağıdaki yüksek seviye gereksinimleri karşıladığından emin olun:

- Seviye 1'de uygulamanın bileşenleri tespit edilmelidir ve uygulamada olmalarının sebebi olmalıdır (ihtiyaç, kullanım vs.).
- Seviye 2'de mimari tanımlanmalıdır ve kod mimariye uygun olmalıdır.
- Seviye 3'te mimari ve tasarım yerinde olmalıdır ve etkin bir biçimde kullanılmalıdır.

Not: Bu bölüm 3.0 sürümünde yeniden tanıtıldı ama ASVS'nin 1.0 sürümünde de aynı mimari kontrolleri bulunmaktadır.

Gereksinimler

#	Tanım	1	2	3	Since
1.1	Tüm uygulama bileşenleri tespit edilmeli ve gerçekten ihtiyaç duyulduğu doğrulanmalıdır.	✓	✓	✓	1.0
1.2	Uygulamanın bir parçası olmayan fakat uygulamanın çalışması için gerekli olan kütüphaneler, modüller, dış kaynaklar tespit edilmeli ve doğrulanmalıdır.		✓	✓	1.0
1.3	Uygulama için üst düzey bir mimarinin tanımlandığı doğrulanmalıdır.		✓	✓	1.0
1.4	Tüm uygulama bileşenlerinin sağladıkları iş veya güvenlik fonksiyonlarının tanımlanmış olduğundan emin olunmalıdır.			✓	1.0
1.5	Uygulamanın parçası olmayan tüm bileşenler doğrulanmalıdır fakat bunlar uygulamanın üzerinde çalıştığı fonksiyonların güvenlik işlevleri açısından tanımlanmış olmalıdır.			✓	1.0
1.6	Hedef uygulama için Spoofing, Tampering, sabotaj, bilgi sızdırma, hizmet kesintisi (DoS) ve yetki yükseltme gibi konuları da içeren bir tehdit modeli üretildiği doğrulanmalıdır.			✓	1.0
1.7	Harici güvenlik hizmeti adı verilen kütüphaneleri de içeren tüm güvenlik kontrollerinin merkezi bir implementasyona sahip olduğu kontrol edilmelidir.		✓	✓	3.0

#	Tanım	1	2	3	Since
1.8	Ağ bölümlendirme, güvenlik duvarı kuralları veya bulut tabanlı güvenlik grupları gibi bileşenlerin birbirinden ayrı tanımladığından emin olunmalıdır.		✓	✓	3.0
1.9	Uygulamanın veri, kontrol ve ara yüz katmanlarının arasında açık bir ayrım olduğundan emin olunmalıdır. Böylece güvenlik üzerine alınan kararlar sistemler üzerine uygulanabilir.		✓	✓	3.0
1.10	İstemci tarafındaki kodlar içerisinde hassas veriler, iş mantığı, gizli anahtarlar ya da diğer özel bilgilerin bulunmadığından emin olunmalıdır.		✓	✓	3.0
1.11	Tüm uygulama bileşenleri, kütüphaneler, modüller, çatılar (frameworks), platform ve işletim sistemlerinin bilinen güvenlik açıklarına karşı korumalı olduğu doğrulanmalıdır.		✓	✓	3.0.1

Referanslar

Daha fazla bilgi için:

- Threat Modeling Cheat Sheet
  https://www.owasp.org/index.php/Application_Security_Architecture_Cheat_Sheet
- Attack Surface Analysis Cheat Sheet:
  https://www.owasp.org/index.php/Attack_Surface_Analysis_Cheat_Sheet

V2: Kimlik Doğrulama Gereksinimleri

Kontrol Amacı

Kimlik doğrulama, bir şeyin (veya birinin) gerçek olduğunun doğrulanmasıdır ya da teyit edilmesidir. Doğrulanmış bir uygulamanın aşağıdaki yüksek seviye gereksinimleri karşıladığından emin olun:

- Bir iletinin göndericisinin dijital kimliği doğrulanmalıdır.
- Yalnızca yetkili kişilerin kimlik doğrulaması yapabilmesi ve kimlik bilgilerinin güvenli bir şekilde taşınması sağlanmalıdır.

Gereksinimler

#	Tanım	1	2	3	Since
2.1	Özel olarak genel paylaşıma açık sayfa ve bileşenler dışındaki tüm gerekli alanlarda kimlik doğrulama mekanizmalarının olduğu doğrulanmalıdır.	✓	✓	✓	1.0
2.2	Kimlik bilgileri form alanında açıkça gösterilmemeli ve kopyalanabilir olmamalıdır.	✓	✓	✓	3.0.1
2.4	Bütün kimlik doğrulama kontrollerinin sunucu tarafında yapıldığı doğrulanmalıdır.	✓	✓	✓	1.0
2.6	Bütün kimlik doğrulama kontrollerinin yanlış giriş yapılması durumda, güvenli bir şekilde başarısız olduğuna ve saldırganların uygulamaya giriş yapmasına izin vermediğine emin olunmalıdır.	✓	✓	✓	1.0
2.7	Parola girdi alanlarının, uzun ve karmaşık parolaların kullanımına izin verdiği ve böylece parola tahmini saldırılarına karşı korunma sağlandığı doğrulanmalıdır.	✓	✓	✓	3.0.1
2.8	Hesaba erişim sağlayabilecek tüm hesap yönetim işlevlerinin (kayıt olma, profil güncelleme, parolamı unuttum, kullanıcı bilgilerini unuttum, devre dışı/kayıp anahtar, yardım paneli ya da interaktif sesli cevap sistemi) en azından birincil kimlik doğrulama mekanizması kadar güvenli olduğu doğrulanmalıdır.	✓	✓	✓	2.0
2.9	Parola değiştirme mekanizmasının an az birincil kimlik doğrulama mekanizması kadar güvenli olduğu doğrulanmalıdır.	✓	✓	✓	1.0

#	Tanım	1	2	3	Since
2.12	Tüm doğrulama girişimlerinin, hassas oturum bilgileri ve parolalar içermeksizin kayıt altına alındığı doğrulanmalıdır. Bu kayıtların içerisindeki isteklerin ve ilgili metadata değerlerinin güvenlik soruşturmalarında kullanılabilecek şekilde tutulduğu doğrulanmalıdır.		✓	✓	3.0.1
2.13	Hesap parolalarının güçlü şifreleme algoritmalarından geçtiğine emin olunmalıdır. Parolanın bir algoritma yardımıyla özeti (hash) alınacaksa “salt” kullanılmalıdır. Bu şifreleme algoritmaları kaba kuvvet (brute force) saldırılarına karşı dayanıklı olmalıdır.		✓	✓	3.0.1
2.16	Kimlik bilgilerinin uygun bir şifreli bağlantı kullanılarak taşındığı ve kullanıcının giriş yapması veya kimlik bilgisi girmesini gerektiren tüm sayfalarda iletişimin şifreli olduğu doğrulanmalıdır.	✓	✓	✓	3.0
2.17	Parolamı unuttum ve diğer parola kurtarma işlevleri mevcut parolayı kullanıcıya hiçbir şekilde göstermemeli ve yeni şifrenin açık metin olarak kullanıcıya gönderilmediği doğrulamalıdır.	✓	✓	✓	2.0
2.18	Giriş, parola sıfırlama veya parolamı/hesabımı unuttum işlevleri aracılığı ile kullanıcı adları listelemenin mümkün olmadığı doğrulanmalıdır.	✓	✓	✓	2.0
2.19	Uygulama çatısı ya da uygulamanın herhangi bir bileşeninde ön tanımlı (default) parola kullanılmadığı doğrulanmalıdır (“admin/password” gibi).	✓	✓	✓	2.0
2.20	Doğrulama ekranları üzerinde yapılan kaba kuvvet ve servis kesintisi saldırıları gibi otomatize saldırıları engellemek adına uygulamaya yapılan isteklerin kısıtlandığı doğrulanmalıdır.	✓	✓	✓	3.0.1
2.21	Uygulamanın dışarıdaki servislere erişim için kullandığı kimlik doğrulama verilerinin şifrelenmiş olduğu ve korunaklı bir yerde saklandığı/depolandığı (kaynak kod içerisinde kesinlikle olmamalı) doğrulanmalıdır.		✓	✓	2.0
2.22	Parolamı unuttum ve diğer kurtarma işlevlerinde, sanal-jeton (soft-token), mobil bildirim veya çevrimdışı doğrulama metotlarının olduğundan emin olunmalıdır. Bir e-postada veya SMS’de rastgele bir parola belirlenip gönderilmesi son çare olmalıdır ve kötü bir yöntem olduğu bilinmelidir.	✓	✓	✓	3.0.1

#	Tanım	1	2	3	Since
2.23	Hesap kilitlenmelerinin “soft” ve “hard” olmak üzere ikiye ayrıldığı ve bu iki durumun birbirini etkilemediği doğrulanmalıdır. Örneğin bir hesap kaba-kuvvet saldırısıyla geçici olarak “soft” kilitlenme durumuna geçtiyse bu “hard” durumunu sıfırlamamalıdır.		✓	✓	3.0
2.24	Gizli soru ve cevap mekanizmasının kullanılması gerekiyorsa; sorular gizlilik ilkelerini ihlal etmemeli ve hesabı kötü niyetli kurtarma işlemlerine karşı koruyabilmelidir. Sorular yeterince zor seçilmelidir.	✓	✓	✓	3.0.1
2.25	Sistemin tanımlanan sayıda (örn. 3 kere) önceden kullanılmış parolaların kullanımına izin vermeyecek şekilde yapılandırılabilir olduğu doğrulanmalıdır.		✓	✓	2.0
2.26	Uygulamanın risk profili dikkate alınarak; uygulamada hassas işlemler yapılırken kimlik doğrulama tekrarı, SMS veya diğer iki adımlı kimlik doğrulama veya işlem imzalamalarının kullanıldığı doğrulanmalıdır.		✓	✓	3.0.1
2.27	Yaygın ve zayıf şifre kullanımının engellenmiş olduğundan emin olunmalıdır.	✓	✓	✓	3.0
2.28	Başarılı ya da başarısız tüm kimlik doğrulama girişimlerinin yanıt süreleri yaklaşık olarak aynı olmalıdır.			✓	3.0
2.29	Kaynak kod içerisinde veya çevrimiçi repository’lerde, API anahtarları veya parolaların saklanmadığından emin olunmalıdır.			✓	3.0
2.31	Kimlik doğrulaması yapılırken, ifşa edilmiş kullanıcı adı + parola ile yetkisiz girişleri engellemek için iki adımlı doğrulama veya güçlü bir kullanıcı doğrulama mekanizması uygulandığından emin olunmalıdır.		✓	✓	3.0
2.32	Güvenilmeyen kişilerin uygulamaların yönetici ara yüzüne ulaşamadıklarından emin olunmalıdır.	✓	✓	✓	3.0
2.33	Riske dayalı bir politikayla yasaklanmadıkça, tarayıcı otomatik tamamlamalarına ve parola yöneticileriyle entegre olmaya izin verilir.	✓	✓	✓	3.0.1

Referanslar

Daha fazla bilgi için:

- OWASP Testing Guide 4.0: Testing for Authentication
  https://www.owasp.org/index.php/Testing_for_authentication
- Password storage cheat sheet
  https://www.owasp.org/index.php/Password_Storage_Cheat_Sheet
- Forgot password cheat sheet
  https://www.owasp.org/index.php/Forgot_Password_Cheat_Sheet
- Choosing and Using Security Questions at
  https://www.owasp.org/index.php/Choosing_and_Using_Security_Questions_Cheat_Sheet

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.

Gereksinimler

#	Tanım	1	2	3	Since
3.1	Özel geliştirilen bir oturum yöneticisinin olmadığını veya kullanılmışsa tüm yaygın oturum yönetimi açıklıklarına karşı güvenli olduğu doğrulanmalıdır.	✓	✓	✓	1.0
3.2	Kullanıcı oturumu sonlandırıldığında oturumun geçersiz kılındığı doğrulanmalıdır.	✓	✓	✓	1.0
3.3	Belirli bir süre boyunca aktif olmayan oturumların, zaman aşımına uğrayarak sonlandırıldığı doğrulanmalıdır.	✓	✓	✓	1.0
3.4	Yönetim seviyesinde, kullanıcının aktif durumundan bağımsız olarak ayarlanabilir bir zaman aşımı süresi olmalıdır ve süre aşıldığında oturumların sonlandırıldığı doğrulanmalıdır.		✓	✓	1.0
3.5	Yalnızca kimlik doğrulaması ile erişilebilen sayfaların tümünün "oturumu sonlandır" linkine sahip olduğu doğrulanmalıdır.	✓	✓	✓	1.0
3.6	Oturum anahtarının çerez başlıkları dışında hiçbir yerde ifşa edilmediği, özellikle de URL üzerinde, hata mesajları ve tutulan kayıtlar içinde tutulmadığı/geçmediği doğrulanmalıdır. Bu kapsamda, uygulamanın oturum çerezlerinin, URL içinde tekrar yazılmasını da desteklemediği ayrıca doğrulanmalıdır.	✓	✓	✓	1.0

#	Tanım	1	2	3	Since
3.7	Başarılı bir kullanıcı girişi sonrası oturum anahtarının değiştiği doğrulanmalıdır.	✓	✓	✓	1.0
3.10	Uygulamanın, yalnızca uygulama çatısı tarafından üretilen oturum anahtarlarını geçerli olarak kabul ettiği doğrulanmalıdır.		✓	✓	1.0
3.11	Kimlik doğrulamada kullanılan oturum anahtarlarının, tahmin saldırılarından korunmaya yetecek seviyede uzun ve karmaşık olduğu doğrulanmalıdır.	✓	✓	✓	1.0
3.12	Kimlik doğrulamada kullanılan oturum anahtarlarını içeren çerezler, oturum anahtarını sitenin sadece belirli bir bölümüne erişim sağlayacak şekilde kısıtlayan dizin değerleri içermelidir. Oturum anahtarlarının "HttpOnly" ve "Secure" özelliklerine sahip olduğuna emin olunmalıdır.	✓	✓	✓	3.0
3.16	Uygulamanın eş zamanlı oturum sayısını kısıtladığından emin olunmalıdır.	✓	✓	✓	3.0
3.17	Her kullanıcı aktif oturumlarını görüntüleyebilmeli ve istenildiği takdirde aktif oturumlarını sonlandırabilmelidir.	✓	✓	✓	3.0
3.18	Başarılı şifre değiştirme işleminden sonra kullanıcıya "Aktif diğer tüm oturumlar kapatılsın mı?" şeklinde uyarı çıkarılmalıdır.	✓	✓	✓	3.0

Referanslar

Daha fazla bilgi için:

- OWASP Testing Guide 4.0: Session Management Testing
  https://www.owasp.org/index.php/Testing_for_Session_Management
- OWASP Session Management Cheat Sheet:
  https://www.owasp.org/index.php/Session_Management_Cheat_Sheet

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.

Gereksinimler

#	Tanım	1	2	3	Since
4.1	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.	✓	✓	✓	1.0
4.4	Direkt nesne erişimlerinin yetki kontrolü ile korunduğu doğrulanmalıdır, öyle ki her kullanıcı sadece yetkisi olan nesnelere erişebilmelidir. Örneğin; bir kullanıcı başka bir kullanıcının hesabını ekstra parametre ekleyerek görmemeli veya değiştirmemelidir.	✓	✓	✓	1.0
4.5	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.	✓	✓	✓	1.0
4.8	Erişim kontrolleri hatalarının güvenli bir şekilde ele alındığı doğrulanmalıdır.	✓	✓	✓	1.0
4.9	Belirli bir kullanıcı rolü için sunum katmanı (presentation layer) tarafından belirtilen erişim kontrollerinin, sunucu tarafında da uygulandığı doğrulanmalıdır.	✓	✓	✓	1.0
4.10	Özel olarak yetkilendirilmediği sürece, erişim kontrolleri tarafından kullanılan tüm kullanıcı verileri ve politika bilgilerinin son kullanıcı tarafından değiştirilemediği doğrulanmalıdır.		✓	✓	1.0

#	Tanım	1	2	3	Since
4.11	Korunan her türlü kaynağa, erişimin kontrolü için merkezi bir mekanizmanın (dış yetkilendirme servislerini çağıran kütüphaneler dahil) olduğu doğrulanmalıdır.			✓	1.0
4.12	Tüm erişim kontrolü hareketlerinin veya tüm yetkisiz erişim hareketlerinin kayıtlarının tutulduğu doğrulanmalıdır.		✓	✓	2.0
4.13	Uygulamanın, yeterince karmaşıklığa sahip ve tahmin edilemez bir anti-CSRF (siteler arası istek sahteciliği) anahtarı ürettiğinden veya benzeri koruma mekanizmasına sahip olduğundan emin olunmalıdır.	✓	✓	✓	2.0
4.14	Sistemin, güvenli fonksiyonlarına, kaynaklarına ya da verilerine sürekli erişimin korumalı olduğu doğrulanmalıdır. Örneğin, bir kaynak yöneticisi yardımı ile bir saat süresince kaynağın düzenlenme sayısına sınırlama getirilmesi ya da tek bir kullanıcı tarafından tüm veri tabanına erişebilir hale gelmesi sistem tarafından engellenmelidir.		✓	✓	2.0
4.15	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.		✓	✓	3.0
4.16	Parametre değiştirme veya aldatma (tampering) yoluyla yetkisiz verilere erişimi engellemek için uygulamanın içeriğe göre yetkilendirmeyi doğru bir şekilde yaptığından emin olunmalıdır.	✓	✓	✓	3.0

Referanslar

Daha fazla bilgi için:

- OWASP Testing Guide 4.0: Authorization

  https://www.owasp.org/index.php/Testing_for_Authorization
- OWASP Cheat Sheet: Access Control

  https://www.owasp.org/index.php/Access_Control_Cheat_Sheet

V5: Kötü Amaçlı Girdi Verilerinin Doğrulama Gereksinimleri

Kontrol Hedefleri

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 çıkmasını sağlamaktadır.

Doğrulanmış bir uygulamanın aşağıdaki yüksek seviye gereksinimleri karşıladığından emin olun:

- Tüm girdilerin doğrulanmış (filtrelenmiş) olması ve amacına uygun oldukları doğrulanmalıdır.
- Harici bir kaynaktan veya istemciden gelen bir veri asla güvenilir olarak kabul edilmemelidir.

Gereksinimler

#	Tanım	1	2	3	Since
5.1	Canlı ortamın 'bellek taşması (buffer overflow)' na olanak tanımadığı veya güvenlik önlemlerinin bellek taşmasına karşı koruma sağladığı doğrulanmalıdır.	✓	✓	✓	1.0
5.3	Sunucu tarafında başarısızlıkla sonuçlanmış girdi kontrollerinin isteğin reddi ile sonuçlanıp kayıt altına alındığına (loglandığına) emin olunmalıdır.	✓	✓	✓	1.0
5.5	Girdi verisi kontrollerinin sunucu tarafında yapıldığı doğrulanmalıdır.	✓	✓	✓	1.0
5.6	Uygulama üzerinde, uygulamanın işleme aldığı her tip veri için tek noktadan veri doğrulama kontrollerinin uygulandığı doğrulanmalıdır.			✓	1.0
5.10	Bütün SQL sorgularının, HQL, OSQL, NOSQL ve saklı yordam'ların (stored procedure) "prepared statement" veya "query parameterization" ile yapıldığı doğrulanmalıdır. Bu sayede SQL enjeksiyonu saldırısının önüne geçilecektir.	✓	✓	✓	2.0
5.11	Uygulamanın 'LDAP Enjeksiyonu (LDAP Injection)'na olanak tanımadığı veya güvenlik önlemlerinin LDAP Enjeksiyonu'na karşı koruma sağladığı doğrulanmalıdır.	✓	✓	✓	2.0

#	Tanım	1	2	3	Since
5.12	Uygulamanın “İşletim Sistemi Komut Enjeksiyonu” (OS Command Injection) zafiyetine olanak tanımadığı veya güvenlik önlemlerinin “İşletim Sistemi Komut Enjeksiyonu”na karşı koruma sağladığı doğrulanmalıdır.	✓	✓	✓	2.0
5.13	Uygulamanın Uzaktan Dosya Dahil Etme(RFI) veya Yerel Dosya Dahil Etme(LFI) saldırılarına olanak tanımadığı veya güvenlik önlemlerinin bu saldırılara karşı koruma sağladığı doğrulanmalıdır.	✓	✓	✓	3.0
5.14	Uygulamanın XPath sorgu değiştirme, XXE (XML External Entity) ve XML Enjeksiyonu saldırıları gibi bilinen XML saldırılarına karşı önlem aldığı doğrulanmalıdır.	✓	✓	✓	2.0
5.15	Yansıtılmış (reflected), yerleşik (stored) ve DOM tabanlı siteler arası betik çalıştırma (XSS) zafiyetlerini önlemek için sunucu tarafında ve web istemci kodlarının içerisinde yer alan bütün girdilerin doğru şekilde “encode” edildiği doğrulanmalıdır.	✓	✓	✓	2.0
5.16	Eğer uygulama otomatik parametre atamasına izin veriyorsa (otomatik değişken bağlama -automatic variable binding-) hassas alanların (rol, parola, hesap özeti gibi) zararlı parametrelerden korunduğu doğrulanmalıdır.		✓	✓	2.0
5.17	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.		✓	✓	2.0
5.18	Sunucu taraflı doğrulamaya ek olarak istemci taraflı doğrulama yapıldığından emin olunmalıdır.		✓	✓	3.0
5.19	Bütün verilerin doğrulandığına emin olunmalıdır. Sadece HTML form alanları değil, REST çağırımları, sorgu parametreleri, HTTP başlıkları, çerezler, batch dosyaları, RSS beslemeleri ve benzeri alanlar gibi girdi içeren alanlar da doğrulanmalıdır. Bunun için pozitif doğrulama (white listing), kötü olduğu bilinen kelimelerin elenmesi (gray listing) veya kötü kabul edilen girdilerin elenmesi (black listing) yöntemleri kullanılabilir.		✓	✓	3.0
5.20	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.		✓	✓	3.0

#	Tanım	1	2	3	Since
5.21	Belirli bir formata uymayan verilerin, izin verilen karakterler, veri uzunluğu ye potansiyel zararlı karakterler gibi (örneğin, 𠀋 veya O'Hara gibi isimlerde bulunan karakterler gibi) her veri için uygulanabilen güvenlik önlemlerinden geçtiği doğrulanmalıdır.		✓	✓	3.0
5.22	WYSIWYG veya benzeri editörlerden gelen güvensiz HTML kodlarının bir HTML temizleyicisi ile uygun şekilde filtrelendiğinden (sanitize) ve doğru bir şekilde "encode" edildiğinden emin olunmalıdır.	✓	✓	✓	3.0
5.23	Otomatik-kaçırma şablonu (auto-escaping template) teknolojisinde "UI escaping"aktif olmalıdır, pasif durumda ise de HTML ayıklamasının (sanitization) aktif olduğundan emin olunmalıdır.		✓	✓	3.0
5.24	Bir DOM içeriğinden diğerine veri aktarılırken ".innerText" ve ".val" gibi güvenli JavaScript metotlarının kullanıldığından emin olunmalıdır.		✓	✓	3.0
5.25	İnternet tarayıcısında JSON ayrıştırılmadan (parsing) önce doğrulanmalıdır. İstemci üzerinde JSON ayrıştırılırken (parsing) eval() kullanılmadığı doğrulanmalıdır.		✓	✓	3.0
5.26	Oturum sonlandıktan sonra kimlik verilerinin istemci depolama alanından (client storage) silindiği doğrulanmalıdır.		✓	✓	3.0

Referanslar

Daha fazla bilgi için:

- OWASP Testing Guide 4.0: Input Validation Testing
  https://www.owasp.org/index.php/Testing_for_Input_Validation
- OWASP Cheat Sheet: Input Validation
  https://www.owasp.org/index.php/Input_Validation_Cheat_Sheet
- OWASP Testing Guide 4.0: Testing for HTTP Parameter Pollution
  https://www.owasp.org/index.php/Testing_for_HTTP_Parameter_pollution_%28OTG-INPVAL-004%29
- OWASP LDAP Injection Cheat Sheet
  https://www.owasp.org/index.php/LDAP_Injection_Prevention_Cheat_Sheet

- OWASP Testing Guide 4.0: Client Side Testing
  https://www.owasp.org/index.php/Client_Side_Testing
- OWASP Cross Site Scripting Prevention Cheat Sheet
  https://www.owasp.org/index.php/XSS_%28Cross_Site_Scripting%29_Prevention_Cheat_Sheet
- OWASP Java Encoding Project
  https://www.owasp.org/index.php/OWASP_Java_Encoder_Project

Otomatik-kaçış (auto-escaping) hakkında daha fazla bilg için:

- Reducing XSS by way of Automatic Context-Aware Escaping in Template Systems
  http://googleonlinesecurity.blogspot.com/2009/03/reducing-xss-by-way-of-automatic.html
- AngularJS Strict Contextual Escaping https://docs.angularjs.org/api/ng/service/$sce
- https://cwe.mitre.org/data/definitions/915.html

V6: Çıktı kodlama (encoding) / kaçış (escaping)

Bu bölüm Uygulama Güvenliği Doğrulama Standardı v2.0'da V5 bölümüne dahil edilmiştir. ASVS 5.16 gereksinimi, XSS saldırılarına karşı önlem almak için bağlamsal çıktı kodlamayı (contextual output encoding) ele alıyor.

V7: Kriptografi İşlemleri Doğrulama Gereksinimleri

Kontrol Hedefleri

Doğrulanmış 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.

Gereksinimler

#	Tanım	1	2	3	Since
7.2	Tüm şifreleme modüllerinin güvenli bir şekilde hata durumuna geçtiği doğrulanmalıdır. Hatalar 'oracle padding'i etkinleştirmeyecek şekilde işlenmelidir.	✓	✓	✓	1.0
7.6	Tüm rastgele üretilen sayılar, dosya isimleri, global eşsiz kimlikleyiciler (GUID) ve karakter dizilerinin saldırgan tarafından tahmin edilemez olmasını sağlamak için onaylanmış bir şifreleme modülü ile rastgele sayı üretildiği doğrulanmalıdır.		✓	✓	1.0
7.7	Uygulama tarafından kullanılan şifreleme modüllerinin FIPS 140-2 veya eşiti bir standarda uygunluğu doğrulanmalıdır.	✓	✓	✓	1.0
7.8	Şifreleme modüllerinin yayınlanmış güvenlik politikaları çerçevesinde işlevselliklerini yerine getirdikleri doğrulanmalıdır.			✓	1.0
7.9	Şifreleme anahtarlarının nasıl yönetileceğine dair açık / belirgin bir politika olduğu doğrulanmalıdır. (Mesela üretilmesi, dağıtılması, iptali, süresinin dolması v.b.) Bu politikanın da doğru şekilde uygulandığı doğrulanmalıdır.		✓	✓	1.0

#	Tanım	1	2	3	Since
7.11	Tüm kriptografik servis kullanıcılarının, anahtar materyallerine doğrudan erişimi olmadığı doğrulanmalıdır. Şifreleme süreçlerinin, şifreleme anahtarları da (master keys) dahil olmak üzere izole edildiği doğrulanmalıdır. Donanımsal anahtarların (HSM) kullanımı tavsiye edilmektedir.			✓	3.0.1
7.12	Kullanıcıyı tanımlamak için kullanılan bilgiler şifrelenerek depolanmalı ve iletişimin korumalı kanallar üzerinden yapıldığına emin olunmalıdır.		✓	✓	3.0
7.13	Bellek okuma (memory dumping) saldırılarında bellek üzerindeki kullanılmayan hassas parolaların, anahtarların ve gizli bilgilerin (secrets) açığa çıkmaması için okunamayacak şekilde değiştirildiğine (üzerine sıfır yazıldığından) emin olunmalıdır.		✓	✓	3.0.1
7.14	Tüm parola ve anahtarların değiştirilebilir olduğu ve bunların kurulum esnasında üretilip değiştirildiği doğrulanmalıdır.		✓	✓	3.0
7.15	Uygulamanın ağır yük altında bile yeterli karmaşıklıkta sayılar üretebildiğinden veya bu gibi durumlarda uygulamanın talebi zarifçe geri çevirdiğinden emin olunmalıdır.			✓	3.0

Referanslar

Daha fazla bilg için:

- OWASP Testing Guide 4.0: Testing for weak Cryptography
  https://www.owasp.org/index.php/Testing_for_weak_Cryptography
- OWASP Cheat Sheet: Cryptographic Storage
  https://www.owasp.org/index.php/Cryptographic_Storage_Cheat_Sheet

V8: Hata Ayıklama ve Kayıt Doğrulama Gereksinimleri

Kontrol Hedefleri

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 kayıt edilmemelidir.
- 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.

Gereksinimler

#	Tanım	1	2	3	Since
8.1	Uygulamanın, saldırganın işini kolaylaştıracak hata mesajları veya yığın yapıları (stack) gibi hassas verileri (oturum bilgisi ve kişisel bilgiler de dahil olmak üzere) açığa çıkarmadığı doğrulanmalıdır.	✓	✓	✓	1.0
8.2	Güvenlik kontrollerindeki hata işleme mantığının "varsayılan/fabrika tanımlı" (default) erişimleri reddettiği doğrulanmalıdır.		✓	✓	1.0
8.3	Güvenlik kayıt kontrol mekanizmalarının, güvenlik ile ilgili başarılı ve başarılı olmayan olayları içeren kayıt kabiliyeti sunduğu doğrulanmalıdır.		✓	✓	1.0
8.4	Sistem günlüklerinin (loglar) olayın gerçekleştiği anda, bilgileri ve zaman çizelgesini ayrıntılı tuttuğunun doğrulaması yapılmalıdır. Bu günlükler, ayrıntılı incelemelerde yeterince bilgi sağlamalıdır.		✓	✓	1.0
8.5	Tutulan kayıtları okuma veya işleme esnasında kullanılan programın, kayıtlar (loglar) içerisindeki verileri kod olarak çalıştırmadığı doğrulanmalıdır.		✓	✓	1.0

#	Tanım	1	2	3	Since
8.6	Güvenlik kayıtlarının izinsiz erişimlere ve değişikliklere karşı korunduğu doğrulanmalıdır.		✓	✓	1.0
8.7	Uygulamanın hassas verileri sistem kayıtlarında (log) tutmadığı doğrulanmalıdır. Bu veriler arasında; yerel gizlilik kanun ve yönetmeliklerinde tanımlanan hassas veriler, risk analizi ile tanımlanan organizasyonel hassas veriler, saldırgana yardımcı olabilecek hassas doğrulama verileri (oturum tanımlayıcılar, parolalar, hash'ler veya API jetonları(token) gibi...) yer almaktadır.		✓	✓	3.0
8.8	Kayıt enjeksiyonunu engellemek adına kayıtlara geçilen verilerde bulunan özel semboller (non-printable semboller gibi) ve alan ayraçları doğru şekilde "encode" edilmelidir.			✓	2.0
8.9	Güvenilir ve güvenilir olmayan kaynaklardan elde edilen kayıtların, kayıt girdilerinde açık bir şekilde birbirinden ayırt edilebilir olduğu doğrulanmalıdır.			✓	2.0
8.10	Bir denetim günlüğünün veya benzerinin, kritik işlemleri reddetmediğine emin olunmalıdır.	✓	✓	✓	3.0
8.11	Güvenlik kayıtlarının yetkisiz değişikliklere karşı korunduğu doğrulanmalıdır.			✓	3.0
8.12	Kayıtların, uygulamadan farklı bir bölümde uygulama ile rotasyonlu çalışacak şekilde tutulduğu doğrulanmalıdır.			✓	3.0
8.13	Kayıtların, doğru bir zamana sahip olduğunun doğrulanması için zaman kaynaklarıyla senkronize edilmelidir.	✓	✓	✓	3.0.1

Referanslar

Daha fazla bilgi için:

- OWASP Testing Guide 4.0 content: Testing for Error Handling https://www.owasp.org/index.php/Testing_for_Error_Handling

V9: Veri Koruma Doğrulama Gereksinimleri

Kontrol Hedefleri

Veri koruması için üç önemli unsur vardır: Gizlilik, Bütünlük ve Erişilebilirlik (İng. 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.

Gereksinimler

#	Tanım	1	2	3	Since
9.1	Son kullanıcı tarafında hassas veri içeren tüm formların önbellekte ya da otomatik tamamlama özellikleri ile saklanmasının önlendiği doğrulanmalıdır.	✓	✓	✓	1.0
9.2	Uygulama tarafından işlenen tüm hassas verilerin bir listesinin tanımlı olduğu ve ayrıca bu verilere nasıl erişileceği ve verilerin nasıl şifrelenip gönderileceği/saklanacağı hakkında belirli bir politikanın var olduğu doğrulanmalıdır. Bu politikanın uygulanmasının dayatıldığı da doğrulanmalıdır.			✓	1.0
9.3	Tüm hassas verilerin sunucuya HTTP mesaj gövdesinde veya başlıklarında gönderildiği doğrulanmalıdır. (Hiçbir hassas veri URL parametresi olarak gönderilmemelidir.)	✓	✓	✓	1.0

#	Tanım	1	2	3	Since

9.4	Uygulamanın, risk durumuna göre aşağıdaki gibi uygun önbellek önleme (anti-caching) başlıklarını gönderdiği doğrulanmalıdır:
	`Expires: Tue, 03 Jul 2001 06:00:00 GMT`
	`Last-Modified: {now} GMT`
	`Cache-Control: no-store, no-cache, must-revalidate, max-age=0`
	`Cache-Control: post-check=0, pre-check=0`
	`Pragma: no-cache`
		✓	✓	✓	1.0

9.5	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.		✓	✓	1.0
9.6	Her tipteki hassas verinin, saklama süresi dolduğu zaman uygulamadan silinmesini sağlamak üzere bir metodun mevcut olduğu doğrulanmalıdır.			✓	1.0
9.7	Uygulamanın, gizli alanlar, Ajax değişkenleri, çerezler ve başlık değerleri gibi güvenilmeyen sistemlere gönderilecek olan değişken sayısını minimum seviyede tuttuğu doğrulanmalıdır.		✓	✓	2.0
9.8	Uygulamanın, olağandışı istek sayısı veya ekran dönüşümü (screen scraping), gibi durumları tespit edebildiği ve gerekli yerlere alarm mesajları gönderebildiği doğrulanmalıdır.			✓	2.0
9.9	Kullanıcı tarafında HTML5 yerel depolama, oturum depolama, indexedDB, düzenli çerezler veya Flash çerezler gibi hassas veri içeren depolamaların yapılmadığından emin olunmalıdır.	✓	✓	✓	3.0.1
9.10	Eğer veriler, veri koruma kanunları altında toplanıyorsa (kişisel veri koruma kanunu kapsamına giren veriler) hassas verilerin kayıt altına alınması veya gerekli bilgilerin kaydedilmesi gerekmektedir.		✓	✓	3.0
9.11	Hassas verilere ihtiyaç duyulmadığında hafızadan hızlıca silinmesi gerekmektedir. Bu bellek okuma (dumping) saldırılarının önüne geçmektedir.		✓	✓	3.0.1

Referanslar

Daha fazla bilgi için:

- User Privacy Protection Cheat Sheet: https://www.owasp.org/index.php/User_Privacy_Protection_Cheat_Sheet

V10: İletişim Güvenliği Doğrulama Gereksinimleri

Kontrol Hedefleri

Doğrulanmış bir uygulamanın aşağıdaki yüksek seviye gereksinimleri karşıladığından emin olun:

- Hassas verilerin iletildiği yerlerde TLS kullanılır.
- Her zaman güçlü algoritmalar ve şifreleme teknikleri kullanılır.

Gereksinimler

#	Tanım	1	2	3	Since
10.1	Güvenilir bir sertifika otoritesinden, tüm Taşıma Katmanı Güvenliği (TLS) sunucu sertifikalarına bir yol (path) sağlanabildiği ve tüm sertifikaların geçerli olduğu doğrulanmalıdır.	✓	✓	✓	1.0
10.3	Yetkilendirilmiş (authenticated) ya da hassas veriler taşıyan tüm bağlantıların –dış ve iç (backend) bağlantıları da kapsamak üzere- Taşıma Katmanı Güvenliği (TLS)'ni kullandığı doğrulanmalıdır. Taşıma Katmanı Güvenliği (TLS) bağlantılarının, başarısız oldukları anda güvensiz bir HTTP bağlantısına dönüşmedikleri doğrulanmalıdır.	✓	✓	✓	3.0
10.4	Başarısız Taşıma Katmanı Güvenliği (TLS) bağlantılarının kayıt altına alındığı doğrulanmalıdır.			✓	1.0
10.5	Tüm istemci sertifikaları için; sertifika sağlayıcılarının önceden yapılandırılmış güvenilir referans noktaları (trust anchors) ve iptal(revocation) bilgileri kullanılarak oluşturulduğu/ doğrulandığı doğrulanmalıdır.			✓	1.0
10.6	Dış sistemlere yapılan ve hassas veri/bilgiler ya da işlevler içeren her bağlantının yetkilendirildiği doğrulanmalıdır.		✓	✓	1.0
10.8	Onaylanmış operasyon modunda (approved operation mode) çalışan uygulamaların tek bir standart Taşıma Katmanı Güvenliği gerçeklemesi kullandığı doğrulanmalıdır.			✓	1.0
10.10	Üretim ortamı ve yedekleme anahtarları ile TLS sertifikası için 'public key pinning'(HPKP) yapıldığına emin olunmalıdır.		✓	✓	3.0.1
10.11	HTTP Strict Transport Security başlığının 'Strict-Transport-Security: max-age=15724800; includeSubdomains ' şeklinde bütün isteklerde ve bütün alt alanlarda(subdomain) yer aldığı doğrulanmalıdır.	✓	✓	✓	3.0

#	Tanım	1	2	3	Since
10.12	Üretim ortamı web sitesi URL'inin web tarayıcısı sağlayıcıları tarafından sağlanmakta olan 'Strict Transport Security' domainleri listesinde yer alması için bu sağlayıcılara gönderildiğine emin olunmalıdır. Lütfen aşağıda referanslara bakınız.			✓	3.0
10.13	Pasif saldırganların trafik kaydetmesini önlemek için ileri seviye şifreleme tekniklerinin kullanıldığından emin olunmalıdır.	✓	✓	✓	3.0
10.14	Online Certificate Status Protocol (OCSP) Stapling gibi bir yöntem ile sertifika iptali yapılabildiği doğrulanmalıdır.	✓	✓	✓	3.0
10.15	Seçilen sertifika otoritesinin bütün sertifika hiyerarşisinde (kök ve aracı sertifikalar dahil) sadece güçlü algoritmalar, protokoller ve şifreleme kullandığına emin olunmalıdır.	✓	✓	✓	3.0
10.16	TLS ayarlarının güncel en iyi pratikler ile yapıldığı doğrulanmalıdır. Konfigurasyonlar, şifreleme metodları ve algoritmalarının güncel olduğu doğrulanmalıdır.	✓	✓	✓	3.0

Referanslar

Daha fazla bilgi için:

- **OWASP – TLS Cheat Sheet.** https://www.owasp.org/index.php/Transport_Layer_Protection_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.
- **Sertifika Sabitleme (Certificate Pinning)**. Daha detaylı bilgi için lütfen bu dokümanı inceleyin; https://tools.ietf.org/html/rfc7469.

Canlı ortamlar ve yedekleme anahtarları için sertifika sabitlemesinin ardındaki mantık iş sürekliliğidir. Detaylar; https://noncombatant.org/2015/05/01/about-http-public-key-pinning/

- OWASP Certificate Pinning Cheat Sheet https://www.owasp.org/index.php/Pinning_Cheat_Sheet
- OWASP Certificate and Public Key Pinning https://www.owasp.org/index.php/Certificate_and_Public_Key_Pinning
- Time of first use (TOFU) Pinning https://developer.mozilla.org/en/docs/Web/Security/Public_Key_Pinning
- **Pre-loading HTTP Strict Transport Security** https://www.chromium.org/hsts

V11: HTTP Güvenliği Doğrulama Gereksinimleri

Kontrol Hedefleri

Doğrulanmış bir uygulamanın aşağıdaki yüksek seviye gereksinimleri karşıladığından emin olun:

- Uygulama sunucusu varsayılan yapılandırmalar dışında uygun bir şekilde sıkılaştırılır.
- HTTP yanıtları, başlıklarda güvenli karakterler içerir.

Gereksinimler

#	Tanım	1	2	3	Since
11.1	Uygulamanın GET, POST gibi sadece izin verilen bir dizi HTTP metodlarını kabul ettiği ve bu dizi dışındaki tüm metodları (TRACE, PUT, DELETE) özellikle blokladığı doğrulanmalıdır.	✓	✓	✓	1.0
11.2	Tüm HTTP cevaplarının UTF-8 vb. gibi güvenli bir karakter setini belirten içerik türü başlığı (content type header) içerdiği doğrulanmalıdır.	✓	✓	✓	1.0
11.3	Güvenilir proxy veya SSO cihazlarından atanan HTTP başlıkları uygulama tarafından doğrulanmalıdır.		✓	✓	2.0
11.4	Üçüncü parti X-Frame tarafından görüntülenmesi gerekmeyen içerikler için uygun bir X-FRAME-OPTIONS başlığının bulunduğundan emin olunmalıdır.		✓	✓	3.0.1
11.5	HTTP başlıklarının veya HTTP yanıtının herhangi bir bölümünün detaylı bir sürüm bilgisi içermediği doğrulanmalıdır.	✓	✓	✓	2.0
11.6	Tüm API yanıtlarının içerisinde X-Content-Type-Options: nosniff ve Content-Disposition: attachment; filename="api.json" veya uygun benzeri içerik tipi kullanıldığından emin olunmalıdır.	✓	✓	✓	3.0
11.7	DOM, XSS, JSON ve JavaScript enjeksiyonu gibi zafiyetlerin etkisini hafifletmek adına Content Security Policy (CSPv2) ilkesinin bulunduğuna emin olunmalıdır.	✓	✓	✓	3.0.1

#	Tanım	1	2	3	Since
11.8	X-XSS-Protection:1; mode=block başlığının HTTP içerisinde var olduğu doğrulanmalıdır. Bu başlık tarayıcının XSS filtrelerini aktifleştirmesini sağlar.	✓	✓	✓	3.0

Referanslar

Daha fazla bilgi için:

- OWASP Testing Guide 4.0: Testing for HTTP Verb Tampering https://www.owasp.org/index.php/Testing_for_HTTP_Verb_Tampering_%28OTG-INPVAL-003%29
- API yanıtlarına “Content-Disposition” HTTP başlığının eklenmesi, istemci ve sunucu arasındaki MIME türünün yanlış yapılandırılmasına dayalı birçok saldırıyı önlemeye yardımcı olur. Bunun yanı sıra “filename” seçeneği “Reflected File Download” saldırılarını önlemeye yardımcı olur.
  https://www.blackhat.com/docs/eu-14/materials/eu-14-Hafif-Reflected-File-Download-A-New-Web-Attack-Vector.pdf
  https://www.webguvenligi.org/docs/ReflectedFileDownload-TR_v1.pdf
- https://www.owasp.org/index.php?title=Content_Security_Policy_Cheat_Sheet&setlang=en

V12: Güvenlik Yapılandırması Doğrulama Gereksinimleri

Bu bölüm Uygulama Güvenliği Doğrulama Standardı v2.0'da V11 bölümüne dahil edilmiştir.

V13: Zararlı Kontrolleri Doğrulama Gereksinimleri

Kontrol Hedefleri

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.

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.

Gereksinimler

#	Tanım	1	2	3	Since
13.1	Tüm zararlı aktivitelerin yeterli ve etkili bir şekilde güvenli alan(sandbox) içerisine alındığı, arındırıldığı veya izole edildiği doğrulanmalıdır.			✓	2.0
13.2	Uygulama kaynak kodunda ve çok sayıda üçüncü parti kütüphanelerde; kimlik doğrulama, erişim kontrolü, giriş kontrolü veya önemli iş akışında arka kapılar, gizli özellikler(easter eggs) ve mantık hataları içermediği doğrulanmalıdır.			✓	3.0.1

Referanslar

Daha fazla bilgi için:

- http://www.dwheeler.com/essays/apple-goto-fail.html

V14: İç Güvenlik Doğrulama Gereksinimleri

Bu bölüm, Uygulama Güvenliği Doğrulama Standardı v2.0'da V13 bölümüne dahil edilmiştir.

V15: İş Mantığı Doğrulama Gereksinimleri

Kontrol Hedefleri

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.

Gereksinimler

#	Tanım	1	2	3	Since
15.1	Uygulamanın iş mantığı akış adımlarını sıralı ve her adımı gerçek -insan benzeri- bir zamanda işlediği, kullanılmayan ve başka bir kullanıcıya ait adımları işlemediği veya çok hızlı gönderilen işlemleri işlemediği doğrulanmalıdır.		✓	✓	2.0
15.2	Uygulamanın işlem limitlerine, dayatmalara sahip olduğu ve ayrıca bu dayatmaların ayarlanabilir bir alarm ve otomatize saldırılara otomatik cevap verebilme gibi özelliklere sahip olduğu doğrulanmalıdır		✓	✓	2.0

Referanslar

Daha fazla bilgi için:

- OWASP Testing Guide 4.0: Business Logic Testing https://www.owasp.org/index.php/Testing_for_business_logic
- OWASP Cheat Sheet: https://www.owasp.org/index.php/Business_Logic_Security_Cheat_Sheet

V16: Dosya ve Kaynakların Doğrulama Gereksinimleri

Kontrol Hedefleri

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.

Gereksinimler

#	Tanım	1	2	3	Since
16.1	URL yönlendirmeleri ve iletimlerinin sadece onaylanmış sitelere yönlendirildiği, onaylanmamış verileri içermediği doğrulanmalıdır.	✓	✓	✓	2.0
16.2	Güvenilmeyen kaynaklardan elde edilen dosya isimleri ve dizin bilgilerinin, dizin gezinme, yerel dosya dahil etme, dosya MIME tipi, işletim sistemi komut enjeksiyonu saldırılarını engellemek için kontrolden geçtiği doğrulanmalıdır.	✓	✓	✓	2.0
16.3	Güvenilmeyen kaynaklardan elde edilen dosyaların, zararlı içerik yüklenmesini engellemek amacıyla anti-virüs sistemleri tarafından taramadan geçirildiği doğrulanmalıdır.	✓	✓	✓	2.0
16.4	Uzak ve yerel dosya ekleme (Remote/Local File Inclusion) zafiyetlerini engellemek için güvenilmeyen verilerin “inclusion”, “class loader” veya “reflection” gibi yapılarda kullanılmadığı doğrulanmalıdır.	✓	✓	✓	2.0
16.5	Uzaktan içerik eklenmesinin önüne geçebilmek için güvensiz verilerin “cross-domain resource sharing” (CORS) içerisinde kullanılmadığına emin olunmalıdır.	✓	✓	✓	2.0
16.6	Güvenilmeyen kaynaklardan elde edilen dosyaların uygulamanın kök dizini (webroot)’nin dışında bir yerde kısıtlı yetkiler ve güçlü doğrulama ile tutulduğu doğrulanmalıdır.		✓	✓	3.0

#	Tanım	1	2	3	Since
16.7	Uygulama sunucusunun uzak/yabancı kaynaklara veya uygulama sunucusu dışındaki kaynaklara erişiminin varsayılan olarak engellenecek şekilde yapılandırıldığı doğrulanmalıdır.		✓	✓	2.0
16.8	Uygulamanın güvenilmeyen kaynaklarca yüklenen kodları çalıştırmadığı doğrulanmalıdır.	✓	✓	✓	3.0
16.9	Flash, Active-X, Silverlight, NACL, istemci taraflı Java veya W3C browser standartları tarafından desteklenmeyen istemci taraflı teknolojiler kullanılmamalıdır.	✓	✓	✓	2.0

Referanslar

Daha fazla bilgi için:

- File Extension Handling for Sensitive Information: https://www.owasp.org/index.php/Unrestricted_File_Upload

V17: Mobil Doğrulama Gereksinimleri

Kontrol Hedefleri

Bu bölüm, mobil uygulamaya özel denetimler içerir. Bu denetimler 2.0'dan kaldırılmıştır. Bu nedenle ilgili ASVS Doğrulama Seviyesi tüm bölümleri ile birlikte ele alınmalıdır. Mobil uygulamalar şunları yapmalıdır:

- Mobil istemcide bulunan tüm güvenlik denetimleri aynı seviyede olmalıdır.
- Cihazda depolanan hassas bilgi varlıkları güvenli bir şekilde tutulmalıdır.
- Cihazdan iletilen tüm hassas veriler, taşıma katmanı güvenliği göz önünde bulundurularak yapılmalıdır.

Gereksinimler

#	Tanım	1	2	3	Since
17.1	UDID (Unique Device Identifier) ya da IMEI gibi ID değerleri cihaz üzerinde tutulmaktadır ve diğer uygulamalar tarafından erişilebilir durumdadır. Bu gibi değerlerin uygulamada bir doğrulama anahtarı (authentication key) gibi kullanılmadığının kontrolü yapılmalıdır.	✓	✓	✓	2.0
17.2	Mobil uygulamanın, hassas verileri diğer uygulamalarında erişebileceği ve şifrelenmemiş (unencrypted) alanlara saklamadığının kontrolü yapılmalıdır. (Örneğin; SD Card, Shared folders..)	✓	✓	✓	2.0
17.3	Hassas verilerin korumasız bir şekilde depolanmadığından emin olunmalıdır. Bu kontrol, sistem korumalı alanlarda da gerçekleştirilmelidir. 'Key chains' gibi...	✓	✓	✓	2.0
17.4	Gizli anahtarlar, API anahtarları ya da parolaların uygulama içerisinde dinamik olarak oluşturulduğu doğrulanmalıdır.	✓	✓	✓	2.0
17.5	Mobil uygulamanın, hassas bilgilerin sızmasına karşı önlem aldığının doğrulaması yapılmalıdır. Örneğin; uygulama arka plana atıldığında ekran görüntüsü (screenshot) kaydedilmektedir. O anda uygulama hassas bilgiler içerebilir. Bir başka örnek; uygulama hassas bilgileri konsol çıktısı olarak yazmamalıdır. Buna karşı uygulama önlem almalıdır.		✓	✓	2.0

#	Tanım	1	2	3	Since
17.6	Uygulamanın gerekli işlemleri yapabilmesi ve kaynakları kullanabilmesi için minimum düzeyde izin istediğinin kontrolü yapılmalıdır. Gereksiz izinler istenmemelidir.		✓	✓	2.0
17.7	Uygulamanın ASLR (Address Space Layout Randomization) yöntemini/korumasını kullandığı doğrulanmalıdır.	✓	✓	✓	2.0
17.8	Uygulamanın, saldırganların uygulamayı debug etmesini engellemeye yönelik anti-debugging yöntemleri kullandığı kontrol edilmelidir. Örneğin; GDB gibi bir araçla uygulamanın debug edilmesine karşı önlem alınmalıdır.			✓	2.0
17.9	Uygulamanın hassas aktiviteleri, intent verilerini ya da içerik sağlayıcılarını (content providers) diğer uygulamaların kullanımına açmadığı kontrol edilmelidir.	✓	✓	✓	2.0
17.10	Uygulamanın, hafızada bulunan hassas verileri artık ihtiyaç duyulmadığında sildiğinden emin olunmalıdır. Artık kullanılmıyorsa üzerine sıfır yazılmalıdır. Bu uygulamayı bellek okuma saldırılarına karşı korur.		✓	✓	3.0.1
17.11	Uygulamanın, aktivitelerden, intent'lerden ya da content provider'lardan dışarı aktarılacak girdileri doğrulandığına emin olunmalıdır.	✓	✓	✓	3.0.1

Referanslar

Daha fazla bilgi için:

- OWASP Mobile Security Project:
  https://www.owasp.org/index.php/OWASP_Mobile_Security_Project
- iOS Developer Cheat Sheet:
  https://www.owasp.org/index.php/IOS_Developer_Cheat_Sheet

V18: Web Servisleri Doğrulama Gereksinimleri

Kontrol Hedefleri

RESTful veya SOAP tabanlı web servislerini kullanan doğrulanmış bir uygulamanın aşağıdakileri yaptığından emin olun:

- Tüm web servisleri uygun yetkilendirme, kimlik doğrulama ve oturum yönetimine sahip olmalıdır.
- Tüm girdilerin kontrolü ve doğrulaması yapılmalıdır.
- SOAP web servisleri katmanı, API kullanımında temel olarak çalışmalıdır.

Gereksinimler

#	Tanım	1	2	3	Since
18.1	İstemci ve sunucu arasında aynı kodlama(encoding) metodunun kullanıldığı doğrulanmalıdır.	✓	✓	✓	3.0
18.2	Web servis uygulaması içindeki yönetim alanının ve yönetim fonksiyonlarının yalnızca servis yöneticileri tarafından erişilebildiğinin doğrulaması yapılmalıdır.	✓	✓	✓	3.0
18.3	XML veya JSON şemalarına girdinin alınmadan önce doğrulandığından emin olunmalıdır.	✓	✓	✓	3.0
18.4	Tüm girdi alanları için uygun uzunluk limitleri olduğu doğrulanmalıdır.	✓	✓	✓	3.0
18.5	SOAP tabanlı web servislerinin en azından (WS-I) Basic Profile'e uyumlu olduğu doğrulanmalıdır. Bu aslında TLS şifreleme anlamına gelmektedir.	✓	✓	✓	3.0.1
18.6	Oturum tabanlı kimlik doğrulama ve yetkilendirme kullanıldığını doğrulayın. Daha fazla bilgi için lütfen bölüm 2,3 ve 4'e bakın. Statik API anahtarı ve benzerlerini kullanmaktan kaçının.	✓	✓	✓	3.0
18.7	REST 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.	✓	✓	✓	3.0.1

#	Tanım	1	2	3	Since
18.8	REST servis için beklenen verinin (xml veya application/json gibi) açıkça tanımlandığından emin olunmalıdır.		✓	✓	3.0
18.9	İstemci ve servis arasında güvenilir bir iletişim sağlamak için mesaj içeriğinin (message payload) imzalandığından emin olunmalıdır. SOAP istekleri için JSON Web Signing ya da WS-Security kullanılabilir.		✓	✓	3.0.1
18.10	Alternatif ve daha az güvenli erişim yollarının var olmadığından emin olunmalıdır.		✓	✓	3.0

Referanslar

Daha fazla bilgi için:

- OWASP Testing Guide 4.0: Configuration and Deployment Management Testing https://www.owasp.org/index.php/Testing_for_configuration_management
- OWASP Cross-Site Request Forgery cheat sheet https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)_Prevention_Cheat_Sheet
- JSON Web Tokens (and Signing) https://jwt.io/

V19. Konfigürasyon Doğrulama Gereksinimleri

Kontrol Hedefleri

Doğrulanmış bir uygulamanın şunları yaptığından emin olun:

- Kütüphaneler ve platformlar günceldir.
- Varsayılan olarak güvenli bir yapılandırma vardır.
- Kullanıcının varsayılan yapılandırmada değişiklikler yaparak gereksiz yere altta yatan sistemlerde güvenlik zayıflıkları başlatmaması için yeterli derecede sıkılaştırma işlemleri yapılmıştır.

Gereksinimler

#	Tanım	1	2	3	Since
19.1	Tüm bileşenler ve güvenlik konfigürasyonları güncel sürümde olmalıdır. Kullanılmayan örnek uygulamalar, platform dokümanları, varsayılan (default) konfigürasyonlar veya test kullanıcıları sistem üzerinden silinmelidir.	✓	✓	✓	3.0
19.2	Özellikle sunucular farklı sistemlerde ise uygulama sunucusu ve veri tabanı sunucusu arasındaki iletişim şifreli olmalıdır.		✓	✓	3.0
19.3	Uygulama sunucusu ve veri tabanı sunucusu gibi sunucuların iletişimini sağlayacak kullanıcının yetkilerinin (ihtiyaç duyduğu kadar) en alt seviyede olduğu doğrulanmalıdır.		✓	✓	3.0
19.4	Uygulamaların yayınlanmalarının (deployment) yeterince korunaklı olduğundan emin olunmalıdır. Saldırganların diğer uygulamalara saldırmasını engellemek için bileşenler farklı sistemlerde veya konteynerlerde (containers) bulunmalıdır.		✓	✓	3.0
19.5	Uygulama derlenme (build) ve yayınlanma (deployment) süreçlerinin güvenli bir şekilde gerçekleştirildiğinden emin olunmalıdır.		✓	✓	3.0
19.6	Yetkili yöneticilerinin, güvenlik konfigürasyonlarının değiştirilmesi/manipule edilmesi (tampering) gibi durumları denetleyebildiklerine emin olunmalıdır.			✓	3.0

#	Tanım	1	2	3	Since
19.7	Tüm uygulama bileşenlerinin imzalanmış/onaylanmış olduğu doğrulanmalıdır.			✓	3.0
19.8	Üçüncü parti uygulamaların güvenilir kaynaklardan geldiği doğrulanmalıdır.			✓	3.0
19.9	ASLR, DEP ve güvenlik denetimleri gibi sistem düzeyinde diller için oluşturulan işlemlerde tüm güvenlik bayraklarının (flags) aktif olduğundan emin olunmalıdır.			✓	3.0
19.10	Tüm uygulama kaynaklarının yine uygulama tarafından barındırıldığından emin olunmalıdır. Örneğin; JavaScript kütüphaneleri, CSS stylesheets ve web fontları CDN veya dış kaynaktan sağlamak yerine uygulamanın kendi içerisinde barındırılmalıdır.			✓	3.0.1

Referanslar

Daha fazla bilgi için:

- OWASP Testing Guide 4.0: Configuration and Deployment Management Testing https://www.owasp.org/index.php/Testing_for_configuration_management

Ek B: Sözlük

- **Erişim Kontrolü** – Kullanıcıların içinde bulundukları kimlik veya gruba göre dosyalara, URL'lere veya diğer kaynaklara erişiminin sınırlandırılmasıdır.
- **Adres Alanı Düzeni Rastgeleleştirme (ASLR)** – Ara bellek taşması saldırlarına karşı koruma sağlayan bir teknik.
- **Uygulama Güvenliği** – Open Systems Interconnection Reference Model(OSI Modeli)'nin uygulama seviyesinde yer alan ve uygulama seviyesindeki güvenlik açıklarına odaklı olarak uygulamanın bileşenlerinin analizidir. Uygulama güvenliği, işletim sistemi veya bağlı olunan ağları odak noktası olarak almaz.
- **Uygulama Güvenliği Doğrulama** – Bir uygulamanın OWASP ASVS baz alınarak teknik olarak incelenmesidir.
- **Uygulama Güvenliği Doğrulama Raporu** – Bir uygulama için yapılan doğrulama ile ilgili genel sonuçları ve destekleyici analizleri içeren dokümandır.
- **Kimlik Doğrulama** – Sunulan kullanıcı kimliğinin doğrulanmasıdır.
- **Otomatize Doğrulama** – Zafiyet imzalarını kullanarak sorunları tespit eden otomatize araçların kullanılmasıdır. (dinamik, statik ya da ikisinin birlikte kullanıldığı araçlar)
- **Arka Kapı (Back Doors)** – Bir uygulamaya yetkisiz giriş sağlayan zararlı kod parçası.
- **Kara Liste (Blacklist)** – İzin verilmeyen bir liste halindeki veri ya da operasyonlar.
- **Basamaklı Stil Şablonu (Cascading Style Sheets)** (CSS) – HTML gibi biçimleme dillerinde yazılan dokümanlarda, sunum işaretlerini tanımlamaya yarayan stil sayfalarıdır.
- **Sertifika Otoritesi (Certificate Authority)** (CA) – Dijital sertifikalar dağıtan kurum.
- **İletişim Güvenliği** – Uygulamaya ait verilerin, uygulama bileşenleri, istemci-sunucu ve dış sistemlerle uygulama arasında iletişimi sırasında korunması.
- **Bileşen (Component)** – Disk ve ağ arabirimleriyle ilişkili, diğer bileşenlerle iletişim kuran bağımsız bir kod birimi.
- **Siteler Arası Betik Çalıştırma** (XSS) – Tipik olarak web uygulamalarında görülen ve istemci tarafındaki içeriğe müdahale edilmesine izin veren bir zafiyet.

- **Kriptografik Modül (Cryptographic modüle)** – Şifreleme algoritmaları ya da şifreleme anahtarları üreten donanım veya yazılım.
- **Servis Dışı Bırakma Saldırıları (DoS)** – Bir uygulamanın kaldırabileceğinden daha fazla istek ile servis veremez duruma getirilmesidir.
- **Dizayn Doğrulama (Design Verification)** – Bir uygulamanın güvenlik mimarisinin teknik olarak değerlendirilmesi
- **Dinamik Doğrulama** – Zafiyet imzalarını kullanarak zafiyetlerini tespit eden otomatik araçlar yoluyla yapılan doğrulama.
- **Easter Eggs** – Özel bir olay veya kullanıcı tarafından tetiklenen ve bu tetikleme anına kadar çalışmayan zararlı kod parçası.
- **Harici Sistemler** – Uygulamanın bir parçası olmayan sunucu taraflı uygulama ya da servis.
- **FIPS 140-2** – Şifreleme modüllerinin tasarımı ve uygulanması sırasında temel alınabilecek bir standart.
- **Global Tekil Tanımlayıcı (Globally Unique Identifier)** (GUID) – Bir yazılımda tanımlayıcı olarak kullanılan benzersiz referans numarası.
- **Hiper Metin Biçim Dili (HTML)** - Tarayıcıda gösterilen web sayfaları ve diğer verilerin yaratılması için kullanılan ana biçim dilidir.
- **Hiper Metin Transfer Protokolü** (HTTP) – 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.
- **Girdi Doğrulama**– Güvenilmeyen kullanıcı girdilerinin doğrulanması ve standartlaştırılmasıdır (canonicalization).
- **Basit Dizin Erişim Protokolü (LDAP)** – Bir ağdaki dağınık dosya bilgi servislerine erişmek ve idame ettirmek için kullanılan uygulama protokolü.
- **Zararlı Kod (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.

- **Zararlı Yazılım (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/
- **Çıktı Doğrulama** – Tarayıcı ya da harici sistemlere gönderilen uygulama çıktısının doğrulanması ve standartlaştırılmasıdır.
- **Kişisel Tanıtıcı Bilgiler** (PII) - Tek bir kişiyi tanımlamak, bağlantı kurmak veya yerini belirlemek için kullanılabilen bilgilerdir.
- **Pozitif doğrulama** – Bkz. whitelist.
- **Güvenlik Mimarisi**– Uygulama tasarımında güvenlik kontrollerinin nerde ve nasıl kullanıldığını tanımlayan bir soyutlamadır.
- **Güvenlik Yapılandırması** – Güvenlik kontrollerinin nasıl kullanılacağını etkileyen işleyiş zamanı uygulama yapılandırmasıdır.
- **Güvenlik Kontrolü** – Güvenlik kontrolü yapan bir fonksiyon ya da bileşen. (Ör: Erişim yetkisi kontrolü)
- **SQL Enjeksiyonu (SQLi)** – Veri tabanlı uygulamalarda, bir girdi noktasına zararlı SQL cümleciklerinin yerleştirilmesi ile gerçekleştirilen bir kod yerleştirme saldırısıdır.
- **Statik Doğrulama** – Uygulama kaynak kodundaki sorunları zafiyetlerin imzalarını kullanarak tespit etmeye çalışan otomatik araçların kullanılması.
- **Doğrulama Hedefi (TOV)** – Eğer OWASP ASVS gereksinimlerine göre bir uygulama doğrulaması gerçekleştiriyorsanız, doğrulama belli bir uygulama üzerinde olacaktır. Bu da kısaca “Doğrulama Hedefi” (Target of Verification) olarak anılmaktadır.
- **Tehdit Modelleme** - Önemli teknik iş varlıklarını, güvenlik alanlarını ve tehdit ajanlarını belirlemek için iyileştirilmiş güvenlik mimarilerinin sürekli geliştirilmesidir.
- **İletişim Katmanı Güvenliği**– İnternet üzerindeki iletişimde güvenliğin sağlanması için uygulanan şifreleme protokolleridir.

- **URI/URL** – 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.
- **Kullanıcı Kabul Testi (UAT)**– Canlıya çıkmadan önce tüm yazılımların canlı ortamdaki gibi davranan bir ortamda test edilmesi.
- **Doğrulayıcı**- Uygulamayı OWASP ASVS gereksinimlerine göre gözden geçiren takım ya da kişidir.
- **Beyaz Liste (Whitelist)** – İzin verilen veri ya da operasyonların listesidir. Ör: girdi doğrulamada kabul edilen karakterler listesi.
- **Genişletilebilir İşaretleme Dili (XML)** – Dokümanların kodlanması(encoding) için bir dizi kurallar sunan işaretleme dili.

Ek C: Referanslar

Aşağıdaki OWASP projeleri, bu standardın kullanıcıları için faydalı olabilir:

- OWASP Testing Guide
  https://www.owasp.org/index.php/OWASP_Testing_Project
- OWASP Code Review Guide
  http://www.owasp.org/index.php/Category:OWASP_Code_Review_Project
- OWASP Cheat Sheets
  https://www.owasp.org/index.php/OWASP_Cheat_Sheet_Series
- OWASP Proactive Controls
  https://www.owasp.org/index.php/OWASP_Proactive_Controls
- OWASP Top 10
  https://www.owasp.org/index.php/Top_10_2013-Top_10
- OWASP Mobile Top 10
  https://www.owasp.org/index.php/Projects/OWASP_Mobile_Security_Project_-_Top_Ten_Mobile_Risks

Benzer şekilde aşağıdaki web sayfaları, bu standardın kullanıcıları için faydalı olabilir:

- MITRE Common Weakness Enumeration - http://cwe.mitre.org/
- PCI Security Standards Council - https://www.pcisecuritystandards.org
- PCI Data Security Standard (DSS) v3.0 Requirements and Security Assessment Procedures https://www.pcisecuritystandards.org/documents/PCI_DSS_v3.pdf