1. AMAÇ

Kıbrıs Aydın Üniversitesinin ihtiyacı olan akademik/ idari personelin, öğrenci, aday öğrenci ya da dış kuruluşlardan gelen ya da eksikliği tespit edilen yazılım / program taleplerini mevcut imkânlar çerçevesinde karşılamak, çözüm üretmek amacı ile Bilgi İşlem Daire Başkanlığınca geliştirilecek/üretilecek olan yazılımlar için izlenecek yolu ve kuralları belirler.

2. KAPSAM

Bu prosedür, Bilgi İşlem Daire Başkanlığınca geliştirilecek/üretilecek olan yazılımlar için uygulanır.

3. UYGULAMA

3.1Yazılım Taleplerinin Alınması / Tespit Edilmesi ve değerlendirilmesi

3.1.1 Öğrencilerin, idari/akademik personelin yazılım geliştirme ihtiyaçları, sorumlu olduğumuz kurum ve kuruluşların yönetmelik, yönerge, resmi yazı ile gelen değişiklikler doğrultusunda BİDB ilgili birim ve üst yönetimle değerlendirilip yapılacak olan değişiklikler ve güncellemelere karar verilip, takvim oluşturulur, UBIS timeline sisteminde kayıtlar takip edilir.

3.1.2 BİDB tarafından gelişen yazılım ve güvenlik teknolojisi doğrultusunda veya daha kullanıcı dostu yazılım geliştirme doğrultusunda gerekli gördüğü güncellemelere karar verip güncellemeler için takvim oluşturulur.

3.2 Yazılım Gereksinimin Yerine Getirilmesi

3.2.1 Güvenli Kodlama İlkelerinde belirtilmiş olan koşullara uyarak yazılımın geliştirilmesi,

3.2.2 Karar verilen ve takvimi oluşturulan yazılım, UBIS programı içerisinde bulunan Timeline uygulamasına ilgili süreç oluşturulur ve uygulamayı geliştirecek olan yazılımcının/yazılımcıların üzerine atanır.

3.2.3 Uygulama geliştirmeleri, ürün ortamında farklı bir ortamda gerçekleştirilir. Bu geliştirme ortamına, yazılım geliştiriciler sadece SVN yöntemini kullanarak, kendileri için özel yetkilendirilmiş olan alanlarda güvenlikleri test edilmiş framework de uygulama geliştirirler.

3.2.4 Yazılımcı uygulamayı geliştirmesinin ardından test görevi verilen farklı bir yazılımcı tarafından test eder.

3.2.5 Eğer bir sorun ile karşılaşılmış ise hatalar ve sorunlar listelenerek uygulamayı geliştiren yazılımcıya iletilir ve bir önceki adımdan süreç devam eder.

3.2.6 Bir sorun ile karşılaşılmamış ise kullanımına sunulacak birimden birkaç personele test yetkisi verilerek, bu personeller tarafından test edilir.

3.2.7 Eğer bir sorun ile karşılaşıldı ise hatalar listelenerek uygulamayı geliştiren yazılımcıya iletilir ve ilgili adımdan süreç devam eder.

3.2.8 Bir sorun ile karşılaşılması ise yazılım devreye alınır ve yazılımın geliştirilmesi doğrultusunda timeline uygulaması üzerinde ilgili süreç kapatılır.

3.3 Güvenli Kodlama İlkeleri

3.3.1 Yazılım geliştirme ortamı ile veri tabanı ortamı birbirinden bağımsız alanlardır. Bu alanlar ile ilgili kritik bilgiler (sunucu adı ve sürümü, kullanılan program sürümü vb.) gizlenir.

3.3.2 Yazılım geliştirme ortamı sadece yazılım geliştiricilerin SVN yöntemi dışında erişimin engelleneceği şekilde, farklı bir virtual host alanında tanımlanıp, erişim fiziksel firewalllar ve switch aracılığı ile engellenir.

3.3.3 Veri tabanı ortamı sadece veri tabanı yöneticilerinin erişimi sağlanacak şekilde, firewalllar ve switch aracılığı ile yönetilir.

3.3.4 Yazılım geliştirme ortamına sadece SVN yöntemi ile kod gönderilir. Bu yöntem ile yazılım kütüphanesi versiyonu tutulmuş olur.

3.3.5 Her bir yazılım geliştiricisi için, ayrı ayrı kullanıcı adı, şifre ve yetkilendirme tanımlanır.

3.3.6 Uygulamayı kullanan, kullanıcıların girmiş oldukları girdiler; temizleme, filtreleme ve SQL Enjeksiyonu kontrol metotlarının kullanımına dikkat edilerek kodlanır.

3.3.7 Uygulamaya yapılan tüm erişim istekleri hem istek hem de yanıt zamanında yetkilendirmeye tabi tutulur.

3.3.8 Kullanıcının, uygulama ortamından talep ettiği her istek, iz kaydı ile saklanır ve yetki kontrolü yapılarak cevap verilir.

3.3.9 Uygulama çatısı, veri tabanı, uygulama sunucusu ve web sunucusu gibi kullanılan yazılımların güvenlik taramaları en üst seviyededir.

3.3.10 Uygulama kullanılırken meydana gelen her başarısız işlemin kaydı alınarak, uygulama geliştiricilere bildirilir.

3.3.11 Ön tanımlı kullanıcı hesapları sistemden, veri tabanından ve uygulamadan kaldırılır.

3.3.12 Hassas bilgiler içeren web sayfalarının tarayıcılarda belleğe alınmaması için autocomplete,

cache-control, pragma gibi gerekli HTTP/HTML başlıkları kullanılır.

3.3.13 Güvenli web trafiği için (SSL) güçlü şifreleme algoritmaları kullanılır.

3.3.14 Tüm parola alanlarında kullanıcı giriş yaparken kullanıcının parolası maskelenir ve açık olarak görüntülenir.

3.3.15 Zayıf parolaların kullanımına izin verilmez.

3.3.16 Uygulamada HTTPS protokolü kullanılır.

3.3.17 Herkese açık olmayan bütün kaynaklara ve sayfalara erişim için sunucu tarafında kimlik doğrulaması yapılır.

3.3.18 Kullanıcı şifreleri şifrelenmiş olarak saklanır. Kontrol hash verisi oluşturulurken salt veri de kullanılır.

3.3.19 Kullanıcılara verilen şifrelerden sonra ilk kullanımda parola değiştirmeye zorlanılır. Parolalar, en az bir büyük, bir küçük harf, bir rakam ve birde özel karakter içermek koşulu ile en az 8 karakter olmalıdır.

3.3.20 Belirli sayıda hatalı giriş yapıldığında hesap bloke edilir.Hatalı şifre ya da kullanıcı adı girildiğinde, sosyal hackerlığı engellemek için tek tip hata mesajı verilerek hatanın kaynağı gizlenir.

3.3.21 Parolaların geçerlilik süresi tanımlanır. Bu süre 90 gündür.

3.3.22 Parola değişiminde eski parola istenir.

3.3.23 Kullanıcılar için her oturum açtıklarında; kendilerine özel, süresi olan çerezler ve session oluşturulur, oturum kapatıldığında, o oturuma ait olan çerezler ve sessionlar sonlandırılır.

3.3.24 Yetkilendirme yaparken “Rol bazlı” yetkilendirmeler yapılır.

3.3.25 Kullanıcı yetkileri, sadece sistem yöneticisi veya yetkilendirilmiş kişiler tarafından yapılır.

3.3.26 Saldırı anında, CAPTCHA vb. güvenlik önlemleri aktif hale getirilir.