Yazılım Güvenliğinde İnsiyatif Alın

Bedirhan Urgun, Ekim 2009, WGT E-Dergi 2. Sayı

Giriş

Son 10 senedir dünya çapında Google, Microsoft, Yahoo gibi kurumsal firmaların güvenli yazılım insiyatifleri almaları bir raslantı değildir. Bu kurumlar, ana problemin sadece yazılımsal zafiyetlerin üzerine giderek çözemeyeceklerini, ürettikleri yazılımların güvenliğini sağlamanın yolunun geliştirme sürecinin bütün noktalarına (ilgili iş süreçleri, yazılım geliştirme safhaları, yasal uyumluluklar, v.b.) dokunmak gerektiği gerçeğinden geçtiğini anladılar. Bu kapsamlı ve doğal olarak uçtan uca güvenlik anlayışı, ancak ve ancak formal ve gelişen bir sürece oturduğu ve tabi ki gerekli desteği aldığı müddetçe başarılı olabilir. 2008 yılında böyle bir sürecin ilk formal tanımları, olgunluk modelleri ile açıklanmaya çalışılmıştır [2, 3].

Olgunluk Modeli Tanımı

Bir organizasyonun herhangi bir konudaki olgunluk seviyesini ve durumunu açıklayan yapısal maddeler bütünüdür [1]. Olgunluk modelleri;

- Bir organizasyonun diğer organizasyonlara göre hangi seviyede olduğunu gösterir.
- Organizasyon içi ve organizasyonlar arası ortak bir dil oluşmasına yardımcı olur.
- Organizasyon içi muhtemel eksikliklerin (gap) bulunmasını ve bu eksikliklerin anlaşılmasını sağlar.
- Alınması gereken aksiyonlarda hedef organizasyona özel önem derecelendirmesini ortaya çıkarır.

Yazılım Güvenliği Olgunluk Modeli

90'lı yılların sonlarına doğru Internet kullanımının yaygınlaşması ve başarılı saldırıların artması ile beraber, yazılım güvenliği öneminin yaygın olarak anlaşılması organizasyonların bir çok alanda bu probleme yönelik aksiyon almalarını gerektirmiştir. Her organizasyonda farklı içeriklerde ve seviyelerde olan bu aksiyonlar geçtiğimiz yıla kadar hiçbir şekilde formal hale getirilmemişti.

Yazılım güvenliğine özel olgunluk modelleri, 2008 yılında iki dokümanın [2, 3] yayınlanması birlikte bir yol haritasına ve tartışma ortamına kavuşmuştur. Yazılım güvenliğine henüz başlamamış, farklı boyutlardaki organizasyonlar bu dokümanlar yardımı ile probleme yaklaşma şansını elde etmişlerdir. Henüz yeni yeni tartışılan ve uygulanan bu modeller ilerde daha da oturacaktır, bu nedenle genel görüş, bir tanesinin seçilip vakit geçirmeden işe başlanılmasıdır [4].

Bu modellerin uygulanması ile beraber organizasyonlar;

- Uygulanan (veya uygulanmayan) yazılım güvenlik süreçleri ile tanımlana modelde nereye oturduklarını öğrenebileceklerdir.

- Uygulanan (veya uygulanmayan) süreçlerde muhtemel eksikliklerini (gap) bulabileceklerdir.
- Bulunan eksikliklere göre alınacak aksiyonların listesini çıkarabilecek ve önemlerine göre bu aksiyonları seviyelendirebileceklerdir.

Yazımızda, aynı zamanda bir OWASP [5] projesi olan SAMM'in [2] bakış açısını inceleyeceğiz. Ayrıca SAMM, diğer yazılım güvenliği olgunluk modeli BSI-MM'den [3] farklı olarak;

- Herhangi bir ticari yapıya bağlı değildir.
- Olgunluk seviyelerini daha detaylı ve küçük organizasyonlar tarafından da uygulanabilir şekilde anlatmaktadır.
- Alınacak aksiyonlara ait adım adım detaylara ve worksheet'lere sahiptir (Bu yardımcı araçlar hali hazırda üretilmektedir).

SAMM (Yazılım Güvence Olgunluk Modeli)

SAMM kullanılarak hali hazırda uygulanmakta olan yazılım güvence süreçleri değerlendirilebilir, bir organizasyon için stratejik bir yol haritası oluşturulabilir, güvenlik aktiviteleri gerçeklenebilir ve hayata geçirilebilir.

SAMM'in sağladığı güvenlik adımları, yazılım geliştirme süreci ile alakalı dört temel iş fonksiyonunun (business function) üzerine oturmaktadır. Bunlar aşağıda sıralanmıştır;

Yönetim (Governance)
Geliştirme (Construction)
Doğrulama (Verification)
İdame (Deployment)

Yukarıdaki her iş fonksiyonu için SAAM üç güvenlik adımı tanımlar. Bu güvenlik adımları, ilgili iş fonksiyonlarının güvencelerini sağlamak için alınması gereken aktiviteleri listeler.

1. Yönetim
    1. Strateji ve Metrikler (Strategy & Metrics)
    2. Politika ve Uyumluluk (Policy & Compliance)
    3. Eğitim ve Kılavuzluk (Education & Guidance)
2. Geliştirme
    1. Tehdit Değerlendirme (Threat Assesment)
    2. Güvenlik Gereksinimleri (Security Requirements)
    3. Güvenli Mimari (Secure Architecture)
3. Doğrulama
    1. Tasarım Denetimi (Design Review)
    2. Kod Denetimi(Code Review)
    3. Güvenlik Testi (Security Testing)
4. İdame
    1. Açıklık Yönetimi (Vulnerability Management)
    2. Sistem Sıkılaştırma (Environment Hardening)
    3. Operasyonel Strateji (Operational Enablement)

En altta SAMM, her bir güvenlik adımının olgunluk seviyelendirmesini birbirini izleyen üç hedef ile tanımlar. Yani ilk seviye ikinci seviyeye göre daha zor ve sofistike hedefleri içerir. Aynı şekilde üçüncü seviye de ikinciye göre daha zor, sofistike ve kapsamlı hedefleri içerir. Bu şekilde organizasyonlar kendi kaynaklarına, analiz ettikleri risklerine göre gerçekleyecekleri seviyeleri seçebilme şansına sahip olmaktadırlar. Bunun yanında SAMM, yeni başlayan organizasyonların hali hazırdaki seviyelerini ölçebilecekleri bir değerlendirme worksheet'i ile beraber gelmektedir. İlgililer bu formu doldurup t0 esnasında hangi seviyede olduklarını ve gerekli gelişim aktivitelerini görebileceklerdir.

Şekil 1, olgunluk modelinin parçalarını hiyerarşik olarak göstermektedir.

Şekil 1: SAMM iş fonksiyonları ve karşılık gelen güvenlik adımları (seviyeler gösterilmemiştir)

Modelde tanımlı güvenlik adımlarının incelenmesi başka ve daha uzun bir yazının konusudur ancak güvenlik adımlarının birbirinden ayrılması bazılarının süreç içerisinde paralel olarak ilerlenmesini sağlar. Şekil 1'e genel bir bakış, güvenli yazılımların sadece teknik perspektifin ağır bastığı (kod analizi, pentestler) bölüm olan **Doğrulama** değil, ancak diğer üç bölümde de yapılacak çalışmalar ile sağlanabileceğini gösterecektir.

Sonuç

Yakın geçmişte yayınlanan olgunluk modelleri [2, 3] ile yazılım güvenliği insiyatifi hem büyük kurumsal şirketlerde hem de daha küçük firmalarda eldeki kaynaklara ve desteğe göre seviye seviye gerçeklenebilir. Bu şekilde ufak adımlar ile başlanabilecek süreç, daha güvenli yazılımların üretilmesinde büyük rol oynayacaktır.

Referanslar

[1] http://en.wikipedia.org/wiki/Capability_Maturity_Model
[2] http://www.opensamm.org/
[3] http://www.bsi-mm.com/
[4] http://www.opensamm.org/2009/06/jeremy-epstein-on-the-value-of-a-maturity-model/
[5] http://www.owasp.org